

Fran Gil

Front-End Development I

USO EXCLUSIVO PARA:

Nombre: Diosbel Mezquia

Cédula de identidad (CI):

1758039927

Email: dmezquiam@gmail.com

DATOS DE DESCARGA:

Fuente: www.forcontu.com

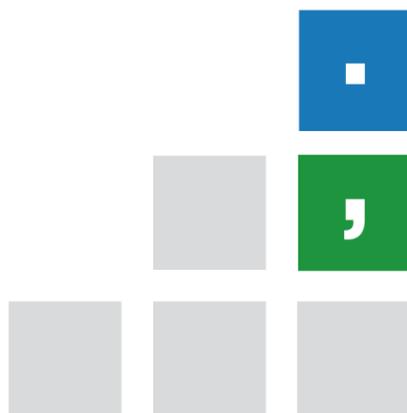
Fecha: 15/12/20 17:26

IP: 191.99.3.224

Código de verificación:

D8FD1PDF00063374008107

Experto en
DRUPAL 8



Versión 1.0 - 25/10/2018

www.forcontu.com

Experto en Drupal 8 Front-End Development I

Fran Gil

Versión: 1.0 (25/10/2018)

Experto en Drupal 8 Front-End Development I

Colección Aprende Drupal con Forcontu

Copyright © 2011-2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este libro, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

ISBN-13 (Edición impresa): 978-84-945012-6-5

ISBN-13 (Edición electrónica, PDF): 978-84-945012-7-2

Autor: Fran Gil

Diseño de portada: Ateigh Design

Directora de Marketing y Distribución: Laura M^a Fornié Alonso

Editado por Forcontu S.L.

Impreso por Liberis Site S.L.

Aunque esta publicación ha sido desarrollada con gran cuidado, ni Forcontu S.L., ni los autores serán responsables de los posibles errores y las consecuencias que pueda causar, directa o indirectamente, la información contenida en este trabajo. Por favor, dirija cualquier recomendación o comentario a info@forcontu.com.

Ayúdenos a mejorar los libros reportando cualquier error que encuentre. Puede hacerlo directamente completando este formulario: <https://www.forcontu.com/libros/reportar-error>

Para más información sobre el contenido de este libro o sobre los canales de distribución, escriba directamente a info@forcontu.com o visite la página web www.forcontu.com.

USO EXCLUSIVO PARA:

Nombre: Diosbel Mezquia

Cédula de identidad (CI):

1758039927

Email: dmezquiam@gmail.com

DATOS DE DESCARGA:

Fuente: www.forcontu.com

Fecha: 15/12/20 17:26

IP: 191.99.3.224

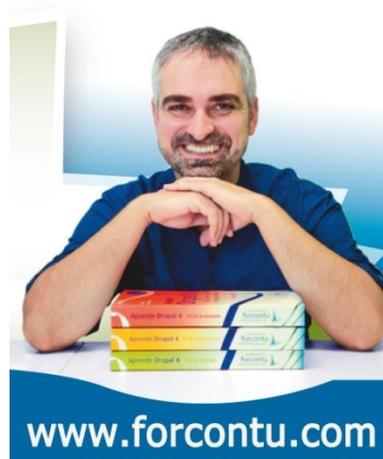
Código de verificación: D8FD1PDF00063374008107

Presentación

Estoy seguro de que si has llegado hasta aquí es porque deseas conocer Drupal a fondo. No importa si sólo has oído hablar de sus numerosas ventajas con respecto a otros CMS, o si ya lo conoces y quieres seguir profundizando en su estudio.

El gran reto al que te enfrentas es, sin duda, **superar la pronunciada curva de aprendizaje de Drupal**. No, no se trata de un mito, llegar a dominar Drupal es una meta realmente costosa en términos de esfuerzo y dedicación.

Esta meta se complica si lo intentas por tu cuenta, sin una guía didáctica más allá de las necesidades específicas de tu primer proyecto. Te verás navegando entre miles de recursos disponibles en Internet (vídeos, entradas de blog, documentación de módulos, etc.) y consultando en foros o chats buscando solucionar con urgencia problemas particulares. Aún así avanzarás en tu proyecto y en tu conocimiento de Drupal, pero los vacíos de conocimiento que se van generando harán que la duda te perseguirá permanentemente con cuestiones como: **¿lo estoy haciendo correctamente?, ¿existirá un módulo mejor para implementar esta funcionalidad?, ¿si utilizo este módulo tendré problemas más adelante para integrar otros módulos?**, etc. Y eso sin hablar del tiempo que te llevará convertirte en un experto en Drupal por ese camino, que puede superar perfectamente los 3 años suponiendo una dedicación exclusiva al desarrollo de sitios web con Drupal.



No te voy a engañar, con nuestros planes formativos **no te ahorrarás ni el esfuerzo ni la dedicación**, pero **te acompañaremos y guiaremos en el aprendizaje de Drupal** para que lo domines en el menor tiempo posible y puedas **desarrollar tus proyectos de forma óptima, productiva y con garantías de éxito**.

Sabrás que te has convertido en un experto en Drupal cuando puedas **desarrollar un proyecto sintiéndote seguro y cómodo**, y eso se consigue con la base necesaria para no perder tiempo en lo cotidiano y focalizar así tus esfuerzos en afrontar los nuevos retos que vayan surgiendo. Porque sí, **Drupal es un entorno en continua evolución** y siempre tendrás nuevos retos a los que enfrentarte, nuevas cosas que aprender y nuevos módulos que probar. Siempre tendrás dudas que consultar, pero notarás que el nivel de tus consultas va subiendo junto con tu grado de experiencia.

Desde 2009 han sido muchas las personas que han aprendido Drupal con los libros y cursos de Forcontu, primero de Drupal 6 y luego de Drupal 7, y que después de todo este tiempo han seguido desarrollando sus proyectos con Drupal y comparten con nosotros su progreso y, como no, su ilusión por dar el salto a Drupal 8 con nosotros.

Ahora comienza, para todos, una nueva etapa con un **Drupal 8 más profesional y más complejo**. Drupal 8 ha dado un importante salto evolutivo con respecto a Drupal 7, por lo que la inversión en adaptación y reciclaje se hace necesaria incluso si llevas años trabajando con Drupal.

En Forcontu estamos desarrollando **unos planes formativos para Drupal 8 mucho más completos** que los anteriores, que te permitan aprender tanto Drupal como otras tecnologías que son necesarias para la implementación de sitios web modernos. De hecho, para esta nueva versión hemos separado los libros y cursos en tres ramas o perfiles profesionales: **Site Building, Back-End Development y Front-End Development**. De esta forma podrás especializarte exclusivamente en uno de los perfiles y obtener tu certificado de **Experto en Drupal 8 (Site Building, Back-End Development o Front-End Development)**, o atreverte con las tres disciplinas y obtener el certificado de **Máster en Drupal 8**, que te certifica como desarrollador Full-Stack, que es el perfil laboral relacionado con Drupal más demandado.

Desde 2012 nuestros libros y cursos están disponibles en español e inglés, y los certificados también se emiten en ambos idiomas para que puedas ampliar tus opciones laborales.

Si ya te has comprometido con Drupal, permítenos acompañarte en esta nueva etapa de tu carrera profesional y guiarte para lograr tu objetivo de convertirte en un Experto en Drupal 8.

Fran Gil
Co-fundador y CEO Forcontu
fran.gil@forcontu.com
@frankgil76

0

Curso Online Experto en Drupal 8 Front-End Development

0.1

Descripción del curso

Qué es Front-End Development

El desarrollo Front-End en Drupal 8 abarca todas las tecnologías relacionadas con la interfaz, que se ejecutan en el navegador web. Podemos agrupar estas tecnologías en tres lenguajes, HTML, CSS y JavaScript. Aunque estas tecnologías se utilizan de base, veremos que existen otras soluciones más avanzadas que facilitan la programación Front-End. Por ejemplo, usaremos SASS para la generación de CSS y librerías JavaScript como jQuery, Angular, React y Vue.

En la primera parte del curso abordaremos temas relacionados con la implementación de temas y la presentación o maquetación de contenidos. En la segunda parte nos adentraremos en la programación JavaScript, haciendo una introducción a las librerías JavaScript más utilizadas.

Modalidad de impartición

El curso de **Experto en Drupal 8 Front-End Development** es un curso **100% práctico y tutorizado**, en modalidad **Online** (Teleformación).

Los participantes dispondrán, además del aula virtual, de un alojamiento web para realizar las prácticas propuestas en el curso. El alumno podrá estudiar y practicar en cualquier momento del día, adaptando el temario y las actividades a sus necesidades profesionales o particulares.

Duración del curso

El curso de **Experto en Drupal 8 Front-End Development** tiene una duración estimada de **3 meses y 180 horas certificadas**, con una dedicación semanal de 15 horas. En este período se incluye la realización de las actividades (120 horas) y del Proyecto Final (60 horas).

Para que puedas tener más margen y tomarte algunas semanas de descanso, tendrás **4 meses para completarlo**. Ten en cuenta que éste es el tiempo máximo, pero si le dedicas más horas o tienes experiencia previa en Drupal y/o en programación, seguro que puedes completarlo en menos tiempo.

Además, si al finalizar el período de matrícula no has podido completar el curso, siempre podrás ampliar la matrícula mes a mes, sin perder todo el trabajo realizado y sin que afecte a tu calificación final.

Ampliación de matrícula

Los alumnos que no hayan podido completar el curso en el tiempo establecido, podrán ampliar la matrícula por períodos mensuales, hasta finalizar el curso. Consulta en nuestra web el coste adicional de esta ampliación.

Objetivos del curso

El objetivo principal del curso **Experto en Drupal 8 Front-End Development** es la capacitación del alumno como **desarrollador Front-End** para la implementación de temas e interfaces de usuario modernas.

Para llegar a este objetivo es necesario cumplir estos objetivos secundarios:

- Conocer las principales herramientas y lenguajes de programación utilizados en el lado del cliente o navegador web.
- Conocer la estructura de los temas y plantillas en Drupal 8.
- Conocer las herramientas y lenguajes utilizados para la maquetación de páginas web.
- Conocer las principales librerías JavaScript para la creación de interfaces modernas e interactivas.

Para qué te prepara

A través de este curso aprenderás a maquetar sitios web en Drupal 8 y mejorar las interfaces que permiten la interacción del usuario con el sitio.

La capacitación adquirida te será de gran ayuda para introducirte en el mercado laboral, mejorar en tu puesto de trabajo o incluso emprender un proyecto propio basado en tecnologías web.

A quién va dirigido

El curso de **Experto en Drupal 8 Front-End Development** va dirigido a:

- Desarrolladores con conocimientos de Drupal 8 que desean profundizar en la maquetación de sitios y programación de interfaces.
- Maquetadores o diseñadores que deseen aplicar sus conocimientos en sitios desarrollados en Drupal 8.
- Cualquier persona que, sin conocimientos de programación previos, desea reorientar su carrera profesional hacia la programación y el desarrollo de sitios web con Drupal 8.

Conocimientos previos necesarios

Para acceder al curso de Experto en Drupal 8 Front-End Development no son necesarios conocimientos de programación. Sí son necesarios conocimientos de Drupal 8 a nivel de Site Building, por lo que se recomienda cursar primero el curso de Experto en Drupal 8 Site Building.

El curso incluye una buena base de programación para personas con ningún o pocos conocimientos de programación. Aún así, debes entender que, si no tienes conocimientos previos de programación, se trata de un reto muy importante que requerirá un esfuerzo adicional al de otros compañeros que sí cuenten con una base de programación.

Aunque no se requiere una titulación académica específica, por su orientación profesionalizadora, es recomendado para diseñadores, desarrolladores web y programadores con titulaciones técnicas universitarias o de formación profesional.

Requisitos técnicos

Para realizar el curso es necesario disponer de un ordenador con conexión a internet y navegador web. Durante el curso te iremos indicando las aplicaciones que debes instalar (cliente FTP, entorno de desarrollo, etc.).

Qué incluye el curso

El curso de Experto en Drupal 8 Front-End Development Incluye los siguientes servicios y materiales:

- **Libros.** Existe mucha información sobre Drupal disponible en libros, blogs, foros, etc. En Forcontu hemos desarrollado unos materiales formativos organizados, en español y compuestos de diferentes objetos de aprendizaje para asegurar la capacitación profesional en Drupal. El curso incluye los libros:
 - o **Experto en Drupal 8 Front-End Development I**
 - o **Experto en Drupal 8 Front-End Development II**

La versión en PDF de los libros se irá actualizando periódicamente para ir incorporando los cambios que se vayan produciendo en los contenidos abordados.

- **Aula virtual.** Dentro del aula encontrarás más información sobre los pasos a seguir durante el curso y las actividades a realizar.
- **Alojamiento web** para la realización de las actividades propuestas y del Proyecto Final de experto.
- **Evaluación continua de las actividades** realizadas durante el curso, aportando feedback sobre los posibles errores cometidos. El aula dispone de un sistema de autocorrección que comprobará inmediatamente si has realizado la actividad correctamente. Esto hace que no tengas que depender del tutor para la corrección de las actividades.
- **Pruebas de Evaluación Continua.** Son conjuntos de actividades adicionales para evaluar un bloque de unidades del curso
- **Evaluación del proyecto final** de Experto.
- **Seguimiento del progreso del alumno.** El tutor hará un seguimiento continuo de tu progreso y responderá a las dudas que te surjan en los foros del aula.
- **Certificado de Experto Forcontu en Drupal 8 Front-End Development** tras superar el curso.
- Posibilidad de obtener el **Certificado de Máster Forcontu en Drupal 8** al completar el resto de cursos de Experto.
- **Carta de recomendación**, para los alumnos que hayan superado el curso con una calificación superior al 95%.
- **Incorporación a la bolsa de trabajo**, donde recibirás las ofertas que publiquen las empresas interesadas en desarrolladores con conocimientos de Drupal.

0.2 Contenidos

El curso de **Experto en Drupal 8 Front-End Development** se divide en las siguientes unidades, distribuidas en dos libros:

Libro **Experto en Drupal 8 Front-End Development I:**

- Unidad 1. Preparando el entorno de desarrollo
- Unidad 2. Instalación y configuración de temas
- Unidad 3. Creación de temas
- Unidad 4. Plantillas Twig en Drupal
- Unidad 5. HTML5
- Unidad 6. CSS3
- Unidad 7. Plantillas y estilos en Drupal
- Unidad 8. Diseño responsive
- Unidad 9. Sass

Libro **Experto en Drupal 8 Front-End Development II:**

- Unidad 10. Introducción a PHP
- Unidad 11. Programación PHP en temas
- Unidad 12. Introducción a Javascript
- Unidad 13. jQuery
- Unidad 14. Bootstrap
- Unidad 15. Foundation
- Unidad 16. Angular
- Unidad 17. React
- Unidad 18. Vue.js

0.3 Metodología

Metodología

El curso de **Experto en Drupal 8 Front-End Development** es un curso **100% práctico y tutorizado**, en modalidad **Online-Teleformación**.

Dispondrás de una guía de trabajo con los pasos a seguir: lectura de cada unidad y prácticas a realizar. A través de los foros del aula virtual, el tutor te guiará en tu proceso de aprendizaje, resolviendo las dudas que te vayan surgiendo durante el curso. También podrás consultar las dudas planteadas por el resto de participantes e incluso intervenir ayudando a tus compañeros.

Se te facilitará un alojamiento web individual para realizar las actividades. Nuestros alojamientos están preparados especialmente para alojar sitios web en Drupal 8. Al trabajar en un servidor remoto, el tutor tendrá acceso a tu sitio web y podrá ver tu trabajo y guiarte en el desarrollo de las actividades propuestas.

Cuando completes y envías una actividad, el sistema intentará corregirla automáticamente. Si detecta algún error, te guiará para corregirla. De esta forma podrás tener un feedback de tu trabajo en cualquier momento, sin depender de la intervención del tutor. Aún así, el tutor estará disponible para resolver cualquier duda y para corregir manualmente aquellas actividades que el autocorrector no haya sido capaz de evaluar.

Podrás realizar las prácticas en cualquier momento, adaptando el temario y las actividades a tu ritmo de trabajo, pero siempre teniendo en cuenta la fecha final de finalización del curso.

El curso se completa con un Proyecto Final acorde a cada especialidad. En el curso de Front-End Development, el Proyecto Final consiste en implementar un tema, maquetar un sitio y mejorar las interfaces con funcionalidades JavaScript añadidas. Si ya tienes un proyecto en mente o has desarrollado en proyecto final de los cursos de Site Building o Back-End, podrás utilizarlo como base para tu Proyecto Final de Front-End.

Plataforma educativa

El estudiante, una vez matriculado en el curso, será dado de alta en la plataforma educativa y el aula correspondiente al módulo a cursar.

Los tutores y estudiantes podrán comunicarse libremente a través de los **foros** (de forma pública para todo el grupo), o del **correo interno** (de forma privada, sólo para los destinatarios indicados) habilitados para cada curso. Por defecto se enviará un duplicado de estas comunicaciones al correo electrónico externo facilitado por el estudiante.

Los estudiantes de un mismo grupo podrán contactar entre sí y compartir sus experiencias y resultados con las actividades y materiales propuestos.

El aula se distribuye de la siguiente forma:

- **Foro Guía** (foro del tutor). En este foro sólo pueden publicar los tutores, y servirá para comunicar a todo el grupo cuestiones generales sobre el curso u otras cuestiones que consideren de interés.
- **Foro Aprende** (foro del curso). En este foro podrán publicar tanto tutores como estudiantes, y se utilizará para discutir cuestiones relacionadas con Drupal y el curso. Los estudiantes podrán plantear aquí sus dudas sobre las actividades y otras consultas generales sobre Drupal.

- **Correo interno.** Permite enviar mensajes a otros compañeros del aula, tanto tutores como estudiantes.
- **Participantes.** Listado de participantes del curso, incluyendo tutores y estudiantes.
- **Usuarios en línea.** Muestra un listado con los participantes que están conectados en ese momento.
- **Perfil.** Permite administrar el perfil del usuario: modificar la contraseña de acceso, modificar los datos personales, cambiar la foto, etc.
- **Calificaciones.** Permite consultar las calificaciones obtenidas.

0.4 Evaluación y Certificación

Evaluación continua

La puntuación máxima que se puede obtener en el curso es de **1000 puntos**, y es necesario completar al menos **800 para superarlo** y obtener el certificado.

Los puntos se distribuyen de la siguiente manera:

- **Actividades.** Cada unidad de estudio tendrá 1 o más actividades. Las actividades pueden tener una puntuación variable en función del esfuerzo estimado requerido. Máximo: 700 puntos

Evaluación: Las actividades no se califican con una puntuación intermedia. Si la actividad enviada no es correcta, se devolverá al alumno con las indicaciones necesarias para solucionar los errores y volver a enviarla. El alumno puede enviar una actividad tantas veces como desee, sin penalización en la calificación final. Una vez la actividad sea validada, el alumno obtendrá la puntuación máxima.

- **Pruebas de Evaluación Continua (PEC).** Cada PEC consta de un conjunto de actividades que sirven de repaso y refuerzo sobre lo estudiado en unidades anteriores. Cada actividad de la PEC se evalúa de forma independiente. El total de puntos de cada PEC es de 50 puntos. Durante el curso se realizarán 2 PEC. Máximo: 100 puntos

Evaluación: La puntuación final de cada PEC depende de la puntuación obtenida en las actividades que la componen. Las actividades no se califican con una puntuación intermedia. Al igual que con las actividades generales, el alumno puede enviar una actividad tantas veces como desee, sin penalización en la calificación final.

- **Proyecto Final.** El trabajo final es un proyecto libre que será previamente validado por el tutor. Máximo: 200 puntos.

El Proyecto Final consta de dos partes:

- o **Fase 1. Planteamiento y análisis.** Máximo: 50 puntos.
- o **Fase 2. Desarrollo.** Máximo: 150 puntos

Evaluación: Al término de cada fase, el tutor evaluará el trabajo realizado y emitirá una

calificación. Si el alumno no obtiene la calificación máxima, recibirá el feedback necesario para realizar las correcciones o ampliaciones necesarias. El alumno puede volver a enviar el proyecto para una nueva evaluación.

Forcontu cuenta con un **Sistema de Autocorrección** de actividades que evalúa de forma automáticamente cada actividad, aportando en el momento del envío la calificación y el feedback sobre los aciertos y errores cometidos en caso de que los haya. Dado que la autocorrección se realiza a nivel de programación, es posible que una pequeña desviación en la solución provoque que el sistema dé por no válida una actividad aparentemente correcta. Si el alumno considera que la autocorrección no está validando correctamente la actividad, podrá enviarla al tutor para su corrección manual.

Certificación

Para superar el curso será necesario completar al menos 800 puntos de los 1000 puntos disponibles (80%). Una vez alcanzados los 800 puntos, el alumno puede dar por completado el curso, o continuar hasta alcanzar la calificación máxima.

Una vez completado el curso, y siempre que se haya alcanzado los 800 puntos, se emitirá el **certificado de Experto Forcontu en Drupal 8 Front-End Development**.

El certificado incluye, entre otros datos, el número de horas de trabajo, el contenido detallado del curso y la calificación final obtenida.

Índice de contenidos

Presentación	v
Curso Online Experto en Drupal 8 Front-End Development	vii
Descripción del curso.....	vii
Contenidos.....	x
Metodología.....	xi
Evaluación y Certificación.....	xii
Índice de contenidos	xv
Unidad 1. Preparando el entorno de desarrollo	3
1.1 Qué es un IDE.....	4
1.2 Instalación y configuración de NetBeans.....	6
1.3 Instalación de Drupal en el servidor de Forcontu.....	22
Unidad 2. Instalación y configuración de temas	29
2.1 Qué son los temas.....	30
2.2 Repositorios de temas.....	31
2.3 Instalación de temas.....	35
2.4 Configuración del tema.....	41
Caso práctico 2.4a Instalación y configuración del tema Mayo.....	44
2.5 Desinstalar un tema.....	54
2.6 Tema de administración.....	56
Unidad 3. Creación de temas	61
3.1 Estructura de un tema.....	62
3.2 Creación de temas.....	67
3.3 Modificación de temas.....	72
3.4 Regiones personalizadas.....	78
3.5 Creación de temas con Drupal Console.....	82
Unidad 4. Plantillas Twig en Drupal	85
4.1 Qué son las plantillas.....	86
4.2 Estructura de plantillas de Drupal.....	89
4.3 Depuración de plantillas.....	96
4.4 Sugerencias de plantillas (template suggestions).....	102
4.5 Twig en Drupal.....	105
Unidad 5. HTML5	123
5.1 Estructura de un documento HTML.....	124
5.2 Etiquetas HTML en el encabezado.....	129
5.3 Estructura del cuerpo.....	134
5.4 Otras etiquetas HTML.....	138
5.5 Formularios HTML.....	143
5.6 HTML en Drupal.....	155
Unidad 6. CSS3	159
6.1 Introducción a CSS.....	160
6.2 Sintaxis de CSS.....	163
6.3 Colores y fondos.....	167
6.4 Estilos de fuentes.....	174
6.5 Agrupación y posicionamiento de elementos.....	179
6.6 Selección avanzada de elementos.....	197
6.7 Estilos de formularios.....	208
6.8 Recursos adicionales.....	213

Unidad 7. Plantillas y estilos en Drupal	217
7.1 Estándares de codificación CSS en Drupal	218
7.2 Librerías CSS en Drupal	229
7.3 Clases CSS desde la interfaz	239
7.4 Plantillas en módulos.....	252
7.5 Módulos adicionales: Advanced CSS/JS Aggregation.....	263
7.6 Módulos adicionales: Twig Vardumper	266
7.7 Módulos adicionales: Twig Tweak.....	268
Unidad 8. Diseño responsive	277
8.1 Introducción al diseño responsive.....	278
8.2 Creación de un tema responsive.....	280
8.3 Layouts responsive en CSS	284
8.4 Media Queries	308
8.5 Breakpoints en Drupal	313
8.6 Imágenes responsive	319
Unidad 9. Sass	327
9.1 Introducción a Sass.....	328
9.2 Sintaxis de Sass (SCSS).....	333
9.3 Instalación y configuración de Sass	348
9.4 Archivos de mapeo en Sass	352
9.5 Compass	355
9.6 Herramientas de automatización: Grunt	367
9.7 Herramientas de automatización: Gulp.....	373
9.8 Configuración de Sass en AdaptiveTheme	380
Diario de cambios (Change log)	385

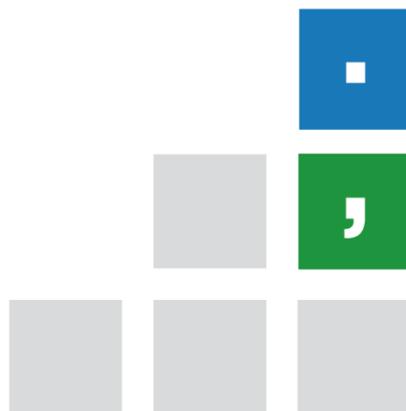
Fran Gil

Front-End Development I



Preparando el entorno de desarrollo

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

1 Preparando el entorno de desarrollo

En esta unidad veremos cómo preparar el entorno de desarrollo para más adelante desarrollar nuestras aplicaciones PHP.

En este curso de desarrollo Front-End nos centraremos en la configuración del entorno para programar en PHP y realizar cambios a nivel de HTML, CSS, plantillas Twig, etc.

De entre los Entornos de Desarrollo Integrados o IDEs disponibles en el mercado, nos hemos decantado por NetBeans. Veremos cómo crear proyectos locales y remotos, incluyendo la configuración para trabajar en el servidor remoto facilitado por Forcontu en sus cursos.

Comparativa D8/D7

Lo estudiado en esta unidad sirve tanto para Drupal 8 como para versiones anteriores.

Contenidos de la Unidad

- 1.1 Qué es un IDE
- 1.2 Instalación y configuración de Netbeans
- 1.3 Instalación de Drupal en el servidor de Forcontu



1.1 Qué es un IDE

Un **Entorno de Desarrollo Integrado** o **IDE** (En inglés, *Integrated Development Environment*) es una aplicación para desarrolladores que proporciona los servicios necesarios para facilitar y optimizar la programación de aplicaciones en diferentes lenguajes de programación. Generalmente, un IDE integra estas funcionalidades:

- **Editor de código fuente.** Es un editor de texto específico para programación, con resaltado de sintaxis (texto en diferentes colores), autocompletar, control de apertura y cierre de llaves, etc.
- **Autocompletado de código.** Sugerencias de código mientras estamos escribiendo, tales como nombres de variables, funciones, métodos, etc. En la programación orientada a objetos, el IDE también nos informa de los métodos disponibles para un determinado objeto.
- **Depurador.** Permite localizar y corregir errores en el código.
- **Compilador e Intérprete.** Herramientas que permiten la ejecución de la aplicación. Como veremos más adelante, PHP es un lenguaje de programación interpretado. Básicamente lo que implica es que, en lugar de compilar el código completo, va ejecutando línea a línea el código de la aplicación, para mostrar el resultado o, si los hubiera, los errores encontrados.

IDEs para desarrollar en PHP y Drupal

Aunque hay muchas soluciones IDE, algunas de los IDE más utilizados para desarrollar en PHP y Drupal son:

- **Netbeans.** Inicialmente enfocado al desarrollo de aplicaciones Java, incluye actualmente soporte para trabajar con aplicaciones PHP, HTML, Javascript, etc. Es un software de código abierto y gratuito, con una gran comunidad de usuarios y desarrolladores.
<https://netbeans.org/>
- **Eclipse PDT.** Eclipse es uno de los entornos más utilizado para trabajar con diferentes lenguajes de programación, especialmente con Java. Eclipse PDT (PHP Development Tools) es una extensión de Eclipse que añade funcionalidad específica para desarrollar el PHP. Es un software de código abierto y gratuito.
<http://www.eclipse.org/pdt/>
- **PHPStorm.** PHPStorm se está posicionando como una de las mejores soluciones para desarrollar aplicaciones en PHP y Drupal. Se trata de un software de pago, con suscripción mensual o anual. Es la opción recomendada cuando ya estés trabajando profesionalmente con Drupal, especialmente en desarrollo back-end, y necesites agilizar y optimizar tu tiempo de desarrollo.

<https://www.jetbrains.com/phpstorm/>

- **Aptana Studio.** Se trata de un IDE de código abierto que también tiene mucha aceptación en el mundo de Drupal.
<http://www.aptana.com/>
- **Komodo Edit.** Se trata de una versión limitada (gratuita y de código abierta) de **Komodo IDE**, que es una solución mucho más completa, pero de pago. Puede ser un buen comienzo para evaluar el software, aunque es muy posible que terminemos necesitando las herramientas de la versión de pago.
<http://komodoide.com/>

Netbeans es el IDE que utilizaremos en este curso. Si ya trabajas con otro IDE, puedes utilizarlo en lugar de Netbeans. Si aún no estás trabajando con ningún IDE, te recomendamos empezar con Netbeans, siguiendo las instrucciones de esta unidad. Más adelante podrás evaluar otras soluciones y decantarte por la que te resulte más cómoda y útil, en función de los requisitos de tu empresa y/o proyectos de desarrollo.

Editores de código fuente

También existen editores de código fuente que, sin llegar al nivel de un IDE, nos permiten editar el código fuente. Además, suelen tener funcionalidades propias de un IDE, como resaltado de sintaxis o autocompletado de código.

Algunas de estos editores son:

- **Atom.** <https://atom.io/>
- **Notepad++.** <https://notepad-plus-plus.org/>
- **Sublime Text.** Es una solución de pago. <https://www.sublimetext.com/>.

Aunque son soluciones totalmente válidas para programar, nuestra recomendación es empezar a trabajar con un IDE más completo, y explorar así todas las funcionalidades que se utilizan en un entorno profesional.

1.2 Instalación y configuración de NetBeans

NetBeans es un IDE de código abierto utilizado principalmente para programar en Java, pero que también permite programar en otros lenguajes como C/C++, XML, HTML y, por supuesto, PHP.

Página oficial: <https://netbeans.org/>

Descarga e instalación

Para descargar NetBeans IDE, entra en: <https://netbeans.org/downloads/>

Selecciona primero el idioma y tu sistema operativo: Windows, Linux o Mac OS X. A continuación, puedes descargar el paquete completo (All), o únicamente el específico para programar en HTML5/Javascript & PHP, que es el que realmente nos interesa [F1.2a].

F1.2a

NetBeans IDE

Página de descarga de NetBeans IDE.

Supported technologies *	Java SE	Java EE	HTML5/Javascript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME		•				•
HTML5/Javascript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

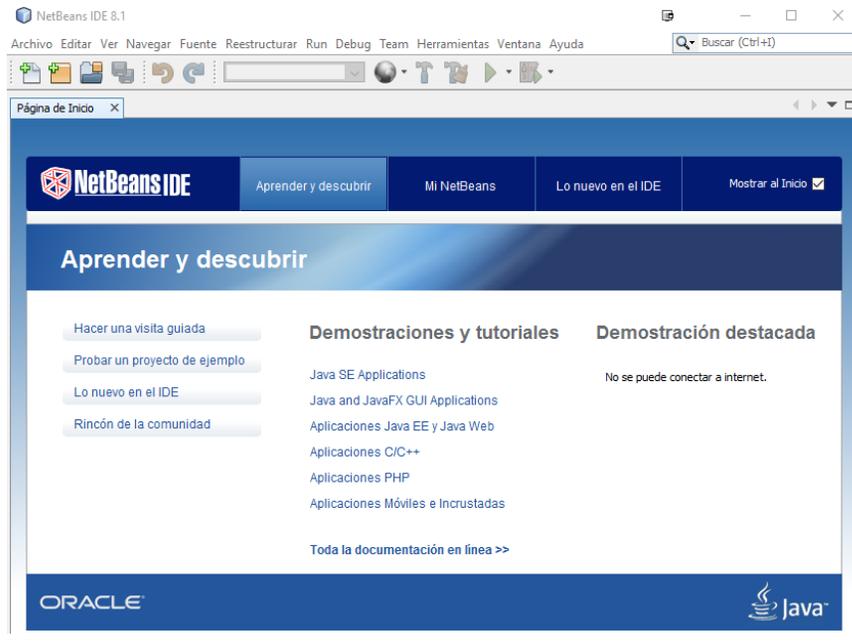
Si necesitas instrucciones de instalación específicas para tu sistema operativo, consulta esta guía (comprueba que la versión del software se corresponda con la instalada):

<https://netbeans.org/community/releases/8.1/install.html>

Al tratarse de una aplicación Java, necesitarás instalar la versión correspondiente de JDK (Java SE Development Kit). El instalador te solicitará instalarlo si no lo detecta en tu equipo durante la instalación. Descarga e instala la última versión de JDK para tu sistema operativo y procesador desde:

<http://www.oracle.com/technetwork/java/javase/downloads>

Una vez instalado, ejecuta NetBeans IDE [F1.2b]:

**F1.2b****NetBeans IDE**

Página de inicio de NetBeans IDE.

Activación de plugins

A continuación, vamos a comprobar que se encuentran activos los Plugins que necesitamos para los lenguajes de programación que vamos a usar. Accede al listado de plugins desde: **Herramientas** ⇒ **Plugins** ⇒ **Instalados**.

Como hemos instalado el paquete específico para PHP, ya aparecen instalados y activados un conjunto de plugins. Si hubiéramos instalado el paquete **All**, tendríamos que activar los plugins en este punto. Con el plugin PHP será suficiente para las pruebas iniciales, aunque más adelante, a medida que estudiemos otras tecnologías, podremos ir incorporando nuevos plugins como Git, Symfony2 Framework, Twig templates y CSS preprocessors, entre otros.

Configurando los estándares de codificación de Drupal

Como veremos más adelante, Drupal tiene sus propios estándares de codificación para que el código contribuido por la comunidad sea lo más homogéneo posible. Por ejemplo, la indentación del código, que normalmente se realiza con el tabulador, en Drupal debe ser de 2 espacios. Veamos como cambiar éste y otros parámetros.

Selecciona los siguientes valores desde **[F1.2c]**:

Herramientas ⇒ **Opciones** ⇒ **Editor** ⇒ **Formato**

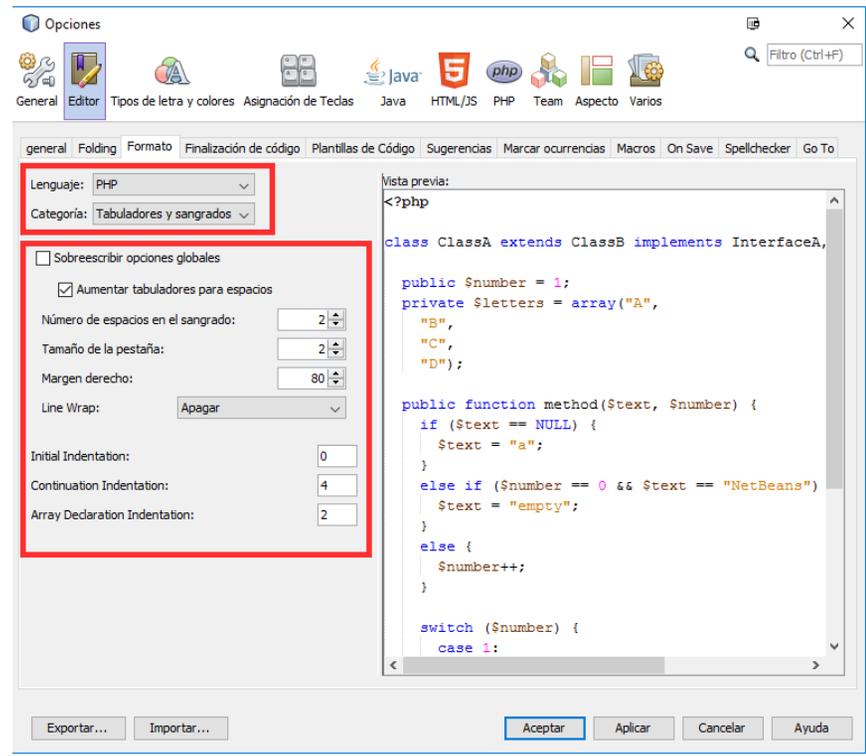
- Lenguaje: PHP
- Categoría: Tabuladores y sangrados
- Sobrescribir opciones globales (desmarcar)
- Aumentar tabuladores para espacios: (marcar)
- Número de espacios en el sangrado: 2

- Tamaño de la pestaña: 2
- Margen derecho: 80
- Line Wrap: Off (apagado)
- Initial Indentation (Indentación inicial): 0
- Continuation Indentation: 4
- Array Declaration Indentation: 2

F1.2c

NetBeans IDE

Configuración de estándares de codificación.

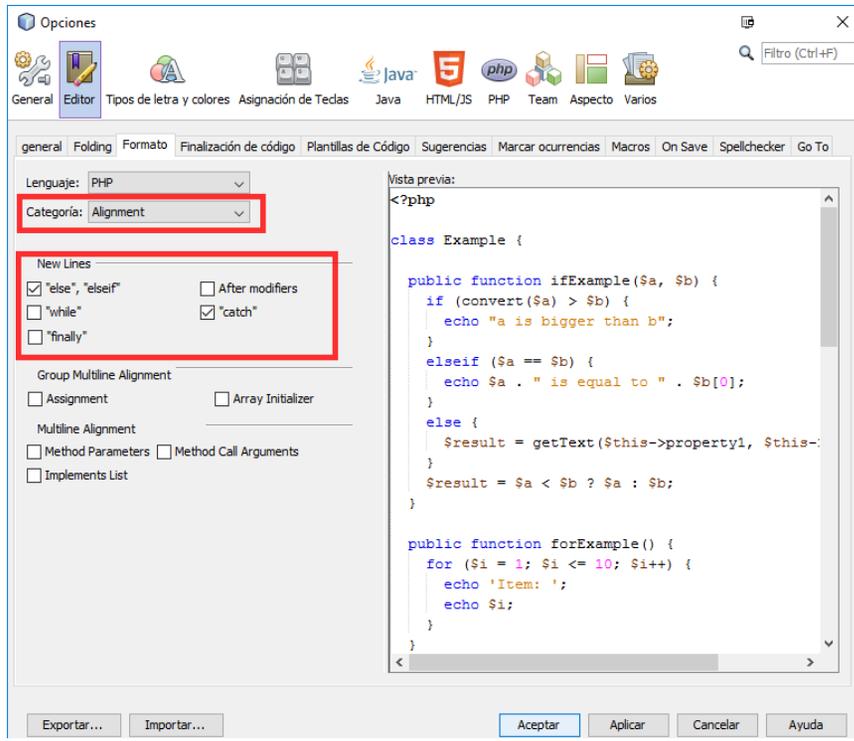


En el cuadro de la derecha (Preview), puedes ver un ejemplo de código PHP y cómo va quedando formateado en función de las opciones seleccionadas.

Desde la misma pestaña, **Formato**, pero seleccionando la categoría **Alignment** [F1.2d], selecciona, dentro de **New lines**:

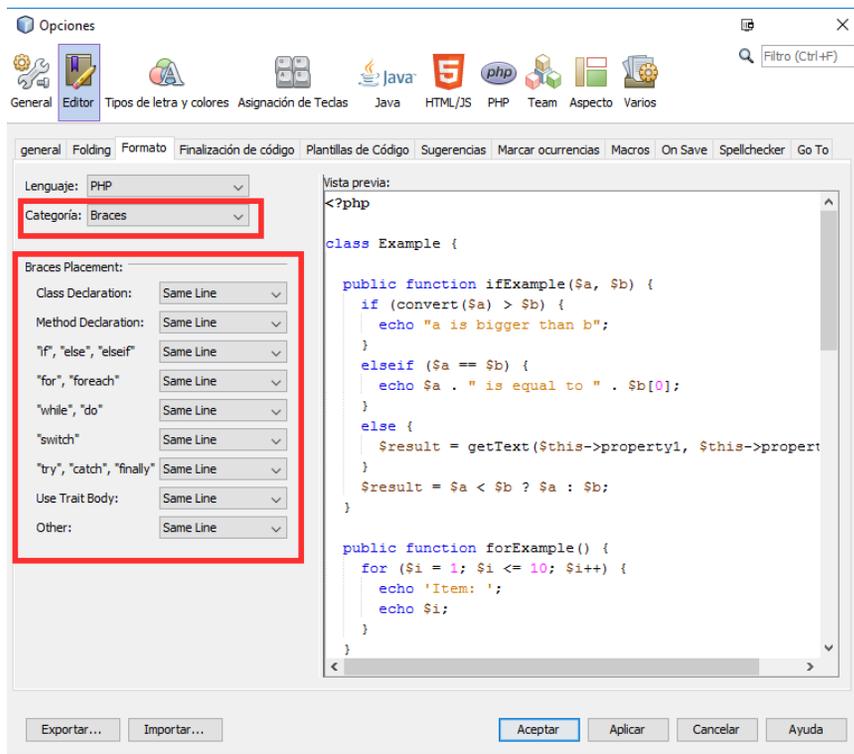
- "else", "elseif"
- "catch"

De esta forma estamos indicando que estas cláusulas comenzarán en línea nueva.



F1.2d
NetBeans IDE
 Configuración de estándares de codificación.

A continuación, selecciona la categoría Braces (llaves {}), y marca todos como Same Line **[F1.2e]**. De esta forma estamos indicando que al apertura de llave { se realiza en la misma línea que el comando. Prueba por ejemplo a marcar los "if" como "New Line", y comprueba el resultado en el código de ejemplo. No olvides dejarlo todo como "Same Line".



F1.2e
NetBeans IDE
 Configuración de estándares de codificación.

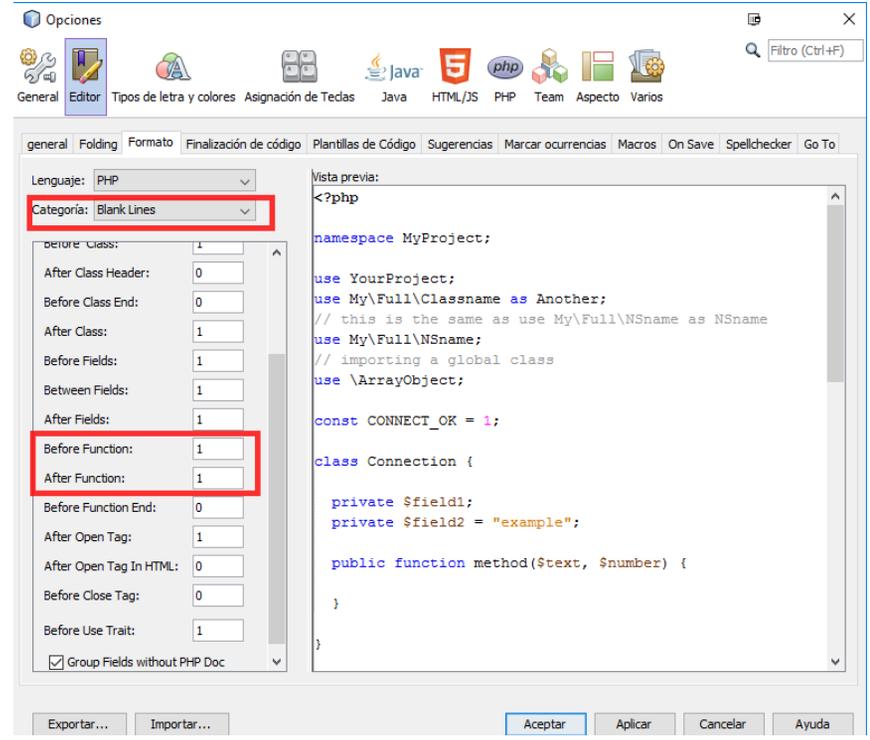
Continuamos con la categoría **Blank lines** (líneas en blanco) **[F1.2f]**. Antes y después de las funciones indicaremos el valor 1:

- Before Function: 1
- After Function: 1

F1.2f

NetBeans IDE

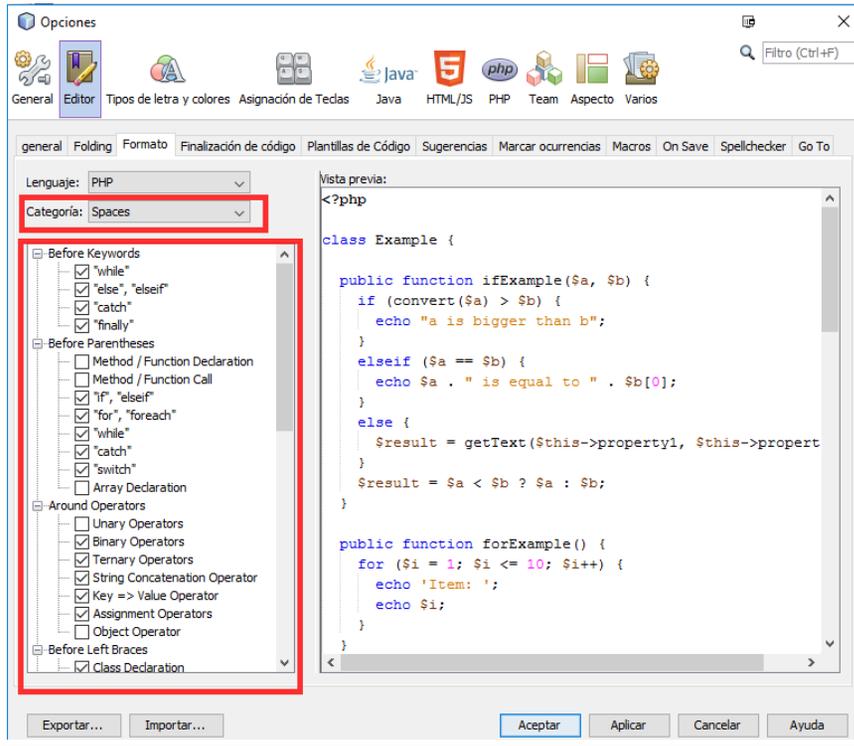
Configuración de estándares de codificación.



Dentro de la categoría **Spaces** **[F1.2g]**, aplica la siguiente configuración:

- Before Keywords:
 - o Activados: while, else/elseif, catch, finally
- Before Parentheses:
 - o Desactivados: Method / Function Declaration, Method / Function Call, Array Declaration
 - o Activados: if/elseif, for/foreach, while, catch, switch
- Around operators:
 - o Desactivados: Unary Operators, Object Operator
 - o Activados: Binary Operators, Ternary Operators, String Concatenation Operator, Key => Value Operator y Assignment Operators.
- Before Left Braces:
 - o Activados: Todos
- Within Parentheses
 - o Desactivados: Todos
- Other:
 - o Desactivados: Before Comma, Before Semicolon

- o Activados: Todos los demás.



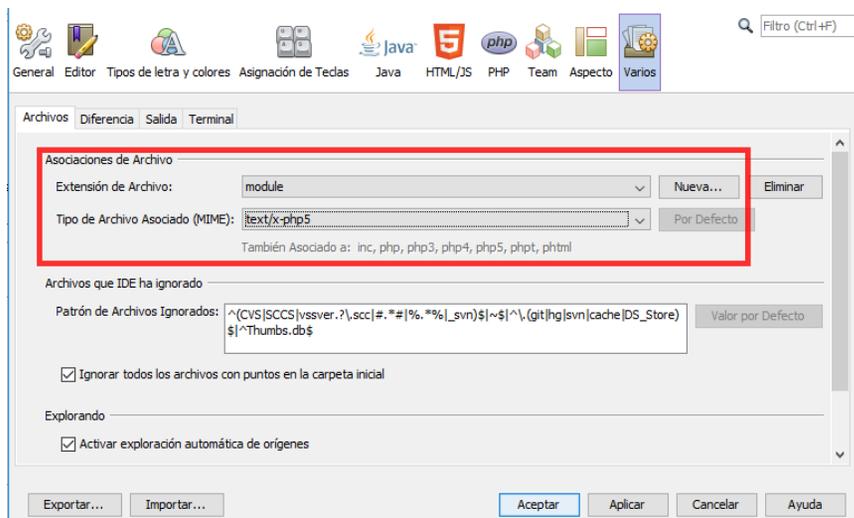
F1.2g
NetBeans IDE
 Configuración de estándares de codificación.

Extensión de archivos

Drupal utiliza extensiones de archivo específicas como .module, .inc, .install, que contienen código PHP. A continuación, debemos indicar esta asociación para que el editor sepa que el contenido de esos archivos es PHP (y aplique la configuración definida anteriormente). Accede a **[F1.2h]**:

Herramientas ⇒ Opciones ⇒ Varios (Miscellaneous) ⇒ Archivos

Añade una nueva extensión (botón Nueva...) y escribe "module". En el Type (MIME), selecciona "text/x-php5" y Aplicar.



F1.2h
NetBeans IDE
 Extensión de archivos.

Siguiendo los mismos pasos, añade el resto de extensiones relacionadas con PHP: inc (ya añadida por defecto), install, test, profile, etc. A medida que te vayas encontrando con otras extensiones que contengan PHP, las puedes ir añadiendo.

Añade también las extensiones **info** (para Drupal 7) y **po** pero en el tipo de archivo selecciona "text/plain".

A medida que avancemos con Drupal 8 iremos viendo otras extensiones de archivos que podrás añadir en la configuración de la misma forma.

Funciones de la API de Drupal

Como hemos instalado el paquete de PHP, las funciones de la API de PHP estarán disponibles en modo **autocompletar**. De esta forma sólo tenemos que empezar a escribir el nombre de una función y el editor nos ayudará a completarla, incluyendo sus parámetros. Vamos a ver ahora cómo **incorporar las funciones de la API de Drupal**.

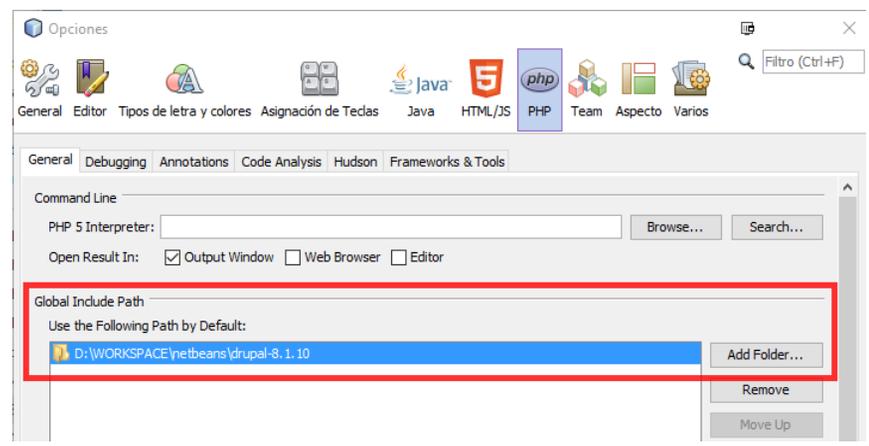
NetBeans sólo necesita leer el código del núcleo de Drupal para incorporar automáticamente las funciones. No hace falta instalar Drupal pero sí descargarlo y descomprimirlo, para indicarle al editor dónde se encuentran los archivos. Una vez descargado y descomprimido, indica la ruta desde **[F1.2i]**:

Herramientas ⇒ Opciones ⇒ PHP ⇒ General (Add Folder)

F1.2i

NetBeans IDE

Funciones de la API de Drupal



Puedes añadir diferentes versiones de Drupal (Drupal 7 y Drupal 8) y también **rutas a módulos contribuidos** con los que trabajas habitualmente.

Dentro de la misma pestaña tenemos la opción PHP 5 Interpreter, donde se pide la ruta de la aplicación encargada de ejecutar (o interpretar) el código PHP. Esto dependerá del software de servidor web que tengamos instalado. En la siguiente unidad veremos cómo instalar algunas de las aplicaciones AMP (Apache, MySQL, PHP) más utilizadas, con las que podremos montar un servidor web y programar en local, aunque **en este curso realizaremos las actividades en un servidor remoto**.

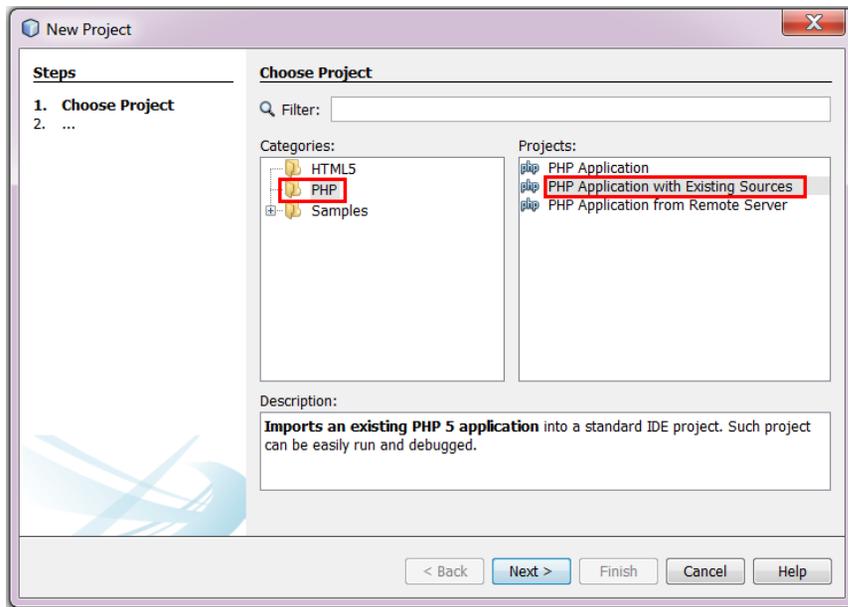
Crear un proyecto en local

Lo primero que tenemos que hacer para empezar a trabajar es crear un proyecto desde **[F1.2k]**:

Archivo (File) ⇒ **New Project**

seleccionando:

Tools ⇒ **PHP** ⇒ **PHP Application with Existing Sources**



F1.2k

NetBeans IDE

Crear un proyecto local.

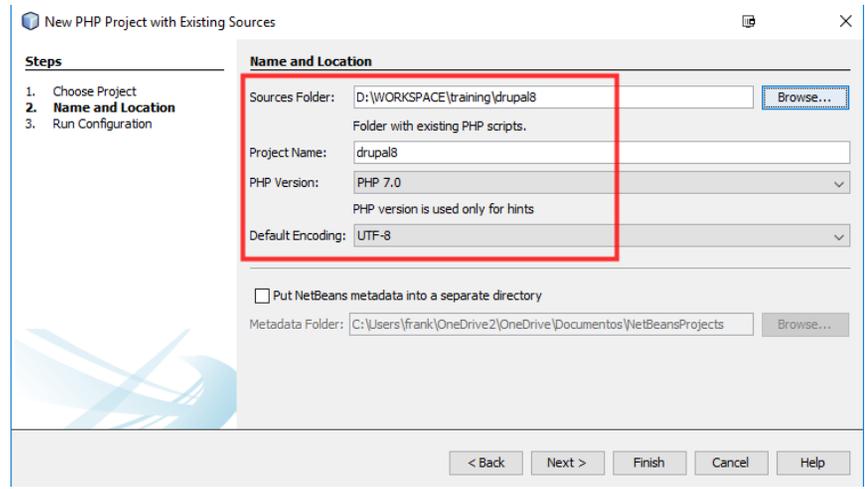
A continuación (Next), seleccionaremos la carpeta donde se encuentra el proyecto. Si tenemos una instalación de Drupal, podemos apuntar directamente a la carpeta **sites**, que será donde estén los módulos y temas a modificar. Sin embargo, si en algún momento necesitamos navegar por el código del núcleo, por ejemplo, para depurar, entonces sí será necesario incluir el proyecto desde el directorio raíz.

Indica también el nombre del proyecto, la versión de PHP (la que hayas instalado en tu servidor local) y la codificación por defecto, UTF-8. **Para Drupal 8 selecciona PHP 7.x [F1.2I]**.

F1.2l

NetBeans IDE

Crear un proyecto local.

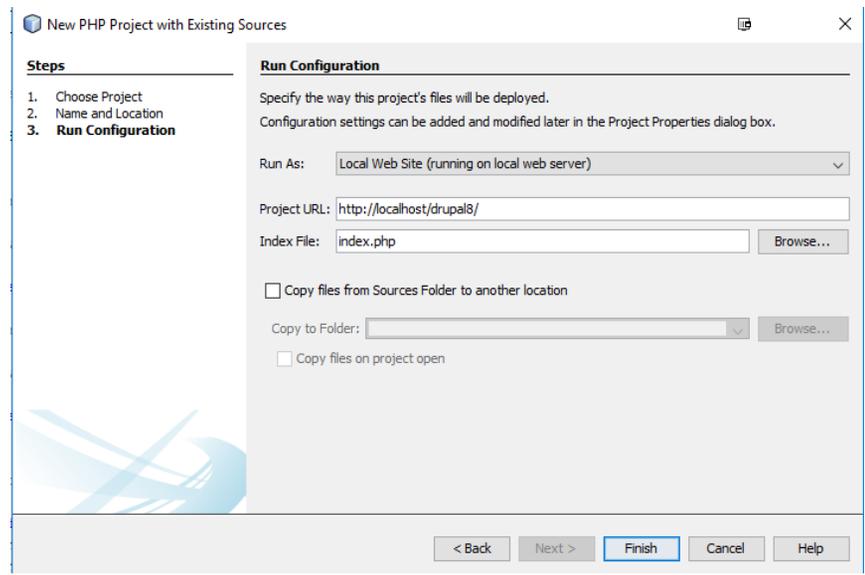


Por último, NetBeans nos pide información sobre el servidor web que se encarga de ejecutar el proyecto, que puede ser local o remoto [F1.2m]. Como ya hemos comentado, en la siguiente unidad se estudiará cómo instalar un servidor web en local, aunque **durante el curso trabajaremos en un servidor remoto**.

F1.2m

NetBeans IDE

Crear un proyecto local.



Haciendo clic en **Finish** se creará el proyecto y podremos navegar por el código, crear o editar archivos, etc.

Crear un proyecto desde servidor remoto

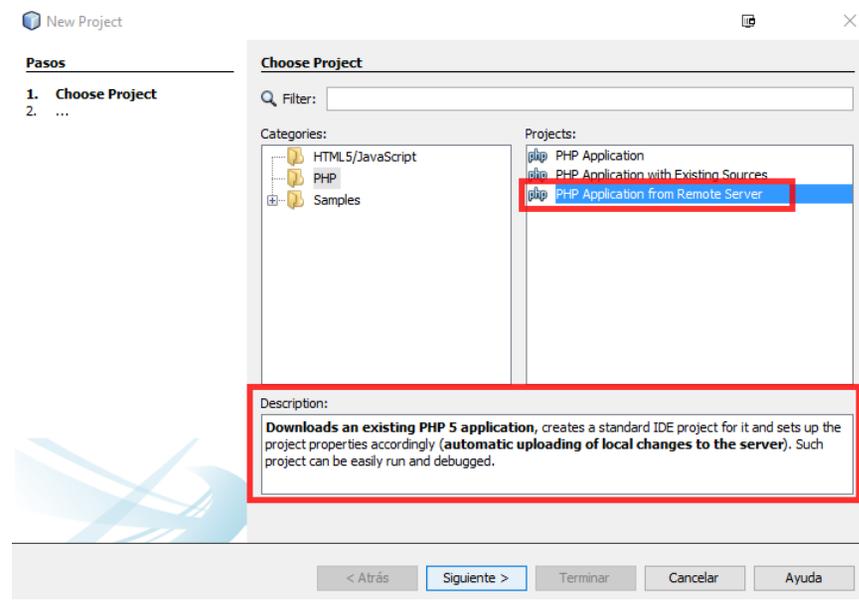
En este caso vamos a considerar que el proyecto no se encuentra en nuestro equipo local, sino que está en un servidor remoto. En este modelo de trabajo, que será el que utilizemos en el curso de Forcontu, los cambios que realicemos en local se subirán automáticamente al servidor remoto.

Crearemos el proyecto **[F1.2n]**:

Archivo (File) ⇒ New Project

seleccionando:

Tools ⇒ PHP ⇒ PHP Application from Remote Server



F1.2n

NetBeans IDE

Crear un proyecto desde servidor remoto.

A continuación (Siguiente, Next), indicaremos un nombre de proyecto y una carpeta local desde donde modificaremos los archivos de la aplicación, que luego se subirán al servidor remoto. Para Drupal 8 seleccionaremos PHP 7.x y codificación por defecto UTF-8 **[F1.2o]**.

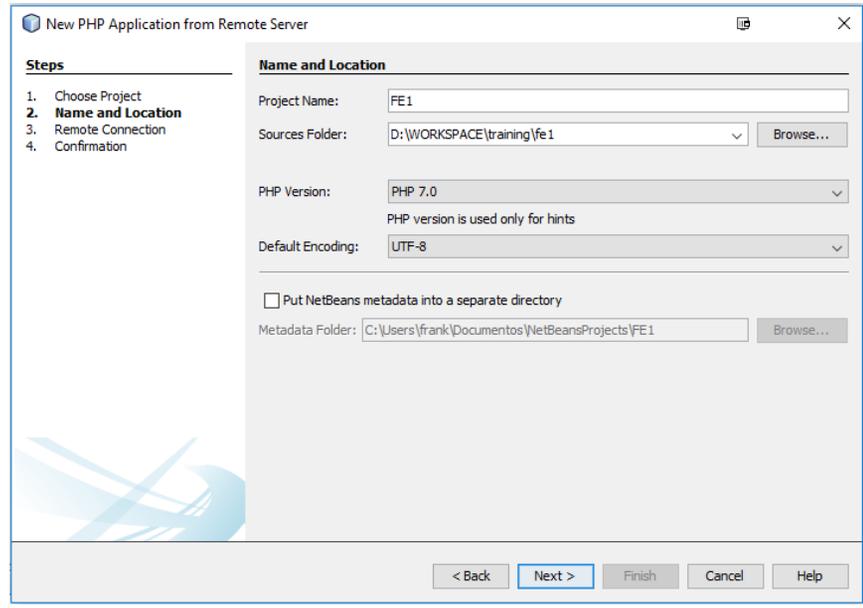
Como primer **ejemplo práctico**, vamos a configurar un proyecto para trabajar con HTML y CSS. Como Drupal está construido con PHP, seleccionaremos PHP, aunque también podríamos seleccionar HTML5/JavaScript para trabajar con proyectos exclusivamente en HTML. Más adelante tendrás que seguir los mismos pasos para crear otros proyectos específicos para trabajar con Drupal.

Nuestro primer proyecto se llamará **FE1** y estará ubicado en una carpeta de nuestro equipo llamada **fe1**. Aunque tienes total libertad para almacenar los proyectos en cualquier carpeta de tu equipo, procura organizarlos bajo una carpeta padre llamada `\proyectos`, `\training`, `\drupal` o similar.

F1.2o

NetBeans IDE

Crear un proyecto desde servidor remoto.



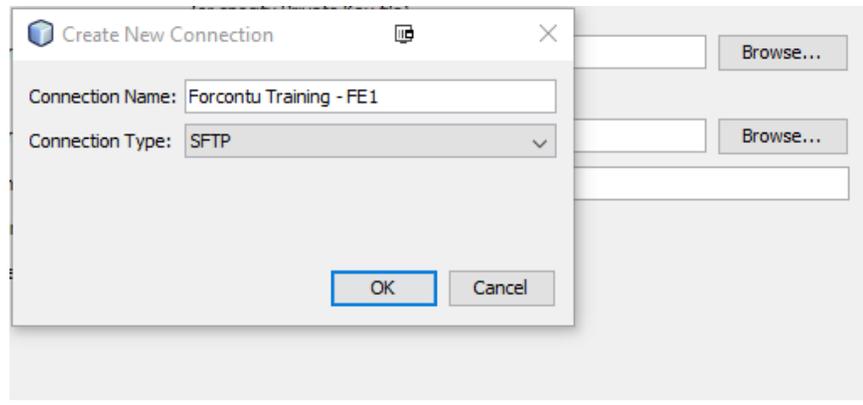
El siguiente paso consiste en configurar la conexión con el servidor remoto:

- La **URL del proyecto** (Project URL) es la ruta desde donde se ejecuta la aplicación. En nuestro caso, para los scripts de PHP utilizaremos la ruta:
http://fe1.xxx.training.forcontu.com (sustituye xxx por tu nombre de usuario).
- En **conexión remota** (Remote Connection), crearemos una nueva conexión por SFTP, haciendo clic en "Manage..." **[F1.2p] [F1.2q]**.

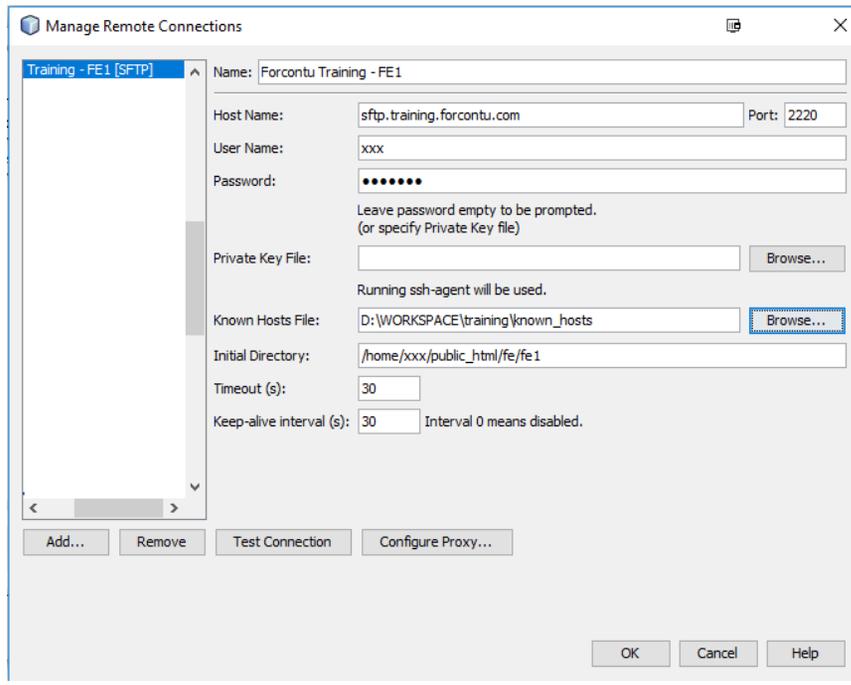
F1.2p

NetBeans IDE

Crear un proyecto desde servidor remoto.

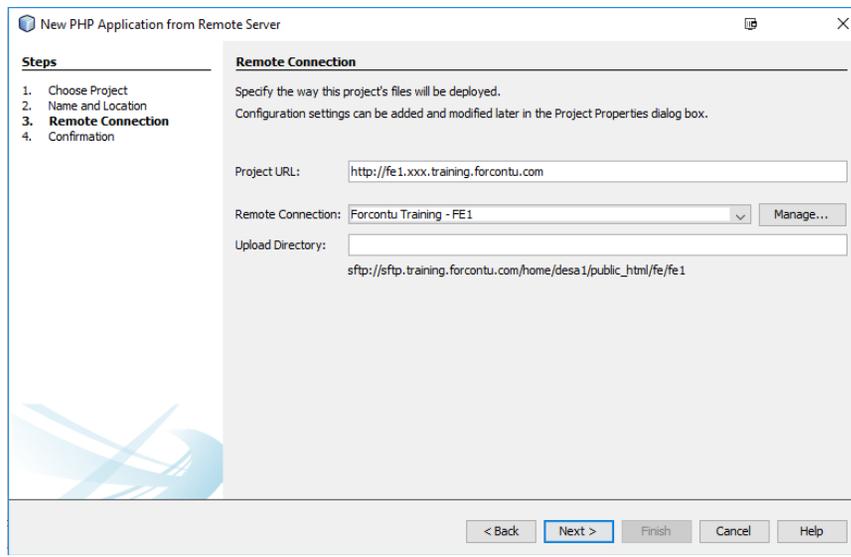


Si estás realizando el curso de Forcontu, los datos de acceso por SFTP al servidor los encontrarás en **Mis alojamientos**. Indica el nombre del servidor (sftp.training.forcontu.com), puerto 2220, tu nombre de usuario y contraseña, y la carpeta raíz del proyecto. Los scripts se ubicarán en: /home/xxx/public_html/fe/fe1 (sustituyendo xxx por tu nombre de usuario).

**F1.2q****NetBeans IDE**

Crear un proyecto desde servidor remoto.

- Deja el **directorio de subida** (Upload directory) vacío, para que los archivos se suban directamente en el raíz o carpeta inicial del proyecto en el servidor remoto **[F1.2r]**.

**F1.2r****NetBeans IDE**

Crear un proyecto desde servidor remoto.

Si intentas contactar ahora, el sistema de dará un error porque no encuentra, en el servidor, ningún archivo para descargar. Para evitar esto, conecta directamente por SFTP (vía Filezilla), y sube un archivo index.php con un script básico de ejemplo (por ahora puedes crearlo con cualquier editor de texto que tengas instalado en tu equipo, tipo block de notas, notepad++, vim o similar):

```
<?php

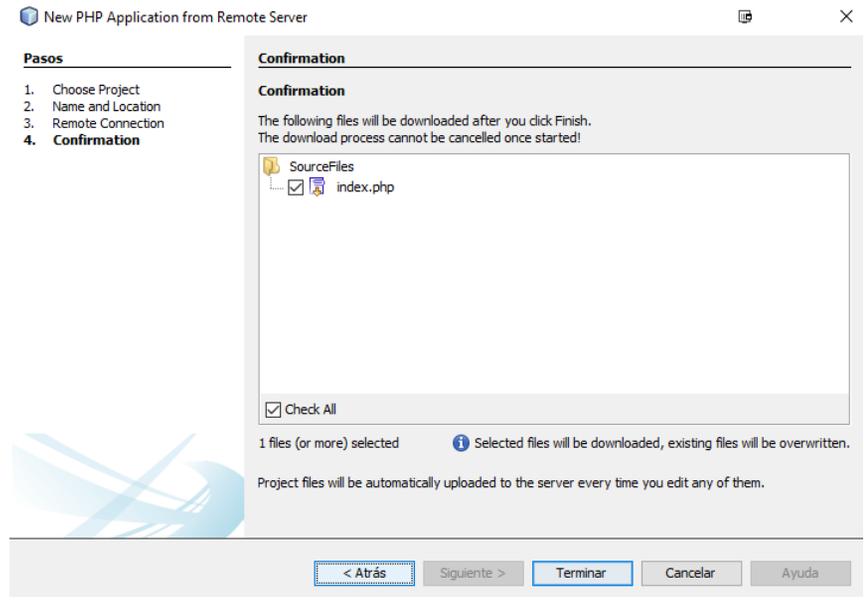
print "Hello, Forcontu";
```

En local, no guardes el archivo en la carpeta del proyecto, para que puedas ver cómo se sincroniza, descargándolo automáticamente desde el servidor **[F1.2s]**.

F1.2s

NetBeans IDE

Crear un proyecto desde servidor remoto.

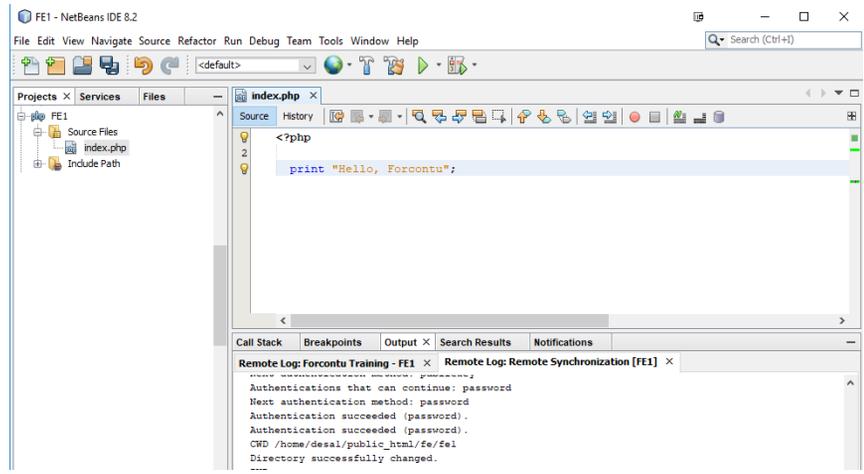


Al **Terminar**, se creará el proyecto y se hará una copia en local de los archivos del servidor. En el nuevo entorno de trabajo de Netbeans, podremos navegar por los archivos del proyecto (ventana Projects), abrir cualquiera de esos archivos para su edición **[F1.2t]**.

F1.2t

NetBeans IDE

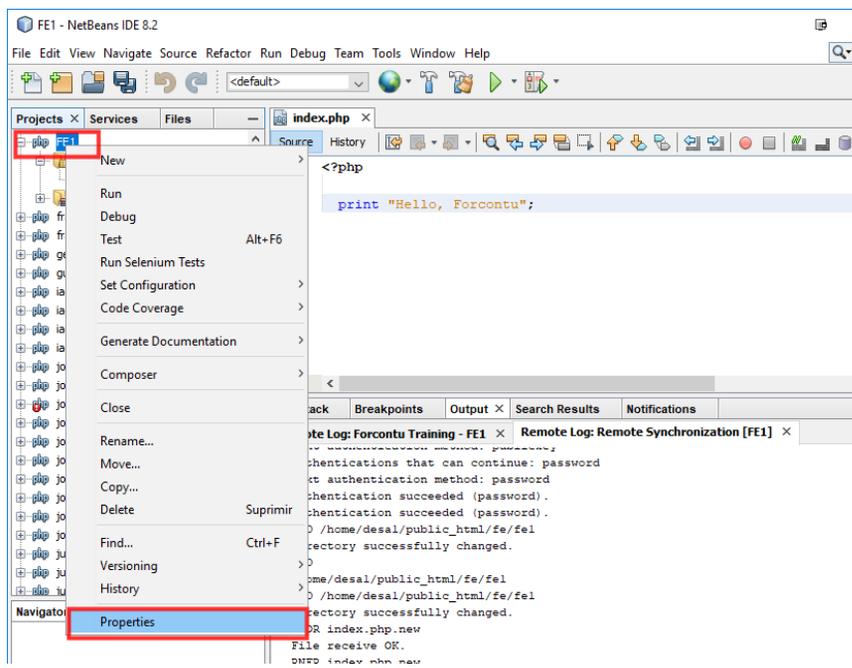
Crear un proyecto desde servidor remoto.



Realiza las siguientes comprobaciones para asegurarte de que todo está correcto:

- Accede desde tu equipo a la carpeta local donde deberían alojarse los archivos del proyecto. Además del archivo **index.php**, deberías encontrar una carpeta **nbproject** donde netbeans almacena la información del proyecto. No debes tocar esta carpeta.
- Ejecuta la URL del proyecto para comprobar que funcione tu script PHP: `http://fe1.xxx.training.forcontu.com/index.php`. Deberías ver el mensaje "Hello, Forcontu" en tu navegador.

Si necesitas realizar modificaciones sobre el proyecto, haz clic sobre el nombre del proyecto con botón derecho y busca la opción Propiedades (properties) **[F1.2u]**.



F1.2u

NetBeans IDE

Modificar las propiedades de un proyecto.

Vamos a modificar el archivo `index.php` para comprobar que los cambios se suben al servidor. Añade, por ejemplo, un signo de exclamación al final de la cadena.

```
<?php

print "Hello, Forcontu!";
```

Cuando un archivo se ha modificado y no se ha guardado aún, el nombre en la pestaña se muestra en **negrita**. Guarda los cambios con **Control + S**. Los cambios se subirán al servidor. Puede pedir confirmación la primera, e incluso lanzar un error de validación del certificado SSL. Simplemente acepta y el servidor quedará guardado como servidor de confianza. Por último, actualiza la página en el navegador para visualizar los cambios.

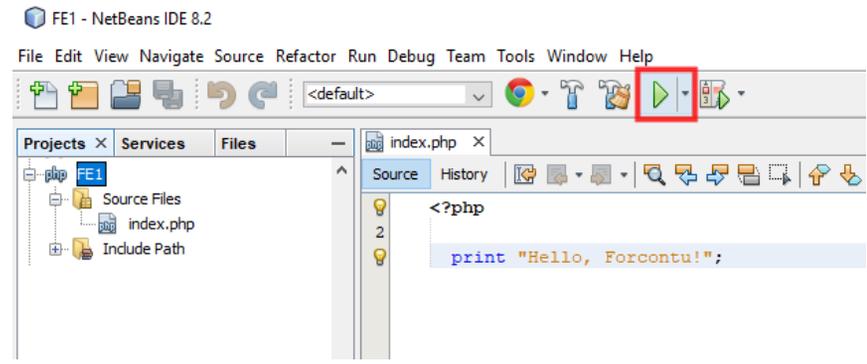
Ejecución del proyecto desde el navegador

En la barra de herramientas encontrarás un icono de Play que permite ejecutar el proyecto. Es una forma rápida y recomendada de ejecutar el proyecto en un navegador. Si haces clic en este botón, se abrirá la URL del proyecto en el navegador por defecto (Internet Explorer) **[F1.2v]**.

F1.2v

NetBeans IDE

Ejecución del proyecto en navegador.

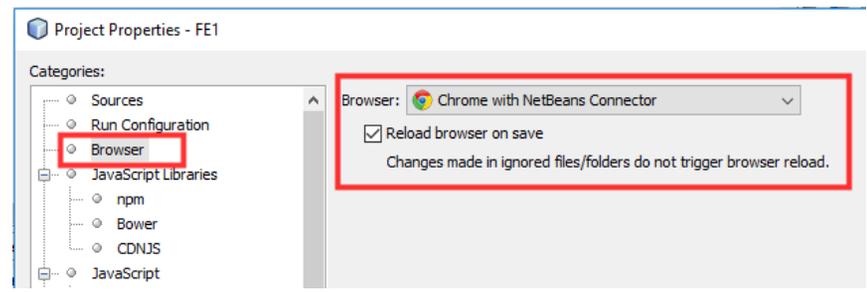


Para cambiar el navegador por defecto, accede a las **propiedades del proyecto** y, en el apartado **Browser**, selecciona **Chrome with NetBeans Connector**, dejando marcada la opción "Reload browser on save" (Recargar el navegador al guardar) **[F1.2w]**.

F1.2w

NetBeans IDE

Selección del navegador por defecto.



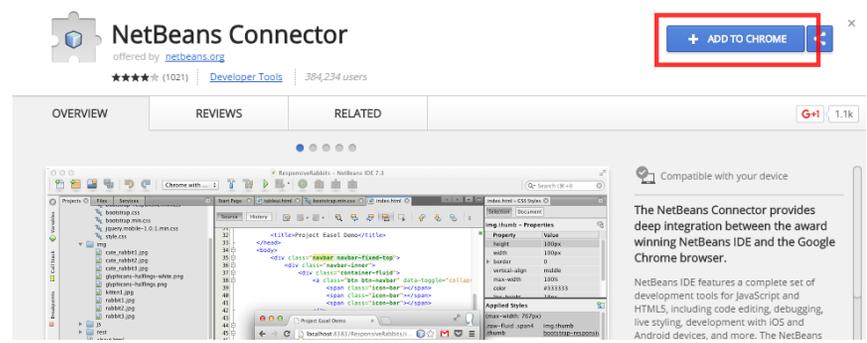
NetBeans Connector es un plugin de Chrome que tenemos que instalar desde Chrome Web Store. El plugin permite la comunicación bidireccional entre el servidor y NetBeans, necesario para depurar el código en remoto **[F1.2x]**.

<https://chrome.google.com/webstore/detail/netbeans-connector/hafdlehqocfodbqjnpecfajgkeejnaa>

F1.2x

NetBeans IDE

Plugin NetBeans Connector para Chrome.



Al ejecutar nuevamente el proyecto, este se abrirá en Chrome y se mostrará un mensaje de NetBeans Connector indicando que está en funcionamiento **[F1.2y]**.



F1.2y NetBeans IDE

Ejecución del proyecto en Chrome con NetBeans Connector.

1.3

Instalación de Drupal en el servidor de Forcontu

Durante este curso vamos a trabajar en un servidor remoto de Formato. Si estás matriculado, encontrarás los datos de acceso al alojamiento en la opción de menú Mis Alojamientos.

Forcontu nos permite acceder a sus servidores vía SSH, con lo que podremos utilizar todas las herramientas de consola disponibles para desarrolladores (Drush, Drupal Console, comandos Linux, etc.).

Instalación por SFTP

La forma clásica que hemos venido utilizando para instalar Drupal es vía SFTP (o FTP). Es el método habitual cuando el proveedor del alojamiento no nos facilita un acceso SSH o cuando no se dispone de los conocimientos necesarios para utilizar la consola del servidor. Este es el método que utilizamos en el curso de Site Building, donde a los participantes no se les enseña a acceder vía consola y ejecutar comandos.

Para instalar Drupal, descargaremos la última versión y la descomprimiremos en nuestro equipo. Luego subiremos los archivos a la carpeta correspondiente, utilizando Filezilla o similar. Por último, realizamos la instalación vía navegador. Encontrarás información detallada paso a paso sobre este método de instalación en el libro Experto en Drupal 8 Site Building I.

Instalación de Drupal desde consola (SSH)

En este nivel vamos a realizar la instalación de una forma más rápida, accediendo al servidor vía consola (SSH). A continuación, usaremos algunos comandos de Linux y Drush. No te preocupes si no tienes experiencia ejecutando comandos desde consola, en próximas unidades veremos estos comandos con más detalle.

Seguiremos estos pasos:

1. Con PuTTY (o software equivalente), accede al servidor vía SSH. Si estás matriculado en el curso de Forcontu, encontrarás los datos de acceso en Mis Alojamientos.
2. Accede a la carpeta `/home/xxx/public_html/fe`. Esta carpeta contiene varias subcarpetas (`fe1`, `fe2`, `fe3`, etc.), que se corresponden con los dominios que también hay disponibles (`fe1.xxx.training.forcontu.com`). Como al entrar al servidor estamos ubicados en `/home/xxx`, ejecuta el comando:

```
$cd public_html/fe
```

Nota: Ojo, el \$ no forma parte del comando y no tienes que escribirlo. Se utiliza para indicar que es un comando de consola. De hecho, en tu consola verás un \$, indicando que está esperando que indiques un comando.

3. Descargamos Drupal usando Drush. Concretamente usaremos el comando **drush dl** (alias de **drush pm-download**).

<https://drushcommands.com/drush-8x/pm/pm-download/>

```
$drush dl drupal --drupal-project-rename=fe1
```

Con estas opciones estamos indicando que instale la última versión de Drupal y que renombre la carpeta del proyecto a fe1, con lo que nuestro sitio quedaría ubicado en:

`/home/xxx/public_html/fe/fe1`

Si queremos seleccionar la versión a instalar, podemos utilizar estas otras opciones en el mismo comando **drush dl**:

```
$drush dl drupal drupal-8 --default-major=8 --select --all --drupal-project-rename=fe1
```

Choose one of the available releases for drupal:

```
[0] : Cancel
[1] : 8.3.x-dev - 2016-Dec-23 - Development
[2] : 8.2.4 - 2016-Dec-07 - Supported, Recommended
[3] : 8.2.3 - 2016-Nov-16 - Security
[4] : 8.2.2 - 2016-Nov-02 -
[5] : 8.2.x-dev - 2016-Oct-23 - Development
[6] : 8.2.1 - 2016-Oct-11 -
[7] : 8.2.0 - 2016-Oct-05 -
```

4. Solo hemos subido los archivos de Drupal, pero aún no hemos realizado la instalación. Una opción sería ir por navegador e instalar Drupal por interfaz:

`http://fe1.xxx.training.forcontu.com`

Pero vamos a seguir trabajando con Drush para realizar la instalación de una forma más rápida, así que saltamos al siguiente paso.

5. Para instalar el sitio utilizamos el comando **drush site-install** (alias **drush si**).

<https://drushcommands.com/drush-8x/core/site-install>

```
$cd fe1
$drush si standard
--db-url='mysql://db_user:db_pass@127.0.0.1/db_name'
--site-name='Experto en Drupal 8 Front-End Development'
--site-mail=yourmail
--account-name=admin
--account-pass=yourpassword
--account-mail=yourmail
--locale=es
```

En primer lugar tenemos que entrar a la carpeta donde se instalará Drupal (cd fe1).

Luego ejecutamos **drush si** (o **drush site-install**) con los siguientes parámetros. Ten en cuenta que **los parámetros van en la misma línea, separados por espacio**. En el código de ejemplo los hemos separado con saltos de línea para facilitar la lectura.

- standard (indica el nombre del perfil de instalación).
- --db_url. Parámetros de acceso a la base de datos. En nuestro caso utilizamos localhost porque la base de datos está en el mismo servidor que el alojamiento. Si estuvieran en servidores separados, habría que indicar la IP o dominio facilitado por el proveedor. **Consulta los datos de acceso a la base de datos en Mis alojamientos** (www.forcontu.com).
- --site-name y --site-mail con los datos del sitio.
- --account-name, --account-pass y --account-mail, con los datos del usuario administrador (uid 1).
- --locale. Si queremos instalar Drupal en otro idioma. Por ejemplo, --locale=es para instalar Drupal en español.

El sistema nos indicará si la instalación se ha realizado correctamente o si se ha producido algún error.

```
You are about to create a
/home/desal/public_html/fe/fe1/sites/default/settings.php file and DROP all tables
in your 'desal_fe1' database. Do you want to continue? (y/n): y
Starting Drupal installation. This takes a while. Consider using the          [ok]
--notify global option.
Installation complete.  User name: admin  User password: *****          [ok]
One translation file imported. 8334 translations were added, 0              [status]
translations were updated and 0 translations were removed.
¡Enhorabuena! Ha instalado Drupal.                                         [status]
3 translation strings were skipped because of disallowed or malformed      [warning]
HTML. See the log for details.
```

Nota: Los warning sobre cadenas de traducción son frecuentes y no implican ningún problema real durante la instalación.

6. Por último, comprueba en tu navegador que el sitio funciona correctamente y que puedes acceder con el usuario administrador indicado.

<http://fe1.xxx.training.forcontu.com>

7. En Forcontu, el acceso a la base de datos se realiza a través de phpMyAdmin. Consulta los datos de acceso completos en Mis alojamientos.

Instalación de módulos contribuidos

Para instalar módulos contribuidos utilizaremos el comando "drush en" (alias de "drush pm-enable", como se explica a continuación. Ejecuta estos comandos desde dentro del directorio raíz de tu sitio (por ejemplo, public_html/fe/fe1).

drush pm-enable

Instala el módulo indicado.

```
drush pm-enable módulo[s]
drush en
```

Si el módulo no está en el servidor, intentará descargarlo desde el repositorio de Drupal.

```
$ drush pm-enable admin_toolbar
admin_toolbar was not found. [warning]
The following projects provide some or all of the extensions not found: [ok]
admin_toolbar
Would you like to download them? (y/n): y
Project admin_toolbar (8.x-1.17) downloaded to [success]
/home/ccc/public_html/fe/fe1/modules/contrib/admin_toolbar.
Project admin_toolbar contains 2 modules: admin_toolbar_tools, admin_toolbar.
The following extensions will be enabled: admin_toolbar
Do you really want to continue? (y/n): y
admin_toolbar was enabled successfully. [ok]
```

Instala los siguientes módulos de uso general durante el desarrollo de un proyecto:

- Admin Toolbar (admin_toolbar). Instala también el submódulo admin_toolbar_tools, utilizando nuevamente el comando "drush en".

https://www.drupal.org/project/admin_toolbar

- Devel. Instala también los submódulos webprofiler, devel_generate y kint.

<https://www.drupal.org/project/devel>

Tras la instalación, vacía la caché del sitio.

drush cache-rebuild

Vacía todas las cachés. Es la alternativa a **drush cc all** en Drupal 8.

```
drush cache-rebuild
drush cr
```

```
$ drush cr
Cache rebuild complete. [ok]
```

<https://drushcommands.com/drush-8x/cache/cache-rebuild/>

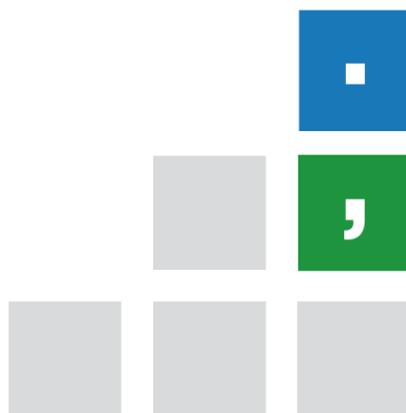
Fran Gil

Front-End Development I

2

Instalación y configuración de temas

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

2 Instalación y configuración de temas

En Drupal, un tema es un conjunto de archivos que permite cambiar el aspecto gráfico del sitio. La elección del tema dependerá de la estructura que queramos para el sitio que estamos diseñando, por eso es importante pensar primero en las secciones que éste tendrá y en su estructura (menús, bloques, columnas, ancho fijo o variable, etc.). Cuanto más se adapte el tema a la estructura final deseada, menos modificaciones habrá que realizar.

En este curso de Front-End comenzamos a estudiar los temas con mayor profundidad, aunque necesitamos repasar algunos de los conceptos estudiados en los niveles de Site Building y desarrollo Back-End.

Comparativa D8/D7

Motor de plantillas

El principal cambio con respecto a los temas es el motor de plantillas. En Drupal 8 se utiliza Twig como motor de plantillas, mucho más potente que el utilizado en Drupal 7 (phptemplate) y sin PHP.

Contenidos de la Unidad

- 2.1 Qué son los temas
- 2.2 Repositorio de temas
- 2.3 Instalación de temas
- 2.4 Configuración del tema
 - Caso práctico 2.4a Instalación y configuración del tema Mayo
- 2.5 Desinstalar un tema
- 2.6 Tema de administración

2

2.1 Qué son los temas

Un **tema** (*theme* en inglés) es un conjunto de archivos que le da al sitio web un aspecto visual determinado. Cambiando un tema por otro es posible modificar el aspecto completo de la web, sin cambiar con ello las funcionalidades implementadas y los contenidos creados.

Con los temas, Drupal **separa los contenidos de la presentación** o diseño gráfico con que se muestran. Los temas están diseñados con **hojas de estilos en cascada o CSS** (*Cascade Style Sheets*, en inglés), lo que hace que sea relativamente sencillo modificarlos y adaptarlos a necesidades específicas.

En este nivel del curso no se pretende que el estudiante aprenda a modificar temas, aunque se dará una breve introducción a la edición de los archivos CSS contenidos en el tema.

En la **Figura [F2.1a]** se muestra un ejemplo de sitio web con Drupal al que se le ha aplicado el tema **Mayo**, que se encuentra disponible en el repositorio de temas gratuitos de Drupal.

F2.1a Sitio demo con tema Mayo

Los temas compartidos por la comunidad se pueden descargar gratis desde drupal.org.

The screenshot shows a Drupal 8 website with the Mayo theme. The header includes the site title "Drupal 8 & Mayo Theme" and a search bar. Below the header is a navigation menu with links: Home, Basic Article, Superfish, Augue Turpis, crawroc, Example.com. The main content area is divided into columns. The "Top first column" features a sunset image and text: "To get the main menu like we want with drop downs will require the Superfish module which pledges to be ready for full release of Drupal 8. Vel sed ac. Sociis pid dolor? Vut placerat pid scelerisque, rhoncus parturient vut et odio magna tempor! Penatibus vut sit lundium turpis velit ac turpis a cum, ac dictumst dolor sociis porta lacus porta cum augue, turpis penatibus! Ridiculus ut magna scelerisque et nunc, ut odio in sit ac est? Ultricies tincidunt. In adipiscing nunc amet! Odio, dolor pulvinar etiam dapibus eros cursus? Parturient ac. Scelerisque cursus elementum a tortor tortor lectus. Adipiscing ac tristique, sed! Ac elementum, vel placerat! Turpis in sit porta ultricies a! Enim lorem sociis sagittis, nunc ultricies? Ac cras nec augue scelerisque porta mattis augue, mattis porta elementum hac integer adipiscing quis porttitor! Eu porta in lorem a enim eu nunc montes aliquet eu nec." The "Top third column" features a purple pansy image and text: "Turpis? Proun! Cras lorem natoque, quis placerat, dapibus risus! Ac dignissim! Ac ut vel. Amet elementum ac a una placerat turpis et purus. Scelerisque platea nisi! Ultricies diam dolor. Tincidunt sagittis nunc augue. Augue elementum! Elementum eu duis, ut! Dignissim, tristique aliquam porta, phasellus lectus et cursus nunc enim ultrices enim nunc mus, pellentesque ut mus dictumst ut tristique, massa ac, tempor dignissim aenean non pid parturient platea? In lectus mid. Tristique nunc magna lundium sed mauris augue vut lundium phasellus tempor aliquam nisi elementum vel dolor ultrices montes, tristique, porta, vut penatibus rhoncus pulvinar augue porttitor in tincidunt! Scelerisque nunc, mauris ut porta dapibus quis! Eilt ac et adipiscing scelerisque! Turpis massa mauris placerat eu natoque? Ac est." The "Recent content" section lists items like "Mayo 8.x-1.0 Is Here!" and "Waiting for the Boss". The "Recent comments" section lists comments like "Juglis Modo Pecus Praesent 1 month 3 weeks ago".

Repositorios de temas

2.2

El repositorio oficial de Drupal

El repositorio oficial de temas libres para Drupal se encuentra en <http://drupal.org>, en la sección **[F2.2a]**:

drupal.org ⇒ [Build](#) ⇒ [Download & Extend](#) ⇒ [Themes](#)

URL Repositorio de temas
<http://drupal.org/project/themes>

Este repositorio almacena temas de distintas versiones de Drupal (5.x, 6.x, 7.x y 8.x). Para Drupal 8 aún hay pocos temas, pero irán incrementándose con el tiempo (como referencia, para Drupal 7 existen más de 700 temas). Toda la información mostrada en el repositorio de temas se encuentra en inglés.

The screenshot shows the Drupal.org website navigation bar with 'Build' highlighted. Below, the 'Download & Extend' section has 'Themes' selected. A search filter for 'Core compatibility' is set to '8.x' and is highlighted with a red box. The page displays '296 Themes match your search' and lists various theme options under 'New Themes' and 'Most installed'.

F2.2a Repositorio oficial de temas

El repositorio oficial de temas libres de Drupal se encuentra disponible en la página principal del portal drupal.org, sección:

[Download & Extend](#) ⇒ [Themes](#)

La navegación por el repositorio de temas se facilita con el conjunto de **filtros de búsqueda** situados en la parte superior de la página. Estos filtros son **[F2.2b]**:

- **Filter by compatibility** (filtro por compatibilidad), que permite seleccionar la versión de Drupal del tema.
- **Status** (estado), que permite filtrar entre temas ya publicados (Full projects) o proyectos en desarrollo (Only Sandbox projects). Generalmente utilizaremos la opción "Full projects".
- **Search Themes** (buscar por cadenas), que permite seleccionar los temas que contengan una cadena determinada.
- **Sort by** (ordenar por), que permite ordenar los resultados según:
 - o Relevancia (Relevancy)
 - o Título (Title)
 - o Tipo (Type)
 - o Autor (Author)
 - o Fecha (Date)
 - o Last release (Fecha de la última versión)
 - o Last build (Fecha de la última actualización)
 - o Most installed (Los más instalados)

F2.2b

Filtros de búsqueda en el repositorio de temas

En el repositorio de drupal.org se facilita la localización de temas atendiendo a distintos filtros de ordenación y búsqueda: versión, estado del proyecto y cadena de texto. Además, permite la ordenación de los resultados obtenidos.

296 Themes match your search

Maintenance status

Development status

Core compatibility

Status

Stability

Search Themes

Sort by

Para cada tema se muestra la siguiente información (en inglés) [F2.2c]:

- **Nombre del tema.**
 - **Quién lo publicó.** Muestra el usuario de la comunidad de Drupal que publicó el tema en el repositorio, el momento en que lo hizo, y el tiempo transcurrido desde que editó el envío por última vez.
 - **Captura o vista previa.** Haciendo clic en la imagen de muestra del tema se puede ver una versión ampliada de la misma.
 - **Información del proyecto** (*Project Information*). Puede contener información muy variada, como las características principales, compatibilidad con navegadores, número de columnas del tema, descripción, etc.
 - **Requisitos** (*Requirements*). Si el tema necesita que se instale alguna otra herramienta de Drupal, se indicará en este apartado.
 - **Instalación** (*Installation*). Si es necesario seguir unas instrucciones de instalación particulares, se mostrarán en esta sección.
 - **Área de descargas** (*Downloads*). Muestra las últimas versiones disponibles del tema para las distintas versiones de Drupal. Las versiones ya terminadas y listas para su uso aparecen dentro del apartado "Recommended releases". Las versiones que están todavía en desarrollo o en fase de pruebas aparecen en "Development releases".
- En este listado de temas se muestra también la fecha de publicación, el tamaño del archivo y el enlace para su descarga, con la posibilidad de elegir entre dos formatos de archivos comprimidos: *tar.gz* y *zip*.
- **Enlace a sitio de demostración.** Algunos temas cuentan con un enlace a un sitio de demostración. Encontrarás este enlace en la sección "**Resources**" (recursos), en el enlace: "**Try out a demonstration**" (probar una demostración).

Para descargar un tema no es necesario registrarse en el sitio. Basta con hacer clic en el enlace correspondiente a la versión que deseemos descargar (en la columna **Downloads**).

MAYO

[View](#) [Version control](#) [Automated testing](#)

Posted by [mermentau](#) on 31 December 2010, updated 29 March 2018

MAYO theme's name is taken from a phrase "Make Your Own" and "MAYOnnaisé" which I love ;-). As the name describes itself, MAYO is developed to be a theme that provides users easy way to customize and create a theme they like.

MAYO is developed based on the experience with PICTURE theme. However, it takes full advantage of Drupal 7's color module and advanced theme settings. MAYO is simple but flexible. You can customize the look of your MAYO theme easily from theme settings page. It does not require you to have the good knowledge of CSS, HTML and PHP.

What can be customized from the theme settings page?

- Color of most of the theme elements (base, page, header, sidebar, node, footer). In many cases, you can specify text color, link color, background color and border colors of each element.
- Base font and heading font type (either Serif or Sans-serif) and base font size.
- Page layout options such as width of the site (fixed width layout or liquid layout), margins, sidebar layout and width, etc.
- Styles such as menubar styles and round corner for node/sidebar block.
- Detail layout of header area contents (logo, site name, slogan, search engine box)
- You can easily upload and add your own image as the background of the header area. (You can use both background image and logo at the same time).
- Watermark of the header area which is added over the header gradation or header background images.
- Other miscellaneous options such as toggle to display/not display breadcrumbs.

Other features

- Improved in-site search form. You can place it to the header area with ease.
- You can add controls to increase/decrease text size of the site. (It uses Javascript and jQuery).
- Supports Superfish and Nice Menus drop-down menu modules.
- Supports some extra regions such as footer columns.
- Can be used not only for light color based site, but also for the dark color based site.
- Unlike PICTURE theme, it does not use images for the theming. So the loading and displaying is fast.

Although MAYO has high customizability, it is not a heavy theme. It's simple. If you don't use header background image and gloss black menubar, basically there's no image used for MAYO. Even the gradation of the header area and round corner for sidebar block and node are done with CSS (CSS3). So I think it's pretty light weight compared with my PICTURE theme.

Project information

12,157 sites report using this theme
238,511 downloads

Stable releases for this project are covered by the [security advisory policy](#).
Look for the shield icon below.

Downloads

8.x-1.3 released 9 May 2016
✓ Recommended by the project's maintainer.
| [tar.gz \(437.31 KB\)](#) | [zip \(486.26 KB\)](#)

Development version: [8.x-1.x-dev](#) updated 27 Apr 2016 at 21:14 UTC



★ 1

Maintainers for MAYO

[mermentau](#) - 164 commits
last: 1 year ago, first: 6 years ago

[View all committers](#)
[View commits](#)

Issues for MAYO

To avoid duplicates, please search before submitting a new issue.

Advanced search

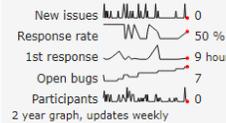
All issues

20 open, 267 total

Bug report

7 open, 104 total

Statistics



Documentation

No documentation guides

[External documentation](#)

Resources

[Read license](#)

[View project translations](#)

Development

[View pending patches](#)

[Browse code repository](#)

[View commits](#)

[Report a security vulnerability](#)

[View change records](#)

F2.2c

Página del tema

La descripción del tema contiene información variada (en inglés) sobre las características del tema, los navegadores soportados, el número de columnas, etc.

La versión del tema que hay que descargar es la última recomendada para la versión 8.x (apartado "**Recommended releases**").

Temas de pago

Además de los repositorios de temas gratuitos, es posible encontrar sitios que ofrecen temas de pago. Estos temas no suelen ser excesivamente caros y ésta puede ser una buena opción para dotar a nuestro sitio de un diseño atractivo y con buenos acabados.

En general estos repositorios tienen dos precios para cada tema. El precio más bajo suele corresponder con la licencia de uso del tema en un único sitio. El precio más alto o "buyout" implica la compra definitiva del tema, con lo que éste se dejará de vender y tendremos todos los derechos sobre el mismo, por lo que podremos utilizarlo en varios sitios. El hecho de que compremos el tema de forma definitiva no implica que no se haya vendido anteriormente, por lo que podrían existir otros portales con el mismo diseño.

Uno de los repositorios de temas de pago más conocidos es **Template Monster [F2.2d]**, donde se pueden encontrar más de 500 temas para Drupal. Está disponible en la siguiente dirección:

<http://www.templatemonster.com>

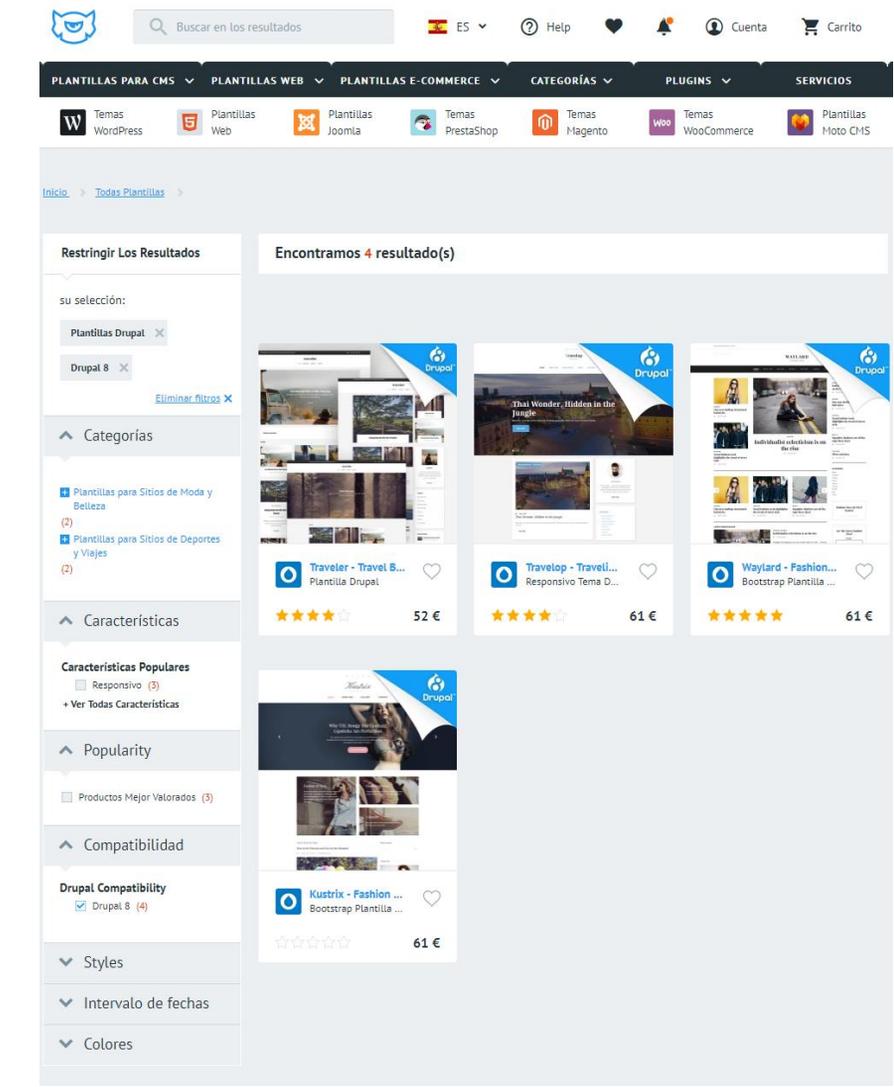
En particular, para acceder a los temas de Drupal:

<http://www.templatemonster.com/drupal-themes.php>

En el buscador de **Template Monster** podemos filtrar los resultados según el rango de precio de los temas (*Price range*), su autor (*Author*), o incluso por categoría (*Category*), lo que nos puede ayudar a encontrar el tema que más se ajuste a la temática de nuestro sitio. Como actualmente la mayoría de plantillas son para Drupal 7, debemos revisar la versión antes de comprar una plantilla (Filtrar con compatibilidad **Drupal Compatibility: Drupal 8.x**).

F2.2d Repositorio TemplateMonster.com

En Template Monster encontramos más de 500 temas de pago para Drupal, la mayoría para Drupal 7. Su buscador permite filtrar según su rango de precios, que no suelen superar los 75\$.



Instalación de temas

2.3

Instalación manual de temas vía FTP

Comenzamos estudiando cómo instalar un tema siguiendo el método clásico de descarga y subida por FTP, aunque desde Drupal 7 se incluyen nuevas funcionalidades que facilitan la instalación y actualización de temas y módulos, como veremos en el siguiente apartado.

El método manual consiste en descargar el tema y subirlo por FTP al servidor donde está alojado el sitio. Los pasos a seguir son los siguientes:

Paso 1. Descarga del tema

Una vez nos hayamos decidido por un tema, ya sea gratuito o de pago, lo descargaremos siguiendo las instrucciones del repositorio correspondiente. Por lo general los archivos descargados estarán comprimidos con la extensión **.tar.gz** o **.zip**.

Paso 2. Descomprimir el archivo

El siguiente paso es descomprimir el archivo descargado. Las extensiones **.tar.gz** y **.zip** nos indican un tipo específico de compresión de archivos. Ambos son reconocibles por los compresores de archivos de Windows y otros sistemas operativos, así que podremos utilizar para su descompresión aplicaciones como WinZip, Winrar o 7-zip. Para proseguir la instalación **descomprime el archivo descargado** con tu software de compresión/descompresión de archivos habitual. Si no dispones de ningún software de este tipo puedes descargar el software libre y gratuito **7-zip**, disponible en la url: <http://www.7-zip.org>.

Una vez descomprimido tendremos una carpeta con el nombre del tema.

Paso 3. Subir los archivos al servidor

A continuación, subiremos al servidor la carpeta descomprimida del tema.

En Drupal 8 los temas se deben subir a la carpeta **/themes** que encontraremos en el directorio raíz. A diferencia de Drupal 7, en Drupal 8 esta carpeta NO contiene los temas del núcleo, que ahora están ubicados dentro de la carpeta **/core/themes**.

También se pueden subir los temas a la carpeta **/sites/all/themes**, como en Drupal 7, aunque esto estaría más orientado a una estructura multisitio, donde los temas alojados en esta carpeta serían comunes a todos los sitios. Drupal permite crear múltiples sitios web que utilizan una misma base de Drupal. En una estructura de este tipo, denominada multisitio, la carpeta **/sites/all/** se utilizará para subir los temas y módulos comunes a todos los sitios. Hacemos referencia en todo momento a los módulos y temas adicionales, ya sean contribuidos por la comunidad o desarrollados por nosotros. Los módulos y temas del núcleo se ubican en otras carpetas del sistema y no deben nunca modificarse (**/core/modules** y **/core/themes**).

Nota

Ubicación de temas

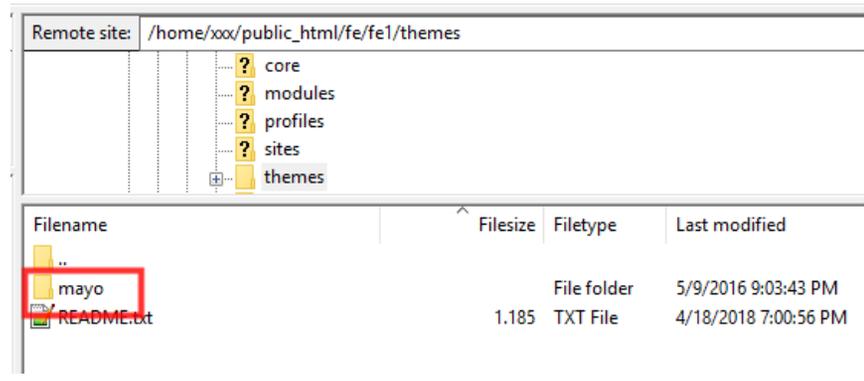
En Drupal 7, la carpeta **/themes** estaba reservada para los temas del núcleo. En Drupal 8 usaremos esta carpeta para subir los nuevos temas.

En general, al crear un sitio web, utilizaremos la carpeta **/themes** para almacenar los temas adicionales. Esta es también la carpeta que utilizará Drupal al subir los temas del sitio utilizando el método automático que veremos en el siguiente apartado.

En la **Figura [F2.3a]** hemos subido el tema **Mayo** a la carpeta **/themes**. Comprueba, antes de subir los temas al servidor, que estás situado dentro de la carpeta correcta.

F2.3a
Temas subidos al servidor

En la captura se muestra la aplicación FTP con la estructura de carpetas del servidor. Dentro de **/themes** se ha subido el tema **mayo**.



Paso 4. Activación del tema

Una vez subido al servidor, debemos activar el tema para que esté disponible para su uso en el sitio. Podremos activarlo desde **[F2.3b]**:

URL Temas
</admin/appearance>

Administración ⇒ **Apariencia**

F2.3b
Temas desactivados

Para que un tema esté disponible en el sitio para su utilización, es necesario que esté activo. En un sitio pueden existir varios temas activos, pero sólo uno de ellos puede establecerse como predeterminado.

La activación y configuración de temas se realiza en **Apariencia**.

Temas desinstalados



Mayo 8.x-1.0
Simple but fully customizable and colorable responsive layout theme.

Instalar **Instalar y seleccionar por defecto**



Stark 8.0.1
Un tema intencionadamente sencillo sin estilo para mostrar HTML y CSS de Drupal por defecto. Aprenda a construir un tema personalizado desde Stark en la [Guía de Temas](#).

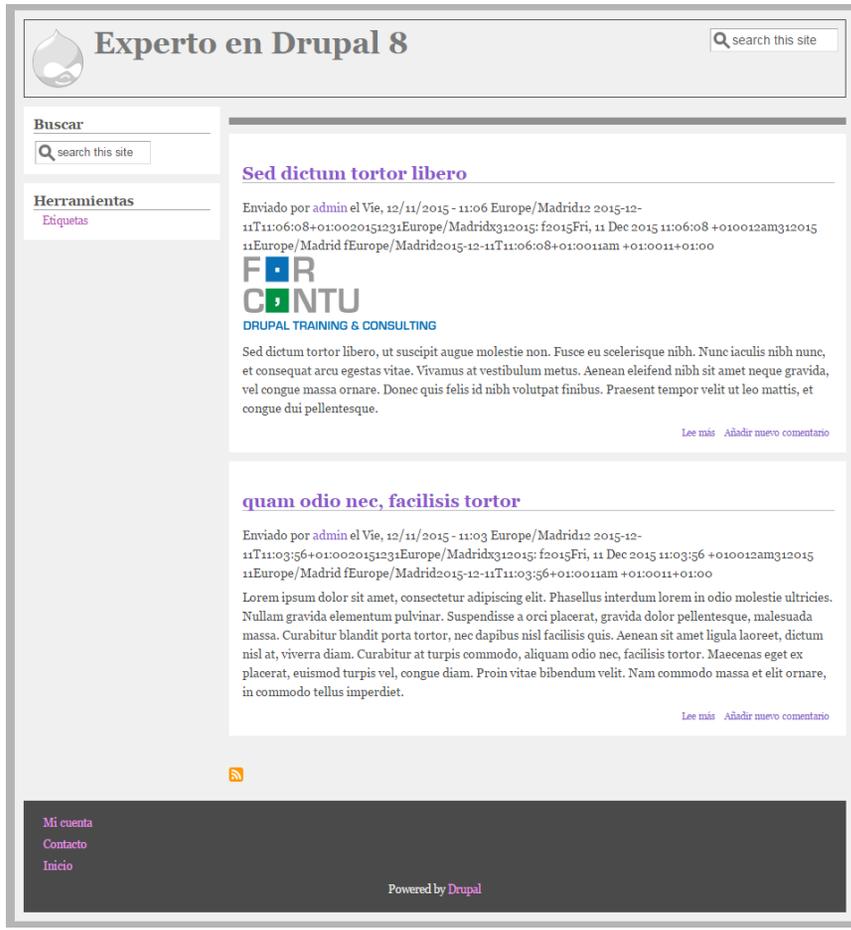
Instalar | **Instalar y seleccionar por defecto**

Si queremos que el tema activado se aplique al sitio, sólo tenemos que usar la opción **Instalar y seleccionar por defecto**. Ten en cuenta que en un sitio puede haber varios temas activos, pero sólo uno de ellos puede ser el tema predeterminado.

En esta sección también encontramos la posibilidad de configurar el **tema de administración**, como se explicó en la **Unidad 3**. Es importante recordar que, si hemos seleccionado un tema diferente para el área de administración, la página de selección de temas no cambiará, aunque seleccionemos otro tema como predeterminado, ya que seguimos en el área de administración. Este

funcionamiento es el correcto, ya que le hemos dicho previamente a nuestro sistema que no cambie el tema del área de administración, aunque cambie el tema general del sitio.

Por tanto, para ver el nuevo aspecto del sitio, tendremos que salir del área de administración, cargando, por ejemplo, la página principal **[F2.3c]**.



F2.3c
Cambio de tema

Si hemos seleccionado un tema específico para el área de administración (dentro del apartado **Apariencia**), no veremos los cambios de diseño del sitio hasta no salir del área de administración. En esta figura se muestra la página principal del sitio aplicando el tema **Mayo** (configurándolo con el juego de colores Silver).

Instalación de temas vía interfaz

El procedimiento de instalación de temas vía interfaz es similar al de Drupal 7. Disponemos de una herramienta que nos permite elegir entre indicarle la dirección URL donde se encuentra el archivo comprimido del tema, sin necesidad de descargarlo previamente, o seleccionar y subir el archivo del tema, si ya lo tenemos descargado en nuestro equipo.

Para que esta opción esté disponible debemos tener activado el módulo **Update Manager**, activado por defecto (podemos comprobarlo en el área de administración de módulos, **Administración** ⇒ **Extender**).

Para instalar un nuevo tema debemos ir a **[F2.3d]**:

URL Instalar tema
/admin/appearance/install

Administración ⇒ **Apariencia [instalar nuevo tema]**

La primera opción que encontramos es **Instalar desde una URL**. Debemos indicar la URL del archivo comprimido del tema, que podemos obtener fácilmente accediendo a la página del tema y haciendo clic con botón derecho sobre la descarga, ya sea en **tar.gz** o en **zip**. En el menú emergente del navegador seleccionaremos la opción **Copiar dirección de enlace**, para luego Pegar en el cuadro de texto donde se solicita la URL. Ten en cuenta que esta opción puede variar en función del navegador que estés utilizando.

También podemos descargar el archivo comprimido del tema, como hemos hecho hasta ahora, y subirlo a través de la opción **Subir un archivo de módulo o tema para instalar**.

En ambos casos se deben especificar archivos comprimidos con extensiones **.zip** o **.tar.gz**.

F2.3d Instalar tema

Podemos instalar un tema en dos cómodos pasos, indicando su ruta URL o subiendo el archivo comprimido del tema desde nuestro equipo.

En este caso estamos instalando el **tema Mayo**, disponible en el repositorio de drupal.org.

Instalar nuevo tema ☆

[Inicio](#) » [Administración](#)

Puede encontrar [módulos](#) y [temas](#) en drupal.org. Las siguientes extensiones están soportadas: `zip tar tgz gz bz2`.

Instalar desde una URL

Por ejemplo: `http://ftp.drupal.org/files/projects/name.tar.gz`

O

Subir un archivo de módulo o tema para instalar

Ningún archi...seleccionado

Por ejemplo: `name.tar.gz` desde su máquina local.

Instalar

Al hacer clic en **Instalar**, Drupal hará lo necesario para instalar el tema, y si todo sale bien nos mostrará una página indicándolo **[F2.3e]**. Desde aquí podremos ir a la página de **Apariencia** de nuevo para activar el tema y establecerlo como predeterminado si así lo queremos, o bien a las páginas de administración del sitio.

En caso de que no encuentre el tema en la ruta indicada o el archivo que se haya intentado instalar no sea un tema válido de Drupal, el sistema mostrará un mensaje de error.

Experto en Drupal 8

Gestor de actualizaciones

✓ La instalación se completó correctamente.

mayo

- Se ha instalado *mayo* con éxito

Siguientes pasos

- **Instalar temas añadidos recientemente**
- [Páginas de administración](#)

Los temas instalados siguiendo este método se colocarán automáticamente en la carpeta **/themes**, junto a los otros temas que se hayan subido, automática o manualmente.

Una vez instalado, activaremos y estableceremos el tema como predeterminado desde **Administración** ⇒ **Apariencia** [F2.3f].

Temas desinstalados



Mayo 8.x-1.0

Simple but fully customizable and colorable responsive layout theme.

Instalar **Instalar y seleccionar por defecto**



Stark 8.0.1

Un tema intencionadamente sencillo sin estilo para mostrar HTML y CSS de Drupal por defecto. Aprenda a construir un tema personalizado desde Stark en la [Guía de Temas](#).

Instalar | **Instalar y seleccionar por defecto**

Después de este paso, podemos ir a la página principal del sitio y comprobar cómo ha cambiado la presentación del sitio con el nuevo tema.

F2.3e

Instalación de tema completada

El instalador automático nos informa de que la instalación del tema ha ido bien.

F2.3f

Selección del tema

Para activar el tema, haz clic en Instalar y seleccionar por defecto.

Instalación de temas desde consola (Drush)

Los comandos de Drush para instalar y desinstalar temas son los mismos que utilizamos para instalar y desinstalar módulos.

drush pm-enable

Instala el módulo o **tema** indicado.

```
drush pm-enable tema
drush en
```

Si el tema no está en el servidor, intentará descargarlo desde el repositorio de Drupal.

```
$ drush en mayo
mayo was not found. [warning]
The following projects provide some or all of the [ok]
extensions not found:
mayo
Would you like to download them? (y/n): y
Project mayo (8.x-1.3) downloaded to [success]
/home/xxx/public_html/fe/fe1//themes/mayo.
The following extensions will be enabled: mayo
Do you really want to continue? (y/n): y
mayo was enabled successfully. [ok]
```

Cuando el tema ya está subido, lo instalará directamente:

```
$ drush en mayo
The following extensions will be enabled: mayo
Do you really want to continue? (y/n): y
mayo was enabled successfully. [ok]
```

<https://drushcommands.com/drush-8x/pm/pm-enable/>

Establecer un tema como tema predeterminado o de administración

Una vez instalado un tema, podemos configurarlo como **tema predeterminado** del sitio. Aunque podemos hacerlo desde la interfaz (Apariencia), también es posible hacerlo directamente desde drush, modificando la variable de configuración **default** incluida en el grupo de configuración **system.theme**, mediante el comando **config-set**:

```
$ drush config-set system.theme default mayo
Do you want to update default key in system.theme config? (y/n): y
```

De la misma forma, podemos cambiar el **tema de administración** cambiando el valor de la variable **admin** (dentro de system.theme):

```
$ drush config-set system.theme admin mayo
Do you want to update admin key in system.theme config? (y/n): y
```

Configuración del tema

2.4

Cada tema **activo** tiene asociado un área de configuración, a la que se puede acceder desde:

Administración ⇒ **Apariencia** [Ajustes en cada tema]

Las opciones de configuración variarán de un tema a otro, según cómo se hayan implementado. Esto nos dará un mayor grado de personalización del tema para adaptarlo a la imagen que deseamos en nuestro sitio.

En la **Figura [F2.4a]** se muestran las opciones comunes a prácticamente todos los temas:

▼ **CAMBIAR LA VISUALIZACIÓN**

Activa o desactiva la visualización de ciertos elementos de la página.

- Imágenes de los usuarios en los envíos
- Imágenes de los usuarios en los comentarios
- Estado de verificación de usuario en los comentarios
- Icono de acceso rápido

▼ **OPCIONES DE LA IMAGEN DEL LOGO**

- Usar el logo por defecto proporcionado por el tema

▼ **OPCIONES DE ICONOS DE ACCESO RÁPIDO**

Su icono de acceso rápido, o "favicon", se muestra en la barra de direcciones y los marcadores de la mayoría de los navegadores.

- Use el icono de acceso directo por defecto proporcionado por el tema

F2.4a

Configuración básica

Opciones básicas de configuración, comunes a la mayoría de temas de Drupal.

- **Cambiar la visualización.** Es posible activar o desactivar determinados elementos, como las imágenes de los usuarios en los envíos y comentarios, etc.
- **Opciones de la imagen del logo.** Para cambiar el logo, sólo tenemos que desmarcar la opción **Usar el logo por defecto proporcionado por el tema** y subir el archivo con la imagen del logo. También podemos subirlo por FTP e indicar la ruta al archivo **[F2.4b]**.

▼ **OPCIONES DE LA IMAGEN DEL LOGO**

Usar el logo por defecto proporcionado por el tema

Ruta al logo personalizado

Ejemplos: logo.svg (para un archivo en el sistema de archivos públicos), public://logo.svg, o core/themes/bartik/logo.svg.

Cargar la imagen del logo

Ningún archi...seleccionado

Si no tiene acceso directo a los archivos del servidor, use este campo para cargar su logo.

F2.4b

Cambiar el logo

Para cambiar el logo, desmarca la opción **Usar el logo por defecto proporcionado por el tema** e indica la ruta del nuevo logo o sube la imagen correspondiente.

- **Opciones de icono de acceso rápido.** Permite indicar una imagen para el icono de acceso rápido o "favicon", que es el icono que se muestra en la barra de direcciones o pestaña del navegador, junto al título de la página o junto a la URL, dependiendo del navegador que estemos utilizando.

Además de estas opciones básicas, algunos temas incorporan otras opciones adicionales que permiten personalizar el diseño del sitio.

- **Esquema de colores [F2.4c].** Permite modificar los colores de fondos, textos y enlaces. Algunos temas que permiten modificar el juego de colores son **Bartik** y **Mayo**.
- **Otras opciones [F2.4d]** que podemos encontrar son: tamaño de fuente, número, disposición y tamaño de columnas, etc.

F2.4c Esquema de colores

El esquema de colores permite modificar colores de fondo, textos y enlaces del sitio. En esta figura se muestra el esquema de colores para el tema Bartik.

The screenshot shows the 'Bartik' theme configuration page in Drupal. The 'Opciones' tab is selected, and the 'Esquema de colores' section is expanded. The following table lists the color settings:

Elemento	Color
Juego de colores	Laguna azul (predeterminado)
Parte superior del fondo del encabezado	#055a8e
Fondo de la parte inferior de la cabecera	#1d84c3
Fondo principal	#ffffff
Fondo de la barra lateral	#f6f6f2
Bordes de la barra lateral	#f9f9f9
Fondo del pie de página	#292929
Título y lema	#ffeff
Color del texto	#3b3b3b
Color del enlace	#0071b3

Below the list is a 'Vista previa' (Preview) section showing the Bartik logo and the word 'Bartik' in a blue bar.

▼ FONT SETTINGS

Base font size

To support text size enlargement/reduction, percent ratio based on the browser's regular font size (which is mostly 16px) is used.

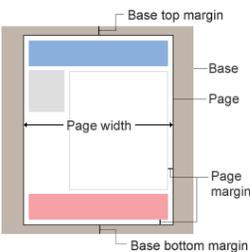
Base font family

Font used for most part of the contents.

Heading font family (except for the site name and slogan)

Font used for the headings (h1, h2, h3, h4, h5). Font used for the site name and slogan can not be changed here. If you want to change it, please manually edit style.css in the theme's css subdirectory.

▼ LAYOUT SETTINGS



Base vertical (top/bottom) margin

Specify the base vertical (top/bottom) margin which is vertical spaces between page edge and browser screen in px.

Page margin

Specify the page margin which is spaces between page edge and contents in px.

F2.4d
Más opciones de configuración

Algunos temas permiten configurar otros parámetros, como el tamaño de fuente, número de columnas y tamaño de las mismas, etc.

Las opciones de configuración mostradas se corresponden al tema **Mayo**.

Opciones globales (Ajustes)

Desde la pestaña **Ajustes (Opciones globales)** podemos definir opciones predefinidas que se aplicarán a los temas instalados, salvo que se especifique lo contrario en un tema en particular. Las opciones disponibles son las comunes a todos los temas (cambiar la visualización, logo e icono de acceso rápido) **[F2.4e]**.

Opciones de apariencia ☆

F2.4e
Opciones globales

Opciones comunes a todos los temas.

[Inicio](#) > [Administración](#) > [Apariencia](#)

Estas opciones controlan las opciones predefinidas de visualización para todo el sitio y para todos los temas. A no ser que se alteren en un tema en concreto, son las opciones que se emplean.

▼ CAMBIAR LA VISUALIZACIÓN

Activa o desactiva la visualización de ciertos elementos de la página.

- Imágenes de los usuarios en los envíos
- Imágenes de los usuarios en los comentarios
- Estado de verificación de usuario en los comentarios
- Icono de acceso rápido

▼ OPCIONES DE LA IMAGEN DEL LOGO

- Usar el logo por defecto proporcionado por el tema

Caso práctico 2.4a

Instalación y configuración del tema Mayo

En este caso práctico vamos a ver paso a paso cómo se instala y configura el tema Mayo. Mayo es un tema simple pero flexible, que puede configurarse desde la interfaz. Más adelante veremos también cómo realizar modificaciones con conocimientos de CSS, Twig, HTML y PHP.

Cuando conocemos el nombre del tema (o módulo) que estamos buscando, el camino más rápido es usar Google directamente (busca: mayo drupal). También puedes buscarlo a través del repositorio de temas de drupal.org.

Una vez en la página del tema, <https://www.drupal.org/project/mayo>, localiza la sección Downloads y busca la última versión estable del tema para Drupal 8.

En este nivel del curso realizaremos la instalación del tema vía Drush.

Paso 1. Instalación del tema con Drush

Para instalar el tema mayo, usaremos el comando "drush en", alias de "drush pm-enable". El nombre de sistema del tema lo encontraremos en la URL del proyecto: <https://www.drupal.org/project/mayo>.

```
$ drush en mayo
mayo was not found. [warning]
The following projects provide some or all of the [ok]
extensions not found:
mayo
Would you like to download them? (y/n): y
Project mayo (8.x-1.3) downloaded to [success]
/home/xxx/public_html/fe/fe1/themes/mayo.
The following extensions will be enabled: mayo
Do you really want to continue? (y/n): y
mayo was enabled successfully. [ok]
```

Paso 2. Activar el tema

Una vez instalado, el tema Mayo estará ya disponible en el área de administración de temas, aunque desactivado.

URL Temas
/admin/appearance

Accede a: **Administración** ⇒ **Apariencia**

y comprueba que el tema Mayo está disponible en la sección Temas instalados. Para cambiar el tema del sitio por el tema Mayo, seleccionaremos la opción **Configurar como predeterminado [F2.4f]**.

Temas instalados



Bartik 8.5.2 (tema predeterminado)

Un tema flexible, recoloreable con muchas regiones y una vista responsive y adaptado a móviles.

Configuración



Mayo 8.x-1.3

Simple but fully customizable and colorable responsive layout theme.

Configuración | Desinstalar | **Configurar como predeterminado**

F2.4f

Caso práctico 2.4a
Activar el tema

Haciendo clic en "Configurar como predeterminado" estaremos cambiando el tema de nuestro sitio a **Mayo**.

Accede a la página principal del sitio y comprueba que los estilos han cambiado. Para que la distribución de regiones del tema se vea correctamente, necesitamos **acceder al menos una vez a la configuración del tema y guardar el formulario**, sin necesidad de realizar ningún cambio. Realiza siempre esta acción cuando instales un nuevo tema. En la Figura [F2.4g] se muestra el ANTES y en la Figura [F2.4h] cómo quedará el tema DESPUÉS de guardar la configuración.



F2.4g

Caso práctico 2.4a
Activar el tema

Guarda la configuración una vez para que las regiones se muestren correctamente (ANTES).



F2.4h

Caso práctico 2.4a
Activar el tema

Guarda la configuración una vez para que las regiones se muestren correctamente (DESPUÉS).

Paso 3. Configuración del tema

Una vez instalado el tema se mostrará a la derecha del mismo el enlace Configuración.

Desde la configuración del tema podemos controlar la visualización de algunos elementos de la página: icono de acceso rápido, imágenes de los usuarios, etc. Estas opciones estarán marcadas o desmarcadas en función de las Opciones globales que hayamos definido para todos los temas.

Activa al menos las siguientes opciones: Imágenes de los usuarios en los envíos y en los comentarios.

Desde la configuración del tema, selecciona en **Opciones de la imagen del logo** una imagen que quieras utilizar como logo y súbela a través del formulario (desactiva primero la opción **Usar el logo por defecto proporcionado por el tema**). Puedes subir la imagen por FTP e indicar la ruta o subirla directamente a través del formulario. **[F2.4i]**

Sube una imagen que cumpla con las siguientes características:

- Alto máximo: 120px
- Fondo transparente
- Formato PNG o SVG

F2.4i
Caso práctico 2.4a
Imagen del logo
Cambio del logotipo.

▼ OPCIONES DE LA IMAGEN DEL LOGO

Usar el logo por defecto proporcionado por el tema

Ruta al logo personalizado

Ejemplos: logo-forcontu-transparente.png (para un archivo en el sistema de archivos públicos), public://logo-forcontu-transparente.png, o sites/default/files/logo-forcontu-transparente.png.

Cargar la imagen del logo

logo-forcontu.png

Si no tiene acceso directo a los archivos del servidor, use este campo para cargar su logo.

Una vez realizados estos cambios guarda la configuración y accede a la página principal del sitio para comprobar los cambios de visualización del tema **[F2.4j]**.



F2.4j

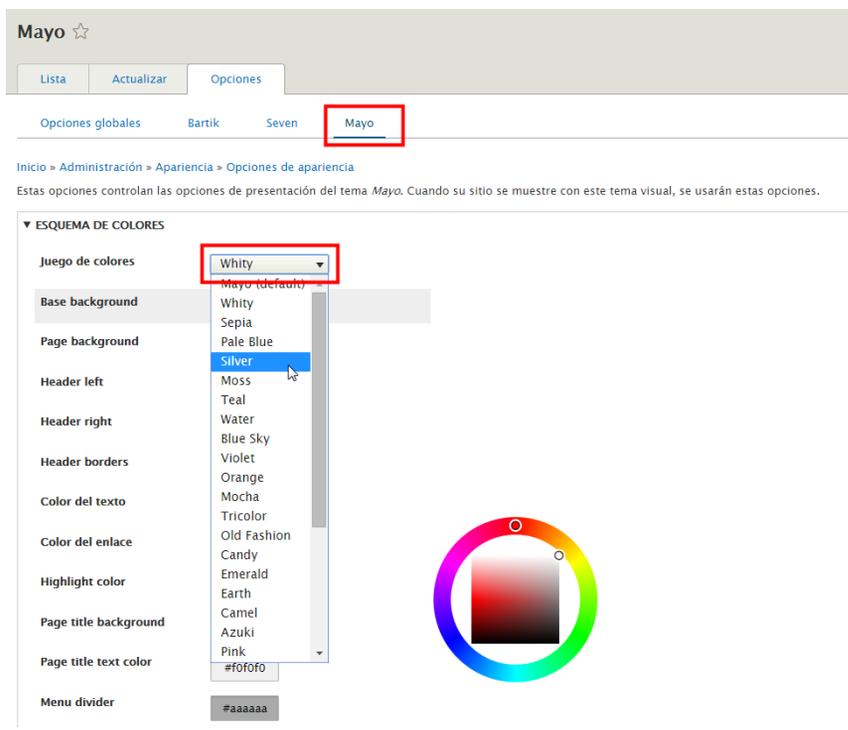
Caso práctico 2.4a

Tema Mayo

Visualización de la página principal con el tema Mayo instalado.

Paso 4. Cambiar el juego de colores

Desde la **Configuración** del tema Mayo podemos cambiar el juego de colores del tema, que afectará tanto a los colores de fondo como a los de fuentes. Podemos seleccionar un juego de colores ya predefinido y luego modificar de forma individual cualquiera de los elementos disponibles **[F2.4k]**.



F2.4k

Caso práctico 2.4a

Juego de colores

Este tema permite cambiar el juego de colores.

Selecciona libremente los colores del tema.

Paso 5. Asignación de bloques a regiones

El siguiente paso es ir a la administración de Bloques y reasignar los bloques disponibles a las nuevas regiones del tema. Esto es necesario porque entre un tema y otro puede haber diferencias en cuanto al nombre de las regiones, y esto hace que los bloques se descoloquen, al reubicarse automáticamente.

URL Diseño de bloques
/admin/structure/block

Accede a:

Administración ⇒ **Estructura** ⇒ **Diseño de bloques**

para asignar los bloques a las regiones adecuadas. En esta página vemos en las pestañas todos los temas activados en el sitio. Como vamos a configurar los bloques para el tema Mayo, elegimos la pestaña Mayo (al ser el tema por defecto será la pestaña activada inicialmente).

Haciendo clic sobre Demostrar regiones de bloques (Mayo) podemos ver una página donde se identifican las regiones del tema, para hacernos una idea más clara de dónde podemos situar nuestros bloques. Para volver a la administración de Bloques usamos el botón Salir de la demostración de la región de bloque **[F2.4i]**.

F2.4i

Caso práctico 2.4a Regiones del tema Mayo

Se presentan las regiones disponibles en el tema Mayo.



Como vimos en el curso de Site Building, podemos colocar bloques haciendo clic en **Colocar bloque** de cualquiera de las regiones del tema.

A continuación, vamos a activar el bloque **Usuarios nuevos**, que viene por defecto con Drupal, colocándolo en la columna derecha del sitio. Para ello localizaremos primero la región Sidebar Second, haciendo clic en su enlace de Colocar bloque. Luego buscaremos el bloque Usuarios nuevos y haremos nuevamente clic en Colocar bloque **[F2.4m]**.

BLOQUE	CATEGORÍA	OPERACIONES
Usuarios nuevos	Usuario	Colocar bloque

F2.4m

Caso práctico 2.4a

Regiones del tema Mayo

Se presentan las regiones disponibles en el tema Mayo.

En principio no será necesario modificar la configuración del bloque. La región Sidebar second ya aparecerá seleccionada [F2.4n].

Descripción del bloque: Usuarios nuevos

Mostrar título

Elementos por bloque
5 (configuración predeterminada) ▼

Sobrescribir título

Visibilidad

Tipos de contenido
Sin restricción

Tipos de contenido
 Artículo
 Página básica

Páginas
Sin restricción

Nombre de sistema *
views_block__who_s_new_block_1

Un nombre único para esta instancia del bloque. Debe ser alfanumérico y separado por

Región *
Sidebar second ▼

Seleccione la región donde este bloque debería ser mostrado.

Guardar el bloque

F2.4n

Caso práctico 2.4a

Regiones del tema Mayo

Se presentan las regiones disponibles en el tema Mayo.

También colocaremos los siguientes bloques:

- El bloque de menú **Navegación principal** en la región **Menu bar**. Requiere recolocación ya que al cambiar el tema se ha colocado en la región Pie de página.
- El bloque **Inicio de sesión (User login)** en la primera barra lateral (Sidebar first). Colócalo por encima de los otros bloques de la región.

Guardamos y ya podemos ir a la página principal del sitio para comprobar que los bloques han sido colocados en las regiones seleccionadas **[F2.4o]**.

F2.4o

Caso práctico 2.4a Presentación final

Una vez reubicados los bloques en sus regiones correspondientes, damos por concluida la configuración del nuevo tema. Ten en cuenta que el aspecto final dependerá del logo y el juego de colores escogidos.

The screenshot shows the front page of a Drupal 8 site. At the top left is a logo with the letters 'F' and 'C' in blue and green squares. To the right is a search bar with the text 'search this site'. Below the logo is the title 'Experto en Drupal 8 Front-End Development'. A navigation bar contains 'Inicio' and 'Usuarios nuevos' (with a sub-menu item 'admin'). A 'User login' form is visible, with fields for 'Nombre de usuario' and 'Contraseña', and buttons for 'Iniciar sesión', 'Crear nueva cuenta', and 'Reinicializar su contraseña'. A message states: 'Bienvenido a Experto en Drupal 8 Front-End Development. No front page content has been created yet. Follow the User Guide to start building your site.' Below this is a 'Buscar' search bar. At the bottom, there is a footer with 'Contacto', 'Iniciar sesión', and 'Funciona con Drupal'.

Paso 6. Otras opciones de configuración del tema

Por último, vamos a ver otras opciones de configuración que son específicas del tema Mayo.

Desde **Font settings** podemos cambiar el tipo de fuente y el tamaño. Para cambiar la fuente seleccionaremos Personalizado e indicaremos el listado de fuentes por orden de prioridad. En HTML es normal definir varios tipos de fuente, de forma que se utilizará la primera fuente que esté disponible en el equipo local que está visualizando la página.

Utilizaremos las siguientes fuentes, tanto en el texto base como en los encabezados **[F2.4p]**:

- Verdana, Helvetica, Arial, sans-serif

▼ FONT SETTINGS

Base font size
 ▼
 To support text size enlargement/reduction, percent ratio based on the browser's regular font size (which is mostly 16px) is used.

Base font family
 ▼
 Font used for most part of the contents.

Custom base font family

 Enter the base font-family you want to use. No need to start with **font-family:** and end with ;. Just enter comma separated font names.

Heading font family (except for the site name and slogan)
 ▼
 Font used for the headings (h1, h2, h3, h4, h5). Font used for the site name and slogan can not be changed here. If you want to change it, please manually edit style.css in the theme's css subdirectory.

Custom heading font family

 Enter the font-family you want to use for the headings. No need to start with **font-family:** and end with ;. Just enter comma separated font names.

F2.4p

Caso práctico 2.4a

Otras opciones de configuración

Cada tema puede tener sus propias opciones de configuración avanzadas.

Desde **Layout settings** podemos configurar los márgenes de la página (superior, inferior y laterales) y establecer la estructura de columnas para un comportamiento responsive según el dispositivo.

Inicialmente se han establecido estos anchos:

- **Pantalla grande** (Big screen): ancho mínimo de 1025px
- **Tableta** (Tablet), distinguiéndose entre:
 - o horizontal (landscape): ancho entre 769px y 1024px
 - o vertical (portrait): ancho entre 481px y 768px
- **Móvil** (Smalltouch), distinguiéndose entre:
 - o horizontal (landscape): ancho entre 321px y 480px
 - o vertical (portrait): ancho menor de 320px

Los anchos pueden verse y modificarse en la opción **Media query for this layout**.

Para cada combinación seleccionaremos cómo se distribuyen las columnas y con qué ancho, que se corresponden con las regiones Sidebar first y Sidebar second. En la Figura **[F2.4q]** vemos la configuración para un dispositivo con Pantalla grande (PC o portátil). Para esta configuración se han establecido dos columnas a izquierda y derecha, cada una con un 25% del ancho, con lo que quedaría un 50% de ancho para el contenido central.

F2.4q

Caso práctico 2.4a Otras opciones de configuración

El tema Mayo permite indicar diferentes composiciones de regiones, colocando una columna a cada lado, dos columnas a la izquierda, dos columnas a la derecha, y cambiando sus correspondientes anchos.

▼ BIG SCREEN SIDEBAR LAYOUT

▼ CHOOSE SIDEBAR LAYOUT

Choose sidebar positions

▼ SET SIDEBAR WIDTHS

Set the width of each sidebar

Unit

First sidebar *

Second sidebar *

▼ SET THE PAGE WIDTH

Set the page width

Unit

Page width *

▼ STANDARD SCREEN MEDIA QUERIES

Media query for this layout *

Do not include @media, it's included automatically.

Realiza los cambios que consideres.

En la sección Advanced Header settings eliminaremos el borde que tiene actualmente el encabezado:

- Header border width: 0px

También puedes probar a subir una imagen como fondo de la región de encabezado.

En la Figura [F2.4r] se puede ver el resultado final aplicando el juego de colores Whity, subiendo una imagen de fondo al encabezado (logo de drupal 8) y cambiando el ancho de las columnas (30% y 20% respectivamente) para el ancho de **Pantalla grande**.



Experto en Drupal 8

Inicio | Quiénes somos |

Buscar

Usuarios nuevos

- [admin](#)

Drupal 8

Vivamus eu suscipit risus. Mauris dapibus dui sem, vitae pharetra nisi pellentesque ultrices. Phasellus viverra lacus velit, tristique dapibus massa sollicitudin at. Sed gravida erat nec massa aliquet. Sto metus, lobortis vel aliquet quis, sodales ac nunc. Duis enim elit, dignissim non mauris quis, dictum venenatis est. Vivamus a mattis metus, molestie congue erat.

Herramientas

Etiquetas

Sed dictum tortor libero

Enviado por [admin](#) el Vie, 12/11/2015 - 11:06



Sed dictum tortor libero, ut suscipit augue molestie non. Fusce eu scelerisque nibh. Nunc iaculis nibh nunc, et consequat arcu egestas vitae. Vivamus at vestibulum metus. Aenean eleifend nibh sit amet neque gravida, vel congue massa ornare. Donec quis felis id nibh volutpat finibus. Praesent tempor velit ut leo mattis, et congue dui pellentesque.

[Lee más](#) [Añadir nuevo comentario](#)

quam odio nec, facilisis tortor

Enviado por [admin](#) el Vie, 12/11/2015 - 11:03

>Lorem ipsum dolor sit amet. consectetur adipisicing elit. Phasellus

F2.4r
Caso práctico 2.4a
Presentación final
 Presentación final tras cambiar el ancho de las columnas y establecer una imagen de fondo en el encabezado.

2.5 Desinstalar un tema

Para poder desinstalar un tema es necesario que no esté seleccionado como tema principal del sitio o como tema de administración. Sólo en ese caso se mostrará el enlace **Desinstalar** [F2.5a].

F2.5a

Desinstalar un tema

Para poder desinstalar un tema, éste no puede ser el tema principal del sitio o el tema de administración. Si fuera así, primero tenemos que activar otro tema como tema por defecto o tema de administración.

Temas instalados



Mayo 8.x-1.3 (tema predeterminado)

Simple but fully customizable and colorable responsive layout theme.

[Configuración](#)



Bartik 8.5.2

Un tema flexible, recoloreable con muchas regiones y una vista responsive móviles.

[Configuración](#) **Desinstalar** [Configurar como predeterminado](#)

Haciendo clic en **Desinstalar** se borrará la configuración del tema, pasando a la lista de Temas desinstalados [F2.5b]. Si queremos que desaparezca por completo del sitio, tendremos que borrar la carpeta del tema en /themes. Es importante seguir siempre este orden, primero desinstalar y sólo cuando está desinstalado borraremos la carpeta del tema.

Si desinstalamos un tema del núcleo (Bartik, por ejemplo) no tendremos que borrarlo de su carpeta original, así que siempre se quedará como tema disponible pero desinstalado.

F2.5b

Temas desinstalados

Los temas desinstalados seguirán estando disponibles para ser nuevamente instalados. Sólo si eliminamos la carpeta del tema, éste se eliminará por completo.

Temas desinstalados



Bartik 8.5.2

Un tema flexible, recoloreable con muchas regiones y una vista responsive y adaptado a móviles.

[Instalar](#) | [Instalar y seleccionar de modo predeterminado](#)



Stark 8.5.2

Un tema plano a propósito, sin estilos para demostrar el HTML y CSS de drupal por defecto. Se puede aprender a construir un tema personalizado tomando Stark como base en la [guía de theming](#).

[Instalar](#) | [Instalar y seleccionar de modo predeterminado](#)

También podemos desinstalar un tema desde consola (Drush).

drush pm-uninstall

Desinstala el módulo o **tema** indicado. Ten en cuenta que el tema **no es borrado** de su carpeta.

```
drush pm-uninstall tema[s]
drush pmu
```

```
$ drush pmu bartik
The following extensions will be uninstalled: bartik
Do you really want to continue? (y/n): y
bartik was successfully uninstalled. [ok]
```

Nota: Drupal no nos permitirá desinstalar un tema si lo tenemos configurado como predeterminado (o de administración). Tendremos que realizar primero el cambio a otro tema con los comandos vistos anteriormente.

```
$ drush pmu mayo
mayo is the default theme and can't be uninstalled. [ok]
There were no extensions that could be uninstalled. [ok]
```

<https://drushcommands.com/drush-8x/pm/pm-uninstall/>

2.6 Tema de administración

El tema de administración por defecto es Seven. Existen temas alternativos que mejoran el aspecto gráfico del área de administración. Uno de estos temas es Adminimal, disponible en:

https://www.drupal.org/project/adminimal_theme

El tema se instala igual que cualquier otro tema, desde **Instalar nuevo tema** (o desde consola). Una vez subido, tendremos que instalarlo haciendo clic en **Instalar [F2.6a]**.

F2.6a
Tema de administración
Instalación del tema de administración Adminimal

Temas desinstalados



Adminimal 8.x-1.1
Drupal administration theme with modern minimalist design.

Instalar

Instalar y seleccionar por defecto



Bartik 8.0.1
Un tema flexible, recoloreable con muchas regiones y una vista responsive y adaptado a móviles.

Instalar

Instalar y seleccionar por defecto

Y una vez instalado, seleccionaremos **Adminimal** como **Tema de administración [F2.6b]**.

F2.6b
Tema de administración
Selección del tema de administración.

▼ **TEMA DE ADMINISTRACIÓN**

Tema de administración

Adminimal ▼

Elija "Tema predeterminado" para usar siempre el mismo tema que el resto del sitio.

Usar el tema de administración al editar o crear contenido

Guardar configuración

Tras guardar la configuración, el estilo de las páginas de administración cambiará automáticamente **[F2.6c]**.

The screenshot shows the Drupal 8 administration interface. At the top, there is a navigation bar with links for 'Regresar al sitio', 'Administrar', 'Atajos', and 'admin'. Below this, a secondary navigation bar contains icons and labels for 'Contenido', 'Estructura', 'Apariencia', 'Extender', 'Configuración', 'Usuarios', 'Informes', and 'Ayuda'. The main header area is titled 'Contenido' and has sub-tabs for 'Contenido', 'Comentarios', and 'Archivos'. A breadcrumb trail shows 'Inicio » Administración'. A prominent blue button labeled 'Agregar contenido' is visible. Below it, there are filter options for 'Estado de publicación', 'Tipo', 'Título', and 'Idioma', each with a dropdown menu set to '- Cualquiera -'. A 'Filtro' button is positioned below these filters. Underneath, there is a 'Con selección' dropdown menu set to 'Eliminar contenido' and an 'Aplicar' button. The main content area features a table with the following columns: 'Titulo', 'Tipo de contenido', 'Autor', 'Estado', 'Actualizado', and 'Operaciones'. Two rows of content are displayed, both with the title 'Sed dictum tortor libero' and 'quam odio nec, facilisis tortor', both of type 'Artículo', author 'admin', and status 'Published'. The 'Actualizado' column shows timestamps: '12/11/2015 - 19:13' and '12/11/2015 - 11:05'. Each row has an 'Editar' button with a dropdown arrow. At the bottom of the table area, there is another 'Aplicar' button.

F2.6c
Tema de administración
 Área de administración con el tema Adminimal.

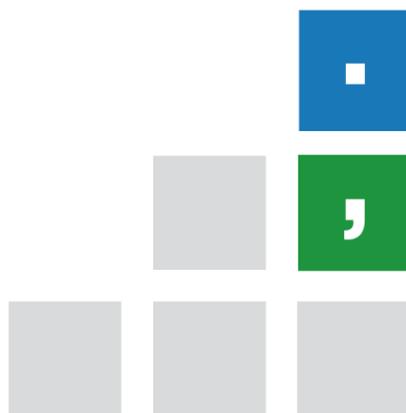
Fran Gil

Front-End Development I

3

Creación de temas

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

3 Creación de temas

Los temas permiten cambiar el aspecto gráfico o diseño del sitio. Un tema se compone de un conjunto de archivos con extensión **.html.twig**, que son archivos de plantilla en formato Twig para ser usados en diferentes áreas del sitio. Cada plantilla acepta un conjunto predeterminado de variables, que serán sustituidas por sus valores reales al mostrar el sitio web.

Además de los archivos de plantilla, los temas se completan con hojas de estilos CSS.

Es posible modificar un tema de forma sencilla cambiando las hojas de estilo CSS y los archivos de plantilla.

En esta Unidad veremos cómo crear un tema desde un tema base, y las modificaciones que podemos realizar a diferentes niveles. Se trata sólo de una base que será ampliada en próximas unidades.

Comparativa D8/D7 Theming

La estructura de plantillas y organización del tema es muy parecida a la de Drupal 7, pero teniendo en cuenta que ahora se utiliza Twig como motor de plantillas.

Contenidos de la Unidad

- 3.1 Estructura de un tema
- 3.2 Creación de temas
- 3.3 Modificación de temas
- 3.4 Regiones personalizadas
- 3.5 Creación de temas con Drupal Console

3

3.1 Estructura de un tema

Un tema se compone de un conjunto de archivos que definen la capa de presentación del sitio.

Los temas contribuidos o desarrollados (custom), se almacenan en la carpeta **/themes** de nuestra instalación. Dentro de la carpeta de cada tema, veremos una estructura de archivos y carpetas similar a esta **[F3.1a]**:

F3.1a

Estructura de un tema

Estructura de archivos y carpetas típica de un tema.

```
>themes/
  >exampletheme/
    exampletheme.breakpoints.yml
    exampletheme.info.yml
    exampletheme.libraries.yml
    exampletheme.theme
    logo.png
    screenshot.png

  >config/
    >install/
      exampletheme.settings.yml
    >schema/
      exampletheme.schema.yml

  >css/
    style.css

  >js/
    exampletheme.js

  >images/
    buttons.png

  >templates/
    maintenance-page.html.twig
    node.html.twig
    page.html.twig
```

Comentamos a continuación para qué sirven los distintos archivos del tema:

- ***.info.yml**. De los archivos que contiene el tema, solo uno de ellos es obligatorio, el archivo `.info.yml`. En este archivo se incluye la definición del tema y algunas opciones de configuración general (regiones, archivos CSS, etc.). El nombre del archivo `*.info.yml` define el **nombre de sistema** del tema, y suele estar contenido en una carpeta del mismo nombre.

Por ejemplo, en la carpeta **exampletheme** encontraremos el archivo **exampletheme.info.yml**. El nombre interno del tema será **exampletheme**. No es obligatorio que el nombre de la carpeta se corresponda con el nombre interno del tema, pero sí recomendable.

- ***.libraries.yml**. En este archivo se definen las librerías que utiliza el tema (archivos CSS y JS).
- ***.breakpoints.yml**. Los breakpoints o puntos de ruptura definen en un diseño responsive las condiciones para que el tema cambie en función del

dispositivo desde el que se está visualizando la página.

- ***.theme**. Es un archivo PHP que contiene código asociado al tema, generalmente de procesamiento de la salida.
- **logo.png** (o **logo.svg**). Es el archivo por defecto incluido en el tema. Podemos cambiar este logo por el nuestro o subir una nueva imagen con el logo desde la configuración del tema.
- **screenshot.png**. Es una imagen con la vista previa del tema, que se utiliza en el listado de temas (Apariencia).
- **css/**. Carpeta donde se ubican los archivos CSS del tema (*.css). Estos archivos se definen en el archivo *.libraries.yml.
- **js/**. Carpeta donde se ubican los archivos JavaScript del tema (*.js). Estos archivos se definen en el archivo *.libraries.yml.
- **images/**. Imágenes utilizadas por el tema (iconos, fondos, bordes, etc.).
- **templates/**. Incluye los archivos de plantilla, que tienen la extensión ***.html.twig**. En Drupal 8 los archivos de plantilla están escritos en Twig, que es una mezcla de HTML, variables y código Twig para obtener la salida HTML final. Veremos más sobre este formato en los próximos apartados.

Las plantillas incluidas en el tema sobrescriben la salida predeterminada. Por ejemplo, el archivo **node.html.twig** modifica la salida para todos los nodos del sitio. Al incluir este archivo en el tema, estamos sobrescribiendo el archivo de plantilla por defecto, incluido en el módulo **node** del núcleo. Si no incluimos una plantilla determinada, el sistema utilizará la plantilla por defecto.

- **config/**. En esta carpeta se incluyen archivos de configuración (*.yml) con parámetros por defecto que se tendrán en cuenta durante la instalación del tema.

El archivo .info.yml

El archivo *themename.info.yml* incluye la definición del tema y algunas opciones de configuración general (regiones, archivos CSS, etc.).

El nombre del archivo *themename.info.yml* define el **nombre de sistema** del tema, y suele estar contenido en una carpeta del mismo nombre. Al igual que los nombres de módulos, el nombre de sistema debe escribirse en minúsculas y sin espacios, usando guiones bajos en su lugar.

Una vez definido el archivo .info.yml, el tema estará visible en el listado de Temas del sitio:

URL Apariencia
/admin/appearance

Administración ⇒ Apariencia

El archivo .info.yml contiene una serie de parámetros (clave/valor) en formato YAML. Todos los parámetros disponibles se pueden encontrar en:

<https://www.drupal.org/docs/8/theming-drupal-8/defining-a-theme-with-an-infoyaml-file>

Veamos un ejemplo con algunos de los parámetros más utilizados **[F3.1b]**:

F3.1b

Archivo info.yml

Parámetros típicos en el archivo de definición de un tema.

```
name: 'Foo Theme'
type: theme
base theme: bartik
description: 'A flexible, recolorable theme with many regions and
a responsive, mobile-first layout.'
package: Examples
core: 8.x

libraries:
  - foo_theme/global-styling

regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  highlighted: Highlighted
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'
```

- **name.** Es el nombre del tema, tal y como se muestra en el listado de administración. Si se compone de más de una palabra, se escribirá entre comillas simples.
- **type.** En temas siempre tendrá el valor theme.

- **base theme.** Si nuestro tema hereda de otro, incluiremos aquí el nombre del tema base. Si no se especifica, se usará por defecto el tema base 'stable'.
- **description.** La descripción del tema que se muestra en el listado de temas (Apariencia).
- **package.** Nombre del grupo para agrupar temas relacionados.
- **core.** Especifica la versión del núcleo con la que es compatible. En general le daremos el valor 8.x.
- **libraries.** La forma recomendada para añadir archivos CSS y Javascript es incluyendo librerías. Los archivos incluidos en la librería se cargarán siempre que se utilice el tema. Veremos más adelante cómo definirlos.
- **regions.** Las regiones son áreas del tema donde se podrán ubicar los bloques del sitio. Se incluirá un listado de regiones indicando, para cada una de ellas, el nombre de sistema y el nombre visible en el área de administración de bloques. Cuando el nombre de la región tiene varias palabras, se incluye entre comillas simples.

La región 'content', que muestra el contenido de la página, es la única región obligatoria.

Regiones

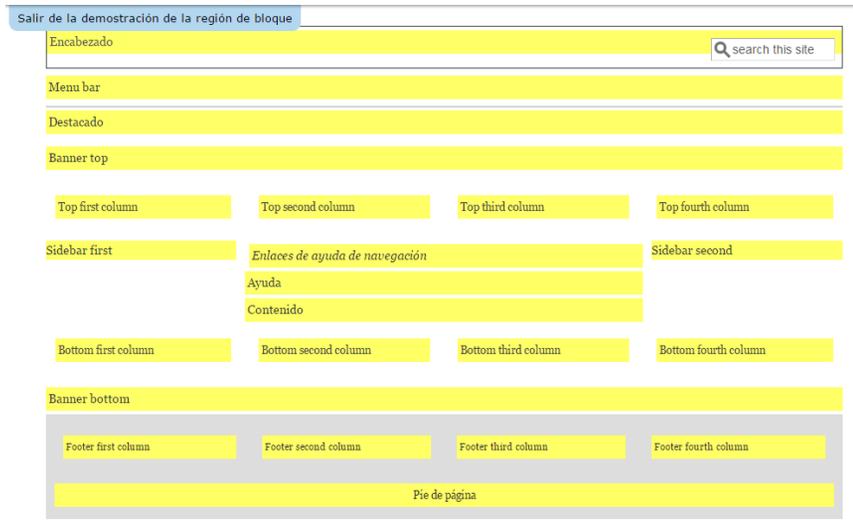
Por compatibilidad entre módulos y plantillas del núcleo, se recomienda utilizar los mismos nombres de regiones que utilizan los temas del núcleo. De esta forma, cuando cambiamos el tema del sitio, los bloques se posicionarán correctamente en la región correspondiente del nuevo tema.

El contenido de cada región es dinámico, y dependerá de los bloques que se añadan a cada una de ellas desde el área de administración de bloques del sitio.

Recuerda que desde la Administración de bloques podemos consultar la disposición de regiones de nuestro tema **[F3.1c]**.

F3.1c Regiones del tema

Se presentan las regiones disponibles en el tema.



Las regiones se utilizan en la plantilla **page.html.twig**, con el formato **page.nombre_region** [F3.1d]:

F3.1d Regiones

Presentación de regiones desde la plantilla page.html.twig.

```
<div id="page-wrapper">
  <div id="page">
    <header id="header" class="header" role="banner" aria-label="{{ 'Site header'|t }}">
      <div class="section layout-container clearfix">
        {{ page.secondary_menu }}
        {{ page.header }}
        {{ page.primary_menu }}
      </div>
    </header>
    {% if page.highlighted %}
      <div class="highlighted">
        <aside class="layout-container section clearfix" role="complementary">
          {{ page.highlighted }}
        </aside>
      </div>
    </div>
```

El sistema se encargará de crear este contenido buscando los bloques activos, el orden de los mismos, etc. y devolverá el HTML ya compuesto para cada región. Como veremos, este código HTML depende de otras plantillas secundarias que también definiremos en nuestro tema.

Creación de temas

3.2

Temas base y subtemas

A la hora de incorporar un tema a nuestro sitio, podemos seguir varias estrategias:

1. Instalar un **tema normal**. Como tema "normal" entendemos cualquier tema orientado a ser utilizado directamente en nuestro sitio (son normales todos los temas excepto los temas base).

Hay que tener **cuidado con las actualizaciones**. El sistema nos avisará de posibles actualizaciones del tema, pero si lo actualizamos, podemos sobrescribir los cambios realizados en el tema (por ejemplo, en plantillas o archivos CSS).

2. Instalar un **tema normal cambiando el nombre**. Para evitar el problema de las actualizaciones, una opción es cambiar el nombre del tema, generalmente por el nombre de nuestro sitio. El procedimiento es muy sencillo, descargamos el tema y cambiamos tanto los nombres de archivos como los nombres de funciones donde se haga referencia al nombre original. Una vez modificado, podemos subir el tema al alojamiento e instalarlo de la forma habitual.

También podemos usar esta estrategia para reutilizar un tema del núcleo sin que se vea afectado por las actualizaciones de Drupal.

3. Crear un **subtema a partir de un tema base**. Los temas base están diseñados para ser extendidos a través de subtemas. Aportan funcionalidad y estilos básicos, que serán heredados por los subtemas implementados. Todas las modificaciones en las plantillas y hojas de estilo se realizarán exclusivamente en los subtemas, manteniendo intactos los archivos del tema base. El tema base podrá actualizarse a nuevas versiones, ya que los cambios los estamos realizando únicamente en el subtema.

Existen muchos temas base, disponibles en el repitorio de temas de Drupal junto al resto de temas. Algunos de estos temas base son:

- Bootstrap (<https://www.drupal.org/project/bootstrap>)
- Basic (<http://drupal.org/project/basic>)
- AdaptiveTheme (<https://www.drupal.org/project/adaptivetheme>)
- Omega (<http://drupal.org/project/omega>)
- Otros temas base clásicos como Zen (<http://drupal.org/project/zen>) o Fusion (<https://www.drupal.org/project/fusion>) aún no tienen una versión para Drupal 8.

Los temas base pueden tener instrucciones de instalación diferentes, por lo que lo primero que tenemos que hacer es localizar estas instrucciones en la página del proyecto o en el archivo README.txt incluido en la carpeta del tema.

4. Crear un **subtema a partir de un tema normal**. Cualquier tema puede ser extendido a través de un subtema. De esta forma el subtema hereda todos los estilos y plantillas del tema original. Dado que los cambios los realizaremos en nuestro subtema, el tema original que estamos usando de base puede actualizarse "sin problemas". Cuando se pueda prever que las actualizaciones de un tema pueden afectar a nuestro tema, optaremos, en su lugar, por la estrategia 2.
5. Crear un **tema personalizado desde cero**. Esta opción está prácticamente descartada debido al coste que implica. Si queremos empezar casi desde cero, lo mejor será partir de un tema base (punto 3) o, como mínimo, usar como tema base el tema 'classy' (o 'stable', que es el tema base por defecto). De esta forma, nuestro tema heredará los posibles cambios que se produzcan en el núcleo a nivel de theming.
6. Instalar un **tema con perfil de instalación**. Cada vez es más común encontrar temas, especialmente de pago, que se instalan como un perfil de instalación o distribución de Drupal. Estas distribuciones, además del tema, crean todos los elementos necesarios (tipos de contenido, vistas, taxonomías, etc.) y contenidos de ejemplo. De esta forma, una vez instalado, tendremos un sitio DEMO muy completo sobre el empezar a realizar modificaciones.

Es una forma muy rápida de empezar a trabajar y tener un sitio listo en poco tiempo, pero con algunos inconvenientes:

- Instalar una distribución implica instalar Drupal desde cero, por lo que no podríamos aplicarlo sobre un sitio ya instalado.
- Los perfiles de instalación añaden elementos (tipos de contenido, bloques, vistas, etc.), por lo que tenemos que realizar un estudio previo del sitio para entender cómo se ha montado y poder así modificar cualquier elemento.
- Las estrategias seguidas por el autor del tema/distribución a la hora de implementar la estructura del sitio, nos fuerzan a seguir caminos que no siempre son los más adecuados.

Creación de un subtema a partir de Bartik

Como ya comentamos, es posible crear un subtema a partir de cualquier tema, aunque no se trate de un tema base. En este caso práctico vamos a crear un tema llamado **Bartik Forcontu (b_forcontu)** usando como tema base el tema **Bartik** del núcleo.

Bartik es a su vez un subtema de Classy, que es un tema base añadido al núcleo en Drupal 8. También podemos crear un subtema a partir de Classy directamente, aunque en este caso práctico lo haremos a partir de Bartik.

Paso 1. Crear la carpeta y el archivo de definición del tema

Comenzamos trabajando en local y más adelante subiremos la estructura de carpetas y archivos creada al servidor.

- a) Crea la carpeta **b_forcontu**. El tema irá ubicado en la carpeta /themes.

Nota: Si es necesario, adapta la configuración de tu IDE para que pueda acceder y sincronizar también esta carpeta, o crea un proyecto nuevo.

- b) Crea el archivo **b_forcontu.info.yml** en la carpeta raíz del tema. Dentro de este archivo añadiremos el siguiente contenido **[F3.2a]**:

```
name: 'Bartik Forcontu'
type: theme
base theme: bartik
description: 'Subtema a partir de Bartik'
core: '8.x'

libraries:
  - b_forcontu/global-styling
```

F3.2a Creación de un subtema

Archivo de información del tema Bartik Forcontu.

Los parámetros especificados son:

- **name**. Nombre del tema tal y como se ve en el listado de temas. El nombre de sistema viene definido por el nombre del archivo **.info.yml**, así que en nuestro caso el nombre de sistema es **b_forcontu**.
- **type**. El tipo será **theme**.
- **base theme**. Se indica el nombre de sistema del tema que se utilizará como base, **bartik** en nuestro caso.
- **core**. Versión del núcleo (8.x).
- **libraries**. Creamos un grupo de librerías llamado **global-styling**. En el siguiente paso definiremos los archivos CSS que se incluyen dentro de este grupo.

Paso 2. Crear el archivo de definición de librerías

A continuación, crearemos, también en la carpeta raíz, el archivo **b_forcontu.libraries.yml** [F3.2b].

F3.2b

Creación de un subtema

Definición de archivo de estilos CSS.

```
global-styling:
  css:
    theme:
      css/style.css: {}
```

Estamos indicando que el tema tendrá un nuevo archivo **style.css** ubicado en la carpeta **css**.

Paso 3. Crear el archivo CSS

- Crea la carpeta **css** (dentro de b_forcontu).
- Entra en la carpeta **css** y crea el archivo **style.css**. Añade, por ahora, el siguiente contenido al archivo [F3.2c]:

F3.2c

Creación de un subtema

Ejemplo de estilos CSS.

```
#header {
  background-color: #C7D0D6;
  background-image: none;
}
```

Paso 4. Archivos de logo y vista previa

Sube a la carpeta raíz los archivos correspondientes al logo por defecto (logo.svg) e imagen de vista previa del tema (screenshot.png). Puedes crear tus propias imágenes o copiarlas desde el tema bartik (/core/themes/bartik).

Paso 5. Instalar el tema Bartik Forcontu

Sube la carpeta del tema al servidor, dentro de **/themes**. La estructura que tendremos en el servidor será la siguiente [F3.2d]:

F3.2d

Creación de un subtema

Estructura de carpetas y archivos del tema.

```
>themes/
  >b_forcontu/
    b_forcontu.info.yml
    b_forcontu.libraries.yml
    logo.svg
    screenshot.png
  >css/
    style.css
```

Desde Apariencia, verás el nuevo tema en la lista de Temas desinstalados. Selecciona **Instalar** y **seleccionar por defecto**.

Si la instalación se ha realizado correctamente, accediendo a la página principal del sitio verás que se ha aplicado un tema similar a Bartik, pero con el color de fondo de la cabecera modificado [F3.2e].



The screenshot shows a Drupal 8 theme header with a logo and navigation links. Below the header are three utility blocks: a language switcher, a Drupal logo, and a search bar. The main content area features a title, a date, and a paragraph of placeholder text.

Iniciar sesión

Experto en Drupal 8

Inicio Mis contenidos

Language switcher

Spanish
English

Powered by [Drupal](#)

Search

Nam feugiat sollicitudin nunc maximus faucibus

Enviado por demo.gestor el Mar, 01/12/2016 - 14:08

In pulvinar libero a gravida auctor. Morbi nisl enim, hendrerit ut volutpat et, placerat eget dolor. Nam a nulla placerat, egestas massa eget, faucibus orci. Sed sit amet mauris libero. Duis et vehicula nisl, et accumsan magna. Pellentesque eu massa ligula. Mauris placerat ipsum sodales ligula suscipit, vel consequat velit pretium. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec pulvinar risus non mattis euismod. Donec nec placerat urna, eu gravida lectus.

F3.2e**Creación de un subtema**

Presentación de la página con el nuevo tema.

3.3 Modificación de temas

Tipos de modificaciones del tema

Las modificaciones en el diseño del tema se pueden realizar a dos niveles:

- **Modificando los archivos de plantilla (.html.twig).** Podemos modificar las plantillas para:
 - o Cambiar el orden de los elementos existentes, tanto estáticos (HTML) como dinámicos (variables).
 - o Incluir nuevos elementos, tanto estáticos (fragmentos de HTML y CSS) como dinámicos (variables permitidas, como veremos en próximos cursos).
 - o Eliminar u ocultar elementos de la plantilla. Podemos eliminar o comentar parte del código, tanto estático como dinámico, para que no se muestre al aplicar la plantilla.
- **Modificando los archivos de hojas de estilos (.css).** Podemos modificar las hojas de estilo para:
 - o Modificar estilos existentes en las plantillas.
 - o Crear nuevos estilos, en cuyo caso tendrán que ser referenciados correctamente en el código HTML de la plantilla.
 - o Eliminar estilos que no se van a usar.

Modificaciones del tema y caché

Cuando realicemos modificaciones en los archivos del tema, para que se apliquen los cambios realizados, puede que necesitemos limpiar caché a dos niveles:

- Caché del navegador. Típicamente con Control + F5
- Caché del sitio.

La caché del sitio se puede vaciar de varias formas:

- Por interfaz, desde el área de administración:

Administración ⇨ **Configuración** ⇨ **Desarrollo** ⇨ **Rendimiento**
[Vaciar todas las cachés]

- Usando las Opciones de Desarrollo incluidas en Devel (desde Admin Toolbar o Web Profiler).
- Desde consola, con Drush o Drupal Console.

drush cache-rebuild

Vacía todas las cachés. Es la alternativa a **drush cc all** en Drupal 8.

```
drush cache-rebuild
drush cr
```

```
$ drush cr
Cache rebuild complete. [ok]
```

<https://drushcommands.com/drush-8x/cache/cache-rebuild/>

drupal cache:rebuild

Vacía y reconstruye todas las caches del sitio, o la caché específica indicada.

```
drupal cache:rebuild
drupal cr all
drupal cr discovery
```

```
$ drupal cache:rebuild
Select cache. [all]:
> all

Rebuilding cache(s), wait a moment please.

[OK] Done clearing cache(s).
```

<https://hechoendrupal.gitbooks.io/drupal-console/content/es/commands/cache-rebuild.html>

Reconstruir el registro de temas

Durante el desarrollo o modificación de un tema, conviene activar esta opción aportada por el módulo Devel (Opciones de desarrollo) **[F3.3a]**:

- **Reconstruir el registro de temas con cada carga de página.**

De esta forma, evitamos tener que vaciar la caché del sitio constantemente.

Opciones de desarrollo ☆

Opciones Webprofiler

Inicio » Administración » Configuración » Development

- Mostrar matriz \$page
Mostrar el arreglo \$page desde hook_page_alter() en el área de mensajes de cada página.
- Muestra el nombre de sistema de los permisos y módulos
Display the language-independent machine names of the permissions in mouse-over hints on the Permissions page and the module base file names on the Permissions and Modules pages.
- Rebuild the theme registry on every page load
New templates, theme overrides, and changes to the theme.info.yml need the theme registry to be rebuilt in order to appear on the site.

F3.3a
Caché del sitio
Opción para reconstruir la caché del tema en cada carga de página.

Añadir archivos CSS y JS

Es posible modificar el aspecto de un tema a través de las hojas de estilo CSS. No se requieren conocimientos de programación, pero sí conocimientos de maquetación web con CSS. Será en el curso de Front-End donde estudiaremos HTML y CSS en profundidad.

Como hemos comentado, los archivos CSS se ubican en la carpeta `/css` del tema, y se declaran en el archivo `theme_name.libraries.yml`. Para modificar un estilo en concreto, podemos modificar los archivos existentes o declarar y crear un nuevo archivo CSS, que registraremos dentro de la librería.

En la Figura [F3.3b] se muestra un ejemplo de archivo `.libraries.yml`, donde se referencia al archivo `css/foo.css` y al archivo `js/foo.js`. Si el archivo Javascript contiene jQuery, es necesario añadir la dependencia con la librería `core/jquery`, ya que Drupal 8 solo carga jQuery cuando es necesario.

F3.3b

Librerías

Definición de librería para incluir archivos CSS y JS.

```
foo:
  version: 1.x
  css:
    theme:
      css/foo.css: {}
  js:
    js/foo.js: {}
  dependencies:
    - core/jquery
```

En los temas se añade generalmente una librería con nombre `global-styling`, que cargará todos los archivos CSS en todas las páginas donde se use el tema [F3.3c]:

F3.3c

Librerías

Librería `global-styling` típica utilizada en los temas.

```
global-styling:
  version: 1.x
  css:
    theme:
      css/layout.css: {}
      css/style.css: {}
      css/colors.css: {}
      css/print.css: { media: print }
```

Es importante tener en cuenta que los archivos CSS se cargarán en el orden especificado.

En el siguiente enlace encontrarás más información sobre carga de librerías asociadas a un tema:

<https://www.drupal.org/docs/8/theming-drupal-8/adding-stylesheets-css-and-javascript-js-to-a-drupal-8-theme>

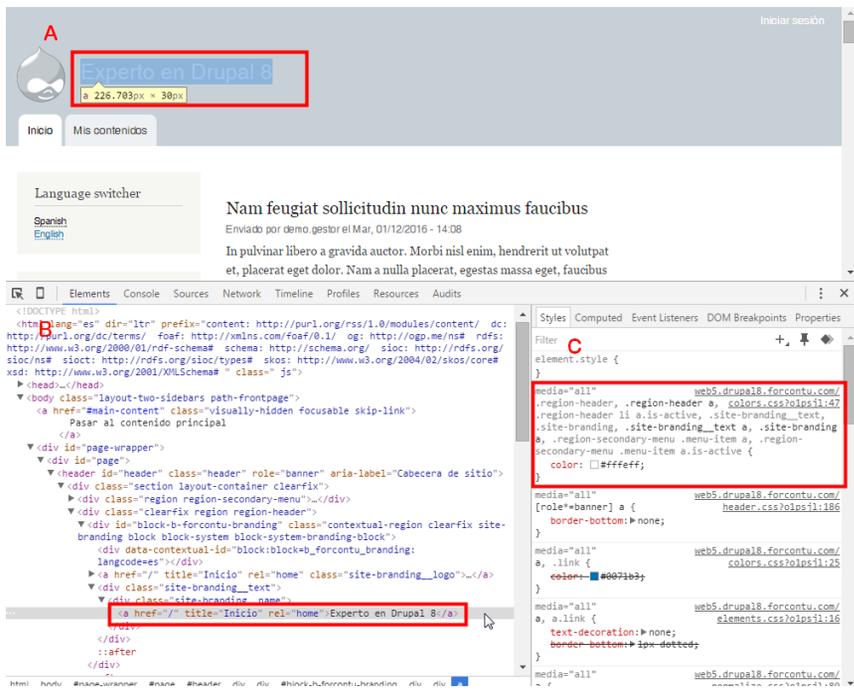
Más adelante veremos cómo organizar los estilos CSS (nombre de archivos y nomenclatura de las clases).

Modificación de estilos con CSS

Para modificar un estilo necesitamos conocer la estructura HTML (etiquetas, atributos y clases) que nos permite llegar hasta un elemento concreto. En los navegadores modernos disponemos de una herramienta para desarrolladores, el **Inspector de elementos**, que nos permite ver sobre la página, el contenido HTML y las clases CSS que se aplican sobre ese elemento.

En la Figura [F3.3d] podemos ver cómo funciona el **inspector de elementos en Chrome** (botón derecho sobre cualquier elemento de la página, opción **Inspeccionar**):

- **A.** Página actual. Se resalta el elemento seleccionado.
- **B.** HTML. Se resalta el código HTML que genera el elemento seleccionado en A.
- **C.** Estilos CSS que se están aplicando a ese elemento.



F3.3d
Inspector de elementos de Chrome para ver el código HTML y los estilos CSS aplicados.

Como ejemplo, si queremos cambiar el color del título de la página y tenemos este código HTML [F3.3e]:

```
<div class="site-branding_text">
  <div class="site-branding_name">
    <a href="/" title="Inicio" rel="home">Experto en Drupal 8</a>
  </div>
</div>
```

F3.3e
Ejemplo de código HTML

Podemos cambiar el color del texto aplicando el siguiente estilo CSS [F3.3f]:

```
.site-branding_text a {
  color: #000;
}
```

F3.3f
Ejemplo de estilos CSS

```
.site-branding__name a{
  color: #000;
}
```

Pero, ¿dónde añadimos este código?

En primer lugar, para poder localizar el archivo CSS correcto, tenemos que desactivar la caché de archivos CSS, desde **Rendimiento (desactivar la opción Combinar archivos CSS)** [F3.3g].

F3.3g

Combinar archivos CSS

Desactivar esta opción durante el desarrollo del proyecto, especialmente durante la modificación de estilos CSS.

Rendimiento ☆

Inicio » Administración » Configuración » Development

▼ VACIAR CACHÉ

Vaciard todas las cachés

▼ CACHÉ

Nota: Drupal provee un módulo de manejo de caché para páginas internas que es recomendado para sitios pequeños y medianos.

Maxima vida de la cache de la página

<no poner en caché> ▼

El tiempo máximo que una página puede estar cacheada en navegadores y proxys. Usado como el valor para max-age en cabeceras Cache-Control.

▼ OPTIMIZACIÓN DE ANCHO DE BANDA

Los recursos externos pueden ser optimizados automáticamente, lo que reducirá tanto el tamaño como el número de peticiones a su sitio web.

Combinar archivos CSS

Juntar archivos JavaScript

Guardar configuración

Una vez desactivada esta opción, si **inspeccionamos** el código y miramos en la **sección C** [F3.3d], veremos que el atributo color se está aplicando en un estilo del archivo **/core/themes/bartik/css/colors.css**. En el siguiente código se resalta el estilo que está afectando realmente al título [F3.3h].

F3.3h

Ejemplo de estilos CSS

```
.region-header,
.region-header a,
.region-header li a.is-active,
.site-branding__text,
.site-branding,
.site-branding__text a,
.site-branding a,
.region-secondary-menu .menu-item a,
.region-secondary-menu .menu-item a.is-active {
  color: #fffeff;
}
```

Tenemos localizado el archivo y el código donde se aplica el estilo que queremos cambiar, pero, **¿podemos modificar directamente este archivo para cambiar el código de color? La respuesta es NO.**

Nuestro tema es un subtema de Bartik, que es un tema del núcleo, y el archivo colors.css es un archivo incluido en Bartik. Si lo modificamos directamente, estaríamos modificando un archivo del núcleo, algo que no se debe hacer bajo ninguna circunstancia. Entre otras cosas, porque cuando actualicemos Drupal,

perderemos el cambio realizado.

Así que el código lo tenemos que incluir en un archivo de nuestro tema, concretamente en el archivo `/themes/b_forcontu/css/style.css` que creamos previamente.

El código que contendrá el archivo, teniendo en cuenta el código que ya contenía anteriormente, será **[F3.3i]**:

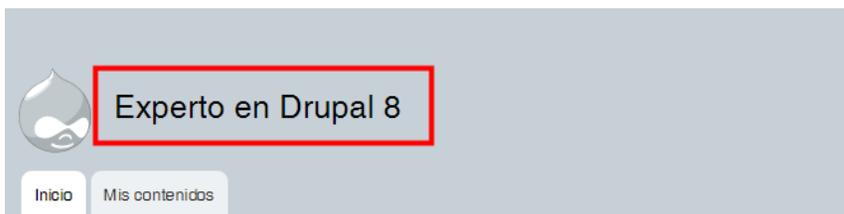
```
#header {
  background-color: #C7D0D6;
  background-image: none;
}
```

```
.site-branding__text a {
  color: #000;
}
```

F3.3i
Ejemplo de estilos CSS

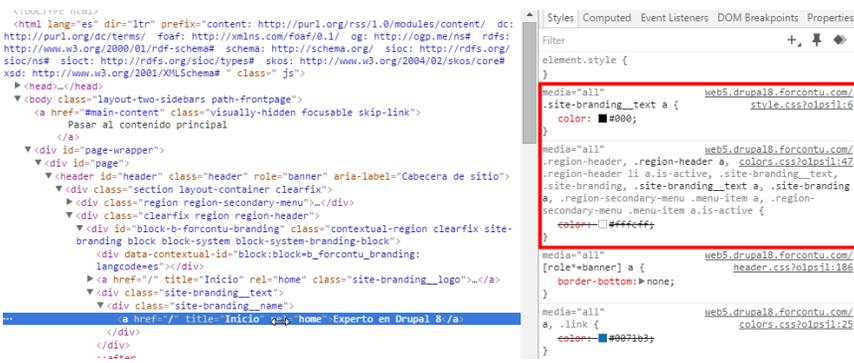
Por último, actualiza la página para ver los cambios (título en negro) **[F3.3j]**. Si no se actualiza, prueba lo siguiente, en este orden:

1. Actualizar la caché de Drupal.
2. Actualizar la caché del navegador (Control + F5).



F3.3j
Estilos CSS
Resultado final.

Utilizando el Inspector de elementos del navegador, podemos observar el nuevo código CSS que se está aplicando al título. El código anterior sigue mostrándose, pero tachado, ya que el código del tema **Bartik Forcontu**, al evaluarse en último lugar, prevalece sobre los estilos aportados por el tema base Bartik **[F3.3k]**.



F3.3k
Estilos CSS
Usamos el inspector de elementos para verificar los nuevos estilos aplicados.

3.4 Regiones personalizadas

Añadir regiones personalizadas

Para añadir regiones personalizadas, seguiremos estos pasos:

1. **Copiar las regiones del tema base** en el archivo `.info.yml` de nuestro tema.

En el archivo `.info.yml` del tema base, encontraremos la definición de regiones, en la variable **regions**. Por ejemplo, en el archivo `bartik.info.yml` se definen las siguientes regiones **[F3.4a]**:

F3.4a

Regiones en Bartik

Regiones definidas en el archivo `bartik.info.yml`. Si no indicamos lo contrario, nuestro tema heredaría estas regiones.

```

name: Bartik
type: theme
base theme: classy
description: 'A flexible, recolorable theme with many regions and
a responsive, mobile-first layout.'
package: Core
version: VERSION
core: 8.x
libraries:
  - bartik/global-styling
ckeditor_stylesheets:
  - css/base/elements.css
  - css/components/captions.css
  - css/components/table.css
  - css/components/text-formatted.css
regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  highlighted: Highlighted
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'

```

Si las regiones de nuestro tema van a ser las mismas que las del tema base, no es necesario añadir estas regiones al archivo `info.yml` de nuestro tema (se heredan directamente). Sin embargo, si queremos modificar las regiones, añadiendo o eliminando, tendremos que copiar la definición completa para, posteriormente, realizar las modificaciones.

- a) Localiza y edita el archivo `/core/themes/bartik/bartik.info.yml`
- b) Copia la variable `regions` junto con toda la definición de regiones
- c) Pega el texto en `/themes/b_forcontu/b_forcontu.info.yml`

- Añadir la nueva región en la variable **regions** de nuestro tema (archivo .info.yml).

Volviendo a nuestro tema, modifica la variable regions añadiendo la nueva región. Elige el orden en función de su ubicación con respecto a otras regiones del tema.

En nuestro tema de ejemplo añadiremos la región 'New region' (new_region) por debajo de la región 'Highlighted' (Destacado) **[F3.4b]**.

```
name: 'Bartik Forcontu'
type: theme
base theme: bartik
description: 'Subtema a partir de Bartik'
core: '8.x'

libraries:
  - b_forcontu/global-styling

regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  highlighted: Highlighted
  new_region: 'New region'
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'
```

F3.4b Añadir región personalizada

Copiamos todas las regiones del tema base y añadimos las regiones personalizadas.

- Añadir el HTML adicional y la variable **page.nombre_region** en la plantilla **page.html.twig** del tema.

Si nuestro tema aún no tiene la plantilla page.html.twig, la copiaremos desde el tema base, generalmente dentro de la carpeta templates. Dentro del archivo de plantilla, añadiremos el código (HTML y Twig) para imprimir la nueva región. Siempre añadimos la condición if para sólo imprimir la región si tiene algún bloque asignado **[F3.4c]**.

```
{% if page.highlighted %}
  <div class="highlighted">
    <aside class="layout-container section clearfix"
role="complementary">
      {{ page.highlighted }}
    </aside>
  </div>
{% endif %}
{% if page.new_region %}
  <div class="new-region">
```

F3.4c Añadir región personalizada

Código de la región añadido en la plantilla page.html.twig.

```

<aside class="layout-container section clearfix"
role="complementary">
  {{ page.new_region }}
</aside>
</div>
{% endif %}
{% if page.featured_top %}
<div class="featured-top">
  <aside class="featured-top__inner section layout-container
clearfix" role="complementary">
    {{ page.featured_top }}
  </aside>
</div>
{% endif %}

```

Nota: También puedes añadir una referencia a la nueva región en el bloque de comentarios de la plantilla.

4. Vacía caché y comprueba que la región esté funcionando correctamente.

Comprueba que en el listado de administración de bloques se muestra la nueva región **[F3.4d]**.

F3.4d

Añadir región personalizada

La nueva región se muestra en el listado de administración de bloques.



También puedes consultar la Demostración de regiones **[F3.4e]**.

F3.4e

Añadir región personalizada

La nueva región se muestra en la demostración de regiones del tema.



Por último, coloca en la región un bloque de prueba para ver si se muestra correctamente. Puedes colocar el bloque "Contenido reciente", pero debes asegurarte de tener creado al menos un contenido.

Eliminar una región

Para eliminar una región, seguiremos los siguientes pasos:

1. **Copiar las regiones del tema base** en el archivo `.info.yml` de nuestro tema.

Si nuestro tema no tiene aún el listado de regiones dentro del archivo `.info.yml`, lo copiaremos desde el tema base, como ya hicimos para añadir regiones.

2. Elimina la definición de la región en el archivo `.info.yml`.
3. Elimina el código que hace referencia a la región en la plantilla `page.html.twig`.

Nota: También puedes eliminar la referencia a la región en el bloque de comentarios de la plantilla.

4. Vacía caché y comprueba que la región ya no se muestra en la página de administración de bloques.

3.5 Creación de temas con Drupal Console

Desde Drupal Console también podemos crear un nuevo tema a partir de un tema base. Se pueden añadir librerías, definir las regiones, los breakpoints, etc.

drupal generate:theme

Genera un tema.

```
drupal generate:theme options
drupal gt
```

Si no introducimos los parámetros adicionales en *options*, la consola los irá solicitando de forma interactiva.

```
$ drupal generate:theme

// Welcome to the Drupal theme generator
Enter the new theme name []:
> Console Theme

Enter the module machine name [console_theme]:
> console_theme

Enter the theme Path [/themes/custom]:
> /themes

Enter theme description [My Awesome theme]:
>

Enter package name [Other]:
> Forcontu

Enter Drupal Core version [8.x]:
>

Base theme (i.e. classy, stable) [false]:
> classy

Enter the global styling library name [global-styling]:
>

Do you want to generate the theme regions (yes/no) [yes]:
> yes

Enter region name [Content]:
>

Enter region machine name [content]:
>

Do you want to add another region (yes/no) [yes]:
> no

Do you want to generate the theme breakpoints (yes/no) [yes]:
> no

Do you confirm generation? (yes/no) [yes]:
> yes

Generated or updated files

1 - be/be2/themes/console_theme/console_theme.info.yml
2 - be/be2/themes/console_theme/console_theme.theme
```

<https://hechoendrupal.gitbooks.io/drupal-console/content/es/commands/generate-theme.html>

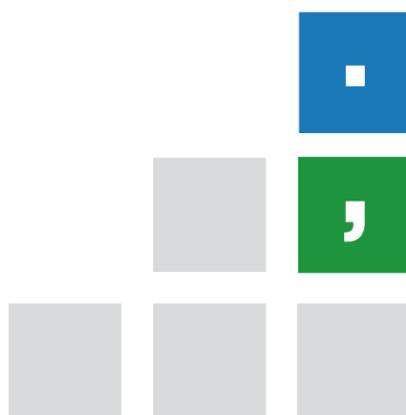
Fran Gil

Front-End Development I

4

Plantillas Twig en Drupal

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

4 Plantillas Twig en Drupal

Twig es el motor de plantillas utilizado en Drupal 8. Con respecto a Drupal 7, que utiliza PHPtemplate, Twig aporta las siguientes ventajas:

- Mayor seguridad. Al no incluir código PHP, no existe la posibilidad de añadir código directamente en la plantilla, lo que evita que se cometan errores y posibles agujeros de seguridad.
- Limpieza del código. Al no permitirnos meter código PHP en las plantillas, el código tendrá que añadirse obligatoriamente donde corresponde, en el módulo que hayamos desarrollado. De este modo toda la lógica de negocio queda en los módulos y no se deriva hacia las plantillas.
- Facilidad de uso y extensibilidad. Como veremos en esta y próximas unidades, el código Twig es muy fácil de usar.

En esta unidad comenzamos el estudio de las plantillas en Drupal y el código Twig.

Comparativa D8/D7

En Drupal 8 se ha cambiado el motor de plantillas a Twig, con importantes diferencias con respecto a PHPtemplate, usado en Drupal 7.

Aunque cambia el código utilizado dentro de las plantillas, la estructura del sistema de plantillas se mantiene prácticamente similar a la de Drupal 7.

Contenidos de la Unidad

- 4.1 Qué son las plantillas
- 4.2 Estructura de plantillas de Drupal
- 4.3 Depuración de plantillas
- 4.4 Sugerencias de plantillas (template suggestions)
- 4.5 Twig en Drupal



4.1 Qué son las plantillas

Qué son las plantillas

Para entender cómo funcionan los temas, es necesario entender primero cómo funcionan las plantillas o templates.

Una plantilla es un archivo de texto con extensión `.html.twig`, donde se mezcla código estático o fijo en HTML, con otros fragmentos de programación (en Twig), que incluyen variables o expresiones. Cuando Drupal carga el sitio web, se sustituirán las variables y expresiones por sus correspondientes valores, que junto con el contenido estático dará lugar al código HTML final que se mostrará en el navegador.

Como ejemplo, revisemos el siguiente código de una plantilla:

```
<h1>{{ title }}</h1>
```

El código Twig empieza con `{{` y termina con `}}`. En este código se imprime por pantalla el contenido de la variable `title`. Esta variable se utiliza para indicar el título de la página, un valor dinámico que depende de la página que se esté cargando en cada momento.

Ahora bien, supongamos que hemos creado dos páginas en nuestro sitio, con los títulos "Aviso legal" y "Quiénes somos". Al aplicar la plantilla, el resultado final será distinto para cada una de las páginas, en concreto:

Página "Aviso legal":

```
<h1>Aviso legal</h1>
```

Página "Quiénes somos":

```
<h1>Quiénes somos</h1>
```

El código finalmente generado es un código estático con marcas HTML, pudiendo incluir también CSS (y Javascript). Este es el código que se envía al navegador del usuario para representar el sitio web y la página particular que se está cargando. Los navegadores incluyen una opción "Ver código fuente" que permite ver el código HTML generado.

Twig vs PHPTemplate

Veamos un ejemplo muy sencillo de plantilla en **PHPTemplate**, que es el motor de plantillas que usa Drupal 7 y versiones anteriores. Las plantillas PHPTemplate son una mezcla de HTML y código PHP, que es el encargado de imprimir los valores dinámicos a partir de ciertos parámetros.

En la siguiente plantilla imprimimos un título de página (`$page_title`) y un listado de elementos de menú (`$menu`).

```
<h1><?php echo $page title ?></h1>

<ul id="navigation">
  <?php foreach ($menu as $item): ?>
    <li><?php echo $item ?></li>
  <?php endforeach ?>
</ul>
```

El HTML resultante depende del valor de esas variables. Por ejemplo, si tenemos los siguientes valores:

```
$page_title = "List of numbers";
$menu = ['One', 'Two', 'Three'];
```

El resultado después de renderizar la plantilla con estos valores será:

```
<h1>List of numbers</h1>

<ul id="navigation">
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
</ul>
```

Como puedes ver, en el resultado renderizado no hay rastro de PHP. Al navegador (o aplicación que realiza la petición) solo llegará la respuesta con el formato de texto solicitado (HTML, JSON, XML, etc.).

Symfony incorpora un motor de plantillas llamado **Twig**, mucho más potente que PHPTemplate.

La principal diferencia es que no utiliza PHP, por lo que no será posible introducir código "basura" (o simplemente mal ubicado) dentro de la plantilla. La plantilla cumple así su objetivo de presentar el contenido, sin que sea posible insertar lógica de programación.

Además, al no contener PHP, el código es más amigable para los perfiles sin conocimientos de programación (diseñadores y maquetadores), evitándose que puedan cometer errores derivados de la modificación de los fragmentos de código PHP.

La plantilla anterior en formato **Twig**, tendría el siguiente aspecto:

```
<h1>{{ page_title }}</h1>
```

```
<ul id="navigation">
  {% for item in menu %}
    <li>{{ item }}</li>
  {% endfor %}
</ul>
```

Por supuesto, si las variables `page_title` y `menu` reciben los mismos valores, el resultado será el mismo que usando la plantilla `PHPTemplate` anterior.

Ventajas de Twig frente a PHPTemplate

Twig es el motor de plantillas utilizado en Drupal 8. En Drupal 7 se utilizaba `PHPTemplate`, un motor de plantillas basado en PHP. Podemos resumir las **ventajas de Twig** frente a `PHPTemplate` en:

- **Mayor seguridad.** Al no incluir código PHP, no existe la posibilidad de añadir código directamente en la plantilla, lo que evita que se cometan errores y posibles agujeros de seguridad.
- **Limpieza del código.** Al no permitirnos meter código PHP en las plantillas, el código tendrá que añadirse obligatoriamente donde corresponde, en el módulo que hayamos desarrollado. De este modo toda la lógica de negocio queda en los módulos y no se deriva hacia las plantillas.
- **Facilidad de uso y extensibilidad.** Como veremos en los cursos de `Back-End Development` y `Front-End Development`, es muy sencillo crear y extender plantillas en formato Twig.

Estructura de plantillas de Drupal

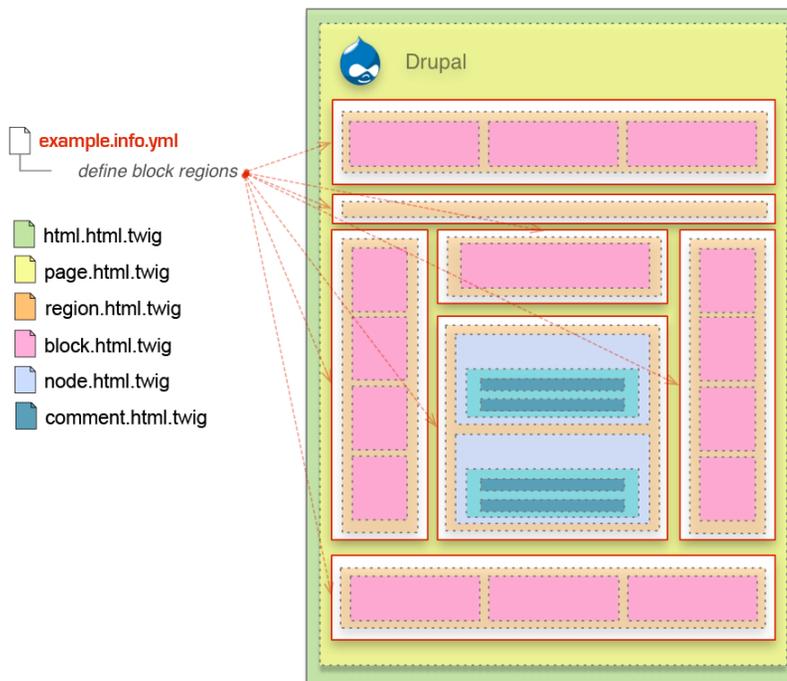
4.2

Una vez entendido el concepto de plantilla, vamos a analizar más a fondo cada uno de los archivos de plantilla que componen un tema de Drupal: **html.html.twig**, **page.html.twig**, **region.html.twig**, **node.html.twig** y **block.html.twig**, entre otros.

Drupal tiene sus propios archivos de plantillas por defecto, que utiliza en el caso de que el tema activado en el sitio no disponga de ellos. Es decir, si el tema no incluye un archivo **block.html.twig**, a la hora de mostrar bloques Drupal utilizará el archivo **block.html.twig** incluido por defecto en el núcleo (este archivo en particular lo aporta el módulo Block del núcleo, y está disponible en **/core/modules/block/templates/block.html.twig**).

Por tanto, si el tema con el que estamos trabajando no incluye un fichero **block.html.twig** pero queremos variar la manera de la que se muestran los bloques, debemos copiar este archivo por defecto en la carpeta **templates** del tema, y a continuación trabajar con él. Drupal, al encontrar un archivo **block.html.twig** en el tema, utilizará éste archivo en lugar del archivo del núcleo.

Cada archivo de plantilla aporta el diseño de un elemento. En la **Figura [F4.4a]** se muestran coloreados los archivos de plantilla principales y cómo afectan a cada elemento del sitio.



F4.2a Plantillas de un tema

Cada archivo de plantilla se encarga de la capa de presentación de un elemento del sitio.

- **html.html.twig** se encarga de configurar la página HTML y de presentar el contenido de la plantilla **page.html.twig**, que a su vez recibe contenidos de otras plantillas interiores, como **block.html.twig**, que genera la salida de presentación de los bloques.

Trabajando de forma anidada se obtiene la presentación final de la página que se está cargando en cada momento.

Imagen obtenida de: <http://drupal.org/node/171194> (adaptada a Drupal 8)

A continuación, describiremos cada uno de estos archivos y veremos algunas de las variables típicas que se pueden utilizar. Como ejemplo tomaremos las plantillas incluidas en el núcleo de Drupal.

La plantilla `html.html.twig`

Este es el archivo de plantilla de más alto nivel, y es el encargado de definir la estructura de la página HTML. Será el encargado de mostrar las etiquetas `<html>`, `<head>` y `<body>`, entre otras [F4.2b].

Si el tema activo (o el tema base) no incluye este archivo, Drupal utilizará la plantilla por defecto, ubicada en:

`/core/modules/system/templates/html.html.twig`.

F4.2b

`html.html.twig`

Plantilla incluida en el módulo system del núcleo.

```
<!DOCTYPE html>
<html{{ html_attributes }}>
  <head>
    <head-placeholder token="{{ placeholder_token|raw }}">
      <title>{{ head_title|safe_join(' | ') }}</title>
      <css-placeholder token="{{ placeholder_token|raw }}">
      <js-placeholder token="{{ placeholder_token|raw }}">
    </head>
  <body{{ attributes }}>
    <a href="#main-content" class="visually-hidden focusable">
      {{ 'Skip to main content'|t }}
    </a>
    {{ page_top }}
    {{ page }}
    {{ page_bottom }}
    <js-bottom-placeholder token="{{ placeholder_token|raw }}">
  </body>
</html>
```

Algunas de las variables que se pueden utilizar en esta plantilla son:

- **head_title**. Título de la página para utilizar en la etiqueta `<title>`.
- **logged_in**. Indica si el usuario está logueado en el sitio.
- **node_type**. Si la página es un nodo, indicará el tipo de contenido.
- **page**. Contenido de la página en HTML, ya procesado y listo para mostrar en el navegador. Este contenido proviene de la plantilla **`page.html.twig`**.

Todas las variables disponibles, se describen en la cabecera de la plantilla:

F4.2c

`html.html.twig`

Cabecera con las variables disponibles en la plantilla.

```
{#
/**
 * @file
 * Default theme implementation for the basic structure of a single Drupal page.
 *
 * Variables:
 * - logged_in: A flag indicating if user is logged in.
 * - root_path: The root path of the current page (e.g., node, admin, user).
 * - node_type: The content type for the current node, if the page is a node.
 * - head_title: List of text elements that make up the head_title variable.
 * May contain one or more of the following:
 * - title: The title of the page.
 * - name: The name of the site.
 * - slogan: The slogan of the site.
 * - page_top: Initial rendered markup. This should be printed before 'page'.
 * - page: The rendered page markup.
 * - page_bottom: Closing rendered markup. This variable should be printed after
 * 'page'.
 * - db_offline: A flag indicating if the database is offline.
 * - placeholder_token: The token for generating head, css, js and js-bottom
 * placeholders.
 *
 * @see template_preprocess_html()
 */
```

```
* @ingroup themeable
*/
#}
```

La plantilla `page.html.twig`

Esta plantilla genera todo el código contenido en la página, y una vez procesada se envía a la plantilla **`html.html.twig`** a través de la variable `{{ page }}`. Algunas de las variables que se pueden utilizar son **[F4.2d]**:

- **base_path**. Ruta de la instalación de Drupal. Si la instalación se ha realizado en el directorio raíz del dominio, devuelve `"/`.
- **is_front**. Indica si la página actual es la portada del sitio.
- **logged_in**. Indica si el usuario está logueado en el sitio.
- **is_admin**. Indica si el usuario es administrador del sitio.
- **front_page**. Enlace a la página de inicio del sitio.
- **\$title**. Título de la página.
- **messages**. Mensajes de error o notificaciones.
- **node**. El objeto *node*, si se está cargando una página en la que se muestra un nodo.
- **Regiones**. Variables que imprimen el contenido de cada región:
 - **page.header**
 - **page.primary_menu**
 - **page.secondary_menu**
 - **page.highlighted**
 - **page.help**
 - **page.content**
 - **page.sidebar_first**
 - **page.sidebar_second**
 - **page.footer**
 - **page.breadcrumb**

Si el tema activado no incluye este archivo, Drupal utiliza el que tiene por defecto, ubicado en **`/core/modules/system/templates/page.html.twig`**.

```
<div class="layout-container">

  <header role="banner">
    {{ page.header }}
  </header>

  {{ page.primary_menu }}
  {{ page.secondary_menu }}

  {{ page.breadcrumb }}

  {{ page.highlighted }}

  {{ page.help }}
```

F4.2d

page.html.twig

Plantilla incluida en el módulo system del núcleo.

```
<main role="main">
  <a id="main-content" tabindex="-1"></a>{# link is in
html.html.twig #}

  <div class="layout-content">
    {{ page.content }}
  </div>{# /.layout-content #}

  {% if page.sidebar_first %}
    <aside class="layout-sidebar-first" role="complementary">
      {{ page.sidebar_first }}
    </aside>
  {% endif %}

  {% if page.sidebar_second %}
    <aside class="layout-sidebar-second" role="complementary">
      {{ page.sidebar_second }}
    </aside>
  {% endif %}

</main>

{% if page.footer %}
  <footer role="contentinfo">
    {{ page.footer }}
  </footer>
{% endif %}

</div>
```

La plantilla `node.html.twig`

Esta plantilla genera la presentación del nodo. El resultado de la plantilla se muestra en la plantilla de página (`page.html.twig`) dentro de la variable de región `{{ page.content }}` [F4.2e].

Si el tema activado no incluye este archivo, Drupal utiliza el que tiene por defecto, ubicado en `/core/modules/node/templates/node.html.twig`.

```
<article{{ attributes }}>

  {{ title_prefix }}
  {% if not page %}
    <h2{{ title_attributes }}>
      <a href="{{ url }}" rel="bookmark">{{ label }}</a>
    </h2>
  {% endif %}
  {{ title_suffix }}

  {% if display_submitted %}
    <footer>
      {{ author_picture }}
      <div{{ author_attributes }}>
        {% trans %}Submitted by {{ author_name }} on {{ date }}{%
endtrans %}
      {{ metadata }}
    </div>
  </footer>
  {% endif %}

  <div{{ content_attributes }}>
    {{ content }}
  </div>

</article>
```

F4.2e

`node.html.twig`

Plantilla incluida en el módulo `node` del núcleo.

A continuación se muestran algunas variables que se pueden utilizar en esta plantilla:

- **node.** El objeto `node`.
- **label.** El título del nodo.
- **content.** El contenido del nodo (o un fragmento o adelanto del nodo si se trata de un listado de nodos). Es un array que contiene todos los elementos del nodo. Se puede imprimir el contenido completo usando la variable `{{ content }}`, o imprimir un elemento en particular `{{ content.field_example }}`.
- **author_picture.** La imagen de usuario del autor del nodo.
- **date.** Fecha de creación del nodo.
- **author_name.** El nombre de usuario del autor del nodo.
- **url.** La URL del nodo.
- **display_submitted.** Indica si se debe publicar la información de envío (autor y fecha del envío).
- **attributes, title_attributes, content_attributes, author_attributes.**

Atributos HTML.

- **view_mode**. Modo de presentación. Por ejemplo, contenido completo (full) o resumen (teaser).
- **teaser**. Indica si el nodo se está mostrando en modo Resumen.
- **page**. Indica si el nodo se está mostrando en modo Contenido completo (full).
- **logged_in**. Indica si el usuario está logueado en el sitio.
- **is_admin**. Indica si el usuario es administrador del sitio.

La plantilla `block.html.twig`

Se utiliza para generar la presentación de los bloques de una página. Si el tema activado no incluye este archivo, Drupal utiliza el que tiene por defecto, ubicado en `/core/modules/block/templates/block.html.twig`.

Algunas de las variables disponibles en esta plantilla son **[F4.2f]**:

- **plugin_id**. Id interno de la implementación del bloque.
- **label**. El título del bloque.
- **content**. El contenido del bloque.
- **attributes**. Atributos HTML asociados al bloque. Por ejemplo, id o class.
- **title_attributes**. Atributos HTML asociados al título del bloque.
- **title_prefix** y **title_suffix**. Prefijo y sufijo del título.

F4.2f

`block.html.twig`

Plantilla incluida en el módulo `block` del núcleo.

```
<div{{ attributes }}>
  {{ title_prefix }}
  {% if label %}
    <h2{{ title_attributes }}>{{ label }}</h2>
  {% endif %}
  {{ title_suffix }}
  {% block content %}
    {{ content }}
  {% endblock %}
</div>
```

Otros archivos de plantilla

Existen muchas más plantillas que se encargan de otras áreas del sitio. Otros ejemplos de archivos de plantilla son:

- **region.html.twig**. Genera la presentación de cualquiera de las regiones del tema. El archivo por defecto de esta plantilla se encuentra en **/core/modules/system/templates/region.html.twig**.
- **comment.html.twig**. Muestra la salida para los comentarios del sitio. El archivo por defecto está en: **/core/modules/comment/templates/comment.html.twig**.
- **maintenance-page.html.twig**. Muestra la salida de la página de mantenimiento del sitio. El archivo por defecto de esta plantilla se encuentra en **/core/modules/system/templates/maintenance-page.html.twig**.

Por último, los módulos también pueden tener plantillas propias. Por ejemplo, el módulo **Forum** incorpora varias plantillas que sirven para configurar cómo se visualizan los mensajes del foro (forums.html.twig, forum-list.html.twig, etc.).

Cada módulo puede incluir varios archivos de plantillas, que utilizará desde su propio código para mostrar las nuevas herramientas que añade al sitio. Si queremos modificar cualquier de estas plantillas nunca debemos hacerlo directamente en el archivo del módulo. Primero copiaremos el archivo en la carpeta **templates** del tema, y ese será el archivo de plantilla que modificaremos, sin cambiar su nombre original.

4.3 Depuración de plantillas

Depuración de plantillas

Una de las mayores complicaciones durante el desarrollo, es localizar qué plantilla es la encargada de generar una determinada salida HTML. Para facilitar esta tarea, Twig incluye una opción de depuración, que activaremos únicamente durante el desarrollo del sitio:

- 1) Localiza el archivo **sites/default/services.yml**. Si el archivo no existe, **copia** el archivo default.services.yml (en la misma carpeta), y renómbralo como services.yml.

Ten en cuenta que, tanto la carpeta default como el archivo services.yml están protegidos contra escritura, así que tendrás que cambiar los permisos para poder copiar y editar el archivo settings.yml. Puedes cambiar los permisos a 777, realizar los cambios, y dejar que el sistema recupere los permisos originales.

- 2) Dentro de la sección **twig.config**, cambiar el valor de **debug** a **true**.

```
parameters:
  twig.config:
    debug: true
```

- 3) **Vacía la caché** del sitio.
- 4) Accede al **Informe de estado**. Este paso recupera los permisos originales de la carpeta **default**.
- 5) No olvides desactivar esta opción en el entorno de producción.

La información de depuración se mostrará como **comentario HTML**, en el código fuente de la página generada. Por ejemplo, en Chrome, podemos usar las opciones **Ver código fuente de página** o **Inspeccionar**.

Cada plantilla comienza con una etiqueta `<!-- THEME DEBUG -->`, e incluye la siguiente información de depuración:

- **THEME HOOK**. Hace referencia a la plantilla base que se utiliza. Por ejemplo, si el valor es `'field'`, la plantilla base será `field.html.twig`.
- **FILE NAME SUGGESTIONS**. Más adelante veremos qué son las sugerencias de plantillas (o sugerencias de nombres de plantilla). Una misma plantilla puede tener diferentes nombres, de forma que actúa de una forma más específica. Por ejemplo:
 - o **html.html.twig**. Es genérica, actúa para todas las páginas.
 - o **html--front.html.twig**. Es más específica, actúa cuando la página es la portada (front) del sitio.

Cuando existen las dos plantillas, se aplica la plantilla más específica. De hecho, el listado ya viene ordenado por orden de prioridad.

En el listado de nombres de plantilla sugeridos, la plantilla que se está actualizando en la página actual viene señalada con una x.

- **BEGIN/END OUTPUT.** Marca el inicio y el fin de la salida HTML devuelta por la plantilla. Tanto en el inicio como en el fin, se indica la ruta completa del archivo de plantilla utilizado. Ten en cuenta que las plantillas pueden contener otras plantillas.

Ejemplo 1: Inicio del archivo, con referencia a la plantilla html, que es la plantilla de mayor nivel. Esta plantilla se cierra justo al final del archivo.

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--front.html.twig
  * html--.html.twig
  x html.html.twig
-->
<!-- BEGIN OUTPUT from
'core/themes/classy/templates/layout/html.html.twig' -->
<!DOCTYPE html>
<html lang="es" ...>
  <head>
  ...
</body>
</html>
<!-- END OUTPUT from 'core/themes/classy/templates/layout/html.html.twig' -->
```

Ejemplo 2: Plantilla que se aplica al título del contenido. En este caso se está usando la plantilla **field--node--title.html.twig**, que es una plantilla específica para los títulos de los nodos.

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'field' -->
<!-- FILE NAME SUGGESTIONS:
  * field--node--title--article.html.twig
  x field--node--title.html.twig
  * field--node--article.html.twig
  * field--title.html.twig
  * field--string.html.twig
  * field.html.twig
-->
<!-- BEGIN OUTPUT from 'core/themes/classy/templates/field/field--
node--title.html.twig' -->
<span property="schema:name" data-quickedit-field-
id="node/9/title/es/teaser" class="field field--name-title field--
type-string field--label-hidden">Adipiscing Nimis Proprius</span>

<!-- END OUTPUT from 'core/themes/classy/templates/field/field--
node--title.html.twig' -->
```

Si queremos modificar la plantilla que presenta el título de los nodos (Ejemplo 2), tenemos que ver primero dónde se encuentra. En este ejemplo, la plantilla está localizada en el tema classy del núcleo, así que **no podemos modificarla directamente**. Lo que haremos será copiar esa plantilla a la carpeta templates de nuestro tema.

Una vez que la plantilla está en nuestro tema, recarga la página y comprueba que la ruta del archivo es la esperada. Vacía caché si no ves el cambio reflejado. El resto de información mostrada es exactamente igual, ya que el nombre de plantilla sigue siendo el mismo, pero ubicada en nuestro tema. Ahora ya podemos modificar la plantilla.

```

<!-- THEME DEBUG -->
<!-- THEME HOOK: 'field' -->
<!-- FILE NAME SUGGESTIONS:
  * field--node--title--article.html.twig
  x field--node--title.html.twig
  * field--node--article.html.twig
  * field--title.html.twig
  * field--string.html.twig
  * field.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/b_forcontu/templates/field--node--
title.html.twig' -->
<span property="schema:name" data-quickedit-field-
id="node/9/title/es/teaser" class="field field--name-title field--
type-string field--label-hidden">Adipiscing Nims Proprius</span>

<!-- END OUTPUT from 'themes/b_forcontu/templates/field--node--
title.html.twig' -->

```

La función dump()

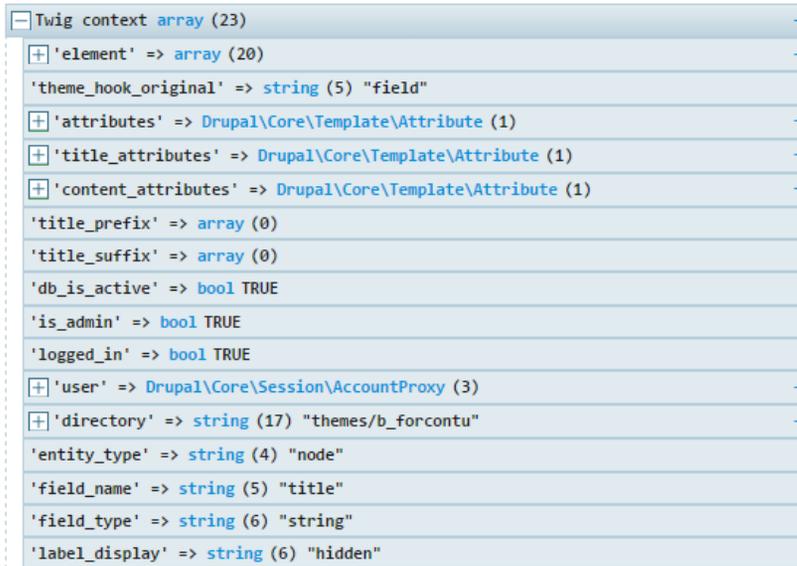
Al activar la depuración en Twig, también tenemos disponible la función dump() que podemos utilizar dentro una plantilla.

- **{{ dump(variable) }}**. Muestra el valor de la variable especificada.
- **{{ dump() }}**. Muestra todas las variables disponibles en la plantilla, con sus correspondientes valores.

Si has instalado el módulo Devel Kint (incluido en Devel), podrás usar la función kint(), que mejora la presentación del resultado.

- **{{ kint(variable) }}**. Muestra el valor de la variable especificada.
- **{{ kint() }}**. Muestra todas las variables disponibles en la plantilla, con sus correspondientes valores.

Tanto en PHP como en Twig, se recomienda utilizar Kint **[F4.3a]**.



F4.3a
Depuración en Twig
 Valores de variables mostrados con Kint.

Web Profiler

El icono **Tema** de Web Profiler muestra información del tema cargado, como el tiempo de renderizado, el número de llamadas a plantillas, bloques cargados, etc. **[F4.3b]**



F4.3b
Web Profiler
 Icono Tema.

Haciendo clic en el icono accedemos a información ampliada: Regiones, listado completo de plantillas que se han utilizado en la carga de la página, etc. **[F4.3c]**

F4.3c
Web Profiler

Información de Tema
ampliada.

Tema	
Nombre	b_forcontu (Editar)
Motor	twig
Base theme	bartik
Regiones	sidebar_first sidebar_second content header primary_menu secondary_menu footer highlighted help page_top page_bottom breadcrumb
Bibliotecas	bartik/global-styling classy/base core/normalize b_forcontu/global-styling
Style sheets removed	
Ruta	themes/b_forcontu
Negotiator	D:\C:\T\DefaultNegotiator::determineActiveTheme

Módulo BigPipe

El módulo BigPipe viene incluido en el núcleo y activado por defecto. Se trata de una técnica de carga de página, desarrollada por Facebook, que permite mostrar las páginas de una forma más rápida.

El modelo clásico de carga de una página, que es también el modelo de Drupal 8 sin BigPipe, implica no empezar a cargar una página hasta que el navegador ha recibido todos sus contenidos (contenido principal, bloques, etc.).

El modelo de carga de BigPipe sigue estos pasos:

- En lugar de ver la página como un "todo", se crean "espacios" vacíos llamados **placeholders**, reservados para los elementos que se van a cargar en la página. Puedes verlo como una región o bloque vacío, esperando a su contenido.
- Al cargar una página, **el sistema entrega primero el contenido principal y los contenidos cacheados** (bloques). Aunque en este punto no están rellenos todos los placeholders, el usuario ya puede ver el contenido principal de la página, con lo que la percepción es que la página carga muy rápido.
- A continuación, con la página ya cargada, el sistema va **entregando al navegador el contenido renderizado del resto de placeholders**, por separado. Si alguno de esos contenidos requiere un tiempo de carga o renderizado mayor, no afectará al rendimiento, simplemente ese placeholder se mostrará vacío y se rellenará cuando la información esté disponible.

Más información sobre la técnica BigPipe:

<https://www.facebook.com/notes/facebook-engineering/bigpipe-pipelining-web-pages-for-high-performance/389414033919>

Desinstalar el módulo BigPipe

Como norma general, **siempre usaremos el módulo BigPipe en producción**. Sin embargo, **conviene desinstalarlo durante el desarrollo del sitio**, especialmente durante el desarrollo Front-End. De lo contrario, el sistema de caché propio de BigPipe nos impedirá tener toda la información que necesitamos sobre las plantillas que se están cargando, ya que muchos de los elementos se cargarán directamente desde caché, sin volver a renderizarlos.

Una vez hayamos finalizado el desarrollo del sitio, no debemos olvidarnos de reinstalar el módulo BigPipe.

4.4

Sugerencias de plantillas
(template suggestions)

Antes de continuar, asegúrate de que el **módulo BigPipe está desinstalado**. Sólo de esta forma podrás ver la información de depuración de todas las plantillas utilizadas para generar la página.

Ya hemos comentado que, para modificar una plantilla, primero tenemos que copiarla a nuestro tema. En muchas ocasiones no nos interesa realizar cambios para todos los casos en que se usa una plantilla, sino solo para casos específicos. Por ejemplo, podríamos necesitar modificar la plantilla de bloque (block.html.twig) sólo para el bloque de búsqueda (bloque del módulo search con nombre de sistema form-block). En estos casos podemos hacer una copia de la plantilla base **block.html.twig** y cambiar el nombre para hacerlo específico: **block--search-form-block.html.twig**. No es nada complicado utilizando la información de depuración de plantillas.

En la Figura [F4.4a] vemos que la plantilla utilizada (marcada con una x) es ya la plantilla específica **block--search-form-block.html.twig**. En este caso es Bartik el que implementa esta plantilla para el bloque de búsqueda, localizada en:

- core/themes/bartik/templates/block--search-form-block.html.twig

Para modificarla, copiaremos esta plantilla a nuestro tema, sin necesidad de cambiar el nombre del archivo.

F4.4a

Sugerencias de plantillas

Plantilla específica para el bloque de formulario de búsqueda.

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'block' -->
<!-- FILE NAME SUGGESTIONS:
  * block--b-forcontu-formulariodebusqueda.html.twig
  x block--search-form-block.html.twig
  * block--search.html.twig
  * block.html.twig
-->
<!-- BEGIN OUTPUT from 'core/themes/bartik/templates/block--search-form-block.html.twig' -->
<div class="search-block-form contextual-region block block-search container-inline" data-drupal-selector="search-block-form" id="block-b-forcontu-formulariodebusqueda" role="search">

  <h2>Formulario de búsqueda</h2>
```

Veamos algunos ejemplos más:

- **page--node--1.html.twig**. Afecta únicamente a la plantilla page cuando se está cargando el nodo con nid 1.
- **page--node.html.twig**. Plantilla page cuando la página cargada es un nodo.
- **block--module-delta.html.twig**. Afecta al bloque *delta* (nombre interno) definido en el módulo *module*. Como vimos anteriormente, la plantilla block--search-form-block.html.twig sigue este patrón.
- **node--type.html.twig**. Afecta a todos los nodos de un tipo determinado. Por ejemplo, node--article.html.twig afectaría a todos los nodos de tipo artículo.

Puedes consultar otras combinaciones posibles en este enlace, aunque lo recomendable es siempre consultar la información de depuración de las plantillas:

<https://www.drupal.org/node/2354645>

Veamos otro ejemplo paso a paso. Queremos que la región Sidebar Second (columna lateral derecha), no se muestre en la página principal del sitio.

- 1) Asegúrate de que el módulo BigPipe está desinstalado.
- 2) Carga la página principal o portada del sitio.
- 3) Comprueba que haya algún bloque asignado a la región lateral derecha y que se esté mostrando. Si no es así, asigna desde la administración de bloques cualquier bloque de los disponibles (por ejemplo, Usuarios nuevos).
- 4) Localiza la información de depuración de la plantilla page.html.twig. Recuerda que en esta plantilla es donde se imprimen las regiones del tema. Busca 'page' en el código fuente para localizar la plantilla más rápido **[F4.4b]**.

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--front.html.twig
  * page--node.html.twig
  x page.html.twig
-->
<!-- BEGIN OUTPUT from
'themes/b_forcontu/templates/page.html.twig' -->
<div id="page-wrapper">
  <div id="page">
```

F4.4b Sugerencias de plantillas

Plantilla genérica
page.html.twig.

La plantilla que se está utilizando para la página principal es **page.html.twig**, ubicada en nuestro tema b_forcontu. Esta es la plantilla genérica para todas las páginas, salvo que se indique lo contrario con un nombre de plantilla más específico.

Entre las sugerencias de plantilla nos indica que, si queremos que sólo actúe sobre la portada, podemos utilizar el nombre page--front.html.twig.

- 5) Copia la plantilla **page.html.twig** en la misma carpeta **templates** del tema **b_forcontu**, renombrándola a **page--front.html.twig**. No elimines la plantilla original page.html.twig, ya que se seguirá utilizando para el resto de páginas que no sean la portada.
- 6) Vacía la caché del sitio, recarga la página principal del sitio y comprueba en el código fuente que ahora la plantilla utilizada es page--front.html.twig **[F4.4c]**.

F4.4c**Sugerencias de plantillas**

Plantilla page específica para la portada page--front.html.twig.

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  x page--front.html.twig
  * page--node.html.twig
  * page.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/b_forcontu/templates/page--front.html.twig' -->
<div id="page-wrapper">
```

- 7) Carga cualquier otra página y comprueba que la plantilla utilizada en ese caso no es la de portada, sino la genérica, page.html.twig.

Para que se cargue el tema principal del sitio, la **página no debe ser de administración**. Carga, por ejemplo, un nodo creado previamente (/node/1).

- 8) Ahora ya podemos realizar cambios en la plantilla de portada, page--front.html.twig. Localiza el código que imprime la región sidebar_second y elimínalo **[F4.4d]**.

F4.4d**Sugerencias de plantillas**

Código a eliminar de la plantilla page--front.html.twig.

```
{% if page.sidebar_second %}
  <div id="sidebar-second" class="column sidebar">
    <aside class="section" role="complementary">
      {{ page.sidebar_second }}
    </aside>
  </div>
{% endif %}
```

- 9) Vuelve a vaciar caché y comprueba que la región lateral derecha desaparece cuando cargas la página de portada, pero se sigue mostrando para el resto de página (que no sean de administración).

Twig en Drupal

4.5

Como ya hemos comentado, en las plantillas el código Twig se mezcla entre código HTML para generar el código HTML final que se envía al navegador. En este apartado veremos la sintaxis de Twig y cómo se usa dentro de Drupal.

Para ampliar la información contenida en este apartado, se aconseja consultar los siguientes enlaces:

- Documentación oficial de Twig:
<http://twig.sensiolabs.org/doc/templates.html>
- Estándares de codificación de Twig:
http://twig.sensiolabs.org/doc/2.x/coding_standards.html
- Estándares de codificación de Twig **en Drupal**:
<https://www.drupal.org/node/1823416>

Sintaxis

En una plantilla Twig podemos encontrar tres tipos de elementos, encapsulados de esta forma:

- `{{ ... }}`. Imprime una variable o el resultado de una expresión. En el ejemplo anterior, `{{ page_title }}` imprime el valor del parámetro `page_title` pasado a la plantilla.
- `{% ... %}`. Se utiliza para ejecutar estructuras de control y decisión. Por ejemplo, un **if** o, como en el ejemplo, un bucle **for**.
- `{# ... #}`. Esta es la forma de añadir un comentario dentro de la plantilla. Este comentario no se mostrará en la salida renderizada final.

Funcionalidad adicional en Twig

Twig incluye un amplio listado de filtros, etiquetas y funciones que se pueden aplicar a los contenidos, siempre con el objetivo único de modificar la presentación y no la lógica de negocio.

Por ejemplo, el filtro **upper** pasa a mayúsculas una variable de texto.

```
<h1>{{ page_title|upper }}</h1>
```

Y el filtro **lower** pasa a minúsculas un texto:

```
<ul id="navigation">
  {% for item in menu %}
    <li>{{ item|lower }}</li>
  {% endfor %}
</ul>
```

Aplicando los filtros anteriores, el resultado sería:

```
<h1>LIST OF NUMBERS</h1>

<ul id="navigation">
  <li>one</li>
  <li>two</li>
  <li>three</li>
</ul>
```

Comentarios

Para escribir comentarios en el código, utilizaremos `{# #}`. Si el comentario tiene una única línea, la apertura y cierre se realiza en la misma línea. Si el comentario tiene varias líneas, los separaremos con un salto de línea. Siempre se debe intentar que las líneas de comentarios tengan menos de 80 caracteres.

```
{# Comentario de una línea #}

{#
  Comentario de varias líneas
  nota: todo este código es un comentario y no se ejecutará.
  tampoco las variables que podamos incluir dentro, como {{ title }}
#}
```

DocBlock

Un DocBlock es un bloque de documentación que se coloca al principio de un archivo de plantilla. Se trata de un bloque de comentarios y, por tanto, estará delimitado por `{# y #}`.

El archivo de plantilla se comenta con una directiva **@file** y una lista con las **variables disponibles (Available variables)**, enviadas por las funciones de preprocesamiento, a las que también se hace referencia a través de directivas **@see**. Por ejemplo:

```
{#
/**
 * @file
 * Default theme implementation to present a feed item in an aggregator page.
 *
 * Available variables:
 * - url: URL to the originating feed item.
 * - title: Title of the feed item.
 * - content: All field items. Use {{ content }} to print them all,
 *   or print a subset such as {{ content.field_example }}. Use
 *   {{ content|without('field_example') }} to temporarily suppress the printing
 *   of a given element.
 * - attributes: HTML attributes for the wrapper.
 * - title_prefix: Additional output populated by modules, intended to be
 *   displayed in front of the main title tag that appears in the template.
 * - title_suffix: Additional output populated by modules, intended to be
 *   displayed after the main title tag that appears in the template.
 */
```

```

*
* @see template_preprocess_aggregator_item()
*
* @ingroup themeable
*/
#)
<article{{ attributes }}>
  {{ title_prefix }}
  <h3>
    <a href="{{ url }}">{{ title }}</a>
  </h3>
  {{ title_suffix }}
  {{ content }}
</article>

```

Para ver otros ejemplos reales de codificación, recomendamos el análisis directo de los archivos del núcleo de Drupal y de otros módulos contribuidos.

La directiva **@ingroup themeable** indica que la plantilla está dentro del grupo **themeable**. Esta directiva solo se incluye en las plantillas por defecto, así que no deberíamos incluirla al sobrescribir una plantilla en nuestro tema.

Variables

Imprimir variables

Para imprimir el valor de una variable, esta se encierra entre llaves dobles `{{ variable }}`. Veamos algunos ejemplos:

- a) Ejemplo de variable simple:

```

{{ title }}

```

- b) Variable con atributos (proveniente de un objeto en PHP):

```

{{ foo.bar }}

```

- c) Variable con elementos (proveniente de un array en PHP):

```

{{ foo['bar'] }}

```

- d) Si el nombre del atributo contiene caracteres especiales (como por ejemplo un guión – que podría ser interpretado como "menos"), podemos utilizar la función **attribute()**:

```

{{ attribute(foo, 'data-bar') }}

```

Sería equivalente a mostrar el valor de 'foo.data-bar'.

Comprobación de variables

Esta es una expresión muy típica en Drupal. Comprobamos si la variable está definida y tiene un valor antes de imprimirla. El código HTML que encapsula a la variable, también se incluye dentro del if.

No es necesario usar el filtro 'is defined' para comprobar si la variable está disponible.

```
{% if foo %}
  <div>{{ foo }}</div>
{% endif %}
```

Asignar variables

Twig nos permite utilizar variables auxiliares dentro de la plantilla. Utilizaremos la etiqueta **set**, entre {% y %} (en lugar de {{ }}).

```
{% set foo = 'bar' %}
{% set foo = [1, 2] %}
```

Estas variables auxiliares se pueden imprimir o utilizar en otras expresiones.

Variables globales

En todas las plantillas están disponibles estas variables globales:

- **_self**. Devuelve un objeto de tipo Twig_Template, con información de la plantilla actual. Los métodos disponibles se pueden consultar en:

https://api.drupal.org/api/drupal/vendor!twig!twig!lib!Twig!Template.php/class/Twig_Template/8

Recuerda que puedes utilizar las funciones de depuración dump() y kint() para ver la estructura y contenidos de cualquier variable.

Si imprimimos la variable directamente, {{ _self }}, devuelve la ruta y nombre del archivo de plantilla actual, desde el directorio raíz del sitio. Por ejemplo:

```
themes/b_forcontu/templates/node--article.html.twig
```

- **_context**. Se trata de un array con información del contexto actual, como todas las variables que están disponibles en la plantilla. Además de las variables que llegan a la plantilla, también muestra las variables que se han definido dentro de la plantilla.

{{ dump(context) }} es equivalente a **{{ dump() }}**. También con kint().

- **_charset**. Juego de caracteres (por ejemplo, "UTF-8").

Filtros

Los filtros se utilizan para modificar las variables de alguna forma. Por ejemplo, un filtro puede servir para pasar una cadena a mayúsculas (upper).

Para usar un filtro sobre una variable, primero se escribe la variable sobre la que se aplicará el filtro, y luego el nombre del filtro separado por el símbolo pipe |. Se pueden concatenar varios filtros, separados por |, que se aplicarán a la variable en orden.

```
{% set foo = 'welcome' %}
{{ foo|upper }}

{# La salida será: WELCOME #}

{{ 'welcome'|upper }}

{# también se puede utilizar el filtro directamente sobre
una cadena, obteniéndose el mismo resultado
#}
```

Los filtros también se pueden aplicar a una sección de código, con la siguiente estructura:

```
{% filter upper %}
  Este texto se convertirá a mayúsculas
{% endfilter %}
```

Algunos de los filtros disponibles son:

- **upper.** Pasa a mayúsculas todas las letras de una cadena.
- **lower.** Pasa a minúsculas todas las letras de una cadena.

```
{{ 'WELCOME'|lower }}
{# Resultado: welcome #}
```

- **capitalize.** Pasa a mayúsculas sólo la primera letra de una cadena.

```
{{ 'welcome home'|capitalize }}
{# Resultado: Welcome home #}
```

- **title.** Pasa a mayúsculas la primera letra de cada palabra de la cadena.

```
{{ 'welcome home'|title }}
{# Resultado: Welcome Home #}
```

- **date.** Permite dar formato a una fecha. Funciona como la función date de PHP.

```
{{ "now"|date("d/m/Y") }}
{# Resultado: 13/02/2016 #}
```

- **join.** Une los elementos de un array en una cadena. Por defecto se utiliza una cadena vacía como separador (sin separación), pero se puede especificar cualquier otro caracter.

```

{{ [1, 2, 3]|join }}
{# Resultado: 123 #}

{{ [1, 2, 3]|join(',') }}
{# Resultado: 1,2,3 #}

```

- **sort.** Ordena un array.

```

{% set foo = [3, 2, 4, 1]|sort %}
{# El contenido de la variable for será: [1, 2, 3, 4] #}

```

- **length.** Devuelve el número de elementos de un array o el número de caracteres de una cadena.

```

{% set count = users|length %}
{# Almacena en la variable count el número de elementos de user #}

```

El listado completo de filtros de Twig se puede consultar en:

<http://twig.sensiolabs.org/doc/filters/index.html>

Filtros específicos de Drupal

Además de los filtros que vienen de base con Twig, Drupal incorpora una serie de filtros adicionales, que se pueden consultar en:

<https://www.drupal.org/docs/8/theming/twig/filters-modifying-variables-in-twig-templates>

Estos filtros están declarados en TwigExtension::getFilters():

<https://api.drupal.org/api/drupal/core!lib!Drupal!Core!Template!TwigExtension.php/function/TwigExtension::getFilters/8>

- **trans (o t).** Se utiliza para **traducir cadenas desde la plantilla.** Al igual que hacemos con todos los textos incluidos directamente en el código de un módulo, los textos que se añaden en una plantilla también deben pasar por la función t() de traducción.

Cuando se trate de una cadena simple, sin patrones de reemplazo, usaremos el filtro |t.

```

<a href="{{ url('<front>') }}" title="{{ 'Home'|t }}" rel="home"
class="site-logo"></a>

<h2>{{ 'Book traversal links for'|t }} {{ book_title }}</h2>

<b>{{ 'Not triggered'|t }}</b>

```

Cuando la cadena tenga variables o patrones de reemplazo, usaremos el filtro `{% trans %}`:

```
{% trans %}
  Submitted by {{ author_name }} on {{ date }}
{% endtrans %}
```

```
{% trans %}Submitted by {{ author_name }} on {{ date }}{% endtrans %}
```

```
{% trans %}
  This {{ token.name }} has a length of: {{ count }}. It contains:
  {{ token.numbers|placeholder }} and {{ token.bad_text }}.
{% endtrans %}
```

- **placeholder**. Devuelve el texto enfatizado (``). También realiza HTML escaping, convirtiendo ciertos caracteres especiales a sus correspondencias en HTML.

```
{% set temp = 'The value is < 5' %}

{{ temp|placeholder }}
```

Imprime:

This value is < 5

El código HTML generado es:

```
<em class="placeholder">This value is &lt; 5</em>
```

Estructuras de control

Estructuras de control: for

El for se puede utilizar como un foreach:

```
<h1>Users</h1>
<ul>
  {% for user in users %}
    <li>{{ user.name }}</li>
  {% endfor %}
</ul>

{# Imprime una lista HTML de nombres de usuario (atributo name)

<h1>Users</h1>
<ul>
  <li>frangil</li>
  <li>laurafornie</li>
  <li>userfoo</li>
</ul>

#}
```

Si necesitamos utilizar tanto la clave como el valor, el equivalente al foreach sería:

```
{% for key, value in items %}
  <div class="{{ key }}">{{ value }}</div>
{% endfor %}
```

Y también podemos usarlo como un for clásico, estableciendo un rango de ejecución:

```
{% for i in range(0, 3) %}
  {{ i }},
{% endfor %}

{# Imprime 0, 1, 2, 3 #}
```

Estructuras de control: if

Permite evaluar una condición para realizar o no un conjunto de acciones.

```
{% if title|length > 0 %}
  <h1>{{ title }}</h1>
{% endif %}

{# Imprime el valor de título (y las etiquetas h1) sólo si
la variable título no está vacía.

#}
```

Funciones

Twig dispone de un conjunto de funciones que podemos utilizar en las plantillas.

Por ejemplo, la función **range**, que ya utilizamos para establecer el rango de valores de la estructura for anterior, devuelve un listado de números entre el mínimo y el máximo indicados. También permite establecer el salto entre valores (tercer parámetro).

```
{% for i in range(0, 6, 2) %}
  {{ i }},
{% endfor %}

{# Imprime 0, 2, 4, 6 #}
```

La función **max** devuelve el valor máximo de entre los facilitados.

```
{{ max(1, 3, 2) }}
```

```
{# Imprime 3 #}
```

El conjunto completo de funciones disponibles por defecto se pueden consultar en:

<http://twig.sensiolabs.org/doc/functions/index.html>

Funciones específicas de Drupal

Además de las funciones que vienen de base con Twig, Drupal incorpora una serie de funciones adicionales, que se pueden consultar en:

<https://www.drupal.org/node/2486991>

url() y path()

Dado un nombre de ruta y sus parámetros, ambas funciones generan la URL correspondiente. La única diferencia es que **url() genera la URL absoluta**, y **path() genera la URL relativa**. La ruta tiene que estar previamente definida en un archivo .routing.yml.

Los parámetros requeridos por las funciones son:

```
url($name, $parameters, $options)
path($name, $parameters, $options)
```

Veamos algunos ejemplos:

- a) URL absoluta a la página de inicio:

```
<a href="{{ url('<front>') }}">{{ 'Home'|t }}</a>
```

Genera el código HTML:

```
<a href="http://www.example.com/">Inicio</a>
```

b) URL relativa a la página de inicio:

```
<a href="{{ path('<front>') }}">{{ 'Home'|t }}</a>
```

Genera el código HTML:

```
<a href="/">Inicio</a>
```

c) URL relativa a la ruta `forcontu_pages.simple` (definida en el módulo Forcontu Pages):

```
<a href="{{ path('forcontu_pages.simple') }}">{{ 'Forcontu Simple Page'|t }}</a>
```

d) URL relativa a la página de perfil de usuario:

```
<a href="{{ path('entity.user.canonical', {'user': user.id}) }}">{{ 'View user profile'|t }}</a>
```

e) URL relativa a un nodo:

```
<a href="{{ path('entity.node.canonical', {'node': node.id}) }}">{{ 'View node page'|t }}</a>
```

link()

La función `link()` genera el código completo del enlace, pasándole como parámetros el texto del enlace, la URL y los atributos adicionales que se añadirán a la etiqueta `<a>`.

Es importante tener en cuenta que la URL no se corresponde con la cadena obtenida por las funciones `url()` y `path()`, sino que debe ser un objeto de tipo `\Drupal\Core\Url`.

```
{% for item in items %}
  <li{{ item.attributes }}>
    {{ link(item.title, item.url, {'class':['foo', 'bar']}) }}
  </li>
{% endfor %}
```

attach_library(\$library)

Adjunta una librería a la plantilla.

```
{{ attach_library('classy/node') }}
```

Argumentos con nombre (named arguments)

En las llamadas a funciones, Twig permite que especifiquemos el nombre de cada argumento, de la siguiente forma:

```
{% for i in range(low=1, high=10, step=2) %}
  {{ i }},
{% endfor %}
```

Los nombres de los argumentos son los establecidos en la declaración de la función.

Las ventajas de los argumentos con nombre frente a los argumentos posicionales son:

- Mejorar la legibilidad del código, al conocer el significado de cada parámetro sin necesidad de recurrir a la API de la función.
- Saltarnos parámetros en la llamada a la función. Los parámetros no establecidos usarán los valores por defecto.
- Reordenar los parámetros pasados a la función. Al estar nombrados, no importará el orden en que se pasen a la función.

También podemos combinar argumentos posicionales y con nombre en la llamada a una función. En ese caso, antepondremos siempre los argumentos posicionales, en el orden esperado por la función.

Herencia de plantillas: block, extends, parent()

Una de las opciones más potentes de Twig es la posibilidad de heredar desde otras plantillas. En la plantilla "padre", o plantilla base, se pueden definir bloques mediante la etiqueta **block**, y esos bloques serán los fragmentos que podrán modificar las plantillas hijas.

Veamos un ejemplo. En primer lugar definimos una plantilla base con nombre de archivo base.html.twig **[F4.5a]**. Esta plantilla servirá como primer nivel, e incluye los elementos típicos de una página HTML (etiquetas <html>, <head> y <body>).

Dentro de esta plantilla se ha definido varios bloques:

- **block head.** Contenido del encabezado <head> del documento. Dentro del encabezado "base" se ha añadido una hoja de estilos css/style.css.
- **block title.** Bloque que a su vez está dentro del bloque head. Sirve para imprimir el atributo title de la página actual. Este bloque está vacío y serán las plantillas hijas las que lo rellenen.
- **block content.** Bloque para el contenido. En la plantilla base este bloque está completamente vacío y serán las plantillas hijas las que lo rellenen.
- **block footer.** Bloque para el pie de página, con el texto de copyright.

F4.5a

Herencia de plantillas

Plantilla base.

```
{# archivo base.html.twig #}
<!DOCTYPE html>
<html>
  <head>
    {% block head %}
      <link rel="stylesheet" href="css/style.css" />
      <title>{% block title %}{% endblock %} - Forcontu</title>
    {% endblock %}
  </head>
  <body>
    <div id="content">{% block content %}{% endblock %}</div>
    <div id="footer">
      {% block footer %}
        &copy; Copyright 2018 by
        <a href="https://www.forcontu.com">Forcontu</a>.
      {% endblock %}
    </div>
  </body>
</html>
```

Ahora vamos a crear una plantilla que hereda de la plantilla anterior, completando la información o añadiendo información adicional en los bloques **[F4.5b]**.

```

{# archivo content.html.twig #}

{% extends "base.html.twig" %}

{% block title %}{{ title }}{% endblock %}
{% block head %}
  {{ parent() }}
  <link rel="stylesheet" href="css/custom.css" />
{% endblock %}
{% block content %}
  <h1>{{ title }}</h1>
  <div>{{ content }}</div>
{% endblock %}

```

F4.5b**Herencia de plantillas**

Plantilla hija, hereda de la plantilla base.

- Al **block title** se le está pasando el valor de la variable `{{ title }}`
- En el **block head** primero llamamos a la función `parent()`, con lo que se reutiliza todo el contenido de ese bloque en la plantilla base. Luego se añade una referencia a otra hoja de estilos.
- Completamos el **block content** usando HTML y las variables `title` y `content`. Por supuesto el sistema se tiene que encargar de que esas variables lleguen a la plantilla.
- No hacemos referencia al **block footer**, así que se utilizará el bloque de la plantilla base.

El HTML resultante una vez procesadas las plantillas con sus correspondientes variables, tendría este aspecto **[F4.5c]**:

```

<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/style.css" />
    <title>Home - Forcontu</title>
    <link rel="stylesheet" href="css/custom.css" />
  </head>
  <body>
    <div id="content"><h1>Home</h1>
    <div>Contenido de la página</div>
  </div>
  <div id="footer">
    &copy; Copyright 2018 by
    <a href="http://www.forcontu.com">Forcontu</a>.
  </div>
</body>
</html>

```

F4.5c**Herencia de plantillas**

Código HTML resultante.

Expresiones

Por último, vamos a ver algunas de las expresiones, tipos de variables y operadores que podemos utilizar en Twig:

Arrays

Los arrays se expresan entre corchetes `[]` y con sus elementos separados por comas.

```
["foo", "bar", "barz"]
```

Los **arrays asociativos** se definen con llaves `{}`, y con elementos clave:valor.

Cuando las claves son cadenas, se pueden escribir entre comillas simples o sin comillas:

```
{ 'foo': 'foo', 'bar': 'bar' }  
{ foo: 'foo', bar: 'bar' }
```

Las claves también pueden ser numéricas:

```
{ 2: 'foo', 4: 'bar' }
```

Las claves también pueden ser expresiones, siempre entre paréntesis:

```
{ (1 + 1): 'foo', (a ~ 'b'): 'bar' }
```

Operaciones matemáticas

Además de los operadores habituales, `+`, `-`, `/`, `%` (resto) y `*`, también podemos usar estos otros operadores:

- `//`. Divide dos números y devuelve el valor entero redondeado.
- `**`. Elevado a. `{ { 2 ** 3 } }` significa 2 elevado a 3, cuyo resultado es 8.

Operadores lógicos

Los operadores lógicos disponibles son `and`, `or` y `not`. Los paréntesis sirven para agrupar expresiones (`expr`).

Operadores de comparación

Además de los operadores básicos: `==`, `!=`, `<`, `>`, `>=` y `<=`, podemos usar los siguientes:

- **starts with**, **ends with**. Comprueba si una cadena empieza o termina por otra cadena.

```
{% if 'foo' starts with 'f' %}
{% endif %}

{% if 'foo' ends with 'o' %}
{% endif %}
```

Operador in

El operador **in** comprueba si un elemento está contenido en un array o en una cadena. También se puede usar negado, **not in**, para comprobar que el elemento no está en la lista. Cuando se cumple la condición, devuelve true.

```
{% if 1 in [1, 2, 3] %}

{% if 5 not in [1, 2, 3] %}

{% if 'cd' in 'abcde' %}
```

Operador is

El operador **is** comprueba si una variable cumple una expresión. Es conocido como operador de test, siendo el test la expresión con la que se compara la variable. También se puede utilizar como negación, **is not**.

```
{% if value is odd %}

{% if post.status is constant('Post::PUBLISHED') %}

{% if post.status is not constant('Post::PUBLISHED') %}
```

Operador ..

Equivale a la función `range()`, creando una secuencia entre un mínimo y un máximo.

```
{{ 1..5 }}

{{ range(1, 5) }}
```

Combinando el operador **..** con el filtro **join**, podemos obtener una cadena cuyos valores están separados por el separador indicado:

```
{{ (1..5)|join(',') }}

{# Salida: 1, 2, 3, 4, 5 #}
```

Operador ~

Convierte todos los operandos en cadenas y las concatena.

```
{% set name = 'Fran' %}
{{ "Hello, " ~ name ~ "!" }}
{# Hello, Fran! #}
```

Operador ternario

El operador ternario (A ? B : C) funciona igual que en PHP. Se evalúa la expresión A. Si el resultado es true, devuelve B, y si es false, devuelve C.

```
{{ foo ? 'yes' : 'no' }}
```

Otras variantes:

```
{{ foo ?: 'no' }} equivale a {{ foo ? foo : 'no' }}
{{ foo ? 'yes' }} equivale a {{ foo ? 'yes' : '' }}
```

Etiqueta spaceless

La etiqueta **spaceless** elimina espacios y saltos de línea entre etiquetas HTML, respetando los espacios que puedan tener las cadenas.

```
{% spaceless %}
  <div>
    <strong>foo bar</strong>
  </div>
{% endspaceless %}

{# Salida: <div><strong>foo bar</strong></div> #}
```

Atributos HTML

En Drupal, muchas plantillas reciben la variable `{{ attributes }}`, con atributos HTML para incorporar en una etiqueta. Podemos imprimir todos los atributos de una vez, con `{{ attributes }}`, o acceder a los elementos dentro de `attributes` de forma individual `{{ attributes.id }}`, `{{ attributes.class }}`, etc.

```
<div id="{{ attributes.id }}" class="{{ attributes.class }}"{{
  attributes }}>
  {{ content }}
</div>
```

Nótese que, en el ejemplo, aunque hemos usado los atributos de forma individual, al final de la etiqueta se añade nuevamente `{{ attributes }}`, para que se impriman otros atributos que puedan provenir de cualquier módulo.

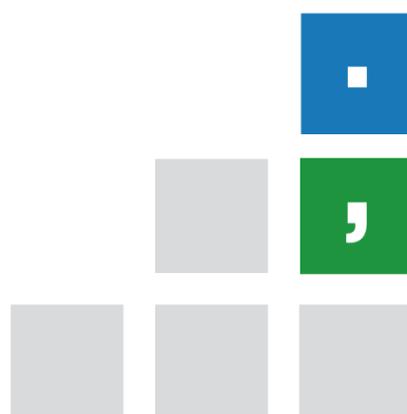
Fran Gil

Front-End Development I

HTML5

5

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

5 HTML5

HTML (HyperText Mark-up Language, Lenguaje de marcado hipertextual) es el código que hace posible presentar la información en Internet. Los navegadores web (Chrome, Firefox, Internet Explorer, Safari, Opera, etc.) interpretan el código HTML y lo convierten en la representación gráfica correspondiente.

En esta unidad vamos a realizar una introducción a la creación de páginas web con HTML, dejando la maquetación con CSS para la próxima unidad. No se pretende abarcar todas las opciones posibles, sino únicamente introducir estos lenguajes para que puedas realizar modificaciones en las plantillas y los contenidos del sitio.

Comparativa D8/D7

Lo estudiado en esta unidad es compatible con cualquier versión de Drupal que utilice HTML5.

Contenidos de la Unidad

- 5.1 Estructura de un documento HTML
- 5.2 Etiquetas HTML en el encabezado
- 5.3 Estructura del cuerpo
- 5.4 Otras etiquetas HTML
- 5.5 Formularios HTML
- 5.6 HTML en Drupal

5

5.1 Estructura de un documento HTML

Introducción a HTML

HTML (*HyperText Mark-up Language*, Lenguaje de marcado hipertextual) es el código que hace posible presentar la información en Internet. Los navegadores web (Chrome, Firefox, Internet Explorer, Safari, Opera, etc.) interpretan el código HTML y lo convierten en la representación gráfica correspondiente.

En el caso particular de Drupal, éste genera el código final HTML teniendo en cuenta las plantillas del tema y la configuración del sitio (módulos, contenidos, bloques, etc.). Una vez generado el HTML, se envía al navegador, que será el encargado de interpretarlo y mostrar el resultado en pantalla. Esta operación se repite cada vez que cargamos una nueva página de nuestro sitio.

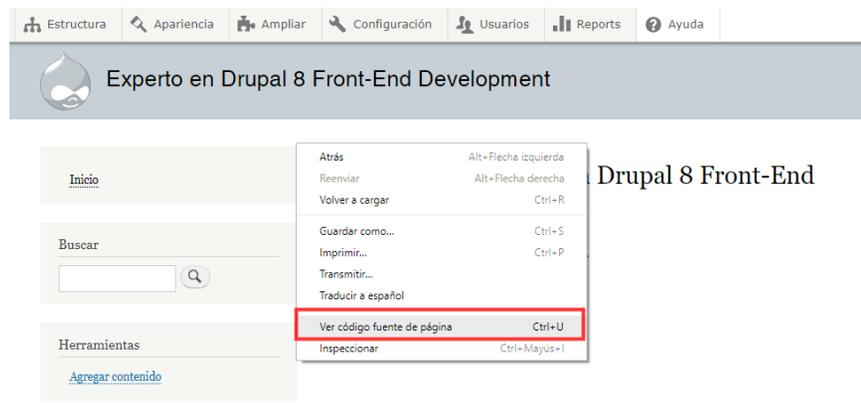
Ver código fuente de página

Para ver el código HTML que llega al navegador, carga la página principal de tu sitio Drupal (fe1). Haciendo clic con botón derecho del ratón, selecciona la opción "Ver código fuente de página" (o "Inspeccionar elemento" si estás sobre algún elemento de la página).

F5.1a

Ver código fuente

El código fuente HTML es accesible desde el navegador.



El código que verás será de este estilo:

```
<!DOCTYPE html>
<html lang="es" dir="ltr" prefix="...">
  <head>
    <meta charset="utf-8" />
    <meta name="Generator" content="Drupal 8 (https://www.drupal.org)" />
    <meta name="MobileOptimized" content="width" />
    <meta name="HandheldFriendly" content="true" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="shortcut icon" href="/core/misc/favicon.ico"
  type="image/vnd.microsoft.icon" />
    <link rel="alternate" type="application/rss+xml" title=""
  href="http://fel.desal.training.forcontu.com/rss.xml" />

    <title>Bienvenido a Experto en Drupal 8 Front-End Development |
  Experto en Drupal 8 Front-End Development</title>
```

```

<link rel="stylesheet"
href="/sites/default/files/css/css_R_6vm3WffQ760L7tOso1MrCvb2yhkuMBF96k
0Uhz_dw.css?p9w72x" media="all" />
</head>
<body class="layout-two-sidebars path-frontpage">
<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-
canvas>
<div id="page-wrapper">
<div id="page">
<header id="header" class="header" role="banner" aria-
label="Cabecera de sitio">
<div class="section layout-container clearfix">
<div class="clearfix region region-header">
<div id="block-b-forcontu-branding" class="clearfix site-branding block
block-system block-system-branding-block">

    <a href="/" title="Inicio" rel="home" class="site-
branding__logo">

</a>
<div class="site-branding__text">
<div class="site-branding__name">
<a href="/" title="Inicio" rel="home">Experto en Drupal 8
Front-End Development</a>
</div>
</div>
</div>
</div>

```

El lenguaje HTML se compone de elementos que se representan mediante etiquetas. La estructura más común es la siguiente, con una etiqueta de apertura <etiqueta> y una etiqueta de cierre </etiqueta>:

<etiqueta>contenido afectado por la etiqueta**</etiqueta>**

Algunos ejemplos que podemos extraer del código HTML anterior:

- <div>... </div>, que permite crear contenedores para agrupar otros elementos.
- <a>... , que permite crear enlaces.

También hay etiquetas sin cierre, como:

- , para añadir una imagen.
- <meta ... />, para incluir información no visible relacionada con la página (metadatos).

Por tanto, conocer HTML no implica más que conocer el conjunto de etiquetas disponibles y su funcionalidad. En este curso abordaremos las etiquetas principales y podrás hacerte una idea del alcance y posibilidades del lenguaje HTML.

Para una referencia completa de HTML5, puedes consultar este enlace:

- HTML5 Tutorial (en inglés):
<https://www.w3schools.com/html/default.asp>

Para ver las etiquetas disponibles, consulta directamente este enlace:

- <https://www.w3schools.com/tags/>

Para probar el código HTML y ver cómo se representaría en el navegador, puedes utilizar el editor online disponible en w3schools:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Sólo tienes que escribir el código HTML en la columna de la izquierda y hacer clic en el botón **Run**. El resultado se mostrará en la columna de la derecha.

F5.2a Editor HTML online



Diferencias entre HTML y XHTML

El código HTML puede estar escrito como HTML o como XHTML (*X* de *eXtensible*), que es una definición de HTML más actualizada y mejor estructurada. XHTML se basa en XML y es más estricto a la hora de cerrar las etiquetas, anidar elementos, etc.

A nivel práctico esto se traduce en pequeñas diferencias. Por ejemplo, en XHTML todas las etiquetas tienen que tener un cierre, incluso las individuales:

- HTML:
``
- XHTML:
``

Dependiendo del tema base utilizado nos podremos encontrar con uno u otro formato. En general se recomienda utilizar el formato XHTML, cerrando todas las etiquetas.

Comentarios en HTML

Los comentarios en HTML se escriben entre los símbolos:

```
<!-- texto del comentario -->
```

Ya vimos un ejemplo de comentarios al activar la depuración de plantillas, como se observa en el siguiente código:

```
<!-- BEGIN OUTPUT from 'themes/b forcontu/templates/page--
front.html.twig' -->
<div id="page-wrapper">
  <div id="page">
    <header id="header" class="header" role="banner" aria-
label="Cabecera de sitio">
```

Es importante recalcar que los comentarios en HTML, aunque no se ven en la página, **sí se ven al inspeccionar el código fuente**, por lo que tenemos que tener cuidado con no anotar información sensible.

Si queremos escribir un comentario en el código que no pueda ver el usuario final, lo tendremos que añadir a nivel de PHP o de plantilla Twig. En este último caso, dado que una plantilla Twig contiene tanto código Twig como código HTML, el comentario tiene que ir como **comentario de Twig** y no como comentario HTML. Observa las diferencias de ambos tipos de comentarios en el siguiente código de plantilla Twig:

```
{#
  Comentario escrito en Twig.
  No se verá en el código fuente de la página.
#}
<!-- Comentario HTML.
  Sí se verá en el código fuente de la página -->
<article{{ attributes }}>
  {{ title_prefix }}
  <h3>
    <a href="{{ url }}">{{ title }}</a>
  </h3>
  {{ title_suffix }}
  {{ content }}
</article>
```

Estructura principal del documento HTML

La etiqueta DOCTYPE

Todo documento HTML5 comienza con una etiqueta DOCTYPE, escrita de la siguiente forma:

```
<!DOCTYPE html>
<html ...>
```

Las etiquetas html, head y body

Estas etiquetas conforman la estructura básica de un documento HTML. Todas las páginas HTML tienen como mínimo estas etiquetas, con sus correspondientes etiquetas de cierre.

El documento HTML empieza y termina con la etiqueta `<html></html>`, que se considera la etiqueta raíz (root). Dentro de la etiqueta `<html>` existen dos bloques principales, el encabezado `<head></head>`, donde se insertan otras etiquetas con información de la página, como el título (`<title></title>`); y el cuerpo `<body></body>`, donde se insertan los contenidos de la página.

A continuación se muestra el código básico de una página HTML.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Título de la página</title>
  </head>

  <body>
    <p>Contenido en un párrafo.</p>
  </body>
</html>
```

Dentro de la etiqueta `html` necesitamos especificar el idioma de la página, a través del atributo `lang`. Por ejemplo, `<html lang="es">`, si la página está en español.

Etiquetas HTML en el encabezado

5.2

La etiqueta `<head>` es un contenedor para otras etiquetas relacionadas con el encabezado de la página. Estas etiquetas aportan información adicional a la página (enlaces a archivos CSS y JS, metadatos, etc.).

https://www.w3schools.com/tags/tag_head.asp

Algunas de las etiquetas que se pueden utilizar en el encabezado son:

`<title>`

Es una etiqueta obligatoria en la que se define el título del documento HTML.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Quiénes somos | Experto en Drupal 8 Front-End
    Development</title>
  </head>
```

No hay que confundir el título del documento o página (page title) que definimos en la etiqueta `<title>` (1), con el título del contenido (2). El título de la página se puede ver en la pestaña del navegador y se utiliza para referenciar la página en los resultados de los buscadores externos tipo Google. El título de la página no tiene por qué coincidir con el título del contenido.

The screenshot shows a browser window with the following elements:

- Browser Tab:** The title bar shows "Quiénes somos | Experto en Drupal 8 Front-End Development" with a red box and the number 1 next to it.
- Page Content:** The main heading of the page is "Quiénes somos" with a red box and the number 2 next to it.
- Page Structure:** The page includes a navigation menu with "Inicio", a "User login" section with a "Nombre de usuario" input field, and a main content area with a Lorem Ipsum placeholder.

F5.2a

`<title>`

Título del documento HTML.

`<meta>`

Las etiquetas `<meta>`, también llamadas metaetiquetas, permiten añadir información adicional o metadata relacionada con la página y los datos que contiene. Algunas de ellas están relacionadas con el posicionamiento SEO, ya que facilitan información a los robots de búsqueda (description, keywords, robots, author, etc.). Otras informan al navegador sobre cómo presentar la información (charset, viewport).

https://www.w3schools.com/tags/tag_meta.asp

Veamos algunos ejemplos de metaetiquetas:

```
<meta charset="utf-8" />
```

- <meta **charset**="" />. Juego de caracteres utilizado en la página. En Drupal 8 utilizamos generalmente utf-8.

```
<meta name="Generator" content="Drupal 8" />
```

- <meta name="**Generator**"... />. Software que genera la página. En nuestro caso, Drupal 8.

```
<meta name="description" content="Experto en Drupal 8" />
```

- <meta name="**description**"... />. Descripción del contenido.

```
<meta name="keywords" content="Drupal, HTML, Forcontu" />
```

- <meta name="**keywords**"... />. Palabras clave para categorizar la página. Hoy en día los buscadores no suelen utilizar directamente esta información, basándose más en el contenido real de la página y las palabras o cadenas que más se repiten o más se destacan dentro del contenido.
- <meta name="**viewport**"... />

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

La metaetiqueta viewport fue introducida en HTML5 para facilitar el diseño responsive. "viewport" hace referencia a la parte visible de la página. A través de esta etiqueta podemos indicarle al navegador cómo debe escalar la página en función del dispositivo.

- 'width=device-width', indica que el ancho de la página se debe adaptar al ancho del dispositivo.
- 'initial-scale=1.0', indica el nivel de zoom inicial cuando el navegador carga la página. El usuario puede posteriormente cambiar el zoom de la página.

Es una etiqueta que se añade por defecto y generalmente no necesitamos modificar sus valores.

- `<meta name="HandheldFriendly"... />`

```
<meta name="HandheldFriendly" content="true" />
```

Indica al navegador que el sitio está ya preparado para navegación con dispositivos móviles.

- `<meta name="MobileOptimized"... />`

```
<meta name="MobileOptimized" content="width" />
```

Esta etiqueta fue introducida por Windows y permite indicar el tamaño por defecto al cargar la página en un dispositivo móvil.

De estas etiquetas de Diseño Responsive, la más utilizada y que han adoptado la mayoría de navegadores es 'viewport'. Veremos más opciones de configuración de estas etiquetas en la unidad dedicada al Diseño Responsive.

- `<meta http-equiv="refresh" content="30">`. Refresca el documento cada x segundos (30 en el ejemplo).

```
<meta http-equiv="refresh" content="30" />
```

<link>

La etiqueta <link> sirve para incluir en la página recursos adicionales. Por ejemplo, se utiliza para añadir archivos de hojas de estilo CSS.

https://www.w3schools.com/tags/tag_link.asp

El parámetro rel es obligatorio, estableciendo el tipo de relación entre la página y el elemento enlazado.

stylesheet

```
<link rel="stylesheet" type="text/css" href="css/styles.css"
media="all" />
```

type

<https://www.iana.org/assignments/media-types/media-types.xhtml>

shortcut icon

```
<link rel="shortcut icon" href="/core/misc/favicon.ico"
type="image/vnd.microsoft.icon" />
```

canonical

```
<link rel="canonical" href="/quienes-somos" />
```

Cuando a un contenido se puede acceder a través de varias URLs distintas, es necesario indicar a los buscadores cuál es la URL principal. Se evita así que los buscadores indexen contenido duplicado, con la penalización que suele suponer a nivel de SEO.

En Drupal es muy común que los contenidos tengan al menos dos URLs, la de sistema (/node/1) y el alias de URL (/quienes-somos). Aunque la URL de sistema podría no llegar a indexarse ni estar accesible para los buscadores, es necesario indicar que la URL canónica, la que se tiene que indexar, es la URL con alias. Esto lo hace Drupal por defecto, sin necesidad de configuración adicional.

```
<link rel="shortlink" href="/node/1" />
```

Además, Drupal añade por defecto otras URLs relacionadas con la edición de contenidos:

```
<link rel="delete-form" href="/node/1/delete" />
<link rel="edit-form" href="/node/1/edit" />
<link rel="version-history" href="/node/1/revisions" />
<link rel="revision" href="/node/1" />
```

<script>

Permite añadir código que se ejecutará en el lado del cliente. Salvo que se especifique lo contrario, el tipo de código esperado será JavaScript.

https://www.w3schools.com/tags/tag_script.asp

Si se especifica el atributo **src**, estaremos haciendo referencia a la URL de un archivo externo donde se encuentra el código a ejecutar.

```
<script
src="/core/assets/vendor/modernizr/modernizr.min.js?v=3.3.1"
/></script>
```

Aunque lo más habitual es hacer referencia a un archivo con el código (src), también permite añadir código directamente, encerrado entre las etiquetas de apertura y cierre.

```
<script>
  Webprofiler = (function () {
    "use strict";

    var key = 'webprofiler/profiler/',
        // ...
    return {
      getPreference: getPreference,

```

```

        setPreference: setPreference,
        load: load,
        ajax: ajax
    }
}() );

Webprofiler.load();
</script>

```

Si no añadimos el atributo 'type', se considerará que el script contiene JavaScript. Si vamos a añadir código en otro lenguaje, lo tendremos que especificar.

```

<script type="application/json" data-drupal-selector="drupal-
settings-json">{...}</script>

```

Comentarios condicionales

Aún es común encontrar comentarios condicionales para añadir scripts (CSS o JS) que sólo se ejecutarán en determinadas versiones de Internet Explorer (IE). Se busca con esto dar alternativas para garantizar la compatibilidad con navegadores "antiguos".

Los comentarios condicionales tienen el formato:

```
<!--[if expresion]> Comentario condicional <![endif]-->
```

Donde las comparaciones posibles son: "menor que" (lt), "mayor que" (gt), "menor o igual que" (lte), "mayor o igual que" (gte). También se pueden utilizar los operadores ! (negado), & (and) y | (or).

En el siguiente ejemplo, se incluirá el script foo.js si el navegador en el que se carga la página es Internet Explorer 8 o una versión anterior.

```

<!--[if lte IE 8]>
<script src="/sites/default/files/js/foo.js"></script>
<![endif]-->

```

Cuando se cumple la condición, el navegador interpreta el código HTML incluido en el comentario. Para el resto de navegadores se considerará un comentario HTML ordinario y se ignorará totalmente.

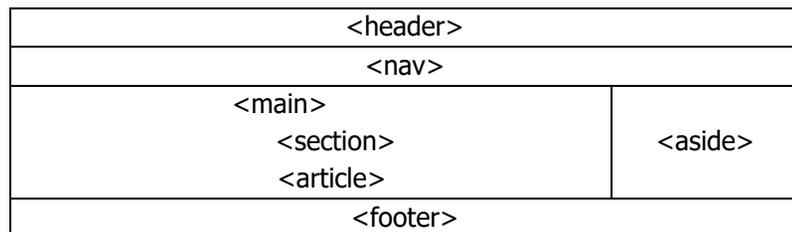
5.3 Estructura del cuerpo

Dentro de la etiqueta `<body>` se definirán los contenidos de la página. En este apartado vamos a ver cómo se organizan los contenidos dentro de la etiqueta `<body>`.

En versiones anteriores de HTML se utilizaban las etiquetas `<div>` para crear contenedores y, con ello, las diferentes regiones o secciones que puede tener una página web (encabezado, columna central, columnas laterales, pie de página, etc.). Las etiquetas `<div>` se consideran **elementos no semánticos**, ya que no aportan por sí mismas información sobre su significado.

En HTML5 se han añadido nuevos **elementos semánticos**, que aportan significado y permiten estructurar mejor los contenidos de la página. Algunos de estos elementos semánticos se representan en la Figura [F5.3a]:

F5.3a
Estructura de página
Elementos semánticos para organizar la estructura de la página.



`<section>`

El elemento `<section>` se utiliza para definir una sección dentro del documento. Una sección se define como una agrupación de contenido relacionado. No existe una forma estricta de aplicar este elemento, así que queda en manos del diseñador definir qué se considera una sección dentro de la página. Debemos cerrar la etiqueta `</section>` al finalizar la sección.

```
<section>
  <h1>Quiénes somos</h1>
  <p>Lorem ipsum dolor sit amet, felis. Nulla facilisi. Aliquam
  rat...</p>
</section>
```

https://www.w3schools.com/tags/tag_section.asp

`<article>`

Un artículo `<article>` es un contenido que podemos leer de forma independiente al resto de la página. Por ejemplo, una entrada en un foro o un blog se puede consultar de forma independiente, sin necesidad de leer otras entradas escritas en la misma página.

```
<article>
  <h1>Question 1</h1>
```

```

<p>Curabitur eleifend urna a suscipit egestas. Aenean vitae
ultrices turpis, sit amet euismod turpis.</p>
</article>
<article>
  <h1>Question 2</h1>
  <p>Nullam commodo viverra nibh vel venenatis. Suspendisse
pharetra semper nulla, vitae pellentesque ligula.</p>
</article>

```

https://www.w3schools.com/tags/tag_article.asp

Anidación de elementos

Los elementos `<section>` y `<article>` se pueden anidar en función de la estructura de la página. Un `<section>` puede contener uno o varios `<article>`, pero también puede ocurrir que un `<article>` se divida en secciones mediante elementos `<section>`. También es habitual encontrar `<section>` que se dividen a su vez en subsecciones con elementos `<section>` adicionales, o artículos que contienen otros artículos.

En el siguiente ejemplo hemos definido una sección conteniendo un artículo. El artículo tiene una sección con información de contacto.

```

<section>
  <article>
    <h1>Question 1</h1>
    <p>Curabitur eleifend urna a suscipit egestas... </p>
    <section>
      <p>Contact information</p>
    </section>
  </article>
</section>

```

`<header>`

Aunque en la Figura [F5.3a] hemos representado el elemento `<header>` como encabezado de toda la página, también se puede utilizar como encabezado de otros elementos, como secciones `<section>` o artículos `<article>`.

En el siguiente ejemplo utilizamos el elemento `<header>` para definir un encabezado en un elemento `<article>`:

```

<article>
  <header>
    <h1>Question 1</h1>
  </header>
  <p>Curabitur eleifend urna a suscipit egestas. Aenean vitae
ultrices turpis, sit amet euismod turpis.</p>
</article>

```

https://www.w3schools.com/tags/tag_header.asp

<footer>

Al igual que el encabezado, el footer define un pie de la página principal o de otros elementos como secciones o artículos. Se puede utilizar para añadir información adicional (contacto, enlaces relacionados, autoría y copyright, etc.).

En el siguiente ejemplo se ha añadido un pie a un artículo, con información sobre el autor del contenido:

```
<article>
  <header>
    <h1>Question 1</h1>
  </header>
  <p>Curabitur eleifend urna a suscipit egestas. Aenean vitae
  ultrices turpis, sit amet euismod turpis.</p>
  <footer>
    <p>Posted by: John Doe</p>
  </footer>
</article>
```

https://www.w3schools.com/tags/tag_footer.asp

<nav>

La etiqueta <nav> se utiliza para definir un bloque con enlaces de navegación, que no es otra cosa que un **menú**.

Como se muestra en el siguiente ejemplo, es muy común el uso del elemento <nav> con estructuras de lista (), que estudiaremos más adelante.

```
<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/quienes-somos">Quiénes somos</a></li>
  </ul>
</nav>
```

https://www.w3schools.com/tags/tag_nav.asp

<main>

Se utiliza para delimitar el contenido principal de la página. Generalmente una página tiene elementos que se repiten entre páginas, como el encabezado y pie, las barras de navegación, las columnas laterales. El contenido principal es el "centro" de la página, el contenido único que no se repite entre páginas.

Sólo debe haber un elemento <main> en la página, y no estar anidado dentro de otros elementos (<section>, <article>, etc.).

```
<header>
  ...
</header>
<main>
  <article>
    <header>
```

```

    <h1>Question 1</h1>
  </header>
  <p>Curabitur eleifend urna a suscipit egestas...</p>
  <footer>
    <p>Posted by: John Doe</p>
  </footer>
</article>
</main>
<footer>
  ...
</footer>

```

https://www.w3schools.com/tags/tag_main.asp

<aside>

El elemento <aside> se utiliza para englobar contenido adicional separado del contenido principal del elemento con el que se relaciona. Generalmente se utiliza para diferenciar las columnas laterales (sidebar) del contenido principal (<main>).

```

<header>
  ...
</header>
<main>
  ... <!-- contenido principal -->
</main>
<aside>
  ... <!-- columna lateral -->
</aside>
<footer>
  ...
</footer>

```

https://www.w3schools.com/tags/tag_aside.asp

Otros elementos semánticos:

- **<details>**. Se utiliza para agrupar información que el usuario puede mostrar u ocultar (plegar). El elemento **<summary>** permite añadir un encabezado visible dentro del elemento <details>.
- **<figure>**. Añade elementos adicionales al contenido, como figuras, imágenes, diagramas, etc. El elemento **<figcaption>** permite añadir un título al elemento <figure>.
- **<mark>**. Permite resaltar texto.
- **<time>**. Permite añadir una fecha/hora.

<div>

Por último, la etiqueta <div> permite delimitar otras secciones o divisiones en la página sin aportar un significado específico.

5.4 Otras etiquetas HTML

En este apartado veremos otras etiquetas o elementos que usaremos dentro de las etiquetas estructurales estudiadas en los apartados anteriores.

Encabezados <h1>...<h6>

Los encabezados (<h1>, <h2>, <h3>, <h4>, <h5> y <h6>) son etiquetas que permiten definir distintos niveles de encabezado. Se aplican a los títulos y subtítulos de las secciones y sirven para estructurar el contenido de una página por niveles y subniveles, como se haría con los índices de cualquier libro de texto. Por defecto los encabezados se representan en diferentes tamaños, siendo **h1** el más grande y **h6** el más pequeño, pero la utilidad real no está basada en el tamaño, sino en la jerarquía. Además, los estilos se pueden cambiar con CSS, pero se ha de mantener una correcta organización de los mismos.

```
<h1>Apartado 1</h1>
<h2>Apartado 1.1</h2>
<h2>Apartado 1.2</h2>
<h3>Apartado 1.2.1</h3>
<h3>Apartado 1.2.2</h3>

<h1>Apartado 2</h1>
```

F5.4a

Encabezados en HTML

Los encabezados h1 a h6 sirven para estructurar el contenido del sitio. Se aplican a los títulos de cada sección

Apartado 1

Apartado 1.1

Apartado 1.2

Apartado 1.2.1

Apartado 1.2.2

Apartado 2

https://www.w3schools.com/tags/tag_hn.asp

<p>,
, <hr>, , <i>

Con la etiqueta <p> estamos escribiendo un párrafo de texto. La etiqueta
 representa un salto de línea. Recuerda que en XHTML se debe abrir y cerrar en la misma etiqueta
.

<hr> dibuja una línea horizontal a modo de separador. La etiqueta resalta el texto en negrita, mientras que la etiqueta <i> lo presenta en cursiva.

```
<h1>Apartado 1</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Aliquam faucibus massa sit amet augue venenatis porttitor. Sed sit
amet nunc vitae justo elementum ornare nec in leo. Pellentesque
rhoncus viverra velit in ullamcorper. </p>
<hr>
<h1>Apartado 2</h1>
<p><strong>Lorem ipsum dolor sit amet</strong>, <i>consectetur
adipiscing elit</i>.<br>Aliquam faucibus massa sit amet augue
venenatis porttitor.<br>Sed sit amet nunc vitae justo elementum
ornare nec in leo.<br>Pellentesque rhoncus viverra velit in
ullamcorper. </p>
```

Apartado 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam faucibus massa sit amet augue venenatis porttitor. Sed sit amet nunc vitae justo elementum ornare nec in leo. Pellentesque rhoncus viverra velit in ullamcorper.

Apartado 2

Lorem ipsum dolor sit amet, *consectetur adipiscing elit*. Aliquam faucibus massa sit amet augue venenatis porttitor. Sed sit amet nunc vitae justo elementum ornare nec in leo. Pellentesque rhoncus viverra velit in ullamcorper.

https://www.w3schools.com/tags/tag_p.asp
https://www.w3schools.com/tags/tag_br.asp
https://www.w3schools.com/tags/tag_hr.asp
https://www.w3schools.com/tags/tag_strong.asp
https://www.w3schools.com/tags/tag_i.asp

La etiqueta se utiliza para poder aplicar estilos a un texto dentro de otro texto. La etiqueta no aporta por sí sola ningún cambio de estilos, que habrá que aplicar vía CSS, como veremos más adelante. En el siguiente ejemplo aplicamos el color rojo a parte del texto.

```
<p>Lorem ipsum dolor sit amet
<span style="color:red">consectetur</span> adipiscing elit.</p>
```

Lorem ipsum dolor sit amet **consectetur** adipiscing elit.

https://www.w3schools.com/tags/tag_span.asp

F5.4b

Ejemplo de uso de etiquetas HTML

Ejemplo de uso de las etiquetas p, hr, br, i.

F5.4c

Ejemplo de uso.

Listas de elementos , ,

Es posible representar listas de elementos tanto no ordenadas (, *unordered list*) como ordenadas (, *ordered list*). En ambos tipos de listas, los elementos se separan unos de otros con las etiquetas .

F5.4d

Listas

Ejemplos de listas no ordenadas (ul) y ordenadas (ol) en HTML.

```
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
</ul>

<ol>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
</ol>
```

- Elemento 1
- Elemento 2
- Elemento 3

1. Elemento 1
2. Elemento 2
3. Elemento 3

https://www.w3schools.com/tags/tag_ul.asp

https://www.w3schools.com/tags/tag_ol.asp

https://www.w3schools.com/tags/tag_li.asp

Enlaces <a>

Algunas etiquetas pueden contener atributos. Los atributos permiten añadir información adicional a los elementos. Por ejemplo, la etiqueta <a> se utiliza para crear hipervínculos o enlaces a otras páginas. La forma en que le indicamos la URL a la que enlaza el elemento es mediante un atributo denominado **href**. La nomenclatura es la siguiente:

```
<a href="https://www.forcontu.com/sobre-forcontu">Sobre
Forcontu</a>
```

El atributo **href** puede contener una url completa a otro sitio web (ruta absoluta) o una url relativa a una página de nuestro sitio web.

```
<a href="/sobre-forcontu">Sobre Forcontu</a>
```

Otros dos atributos importantes para la etiqueta <a> son **title** y **target**. El atributo title permite indicar un texto descriptivo del enlace. El atributo target sirve para indicar en qué página se abrirá la página enlazada (target="_self" indica que la página se abrirá en la misma ventana, sustituyendo a la página actual; target="_blank" indica que la página se abrirá en una nueva ventana).

```
<a href="http://www.forcontu.com/sobre-forcontu" title="Más información sobre Forcontu" target="_blank">Sobre Forcontu</a>
```

https://www.w3schools.com/tags/tag_a.asp

Tablas `<table>`, `<tr>`, `<td>`, `<th>`

Las tablas se usan cuando queremos mostrar datos en forma de tabla. En las tablas las etiquetas `<tr></tr>` definen las filas, y las etiquetas `<td></td>` definen las columnas en cada fila. Cuando las celdas hacen referencia a un encabezado, reemplazamos la etiqueta `<td>` por `<th>`.

En el siguiente ejemplo hemos añadido un borde a los elementos de la tabla para una mejor visualización. Veremos más adelante el uso de `<style>` para aplicar estilos CSS.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
border: 1px solid black;
}
</style>
</head>
<body>

<table>
<tr>
<th>Encabezado 1</th>
<th>Encabezado 2</th>
</tr>
<tr>
<td>Fila 1, columna 1</td>
<td>Fila 1, columna 2</td>
</tr>
<tr>
<td> Fila 2, columna 1</td>
<td> Fila 1, columna 2</td>
</tr>
</table>

</body>
</html>
```

F5.4e

Tablas

Ejemplo de creación de tabla en HTML.

Encabezado 1	Encabezado 2
Fila 1, columna 1	Fila 1, columna 2
Fila 2, columna 1	Fila 1, columna 2

https://www.w3schools.com/tags/tag_table.asp

Imágenes

Para insertar imágenes se utiliza la etiqueta . Hay dos atributos obligatorios:

- **src.** Indica la url de la imagen que queremos mostrar.
- **alt.** Indica un texto alternativo o descriptivo de la imagen para los navegadores que no pueden mostrar imágenes o que tienen esta opción deshabilitada.

Otros atributos opcionales son:

- **width.** Indica el ancho de la imagen en píxeles.
- **height.** Indica el alto de la imagen en píxeles.

Si queremos enlazar una imagen con otro contenido, sólo tenemos que anidar la etiqueta dentro de una etiqueta de enlace <a>.

F5.4f

Imágenes

Uso de imágenes en HTML.

```
<p>
Esta imagen es un enlace:
<a href="https://www.forcontu.com">

</a>
</p>
```

Esta imagen es un enlace: 

https://www.w3schools.com/tags/tag_img.asp

Formularios HTML

5.5

Los formularios permiten solicitar información al usuario y enviarla a otra página para su procesamiento.

En Drupal los formularios se pueden crear de varias formas:

- Módulos de creación de formularios como Webform o Contact (curso Site Building).
- Formularios programados utilizando la API de Formularios (curso Back-End Development).

Independientemente de cómo se generen estos formularios, el resultado final es código HTML interpretable por el navegador. Por tanto, necesitamos conocer los elementos de formulario de HTML no para crear formularios desde cero, sino para, una vez generados desde Drupal, poder maquetarlos e interactuar con ellos a nivel de Front-End.

<form>

El formulario se define mediante la etiqueta <form>. Dentro de esta etiqueta se anidarán los elementos de formulario. Veamos un ejemplo de formulario básico con un elemento de entrada de texto y un botón de envío o submit.

```
<form action="/process.php" name="customform" method="post">
  Subject: <input type="text" name="subject" value="Lorem ipsum"><br>
  <input type="submit" value="Enviar">
</form>
```

F5.5a

Formularios

Uso de la etiqueta <form> para definir un formulario HTML.

Subject:

Algunos de los atributos que puede llevar la etiqueta <form> son:

- **action.** Permite especificar la URL a la que se enviará la información del formulario.
- **method.** Indica el método HTTP que se utiliza para enviar el formulario. Los valores posibles son:
 - o get. Los valores del formulario se transmiten a través de la URL, en parejas nombre/valor.
 - o post. Los valores del formulario se transmiten internamente, en el cuerpo del request HTTP.
- **name.** Nombre del formulario.

https://www.w3schools.com/tags/tag_form.asp

https://www.w3schools.com/html/html_forms.asp

<input>

La etiqueta <input> añade campos al formulario que permite al usuario introducir datos. El campo <input> tiene múltiples variantes, en función del atributo 'type'. Algunas de estas variantes son:

- type="text". Entrada de texto de una línea.
- type="email". Entrada de texto para introducir un email.
- type="search". Campo de texto utilizado en búsquedas.
- type="url". Campo de texto utilizado para introducir una URL.
- type="password". Entrada de texto para introducir una contraseña de forma oculta.
- type="hidden". Permite enviar un valor de forma oculta. El campo no se muestra en el formulario, pero sí que está disponible en el código HTML.
- type="checkbox". Cuadro de check único.
- type="radio". Selector de radio único.
- type="number". Selección de un número dentro de un rango establecido.
- type="range". Selección de un número dentro de un rango establecido, utilizando un control deslizante (slider).
- type="date". Selección de fecha.
- type="color". Selección de color.
- type="file". Selección de archivo que se adjuntará al formulario.
- type="submit". Botón de envío del formulario.
- type="image". Similar al submit, pero define una imagen como botón de envío.
- type="reset". Botón de reset para limpiar el formulario, devolviendo todos los campos a sus valores por defecto.

Además del tipo, la etiqueta <input> puede tener estos otros atributos:

- disabled="disabled". Indica que el elemento debería estar deshabilitado. Se mostrará el valor actual pero no podrá ser modificado.
- name. Nombre del elemento.
- required="required". Indica que el elemento debe completarse para poder enviar el formulario.
- value. Valor del elemento.

https://www.w3schools.com/tags/tag_input.asp

A continuación, veremos con más detalle algunas de las variantes de <input> junto con los atributos específicos que pueden utilizar. Todos los tipos de input se pueden consultar en:

https://www.w3schools.com/html/html_form_input_types.asp

<input type="text">

Define un elemento de entrada de texto de una única línea.

```
<form>
First name: <input type="text" name="firstname"><br>
Last name: <input type="text" name="lastname">
</form>
```

F5.5b
Input text

First name:

Last name:

Algunos de los atributos específicos que podemos utilizar en un input de tipo text son:

- **maxlength.** Número máximo de caracteres que se pueden escribir en el campo.
- **size.** Tamaño del campo.

En el siguiente campo el tamaño de la caja de texto es de 50 caracteres (size), pero se puede seguir escribiendo hasta llegar a 70 (maxlength).

```
<form>
Subject:
<input type="text" name="subject" size="50" maxlength="70">
</form>
```

F5.5c
Input text

Subject:

<input type="email">

Entrada de texto para introducir un email. Es similar al campo de tipo text, pero específico para introducir un email. Al enviarse el formulario, el navegador comprobará que se haya introducido una dirección de correo válida.

```
<form>
E-mail:
<input type="email" name="email">
<input type="submit">
</form>
```

F5.5d
Input email

E-mail:

 Incluye un signo "@" en la dirección de correo electrónico. La dirección "example" no incluye el signo "@".

El campo email acepta los mismos atributos que el campo de tipo text (size, maxlength, etc.).

<input type="search">

Campo de texto utilizado en búsquedas. Se utiliza exactamente igual que un campo de tipo text.

<input type="url">

Entrada de texto para introducir una dirección URL. Es similar al campo de tipo text, pero específico para introducir una URL. Al enviarse el formulario, el navegador comprobará que se haya introducido una URL válida.

F5.5e Input url

```
<form>
  Escribe una URL:
  <input type="url" name="customurl">
  <input type="submit">
</form>
```

Escribe una URL:

Ingresa una URL.

<input type="password">

Entrada de texto para introducir una contraseña de forma oculta. Ten en cuenta que, aunque visualmente la contraseña se enmascara con asteriscos, el texto se transmite como texto plano, igual que el texto introducido en un campo de tipo text.

F5.5f Input password

```
<form>
  User name: <br>
  <input type="text" name="username"><br>
  User password: <br>
  <input type="password" name="password">
</form>
```

User name:

User password:

<input type="hidden">

Permite enviar un valor de forma oculta. El campo no se muestra en el formulario, pero sí que está disponible en el código HTML. Por tanto, no se debe enviar a través de este campo información sensible o secreta.

```
<input type="hidden" id="productId" name="productId" value="1234">
```

<input type="checkbox">

Añade un cuadro de selección o check, conocido como checkbox.

```
<form>
<input type="checkbox" name="acceptterms" value="yes">I have read
and I accept the terms and conditions
</form>
```

F5.5g
Input checkbox

I have read and I accept the terms and conditions

Si queremos que el cuadro esté activado por defecto, podemos añadir el atributo **checked**:

```
<form>
<input type="checkbox" name="acceptterms" value="yes" checked>I
have read and I accept the terms and conditions
</form>
```

Para crear varios elementos de tipo checkbox asociados, como si se tratara de una lista, añadiremos elementos input de tipo checkbox con el mismo valor de "name".

```
<form>
Selecciona tus deportes favoritos:<br>
<input type="checkbox" name="sports"
value="Baloncesto">Baloncesto<br>
<input type="checkbox" name="sports" value="Fútbol">Fútbol<br>
<input type="checkbox" name="sports" value="Tenis">Tenis<br>
<input type="checkbox" name="sports"
value="Balonmano">Balonmano<br>
</form>
```

F5.5h
Input checkbox múltiple

Selecciona tus deportes favoritos:

- Baloncesto
 - Fútbol
 - Tenis
 - Balonmano
-

Al enviar el formulario, el resultado se recibirá como una única variable de tipo array, con los valores seleccionados.

<input type="radio">

Selector de radio único. Es similar al checkbox pero con un botón de radio (círculo). Al contrario que con el cuadro de tipo checkbox, el elemento de radio no puede deseleccionarse una vez seleccionado.

F5.5i Input radio

```
<form>
<input type="radio" name="acceptterms" value="terms">I have read
and I accept the terms and conditions
</form>
```

I have read and I accept the terms and conditions

Para crear varios elementos de tipo radio asociados, como si se tratara de una lista, añadiremos elementos input de tipo radio con el mismo valor de "name". Al tratarse de botones de tipo radio, sólo se podrá seleccionar uno de los elementos. Al seleccionar otro elemento, el elemento seleccionado anteriormente se desmarcará.

F5.5j Input radio múltiple

```
<form>
  Select your age:<br>
  <input type="radio" name="age" value="10">0-10<br>
  <input type="radio" name="age" value="20">11-20<br>
  <input type="radio" name="age" value="30">21-30
</form>
```

Select your age:

0-10

11-20

21-30

Al enviar el formulario, el resultado se recibirá como una única variable de tipo array, con los valores seleccionados.

<input type="number">

Selección de un número dentro de un rango establecido. Podemos establecer el valor mínimo (min) y el valor máximo (max).

F5.5k Input number

```
<form>
  Escribe un número (entre 1 y 10):
  <input type="number" name="quantity" min="1" max="10">
  <input type="submit">
</form>
```

Escribe un número (entre 1 y 10):

También se puede indicar un salto con el atributo **step**. Por ejemplo, si min="1" y step="2", los valores posibles serán 1, 3, 5, 7...

<input type="range">

Selección de un número dentro de un rango establecido, utilizando un control deslizante (slider). Es similar a number y también se pueden utilizar los atributos min, max y step.

```
<form>
  Points:
  <input type="range" name="points" min="0" max="100">
  <input type="submit">
</form>
```

F5.51
Input range



<input type="date">

Añade un selector de fecha. Dependiendo de la compatibilidad del navegador se mostrará un calendario desplegable o no.

```
<form>
  Fecha de nacimiento:
  <input type="date" name="bday">
  <input type="submit">
</form>
```

F5.5m
Input date



El tipo de input 'datetime-local' es similar pero además de la fecha se permite introducir una hora (hora:minutos).

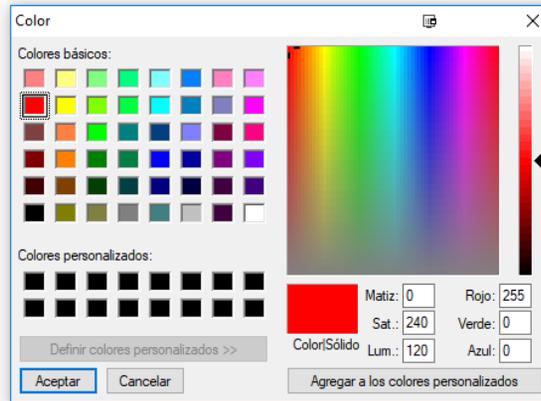
<input type="color">

Añade un selector de color.

F5.5n Input color

```
<form action="/action_page.php">
  Selecciona el color de fondo
  <input type="color" name="backgroundcolor" value="#ff0000">
  <input type="submit">
</form>
```

Selecciona el color de fondo



<input type="file">

Permite seleccionar y adjuntar un archivo al formulario.

F5.5o Input file

```
<form>
  Select a file:
  <input type="file" name="myFile">
  <input type="submit">
</form>
```

Select a file: No se eligió archivo

<input type="submit">

El botón de submit permite enviar el formulario a la dirección definida en el atributo action del elemento form.

F5.5p Input submit

```
<form action="/process.php" name="customform" method="post">
  <input type="submit" value="Enviar">
</form>
```

<input type="image">

El elemento de tipo image permite cambiar el botón de submit por una imagen. La ruta de la imagen se especificará en el atributo src. Si añadimos un elemento de tipo image no necesitamos añadir uno de tipo submit, ya que el funcionamiento es exactamente igual.

```
<form action="/process.php" name="customform" method="post">
  <input type="image" src="img_submit.gif" alt="Submit" width="48"
  height="48">
</form>
```

F5.5q
Input image

<input type="reset">

Botón de reset para limpiar el formulario, devolviendo todos los campos a sus valores por defecto.

```
<form action="/action_page.php">
  First name: <input type="text" name="firstname">
  <br>
  Last name: <input type="text" name="lastname">
  <br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

F5.5r
Input reset

First name:

Last name:

<label>

Añade una etiqueta asociada a un elemento de tipo input. Se trata de información textual asociada al elemento de entrada o input, generalmente utilizado para mejorar la accesibilidad.

Para asociar la etiqueta label con un elemento, se puede hacer de dos formas:

1. Utilizando el atributo for, donde se indicará el id del elemento input al que está vinculado.

```
<label for="name">Nombre</label>
<input type="text" id="name" name="name">
```

2. Anidando el elemento input dentro de la etiqueta label.

```
<label>Nombre <input type="text" id="name" name="name"></label>
```

En ambos casos el resultado será un texto asociado al campo.

F5.5s label

Nombre

https://www.w3schools.com/tags/tag_label.asp

<textarea>

Añade un cuadro de texto de varias líneas. El texto que se añada entre las etiquetas de apertura y cierre se mostrará dentro del cuadro, y podrá ser modificado.

Algunos atributos típicos son:

- **rows.** Número de filas.
- **cols.** Número de caracteres por columna.

F5.5t textarea

```
<textarea rows="5" cols="80">
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum tincidunt, quam nec fringilla rutrum, augue risus
tincidunt augue, vitae posuere dolor libero eget ligula.
Pellentesque eu lobortis massa.
</textarea>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tincidunt, quam nec fringilla rutrum, augue risus tincidunt augue, vitae posuere dolor libero eget ligula. Pellentesque eu lobortis massa.

https://www.w3schools.com/tags/tag_textarea.asp

<select>, <option>

Crea una lista desplegable de selección de elementos. Cada elemento de la lista se define mediante una etiqueta <option>.

F5.5u select

```
Selecciona Marca:
<select name="car">
  <option value="fiat">Fiat</option>
  <option value="ford">Ford</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

Selecciona Marca:

Fiat
Ford
Opel
Audi

Añadiendo el atributo `<select multiple>` estamos indicando que el usuario puede seleccionar varios valores de la lista.

Selecciona Marca:

F5.5v
select múltiple

https://www.w3schools.com/tags/tag_select.asp

`<optgroup>`

Dentro de un select, la etiqueta `<optgroup>` permite agrupar las opciones en grupos.

```
<select>
  <optgroup label="Coches">
    <option value="fiat">Fiat</option>
    <option value="ford">Ford</option>
  </optgroup>
  <optgroup label="Motos">
    <option value="bmw">BMW</option>
    <option value="kawasaki">Kawasaki</option>
  </optgroup>
</select>
```

F5.5w
optgroup

`<fieldset>`, `<legend>`

El elemento `fieldset` permite agrupar elementos de un formulario. La etiqueta `<legend>` añade un texto a modo de título del grupo.

```
<fieldset>
  <legend>Datos personales:</legend>
  Name: <input type="text"><br>
  Email: <input type="text"><br>
  Date of birth: <input type="text">
</fieldset>

<fieldset>
  <legend>Dirección:</legend>
  Calle: <input type="text"><br>
  Ciudad: <input type="text"><br>
</fieldset>
```

F5.5x
fieldset

Datos personales:

Name:

Email:

Date of birth:

Dirección:

Calle:

Ciudad:

<button>

Además del botón de submit, podemos añadir al formulario botones adicionales con sus propias URLs de submit.

F5.5y button

```
<form action="/action_page.php" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <button type="submit">Submit</button><br>
  <button type="submit" formaction="/action_page2.php">Submit to
  another page</button>
</form>
```

First name:

Last name:

https://www.w3schools.com/tags/tag_button.asp

HTML en Drupal

5.6

Ahora sabemos un poco más sobre HTML, pero, ¿cómo lo usamos dentro de Drupal? Pues básicamente existen dos puntos donde se puede introducir HTML en Drupal:

1. En las **plantillas TWIG**. Ya vimos en unidades anteriores que las plantillas contienen HTML y Twig. También estudiamos la forma de añadir las plantillas al tema para poder modificarlas.
2. En los contenidos que acepten texto con **formato de texto**. Al editar un contenido, éste puede tener un campo de tipo **Texto largo con formato** que acepte diferentes formatos de texto. El formato Full HTML (HTML completo) es el más permisivo, al no limitar las etiquetas HTML que pueden ser utilizadas.

Como ejemplo, crea una nueva Página básica. En el cuerpo, selecciona HTML Completo. Si utilizas los botones del editor WYSIWYG, estarás añadiendo HTML, pero sin ver el código HTML que se está generando. Para ver ese código HTML, haz clic en el botón **Fuente HTML [F5.6a]**.

Crear Página básica ☆

[Inicio](#) > [Agregar contenido](#)

Título *

Ejemplo HTML

Cuerpo (Editar resumen)

B *I* ~~S~~ ^{x²} _{x₂} U

Formato Fuente HTML

Lorem ipsum **dolor sit amet**, consectetur adipiscing elit.

- Suspendisse eget nibh suscipit
- Tincidunt magna ut, euismod odio
- Ut in ultricies sapien.

body ul li

Formato de texto **HTML completo** [Acerca de formatos de texto ?](#)

F5.6a

HTML en campos de texto

Edición del contenido usando los botones del editor WYSIWYG.

Ahora podrás ver el código HTML generado y escribir código HTML directamente **[F5.6b]**.

F5.6b**HTML en campos de texto**

Fuente HTML activada.

Título *
Ejemplo HTML

Cuerpo (Editar resumen)

B I S x² x₂ I_x | ☰ ☲ ☱ ☳ ☴ ☵ ☶ ☷ | Formato - | Fuente HTML

```
<p>Lorem ipsum <strong>dolor sit amet</strong>, consectetur adipiscing elit.</p>
<ul>
  <li>Suspendisse eget nibh suscipit</li>
  <li>Tincidunt magna ut, <em>euismod odio</em></li>
  <li>Ut in ultricies sapien.</li>
</ul>
```

Formato de texto HTML completo ▾ [Acerca de formatos de texto ?](#)

Por último, guarda el contenido para ver el resultado **[F5.6c]**:

F5.6c**HTML en campos de texto**

HTML generado.

Lorem ipsum **dolor sit amet**, consectetur adipiscing elit.

- Suspendisse eget nibh suscipit
- Tincidunt magna ut, *euismod odio*
- Ut in ultricies sapien.

Esto último es aplicable a cualquier entidad que tenga campos de tipo **Texto largo con formato**. Por tanto, podremos utilizar HTML en otros elementos de Drupal como Bloques personalizados, Taxonomías, Comentarios, Usuarios, etc.

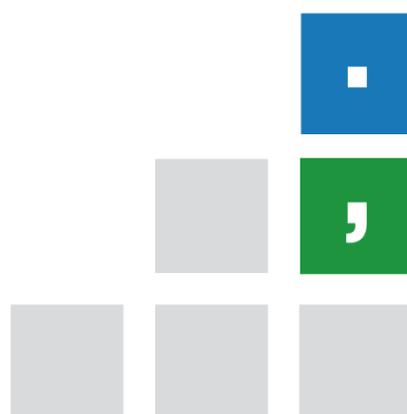
Fran Gil

Front-End Development I

CSS3

6

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

6 CSS3

Para la correcta presentación de las páginas en Internet, complementamos el código HTML con las hojas de estilo en cascada (CSS, Cascading Style Sheets). HTML se utiliza para definir la estructura y contenidos de la página, mientras que CSS se encarga de añadir los estilos que tendrán de los distintos elementos (color, forma, tamaño, posición, etc.).

En esta unidad veremos una amplia introducción a CSS3, que podrás completar con recursos externos y documentación más específica.

Comparativa D8/D7

Lo estudiado en esta unidad es compatible con cualquier versión de Drupal que utilice CSS3.

Contenidos de la Unidad

- 6.1 Introducción a CSS
- 6.2 Sintaxis de CSS
- 6.3 Colores y fondos
- 6.4 Estilos de fuentes
- 6.5 Agrupación y posicionamiento de elementos
- 6.6 Selección avanzada de elementos
- 6.7 Estilos de formularios
- 6.8 Recursos adicionales

6

6.1 Introducción a CSS

Para la correcta presentación de las páginas en Internet, complementamos el código HTML con las hojas de estilo en cascada (CSS, *Cascading Style Sheets*). HTML se utiliza para definir la estructura y contenidos de la página, mientras que CSS se encarga de añadir los estilos que tendrán de los distintos elementos (color, forma, tamaño, posición, etc.).

Para una referencia completa de CSS, puedes consultar este enlace:

- CSS Tutorial (en inglés):
<https://www.w3schools.com/css/default.asp>

Existen varias formas de añadir estilos CSS a una página:

1. **Inline.** Mediante el atributo "style", que se puede aplicar a casi cualquier etiqueta HTML.

```
<p style="font-size:14px;">El tamaño de letra es de 14 pixeles</p>
```

2. **Interno.** Utilizando un bloque de estilos en el encabezado de la página, mediante la etiqueta <style></style>:

```
<html>
<head>
  <title>Página con estilos CSS</title>
  <style type="text/css">
    h1 {font-size: 30px; font-family: arial; color: red;}
    h2 {font-size: 15px; font-family: arial;}
    p {font-size: 8px; font-family: arial;}
  </style>
</head>
<body>
  <h1>Texto encabezado h1</h1>
  <h2>Texto encabezado h2</h2>
</body>
</html>
```

3. **Externo.** Separando los estilos CSS en un archivo **.css**, que enlazaremos desde la página web. Veamos cómo referenciar el archivo **.css** desde un archivo HTML.

Archivo HTML:

```
<html>
<head>
  <title>Página con estilos CSS</title>
  <link rel="stylesheet" type="text/css" href="estilos.css" />
</head>
<body>
  <h1>Texto encabezado h1</h1>
  <h2>Texto encabezado h2</h2>
</body>
</html>
```

Archivo CSS (estilos.css):

```
h1 {
  font-size: 30px;
  font-family: arial;
  color: red;
}
h2 {
  font-size: 15px;
  font-family: arial;
}
p {
  font-size: 8px;
  font-family: arial;
}
```

El método recomendado es el utilizado en el punto 3, enlazando a uno o varios archivos **.css** que contengan todos los estilos del sitio. De esta forma, cuando se realicen modificaciones en la hoja de estilos, sólo habrá que modificar los archivos **.css** para que los cambios se vean reflejados directamente en todas las páginas que hagan uso de estos, ahorrándonos el coste de tener que modificar individualmente cada una de las páginas. Ésta es, además, la manera más limpia de trabajar, porque permite separar el diseño de la estructura y contenidos de la web.

Mismo contenido, distintos estilos

Para hacerte una idea del potencial de las hojas de estilo CSS, visita esta URL:

https://www.w3schools.com/css/css_intro.asp

Aquí encontrarás un ejemplo de contenido HTML al que se le aplican diferentes hojas de estilo. Al hacer clic sobre los enlaces "Stylesheet 1", "Stylesheet 2", etc., verás cómo cambian completamente los estilos (colores, fondos, colocación de elementos, tamaño de letras, etc.), pero manteniendo los mismos contenidos.

Welcome to My Homepage
Use the menu to select different Stylesheets

Stylesheet 1
Stylesheet 2
Stylesheet 3
Stylesheet 4
No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
[No Stylesheet](#).

Side-Bar
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi.

F6.1a

Hojas de estilo

Probando diferentes hojas de estilo sobre un mismo contenido HTML.

F6.1b

Hojas de estilo

Probando diferentes hojas de estilo sobre un mismo contenido HTML.



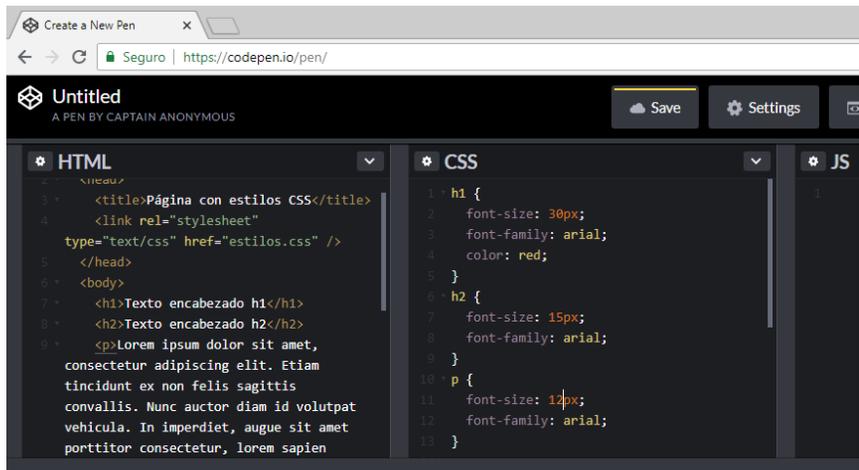
Probando el CSS

En la próxima unidad veremos cómo trabajar con los estilos dentro de Drupal. Para probar el código de esta unidad fuera de Drupal, puedes utilizar herramientas online como: <https://codepen.io/pen/>

F6.1c

Probando el CSS

Herramienta online para probar el funcionamiento de HTML y CSS.



Texto encabezado h1

Texto encabezado h2

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam tincidunt ex non felis sagittis convallis. Nunc auctor diam id volutpat vehicula. In imperdiet, augue sit an ut condimentum leo ex vel elit. Nulla dui magna, interdum ut auctor eu.

Este tipo de herramientas tienen, generalmente, una ventana para introducir HTML, otra para CSS, como si se tratara de un archivo .css externo (método 3 visto anteriormente), e incluso una ventana para añadir Javascript, que también estudiaremos en próximas unidades. En una cuarta ventana veremos la salida final de la página, teniendo en cuenta la combinación del código anterior (HTML + CSS + Javascript).

Sintaxis de CSS

6.2

Conocer CSS implica conocer qué propiedades pueden ser aplicadas a cada elemento HTML y qué valores posibles puede tener cada propiedad. La estructura básica consiste en un selector y un bloque de estilos que se aplicarán a ese selector.

```
selector {
  propiedad: valor;
  propiedad: valor;
  propiedad: valor;
}
```

El selector identifica al elemento HTML sobre el que se aplicarán los estilos asociados (en el formato **propiedad: valor;**)

Un selector puede ser directamente una etiqueta HTML. En el siguiente ejemplo se estarían aplicando las propiedades de color, fuente y tamaño a todas las etiquetas <p> de la página.

```
p {
  color: #00FF00;
  font-family: verdana;
  font-size: 12px;
}
```

Una misma propiedad puede aplicarse a distintas etiquetas HTML. Por ejemplo, la propiedad "color", que define el color que tendrá el elemento puede aplicarse a las etiquetas <h1> a <h6>, <p>, <a>, etc.

```
p {
  color: #00FF00;
}

h1 {
  color: #00FF00;
}
```

En el ejemplo anterior, estamos aplicando el mismo color a las etiquetas <p> y <h1>, así que sería equivalente a agruparlas de la siguiente forma:

```
p, h1 {
  color: #00FF00;
}
```

El selector class

Si definimos los estilos utilizando como selector una etiqueta HTML, como en los ejemplos anteriores, cualquier cambio de estilos en un elemento se aplicará a todos los elementos del mismo tipo. Por ejemplo, todos los párrafos `<p>` tendrán un estilo único, definido en la hoja de estilos.

En la práctica será necesario aplicar unos estilos determinados a los elementos de una sección, pero no a todos los elementos del mismo tipo. En Drupal, por ejemplo, podemos distinguir entre los elementos de la región de contenidos y los elementos que se utilicen en los bloques. Incluso es posible diferenciar entre los estilos de distintos bloques.

Veamos en primer lugar cómo agrupar elementos con el atributo **class**. En el siguiente ejemplo vamos a definir dos tipos de enlaces, *enlace-noticia* y *enlace-generico*. Para referirnos a una clase en la hoja de estilos le antepondremos un punto.

```
a.enlace-noticia {
  color: #000000;
  text-decoration: none;
}

a.enlace-generico {
  color: #FFCD00;
  text-decoration: underline;
}
```

De este modo podremos utilizar varias clases para distinguir unos enlaces de otros. La forma de referenciar a estas clases, en la etiqueta `<a>` de HTML, es mediante el atributo **class**:

```
<p>Enlaces de distintos tipos: </p>
<ul>
  <li><a href="/noticia1" class="enlace-noticia">Enlace a
  noticia</a></li>
  <li><a href="/contacto" class="enlace-generico">Enlace a
  formulario de contacto</a></li>
</ul>
```

Para declarar los estilos a través de una clase, no es necesario indicar el tipo de elemento HTML:

```
.enlace-noticia {
  color: #000000;
  text-decoration: none;
}

.enlace-generico {
  color: #FFCD00;
  text-decoration: underline;
}
```

Ahora la declaración de la clase es más genérica y se podrá aplicar a diferentes elementos HTML (`<p>`, `<a>`, etc.).

Por otra parte, sobre un mismo elemento HTML se pueden aplicar varias clases de estilos, indicándolas dentro del atributo class separadas por espacio:

```
<p class="noticia portada nodo">Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus aliquet non risus vel
fermentum. Cras a blandit lorem, eget ultricies nibh. Cras odio
lorem, finibus in luctus ac, pharetra sed dolor. </p>
```

```
.noticia {
  color: red;
  text-decoration: none;
  font-family: arial;
}

.portada {
  font-size: 20px;
}

.nodo {
  color: green;
}
```

Al párrafo anterior se le aplicarán los estilos definidos en las clases .noticia, .portada y .node, en ese orden. Cuando una propiedad está duplicada (color aparece en .noticia y en .nodo), al aplicarse en orden, la propiedad "ganadora" será la que se indique en último lugar. En el ejemplo anterior estos serían los estilos finales que se aplicarían a ese párrafo (la propiedad **color: red;** queda anulado en favor de **color: green;** que se aplica posteriormente) [F6.2a]:

```
color: red;
text-decoration: none;
font-family: arial;
font-size: 20px;
color: green;
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus aliquet non risus vel fermentum. Cras a blandit lorem, eget ultricies nibh. Cras odio lorem, finibus in luctus ac, pharetra sed dolor.

Esto se conoce sobrescribir estilos, y es la forma natural de trabajar con estilos CSS. El navegador aporta unos estilos por defecto, y nuestro sitio web Drupal también aportará sus estilos. Por tanto, nosotros trabajaremos siempre sobre un conjunto de estilos de base que tendremos que sobrescribir con otros estilos más específicos. Estos estilos se definirán dentro del tema del sitio.

El selector id

En algunas ocasiones tenemos elementos de HTML que consideramos únicos y a los que les queremos aplicar un conjunto de estilos específicos, estilos que no se van a reutilizar para otros elementos. En estos casos utilizamos el atributo **id**, que tiene que ser único para toda la hoja de estilos y para todo el documento HTML.

```
<h1 id="titulo-principal">Titulo</h1>
```

Para referirnos al **id** de un elemento en la hoja de estilos le antepondremos el

F6.2a

Aplicando estilos CSS

Ejemplo de aplicación de estilos con clases.

símbolo #. Por ejemplo, queremos aplicar estilos particulares al título de la página:

```
#titulo-principal {
  color: #FFCD00;
  text-decoration: underline;
}
```

Agrupación de selectores

Todos los selectores vistos (etiquetas HTML, clases e ids) se pueden agrupar para aplicarles estilos comunes.

```
h1 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}

.general {
  text-align: center;
  color: red;
}
```

En este ejemplo se aplican los mismos estilos a los selectores h1, p y .general. Podemos ahorrar código agrupando los selectores en una línea, separándolos por coma de la siguiente forma:

```
h1, p, .general {
  text-align: center;
  color: red;
}
```

Comentarios en CSS

En las hojas de estilo CSS podemos añadir comentarios comenzando con /* y terminando con */. Pueden ser una línea o de varias líneas, y colocarse entre reglas o entre propiedades.

```
p {
  color: red;
  /* Comentario en una línea */
  text-align: center;
}

/* Comentario en varias líneas
  Lorem ipsum dolor sit amet,
  consectetur adipiscing elit. */
```

Ten en cuenta que, al igual que los comentarios HTML, los comentarios en CSS son visibles por cualquier usuario que inspeccione la hoja de estilos desde su navegador.

Colores y fondos

6.3

En este apartado veremos algunas propiedades CSS que permiten configurar el color de los elementos y el fondo. Algunas de las propiedades más utilizadas son:

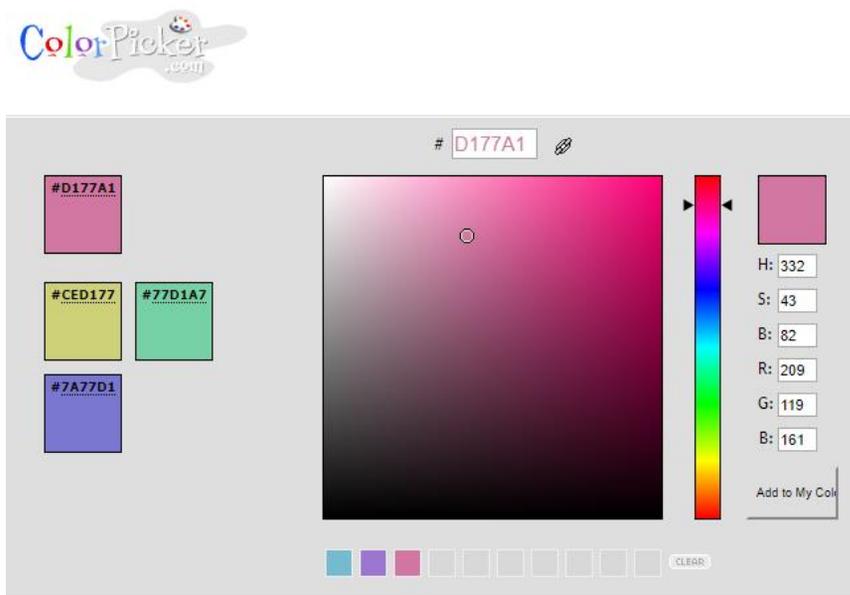
color

Describe el color de la fuente de un elemento. Por ejemplo, si queremos aplicar el color verde al texto contenido en los párrafos <p>, escribiremos en la hoja de estilos:

```
p {
  color: #00FF00;
}
```

La propiedad **color** se puede aplicar a cualquier elemento que pueda contener texto, y sólo afectará al texto.

En HTML los colores se expresan comúnmente en formato hexadecimal, siguiendo la nomenclatura #RRGGBB (red, green, blue). El software de diseño gráfico (del tipo Photoshop o Gimp) ya viene preparado para facilitarnos estos valores, pero también podemos usar alguna de las herramientas online que nos permiten seleccionar un color y nos devuelven su valor hexadecimal listo para usar en nuestras hojas de estilo (Por ejemplo, <http://www.colorpicker.com>). **[F6.3a]**



F6.3a

Selección de colores

En muchos programas de tratamiento de imágenes, como Gimp o Photoshop, podemos ver los códigos RGB en formato hexadecimal de los colores seleccionados. También encontramos servicios online como ColorPicker, que nos permite obtener el código del color seleccionado.

HTML soporta un conjunto de colores que pueden ser referenciados por su nombre (Por ejemplo: **color: Green;**). En este enlace puedes ver el listado completo:

https://www.w3schools.com/colors/colors_names.asp

Aunque el formato más utilizado es el Hexadecimal (HEX), los colores se pueden representar en cualquiera de estos formatos:

- Por **nombre** de color. Simplemente especificaremos un nombre de color válido (consulta la lista anterior).

```
p {  
  color: Tomato;  
}
```

- **RGB**. Se especifica el color en función de la intensidad de cada color (rojo, verde, azul), indicando un valor entre 0 (menos intensidad) y 255 (mayor intensidad). El formato es el siguiente:

rgb(red, green, blue)

```
p {  
  color: rgb(255, 99, 71);  
}
```

- **HEX**. Equivale al formato RGB, pero los valores se expresan en hexadecimal.

```
p {  
  color: #ff6347;  
}
```

- **HSL**. El color se introduce a través de los valores de tono, saturación y brillo (hue, saturation, lightness).

```
p {  
  color: hsl(9, 100%, 64%);  
}
```

- **RGBA**. Similar al formato RGB, pero añadiendo un canal alpha que especifica la opacidad o transparencia.

0 equivale a totalmente transparente, y 1 a totalmente opaco (sin transparencia). Podemos utilizar valores intermedios (0.5 equivale a un 50% de transparencia).

```
p {  
  color: rgba(255, 99, 71, 0.5);  
}
```

- **HSLA**. Similar al formato HSL, pero añadiendo un canal alpha que especifica la opacidad o transparencia.

```
p {  
  color: hsla(9, 100%, 64%, 0.5);  
}
```

background

La propiedad background se divide en las siguientes propiedades:

background-color. Describe el color de fondo de los elementos. Si queremos cambiar el color de fondo de una página, aplicamos la propiedad a la etiqueta `<body>`, pero también podemos aplicar el color de fondo a otros elementos.

```
body {
  background-color: Beige;
}

div {
  background-color: #7A77D1;
}
```

background-image. La propiedad background-image permite insertar una imagen de fondo. La URL será relativa cuando la imagen esté almacenada en el mismo sitio, aunque también se aceptan URLs absolutas para referenciar a imágenes externas.

```
body {
  background-image: url("images/fondo.png");
}
```

background-repeat. Permite indicar si queremos que la imagen de fondo se repita. Los valores posibles son:

```
background-repeat: repeat-x; /* se repite en horizontal */
background-repeat: repeat-y; /* se repite en vertical */
background-repeat: repeat; /* se repite en horizontal y vertical */
background-repeat: no-repeat; /* no se repite */
```

background-attachment. Esta propiedad permite indicar si la imagen está fija o se desplaza con el elemento contenedor.

```
background-attachment: scroll; /* la imagen se desplaza */
background-attachment: fixed; /* la imagen está fija */
```

background-position. Permite definir la posición de la imagen de fondo con respecto al elemento que la contiene. Por defecto la imagen se coloca en la esquina superior izquierda (*top left*). La forma de indicar la posición es siempre mediante las coordenadas *x* e *y*, que se pueden expresar en unidades fijas, como píxeles, o en porcentaje con respecto al elemento que la contiene. También se pueden utilizar las palabras reservadas: *top* (superior), *bottom* (inferior), *center* (centro), *left* (izquierda) y *right* (derecha). Algunos ejemplos de uso son:

- **background-position: 10px 20px;** (la imagen se posiciona a 10 píxeles del margen superior y a 20 píxeles del margen izquierdo)
- **background-position: 50% 25%;** (la imagen se posiciona en el centro vertical de la página y a un 25% del margen izquierdo).
- **background-position: top left;** (la imagen se posiciona en la esquina superior izquierda).

background. Esta propiedad agrupa los valores de las propiedades anteriores en una única propiedad. Por ejemplo, el conjunto de propiedades:

```
body {
  background-color: #FFDD35;
  background-image: url("fondo.png");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: left top;
}
```

Es equivalente a:

```
body {
  background: #FFDD35 url("fondo.png") no-repeat fixed left top;
}
```

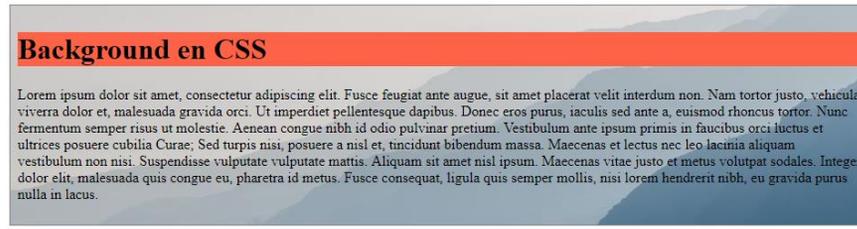
Veamos un ejemplo de HTML al que le aplicamos diferentes estilos de fondo:

```
<div class="mountain">
<h1>Background en CSS</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce feugiat ante augue, sit amet placerat velit interdum non. Nam tortor justo, vehicula viverra dolor et, malesuada gravida orci. Ut imperdiet pellentesque dapibus. Donec eros purus, iaculis sed ante a, euismod rhoncus tortor. Nunc fermentum semper risus ut molestie. Aenean congue nibh id odio pulvinar pretium. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed turpis nisi, posuere a nisl et, tincidunt bibendum massa. Maecenas et lectus nec leo lacinia aliquam vestibulum non nisi. Suspendisse vulputate vulputate mattis. Aliquam sit amet nisl ipsum. Maecenas vitae justo et metus volutpat sodales. Integer dolor elit, malesuada quis congue eu, pharetra id metus. Fusce consequat, ligula quis semper mollis, nisi lorem hendrerit nibh, eu gravida purus nulla in lacus.</p>
</div>
```

Estilo 1: Se muestra una imagen de fondo. Como no se especifica lo contrario, la imagen se repetirá en horizontal y en vertical. Aunque esto no se aprecia con el contenido HTML que hemos definido, si añadimos más texto el efecto sería el mostrado en la Figura [F6.3c].

```
.mountain {
  width: 100%;
  border: 1px solid gray;
  padding: 8px;
  background: #e1efbb url('https://www.w3schools.com/css/mountain.jpg');
}

h1 {
  background: Tomato;
}
```



F6.3b
background
Aplicando colores e imágenes de fondo.



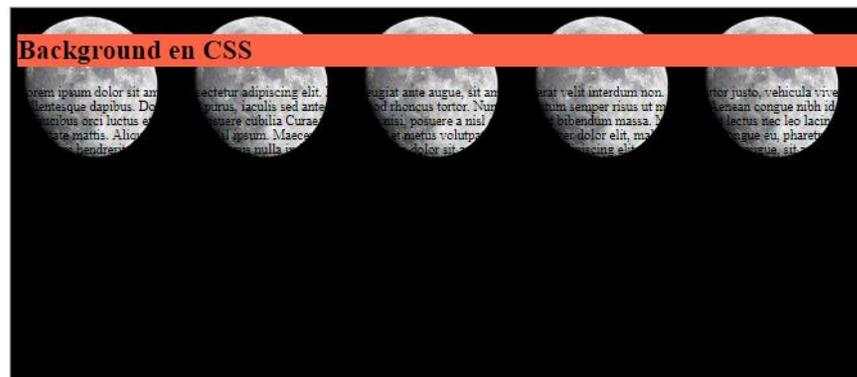
F6.3c
background
Aplicando colores e imágenes de fondo.

Estilo 2: Se añade una imagen que se repite en horizontal, empezando en la esquina superior izquierda. La imagen no se repite en vertical. **[F6.3d]**.

```
.mountain {
  width: 100%;
  border: 1px solid gray;
  padding: 8px;
  background: #000 url('https://interactive-examples.mdn.mozilla.net/media/examples/moon.jpg') repeat-x fixed left top;
}

h1 {
  background: Tomato;
}
```

<https://interactive-examples.mdn.mozilla.net/media/examples/moon.jpg>



F6.3d
background
Aplicando colores e imágenes de fondo.

opacity

La propiedad `opacity` permite especificar un valor de opacidad/transparencia de un elemento.

Aunque se utiliza generalmente con imágenes o colores de fondo, su aplicación afecta a cualquier tipo de elemento. Ten en cuenta que si se aplica transparencia a un elemento contenedor, tipo `div`, la transparencia se aplicará a todos los elementos que contenga (salvo que se especifiquen estilos específicos para esos elementos). En el siguiente ejemplo vemos este efecto, en el que la transparencia afecta al color de fondo pero también al texto (<p>) incluido en el contenedor donde se aplica la transparencia **[F6.3e]**.

```
<div class="style1"><p>100% opacity</p></div>
<div class="style2"><p>60% opacity</p></div>
<div class="style3"><p>30% opacity</p></div>
<div class="style4"><p>10% opacity</p></div>
```

```
div {
  padding: 50px;
  color: black;
  width: 100px;
  float: left;
}

.style1 {
  background: rgb(71, 55, 206);
  opacity: 1;
}

.style2 {
  background: rgb(71, 55, 206);
  opacity: 0.6;
}

.style3 {
  background: rgb(71, 55, 206);
  opacity: 0.3;
}

.style4 {
  background: rgb(71, 55, 206);
  opacity: 0.1;
}
```

F6.3e

Opacity

Aplicando transparencia a elementos HTML.



Para que la transparencia afecte únicamente al color de fondo (y no al texto), podemos recurrir al formato de color **rgba**, que permite añadir el grado de opacidad directamente en la definición del color **[F6.3f]**.

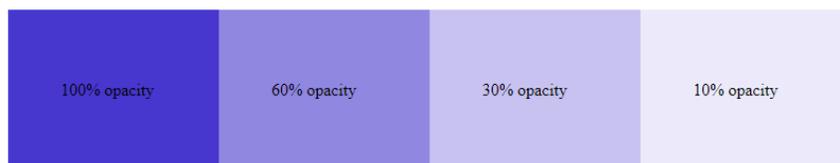
```
div {
  padding:50px;
  color:black;
  width: 100px;
  float: left;
}

.style1 {
  background:rgb(71, 55, 206);
}

.style2 {
  background:rgba(71, 55, 206, 0.6);
}

.style3 {
  background:rgba(71, 55, 206, 0.3);
}

.style4 {
  background:rgba(71, 55, 206, 0.1);
}
```

**F6.3f****Opacity**

Aplicando transparencia a elementos HTML.

https://www.w3schools.com/css/css_image_transparency.asp

6.4 Estilos de fuentes

En este apartado vamos a aplicar propiedades CSS relacionadas con los textos.

Estilos de fuentes

font-family. Con esta propiedad especificamos una lista ordenada de fuentes que se aplicarán al texto del elemento. Se especifica una lista porque es posible que las fuentes indicadas no existan en el ordenador del usuario que está visualizando la página. Esto es importante tenerlo en cuenta a la hora de usar fuentes "especiales". Siempre será mejor usar las fuentes más comunes como Arial, Verdana, Helvetica, etc.

```
h1 {
  font-family: arial, verdana, sans-serif;
}
h2 {
  font-family: "Times New Roman", serif;
}
```

Cuando el nombre de fuente contiene espacios, se indica entre comillas, como es el caso de "Times New Roman".

Es una buena práctica terminar la lista de las fuentes con una fuente básica, para asegurarnos de que el navegador tendrá un comportamiento predecible si no encuentra instalada ninguna de las fuentes anteriores (por ejemplo, terminar la lista con *serif*, *sans-serif*, *monospace*).

font-style. Permite definir estilos para el texto, aceptando los valores: *normal* (sin estilo), *italic* (cursiva) y *oblique* (inclinada, dependiendo de la fuente puede ser similar a *italic*).

font-weight. Define la intensidad o peso de la fuente, aceptando los valores *normal* y *bold*. El valor *bold* muestra el texto en negrita.

font-size. Establece el tamaño de la fuente. El tamaño se puede especificar de muchas formas. Algunas de las más comunes son:

- De forma absoluta, en píxeles (px) o puntos (pt):

```
h1 {
  font-size: 30px;
}
h1 {
  font-size: 12pt;
}
```

- De forma relativa, en porcentaje (%) o en relación al valor de referencia del navegador del usuario (em):

```
h1 {
  font-size: 110%;
}
```

```

}
h1 {
  font-size: 2em;
}

```

Por accesibilidad se recomienda el uso de medidas relativas, porque así el usuario podrá modificar el tamaño de la letra en su navegador y la página web se adaptará a los nuevos tamaños de fuente. En este caso el usuario selecciona el valor de la fuente base o de referencia en su navegador. En el ejemplo anterior, el elemento h1 tiene un tamaño de 2em, esto es, el doble del tamaño especificado por el navegador del usuario.

font. Por último, la propiedad font combina algunas de las propiedades anteriores, siguiendo este orden de elementos (font-style | font-weight | font-size | font-family):

```

p {
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
  font-family: arial, sans-serif;
}

```

Es equivalente a:

```

p {
  font: italic bold 30px arial, sans-serif;
}

```

https://www.w3schools.com/css/css_font.asp

Estilos de texto

Estas propiedades se utilizan para definir los estilos del texto.

color. Ya vimos en el apartado anterior cómo aplicar esta etiqueta para definir el color del texto.

text-align. Permite definir la alineación horizontal del texto. Los valores posibles de esta propiedad son:

- left (texto alineado a la izquierda)
- right (texto alineado a la derecha)
- center (texto con alineación centrada)
- justify (texto con justificación completa)

```

td {
  text-align: left;
}

p {
  text-align: justify;
}

```

text-decoration. Permite añadir algunos efectos al texto, aceptando los siguientes valores:

- **none** (es el valor por defecto, sin ningún efecto especial)
- **underline** (subrayado)
- **overline** (línea por encima del texto)
- **line-through** (tachado)

```
p {  
  text-decoration: underline;  
}
```

text-transform. Permite realizar algunas transformaciones del texto, aceptando los siguientes valores:

- **capitalize** (pone en mayúscula la primera letra de cada palabra)
- **uppercase** (convierte todas las letras a mayúscula)
- **lowercase** (convierte todas las letras a minúscula)

```
h1 {  
  text-transform: uppercase;  
}
```

letter-spacing. Define el espaciado entre caracteres.

```
p {  
  letter-spacing: 3px;  
}
```

line-height. Define el espacio entre líneas de texto.

```
p {  
  line-height: 1.2;  
}
```

word-spacing. Define el espacio entre palabras de un texto.

```
h1 {  
  word-spacing: 10px;  
}  
  
p {  
  word-spacing: -5px;  
}
```

https://www.w3schools.com/css/css_text.asp

Estilos de enlaces

Para los enlaces (etiqueta <a>) se pueden aplicar muchas de las propiedades vistas hasta ahora. Sin embargo, los enlaces en HTML guardan el estado del mismo (por ejemplo, si ha sido visitado o si el cursor está sobre el enlace) y es posible aplicar propiedades específicas para cada estado. Los posibles estados del enlace, también conocidos como pseudoclasas son:

- **a** (estilos generales para todos los enlaces, independientemente del estado)
- **a:link** (enlace no visitado)
- **a:visited** (enlace visitado)
- **a:hover** (se activa cuando se pasa el puntero del ratón por encima del enlace)
- **a:active** (enlace activo, a la página que se está visitando)

La combinación adecuada dependerá de los estilos que queramos aplicar a nuestro sitio web. Veamos algunos ejemplos:

```
a {
  color: #000000;
  text-decoration: none;
}
a:link {
  color: #FF0000;
}
a:visited {
  color: #00FF00;
}
a:hover {
  color: orange;
  font-style: italic;
  font-weight: bold;
  text-decoration: underline;
}
a:active {
  background-color: #FFFF00;
}
```

Cuando se establecen estilos para varios estados de enlaces, hay que tener en cuenta el siguiente orden:

- a:hover debe definirse después de a:link y a:visited
- a:active debe definirse después de a:hover

https://www.w3schools.com/css/css_link.asp

La etiqueta **** es un elemento neutro que no añade nada al documento por sí solo, pero que es muy útil para aplicar estilos a determinados fragmentos de texto. Por ejemplo, podríamos utilizar este elemento para destacar en rojo determinados fragmentos de texto.

HTML:

```
<h1>Ejemplo de utilización de span</h1>
<p>Lorem ipsum <span class="destacado">Drupal</span> sit amet,
consectetur adipiscing elit. Praesent ut ornare nulla. Phasellus
vitae lacus rutrum erat bibendum venenatis. Mauris faucibus ex ut
nunc tristique suscipit eu quis nulla. Integer eleifend sit amet
ante sed auctor. Aenean <span class="destacado">Drupal</span>
tortor tellus, non tristique sapien facilisis non. Quisque
pellentesque metus tortor, nec viverra tellus euismod id. Donec
varius quis erat sed eleifend. Sed elit libero, tincidunt at
tincidunt sed, ornare non sapien. Aliquam vitae nulla ultrices,
vehicula neque sit amet, efficitur <span
class="destacado">Drupal</span>. </p>
```

CSS:

```
span.destacado {
  color: White;
  background-color: DodgerBlue;
}
```

F6.4a

Etiqueta span

Con span podemos aplicar estilos a fragmentos de texto.

Ejemplo de utilización de span

Lorem ipsum **Drupal** sit amet, consectetur adipiscing elit. Praesent ut ornare nulla. Phasellus vitae lacus rutrum erat bibendum venenatis. Mauris faucibus ex ut nunc tristique suscipit eu quis nulla. Integer eleifend sit amet ante sed auctor. Aenean **Drupal** tortor tellus, non tristique sapien facilisis non. Quisque pellentesque metus tortor, nec viverra tellus euismod id. Donec varius quis erat sed eleifend. Sed elit libero, tincidunt at tincidunt sed, ornare non sapien. Aliquam vitae nulla ultrices, vehicula neque sit amet, efficitur **Drupal**.

Agrupación y posicionamiento de elementos

6.5

Contenedores <div>

En la Unidad 5 (apartado 5.3) estudiamos los nuevos elementos semánticos que permiten crear regiones o secciones dentro de una página web (<header>, <section>, <nav>, <main>, <article>, <aside>, etc.). La etiqueta **<div>** se sigue utilizando para definir contenedores que no requieren un significado propio. Aunque en este apartado hablaremos de <div> como contenedor genérico, la agrupación de elementos se puede realizar con cualquiera de los elementos semánticos anteriores.

Los elementos <div> permiten crear regiones o bloques con distintos estilos **[F6.5a]**.

HTML:

```
<h1>Ejemplo de utilización de div</h1>
<div id="lista1">
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ul>
</div>
<div id="lista2">
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ul>
</div>
```

CSS:

```
#lista1 {
  background: #0000FF;
  color: #FFFFFF;
}
#lista2 {
  background: #FF0000;
  color: #FFFFFF;
}
```

Ejemplo de utilización de div

- Elemento 1
- Elemento 2
- Elemento 3

- Elemento 1
- Elemento 2
- Elemento 3

F6.5a Etiqueta div

Contenedor div para agrupar elementos.

Márgenes y separación entre elementos

Para separar los distintos elementos, disponemos en CSS de las siguientes propiedades **[F6.5b]**:

- **margin.** Define el espacio con respecto al elemento contenedor (elemento previo). Es el espacio entre el elemento previo y el borde del elemento actual.
- **border.** Define el borde o límite del elemento.
- **padding.** Define el espacio dentro del mismo elemento desde el borde del mismo al contenido. Es conocido también como relleno.

Todo elemento tiene cuatro lados: superior (top), derecho (right), inferior (bottom) e izquierdo (left), y siempre se referencia en ese orden.

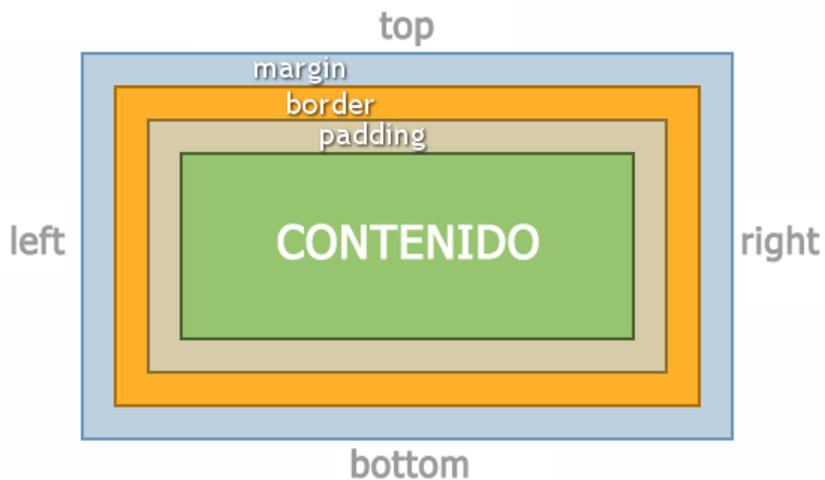
Cuando aplicamos un determinado margen a un elemento lo estamos haciendo con respecto a la página o con respecto a un elemento contiguo. Por ejemplo, si aplicamos un margen superior de 10px a una tabla que tiene justo encima una imagen, lo que estamos haciendo es separar la tabla un espacio de 10px con respecto a la imagen.

F6.5b Etiqueta div Diferencias entre padding y margin

En esta figura podemos ver a qué zonas afectan cada una de las propiedades relacionadas con padding y margin.

El padding sirve para indicar un espacio entre el comienzo del elemento y su contenido real (o borde).

El margin sirve para especificar espacios entre elementos, o de un elemento con respecto al que lo contiene.



Veamos un ejemplo de utilización de márgenes (margin) y relleno (padding), aplicado a un párrafo de texto.

HTML:

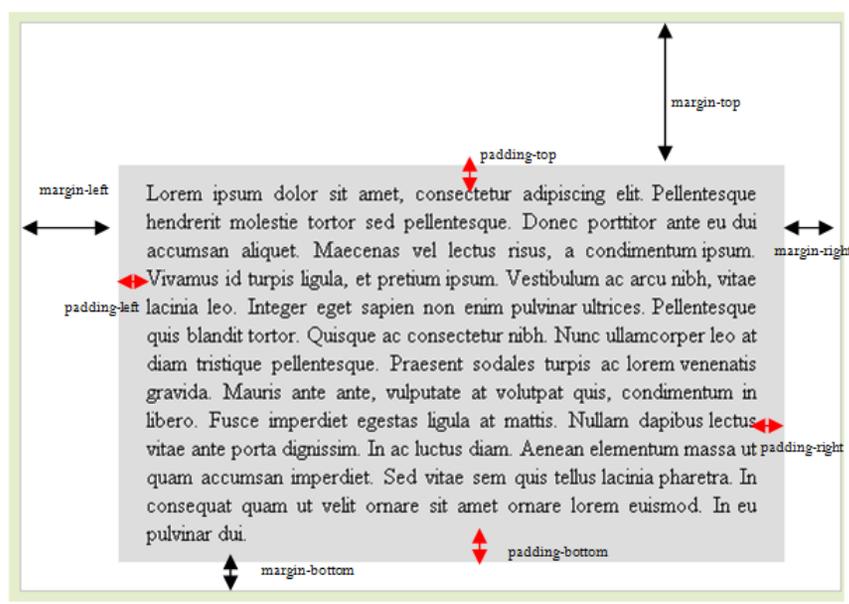
```
<p id="parrafo-principal">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent
  ut ornare nulla. Phasellus vitae lacus rutrum erat bibendum
  venenatis. Mauris faucibus ex ut nunc tristique suscipit eu quis
  nulla. Integer eleifend sit amet ante sed auctor...</p>
```

CSS:

```
body {
  margin-top: 100px;
  margin-right: 40px;
  margin-bottom: 10px;
  margin-left: 70px;
}

#parrafo-principal {
  text-align: justify;
  background-color: #DDDDDD;
  padding-top: 10px;
  padding-right: 20px;
  padding-bottom: 10px;
  padding-left: 20px;
}
```

En la Figura **[F6.5c]** se presenta un párrafo con fondo gris. Los padding son espacios dentro del propio <p>, entre el borde del elemento y el texto contenido. Los márgenes son los espacios entre el elemento superior (body) que contiene al elemento <p>.



F6.5c

Uso de padding y margin

Ejemplo de utilización de padding y margin.

Las propiedades margin y padding se puede reducir a una única línea:

CSS:

```
body {
  margin: 100px 40px 10px 70px;
}

#parrafo-principal {
  text-align: justify;
  background-color: #DDDDDD;
  padding: 10px 20px 10px 20px;
}
```

Para acordarte del orden de los parámetros, empieza siempre en top y continúa en el sentido de las agujas del reloj: top, right, bottom y left.

En nuestro ejemplo podemos simplificar aún más la definición del padding:

CSS:

```
#parrafo-principal {
  text-align: justify;
  background-color: #DDDDDD;
  padding: 10px 20px;
}
```

Cuando la propiedad sólo tiene dos valores, equivale a:

- top y bottom tienen el mismo valor de padding (10px)
- left y right tienen el mismo valor de padding (20px).

Si a la propiedad margin o padding le asignamos un único valor, estaremos estableciendo el mismo valor para todos los lados del elemento.

Enlaces relacionados:

https://www.w3schools.com/css/css_padding.asp

https://www.w3schools.com/css/css_margin.asp

Bordes

Estas propiedades se utilizan para definir estilos relacionados con los bordes de los elementos:

border-width. Define el ancho del borde. Se puede utilizar un valor en píxeles o los valores *thin*, *medium* y *thick*. También se pueden establecer valores de borde independientes para cada lado del elemento, siguiendo el orden top, right, bottom, left.

border-color. Color del borde del elemento. El color también se puede establecer diferente para cada lado del elemento.

border-style. Define el estilo del borde, pudiéndose aplicar los valores: *none* (ningún estilo), *dotted*, *dashed*, *solid*, *double*, *groove*, *ridge*, *inset*, *outset*. El estilo también se puede establecer diferente para cada lado del elemento.

border. Permite agrupar las propiedades de borde en una sola propiedad.

```
p {
  border-width: 1px;
  border-style: solid;
  border-color: blue;
}
```

Es equivalente a:

```
p {
  border: 1px solid blue;
}
```

Veamos algunos ejemplos [F6.5d]:

HTML:

```
<p class="uno">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum
  dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit
  amet, consectetur adipiscing elit.</p>
<p class="dos">
  Quisque laoreet, ligula sed dictum dapibus, augue lectus accumsan
  magna, ac imperdiet massa nisi vitae purus. Quisque laoreet, ligula
  sed dictum dapibus, augue lectus accumsan magna, ac imperdiet.</p>
<p class="tres">
  Donec scelerisque urna lorem, efficitur blandit mauris tristique a.
  Aliquam blandit ipsum ac elementum consectetur. Quisque auctor urna
  ex, quis tincidunt arcu dictum ac. </p>
```

CSS:

```
p.uno {
  border: 2px dotted blue;
}
p.dos {
  border: thick solid red;
}
p.tres {
  border-style: dotted dashed solid double;
}
```

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Quisque laoreet, ligula sed dictum dapibus, augue lectus accumsan magna, ac imperdiet massa nisi vitae purus. Quisque laoreet, ligula sed dictum dapibus, augue lectus accumsan magna, ac imperdiet massa nisi vitae purus.

Donec scelerisque urna lorem, efficitur blandit mauris tristique a. Aliquam blandit ipsum ac elementum consectetur. Quisque auctor urna ex, quis tincidunt arcu dictum ac. Nulla accumsan varius placerat.

F6.5d Bordes

Diferentes estilos de borde.

border-radius. Esta propiedad permite añadir esquinas redondeadas a los bordes de un elemento. El valor es el número de píxeles que se usará para redondear la esquina [F6.5e].

CSS:

```
p.uno {
  border: 2px solid red;
  border-radius: 10px;
}
p.dos {
  border: 2px solid blue;
  border-radius: 5px;
}
```

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Quisque laoreet, ligula sed dictum dapibus, augue lectus accumsan magna, ac imperdiet massa nisi vitae purus.

F6.5e Bordes redondeados

Bordes redondeados con la propiedad border-radius.

https://www.w3schools.com/css/css_border.asp

Alto y ancho

Las propiedades **width** y **height** permiten definir el ancho y alto, respectivamente, de los elementos. Los elementos sobre los que especifiquemos estas propiedades tendrán un ancho y alto fijo, establecido en px o en porcentaje **[F6.5f]**.

HTML:

```
<div id="caja1">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Pellentesque hendrerit molestie tortor sed pellentesque. Donec
  porttitor ante eu dui accumsan aliquet. Maecenas vel lectus risus,
  a condimentum ipsum. Vivamus id turpis ligula, et pretium ipsum.
  Vestibulum ac arcu nibh, vitae lacinia leo.</p>
</div>
<div id="caja2">
  <p>Integer tempus mauris ut neque maximus posuere pharetra non
  mauris. Nunc id magna est. Pellentesque quis dictum risus. Nullam
  mollis mollis elit, a sodales neque dapibus ut. Maecenas ac
  ullamcorper dui. </p>
</div>
```

CSS:

```
#caja1 {
  text-align: justify;
  width: 500px;
  height: 100px;
  padding: 10px;
  background-color: #DDD;
}
#caja2 {
  text-align: justify;
  width: 250px;
  height: 150px;
  padding: 10px;
  background-color: #BBB;
}
```

F6.5f Propiedades width y height

Estas propiedades establecen el ancho y alto de los elementos, respectivamente.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque hendrerit molestie tortor sed pellentesque. Donec porttitor ante eu dui accumsan aliquet. Maecenas vel lectus risus, a condimentum ipsum. Vivamus id turpis ligula, et pretium ipsum. Vestibulum ac arcu nibh, vitae lacinia leo.

Integer tempus mauris ut neque maximus posuere pharetra non mauris. Nunc id magna est. Pellentesque quis dictum risus. Nullam mollis mollis elit, a sodales neque dapibus ut. Maecenas ac ullamcorper dui.

Una variación de estas propiedades son **max-width**, **max-height**, **min-width** y **min-height**. Con ellas podemos establecer los tamaños máximos y mínimos. Si establecemos un ancho máximo, el elemento tomará el ancho que necesite, pero sin superar el máximo indicado. Si especificamos un mínimo, el elemento tomará como mínimo el valor indicado, pero además podrá utilizar el tamaño que necesite a partir de ese valor para mostrarse entero **[F6.5g]**.

CSS:

```
#cajal {
  text-align: justify;
  width: 100%;
  min-height: 150px;
  padding: 10px;
  background-color: #DDD;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque hendrerit molestie tortor sed pellentesque. Donec porttitor ante eu dui accumsan aliquet. Maecenas vel lectus risus, a condimentum ipsum. Vivamus id turpis ligula, et pretium ipsum. Vestibulum ac arcu nibh, vitae lacinia leo.

F6.5g Propiedades width y height

Variación de las propiedades de ancho y alto.

https://www.w3schools.com/css/css_dimension.asp

Posicionamiento de elementos

float. Los elementos (junto con su contenido) se pueden hacer flotar a la derecha o a la izquierda usando la propiedad **float**. La propiedad float puede tener los valores: left (izquierda), right (derecha) o none (ninguna). Vemos un ejemplo de aplicar propiedad **float** a un elemento imagen, posicionándolo a la izquierda del texto [F6.5h].

HTML:

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis...</p>
```

CSS:

```
.imagen {
  float: left;
  padding: 5px;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus.

F6.5h float

Imagen flotante a la izquierda del texto.

Sin necesidad de cambiar el orden de los elementos a nivel de HTML, podemos hacer flotar la imagen a la derecha [F6.5i]:

CSS:

```
.imagen {
  float: right;
  padding: 5px;
}
```

F6.5i

float

Imagen flotante a la derecha del texto.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus.



En el siguiente ejemplo se ha aplicado esta propiedad para crear 3 columnas de texto utilizando contenedores <div> **[F6.5j]**.

HTML:

```
<p>Lorem ipsum dolor sit amet,... </p>
<div id="column1">
  <p>Praesent consequat aliquam ligula. Vivamus risus erat... </p>
</div>
<div id="column2">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing... </p>
</div>
<div id="column3">
  <p>Proin vitae urna ut tortor pretium placerat...</p>
</div>
```

CSS:

```
#column1 {
  float:left;
  width: 60%;
  background-color: #DDD;
}
#column2 {
  float:left;
  width: 20%;
  background-color: #EEE;
}
#column3 {
  float:left;
  width: 20%;
  background-color: #CCC;
}
```

F6.5j

Propiedades float y width

Combinación de float y width para crear columnas.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus.

Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. Praesent consequat aliquam ligula. Vivamus risus erat, iaculis iaculis ornare maximus, luctus ac magna. Nulla blandit arcu eget neque ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis lisis condimentum nulla. Praesent consequat aliquam ligula.

Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare am. Aliquam mattis nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus.

https://www.w3schools.com/css/css_float.asp

clear. La propiedad **clear** se usa para controlar cómo se comportan los elementos que siguen a los elementos flotantes de un documento. Esta propiedad se utiliza conjuntamente con **float**, cuando queremos que elementos que tienen cerca un elemento flotante (mediante "float") dejen de alinearse junto a él. Sus posibles valores son *left* (el elemento no permite flotantes a su izquierda), *right* (el elemento no permite flotantes a su derecha), *both* (el elemento no permite flotantes a ningún lado) o *none* (el elemento permite flotantes a ambos lados).

En el siguiente ejemplo volvemos a la imagen flotante sobre un párrafo de texto. En este caso bloquearemos el párrafo con **clear** para no admitir elementos flotantes ni a izquierda ni a derecha **[F6.5k]**.

HTML:

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis...</p>
```

CSS:

```
.imagen {
  float: left;
}
p {
  clear: both;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat.

F6.5k float y clear

Clear nos permite bloquear elementos flotantes a los lados de otro elemento.

https://www.w3schools.com/cssref/pr_class_clear.asp

position. La propiedad **position** especifica el tipo de posicionamiento que utilizará el elemento: static, relative, fixed, absolute y sticky.

- **static.** Los elementos HTML están posicionados como static por defecto. Con static no estamos especificando ninguna posición en concreto, de forma que el elemento se colocará atendiendo al flujo normal de la página.

En el siguiente ejemplo, el div irá colocado entre los dos párrafos **[F6.5l]**.

HTML:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
urna ut tortor pretium placerat...</p>
<div class="box1">
Este div está definido como static
```

```
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
urna ut tortor pretium placerat. Vivamus dui elit, ultricies
ornare lorem vitae, </p>
```

CSS:

```
div.box1 {
  position: static;
  border: 2px solid Red;
}
```

F6.5I position: static

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Este div está definido como static

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae,

- **relative.** La posición relativa se establece a partir de la posición normal del elemento (static). Se pueden las propiedades top, right, bottom y left para desplazar el elemento con respecto a esa posición. Ten en cuenta que esto puede hacer que el elemento se solape con otros elementos, ya que el desplazamiento no provoca el desplazamiento del resto de elementos **[F6.5m]**.

CSS:

```
div.box1 {
  position: relative;
  left: 30px;
  top: 20px;
  border: 2px solid Red;
}
```

F6.5m position: relative

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Este div está definido como relative

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae,

- **fixed.** Con la posición fixed el elemento se fija con respecto al "viewport", que es la parte visible de la página. Incluso aunque hagamos scroll en la página, el elemento se quedará siempre en la misma posición.

En el siguiente ejemplo se ha colocado el div en la esquina inferior derecha, poniendo a 0 los valores de bottom y right **[F6.5n]**.

CSS:

```
div.box1 {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
```

```
border: 2px solid Red;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

F6.5n
position: fixed

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae,

Este div está definido como fixed

- **absolute.** Posición **absolute** es parecido a **fixed** pero la posición se establece tomando como referencia el elemento superior más cercano con una posición establecida (cualquier posición que no sea static).

Como ejemplo hemos definido un div con posición absolute dentro de un div con posición relativa [F6.5o].

HTML:

```
<div class="relative">Este div está definido como: relative
  <div class="absolute">Este div está definido como:
absolute</div>
</div>
```

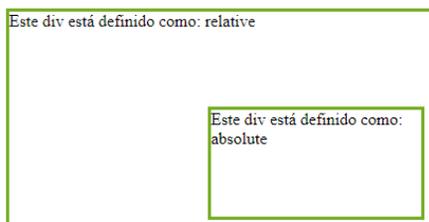
CSS:

```
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  bottom: 5px;
  right: 5px;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

F6.5o
position: absolute



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae,

- **sticky**. Esta posición se basa en la posición del scroll. Es muy típico usar una barra de menú o encabezado de una tabla con posición sticky en la parte superior de la página, para que no desaparezca al hacer scroll **[F6.5p]** **[F6.5q]**.

HTML:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat...</p>
<div class="box1">
Este div está definido como sticky
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, </p>
```

CSS:

```
div.box1 {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
```

F6.5p**position: sticky**

Posición inicial del elemento.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, facilisis condimentum nulla. nibh ut interdum vehicula. Aliquam convallis sed tellus et dapibus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Este div está definido como sticky

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem

F6.5q**position: sticky**

Posición del elemento fijada en la parte superior tras hacer scroll.

Este div está definido como sticky

adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum

https://www.w3schools.com/css/css_positioning.asp

z-index. Los elementos posicionados (cualquier posición excepto static) pueden solaparse, de forma que unos quedarán por delante de otros. Si no se especifica un orden, el último elemento que aparezca en el HTML, será el que se muestre por encima del resto. La propiedad **z-index** permite ordenar los elementos cuando estos se solapan. El valor de z-index puede ser un entero negativo o positivo, de forma que el elemento con mayor z-index se colocará delante.

Como ejemplo vamos a colocar una imagen con posición absoluta, de forma que se quedará por delante del texto **[F6.5r]**.

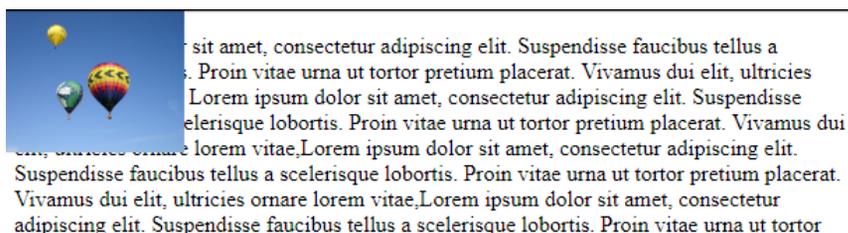
HTML:

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
urna ut tortor pretium placerat. Vivamus dui elit, ultricies
ornare lorem vitae, </p>
```

CSS:

```
img {
  position: absolute;
  left: 0px;
  top: 0px;
}
```

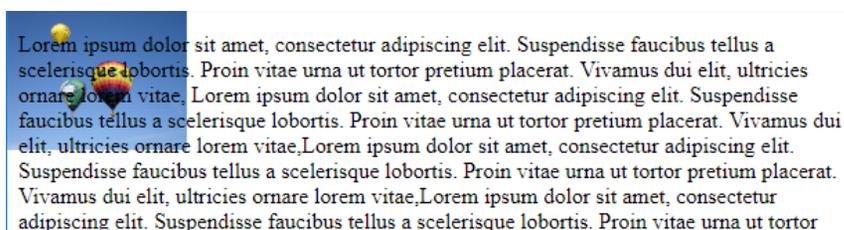


F6.5r
z-index
Orden de elementos
cuando se superponen.

Aplicando un valor negativo a z-index, estaremos enviando la imagen hacia atrás, por lo que el texto quedará por delante **[F6.5s]**.

CSS:

```
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
```



F6.5s
z-index
Orden de elementos
cuando se superponen.

overflow. La propiedad overflow controla qué ocurre cuando el contenido es muy grande para encajar dentro de un área determinada. Funciona con bloques donde se haya establecido un alto (height) específico. Los valores posibles son:

- **visible.** El contenido sobrante se muestra, aunque quede fuera del área. Éste es el valor por defecto.
- **hidden.** El contenido sobrante se oculta. Sólo se mostrará el contenido que quepa dentro del área.
- **scroll.** El contenido sobrante se oculta, pero se añade una barra de scroll para poder acceder a ese contenido.
- **auto.** Es similar a scroll, pero el scroll sólo aparece cuando el contenido se sobresale del área.

Como ejemplo mostraremos un texto dentro de un contenedor div limitado en tamaño, de forma que el texto se sobresalga.

HTML:

```
<div>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
  urna ut tortor pretium placerat...</p>
</div>
```

Como el valor por defecto de overflow es visible, en este primer caso el texto sobrante se muestra fuera de la caja (en gris).

CSS:

```
div {
  width: 600px;
  height: 100px;
  background-color: DarkGray;
}
```

F6.5t z-index

Orden de elementos cuando se superponen.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet,

Añadiendo un overflow hidden, el texto sobrante se cortará y no se mostrará [F6.5u].

CSS:

```
div {
  width: 600px;
  height: 100px;
  background-color: DarkGray;
  overflow: hidden;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit.

F6.5u z-index

Orden de elementos cuando se superponen.

Añadiendo un overflow scroll, el texto sobrante se cortará según el tamaño del contenedor, pero se mostrará un scroll para poder ver el resto de texto **[F6.5v]**.

CSS:

```
div {
  width: 600px;
  height: 100px;
  background-color: DarkGray;
  overflow: scroll;
}
```

scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, ultricies ornare lorem vitae, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium

F6.5v z-index

Orden de elementos cuando se superponen.

En este caso, como el texto se sobresale del contenedor, si establecemos overflow a auto, se mostrará un scroll, igual que si utilizáramos directamente el valor scroll. Sin embargo, si el texto no sobresale, no se mostrará el scroll **[F6.5w]**.

CSS:

```
div {
  width: 600px;
  height: 100px;
  background-color: DarkGray;
  overflow: auto;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis.

F6.5w z-index

Orden de elementos cuando se superponen.

https://www.w3schools.com/css/css_overflow.asp

display. La propiedad display determina si un elemento se mostrará y cómo (block o inline). El valor de display por defecto para la mayoría de los elementos es block o inline.

- **block.** En modo de presentación block, el elemento comienza en una nueva línea y utiliza todo el ancho disponible. El siguiente elemento comenzaría en la siguiente línea. Algunos elementos de tipo block son: `<div>`, `<h1>`..`<h6>`, `<p>`, `<form>`, `<header>`, `<footer>`, `<section>`
- **inline.** Los elementos inline se presentan en línea con los elementos anteriores y posteriores, utilizando únicamente el ancho que necesitan. Algunos elementos de tipo inline son: ``, `<a>`, ``

Veamos con un ejemplo la diferencia entre block e inline:

HTML:

```
<div>El div es un elemento de tipo block.</div>
<p>El párrafo es un elemento de tipo block, <span> pero span es de
tipo inline</span>, así que se muestra dentro del texto, sin salto
de línea.</p>
```

El CSS añadido no es necesario, ya que los valores de display son los que tienen los elementos por defecto. Los estilos definidos se utilizan únicamente para destacar los elementos y diferenciarlos del resto.

CSS:

```
div {
  background-color: DarkGray;
  height: 30px;
}

span {
  background-color: Red;
  color: White;
}
```

El div es un elemento de tipo block.

El párrafo es un elemento de tipo block, pero span es de tipo inline, así que se muestra dentro del texto, sin salto de línea.

F6.5x

display

Elementos de tipo block e inline.

Es posible cambiar el comportamiento por defecto de un elemento mediante la propiedad display, asignando el valor block o inline según corresponda.

Como ejemplo, vamos a representar un listado de elementos (``) en horizontal (inline), en lugar de la representación por defecto que es vertical, al tratarse de elementos de tipo block. Esta representación en horizontal se utiliza habitualmente en los menús de la página.

HTML:

```
<ul>
  <li><a href="/">Inicio</a></li>
  <li><a href="/nosotros">Sobre Nosotros</a></li>
  <li><a href="/servicios">Nuestros servicios</a></li>
</ul>
```

Sin añadir ningún estilo específico, la lista se mostrará en vertical.

- [Inicio](#)
- [Sobre Nosotros](#)
- [Nuestros servicios](#)

F6.5y
display

Lista de elementos con su valor de display por defecto (block).

Pero, si cambiamos el display de los elementos a inline, el resultado será un listado de enlaces en horizontal:

CSS:

```
li {
  display: inline;
}
```

[Inicio](#) [Sobre Nosotros](#) [Nuestros servicios](#)

F6.5z
display

Lista de elementos en modo inline.

https://www.w3schools.com/css/css_display_visibility.asp

Visibilidad de elementos

Existen dos formas de ocultar un elemento desde CSS. Es importante tener en cuenta que el elemento no se muestra en la página, pero sí sigue estando disponible y visible si se consulta el código HTML de la página. Por tanto, no debe utilizarse para ocultar información sensible a la que el usuario no debería poder acceder.

display: none;

Este método oculta el elemento y la página se renderiza ignorándolo por completo, como si no existiera.

visibility: hidden;

Este método oculta el elemento, pero el espacio ocupado por éste queda reservado, aunque se muestre en blanco.

Veamos un ejemplo aplicando ambas propiedades. En la primera Figura **[F6.5aa]** se muestra el HTML sin ocultar ninguno de los elementos (una imagen a la izquierda de un texto).

Selección avanzada de elementos

6.6

En los apartados anteriores hemos estudiado la sintaxis básica de una regla CSS, donde el selector puede ser un elemento HTML, una clase o un id. Pero no siempre es tan directo llegar al elemento al que queremos dar formato, o no podemos darle formato sin que afecte a otros elementos que cumplan la misma regla de selección.

En este apartado estudiaremos otros mecanismos de selección más avanzados, que nos permitirán llegar de una forma más específica a elementos particulares sobre los que aplicar estilos.

Selector descendente

Se considera que un elemento es descendiente de otro cuando se encuentra entre sus etiquetas de apertura y cierre. Por ejemplo:

HTML:

```
<div>
  <p>Lorem ipsum dolor sit <span>amet</span>, consectetur</p>
  <ul>
    <li><a href="/">Inicio</a></li>
    <li><a href="/nosotros">Sobre Nosotros</a></li>
    <li><a href="/servicios">Nuestros servicios</a></li>
  </ul>
</div>
<p>Donec rutrum magna non molestie, pretium faucibus tellus
feugiat. Aenean ut sem ut lorem interdum fermentum nec eget
velit.</p>
```

- El elemento `` es descendiente de `<p>`, que a su vez es descendiente de `<div>`.
- Los elementos `<a>` son descendientes de sus respectivos ``, que son descendientes de ``, que a su vez es descendiente de `<div>`.

Si queremos llegar a un elemento que se encuentra dentro de otro, podemos hacerlo añadiendo los selectores (elemento, clase o id) **separados por espacio [F6.6a]**:

CSS:

```
p {
  color: Red;
}

div p {
  color: Green;
  font-weight: bold;
}

div p span {
  color: Gray;
  text-decoration: underline;
}
```

F6.6a**Selector descendente**

Ejemplo de anidación de selectores.

Lorem ipsum dolor sit amet, consectetur.

- [Inicio](#)
- [Sobre Nosotros](#)
- [Nuestros servicios](#)

Donec rutrum magna non molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit.

- La primera regla afecta a cualquier <p>, independientemente de si tiene ascendentes o no. Todos los <p> se mostrarán en rojo.
- La segunda regla afecta a los elementos <p> que estén dentro de un <div>, así que sobrescribe el color del primer <p>, pasando a verde.
- La tercera regla afecta al elemento que esté dentro de un <p> que a su vez esté dentro de un <div>.

La descendencia no hace referencia al elemento de nivel inmediatamente superior. En el ejemplo anterior, la etiqueta es descendiente de <p> pero también es descendiente de <div>, por lo que la siguiente regla también sería válida:

CSS:

```
div span {
  color: Gray;
  text-decoration: underline;
}
```

Repasemos las combinaciones estudiadas hasta ahora:

- **p.error** { ... }. Selecciona los elementos de tipo <p> con el atributo class="error".
- **p .error** { ... }. Selecciona los elementos con atributo class="error" (de cualquier tipo) que estén dentro de un elemento de tipo <p>.
- **p, .error** { ... }. Selecciona todos los elementos de tipo <p> y todos los elementos con el atributo class="error" (de cualquier tipo).
- **p.error, .status** { ... }. Selecciona los elementos de tipo <p> con atributo class="error" y todos los elementos con el atributo class="status" (de cualquier tipo).

Las mismas combinaciones se pueden aplicar a ids en lugar de clases, sustituyendo el . por #. Por ejemplo:

- **p #error** { ... }. Selecciona los elementos con atributo id="error" (de cualquier tipo) que estén dentro de un elemento de tipo <p>.

Agrupación de clases

Podemos definir un selector que sólo afecte a los elementos con varias clases definidas en su propiedad class.

En el siguiente ejemplo hemos definido un <p> con class="error general mensaje" (varias clases separadas por espacio).

HTML:

```
<p class="error">Donec rutrum magna non molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit.</p>
<p class="error general mensaje">Nullam nec blandit mauris. Nullam vitae ex eu dui efficitur iaculis eget ut dui. Aliquam fringilla ornare nulla et interdum. </p>
```

El selector .error.general (sin espacios), indica que el elemento debe tener esas dos clases **[F6.6b]**.

CSS:

```
.error.general {
  color: Blue;
}

.error {
  color: Red;
}

.general {
  color: Green;
}
```

Donec rutrum magna non molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit.

Nullam nec blandit mauris. Nullam vitae ex eu dui efficitur iaculis eget ut dui. Aliquam fringilla ornare nulla et interdum.

F6.6b

Agrupación de clases

Ejemplo de agrupación de clases.

Nota: no importa el orden de las clases en la propiedad "class", y tampoco importa si el elemento tiene asignadas otras clases.

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario **[F6.6c]**:

HTML:

```
<div id="menu">
  <ul>
    <li class="destacado"><p class="error general">primer elemento</p></li>
    <li><p class="error general">segundo elemento</p></li>
    <li><p class="error general">tercer elemento</p></li>
  </ul>
</div>

<p class="error general">Donec rutrum magna non molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit.</p>
```

CSS:

```
div#menu li.destacado p.error.general {
  color: Blue;
}
```

Sólo se verá afectado por esta regla el primer elemento de la lista.

F6.6c

Agrupación de clases

Ejemplo de agrupación de clases.

- primer elemento
- segundo elemento
- tercer elemento

Donec rutrum magna non molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit.

Combinación de selectores

Los combinadores son operadores que se pueden añadir entre selectores para determinar la relación entre ellos. Como vimos anteriormente, el espacio indica descendencia de un elemento con respecto a otro. Veamos otros combinadores posibles:

- espacio. Elemento descendente (visto anteriormente).
- >. Hijo (equivale a un descendente directo, de primer nivel).
- +. Elemento hermano (mismo nivel) adyacente.
- ~. Elemento hermano (mismo nivel) general.

Selector hijo (>). Selecciona todos los hijos o descendientes inmediatos de un elemento.

Veamos como ejemplo estas dos listas () anidadas **[F6.6d]**.

HTML:

```
<div>
  <ul>
    <li>primer elemento</li>
    <li>segundo elemento</li>
    <li>
      <ul>
        <li>Elemento 2.1</li>
        <li>Elemento 2.2</li>
      </ul>
    </li>
    <li>tercer elemento</li>
  </ul>
</div>
```

CSS:

```
div > ul {
  list-style-type: circle;
}

li > ul {
  list-style-type: square;
}
```

- La primera regla afecta únicamente al que es hijo directo de un <div>, que será la lista exterior (círculo).
 - La segunda regla afecta únicamente al que es hijo directo de un , que será la lista interior (cuadrado).
- o primer elemento
 - o segundo elemento
 - Elemento 2.1
 - Elemento 2.2
 - o tercer elemento

F6.6d**Selector hijo >**

Ejemplo de uso de selector hijo.

Selector hermano adyacente (+). Selecciona los elementos hermanos adyacentes. Los elementos hermanos están dentro del mismo elemento (mismo padre) **[F6.6e]**.

HTML:

```
<h1>Título</h1>
<p>Primer párrafo</p>
<p>Segundo párrafo</p>
<p>Tercer párrafo</p>
```

CSS:

```
h1 + p {
  color: Red;
}
```

En este ejemplo estamos seleccionando los <p> que sean hermanos adyacentes de un elemento <h1>. Sólo se verá afectado el primer párrafo, ya que los otros párrafos, aunque son hermanos de <h1>, no son adyacentes.

Título

Primer párrafo

Segundo párrafo

Tercer párrafo

F6.6e**Selector hermano adyacente +**

Ejemplo de uso de selector de hermano adyacente.

Selector hermanos general (~). Selecciona todos los elementos hermanos (mismo padre) **[F6.6f]**.

HTML:

```
<h1>Título</h1>
<p>Primer párrafo</p>
<p>Segundo párrafo</p>
<p>Tercer párrafo</p>
```

CSS:

```
h1 ~ p {
  color: Red;
}
```

En este ejemplo estamos seleccionando los `<p>` que sean hermanos de un elemento `<h1>`. En este caso se verán afectados todos los párrafos.

F6.6f

Selector hermanos ~

Ejemplo de uso del selector de hermanos general.

Título

Primer párrafo

Segundo párrafo

Tercer párrafo

Pseudo-clases

Las pseudo-clases se utilizan para definir un estado específico de un elemento.

```
selector:pseudo-class {
  propiedad: valor;
  propiedad: valor;
}
```

Ya estudiamos algunas pseudo-clases asociadas a enlaces: **a:link**, **a:visited**, **a:hover** y **a:active**.

Las pseudo-clases se pueden combinar con clases e ids. En el siguiente ejemplo la regla sólo afectará a los enlaces con clase "destacado" en su estado **hover**:

CSS:

```
a.destacado:hover {
  color: Red;
}
```

:hover

Un elemento estará en este estado cuando se pase el ratón por encima. Aunque lo hemos visto aplicado a enlaces, se puede utilizar con otros elementos (`<div>`, `<p>`, ``, etc.).

En el siguiente ejemplo la imagen se muestra transparente (opacity: 0.3) **[F6.6g]** y cuando se pasa el ratón por encima, recupera toda su opacidad (opacity: 1) **[F6.6h]**.

HTML:

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
urna ut tortor pretium placerat. Vivamus dui elit,... </p>
```

CSS:

```
img {
  opacity: 0.3;
```

```
float: left;
padding: 5px;
}
```

```
img:hover {
opacity: 1;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a

F6.6g
:hover
Uso de :hover



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae urna ut tortor pretium placerat. Vivamus dui elit, Lore ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse faucibus tellus a

F6.6h
:hover
Uso de :hover

:first-child

Permite seleccionar el primer hijo de cualquier elemento. En el siguiente ejemplo se seleccionan todos los elementos <p> que sean primer hijo de cualquier otro elemento **[F6.6i]**.

HTML:

```
<div>
  <p>Primer párrafo</p>
  <p>Segundo párrafo</p>
  <p>Tercer párrafo</p>
</div>
```

CSS:

```
p:first-child {
color: Red;
}
```

Sólo se selecciona el primer <p>.

Primer párrafo

Segundo párrafo

Tercer párrafo

F6.6i
:first-child
Uso de :first-child

:last-child

Similar al anterior, pero seleccionando el último hijo de cualquier elemento **[F6.6j]**.

CSS:

```
p:last-child {  
  color: Red;  
}
```

Sólo se selecciona el último <p>.

F6.6j**:last-child**

Uso de :last-child

Primer párrafo

Segundo párrafo

Tercer párrafo

:nth-child(n)

Similar a los anteriores, pero especificando la posición del hijo que queremos seleccionar. Por ejemplo, **:nth-child(2)** seleccionará todos los segundos hijos de otros elementos **[F6.6k]**.

CSS:

```
p:nth-child(2) {  
  color: Red;  
}
```

Sólo se selecciona el segundo <p>.

F6.6k**:nth-child(n)**

Uso de :nth-child(n)

Primer párrafo

Segundo párrafo

Tercer párrafo

:first-of-type, :last-of-type, :nth-of-type(n)

Estas pseudo-classes funcionan igual que las anteriores, pero sólo se tienen en cuenta los hijos de un tipo determinado, sin importar si hay hijos de otro tipo **[F6.6l]**.

HTML:

```
<div>  
  <h1>Título</h1>  
  <p>Primer párrafo</p>  
  <p>Segundo párrafo</p>  
  <p>Tercer párrafo</p>  
</div>
```

CSS:

```
p:first-of-type {
  color: Red;
}
```

El primer hijo del <div> es <h1>, pero el primero de tipo <p> se corresponde con el primer párrafo.

Título

Primer párrafo

Segundo párrafo

Tercer párrafo

F6.6I

:first-of-type

Uso de :first-of-type

Puedes consultar otras pseudo-clases disponibles en:

https://www.w3schools.com/css/css_pseudo_classes.asp

Pseudo-elementos

Los pseudo-elementos permiten aplicar estilos a partes específicas de un elemento. Por ejemplo, cambiar el color de la primera línea de un texto, o incluso insertar contenido adicional antes o después del contenido de un elemento.

```
selector::pseudo-element {
  propiedad: valor;
  propiedad: valor;
}
```

::first-line

Selecciona la primera línea de un elemento de texto. En el siguiente ejemplo se amplía el tamaño y se cambia el color de las primeras líneas de los elementos <p> **[F6.6m]**.

HTML:

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec
  rutrum magna non urna molestie, pretium faucibus tellus feugiat.
</p>
<p>Nunc quis arcu ut risus pulvinar malesuada. Sed non condimentum
  metus. Vestibulum commodo facilisis justo vitae ullamcorper.
</p>
```

CSS:

```
p::first-line {
  color: Red;
  font-size: 20px;
}
```

F6.6m**::first-line**

Uso de ::first-line

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum magna non urna molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit. Aliquam nulla turpis, ultricies non urna eget, viverra hendrerit risus. Etiam hendrerit lectus neque.

Nunc quis arcu ut risus pulvinar malesuada. Sed non condimentum metus. Vestibulum commodo facilisis justo vitae ullamcorper. Fusce eleifend turpis in pulvinar pretium. Sed magna felis, malesuada sagittis velit sit amet, dapibus aliquam orci.

Nota: Ten en cuenta que el elemento <p> es fluido y el texto presentado en cada línea cambia junto con el tamaño de la página.

::first-letter

Selecciona la primera letra del elemento. Aplicado al HTML anterior **[F6.6n]**:

CSS:

```
p::first-letter {
  color: Red;
  font-size: 20px;
}
```

F6.6n**::first-letter**

Uso de ::first-letter

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum magna non urna molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit. Aliquam nulla turpis, ultricies non urna eget, viverra hendrerit risus. Etiam hendrerit lectus neque.

Nunc quis arcu ut risus pulvinar malesuada. Sed non condimentum metus. Vestibulum commodo facilisis justo vitae ullamcorper. Fusce eleifend turpis in pulvinar pretium. Sed magna felis, malesuada sagittis velit sit amet, dapibus aliquam orci.

::before, ::after

Permite añadir contenido antes o después del contenido de un elemento, respectivamente.

Aplicado al HTML anterior, añadimos el texto "Importante: " antes de cada párrafo **[F6.6o]**.

CSS:

```
p::before {
  content: "Importante: ";
  color: Red;
  font-size: 20px;
}
```

Importante: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum magna non urna molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit. Aliquam nulla turpis, ultricies non urna eget, viverra hendrerit risus. Etiam hendrerit lectus neque.

F6.6o
::before
 Uso de ::before

Importante: Nunc quis arcu ut risus pulvinar malesuada. Sed non condimentum metus. Vestibulum commodo facilisis justo vitae ullamcorper. Fusce eleifend turpis in pulvinar pretium. Sed magna felis, malesuada sagittis velit sit amet, dapibus aliquam orci.

Nota: Ten en cuenta que el contenido adicional se añade antes o después, pero siempre **dentro** de la etiqueta, por lo que se verá afectado por los estilos que se hayan aplicado a ese elemento.

::selection

Permite modificar los estilos del texto seleccionado por el usuario (por ejemplo, para copiar el texto) **[F6.6p]**.

CSS:

```
::selection {
  color: white;
  background: Green;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum magna non urna molestie, pretium faucibus tellus feugiat. Aenean ut sem ut lorem interdum fermentum nec eget velit. Aliquam nulla turpis, ultricies non urna eget, viverra hendrerit risus. Etiam hendrerit lectus neque.

F6.6p
::selection
 Uso de ::selection

Nunc quis arcu ut risus pulvinar malesuada. Sed non condimentum metus. Vestibulum commodo facilisis justo vitae ullamcorper. Fusce eleifend turpis in pulvinar pretium. Sed magna felis, malesuada sagittis velit sit amet, dapibus aliquam orci.

https://www.w3schools.com/css/css_pseudo_elements.asp

6.7 Estilos de formularios

Ya estudiamos en la unidad 5 los distintos elementos de formulario disponibles en HTML. En este apartado veremos cómo aplicar estilos a estos elementos. Aunque en estos ejemplos utilizaremos directamente las etiquetas como selector de CSS, podemos aplicar clases e ids de la misma forma que hemos hecho con el resto de elementos estudiados.

input

Los elementos input permiten la introducción de texto, contraseñas, números, etc., en función del tipo definido en type. Los ejemplos aquí mostrados hacen referencia a [type=text], pero se pueden aplicar en otros tipos de elementos de tipo input.

HTML:

```
<form>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname">
  <label for="lname">Last Name</label>
  <input type="text" id="lname" name="lname">
</form>
```

Comenzamos maquetando cuadros de entrada de texto [type=text]. A este elemento se le pueden aplicar algunas de las propiedades ya estudiadas, como width, padding, margin, border, border-radius, etc. **[F6.7a]**

CSS:

```
input[type=text] {
  width: 100%;
  box-sizing: border-box;
  padding: 12px 10px;
  margin: 8px 0;
  border: 2px solid green;
  border-radius: 10px;
}
```

F6.7a

input

Aplicación de estilos en elementos de tipo input.

First Name

Last Name

La propiedad **box-sizing: border box;** se utiliza para que el padding y el borde se incluyan dentro del ancho/alto del elemento. De esta forma el elemento no se hace más grande al añadir un borde o un padding. Puedes leer más sobre esta propiedad y los valores posibles en:

https://www.w3schools.com/cssref/css3_pr_box-sizing.asp

Para aplicar color de fondo podemos añadir la propiedad `background-color` **[F6.7b]**.

CSS:

```
input[type=text] {
  width: 100%;
  box-sizing: border-box;
  padding: 12px 10px;
  margin: 8px 0;
  border: 2px solid green;
  border-radius: 10px;
  background-color: #3CBC8D;
  color: white;
}
```

First Name

John

Last Name

Doe

F6.7b

input

Aplicación de estilos en elementos de tipo input.

También podemos cambiar los estilos para el estado "focus", cuando el usuario está rellenando el elemento **[F6.7c]**.

CSS:

```
input[type=text]:focus {
  background-color: lightblue;
  color: Black;
  outline: none;
}
```

First Name

John

Last Name

Doe

F6.7c

input

Aplicación de estilos en elementos de tipo input.

Al aplicar redondeado a las esquinas, añadimos la propiedad `outline: none;` que evita que en ciertos navegadores se vea un borde de fondo con las esquinas rectas.

Icono y placeholder

La propiedad `placeholder` se utiliza para añadir un texto de ayuda dentro de un cuadro de texto. Ese texto no se envía, y desaparece cuando el usuario empieza a escribir el contenido real que se enviará dentro del elemento.

En el siguiente ejemplo añadimos a un cuadro de texto (que podría ser un input de tipo search) un icono de búsqueda y un texto de ayuda o placeholder **[F6.7d]**:

HTML:

```
<form>
  <input type="text" name="search" placeholder="Search...">
</form>
```

CSS:

```
input[type=text] {
  width: 100%;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
  background-color: white;
  background-image: url('search-icon.png');
  background-position: 10px 10px;
  background-repeat: no-repeat;
  padding: 12px 20px 12px 40px;
}
```

F6.7d**input**

Aplicación de estilos en elementos de tipo input.

**textarea**

En el siguiente ejemplo aplicamos estilos a un elemento de tipo textarea. Destacamos la propiedad `resize: none`; que evita que el usuario pueda cambiar el tamaño del textarea (generalmente desde la esquina inferior derecha) **[F6.7e]**.

HTML:

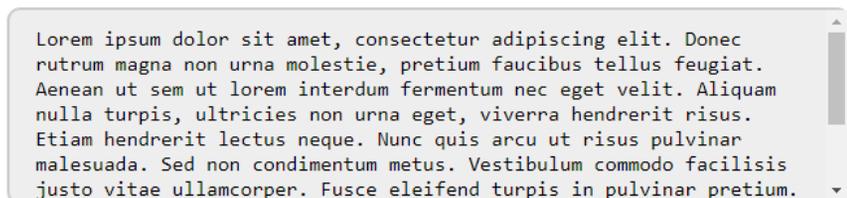
```
<form>
  <textarea></textarea>
</form>
```

CSS:

```
textarea {
  width: 100%;
  height: 150px;
  padding: 12px 20px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 10px;
  background-color: #eee;
  font-size: 16px;
  resize: none;
  outline: none;
}
```

F6.7e**textarea**

Aplicación de estilos en elementos de tipo textarea.



select

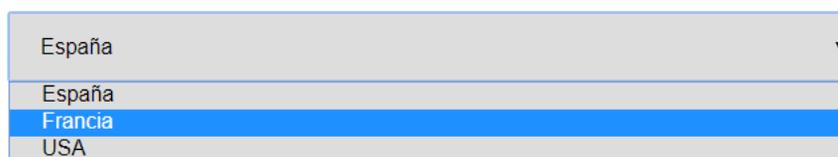
Aplicamos estilos a una lista de selección select **[F6.7f]**.

HTML:

```
<form>
  <select id="country" name="country">
    <option value="es">España</option>
    <option value="fr">Francia</option>
    <option value="usa">USA</option>
  </select>
</form>
```

CSS:

```
select {
  width: 100%;
  padding: 16px 20px;
  border: none;
  border-radius: 4px;
  background-color: #ddd;
  font-size: 16px;
}
```



F6.7f

select

Aplicación de estilos en elementos de tipo select.

Estilos de botones

Los estilos de botón se pueden aplicar a los siguientes elementos:

- input type="button"
- input type="submit"
- button
- a (enlace)

En el siguiente ejemplo todos los elementos, aunque diferentes, comparten la clase `.button`, que aporta estilos comunes. Posteriormente hemos aplicado estilos adicionales para pintar los botones con diferentes colores **[F6.7g]**.

HTML:

```
<form>
  <button class="button button1">Button</button>
  <input type="button" class="button button2" value="Input
button">
  <input type="submit" class="button button3" value="Submit">
  <a href="#" class="button button4">Link </a>
</form>
```

CSS:

```
.button {
  background-color: #4b74d0;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  font-family: Arial;
}

.button1 {background-color: #4CAF50;} /* Verde */
.button2 {background-color: #008CBA;} /* Azul */
.button3 {background-color: #f44336;} /* Rojo */
.button4 {background-color: #555555;} /* Negro */
```

F6.7g Botones

Aplicación de estilos en botones.



Consulta este enlace para ver otros estilos como sombras, hover, transparencia, grupos de botones, etc.:

https://www.w3schools.com/css/css3_buttons.asp

Consulta este otro enlace para profundizar más en los estilos de elementos de formulario:

https://www.w3schools.com/css/css_form.asp

Recursos adicionales

6.8

Herramientas de depuración HTML y CSS

Chrome Devtools:

- HTML y DOM
<https://developers.google.com/web/tools/chrome-devtools/beginners/html?hl=es>
- Inspector de elementos en Chrome (Ampliación de Unidad 3.3)
<https://developers.google.com/web/tools/chrome-devtools/css/?hl=es>
<https://developers.google.com/web/tools/chrome-devtools/inspect-styles/?hl=es>

Editores online de HTML, CSS y JavaScript:

- W3Schools Online Code Editor:
<https://www.w3schools.com/tryit/>
- Codepen
<https://codepen.io/pen/>

Documentación y ejemplos

En Internet encontrarás muchos tutoriales relacionados con HTML y CSS. Verifica que la información está actualizada y que hace referencias a CSS3.

Además de tutoriales específicos, consulta siempre el código HTML y CSS de las páginas o elementos que te gusten. Si puedes verlo en otra página web, es muy probable que puedas replicar el mismo estilo en tu sitio.

Dentro del tutorial de CSS de w3schools.com, consulta la sección **CSS Advanced**. Destacamos los siguientes contenidos:

- Menús desplegables:
https://www.w3schools.com/css/css_dropdowns.asp
- Selección de atributos
https://www.w3schools.com/css/css_attribute_selectors.asp
- Unidades en CSS:
https://www.w3schools.com/css/css_units.asp

- Esquinas redondeadas.
https://www.w3schools.com/css/css3_borders.asp
- Múltiples fondos.
https://www.w3schools.com/css/css3_backgrounds.asp
- Gradientes.
https://www.w3schools.com/css/css3_gradients.asp
- Efectos de sombras
https://www.w3schools.com/css/css3_shadows.asp
- Web Fonts
https://www.w3schools.com/css/css3_fonts.asp
- Fuentes de Google:
https://www.w3schools.com/howto/howto_google_fonts.asp

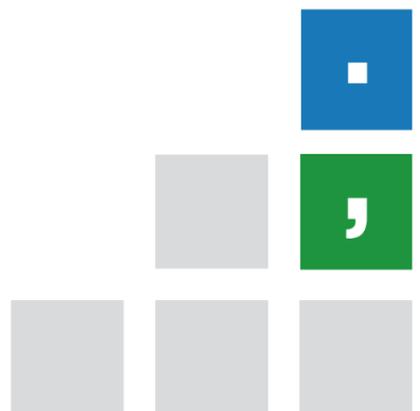
Fran Gil

Front-End Development I



Plantillas y estilos en Drupal

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

7 Plantillas y estilos en Drupal

En unidades anteriores hemos estudiado Twig, HTML y CSS. En esta unidad veremos cómo aplicar todo lo anterior en Drupal 8, siguiendo los estándares de codificación CSS, añadiendo clases CSS directamente desde la interfaz y modificando las plantillas cuando sea necesario.

También veremos algunos módulos adicionales relacionados con los archivos CSS y las plantillas Twig.

Comparativa D8/D7

Aunque los estándares estudiados también se pueden aplicar a Drupal 7, han sido desarrollados principalmente para su aplicación en Drupal 8.

Los módulos relacionados con las plantillas Twig también son exclusivos de Drupal 8.

Contenidos de la Unidad

- 7.1 Estándares de codificación CSS en Drupal
- 7.2 Librerías CSS en Drupal
- 7.3 Clases CSS desde la interfaz
- 7.4 Plantillas en módulos
- 7.5 Módulos adicionales: Advanced CSS/JS Aggregation
- 7.6 Módulos adicionales: Twig Vardumper
- 7.7 Módulos adicionales: Twig Tweak



7.1 Estándares de codificación CSS en Drupal

Drupal es un software que se desarrolla en Comunidad, de ahí que sea muy importante que el código implementado por diferentes desarrolladores sea lo más homogéneo posible. Al igual que se ha hecho con la programación Back-End, los desarrolladores Front-End han llegado a un consenso para establecer unos estándares de codificación en Drupal.

En teoría estos estándares están enfocados, principalmente, a compartir en comunidad. Si vas a desarrollar un módulo o tema, o modificar uno existente para compartir el resultado, debes seguir obligatoriamente los estándares de codificación.

Si no vas a compartir el resultado, por tratarse de un desarrollo interno, podrías pensar en saltarte estos estándares, pero te perderías las siguientes ventajas:

- **Desarrollo homogéneo dentro de tu equipo de trabajo.** Si tu equipo de trabajo sigue los estándares, asegurarás una codificación entendible incluso para futuras incorporaciones.
- **Mantenimiento.** Seguir los estándares facilita el mantenimiento cuando hay que hacer modificaciones y ampliaciones, ganando en productividad.
- **Rendimiento.** En general se persigue la mejora del rendimiento al aplicar buenas prácticas de desarrollo.
- **Evitar errores típicos.** Siguiendo los estándares se evita cometer errores típicos que a la larga afectan al mantenimiento y al rendimiento.
- **Compartir código.** Aunque un proyecto no se inicie con la idea de ser compartido, quizás más adelante surga esa posibilidad. Si no has seguido los estándares, tendrás que recodificar para adaptar el código a ellos.

Nuestra recomendación es, por tanto, seguir siempre los estándares de codificación propuestos por la comunidad.

Los estándares de codificación CSS en Drupal se basan en las metodologías SMACSS y BEM, que son de aplicación general a cualquier página web. Así que primero vamos a ver brevemente en qué se basan estas metodologías y cómo las utiliza Drupal 8.

SMACSS

SMACSS (**S**calable and **M**odular **A**rchitecture for **CSS**) es utilizado por Drupal para organizar las reglas CSS dentro de las librerías, repartiéndolas entre las siguientes categorías:

- **Base.** Contiene CSS muy genérico que se utiliza de base, y se aplica directamente a elementos HTML **sin clases**. Drupal 8 incluye en el núcleo la librería `normalize.css`, que cumple esta función de CSS base.

<https://api.drupal.org/api/drupal/core/assets/vendor/normalize-css/normalize.css/8.5.x>

Los temas pueden incluir también reglas CSS de uso general, que se ubicarán en la carpeta base. Por ejemplo, en el tema seven encontramos:

CSS: `\core\themes\seven\css\base\elements.css`

```
/**
 * Generic elements.
 */
body {
  color: #333;
  background: #fff;
  font: normal 81.3%/1.538em "Lucida Grande", "Lucida Sans
Unicode", "DejaVu Sans", "Lucida Sans", sans-serif;
}
a,
.link {
  color: #0074bd;
  text-decoration: none;
}
a:hover,
.link:hover,
a:focus,
.link:focus {
  text-decoration: underline;
  outline: 0;
}
```

- **Layout.** Estilos relacionados con la estructura o composición de la página (regiones, secciones, etc.). Se evita colocar aquí estilos como colores o bordes, definiéndose únicamente estilos relacionados con el tamaño y posición de las regiones (width, float, margin, padding).

CSS: `\core\themes\seven\css\layout\layout.css`

```
/**
 * Add spacing to bottom of pages
 */
.page-content {
  margin-bottom: 80px;
}
```

- **Component.** Dentro de esta categoría entran muchos elementos de Drupal que podemos encontrar en la interfaz (elementos de formularios, menús, campos, paginadores, tablas, pestañas, etc.).

CSS: \core\themes\seven\css\components\buttons.css

```
.button--danger: hover,
.button--danger: focus,
.button--danger: active {
  color: #ff2a00;
  text-decoration: underline;
  text-shadow: none;
  padding: 0;
  border: 0;
  box-shadow: none;
  background: none;
}
```

- **State.** Aquí se definirían estados relacionados con los estilos. En Drupal los estilos de esta categoría se incluyen junto a los componentes a los que afecta el cambio de estado.

CSS: \core\themes\seven\css\components\buttons.css

```
.button--danger: disabled,
.button--danger.is-disabled {
  color: #737373;
  cursor: default;
  text-decoration: none;
  -webkit-font-smoothing: antialiased;
  padding: 0;
  border: 0;
  box-shadow: none;
  background: none;
}
```

- **Theme.** Aquí se añadirán los estilos relacionados con el aspecto final (color, fondo, bordes, etc.) de los elementos definidos en los otros apartados.

CSS: \core\themes\seven\css\theme\colors.css

```
/**
 * Reusable colors.
 */
.color-success {
  color: #325e1c;
  background-color: #f3faef;
}
.color-warning {
  color: #734c00;
  background-color: #fdf8ed;
}
.color-error {
  color: #a51b00;
  background-color: #fcf4f2;
}
```

Estos ejemplos se han extraído del tema **Seven** del núcleo, que divide sus archivos CSS siguiendo la estructura propuesta por SMACSS:

seven

css

base

element.css
print.css
typography.css

components

breadcrumb.css
buttons.css
dialog.css
form.css
pager.css
tabs.css

...

layout

layout.css
node-add.css

theme

ckeditor-dialog.css
colors.css
install-page.css
maintenance-page.css

Como hemos comentado, en Drupal 8 no crearemos la carpeta **state**. Los estilos irán dentro de components, junto a los otros estilos de cada componente.

Aquí puedes encontrar más información sobre SMACSS: <https://smacss.com/>

BEM

El modelo BEM (Block Element Modifier) determina cómo se nombrarán las clases CSS. La estructura general a utilizar es:

```
.bloque__elemento--modificador { }
```

- Un **bloque** es una entidad independiente dentro de la interfaz de Drupal. Por ejemplo: un nodo, un paginador, un menú, un botón, una tabla, etc. Como estamos utilizando también la metodología SMACSS, en Drupal llamaremos **componentes** a estos bloques.
- Un **elemento** forma parte de un bloque, y no tiene significado fuera de él. Por ejemplo, un menú de navegación (bloque) contiene elementos de menú (elementos). Un bloque puede no tener elementos.
- Los **modificadores** son propiedades o estados de los bloques o elementos.

Veamos algunos ejemplos extraídos del tema Seven:

- `.button {}`, bloque/componente
- `.button--primary {}`, bloque/componente con modificador
- `.button--danger {}`, bloque/componente con modificar

- `.pager__item {}`, bloque/componente con elemento
- `.pager__item--first {}`, bloque/componente con elemento y modificador
- `.pager__item--next {}`, bloque/componente con elemento y modificador
- `.pager__item.is-active {}`, ejemplo con estilo aplicado a estado del componente (clase `is-active`). Como hemos comentado, se incluye en el mismo archivo de estilos del componente.

Otras reglas a tener en cuenta al nombrar las clases CSS:

- Los nombres de bloques (componentes), elementos y modificadores deben ser cortos y con significado.
- No uses caracteres especiales, sólo letras y números y guión medio (aunque también conviene evitar este último).
- No uses guión bajo como separador de palabras.
- Como Drupal se desarrolla en inglés, se recomienda que los nuevos nombres de clases estén escritos en inglés. Es algo totalmente obligatorio si vas a compartir tu código.

Puedes leer más sobre la metodología BEM en general en este enlace:

<https://www.smashingmagazine.com/2014/07/bem-methodology-for-small-projects/>

Guía de formato de CSS en Drupal 8

Puedes consultar la guía completa a partir de este enlace (menú lateral):

<https://www.drupal.org/node/1886770>

Repasaremos aquí algunas de las normas más importantes:

Indentación

- Como en el resto de programación de Drupal (PHP, YAML, JavaScript, Twig, etc.), usaremos 2 espacios para indentar, nunca tabulador.

```
@media all and (min-width: 851px) {
  .layout-container {
    max-width: 1290px;
  }
}
```

Líneas en blanco

- Las reglas se organizan en grupos de reglas relacionadas, siendo recomendado añadir un comentario como encabezado del grupo.
- Las reglas dentro de un grupo no irán separadas por una línea en blanco.
- Los grupos de reglas sí irán separados por una línea en blanco.
- Al final de cada línea no puede haber ni espacios ni tabuladores.
- Los archivos .css terminan con una única línea en blanco.

```
/**
 * Container
 */
.layout-container {
  max-width: 860px;
  margin-left: auto;
  margin-right: auto;
  box-sizing: border-box;
}

/**
 * Main
 */
.layout-main-wrapper {
  min-height: 300px;
}
.layout-main {
  margin-top: 20px;
  margin-bottom: 40px;
}
```

Comentarios

- Cada archivo debería comenzar con un comentario describiendo lo que contiene, separado con un salto de línea de la primera regla o comentario específico de un grupo de reglas.

```
/**
 * @file
 * Styles for Bartik's messages.
 */
```

```
.messages__wrapper {
}
```

- En comentarios de varias líneas, generalmente antes de un conjunto de reglas, utilizaremos este formato:

```
/**
 * Reusable heading classes are included to help modules change
 * the styling of headings on a page without affecting accessibility.
 */
```

```
h1,
.heading-a {
  margin: 1em 0 0.5em;
  font-weight: inherit;
  font-size: 1.357em;
  color: #000;
}
```

- En comentarios de una línea utilizaremos este formato:

```
/* This is needed to reset styles in item-list.theme.css */
.item-list ul {
  list-style: none;
  margin: 0 0 0.25em 0;
  padding: 0; /* LTR */
}
[dir="rtl"] .item-list ul {
  padding: 0;
}
.item-list ul li {
  /* This is needed to reset styles in item-list.theme.css */
  margin: 0;
  padding: 0.2em 0.5em 0 0; /* LTR */
}
```

Formato de regla

- Cuando una regla tiene varios selectores, se colocarán en líneas independientes, separados por coma.
- La apertura de la declaración { debe ir colocada en la misma línea del selector (o último selector), separada por un espacio.
- El cierre } irá en una línea independiente, a la misma altura que el comienzo del selector.
- Las propiedades declaradas dentro de la regla irán indentadas correctamente y siempre una propiedad por línea.

```
table tr th,
table tr th a,
table tr th a:hover,
table tr th a:focus {
  color: #fff;
  font-weight: bold;
}
```

Cuando en un bloque tenemos muchas reglas relacionadas con una única propiedad, se acepta este formato en una línea:

```
.selector1 { width: 10%; }
.selector2 { width: 20%; }
.selector3 { width: 30%; }
```

Formato de propiedades

- El formato de declaración de una propiedad es siempre:

```
propiedad: valor;
```

Por ejemplo:

```
display: block;
```

- No olvides el espacio entre los dos puntos : y el valor de la propiedad.
- Todas las propiedades llevan un ; al final, incluso la última propiedad definida en una regla.
- Cuando se utilice el formato HEX en colores, lo escribiremos en minúsculas y, siempre que se pueda, abreviado. Por ejemplo: #fff en lugar de #ffffff;
- Cuando haya que usar comillas, usaremos comillas dobles. Por ejemplo:


```
font-family: Georgia, "Times New Roman", Times, serif;
```
- Si una propiedad no requiere comillas, no las usaremos. Un caso muy típico es añadir comillas a las URLs:

Correcto: `background-image: url(path/image.png);`

Incorrecto: `background-image: url("path/image.png");`

- Cuando una propiedad tenga un valor 0, evitaremos poner las unidades.

Correcto: `margin: 0;`

Incorrecto: `margin: 0px;`

Orden de las propiedades en una regla

Aunque el orden de las propiedades no es obligatorio, se recomienda agrupar las propiedades según este orden:

1. Posicionamiento. Propiedades como position, float, clear, top, right, bottom, left, z-index, etc.
2. Modelo de caja (box model). Propiedades relacionadas con los contenedores, como display, height, width, margin, padding, border, etc.
3. Otras declaraciones. Cualquier otra propiedad no incluida o relacionada con las anteriores. Por ejemplo: background, color, font-family, etc.

Dentro de cada uno de estos grupos no se establece un orden en particular.

Veamos un ejemplo:

```
.selector {
  /* Positioning declarations */
  position: absolute;
  top: 0;
  left: 0;
  z-index: 10;
  /* Box model declarations */
  display: inline-block;
  width: 100%;
  padding: 10px;
  padding: 0.625rem;
  border: 1px solid #333;
  /* Other declarations */
  background: #000;
  color: #fff;
  font-family: sans-serif;
  font-size: 18px;
  font-size: 1.125rem;
}
```

Arquitectura CSS

En este apartado vamos a ver cómo deberíamos crear y organizar los estilos CSS. La arquitectura propuesta persigue que el código CSS sea predecible, reutilizable, mantenible y escalable. Puedes leer más sobre la arquitectura de CSS propuesta en Drupal 8 en:

<https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8>

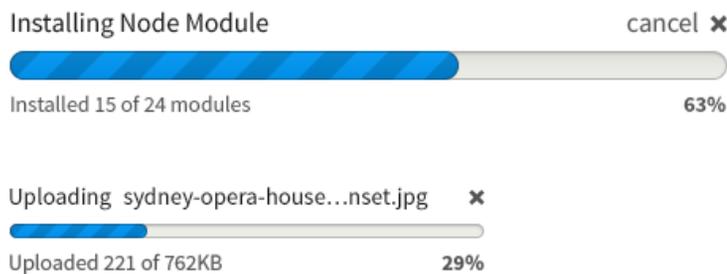
Veamos algunas de las mejores prácticas descritas:

1. Evitar dependencia de la estructura HTML de la página. Un elemento debería poder seleccionarse de forma que no depende de la página donde se muestre. Por ello lo recomendado siempre es seleccionar el elemento a través de una clase añadida al propio elemento. Cuando esa clase no está disponible, deberíamos añadir la clase al elemento directamente.
2. Definir los elementos de un componente con sus propias clases. Esto viene de la metodología BEM vista anteriormente:
 - `.pager`: clase para el componente.
 - `.pager__item`: clase para el elemento **item** dentro del componente **pager**.
3. Extender los componentes usando modificadores. Esto viene también de la metodología BEM vista anteriormente:
 - `.pager__item--first` `{}`
 - `.pager__item--next` `{}`
4. Separar los estilos en base al modelo SMACSS. Ya comentamos anteriormente que los estilos deberían separarse según la estructura

Base, Layout, Component, State y Theme. En el próximo apartado veremos cómo crear las librerías CSS creando los archivos .css mínimos necesarios.

5. Formatos de nombres de clases. En general hay que utilizar nombres de clases que aporten significado.
6. Evitar el uso de id como selector de reglas CSS. Para que una regla sea reutilizable, utilizaremos como seleccionador clases en lugar de ids, que son más específicos y sólo afectan a un elemento en particular.

Analizaremos el ejemplo propuesto en la propia guía de Drupal. Se trata de una barra de progreso que indica el porcentaje de realización de una tarea (instalar un módulo, subir un archivo, etc.) **[F7.1a]**.



F7.1a

Ejemplo clases

Definición de clases para un componente barra de progreso.

HTML

```
<div class="progress">
  <label class="label">Installing Node Module</label>

  <!-- progress element (sub-object) -->
  <div class="progress_track">
    <div class="progress_bar js-progress-percent" style="width:
63%"></div>
  </div>
  <div class="progress_description">
    <div class="layout-pull">Installed 15 of 24 modules</div>
    <strong class="layout-push">63%</strong>
  </div>

  <button class="progress_cancel" href="#"
title="cancel">cancel</button>
</div>
```

La clase principal para referenciar al componente se ha denominado: **.progress**. Los elementos del componente se han denominado:

- .progress_track
- .progress_description
- .progress_bar
- .progress_cancel

De esta forma, para aplicar estilos a un elemento, no tenemos que hacer referencia a la clase de nivel superior (.progress), sino que utilizaremos directamente el

nombre de clase asociado al elemento (.progress__bar).

- Incorrecto: `.progress .progress__bar {}`
- Correcto: `.progress__bar {}`

HTML

```
<div class="progress progress--small">
  <label class="label label--small">Uploading sydney-opera-house-
  sunset.jpg</label>
  <div class="progress__track">
    <div class="progress__bar js-progress-percent" style="width:
    29%"></div>
  </div>
  <div class="progress__description">
    <div class="layout-pull">Uploaded 221 of 762KB</div>
    <strong class="layout-push">29%</strong>
  </div>
  <a class="progress__cancel" href="#" title="cancel"><span
  class="visually-hidden">cancel</span></a>
</div>
```

Ahora definimos un segundo componente que es en realidad una **variante** del anterior (.progress--small).

- `.progress--small .progress__track {}`
- `.progress--small .progress__bar {}`
- `.progress--small .progress__cancel {}`

Ojo, aquí sí usamos el selector descendente, de forma que no será necesario crear subclases para los elementos (tipo `.progress--small__track`, `progress--small__bar`). Como vimos en el código HTML, sólo es necesario añadir la clase `progress--small`, dejando el resto de elementos HTML iguales que los del elemento `progress`.

```
<div class="progress progress--small">
```

En resumen, estas son las clases definidas para el componente y su variante:

CSS

```
/**
 * Componente Progress Bar
 */
.progress {}
.progress__track {}
.progress__bar {}
.progress__description {}
.progress__cancel {}
.progress__cancel:focus,
.progress__cancel:hover {}

/**
 * Variante Progress Bar small
 */
.progress--small .progress__track {}
.progress--small .progress__bar {}
.progress--small .progress__cancel {}
```

Librerías CSS en Drupal

7.2

Organización de archivos CSS en temas

Como ya hemos comentado, se recomienda la organización de estilos CSS siguiendo la metodología SMACSS.

Desde un punto de vista práctico definiremos en el tema las carpetas:

- css/base
- css/components
- css/layout
- css/theme

Recuerda que los estilos correspondientes a la categoría State se incluyen dentro de components, asociando los estilos de estados de un componente al propio componente. Dentro de cada carpeta se definirán uno o varios archivos css agrupando los estilos.

Como ejemplo vemos la definición de estilos en el tema Seven del núcleo:

Archivo `\core\themes\seven\seven.libraries.yml`

```
global-styling:
  version: VERSION
  css:
    base:
      css/base/elements.css: {}
      css/base/typography.css: {}
      css/base/print.css: {}
    component:
      css/components/admin-list.css: {}
      css/components/content-header.css: {}
      css/components/container-inline.css: {}
      css/components/container-inline.module.css: {}
      css/components/breadcrumb.css: {}
      css/components/buttons.css: {}
      css/components/messages.css: {}
      ...
    theme:
      css/theme/colors.css: {}
  layout:
    css/layout/layout.css: {}
```

Ya vimos anteriormente cómo añadir un archivo `.libraries.yml`, donde se declaran las librerías (CSS y JS) que tendrá el tema.

```
foo:
  version: 1.x
  css:
    theme:
      css/foo.css: {}
  js:
    js/foo.js: {}
  dependencies:
    - core/jquery
```

En los temas se añade generalmente una librería con nombre **global-styling**, que cargará todos los archivos CSS en todas las páginas donde se use el tema.

Como generalmente crearemos un subtema a partir de un tema base, podemos definir una estructura mínima de archivos CSS, siguiendo la estructura propuesta en SMACSS **[F3.3c]**:

```
global-styling:
  version: 1.x
  css:
    base:
      css/base/base.css: {}
    layout:
      css/layout/layout.css: {}
    component:
      css/components/components.css: {}
    theme:
      css/theme/theme.css: {}
```

Es importante tener en cuenta que los archivos CSS se cargarán en el orden especificado.

Organización de archivos CSS en módulos

Aunque el desarrollo de módulos custom no entra dentro de las tareas del desarrollador Front-End, sí que debe orientar al desarrollador Back-End en los siguientes aspectos:

- Plantillas y estilos a aplicar sobre los elementos definidos en el módulo.
- Ubicación y organización de los archivos CSS dentro del módulo.

Dentro del módulo se creará una carpeta css con los siguientes archivos:

- `module_name/css/module_name.module.css`. Aquí se incluyen los estilos mínimos (categorías layout, component y state).
- `module_name/css/module_name.theme.css`. Estilos dentro de la categoría theme.
- `module_name/css/module_name.admin.css`. Estilos mínimos para las páginas de administración del módulo (categorías layout, component y state).
- `module_name/css/module_name.admin.theme.css`. Estilos dentro de la categoría theme para las páginas de administración.

Como excepción, si un módulo tiene un archivo CSS asociado exclusivamente a una plantilla, el nombre del archivo CSS será igual al de la plantilla. Por ejemplo:

- Plantilla: `system-plugin-ui-form.html.twig`
- Archivo CSS: `system-plugin-ui-form.css`

Aplicando CSS a elementos existentes

Cuando cargamos una página web, el sistema se encarga de generar el HTML junto con todas las clases CSS que se utilizan o se podrían utilizar.

Drupal está preparado para que podamos maquetar cualquier elemento añadiendo estilos CSS desde el tema, sin tener que tocar HTML (plantillas) para añadir clases adicionales. De hecho, este exceso de clases hace que en la práctica muchas de ellas queden inutilizadas, lo que ha sido desde sus inicios una queja por parte de algunos desarrolladores Front-End, que abogan por un HTML más "limpio".

Nosotros nos quedaremos con la idea de que este modelo hace que la maquetación sea más sencilla, limitando la necesidad de intervenir a nivel de HTML para añadir clases adicionales.

Los nombres de clases añadidos por Drupal son semánticos, aportando un significado que nos da información contextual y del tipo de elemento al que afectan. Siempre teniendo en cuenta que se trata de terminología de Drupal y, por tanto, debemos conocer con qué tipo de elementos estamos tratando.

Aplicar estilos CSS sobre clases ya generadas implica inspeccionar y analizar el HTML generado. Como ya vimos en el apartado anterior, un elemento debería poder seleccionarse de forma que no dependa de la página donde se muestre. Es por ello lo recomendado siempre es seleccionar el elemento a través de una clase añadida al propio elemento. Si la clase no está disponible, podemos usar el selector descendente, buscando una clase padre, aunque también es posible modificar la plantilla para añadir la clase al elemento directamente.

Ejemplo 1: Nodo completo

Como ejemplo práctico vamos a analizar una versión simplificada del HTML generado al cargar un nodo de tipo artículo (article):

HTML: Contenido de tipo Artículo (article)

```
<body class="layout-two-sidebars user-logged-in path-node page-
node-type-article">

  <h1 class="title page-title">
    <span class="field field--name-title field--type-string field-
-label-hidden">Quiénes somos</span>
  </h1>

  <div class="content">

    <article class="contextual-region node node--type-article
node--view-mode-full clearfix">
      <div class="node__content clearfix">
        <div class="clearfix text-formatted field field--name-body
field--type-text-with-summary field--label-hidden field__item">

          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque eget commodo quam. In hac habitasse platea dictumst.
Donec ultrices ultricies enim id hendrerit...</p>
        </div>
      </div>
    </article>
  </div>
</body>
```

En el elemento `<body>` se añaden clases que permiten añadir estilos globales a toda la página:

- **.layout-two-sidebars.** Indica el diseño o layout seleccionado, dependiendo del tema y su configuración.
- **.user-logged-in.** Indica que el usuario visitante está registrado.
- **.path-node.** Indica que se está cargando una página de nodo o contenido.
- **.page-node-type-article.** Indica que el nodo que se está cargando es de tipo article.

En Drupal el título del contenido se separa del contenido en sí, en un `<h1>` independiente. Esta etiqueta `<h1>` tiene las clases `.title` y `.page-title`, indicando que se trata de un título de página. Además contiene un `` donde se añaden clases relacionadas con el campo donde se almacena el título, que es un campo de texto (`field--name-title`, `field--type-string`).

El elemento HTML `<article>` engloba el contenido del nodo. No hay que confundir este `<article>` con el nombre de sistema del tipo de contenido, que también es `article`. A este elemento se le han añadido clases semánticas que identifican el tipo de elemento de Drupal, un nodo o contenido (`.node`), y el tipo de contenido (`.node--type-article`), identificado como variante de `.node`.

```
<article class="contextual-region node node--type-article
node--view-mode-full clearfix">
```

También se ha añadido una clase, variante de `.node`, que identifica el modo de presentación.

- `.node.` Nodo.
- `.node--type-article.` Variante de nodo para el tipo `article`.
- `.node--view-mode-full.` Variante de nodo para el modo de presentación contenido completo.

Dentro de `<article>` se muestran los campos del tipo de contenido con sus correspondientes clases. El cuerpo o `body` es un campo de tipo Texto con formato y resumen, con las clases: `.text-formatted`, `field--name-body`, `field--type-text-with-summary`.

Teniendo este HTML generado, vamos a ver cómo utilizar las clases en diferentes situaciones. Lo haremos primero utilizando exclusivamente las clases CSS disponibles:

- a) Seleccionar todos los títulos.
`.page-title {}`
- b) Seleccionar los títulos de cualquier tipo de nodo.
`.path-node .page-title {}`
- c) Seleccionar los títulos del tipo de contenido artículo.
`.page-node-type-article .page-title {}`

- d) Seleccionar todos los campos de tipo Texto largo con formato y resumen.
`.field--type-text-with-summary {}`
- e) Seleccionar todos los campos con nombre Body.
`.field--name-body {}`
- f) Seleccionar todos los campos Body pero dentro del tipo de contenido Artículo.
`.node--type-article .field--name-body {}`

En los casos en los que hemos tenido que ser más específicos (b, c y f), nos hemos visto obligados a utilizar el **selector descendente**. Aunque es totalmente correcto a nivel de CSS, estaríamos incumpliendo el punto 1 de las mejores prácticas:

1. Evitar dependencia de la estructura HTML de la página. Un elemento debería poder seleccionarse de forma que no depende de la página donde se muestre. Es por ello lo recomendado siempre es seleccionar el elemento a través de una clase añadida al propio elemento. **Cuando esa clase no está disponible, deberíamos añadir la clase al elemento directamente.**

En la situación actual, si queremos seleccionar el campo body en los tipos de contenido artículo, página básica y webform, deberíamos añadir la siguiente regla:

```
.node--type-article .field--name-body,
.node--type-page .field--name-body,
.node--type-webform .field--name-body {}
```

Sin embargo, si añadimos una clase específica para el elemento (variante):

```
<div class="clearfix text-formatted field field--name-body
field--name-body-large field--type-text-with-summary field--label-
hidden field__item">
```

Estaríamos simplificando la regla anterior a:

```
.field--name-body-large {}
```

Eso sí, tendríamos que asegurarnos de que esta nueva clase se imprime para esos tipos de contenido, y eso **implica modificar la plantilla** correspondiente, como veremos más adelante.

Ejemplo 2: Portada con resúmenes

En el siguiente ejemplo se muestra la página de portada del sitio, identificada en el `<body>` con la clase `.path-frontpage`.

HTML: Portada del sitio (artículo en modo resumen)

```
<body class="layout-two-sidebars user-logged-in path-frontpage">

  <main id="content" class="column main-content"
role="main">
```

```

        <section class="section">
          <a id="main-content" tabindex="-1"></a>
          <div class="content">
<div class="contextual-region view view-frontpage view-id-
frontpage view-display-id-page_1 ">
  <div class="view-content">
    <div class="views-row">

<article class="contextual-region node node--type-article node--
promoted node--view-mode-teaser clearfix">
  <header>
    <h2 class="node_title">
      <a href="/node/3" rel="bookmark">
<span class="field field--name-title field--type-string field--
label-hidden">Lorem ipsum</span></a>
    </h2>
  </header>

  <div class="node_content clearfix">
    <div class="clearfix text-formatted field field--name-body
field--type-text-with-summary field--label-hidden field_item">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse faucibus tellus a scelerisque lobortis. Proin vitae
urna ut tortor pretium placerat. Vivamus dui elit, ultricies
ornare lorem vitae, facilisis condimentum nulla. Praesent
consequat aliquam ligula. Vivamus risus erat, iaculis iaculis
ornare maximus, luctus ac magna. Nulla blandit arcu eget neque
ullamcorper aliquam. Aliquam mattis nibh ut interdum vehicula.
Aliquam convallis sed tellus et dapibus.
      </p>
    </div>
  </div>

</article>

</div>
</body>

```

La portada muestra un listado de resúmenes de nodos promocionados a la portada. Este listado es una **vista**, identificada también en las clases.

```

<div class="contextual-region view view-frontpage view-id-
frontpage view-display-id-page_1 ">
  <div class="view-content">
    <div class="views-row">

```

Cada resumen de nodo tiene su propio `<article>`, que incluye el título (`<h2>` en este caso) y el resto de campos asignados en el modo de presentación Resumen (Teaser).

```

<article class="contextual-region node node--type-article
node--promoted node--view-mode-teaser clearfix">

```

Encontramos las siguientes clases:

- `.node`. Nodo.
- `.node--type-article`. Variante de nodo para el tipo article.
- `.node--view-mode-teaser`. Variante de nodo para el modo de presentación resumen.
- `node--promoted`. Variante de nodo para nodos promocionados a la portada.

La selección de clases sería igual que la vista anteriormente. Por ejemplo:

- a) Seleccionar todos los campos Body dentro del tipo de contenido Artículo cuando está promocionado a la portada y se presenta en modo resumen.

```
.node--type-article.node--promoted.node--view-mode-teaser .field--name-body {}
```

Nótese que las tres primeras clases están unidas, ya que el elemento tiene que tener todas esas clases. Luego usamos el selector descendente para localizar a la clase hija `.field--name-body`.

Como comentamos antes, con una clase específica para esa variante del elemento, podemos ahorrar código y hacerlo más mantenible y reutilizable.

Ejemplo 3: Bloque

En el siguiente ejemplo se muestra el código HTML y las clases generadas para el bloque Usuarios nuevos (Who's new):

HTML: Bloque Usuarios nuevos

```
<div class="region region-sidebar-second">
  <div class="views-element-container contextual-region block
block-views block-views-blockwho-s-new-block-1" id="block-b-
forcontu-views-block-who-s-new-block-1">
  <h2>Usuarios nuevos</h2>
  <div class="content">
    <div class="view-content">
      <div class="item-list">
        <ul>
          <li>
            <div class="views-field views-field-name"><span
class="field-content"><a title="Ver perfil del usuario."
href="/user/1" lang="" about="/user/1" typeof="schema:Person"
property="schema:name" datatype=""
class="username">admin</a></span>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>
</div>
```

Se trata de un bloque generado desde Vistas, de ahí que se muestren algunas clases relacionadas con views:

```
<div class="views-element-container contextual-region block
block-views block-views-blockwho-s-new-block-1">
```

El sistema añade una clase específica para cada bloque, usando su nombre de sistema (`.block-views-blockwho-s-new-block-1`).

Algunos selectores:

a) Seleccionar todos los bloques:

```
.block {}
```

b) Seleccionar todos los bloques generados desde vistas:

```
.block-views {}
```

c) Seleccionar todos los bloques colocados en la columna derecha (Sidebar second):

```
.region-sidebar-second .block {}
```

Si queremos seleccionar varios bloques para aplicar un estilo determinado, pero esos bloques no guardan una relación entre sí, tendremos que añadir una clase adicional que los identifique. Por ejemplo:

```
.block--small {}
```

Veremos en el próximo apartado módulos adicionales que nos permiten añadir clases adicionales en módulos específicos.

Añadir clases en plantillas

Hemos comentado que en algunos casos no es suficiente con las clases generadas, de forma que necesitamos añadir clases adicionales a determinados elementos. Uno de las formas de conseguir esto es modificando las plantillas.

Ejemplo 1: Plantilla de nodo

En la plantilla general de nodo (`node.html.twig`), se utiliza una variable `classes` donde se registran las clases que se añadirán a la etiqueta `<article>`.

Twig: `node.html.twig`

```
{%
  set classes = [
    'node',
    'node--type-' ~ node.bundle|clean_class,
    node.isPromoted() ? 'node--promoted',
    node.isSticky() ? 'node--sticky',
    not node.isPublished() ? 'node--unpublished',
    view_mode ? 'node--view-mode-' ~ view_mode|clean_class,
    'clearfix',
  ]
%}
<article{{ attributes.addClass(classes) }}>
```

Posteriormente, en la etiqueta `<article>` se utiliza la función `addClass` para añadir todas las clases indicadas.

```
<article{{ attributes.addClass(classes) }}>
```

Si añadimos la clase `'node--large'` al listado, se mostrará junto a las otras clases asignadas. Aquí podemos aplicar cierta lógica de programación, o copiar la plantilla con otro nombre (como vimos en el apartado 4.4, Sugerencias de plantillas).

Twig: `node.html.twig` (modificado)

```
{%
  set classes = [
    'node',
    'node--type-' ~ node.bundle|clean_class,
    node.isPromoted() ? 'node--promoted',
    node.isSticky() ? 'node--sticky',
    not node.isPublished() ? 'node--unpublished',
    view_mode ? 'node--view-mode-' ~ view_mode|clean_class,
    (node.bundle == 'page') or (node.bundle == 'article') ? 'node--large',
    'clearfix',
  ]
%}
<article{{ attributes.addClass(classes) }}>
```

Hemos añadido una comprobación de tipo de contenido (`node.bundle`). Si el tipo es `page` o `article`, se añade la clase `'node--large'`.

Ejemplo 2: Plantilla de bloque

En este ejemplo hemos creado una plantilla específica para el bloque 'custom-block', a partir de la plantilla genérica block.html.twig. Al tratarse de una plantilla específica, podemos añadir la clase adicional directamente.

Twig: block--**custom-block**.html.twig

```
{%
  set classes = [
    'block',
    'block-' ~ configuration.provider|clean_class,
    'block-' ~ plugin_id|clean_class,
    'block--small',
  ]
%}
<div{{ attributes.addClass(classes) }}>
```

Clases CSS desde la interfaz

7.3

Ya sabemos que, cuando necesitamos añadir clases adicionales a un elemento, lo podemos hacer modificando la plantilla correspondiente. Sin embargo, hay algunos módulos adicionales que permiten añadir clases CSS adicionales desde la interfaz de administración. Algunos de estos módulos ya han sido estudiados en el curso de Site Building, por lo que aquí sólo veremos las opciones relacionadas con la inyección de clases CSS.

Clases en bloques

El módulo **Block Class** permite añadir clases adicionales a los bloques.

https://www.drupal.org/project/block_class

Una vez instalado, accede a la configuración de cualquier bloque y añade una o varias clases, separadas por espacio **[F7.3a]**.

Configurar bloque ☆

Configurar bloque Devel

Inicio » Administración » Estructura » Diseño de bloques

Descripción del bloque: Usuarios nuevos

Mostrar título

Elementos por bloque

5 (configuración predeterminada) ▼

Sobrescribir título

Visibilidad

Tipos de contenido

Sin restricción

Páginas

Sin restricción

Tipos de contenido

Artículo

Página básica

Nombre de sistema *

b_forcontu_views_block__who_s_new_block_1

Un nombre único para esta instancia del bloque. Debe ser alfanumérico y separado por guiones bajos.

Región *

Sidebar second ▼

Seleccione la región donde este bloque debería ser mostrado.

Clase(s) CSS

block--small

Customize the styling of this block by adding CSS classes. Separate multiple classes by spaces.

[Guardar el bloque](#) [Eliminar bloque](#)

F7.3a

Módulo Block Class

Permite añadir clases CSS a bloques.

Las clases se añadirán en el contenedor principal del bloque.

HTML

```
<div class="views-element-container block--small contextual-region
block block-views block-views-blockwho-s-new-block-1" id="block-b-
forcontu-views-block-who-s-new-block-1">
```

Clases en campos

El módulo **Field Formatter Class** permite añadir clases a los campos de cualquier entidad.

https://www.drupal.org/project/field_formatter_class

Las clases se añaden desde la pestaña **Administrar/Gestionar presentación** de la entidad (tipo de contenido, usuario, taxonomías, etc.), accediendo a la configuración del campo. Recuerda seleccionar siempre el modo de presentación correcto.

Por ejemplo, añadimos las siguientes clases CSS al campo **Body** del tipo de contenido artículo, en el modo de presentación Predeterminado **[F7.3b]**:

F7.3b Módulo Field Formatter Class

Permite añadir clases CSS a los campos de cualquier entidad.



Las clases se añadirán en el contenedor del campo:

HTML

```
<div class="body--custom body--highlighted clearfix text-formatted
field field--name-body field--type-text-with-summary field--label-
hidden field_item quickedit-field">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...
```

Clases en vistas

Esta opción está disponible en las vistas sin necesidad de instalar ningún módulo adicional. Edita la vista y despliega la pestaña Avanzado. En la opción **Clase CSS** [F7.3c] podrás añadir clases adicionales, separadas por espacio [F7.3d].

The screenshot shows the 'AVANZADO' configuration tab for a Drupal view. Under the 'FILTROS CONTEXTUALES' section, the 'Clase CSS' field is highlighted with a red box and contains the text 'Ninguno'. Other visible options include 'FORMULARIO EXPUESTO' (Básico | Configuración), 'OTRO(S)' (Nombre de sistema: block_1), and 'Opciones de configuración de consulta' (Configuración).

F7.3c Clases en vistas

Opción incluida en el núcleo para añadir clases CSS en vistas.

The screenshot shows a configuration dialog titled 'Usuarios nuevos: Clase CSS'. It features a text input field containing 'views--highlighted' and a label 'Nombre(s) de clase(s) CSS'. Below the input is the instruction 'Separe múltiples clases con espacios'. At the bottom, there are two buttons: 'Aplicar' and 'Cancelar'.

F7.3d Clases en vistas

Opción incluida en el núcleo para añadir clases CSS en vistas.

Las clases se añadirán en el contenedor de la vista:

HTML

```
<div class="views-element-container block block-views block-views-blockwho-s-new-block-1" id="block-b-forcontu-views-block-who-s-new-block-1">
  <h2>Usuarios nuevos</h2>
  <div class="content">
    <div>
      <div class="views--highlighted contextual-region view view-who-s-new view-id-who_s_new view-display-id-block_1">
```

Display Suite

Ya estudiamos Display Suite en la **Unidad 32** del libro Experto en Site Building III. Vamos a repasar las opciones para añadir clases adicionales a determinados elementos de Drupal.

Etiquetas personalizadas

Desde la pestaña **Custom wrappers** podemos cambiar las etiquetas HTML por defecto que se utilizan como contenedores o encapsuladores de cada región. Por defecto se utiliza la etiqueta `<div>`, pero desde aquí podemos establecer otros valores, además de añadir propiedades adicionales **[F7.3e]**.

F7.3e

Etiquetas personalizadas

Definición de etiquetas contenedoras personalizadas.

The screenshot shows the configuration interface for the 'Layout for autor in full' region. The 'Custom wrappers' tab is selected and highlighted with a red box. The configuration options are as follows:

- Choose a wrapper. All Display Suite layouts support this option.**
- Wrapper for Left:** Div
- Wrapper for Right:** Div
- Outer wrapper:** Div
- Layout attributes:** (Text input field)
- E.g.:** role|navigation,data-something|some value
- Añadir enlace:** Sin enlace
- This will add an onclick attribute on the layout wrapper.**

Clases personalizadas en regiones

Desde **Clases personalizadas (Custom classes)** podemos añadir clases CSS que se aplicarán en las regiones indicadas.

Primero añadiremos nuevas clases desde **[F7.3f]**:

Administración ⇒ **Estructura** ⇒ **Display suite** ⇒ **Clases**

Classes ☆

Presentaciones
Classes
Campos

Inicio » Administración » Estructura » Presentaciones

CSS classes for regions

autor-left|Autor columna izquierda
autor-right|Autor columna derecha

Configure CSS classes which you can add to regions on the "manage display" screens. Add multiple CSS classes line by line.

If you want to have a friendly name, separate class and friendly name by |, but this is not required. eg:

```
class_name_1
class_name_2|Friendly name
class_name_3
```

CSS classes for fields

autor-nacimiento|Campo autor fecha nacimiento
autor-pais|Campo autor país

Configure CSS classes which you can add to fields on the "manage display" screens. Add multiple CSS classes line by line.

If you want to have a friendly name, separate class and friendly name by |, but this is not required. eg:

```
class_name_1
class_name_2|Friendly name
class_name_3
```

Guardar configuración

F7.3f Estilos para regiones

Listado de estilos que se aplicarán en las regiones. Estos estilos deberán definirse en la hoja de estilos del tema (archivos CSS).

Podemos especificar clases simplemente indicando el nombre de la clase, o mediante la notación:

clase|Nombre descriptivo

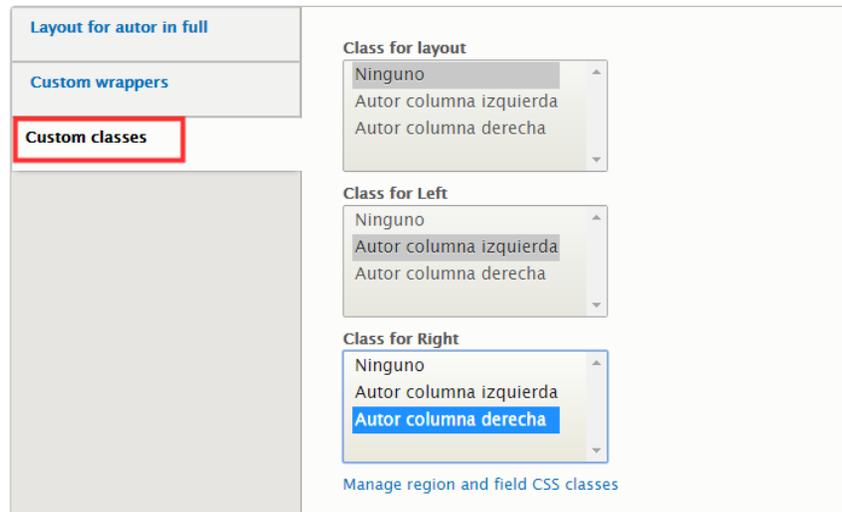
En esta notación, la clase es la cadena que se encuentra a la izquierda del símbolo | (barra vertical) y a la derecha podemos colocar un nombre más amigable que será el que veamos desde el área de administración de **Display suite**.

Una vez añadidas las clases, tendremos que asignarlas a las regiones adecuadas desde la pestaña **Clases personalizadas (Custom classes) [F7.3g]**.

F7.3g

Clases adicionales

Los Estilos creados para las regiones estarán disponibles en cada presentación para ser asignados a las regiones del diseño.



También podemos asignar clases que se aplicarán en todo el diseño (**class for layout**), en un nivel superior a las regiones.

Todas las clases añadidas se incluirán en el HTML generado, lo que permitirá aplicar estilos a cada región desde la hoja de estilos CSS del tema.

```
<div class="content">
  <div data-history-node-id="12" class="node node--type-autor
node--view-mode-full ds-2col clearfix">

  <div class="autor-left group-left">
    contenido región izquierda
  </div>

  <div class="autor-right group-right">
    contenido región derecha
  </div>

</div>
```

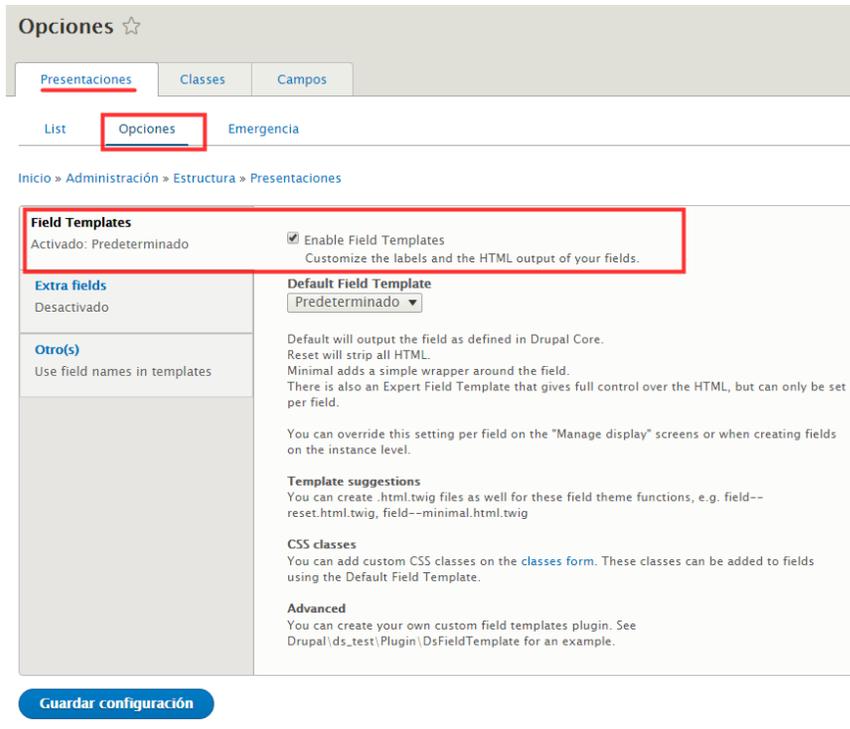
Clases personalizadas en campos

También podemos añadir clases personalizadas que podremos asignar a la presentación de campos. Las clases para campos también se añaden desde **[F7.3f]**:

Administración ⇒ Estructura ⇒ Display suite ⇒ Clases

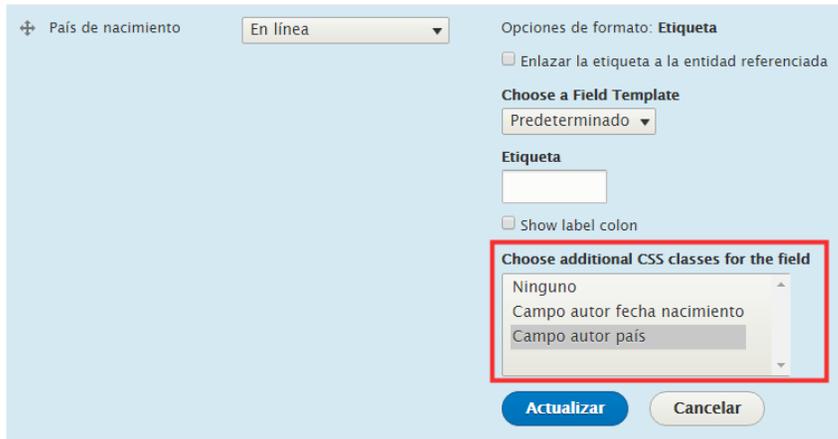
Para que las clases estén disponibles en los campos, tenemos que activar la opción Activar plantillas de campos (Enable Field Templates) desde las Opciones de configuración de Display Suite **[F7.3h]**:

Administración ⇒ Estructura ⇒ Display suite ⇒ Opciones [Field Templates]



F7.3h
Estilos para campos
 Activación de plantillas de campos para poder añadir estilos a los campos.

Una vez activada esta opción, podremos asignar clases a cualquiera de los campos disponibles en **Administrar presentación [F7.3i]**.



F7.3i
Estilos para campos
 Selección de clases de estilos en la presentación de un campo.

Estas clases se añadirán al código HTML generado:

```
<div class="autor-pais field field--name-field-autor-pais field--type-entity-reference field--label-inline">
  <div class="field_label">País de nacimiento</div>
  <div class="field__item">India</div>
</div>

<div class="autor-nacimiento field field--name-field-autor-nacimiento field--type-datetime field--label-inline">
  <div class="field_label">Fecha de nacimiento</div>
  <div class="field__item">25-06-1903</div>
</div>
```

Clases en Webform

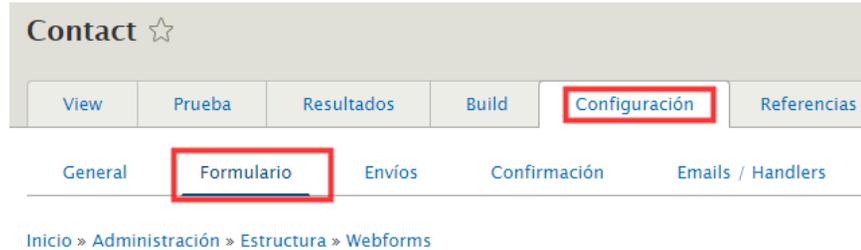
El módulo Webform incluye opciones de configuración que permiten añadir clases CSS tanto a formularios completos como a elementos individuales.

Las clases generales para todo el formulario se pueden añadir editando el formulario, pestaña Configuración [Formulario] [F7.3j]:

F7.3j

Webform

Estilos aplicados a todo el formulario.

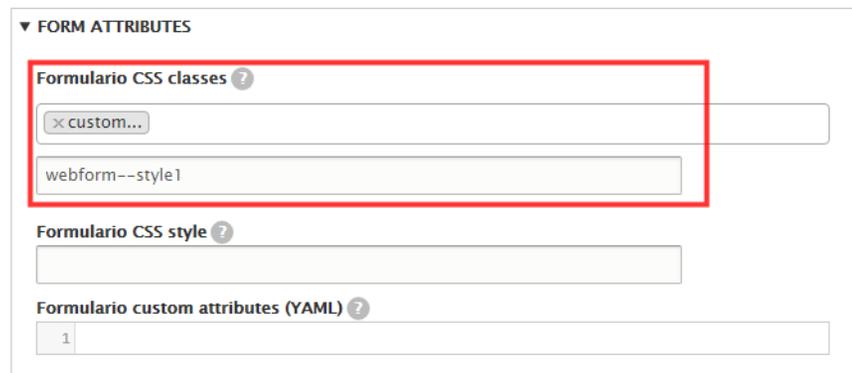


Selecciona "custom..." para añadir clases adicionales no incluidas en el listado [F7.3j]:

F7.3k

Webform

Estilos aplicados a todo el formulario.

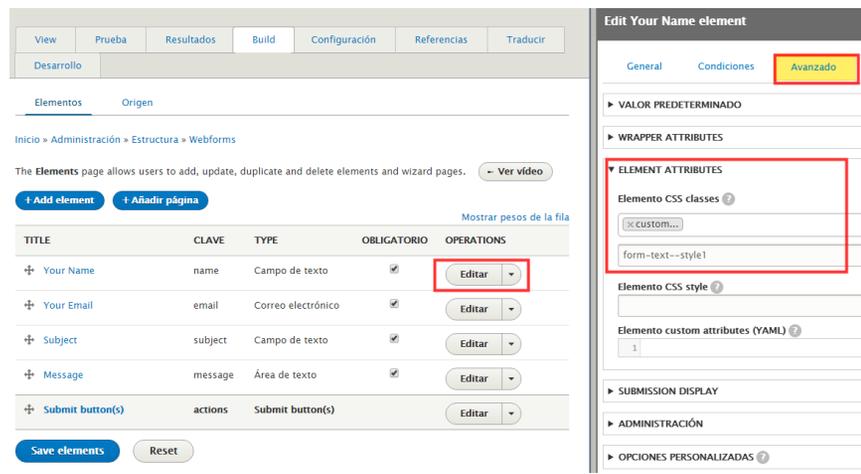


Desde la pestaña Build [Elementos], selecciona un elemento de formulario para aplicar clases específicas (pestaña Avanzado, apartado Element Attributes). Selecciona "custom..." para añadir nuevas clases [F7.3l]:

F7.3l

Webform

Estilos aplicados a un campo en particular.



A continuación, se muestra el HTML resultante con las clases aplicadas. Las clases generales se añaden en la etiqueta <form>, mientras que las clases de elemento se añaden directamente en la etiqueta del elemento (en este ejemplo, un elemento <input>).

```
<form class="webform-submission-form webform-submission-add-form
webform-submission-contact-form webform-submission-contact-add-
form webform--style1 js-webform-details-toggle webform-details-
toggle contextual-region" data-drupal-selector="webform-
submission-contact-add-form" action="/form/contact" method="post"
id="webform-submission-contact-add-form">
  <input class="form-text--style1 form-text required" data-
drupal-selector="edit-name" type="text" id="edit-name" name="name"
value="admin" size="60" maxlength="255" required="required">
```

Fuentes adicionales

El módulo **@font-your-face** permite añadir fuentes de texto adicionales de diferentes proveedores:

<https://www.drupal.org/project/fontyourface>

Nota: Al usar una fuente externa es importante que verifiques **la licencia de uso**, por si podría tener alguna limitación o pago requerido, especialmente si el portal requiere una **licencia de uso Comercial**.

El módulo tiene un submódulo principal (@fontyourface) y submódulos independientes por cada proveedor de fuentes. La recomendación es que revises primero las fuentes en los enlaces de cada proveedor, hasta localizar la fuente o fuentes que usarás en tu sitio. De esta forma sólo tendrás que instalar los módulos que realmente vayas a utilizar. También puedes instalarlos todos, navegar entre las fuentes desde tu sitio, y desinstalar los módulos que finalmente no vayas a utilizar. Los submódulos disponibles son:

- @fontyourface. Módulo base para gestionar las fuentes.
- @fontyourface - Adobe Edge Fonts
<https://edgwebfonts.adobe.com/fonts>
- @fontyourface - Font Squirrel
<https://www.fontsquirrel.com/>
- @fontyourface - Fonts.com
<https://www.fonts.com/>
- @fontyourface - Google Fonts
<https://fonts.google.com/>
- @fontyourface - Local Fonts (fuentes locales instaladas en el servidor).
- @fontyourface - Typekit
<https://typekit.com/fonts>

Accede a la configuración desde:

Administración ⇒ **Apariencia** ⇒ **@font-your-face [Configuración]**

Importa todas las fuentes (o las de un proveedor en particular) para que estén disponibles en el sitio **[F7.3m]**:

F7.3m

@font-your-face

Importación de fuentes de diferentes proveedores.

The screenshot shows the 'Font settings' configuration page for the '@font-your-face' module. It includes tabs for 'Lista', '@font-your-face', 'Actualizar', and 'Configuración'. Below the tabs are sub-tabs for 'Browse', 'Font display', and 'Configuración'. The main content area contains a checkbox for 'Load all enabled fonts' which is checked, with a warning message below it. There is an 'IMPORT' section with two buttons: 'Import from google_fonts_api' and 'Import all fonts'. At the bottom, there is a blue 'Guardar configuración' button.

Desde la pestaña **Explorar (Browse)**, podrás consultar las fuentes importadas. Activa únicamente las fuentes que vayas a utilizar (enlace **Activar** en cada fuente) **[F7.3n]**.

F7.3n

@font-your-face

Activa las fuentes que vayas a utilizar en tu sitio.

The screenshot shows the 'Font Selector' configuration page. It has tabs for 'Lista', '@font-your-face', 'Actualizar', and 'Configuración', with sub-tabs for 'Browse', 'Font display', and 'Configuración'. Below the tabs, there are several dropdown menus for 'Name' (set to 'baloo'), 'Font Provider', 'CSS Style', 'CSS Weight', 'Classification', 'Language/subset', and 'Tags'. An 'Apply' button is present. Below the filters, there are four font preview cards for 'Baloo Bhai regular' from 'google_fonts_api'. The first card is for 'gujarati' and is active. The second card is for 'latin' and is deactivated. The third card is for 'latin-ext' and is active. The fourth card is for 'vietnamese' and is active. Each card shows the font name, provider, and a large 'AaGg' sample.

Una vez activada la fuente, está ya disponible en el sitio. Para usar la fuente dentro de un estilo, tenemos dos opciones:

1. Usar la fuente directamente, añadiendo la fuente en cualquier regla CSS, en los archivos CSS del tema.

Desde Explorar, haciendo clic en el nombre de la fuente accederás a información detallada sobre cómo usar la fuente y el aspecto que tendrá aplicada a diferentes elementos HTML (encabezados, listas, etc.).

El código CSS señalado es el que debes utilizar dentro de cualquier regla CSS **[F7.3o]**:

Name

Baloo Bhai regular (latin) ☆

Ver
Devel

F7.3o
[@font-your-face](#)
 Detalle de fuente.

[Inicio](#) » [Administración](#) » [Apariencia](#) » [Font Selector](#)

If you wish to use the font in your theme / css file, use:

```
font-family: 'Baloo Bhai';
font-style: normal;
font-weight: normal;
```

▼ DETAILS

google_fonts_api

Font URL
[https://www.google.com/webfonts/family?family=Baloo Bhai&subset=latin#regular](https://www.google.com/webfonts/family?family=Baloo+Bhai&subset=latin#regular)

CSS Family
 Baloo Bhai

CSS Style
 normal

CSS Weight
 normal

PREVIEW

h1. This is a very large header.

h2. This is a large header.

h3. This is a medium header.

2. El módulo incluye una opción para utilizar la fuente sin necesidad de tocar los archivos CSS del sitio. Desde la pestaña Font Display, añade una nueva presentación de fuente (Add Font Display) **[F7.3p]**:

- Escribe una etiqueta para este display.
- Selecciona la Fuente de entre las fuentes activadas.
- Puedes añadir un listado con las fuentes alternativas que se utilizarán en caso de que no se pueda cargar la fuente o que no sea compatible con el navegador del usuario.
- Selecciona un selector o escribe un selector personalizado (Other).
- Por último, selecciona el tema donde se aplicará el estilo.

F7.3p

@font-your-face

Crear un Font display para añadir directamente la fuente a una clase o regla CSS.

Add Font display ☆

Inicio » Administración » Apariencia » Font Selector » Font display

Etiqueta *
 Nombre de sistema: node_title [Editar]

Label for the Font display.

Fuente *

Select the font to use as part of the font style

USAGE

If you wish to skip using the font display and add the css directly to your theme, copy/paste the following for the font into your theme css file:

```
font-family: 'Baloo Bhai';
font-style: normal;
font-weight: normal;
```

Fallback fonts

Fallback fonts in case selected font fails to load.

Preset Selectors *

Use preset selectors to easily display your font.

Selectores

Selects the selected font will apply to. Note that all pages will have a "fontyourface" class on the body tag. You can use that to specify a font.

Tema *

Select theme this display will work for.

[Guardar](#)

Los displays creados se mostrarán ahora listados en la pestaña Font display [F7.3q]:

F7.3q

@font-your-face

Listado de presentaciones de fuentes creadas.

Font display ☆

Lista
@font-your-face
Actualizar
Configuración

Explorar
Font display
Configuración

Inicio » Administración » Apariencia » Font Selector

[+ Add Font display](#)

FONT DISPLAY	NOMBRE DE SISTEMA	FUENTE	TEMA	OPERATIONS
Node title	node_title	Baloo Bhai regular (latin)	b_forcontu	Editar ▼

Como ejemplo hemos añadido la fuente 'Baloo Bhai' a la clase **.page-title**, generándose el siguiente código:

```
.page-title {
  font-family: 'Baloo Bhai',Georgia,"Times New Roman",Times;
  font-style: normal;
  font-weight: normal;
}
```

El estilo se aplicará en los títulos de los contenidos donde se use esta clase (por defecto) **[F7.3r]**:

Quiénes somos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque eget commodo quam. In hac habitasse platea dictumst. Donec ultrices ultricies enim id hendrerit. Suspendisse egestas ex mauris, vitae ultricies libero aliquam a. Mauris pellentesque id mauris nec pretium. Quisque tristique dui egestas sapien mollis dapibus. Vestibulum suscipit dictum metus ut gravida. Nullam vitae orci vel mauris tincidunt bibendum sed sit amet enim. Morbi pellentesque mauris lacus, eget volutpat risus finibus ut. Duis eu dictum sem. Aliquam ornare, ex eu tempus mattis, ipsum ante tincidunt sapien, at congue felis dui vitae massa. Vestibulum hendrerit lectus id lectus porttitor eleifend.

F7.3r

@font-your-face

Resultado final aplicando la fuente a los títulos de los nodos.

Aquí vemos el código incluido en el encabezado de la página para cargar la fuente desde la página del proveedor externo:

```
<head>
  <link rel="stylesheet"
  href="https://fonts.googleapis.com/css?family=Baloo+Bhai:regular&
  mp;subset=latin" media="all">
</head>
```

7.4 Plantillas en módulos

En el **apartado 4.3** vimos cómo activar la depuración de plantillas para conocer qué plantilla se está utilizando para cada fragmento de HTML. Si necesitamos realizar algún cambio a nivel de HTML, como añadir nuevas etiquetas o clases para aplicar estilos, sólo tenemos que copiar la plantilla en nuestro tema y hacer allí las modificaciones.

La depuración de plantillas también nos aporta información sobre las sugerencias de plantillas (**apartado 4.4**), que son nombres alternativos para crear plantillas más específicas.

En este apartado vamos a revisar las plantillas que incluyen algunos módulos de uso común y sus correspondientes sugerencias de plantillas. También veremos qué módulos permiten utilizar código Twig desde la interfaz.

html.html.twig

Aunque hemos comentado que lo ideal es aplicar estilos a nivel de componentes, en ocasiones es necesario diferenciar a un nivel superior, en función del diseño o layout de la página.

Las clases de nivel más alto para el contenido se especifican en la etiqueta <body>, desde la plantilla **html.html.twig**. Por ejemplo, al cargar un node de tipo artículo (article), tendremos las siguientes clases disponibles:

```
<body class="layout-two-sidebars user-logged-in path-node page-node-type-article">
```

Las sugerencias de plantillas dependerán del tipo de página que estemos cargando:

- html.html.twig. Plantilla original de base.

Página principal del sitio:

```
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--front.html.twig
  * html--node.html.twig
  x html.html.twig
-->
```

- html--node.html.twig. 'node' se corresponde con el nombre de sistema de la vista de portada por defecto.
- html--front.html.twig. 'front' hace referencia a la página principal del sitio independientemente de la página o vista que se utilice como portada.

Página de nodo:

```
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--node--3.html.twig
  * html--node--%.html.twig
  * html--node.html.twig
  x html.html.twig
-->
```

- html--node.html.twig. 'node' se corresponde con el nombre de sistema de la vista de portada por defecto. También es la base de URL de un nodo /node.
- html--node--3.html.twig. 3 se corresponde con un id de nodo en particular. Permite implementar una plantilla específica para el nodo indicado.
- html--node--%.html.twig. % se corresponde con un parámetro de URL numérico. Esta plantilla es válida para contenidos con URL de tipo node/%. Permite diferenciar de /node (sin parámetro).

Página de vista:

```
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--documents.html.twig
  x html.html.twig
-->
```

- html--documents.html.twig. 'documents' se corresponde con el nombre de sistema de la vista.

Página de formulario (webform):

```
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--webform--contact.html.twig
  * html--webform.html.twig
  x html.html.twig
-->
```

- html--webform--contact.html.twig. Plantilla específica para un formulario. 'contact' se corresponde con el nombre de sistema del formulario.
- html--webform.html.twig. Plantilla genérica para todos los formularios de webform.

Página de panel (page manager):

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * html--homepage.html.twig
  x html.html.twig
-->
```

- html--homepage.html.twig. 'homepage' se corresponde con el nombre de sistema de la página.

Página de perfil de usuario:

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
  * html--user--1.html.twig
  * html--user--%.html.twig
  * html--user.html.twig
  x html.html.twig
-->
```

- html--user.html.twig. Plantilla para todas las páginas de perfil de usuario.
- html--user--1.html.twig. 1 se corresponde con un id de usuario en particular. Permite implementar una plantilla específica para el usuario indicado.
- html--user--%.html.twig. % se corresponde con un parámetro de URL numérico. Esta plantilla es válida para contenidos con URL de tipo user/%. Permite diferenciar de /user (sin parámetro).

page.html.twig

Veamos algunas sugerencias de plantilla para page.html.twig, que es la plantilla encargada del contenido del <body> de la página.

Página principal del sitio:

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  x page--front.html.twig
  * page--node.html.twig
  * page.html.twig
-->
```

- page--node.html.twig. 'node' se corresponde con el nombre de sistema de la vista de portada por defecto.
- page--front.html.twig. 'front' hace referencia a la página principal del sitio independientemente de la página o vista que se utilice como portada.

Página de nodo:

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--node--3.html.twig
  * page--node--%.html.twig
  * page--node.html.twig
  x page.html.twig
-->
```

- page--node.html.twig. 'node' se corresponde con el nombre de sistema de la vista de portada por defecto. También es la base de URL de un nodo /node.
- page--node--3.html.twig. 3 se corresponde con un id de nodo en particular. Permite implementar una plantilla específica para el nodo indicado.
- page--node--%.html.twig. % se corresponde con un parámetro de URL

numérico. Esta plantilla es válida para contenidos con URL de tipo `node/%`. Permite diferenciar de `/node` (sin parámetro).

Página de vista:

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--documents.html.twig
  x page.html.twig
-->
```

- `page--documents.html.twig`. 'documents' se corresponde con el nombre de sistema de la vista.

Página de formulario (webform):

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--contact.html.twig
  x page.html.twig
-->
```

- `page--contact.html.twig`. Plantilla específica para un formulario. 'contact' se corresponde con el nombre de sistema del formulario.

Página de panel (page manager):

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--homepage.html.twig
  x page.html.twig
-->
```

- `page--homepage.html.twig`. 'homepage' se corresponde con el nombre de sistema de la página.

Página de perfil de usuario:

```
<!-- THEME HOOK: 'page' -->
<!-- FILE NAME SUGGESTIONS:
  * page--user--1.html.twig
  * page--user--%.html.twig
  * page--user.html.twig
  x page.html.twig
-->
```

- `page--user.html.twig`. Plantilla para todas las páginas de perfil de usuario.
- `page--user--1.html.twig`. 1 se corresponde con un id de usuario en particular. Permite implementar una plantilla específica para el usuario indicado.
- `page--user--%.html.twig`. % se corresponde con un parámetro de URL numérico. Esta plantilla es válida para contenidos con URL de tipo `user/%`. Permite diferenciar de `/user` (sin parámetro).

layout.html.twig

Los módulos tipo Panels, Panelizer y Display Suite utilizan diferentes diseños o estructuras de página llamados **layouts** (dos columnas, dos columnas apiladas, tres columnas, etc.).

Los layouts sólo afectan a la región del contenido, y no a la página completa. Por ejemplo, si seleccionamos el layout "2 columnas" para la presentación de un nodo, los elementos del contenido (título, autor, campos, etc.) se podrán distribuir entre las regiones Left y Right. El resto de regiones del tema (columnas laterales, encabezado, pie, etc.) no se verán alterados, conteniendo los bloques que se hayan colocado desde la administración de bloques.

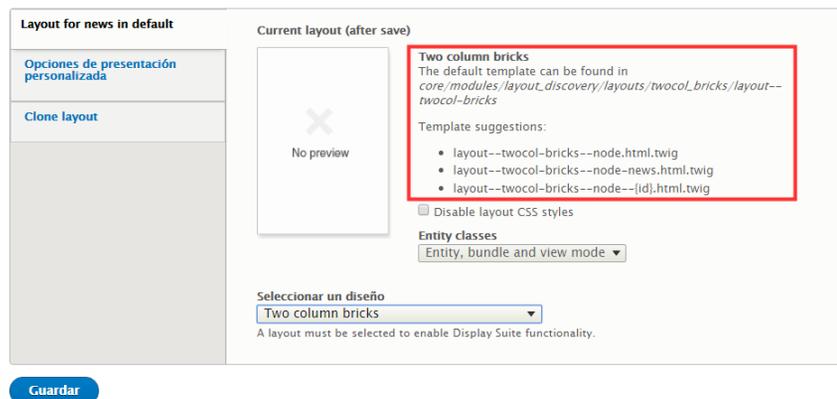
Layout discovery es el módulo del núcleo que se utiliza como base para cargar el layout seleccionado. Este módulo incluye algunos layouts básicos, que pueden complementarse con los layouts facilitados por otros módulos. La plantilla de base es **layout.html.twig**, que será reemplazada por la plantilla específica con la estructura de página seleccionada.

Como ejemplo vamos a utilizar Display Suite para modificar el layout del tipo de contenido News. Desde Administrar/Gestionar presentación del tipo de contenido, seleccionamos primero la presentación (Predeterminado, Contenido completo, Resumen, etc.) y luego uno de los diseños o layouts disponibles.

F7.4a

Layouts

Selección de layout en un tipo de contenido.



Las plantillas del módulo Layout discovery tienen el prefijo 'layout':

- layout--onecol.html.twig
- layout--threecol-25-50-25.html.twig
- layout--threecol-33-34-33.html.twig
- layout--twocol.html.twig
- layout--twocol-bricks.html.twig
- layout.html.twig

Las plantillas del módulo Display Suite tienen el prefijo 'ds':

- ds-2col-fluid.html.twig
- ds-2col-stacked-fluid.html.twig
- ds-2col-stacked.html.twig

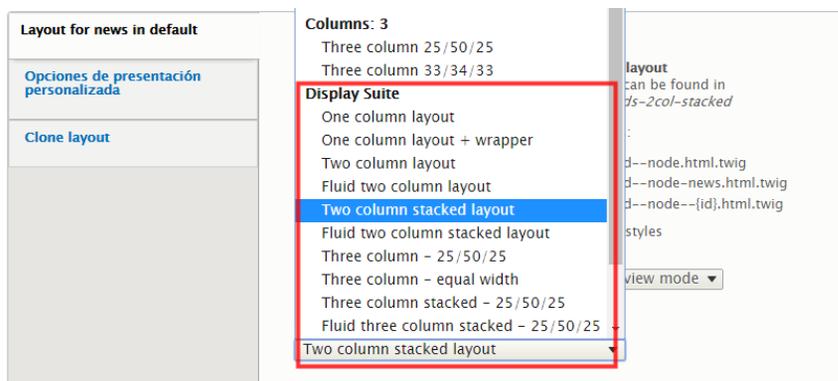
Página de nodo con layout:

Como ejemplo hemos configurado el tipo de contenido News (news) para que utilice el layout "Two column bricks" (del núcleo). Utilizando la depuración de plantillas vemos las sugerencias de plantillas alternativas:

```
<!-- THEME HOOK: 'layout__twocol_bricks' -->
<!-- FILE NAME SUGGESTIONS:
  * layout--twocol-bricks--node--51.html.twig
  * layout--twocol-bricks--node-news-full.html.twig
  * layout--twocol-bricks--node-news.html.twig
  * layout--twocol-bricks--node-full.html.twig
  * layout--twocol-bricks--node.html.twig
  x layout--twocol-bricks.html.twig
  * layout.html.twig
-->
```

- layout.html.twig. Plantilla base, si no aplicamos un layout específico.
- layout--twocol-bricks.html.twig. Plantilla que se utilizará por defecto para todas las entidades que utilicen este layout.
- layout--twocol-bricks--node.html.twig. Plantilla que se utilizará en todos los nodos que utilicen este layout, independientemente del tipo de contenido.
- layout--twocol-bricks--node-full.html.twig. Plantilla que se utilizará en todos los nodos que utilicen este layout, en el modo de presentación Contenido completo (full).
- layout--twocol-bricks--node-news.html.twig. Plantilla que se utilizará en todos los nodos de tipo news que utilicen este layout.
- layout--twocol-bricks--node-news-full.html.twig. Plantilla que se utilizará en todos los nodos de tipo news que utilicen este layout, en el modo de presentación Contenido completo (full).
- layout--twocol-bricks--node--51.html.twig. Plantilla que se utilizará exclusivamente en el nodo con Id 51.

Los layouts aportados por los diferentes módulos se muestran en el mismo selector, agrupados por el nombre del módulo. En la Figura se muestran los layouts disponibles gracias a Display Suite.



F7.4b

Layouts

Selección de layout en un tipo de contenido. Listado de layouts de Display Suite.

block.html.twig

Los bloques utilizan como base la plantilla `block.html.twig`. Algunas de las sugerencias disponibles son:

```
<!-- THEME HOOK: 'block' -->
<!-- FILE NAME SUGGESTIONS:
  * block--b-forcontu-search.html.twig
  x block--search-form-block.html.twig
  * block--search.html.twig
  * block.html.twig
-->
```

El patrón de base utilizado para las plantillas de bloques es:

- `block--module-delta.html.twig`. En `block--search-form-block.html.twig`, el módulo es `search` y el valor de `delta` `form-block`, identificando al bloque de búsqueda. Esta plantilla será exclusiva para ese bloque.
- `block--module.html.twig`. En `block--search.html.twig` sólo indicamos el nombre del módulo (`search`), así que la plantilla afectará a todos los bloques generados por el módulo.
- `block--b-forcontu-search.html.twig`. Esta variante de plantilla afectará a todos los bloques del módulo `search` cuando el tema activo sea `Bartik Forcontu` (`b_forcontu`).

Tipos de contenido

Los nodos para los que no se ha seleccionado un layout específico, utilizarán la plantilla `node.html.twig` como base. Si se ha seleccionado un layout, se utilizará la plantilla correspondiente al layout y se ignorará cualquier plantilla derivada de `node.html.twig`.

```
<!-- THEME HOOK: 'node' -->
<!-- FILE NAME SUGGESTIONS:
  * node--29--full.html.twig
  * node--29.html.twig
  * node--article--full.html.twig
  x node--article.html.twig
  * node--full.html.twig
  * node.html.twig
-->
```

- `node.html.twig`. Plantilla de base para todos los nodos.
- `node--full.html.twig`. Plantilla para todos los nodos en modo de presentación Contenido completo (full).
- `node--article.html.twig`. Plantilla para todos los nodos de tipo Artículo (article).
- `node--article--full.html.twig`. Plantilla para todos los nodos de tipo Artículo (article) en modo de presentación Contenido completo (full).
- `node--29.html.twig`. Plantilla específica para el nodo con Id 29.
- `node--29--full.html.twig`. Plantilla específica para el nodo con Id 29 en modo de presentación Contenido completo (full).

Campos

Cada campo de la entidad tiene su propia plantilla, con sus correspondientes sugerencias para hacerla más específica. Como ejemplo veamos las variaciones para el campo `field_image` añadido al tipo de contenido Artículo:

```
<!-- THEME HOOK: 'field' -->
<!-- FILE NAME SUGGESTIONS:
  * field--node--field-image--article.html.twig
  * field--node--field-image.html.twig
  * field--node--article.html.twig
  * field--field-image.html.twig
  * field--image.html.twig
  x field.html.twig
-->
```

- `field.html.twig`. Plantilla para base para todos los campos.
- `field--image.html.twig`. Plantilla para todos los campos de tipo Imagen (image).
- `field--field-image.html.twig`. Plantilla para el campo 'field_image', independientemente de la entidad donde esté asignado.
- `field--node--article.html.twig`. Plantilla para cualquier campo de los nodos de tipo artículo.
- `field--node--field-image.html.twig`. Plantilla para el campo 'field_image' dentro de nodos.
- `field--node--field-image--article.html.twig`. Plantilla para el campo 'field_image' dentro de nodos de tipo artículo.

Vistas

Terminamos este apartado analizando las sugerencias de plantillas disponibles para las vistas. En Drupal 7 las sugerencias de plantillas y la plantilla utilizada actualmente podía consultarse directamente desde la interfaz de administración de la vista. Esta opción se ha eliminado en Drupal 8, pero podemos consultar las sugerencias de plantilla desde la depuración de plantillas, como hemos hecho con el resto de plantillas del sitio.

Nota: Si no se muestran las sugerencias de plantillas de vistas en tu sitio, revisa esta issue y aplica el parche adecuado:

<https://www.drupal.org/project/drupal/issues/2923634>

Para ilustrar este caso hemos creado la vista Artículos, que muestra una tabla con los contenidos de tipo Artículo. Para cada contenido muestra el título, la fecha de creación y el autor.

F7.4c**Plantillas en vistas**

Vista de ejemplo en formato tabla.

Artículos

Título	Fecha	Escrito por
Macto Nunc Paratus Ymo	Lun, 08/13/2018 - 04:24	admin
Interdico Lobortis	Mar, 07/10/2018 - 11:50	admin
Hos Premo Ullamcorper	Mar, 06/26/2018 - 20:19	admin
Abdo Gemino Nutus Usitas	Vie, 06/22/2018 - 20:09	admin
Et Interdico Pagus Paratus	Dom, 06/17/2018 - 08:55	Anónimo
Humo	Vie, 06/15/2018 - 11:46	admin
Augue In Lobortis Molior	Jue, 06/14/2018 - 15:58	Anónimo
Jumentum Neque Paulatim	Mié, 06/13/2018 - 20:38	admin
Decet Ullamcorper	Mar, 05/29/2018 - 04:12	admin
Aliquip	Jue, 05/03/2018 - 11:02	admin

1 2 [Next >](#) [Last >](#)

La plantilla de nivel más alto en la vista es `views-view.html.twig`. Como veremos, esta plantilla engloba a otras vistas.

```
<!-- THEME HOOK: 'views_view' -->
<!-- FILE NAME SUGGESTIONS:
 * views-view--articulos--page-1.html.twig
 * views-view--page-1.html.twig
 * views-view--default.html.twig
 * views-view--articulos--page.html.twig
 * views-view--page.html.twig
 * views-view--articulos.html.twig
 x views-view.html.twig
-->
```

- `views-view.html.twig`. Plantilla base para todas las vistas.
- `views-view--articulos.html.twig`. Plantilla base para la vista 'articulos'.
- `views-view--page.html.twig`. Plantilla base para todas las vistas cuya presentación sea de tipo página (page).
- `views-view--articulos--page.html.twig`. Plantilla base para las presentaciones de tipo Page de la vista 'articulos'.
- `views-view--default.html.twig`. Plantilla base para todas las vistas en la presentación por defecto.
- `views-view--page-1.html.twig`. Plantilla base para la presentación con nombre interno 'page-1'.

El siguiente nivel de la plantilla tiene que ver con el formato. Los nombres de sistema de los formatos de vista son:

- Tabla (table).
- Cuadrícula (grid).
- Lista sin formato (unformatted).
- Lista HTML (list).

La plantilla utilizada tendrá la nomenclatura:

- `views-view-formato.html.twig`

En nuestra vista de ejemplo, el formato utilizado es table y, por tanto, la plantilla base para aplicar este formato será:

- views-view-table.html.twig.

```
<!-- THEME HOOK: 'views_view_table' -->
<!-- FILE NAME SUGGESTIONS:
  * views-view-table--articulos--page-1.html.twig
  * views-view-table--page-1.html.twig
  * views-view-table--default.html.twig
  * views-view-table--articulos--page.html.twig
  * views-view-table--page.html.twig
  * views-view-table--articulos.html.twig
  x views-view-table.html.twig
-->
```

Las variantes nos permiten discriminar según el nombre de la vista y el tipo de presentación, siendo la más específica:

- views-view-table--articulos--page-1.html.twig. Plantilla específica para el formato table de la presentación 'page-1' en la vista 'articulos'.

Si la vista muestra campos, usará la plantilla 'views-view-field.html.twig' como base. A continuación, se muestran las variantes disponibles para el campo 'title' dentro de la vista:

```
<!-- THEME HOOK: 'views_view_field' -->
<!-- FILE NAME SUGGESTIONS:
  * views-view-field--articulos--page-1--title.html.twig
  * views-view-field--articulos--page-1.html.twig
  * views-view-field--page-1--title.html.twig
  * views-view-field--page-1.html.twig
  * views-view-field--articulos--page--title.html.twig
  * views-view-field--articulos--page.html.twig
  * views-view-field--page--title.html.twig
  * views-view-field--page.html.twig
  * views-view-field--articulos--title.html.twig
  * views-view-field--articulos.html.twig
  * views-view-field--title.html.twig
  x views-view-field.html.twig
-->
```

- views-view-field--articulos--page-1--title.html.twig. Plantilla exclusiva para el campo 'title' dentro de la presentación 'page-1' de la vista 'articulos'.

Además de las plantillas para los resultados de la vista, también podremos modificar otras plantillas para:

- views-exposed-form.html.twig. Muestra el formulario de filtros expuestos.
- pager.html.twig. Paginador.

Para consultar información adicional sobre las plantillas en vistas, visita este enlace:

https://api.drupal.org/api/drupal/core%21modules%21views%21views.theme.inc/group/views_templates/8.5.x

Twig desde la interfaz de vistas

En el curso de Experto en Drupal 8 Site Building ya vimos cómo utilizar patrones de reemplazo dentro de las vistas, reescribir la salida de un campo y definir campos generales de tipo Texto personalizado. Recordamos en este punto que estos campos aceptan tanto HTML como Twig, con lo que podemos aplicar las funciones y filtros de Twig estudiados en este nivel.

En la Figura [F7.4d] usamos el nombre de usuario `{{ uid }}` pero aplicando los filtros:

- `|striptags`. Elimina el HTML que pudiera contener el campo `{{ uid }}`. Como este campo es un enlace, eliminaría el enlace y dejaría únicamente el nombre del usuario.
- `|upper`. Pasa a mayúsculas el nombre del usuario.

F7.4d

Twig en vistas

Utilización de Twig desde la interfaz de Vistas.

Texto

Este artículo fue creado por `{{ uid|striptags|upper }}`

The text to display for this field. You may include HTML or [Twig](#). You may enter data from this view as per the "Replacement patterns" below.

▼ PATRONES DE REEMPLAZO

The following replacement tokens are available for this field. Note that due to rendering order, you cannot use fields that come after this field; if you need a field not listed here, rearrange your fields.

- `{{ title }}` ==
- `{{ created }}` ==
- `{{ uid }}` ==
- `{{ nothing }}` == Texto personalizado

El resultado se muestra en la Figura [F7.4e]:

F7.4e

Twig en vistas

Utilización de Twig desde la interfaz de Vistas.

Abdo Gemino Nutus Usitas	Vie, 06/22/2018 - 20:09	admin	Este artículo fue creado por ADMIN
Et Interdico Pagus Paratus	Dom, 06/17/2018 - 08:55	Anónimo	Este artículo fue creado por ANÓNIMO

Módulos adicionales: Advanced CSS/JS Aggregation

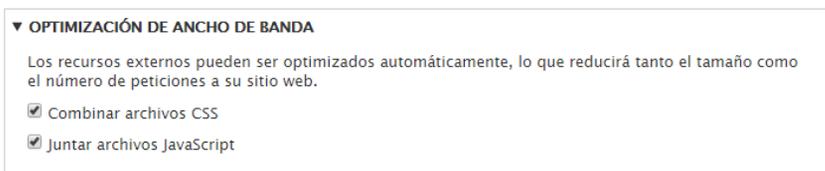
7.5

Combinar archivos CSS

Ya vimos en el apartado 3.3 que para depurar los estilos CSS desde el navegador, era necesario desactivar la opción Combinar archivos CSS desde:

Administración ⇒ Configuración ⇒ Desarrollo ⇒ Rendimiento

Cuando pasamos el sitio a producción tenemos que asegurarnos de que esta opción vuelve a estar activa.



F7.5a
Combinar archivos CSS
 Opción disponible dentro de Rendimiento.

Advanced CSS/JS Aggregation

El módulo **Advanced CSS/JS Aggregation** añade opciones para mejorar el rendimiento de las librerías CSS y JS. El módulo está disponible en:

<https://www.drupal.org/project/advagg>

Entre los submódulos incluidos, destacamos los siguientes relacionados con el tratamiento de CSS:

- **Advanced CSS/JS Aggregation.** Módulo base. Aporta opciones de compresión de CSS.
- **AdvAgg Bundler.** Permite crear diferentes archivos CSS y separarlos según la configuración especificada.
- **AdvAgg CDN.** Usa un CDN para cargar las librerías CSS y JS. Un CDN es una red de distribución de contenidos (CDN, content delivery network en inglés). Básicamente se trata de externalizar ciertos archivos, generalmente estáticos, para maximizar el ancho de banda y que la página cargue más rápido.
- **AdvAgg CSS/JS Validator.** Comprueba si los archivos CSS y JS tienen errores sintácticos.
- **AdvAgg Minify CSS.** Permite seleccionar el método de reducción de los archivos CSS (núcleo o YUI Compressor).

Una vez instalados los módulos, accede a la configuración desde **[F7.5b]**:

Administración ⇒ Configuración ⇒ Desarrollo
⇒ Rendimiento [Advanced CSS/JS Aggregation]

F7.5b
Advanced CSS/JS
Aggregation
Resultado f

AdvAgg: Configuration ☆

Rendimiento | Rendimiento | Advanced CSS/JS Aggregation | Advanced CSS/JS Aggregation Asset Validation

Configuración | Información | Operations | Bundler | CSS Minification

Inicio » Administración » Configuración » Development » Rendimiento

GLOBAL OPTIONS

Enable advanced aggregation
Uncheck this box to temporarily disable AdvAgg functionality.

Use DNS Prefetch for external CSS/JS.
Start the DNS lookup for external CSS and JavaScript files as soon as possible.

SERVER CONFIG
Cache-Control: Immutable

Your server can send Cache-Control: immutable header for all optimized files. This should improve performance for some users. Current [browser support](#)

With Nginx, AdvAgg can't set the headers for optimized files in a performant manner. However, you can easily do so in your server config. See the [manual for instructions](#).

AdvAgg Cache Settings

Development

Bajo

Normal

Alto

No performance data yet but most use cases will probably want to use the Normal cache mode.

F7.5c
Advanced CSS/JS
Aggregation
Resultado f

▼ COMPRESSION OPTIONS

AdvAgg can't configure your Nginx server to automatically serve compressed assets. See the [AdvAgg manual](#) for instructions on manually doing so.

Gzip CSS assets
This should be enabled unless you are experiencing corrupted compressed asset files.

Gzip JavaScript assets
This should be enabled unless you are experiencing corrupted compressed asset files.

Brotli compress CSS assets
Brotli compression is not available on your server. See [PHP Brotli](#) page for more information.

Brotli compress JavaScript assets
Brotli compression is not available on your server. See [PHP Brotli](#) page for more information.

▼ CSS OPTIONS

Combine CSS files by using media queries
Will combine more CSS files together because different CSS media types can be used in the same file by using media queries. Use cores grouping logic needs to be unchecked in order for this to work. Also noted is that due to an issue with IE9, compatibility mode is forced off if this is enabled.

Fix improperly set type
If type is external but does not start with http, https, or // change it to be type file. If type is file but it starts with http, https, or // change type to be external. Note that if this is causing issues, odds are you have a double slash when there should be a single; see [this issue](#)

Do not change external to file if on same host.
If a CSS file is set as external and is on the same hosts do not convert to file.

Desde la subpestaña Configuración podemos aplicar, entre otras, las siguientes opciones de configuración general:

- Activar o desactivar Advanced Aggregation.
- Comprimir con Gzip

Desde la pestaña Advanced CSS/JS Aggregation Asset Validation podemos comprobar si los archivos CSS son correctos a nivel sintáctico **[F7.5d]**:

AdvAgg: CSS Validator using CSSLint ☆

Rendimiento
Rendimiento
Advanced CSS/JS Aggregation
Advanced CSS/JS Aggregation Asset Validation

CSSLint
W3.org CSS Validator
JSHint

[Inicio](#) » [Administración](#) » [Configuración](#) » [Development](#) » [Rendimiento](#) » [AdvAgg: Configuration](#)

▼ **THEMES**

Directory: themes

Check directory and all subdirectories

themes/mayo/css/style.css

- No errors

themes/mayo/css/messages.css

- No errors

themes/mayo/css/maintenance-page.css

- No errors

themes/mayo/css/colors.css

- No errors

F7.5d
Advanced CSS/JS
Aggregation
 Resultado f

7.6 Módulos adicionales: Twig Vardumper

El módulo Twig Vardumper aporta una alternativa a las funciones de depuración de variables, mejorando la presentación de la información y el rendimiento.

https://www.drupal.org/project/twig_vardumper

Para instalar este módulo, accede por SSH y ejecuta estos comandos desde la carpeta del sitio (por ejemplo, fe/fe1):

- a) Descarga del módulo junto con las librerías necesarias vía composer:

```
composer require drupal/twig_vardumper
```

- b) Instalación del módulo:

```
drush en twig_vardumper
```

Una vez instalado, podremos utilizar las siguientes alternativas:

- Núcleo:


```
{{ dump() }}
```

```
{{ dump(variable_name) }}
```
- Módulo Kint (submódulo de Devel):


```
{{ kint() }}
```

```
{{ kint(variable_name) }}
```
- Módulo Twig Vardumper (utiliza symfony/var-dumper):


```
{{ vardumper() }}
```

```
{{ vardumper(variable_name) }}
```

En todos los casos, si no se indica un nombre de variable, se mostrarán todas las variables disponibles en la plantilla.

F7.6a Twig Vardumper

Arriba, salida con kint().
Abajo, salida con vardumper().

```

classes array (7)
  string (4) "node"
  string (18) "node--type-article"
  string (14) "node--promoted"
  string (0) ""
  string (0) ""
  string (20) "node--view-mode-full"
  string (8) "clearfix"

array:7 [▼]
  0 => "node"
  1 => "node--type-article"
  2 => "node--promoted"
  3 => ""
  4 => ""
  5 => "node--view-mode-full"
  6 => "clearfix"
  ]
  
```

En la siguiente Figura se muestra la salida con vardumper() mostrando todas las variables disponibles en una plantilla:

```
array:30 [▼
  "elements" => array:20 [▶]
  "theme_hook_original" => "node"
  "attributes" => Attribute {#2689 ▶}
  "title_attributes" => Attribute {#256 ▶}
  "content_attributes" => Attribute {#3511 ▶}
  "title_prefix" => []
  "title_suffix" => array:3 [▶]
  "db_is_active" => true
  "is_admin" => true
  "logged_in" => true
  "user" => AccountProxy {#464 ▶}
  "directory" => "themes/b_forcontu"
  "view_mode" => "full"
  "teaser" => false
  "node" => Node {#1019 ▶}
  "date" => Markup {#418 ▶}
  "author_name" => Markup {#154 ▶}
  "url" => "/node/23"
  "label" => array:21 [▶]
  "page" => true
  "content" => array:5 [▼
    "field_image" => array:17 [▶]
    "body" => array:17 [▶]
    "field_tags" => array:19 [▶]
    "links" => array:2 [▶]
    "comment" => array:17 [▶]
  ]
  "author_attributes" => Attribute {#2468 ▶}
  "display_submitted" => true
  "author_picture" => array:6 [▶]
  "#cache" => array:1 [▶]
  "metadata" => Markup {#262 ▶}
  "twig_debug" => true
  "theme_hook_suggestions" => array:5 [▶]
  "theme_hook_suggestion" => "node"
  "classes" => array:7 [▶]
]
```

F7.6b
Twig Vardumper
 Salida con vardumper().

7.7 Módulos adicionales: Twig Tweak

El módulo Twig Tweak añade al sistema filtros y funciones adicionales para usar directamente dentro de las plantillas twig.

https://www.drupal.org/project/twig_tweak

Puedes consultar todas las funciones y filtros disponibles en esta URL:

<https://www.drupal.org/docs/8/modules/twig-tweak/cheat-sheet-8x-2x>

Las funciones y filtros están definidos en el archivo del módulo:

```
/modules/twig_tweak/src/TwigExtension.php
```

Si tienes dudas, puedes consultar este archivo para ver cómo trabaja cada función internamente, a nivel de Back-End.

Funciones

drupal_view()

Permite embeber una vista desde una plantilla. Internamente utiliza la función de la API **views_embed_view()**:

https://api.drupal.org/api/drupal/core!modules!views!views.module/function/views_embed_view/8.6.x

Además del nombre de sistema y display de la vista, se pueden pasar parámetros adicionales, que serán equivalentes a los parámetros que acepta la vista.

```
{{ drupal_view('who_s_new', 'block_1') }}
{{ drupal_view('custom_view', 'page_1', 3) }}
```

drupal_view_result()

Es similar a la función anterior, pero sólo muestra los resultados de la vista. También permite el paso de parámetros. Internamente utiliza la función **views_get_view_result()**:

https://api.drupal.org/api/drupal/core!modules!views!views.module/function/views_get_view_result/8.6.x

```
{{ drupal_view_result('who_s_new', 'block_1') }}
```

drupal_block()

Representa un bloque, sin necesidad de que el bloque esté activo en la administración de Bloques.

El primer parámetro es el nombre de sistema del bloque. El segundo parámetro es un array de configuración del bloque. El tercer parámetro (wrapper) es un booleano para indicar si se utilizará o no la plantilla del bloque para renderizar el elemento.

```

{{ drupal_block('system_branding_block', {label: 'Branding',
use_site_name: false}) }}

{# Muestra el bloque sin el HTML encapsulador, saltándose la
plantilla block.html.twig. #}
{{ drupal_block('system_branding_block', wrapper=false) }}

```

drupal_region()

Imprime una región completa. Como primer parámetro indicaremos el nombre de sistema de la región. El segundo parámetro es opcional y permite indicar el nombre del tema donde se cargará la región. Si estamos trabajando dentro de la plantilla de un tema no es necesario indicar el nombre del tema.

```

{{ drupal_region('sidebar_first') }}

```

drupal_entity()

Renderiza cualquier entidad. Los parámetros de la función son:

- Tipo de entidad ('node', 'block_content', 'user', etc.)
- Id de la entidad. Si no se indica un valor, se carga la entidad teniendo en cuenta la URL.
- Modo de presentación ('full', 'teaser', etc.)
- Código de idioma. Si no se especifica se tendrá en cuenta el idioma actual.
- Comprobar acceso. Si TRUE (por defecto), se comprobará si el usuario tiene acceso para ver la entidad.

```

{{ drupal_entity('block_content', 1) }}

{# Carga el nodo con ID 3, en modo resumen #}
{{ drupal_entity('node', 3, 'teaser') }}

{# Carga el resumen del nodo actual según la ruta. #}
{{ drupal_entity('node', null, 'teaser') }}

```

drupal_entity_form()

Permite renderizar un formulario de creación o edición de una entidad, y procesarlo. Los parámetros de la función son:

- Tipo de entidad ('node', 'block_content', 'user', etc.)
- Id de la entidad. Si se indica un valor, se editará la entidad. Si no se indica un valor, se creará una nueva entidad.
- Modo de presentación del formulario (opcional)
- Valores. Se puede pasar un array con valores (clave/valor) que se añadirán a los elementos del formulario. La clave se debe corresponder con el nombre de sistema del elemento de formulario. Si se va a crear una nueva entidad, se debe indicar el tipo o bundle (por ejemplo, el tipo de contenido al crear un nodo).
- Comprobar acceso. Si TRUE (por defecto), se comprobará si el usuario tiene acceso para ver la entidad.

```
{# Formulario de edición del nodo con Id 3 #}
{{ drupal_entity_form('node', 3) }}

{# Formulario de creación de un nodo. Se debe especificar el tipo
de bundle en valores #}
{{ drupal_entity_form('node', values={type: 'article'}) }}
```

drupal_field()

Renderiza un único campo de la entidad especificada. Los parámetros son similares a los de **drupal_entity()**, pero anteponiendo el nombre del campo.

```
{{ drupal_field('field_image', 'node', 1) }}
```

drupal_menu()

Renderiza un menú, según los siguientes parámetros de configuración:

- Nombre del menú.
- Nivel inicial.
- Número máximo de niveles a mostrar.
- Indica si se deben expandir todos los enlaces de menú (valor por defecto, FALSE).

```
{# Muestra el menú 'main' completo #}
{{ drupal_menu('main') }}

{# Muestra el menú de administración especificando nivel inicial y
nivel de profundidad #}
{{ drupal_menu('admin', 2, 3) }}
```

drupal_form()

Dado un Form ID, renderiza el formulario y lo procesa.

```
{# Formulario de búsqueda #}
{{ drupal_form('Drupal\\search\\Form\\SearchBlockForm') }}
```

drupal_image()

Renderiza una imagen. Se pueden aplicar estilos de imagen de Drupal y atributos CSS adicionales. Los parámetros son:

- El primer parámetro referencia la imagen a través de su fid, uuid o uri.
- Estilo de imagen.
- Atributos adicionales.
- Responsive. Si se asigna el valor TRUE, el estilo de imagen se buscará entre los estilos de imagen responsive definidos en la interfaz. Si el valor es FALSE, el estilo de imagen deberá estar definido entre los estilos "normales".
- Comprobar acceso. Si TRUE (por defecto), se comprobará si el usuario tiene acceso para ver la entidad.

```
{# Imagen con estilo large #}
{{ drupal_image('public://sample.jpg', 'large') }}
```

```
{# Imagen con estilo responsive. #}
{{ drupal_image('public://sample.jpg', 'wide', responsive=true) }}
```

drupal_token()

Sustituye un token por su valor correspondiente. Por ejemplo, 'node:title' imprimirá el título del nodo, siempre teniendo en cuenta el contexto actual.

```
{# Nombre del sitio #}
{{ drupal_token('site:name') }}
```

```
{# Tokens dentro de un contexto #}
{{ drupal_token('node:title', {node: node}) }}
```

```
{# Reemplazo de varios tokens dentro de una cadena, con el filtro
|token_replace #}
{{ '<h1>[site:name]</h1><div>[site:slogan]</div>'|token_replace }}
```

drupal_config()

Imprime el valor de una variable de configuración, especificando el nombre del objeto de configuración y la variable.

```
{# Imprime el nombre del sitio #}
{{ drupal_config('system.site', 'name') }}
```

drupal_dump(), dd()

Imprime información de depuración de una variable. `dd()` es un alias de `drupal_dump()`. Si no se especifica el nombre de una variable, se mostrarán todas las variables disponibles en la plantilla:

```
{{ drupal_dump(var) }}

{{ dd(var) }}
```

drupal_title()

Muestra el título de la página actual.

```
{{ drupal_title() }}
```

drupal_url(), drupal_link()

Funciones para generar la URL y enlace de una ruta interna.

```
{# Genera una URL absoluta #}
{{ drupal_url('node/1', {absolute: true}) }}

{# Genera un enlace <a> a un nodo #}
{{ drupal_link('Example'|t, 'node/1') }}
```

drupal_messages()

Muestra los mensajes de estado de Drupal.

```
{{ drupal_messages() }}
```

drupal_breadcrumb()

Muestra las migas de pan para la página actual:

```
{{ drupal_breadcrumb() }}
```

Filtros

|php

Aunque es una muy mala práctica y está **totalmente desaconsejado**, también incluye un filtro PHP (|php), lo que permite añadir cualquier código PHP directamente en la plantilla.

El filtro está activado por defecto. Para activarlo es necesario añadir la siguiente variable de configuración en el archivo settings.php:

```
$settings['twig_tweak_enable_php_filter'] = TRUE;
```

```
{# Devuelve el año actual usando la función date() de PHP #}
{{ 'return date("Y");'|php }}
```

|preg_replace()

Utiliza la función preg_replace() de PHP para realizar un reemplazo de cadena usando una expresión regular. El orden de los parámetros es: texto (a la izquierda del filtro), patrón de búsqueda y texto de reemplazo.

<http://www.php.net/manual/es/function.preg-replace.php>

```
{{ 'Drupal - community plumbing!'|preg_replace('/(Drupal)/',
'<b>$1</b>') }}
```

|image_style()

Devuelve la URL de una imagen aplicando el estilo de imagen indicado.

```
{{ 'public://images/sample.jpg'|image_style('thumbnail') }}
```

|transliterate

Realiza una transliteración de la cadena a ASCII. Por ejemplo, á será sustituido por a. Esta transformación es similar a la que realiza Pathauto al generar URLs automáticamente.

```
{# Devuelve 'animo' #}
{{ 'ánimo'|transliterate }}
```

|check_markup()

Filtra un texto según el formato de texto indicado. Los formatos de texto se pueden consultar en Configuración, Autoría del contenido, Formato y editores de texto.

```
{# Filtra el texto aplicando el filtro HTML Restringido #}
{{ '<b>bold</b>'
<strong>strong</strong>|check_markup('restricted_html') }}
```

|truncate()

Corta un texto a un número especificado de caracteres. El primer parámetro indica el número de caracteres. El segundo parámetro a TRUE indica que no se cortarán palabras. Un tercer parámetro permite añadir ... al final de la cadena.

```
{{ 'Some long text'|truncate(10, true, true) }}
```

|with, |without

El filtro **without** está incluido en el núcleo de Drupal y permite eliminar un elemento de un array. Realmente lo que hace es una copia del array sin el elemento indicado.

El filtro **with** es similar al anterior, pero añadiendo un nuevo elemento al array.

```
{{ item|with('#title', 'Example') }}
```

|view

Devuelve la representación de una entidad o un campo.

```
{# Presentación de un nodo #}
{{ node|view }}
```



```
{# Presentación de un campo #}
{{ node.field_image|view }}
```

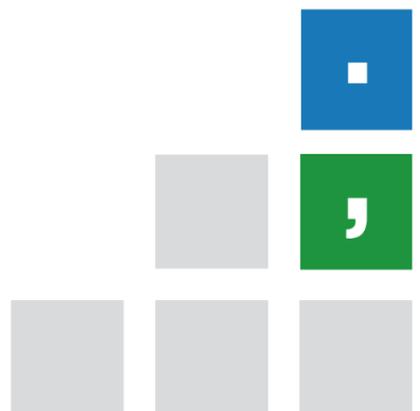
Fran Gil

Front-End Development I

8

Diseño responsive

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

8 Diseño responsive

Llamamos diseño responsive (también conocido como adaptativo o responsivo) al diseño web que permite que una página web se visualice correctamente en diferentes dispositivos (ordenadores de sobremesa o portátiles, tablets, teléfonos móviles, etc.), adaptándose automáticamente a las diferentes resoluciones.

En esta unidad veremos:

- Cómo crear un tema responsive a partir de un tema base.
- Tipos de layouts CSS responsive.
- Cómo definir Media queries y breakpoints para crear reglas CSS condicionales.
- Cómo crear estilos de imágenes responsive y aplicarlos dentro de Drupal.

Comparativa D8/D7

Los temas generales relacionados con HTML y CSS, como la definición de layouts y media queries son de aplicación tanto en Drupal 8 como Drupal 7. El resto de temas tratados son específicos para Drupal 8.

Contenidos de la Unidad

- 8.1 Introducción al diseño responsive
- 8.2 Creación de un tema responsive
- 8.3 Layouts responsive en CSS
- 8.4 Media Queries
- 8.5 Breakpoints en Drupal
- 8.6 Imágenes responsive



8.1 Introducción al diseño responsive

Qué es el diseño responsive

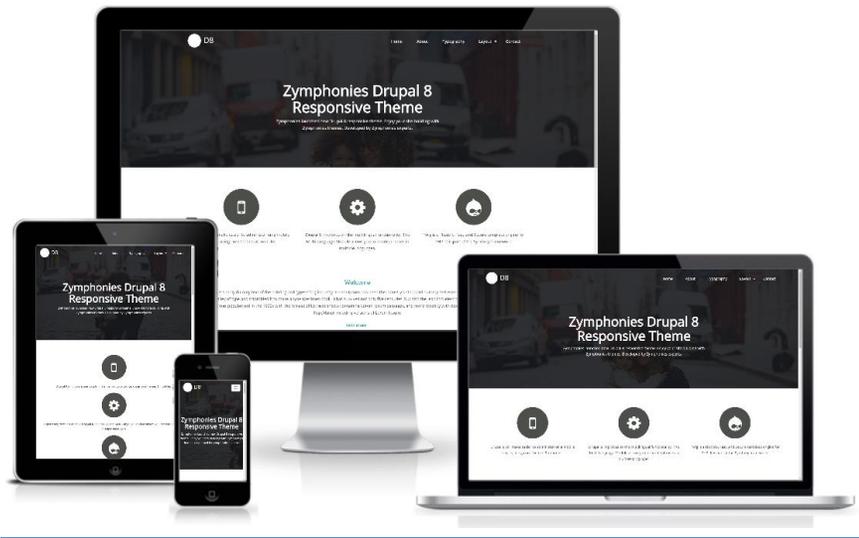
Llamamos diseño responsive (también conocido como adaptativo o responsivo) al diseño web que permite que una página web se visualice correctamente en diferentes dispositivos (ordenadores de sobremesa o portátiles, tablets, teléfonos móviles, etc.), adaptándose automáticamente a las diferentes resoluciones.

Utilizando HTML y CSS indicaremos cómo deben comportarse los distintos elementos de la página, pudiendo cambiar el tamaño, mover o incluso esconder elementos en función del dispositivo.

F8.1a

Diseño responsive

Ejemplo de diseño responsive generado desde <http://ami.responsivedesign.is>, utilizando el tema Drupal8 Zymphonies Theme, disponible en el repositorio oficial de Drupal.



Qué es el Viewport

Ya hablamos del Viewport en el apartado 5.2, cuando hablamos de las etiquetas <meta> en HTML.

```
<meta name="viewport"... />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

La metaetiqueta viewport fue introducida en HTML5 para facilitar el diseño responsive. "viewport" hace referencia a la parte visible de la página. A través de esta etiqueta podemos indicarle al navegador cómo debe escalar la página en función del dispositivo.

- 'width=device-width', indica que el ancho de la página se debe adaptar al ancho del dispositivo.

- 'initial-scale=1.0', indica el nivel de zoom inicial cuando el navegador carga la página. El usuario puede posteriormente cambiar el zoom de la página.

Es una etiqueta que se añade por defecto y generalmente no necesitamos modificar sus valores.

En general podemos decir que el objetivo del diseño responsive es **evitar el scroll horizontal**. Los usuarios sí están acostumbrados al scroll vertical, pero el scroll horizontal se considera una mala experiencia de navegación. La forma de lograr esto es dejando que el contenido se adapte al tamaño del Viewport, evitando elementos con anchos fijos y el diseño específico para una única resolución de pantalla.

8.2 Creación de un tema responsive

En este apartado vamos a crear un tema responsive para poner en práctica los contenidos de la unidad. Utilizaremos como tema base **AdaptiveTheme**, aunque en el repositorio de Drupal encontrarás otros temas base responsive como Bootstrap, Basic o Zen.

En el caso concreto de AdaptiveTheme, el tema incluye un Generador de temas, que tendremos que instalar para crear nuestro subtema. Veamos la creación del tema paso a paso.

Paso 1. Instalación del tema AdaptiveTheme

El tema AdaptiveTheme está disponible en:

<https://www.drupal.org/project/adaptivetheme>

Lo instalaremos desde:

Administración ⇒ **Apariencia [Instalar nuevo tema]**

El tema **AT Core** no aparecerá en el listado, ya que se trata de un tema oculto que solo se utilizará como base para el tema que crearemos.

A continuación, instala la última versión del módulo **AT Tools**, disponible en:

https://www.drupal.org/project/at_tools

Activa los módulos **AT Tools** y **AT Theme Generator**, incluidos en AT Tools.

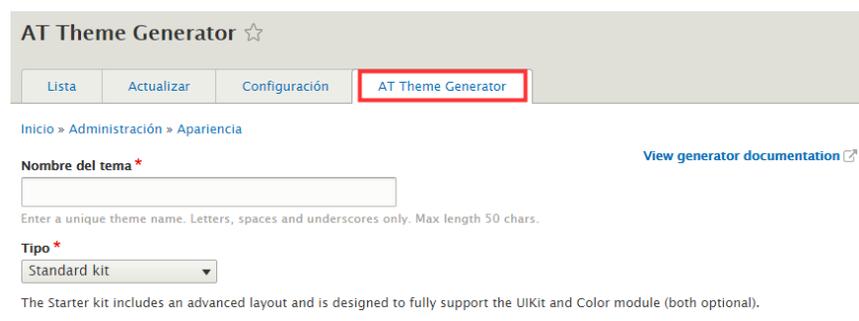
Paso 2. Crear el subtema con AT Theme Generator

En Apariencia, encontrarás una nueva pestaña llamada AT Theme Generator, que utilizaremos para generar nuestro tema **[F8.2a]**. Si no se muestra la pestaña, vacía la caché del sitio o accede directamente a la URL: `/admin/appearance/at_generate`.

URL Apariencia
/admin/appearance

F8.2a Subtema con AdaptiveTheme

Generador de subtema.



Completaremos la información indicando el nombre del tema, que generalmente será el nombre del proyecto. En nuestro ejemplo escribiremos **AT Forcontu**, con nombre de sistema **at_forcontu**. El tipo de tema a generar será **Standard kit (juego estándar) [F8.2b]**.

Activa también las opciones Templates (plantillas), UI Kit (juego UI) y Color Module (módulo de color). Por último, haz clic en Enviar para generar el tema.

AT Theme Generator ☆

List
Actualizar
Ajustes
AT Theme Generator

[Inicio](#) » [Administración](#) » [Apariencia](#) [View generator documentation](#)

Nombre del tema *

Nombre de sistema: at_forcontu [\[Editar\]](#)

Enter a unique theme name. Letters, spaces and underscores only. Max length 50 chars.

Type *

juego estándar

The Starter kit includes an advanced layout and is designed to fully support the UIKit and Color module (both optional).

OPCIONES

- SCSS (SASS/Grunt)

Includes all SCSS partials for the UIKit (user facing styles), Layouts and Layout Plugin partials, Grunt tasks, and Bower/NPM package management. If you have no use for SCSS un-check this option.
- Módulo de color

Provides basic Color module support so you can modify the color of your theme in theme settings.
- Block Config

Include configuration for blocks. Un-check this setting if you want your theme to inherit the default themes block configuration.
- Plantillas

Include copies of Drupals front end twig templates (page.html.twig is always included regardless of this setting).
- theme-settings.php

Include a theme-settings.php file. Includes skeleton code for the form alter, custom validation and submit functions.

Layout type

- Flexbox
- Floats

All modern browsers including IE11 are compatible with Adaptivetheme's Flexbox layouts. If you need to support legacy browsers such as IE8/9 select Floats. See caniuse.com for more info.

Descripción

Descriptions are used on the Appearance list page.

Cadena de versión

Numbers, hyphens and periods only. E.g. 8.x-1.0

Save to

Themes directory

Select the directory to save the new theme in. If you select "Themes" your new theme will be saved to the same directory as adaptivetheme.

Submit

F8.2b
Subtema con AdaptiveTheme
 Generador de subtema.

Se mostrará un mensaje indicando que el tema ha sido creado, estando disponible en la carpeta /themes/at_forcontu **[F8.2c]**.

✓ A new theme **AT Forcontu**, with then machine name: `at_forcontu`, has been generated.

You can find your theme here: `themes/at_forcontu`

Click the List tab to view the themes list and enable your new theme.

F8.2c
Subtema con AdaptiveTheme
 Tema AT Forcontu generado.

En la carpeta /themes estarán disponibles los dos temas, el tema generado y el tema base del que depende. En ningún caso debemos eliminar la carpeta del tema

/adaptivetheme [F8.2d].

F8.2d

Subtema con AdaptiveTheme

Localización del tema generado.

```
>themes/
  >adaptivetheme/
    >at_forcontu/
```

Si editamos el archivo de definición del tema, **at_forcontu.info.yml**, veremos una referencia al subtema [F8.2e]:

F8.2e

Subtema con AdaptiveTheme

Archivo de información del tema generado.

```
name: 'AT Forcontu'
version: 8.x-1.0
type: theme
'base theme': at_core
'base theme original': 8.x-3.1
'subtheme type': adaptive_subtheme
layout: page-layout
description: 'Subtema a partir de AdaptiveTheme <br>Base: at_core (8.x-3.1) <br>Generated: Sáb 15th, Sep, 2018 - 19:13'
core: 8.x
regions:
...
```

Paso 3. Instalar el nuevo tema AT Forcontu

Desde el listado de temas desactivados, instalaremos el nuevo tema creado, AT Forcontu (Instalar y seleccionar por defecto). También podemos **desinstalar el módulo AT Theme Generator**, ya que solo se utiliza durante la generación del subtema. Accediendo a la página principal del sitio veremos la apariencia del nuevo tema instalado [F8.2f].

F8.2f

Subtema con AdaptiveTheme

Activación del nuevo tema generado.

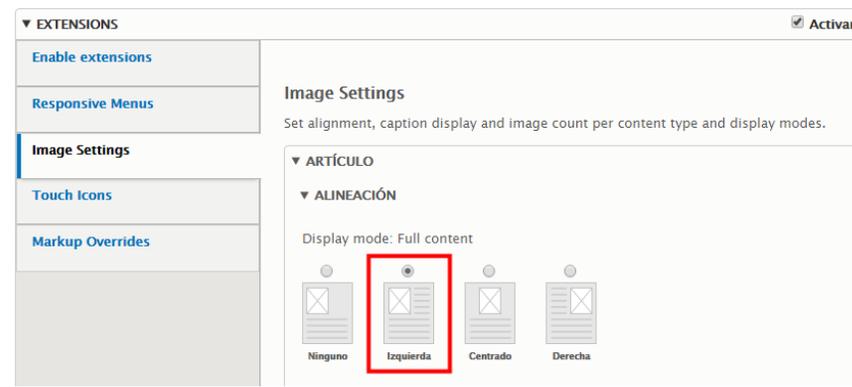


Paso 4. Configuración del tema

Desde las **Opciones del tema** podemos acceder a muchas opciones de configuración proporcionadas por el tema base AdaptiveTheme. Algunas de estas opciones son:

- Habilitar extensiones (Enable extensions). Permite activar los grupos de opciones de configuración.
- Menús responsive. Configuración de los menús del sitio.
- Alineación de imágenes y estilos de títulos por tipo de contenido.
- Fuentes. Permite añadir fuentes de texto especiales, como por ejemplo las fuentes de Google.
- Slideshows. Permite añadir slideshows de imágenes.
- Esquema de colores. Permite cambiar el juego de colores del tema.
- Opciones básicas. Opciones comunes a todos los temas (elementos a mostrar, imagen del logo e imagen de icono de acceso rápido).

Como ejemplo, vamos a añadir un nuevo logo, cambiar el juego de colores, y establecer Alineación izquierda para las imágenes mostradas en el tipo de contenido Artículo **[F8.2g]**.



F8.2g
Subtema con AdaptiveTheme
Configuración del tema.

El resultado es el mostrado en la Figura **[F8.2h]**.



F8.2h
Subtema con AdaptiveTheme
Visualización de la página
tras la configuración del
tema.

8.3 Layouts responsive en CSS

El layout es la base estructural de la página, donde se definen las regiones o contenedores y cómo se comportan cuando cambia el tamaño de pantalla.

Por ejemplo, si definimos un layout con dos columnas y establecemos los anchos a 30% y 70%, el ancho se adaptará al tamaño de la pantalla, respetando esos porcentajes.

En este apartado vamos a introducir las siguientes estrategias de diseño o construcción de layouts responsive en CSS:

- **Modelo de cajas clásico.** Utiliza principalmente la propiedad **float** para colocar los contenedores o cajas.
- **Grid basado en columnas.** Se divide la página en columnas de igual ancho (generalmente 12 columnas). El ancho de cada elemento se indica en número de columnas, especificando un nombre de clase. Es más flexible que el método clásico y no requiere el uso de float para cada elemento.
- **Flexbox.** Se basa en la creación de contenedores flexibles utilizando la propiedad **display: flex**. En los contenedores flexibles podemos utilizar nuevas propiedades "flex" para indicar cómo se comportarán los elementos que contiene (alineación, organización como fila o columna, espacio entre elementos, etc.).
- **CSS Grid Layout.** Utiliza un grid o cuadrícula con filas y columnas para organizar los elementos de la página. Es una evolución de Flexbox.

box-sizing: border-box;

Antes de entrar en detalle de cada uno de los modelos de layout, tenemos que hablar de la propiedad box-sizing. Esta propiedad define cómo se calcula el ancho y alto real de un elemento, teniendo en cuenta que el padding y el borde del elemento ocupan espacio.

Los valores posibles de box-sizing son:

- **box-sizing: content-box;**
Es el valor por defecto. El ancho/alto especificado sólo incluye el contenido. El ancho/alto añadido por las propiedades border y padding debe sumarse para calcular el ancho total del elemento.

En el siguiente ejemplo, el contenedor tiene un **ancho real de 114px** = 100px (content) + 2 x 2px (border) + 2 x 5px (padding);

```
div {
  box-sizing: border-content;
  width: 100px;
  border: 2px solid red;
  padding: 5px;
  float: left;
}
```

- **box-sizing: border-box;**

El ancho y alto especificado incluye el contenido, el borde y el padding.

En el siguiente ejemplo, el contenedor tiene un **ancho real de 100px**. Como los valores de border y padding son fijos, el ancho del contenido se ajusta automáticamente para que el total no pase de los 100px = 86px (content) + 2 x 2px (border) + 2 x 5px (padding).

```
div {
  box-sizing: border-box;
  width: 100px;
  border: 2px solid red;
  padding: 5px;
  float: left;
}
```

Para tener mayor control sobre la disposición de elementos, nos interesa este segundo valor (box-sizing: border-box;). De esta forma se evita un efecto muy típico, que es que contenedores que suman el 100% de ancho, no se puedan colocar en la misma fila por falta de espacio real. Como ejemplo vamos a intentar colocar dos cajas una al lado de otra:

```
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse ac augue ut arcu volutpat finibus. Morbi interdum vel
ligula in egestas. Sed commodo iaculis elit ac maximus. Etiam quis
placerat lacus, nec laoreet metus. </div>
```

```
<div>Vestibulum fringilla sem lacus, eget pharetra purus imperdiet
ut. Aliquam pharetra eu tortor sit amet viverra. Mauris lobortis
vel leo mattis ullamcorper. Nulla a nunc facilisis, tristique
massa vel, pretium diam. </div>
```

Con **border-content**:

```
div {
  box-sizing: border-content;
  width: 50%;
  border: 2px solid red;
  padding: 5px;
  float: left;
}
```

Al 50% de ancho de cada caja hay que sumarle el tamaño del border y del padding, por lo que no caben en una misma fila **[F8.3a]**.

F8.3a

border-content

Cajas con box-sizing:
border-content;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ac augue ut arcu volutpat finibus. Morbi interdum vel ligula in egestas. Sed commodo iaculis elit ac maximus. Etiam quis placerat lacus, nec laoreet metus.

Vestibulum fringilla sem lacus, eget pharetra purus imperdiet ut. Aliquam pharetra eu tortor sit amet viverra. Mauris lobortis vel leo mattis ullamcorper. Nulla a nunc facilisis, tristique massa vel, pretium diam.

Con **border-box**:

```
div {
  box-sizing: border-box;
  width: 50%;
  border: 2px solid red;
  padding: 5px;
  float: left;
}
```

El tamaño real de cada caja es del 50% del ancho de pantalla, por lo que encajan perfectamente una al lado de la otra, ocupando el 100% del ancho disponible **[F8.3b]**:

F8.3b

border-box

Cajas con box-sizing:
border-box;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse ac augue ut arcu volutpat finibus. Morbi interdum vel ligula in egestas. Sed commodo iaculis elit ac maximus. Etiam quis placerat lacus, nec laoreet metus.

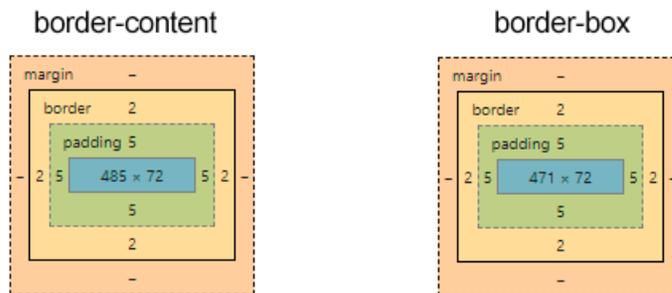
Vestibulum fringilla sem lacus, eget pharetra purus imperdiet ut. Aliquam pharetra eu tortor sit amet viverra. Mauris lobortis vel leo mattis ullamcorper. Nulla a nunc facilisis, tristique massa vel, pretium diam.

En la figura **[F8.3c]** vemos los tamaños de un mismo elemento en función del valor de la propiedad box-sizing. A la derecha el contenido del elemento se reduce a un ancho de 471px para que el ancho total se corresponda con los 485px del elemento completo.

F8.3c

box-sizing

Comparativa de tamaños para un mismo elemento cambiando el valor de box-sizing.



Para generalizar el uso de border-box a todos los elementos de la página, podemos añadir el código:

```
* {
  box-sizing: border-box;
}
```

CSS responsive clásico (float)

En el **apartado 6.5** ya estudiamos las propiedades de los contenedores y el posicionamiento de elementos con float. Este es el modelo clásico de posicionamiento que podemos utilizar también en diseño responsive. Para que el diseño sea responsive y se adapte al tamaño de la pantalla, utilizaremos medidas en porcentaje (%).

HTML:

```
<div class="header">
  <h1>Lorem ipsum</h1>
</div>

<div class="clearfix">
  <div class="column menu">
    <ul>
      <li>Phasellus porttitor</li>
      <li>Curabitur velit</li>
      <li>Nam mattis</li>
      <li>Cras sit amet</li>
    </ul>
  </div>

  <div class="column content">
    <h1>Cras sit amet</h1>
    <p>Nunc sit amet justo eget velit egestas iaculis. Donec
    ullamcorper libero at dui aliquam, at aliquet tortor ultricies.
    Mauris sed iaculis risus. Curabitur venenatis orci et elit iaculis
    blandit.</p>
    <p>Donec dictum tincidunt lacus, ut lobortis tortor feugiat
    vel. Cras commodo sapien eu porta lobortis. Aenean gravida, libero
    non eleifend commodo, neque lorem maximus turpis, sit amet pretium
    mauris erat et lectus. Curabitur tristique ligula in odio
    elementum venenatis.</p>
  </div>
</div>

<div class="footer">
  <p>Footer Text</p>
</div>
```

CSS:

```
* {
  box-sizing: border-box;
}
.header, .footer {
  background-color: grey;
  color: white;
  padding: 15px;
}
.column {
  float: left;
  padding: 15px;
}
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.menu {
  width: 25%;
}
.content {
  width: 75%;
}
```

```

}
.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
.menu li {
  padding: 8px;
  margin-bottom: 8px;
  background-color: #0088dd;
  color: #ffffff;
}

```

F8.3d
Maquetación con float
 Maquetación clásica con float.

Lorem ipsum

Phasellus porttitor	<p>Cras sit amet</p> <p>Nunc sit amet justo eget velit egestas iaculis. Donec ullamcorper libero at dui aliquam, at aliquet tortor ultricies. Mauris sed iaculis risus. Curabitur venenatis orci et elit iaculis blandit.</p> <p>Donec dictum tincidunt lacus, ut lobortis tortor feugiat vel. Cras commodo sapien eu porta lobortis. Aenean gravida, libero non eleifend commodo, neque lorem maximus turpis, sit amet pretium mauris erat et lectus. Curabitur tristique ligula in odio elementum venenatis.</p>
Curabitur velit	
Nam mattis	
Cras sit amet	

Footer Text

clearfix hack

En el ejemplo anterior hemos utilizado un código CSS "especial" que evita el efecto mostrado en la Figura [F8.3e]. Cuando en el interior de un contenedor hay elementos hijos flotando (float), el alto del contenedor no tiene en cuenta el alto de esos elementos hijos, pudiendo estos hijos sobresalir de los márgenes del contenedor.

F8.3e
clearfix hack
 Sin utilizar el hack el elemento hijo se sale del elemento padre.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam pharetra congue diam, sit amet ultricies risus ultrices at. Praesent feugiat a lectus in gravida. Pellentesque elit eros, rhoncus ut nisl semper, accumsan sodales lectus. Sed mattis dolor consequat condimentum tincidunt. Fusce id sem id mi blandit fringilla. Pellentesque faucibus ex non facilisis sollicitudin.



Para evitar esto se utiliza lo que se conoce como "clearfix hack", que es un código CSS que añadiremos como rutina a los contenedores que puedan sufrir este efecto.

El clearfix hack clásico es:

```

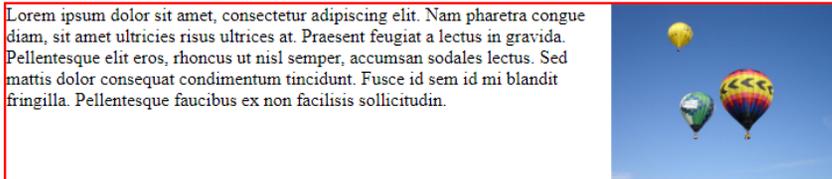
.clearfix {
  overflow: auto;
}

```

Pero nosotros utilizaremos una alternativa más utilizada actualmente por tener menos efectos secundarios en versiones de navegadores más antiguos:

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

En la Figura [8.3f] vemos la corrección que realiza este hack.



F8.3f
clearfix hack

Utilizando el hack el elemento padre se adapta al tamaño del elemento hijo.

A continuación, mostramos en código la forma de aplicar el hack. El contenedor (<div>) tendrá asignada la clase .clearfix. En el CSS vemos que la imagen está flotando dentro del contenedor. Puedes probar el funcionamiento del clearfix hack eliminando la clase .clearfix del contenedor padre.

HTML:

```
<div class="clearfix">
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam pharetra congue diam, sit amet ultricies risus ultrices at. Praesent feugiat a lectus in gravida. Pellentesque elit eros, rhoncus ut nisl semper, accumsan sodales lectus. Sed mattis dolor consequat condimentum tincidunt. Fusce id sem id mi blandit fringilla. Pellentesque faucibus ex non facilisis sollicitudin.
</div>
```

CSS:

```
div {
  border: 2px solid red;
}

img {
  float: right;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

Enlaces de ampliación:

- Modelo de cajas y posicionamiento CSS (Diego C Martin)
<https://www.diegocmartin.com/modelo-de-cajas-y-posicionamiento-css/>
- CSS Layout - float and clear (w3schools.com)
https://www.w3schools.com/Css/css_float.asp

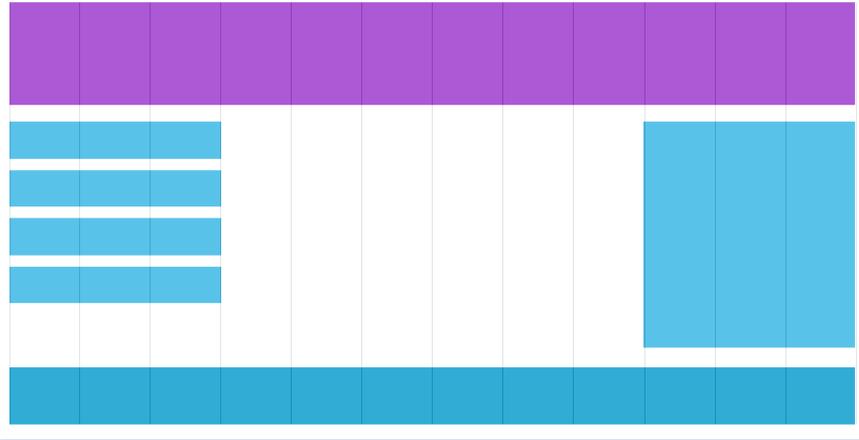
Grid basado en columnas

En este modelo de layout, la página se divide en columnas de igual ancho (generalmente 12 columnas). El ancho de cada elemento se indica en número de columnas, especificando un nombre de clase **[F8.3g]**.

F8.3g

Grid basado en columnas

Página dividida en un grid de 12 columnas.



Puedes probar el comportamiento responsive de este grid en:

https://www.w3schools.com/css/tryresponsive_grid.htm

Para construir el grid, tenemos que calcular el ancho de cada columna en porcentaje. Por ejemplo, si vamos a crear un grid de 2 columnas, el ancho de una columna será del 50%, y el ancho de dos columnas será del 100%.

Como hemos comentado, 12 es el número de columnas habitual. Dividiendo 100% / 12 columnas obtenemos el ancho de una columna, que será 8.33%.

A continuación, definiremos las clases para las 12 columnas, con el prefijo col- y el ancho correspondiente al número de columnas que ocupará un elemento. Por ejemplo, .col-3 será el equivalente a tres columnas, y por tanto su ancho será de $3 * 8.33\% = 25\%$ (se corresponde con un cuarto del ancho total).

CSS:

```
* {
  box-sizing: border-box;
}

.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

```
[class*="col-"] {
```

```

float: left;
padding: 15px;
border: 1px solid red;
}

.row::after {
  content: "";
  clear: both;
  display: table;
}

```

Además de las clases `.col-*`, hemos añadido:

- box-sizing: border-box; para todos los elementos.
- Estilos específicos para todas las columnas, especialmente float: left;
- El clearfix hack visto anteriormente. En este caso la clase se ha llamado `.row` en lugar de `.clearfix`, pero el funcionamiento es el mismo.

Ahora sólo tenemos que indicar el número de columnas que ocupa cada elemento. Por ejemplo, si queremos añadir dos contenedores que ocupen 3 (25%) y 9 (75%) columnas respectivamente **[F8.3h]**:

```

<div class="row">
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere ipsum dolor, sed lobortis tellus vestibulum ut. Quisque hendrerit ipsum sed lacus pulvinar, vitae hendrerit sapien tempor. Quisque rhoncus lacinia leo at placerat. Mauris id nisi vel est imperdiet convallis. Vestibulum consectetur pellentesque nibh, at pharetra ipsum. Aenean egetas consequat feugiat. Nullam facilisis ipsum sed aliquam tempus.

Nullam at sollicitudin leo, eget sagittis purus. Nunc porta justo placerat dignissim efficitur. In ac feugiat libero. Pellentesque gravida erat at diam auctor, eu molestie odio porttitor. Maecenas dui elit, ornare sit amet condimentum eu, elementum a justo. Nullam euismod ultricies ultrices. Curabitur in commodo lectus. Aenean ut feugiat purus, nec vulputate nisi. Fusce ipsum arcu, tempor eget rutrum sagittis, tempus eget dui. Mauris arcu dui, euismod vitae neque et, pulvinar vehicula sapien. Sed placerat hendrerit sapien ac pretium. Praesent tempus dui sed tempor volutpat. Proin elit ligula, suscipit sed mauris eget, dictum mollis libero. Aliquam erat volutpat. Suspendisse potenti. Duis volutpat mi id dolor vestibulum pretium. Vivamus id diam nec tortor laoreet ornare eget in metus. Ut eget lectus velit. Phasellus ac tellus non ex suscipit euismod. Proin eget nisi id nisi placerat interdum quis in ante.

F8.3h

Grid basado en columnas

Representación de columnas de 25% y 75%.

En el siguiente ejemplo se representan varias filas con diferentes anchos de columnas **[F8.3i]**:

```

<div class="row">
  <div class="col-12">...</div><!-- 100% -->
</div>

<div class="row">
  <div class="col-6">...</div> <!-- 50% -->
  <div class="col-6">...</div> <!-- 50% -->
</div>

<div class="row">
  <div class="col-9">...</div> <!-- 75% -->
  <div class="col-3">...</div> <!-- 25% -->
</div>

```

F8.3i

Grid basado en columnas

Representación de varias filas con columnas de diferentes anchos.

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere ipsum dolor, sed lobortis tellus vestibulum ut. Quisque hendrerit ipsum sed lacus pulvinar, vitae hendrerit sapien tempor. Quisque rhoncus lacinia leo at placerat. Mauris id nisl vel est imperdiet convallis. Vestibulum consectetur pellentesque nibh, at pharetra ipsum. Aenean egestas consequat feugiat. Nullam facilisis ipsum sed aliquam tempus.</p>	
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis posuere ipsum dolor, sed lobortis tellus vestibulum ut. Quisque hendrerit ipsum sed lacus pulvinar, vitae hendrerit sapien tempor. Quisque rhoncus lacinia leo at placerat. Mauris id nisl vel est imperdiet convallis. Vestibulum consectetur pellentesque nibh, at pharetra ipsum. Aenean egestas consequat feugiat. Nullam facilisis ipsum sed aliquam tempus.</p>	<p>Nullam at sollicitudin leo, eget sagittis purus. Nunc porta justo placerat dignissim efficitur. In ac feugiat libero. Pellentesque gravida erat at diam auctor, eu molestie odio porttitor. Maecenas dui elit, ornare sit amet condimentum eu, elementum a justo. Nullam euismod ultricies ultrices. Curabitur in commodo lectus. Aenean ut feugiat purus, nec vulputate nisl. Fusce ipsum arcu, tempor eget rutrum sagittis, tempus eget dui. Mauris arcu dui, euismod vitae neque et, pulvinar vehicula sapien. Sed placerat hendrerit sapien ac pretium. Praesent tempus dui sed tempor volutpat. Proin elit ligula, suscipit sed mauris eget, dictum mollis libero. Aliquam erat volutpat. Suspendisse potenti. Duis volutpat mi id dolor vestibulum pretium. Vivamus id diam nec tortor laoreet ornare eget in metus. Ut eget lectus velit. Phasellus ac tellus non ex suscipit euismod. Proin eget nisi id nisl placerat interdum quis in ante.</p>
<p>Nullam at sollicitudin leo, eget sagittis purus. Nunc porta justo placerat dignissim efficitur. In ac feugiat libero. Pellentesque gravida erat at diam auctor, eu molestie odio porttitor. Maecenas dui elit, ornare sit amet condimentum eu, elementum a justo. Nullam euismod ultricies ultrices. Curabitur in commodo lectus. Aenean ut feugiat purus, nec vulputate nisl. Fusce ipsum arcu, tempor eget rutrum sagittis, tempus eget dui. Mauris arcu dui, euismod vitae neque et, pulvinar vehicula sapien. Sed placerat hendrerit sapien ac pretium. Praesent tempus dui sed tempor volutpat. Proin elit ligula, suscipit sed mauris eget, dictum mollis libero. Aliquam erat volutpat. Suspendisse potenti. Duis volutpat mi id dolor vestibulum pretium. Vivamus id diam nec tortor laoreet ornare eget in metus. Ut eget lectus velit. Phasellus ac tellus non ex suscipit euismod. Proin eget nisi id nisl placerat interdum quis in ante.</p>	<p>Nullam at sollicitudin leo, eget sagittis purus. Nunc porta justo placerat dignissim efficitur. In ac feugiat libero. Pellentesque gravida erat at diam auctor, eu molestie odio porttitor. Maecenas dui elit, ornare sit amet condimentum eu, elementum a justo. Nullam euismod ultricies ultrices. Curabitur in commodo lectus. Aenean ut feugiat purus, nec vulputate nisl. Fusce ipsum arcu, tempor eget rutrum sagittis, tempus eget dui. Mauris arcu dui, euismod vitae neque et, pulvinar vehicula sapien. Sed placerat hendrerit sapien ac pretium. Praesent tempus dui sed tempor volutpat. Proin elit ligula, suscipit sed mauris eget, dictum mollis libero. Aliquam erat volutpat. Suspendisse potenti. Duis volutpat mi id dolor vestibulum pretium. Vivamus id diam nec tortor laoreet ornare eget in metus. Ut eget lectus velit. Phasellus ac tellus non ex suscipit euismod. Proin eget nisi id nisl placerat interdum quis in ante.</p>

Por último, mostramos un ejemplo con una estructura de página típica con encabezado, columna a la izquierda del contenido y pie de página **[F8.3j]**.

```

<div class="row">
  <div class="col-12 header">
    <h1>Lorem ipsum</h1>
  </div>
</div>
<div class="row">
  <div class="col-3 menu">
    <ul>
      <li>Phasellus porttitor</li>
      <li>Curabitur velit</li>
      <li>Nam mattis</li>
      <li>Cras sit amet</li>
    </ul>
  </div>

  <div class="col-9">
    <h1>Cras sit amet</h1>
    <p>Nunc sit amet justo eget velit egestas iaculis. Donec ullamcorper libero at dui aliquam, at aliquet tortor ultricies. Mauris sed iaculis risus. Curabitur venenatis orci et elit iaculis blandit.</p>
    <p>Donec dictum tincidunt lacus, ut lobortis tortor feugiat vel. Cras commodo sapien eu porta lobortis. Aenean gravida, libero non eleifend commodo, neque lorem maximus turpis, sit amet pretium mauris erat et lectus. Curabitur tristique ligula in odio elementum venenatis.</p>
  </div>
</div>
<div class="row">
  <div class="col-12 footer">
    <p>Footer Text</p>
  </div>
</div>

```

**F8.3j****Grid basado en columnas**

Estructura de página típica con encabezado, columna izquierda, área de contenido y pie de página.

Enlaces de ampliación:

- Responsive Web Design - Grid-View (w3schools.com)
https://www.w3schools.com/css/css_rwd_grid.asp

CSS Flexbox

El modelo Flexbox se basa en contenedores flexibles o configurables. Cada contenedor puede albergar varios elementos hijos, que se mostrarán de diferentes formas en función de la configuración del contenedor.

Un contenedor Flexbox es un elemento contenedor (por ejemplo, <div>) con la propiedad **display: flex**. Una vez establecido el tipo de display a flex, el elemento padre puede tener asociadas otras propiedades "flex" para configurar su contenido.

En la Figura [F8.3k] vemos un contenedor flex que ocupa el 100% del ancho de pantalla. Dentro tiene 4 elementos que hemos representado en forma de bloques numerados.

F8.3k CSS Flexbox

Ejemplo de contenedor flex.



Propiedades del contenedor flex

- **display**. Lo primero que debemos hacer es convertir un contenedor en flex, mediante la propiedad display.

```
.flex-container {
  display: flex;
}
```

- **flex-direction [F8.3l][F8.3m]**. Estable el orden y dirección de los elementos hijos del contenedor. Los valores posibles son:
 - o **row**. Fila, de izquierda a derecha. Es el valor por defecto.
 - o **row-reverse**. Fila, de derecha a izquierda.
 - o **column**. Columna, de arriba abajo.
 - o **column-reverse**. Columna, de abajo a arriba.

```
.flex-container {
  display: flex;
  flex-direction: row | row-reverse | column | column-reverse;
}
```

row

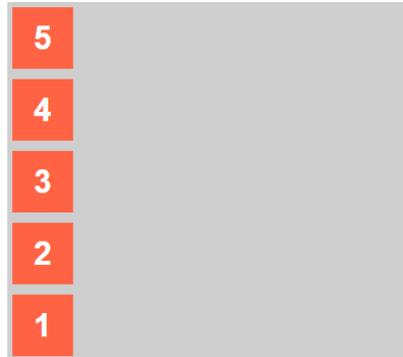


row-reverse



F8.3l CSS Flexbox

Dirección de los elementos del contenedor con flex-direction (filas).

column**column-reverse****F8.3m**
CSS Flexbox

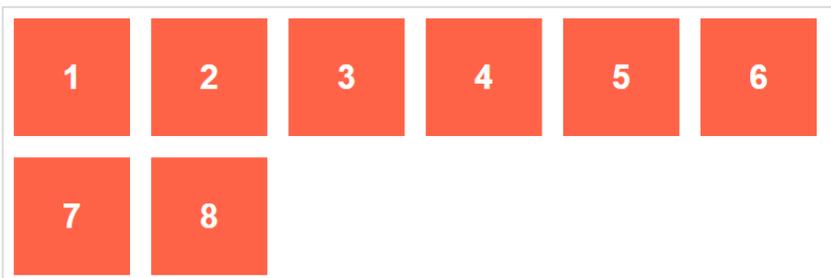
Dirección de los elementos del contenedor con flex-direction (columnas).

- **flex-wrap [F8.3n]**. Por defecto (**nowrap**), los elementos del contenedor se mantienen en una única línea, comprimiéndose o expandiéndose en función del ancho disponible. Si establecemos este atributo como **wrap**, los elementos pasarán a la siguiente fila cuando se queden sin espacio en la fila actual. El valor **wrap-reverse** desplaza los elementos a la siguiente fila en orden inverso (de abajo hacia arriba).

```
.flex-container {
  display: flex;
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

nowrap

F8.3n
CSS Flexbox
Propiedad flex-wrap.

wrap**wrap-reverse**

- **flex-flow.** Combina las propiedades anteriores, flex-direction y flex-wrap.

```
.flex-container {
  display: flex;
  flex-flow: row wrap;
}
```

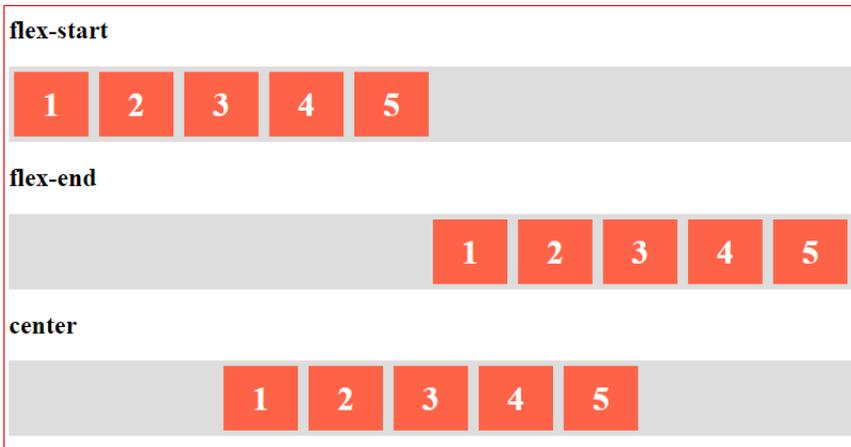
- **justify-content [F8.3o] [F8.3p].** Tipo de **justificación horizontal** de los elementos del contenedor. Los valores posibles son:
 - o **flex-start.** Alineación izquierda.
 - o **flex-end.** Alineación derecha.
 - o **center.** Alineación centrada.
 - o **space-between.** Justificación completa, dejando espacios en medio de los elementos para que el primer elemento esté completamente a la izquierda y el último elemento completamente a la derecha. Es similar a la justificación completa del texto.
 - o **space-around.** Los elementos se distribuyen dejando un espacio entre ellos. Entre un elemento y el borde del contenedor habrá 1 espacio, pero entre dos elementos centrales, habrá 2 espacios, uno por cada elemento.
 - o **space-evenly.** Es similar al anterior pero siempre habrá 1 único espacio entre elementos y entre un elemento y el borde del contenedor.

```
.flex-container {
  display: flex;
  justify-content: flex-start | flex-end | center | space-between |
  space-around | space-evenly;
}
```

F8.3o

CSS Flexbox

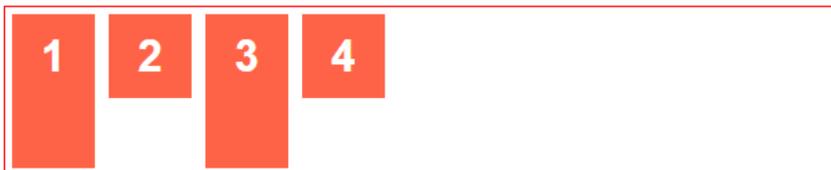
Propiedad justify-content.



space-between**space-around****space-evenly****F8.3p****CSS Flexbox**

Propiedad justify-content.

- **align-items [F8.3q][F8.3r]**. Tipo de **justificación vertical** de los elementos del contenedor. Los valores posibles son:
 - o **flex-start**. Los elementos se alinean en la línea superior del contenedor.
 - o **flex-end**. Los elementos se alinean en la línea inferior del contenedor.
 - o **center**. Los elementos se alinean en el centro (vertical) del contenedor.
 - o **baseline**. La línea base se corresponde con la base del texto que contiene el elemento (ver Figura [F8.3r]).
 - o **stretch** (por defecto). Los hijos se extienden para ajustarse al alto del contenedor.

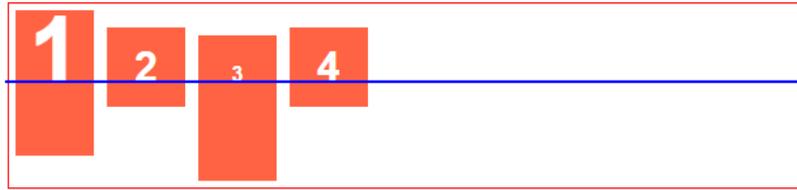
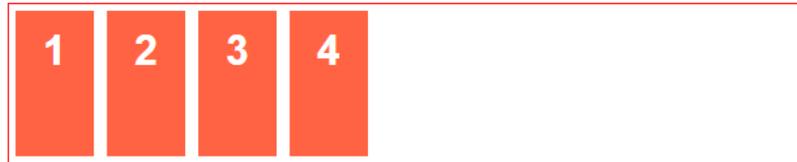
flex-start**F8.3q****CSS Flexbox**

Propiedad align-items.

flex-end**center**

F8.3r**CSS Flexbox**

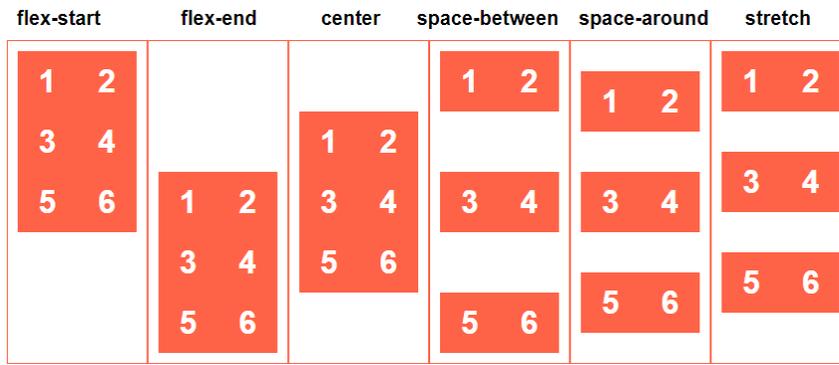
Propiedad align-items.

baseline**stretch**

```
.flex-container {
  display: flex;
  align-items: flex-start | flex-end | center | baseline | stretch;
}
```

- **align-content [F8.3s]**. Tipo de **justificación vertical** del conjunto completo de elementos o contenido dentro del contenedor. Los valores posibles son:
 - o **flex-start**. El contenido completo se alinea en la línea superior del contenedor.
 - o **flex-end**. El contenido completo se alinea en la línea inferior del contenedor.
 - o **center**. El contenido completo se alinea en el centro (vertical) del contenedor.
 - o **space-between**. Justificación completa, dejando espacios en medio de los elementos para que el primer elemento esté completamente arriba y el último elemento completamente abajo.
 - o **space-around**. Los elementos se distribuyen dejando un espacio vertical entre ellos.
 - o **stretch** (por defecto). El primer elemento se alinea en la parte superior y el resto de elementos se separan verticalmente para ocupar todo el espacio disponible.

```
.flex-container {
  display: flex;
  align-content: flex-start | flex-end | center | space-between |
  space-around | stretch;
}
```



F8.3s
CSS Flexbox
 Propiedad align-content.

Veamos algunos ejemplos más combinando estas propiedades:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
</div>
```

- a) Elementos en fila y justificación centrada. En las figuras **[F8.3t]** y **[F8.3u]** vemos el efecto de reducción de los elementos cuando se reduce el ancho del contenedor.

```
.flex-container {
  display: flex;
  flex-direction: row;
  justify-content: center;
  background-color: #029200;
}

.flex-container > div {
  background-color: #FFF;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 50px;
  font-size: 30px;
}
```



F8.3t
CSS Flexbox
 Ejemplo combinando propiedades (extendido).



F8.3u
CSS Flexbox
 Ejemplo combinando propiedades (comprimido).

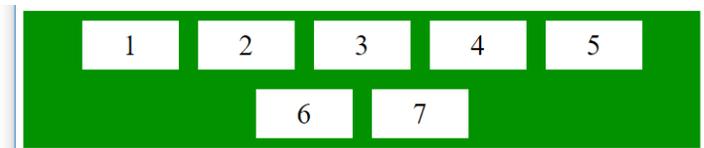
- b) Elementos en fila, justificación centrada y wrap activado. Ahora, cuando reducimos el ancho del contenedor, los elementos que no caben pasan a la siguiente línea **[F8.3v]**.

```
.flex-container {
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: center;
  background-color: #029200;
}
```

F8.3v

CSS Flexbox

Ejemplo combinando propiedades (comprimido).



- c) Elementos en columna, con wrap activado. Vemos cómo se distribuyen los elementos en columnas cuando se reduce el alto del contenedor **[F8.3w]**.

```
.flex-container {
  display: flex;
  flex-flow: column wrap;
  background-color: #029200;
  height: 250px;
}
```

F8.3w

CSS Flexbox

Ejemplo combinando propiedades (comprimido).



Propiedades de los elementos hijos (flex items)

Los elementos hijos también pueden tener propiedades específicas aportadas por flex:

- **order**. Los elementos se colocan por defecto en el orden establecido en el código HTML. El atributo orden, cuyo valor por defecto es 0, permite cambiar el orden de los elementos. Los elementos se mostrarán de menor a mayor orden. Cuando el número de orden es el mismo, se muestran según el orden original en el HTML.

```
.item {
  order: <integer>; /* default 0 */
}
```

En el siguiente ejemplo los elementos 2, 3 y 4 conservan su peso original (0). El elemento 5 se coloca en primer lugar con un peso menor (-1), y el elemento 1 se coloca en último lugar con un peso mayor (3) **[F8.3x]**.

```
.item1 {
  order: 3;
}
.item5 {
  order: -1;
}
```



F8.3x CSS Flexbox

Propiedades de elementos hijos (order).

- **flex-grow**. Permite que unos elementos crezcan más que otros cuando el contenedor tiene espacio suficiente para ello. El valor por defecto de todos los elementos es 1. Si establecemos el valor 2 para un elemento, este crecerá el doble que los demás **[F8.3y]**.

```
.item {
  flex-grow: <number>; /* default 0 */
}
```

En el siguiente ejemplo el elemento 1 puede crecer el doble (2), y el elemento 3 el cuádruple (4).

```
.item1 {
  flex-grow: 2;
}
.item3 {
  flex-grow: 4;
}
```



F8.3y CSS Flexbox

Propiedades de elementos hijos (flex-grow).

- **flex-shrink**. Permite que unos elementos se reduzcan más que otros cuando se reduce el espacio del contenedor y los elementos deben redimensionarse. El valor por defecto es 1. Si establecemos el valor 2 para un elemento, este podrá reducirse a la mitad de ancho que el resto de elementos.

```
.item {
  flex-shrink: <number>; /* default 1 */
}
```

En el siguiente ejemplo el elemento 1 puede reducirse el doble (2), y el elemento 3 el cuádruple (4) **[F8.3z]**.

```
.item1 {
  flex-shrink: 2;
}

.item3 {
  flex-shrink: 4;
}
```

F8.3z**CSS Flexbox**

Propiedades de elementos hijos (flex-shrink).



- **flex-basis.** Permite definir el tamaño por defecto de un elemento antes de que se distribuya el espacio libre disponible. El valor **auto** implica que se utilizará el tamaño definido en las propiedades del elemento (width o height). Si queremos establecer otro valor, podemos hacerlo como unidad de medida (% , rem, px, etc.).

```
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

- **flex.** Esta propiedad agrupa a las propiedades flex-grow, flex-shrink y flex-basis vistas anteriormente.

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

- **align-self.** Es equivalente a align-items, pero aplicado a un elemento en particular. Los valores posibles son los mismos que para align-items (flex-start, flex-end, center, baseline y stretch).

```
.item {
  align-self: auto | flex-start | flex-end | center | baseline
              | stretch;
}
```

Enlaces de ampliación:

- CSS Flexbox (w3schools.com)
https://www.w3schools.com/css/css3_flexbox.asp
- A Complete Guide to Flexbox
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

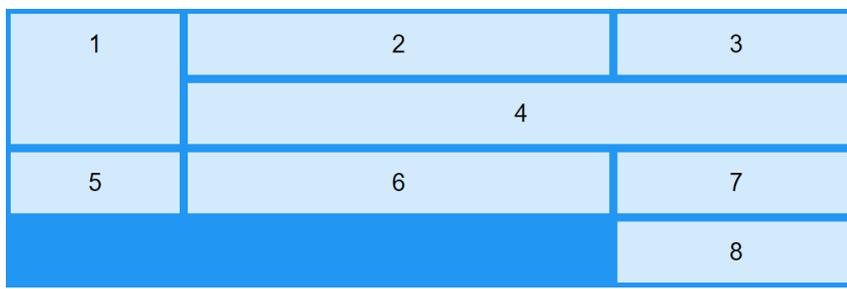
CSS Grid Layout

El modelo CSS Grid Layout se puede considerar una mezcla entre el Grid basado en columnas y los contenedores de Flexbox. CSS Grid Layout permite crear contenedores flexibles con elementos hijos, en un formato de cuadrícula con filas y columnas **[F8.3aa]**.

El contenedor se define con `display: grid`, y a partir de ahí podemos utilizar las propiedades específicas de CSS Grid Layout.

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: 20% 50% auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}
.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
.item1 {
  grid-row-start: 1;
  grid-row-end: 3;
}
.item4 {
  grid-column-start: 2;
  grid-column-end: 4;
}
.item8 {
  grid-column-start: 3;
  grid-column-end: 4;
}
```



F8.3aa
CSS Grid Layout
 Disposición de regiones con CSS Grid Layout.

- **grid-template-columns.** Define el número de columnas y sus anchos. En nuestro ejemplo hemos definido 3 columnas, con anchos del 20%, 50% y auto (restante). Si todos los anchos se indican como auto, las columnas tendrán el mismo ancho.
- **grid-template-rows.** Similar a la propiedad anterior pero para establecer el alto de las filas.
- **grid-row-gap.** Ancho de las líneas de separación de filas.
- **grid-column-gap.** Ancho de las líneas de separación de columnas.
- **grid-gap.** Propiedad que agrupa los valores de **grid-row-gap** y **grid-column-gap**.
- Las propiedades **justify-items**, **align-items**, **place-items**, **justify-content**, **align-content** y **place-content** son muy parecidas a las propiedades de posicionamiento de elementos de Flexbox. Consulta los enlaces recomendados para ver los valores específicos.

A nivel de elementos dentro del contenedor, podemos utilizar estas otras propiedades:

- **grid-column-start**, **grid-column-end.** Determina la posición horizontal del elemento dentro del grid, indicando la columna donde empieza y la columna donde termina (sin incluirla). Se puede utilizar la propiedad agrupada **grid-column**, separando los valores de inicio y fin por una /.
- **grid-row-start**, **grid-row-end.** Determina la posición vertical del elemento dentro del grid, indicando la fila donde empieza y la fila donde termina (sin incluirla). Se puede utilizar la propiedad agrupada **grid-row**, separando los valores de inicio y fin por una /.

En nuestro ejemplo, el elemento 1 comienza en la fila 1 y termina en la fila 3. Ocupa, por tanto, las filas 1 y 2. El elemento 4 ocupa las columnas 2 y 3, ya que comienza en la 2 y termina en la 4.

Enlaces de ampliación:

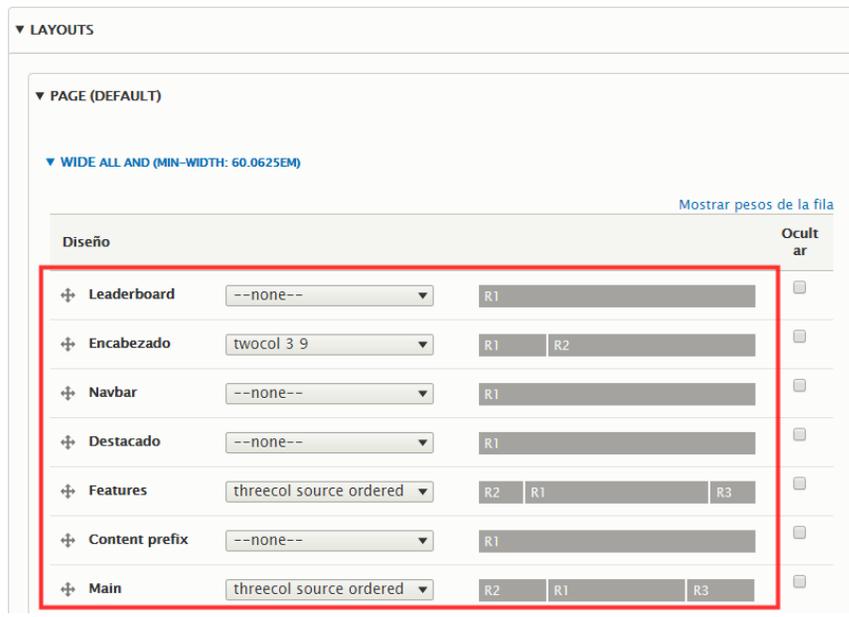
- **A Complete Guide to Grid**
<https://css-tricks.com/snippets/css/complete-guide-grid/>
- **CSS Grid Layout Module (w3schools.com)**
https://www.w3schools.com/css/css_grid.asp
- **CSS Grid Layout (MDN web docs)**
https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout

Configuración de layouts en AdaptiveTheme

En los subtemas generados a partir de AdaptiveTheme podemos configurar los layouts desde la interfaz de administración:

Administración ⇒ Apariencia ⇒ Tema [Layouts]

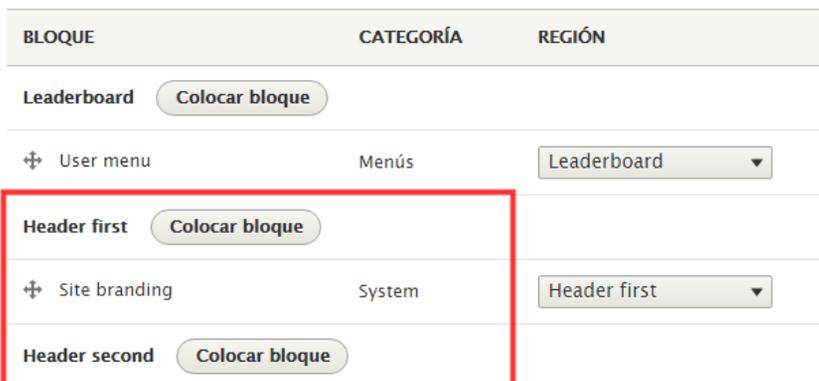
Aunque hablaremos de las Media Queries en el próximo apartado, desplegando el apartado **Page** veremos que los layouts están separados en función del ancho del dispositivo. Podemos seleccionar el layout para cada región del tema. En AdaptiveTheme los layouts son distribuciones preconfiguradas de columnas, sobre un grid de 12 columnas. Por ejemplo, "twocol 3 9" será un layout con dos columnas de 3 (25%) y 9 (75%) **[F8.3ab]**.



F8.3ab
Layouts en AdaptiveTheme

Selección de layouts para las diferentes regiones del tema.

Al asignar layouts de varias columnas a una región, ésta se dividirá en subregiones en la página de administración de bloques. Por ejemplo, el Header o encabezado al que hemos asignado el layout "twocol 3 9", generará las regiones **Header first** y **Header second** **[F8.3ac]**.



F8.3ac
Layouts en AdaptiveTheme

Subregiones creadas según el layout seleccionado.

No hay bloques en esta región

Si analizamos el código HTML generado al cargar la página **[F8.3ad]**, veremos que cada región es un contenedor Flexbox que contiene a su vez otro contenedor Flexbox. Dentro de este segundo contenedor encontraremos las subregiones como elementos del contenedor.

Los bloques colocados dentro de una región no son elementos de Flexbox directos. Esto quiere decir que podemos aplicar propiedades de hijos de contenedor Flexbox a las subregiones, como cambiar el orden, pero no a los bloques en sí.

CSS: `at_forcontu/layout/page-layout/css/page-layout-global.css`

```
.page {
  display: flex;
  flex-flow: column;
  width: 100%;
}

.l-rw {
  display: flex;
  flex-wrap: wrap;
  flex-direction: row;
  max-width: 75em;
  width: 100%;
}

.l-r {
  flex: 0 0 auto;
  padding: 0 0.625em;
  width: 100%;
}
```

F8.3ad Layouts en AdaptiveTheme

Identificación de contenedores flexbox en el código generado.



Los anchos de las columnas se aplicarán en función del layout escogido. Por ejemplo, para el layout en 2 columnas 3-9, los porcentajes serán 25% y 75%, respectivamente.

CSS: `at_forcontu/layout/page-layout/css/twocol/twocol-3-9.css`

```
.twocol-3-9 .arc--2 .l-r {
  width: 25%;
}

.twocol-3-9 .arc--2 .l-r:last-child {
  width: 75%;
}
```

El contenedor principal que incluye la región de contenido y las columnas laterales (sidebar-first y sidebar-second), también es de tipo flexbox, siendo estas regiones elementos hijos del contenedor flex. Esto nos permite cambiar fácilmente el orden de las regiones.

Veamos primero el código HTML generado al cargar la página.

HTML

```

<div class="l-pr page__row pr-main regions-multiple" id="main-content">
  <div class="l-rw regions container pr-main__rw arc--3 hr--1-2-3"
  data-at-regions="">

    <!-- contenido -->
    <div data-at-region="1" data-at-block-count="4" class="l-r
  region pr-main__content" id="rid-content">...</div>

    <!-- columna izquierda -->
    <div data-at-region="2" data-at-block-count="2" class="l-r
  region pr-main__sidebar-first" id="rid-sidebar-first">...<div>

    <!-- columna derecha -->
    <div data-at-region="3" data-at-block-count="2" class="l-r
  region pr-main__sidebar-second" id="rid-sidebar-second">...<div>
  </div>
</div>

```

Y el CSS encargado de organizar el tamaño y orden de estas regiones.

CSS: `at_forcontu/layout/page-layout/css/page-layout-global.css`

```

.l-rw {
  display: flex;
  flex-wrap: wrap;
  flex-direction: row;
  max-width: 75em;
  width: 100%;
}

/* contenido */
.pr-main .arc--3 .l-r:first-child {
  width: 50%;
  order: 2;
}

/* columna izquierda */
.pr-main .arc--3 .l-r:nth-child(2) {
  width: 25%;
  order: 1;
}

/* columna derecha */
.pr-main .arc--3 .l-r:last-child {
  width: 25%;
  order: 3;
}

```

Aunque en el código HTML las regiones están definidas en este orden: contenido, columna izquierda, columna derecha, la propiedad `order` de flexbox aplica el siguiente orden: columna izquierda (1), centro (2), columna derecha (3). Prueba a cambiar el orden de estos elementos desde el inspector de objetos del navegador.

Nota: El código CSS en `AdaptiveTheme` se genera a partir de archivos con código SCSS, que estudiaremos en la unidad 9 (SASS).

8.4 Media Queries

Las Media Queries son reglas definidas con la cláusula **@media** que sólo se aplican cuando se cumplen las condiciones especificadas. Generalmente las Media Queries se utilizan para aplicar estilos diferentes en función del ancho de pantalla del dispositivo (viewport) que está visitando la página.

Comencemos analizando un ejemplo:

```
body {
  background-color: Red;
  color: #fff;
}

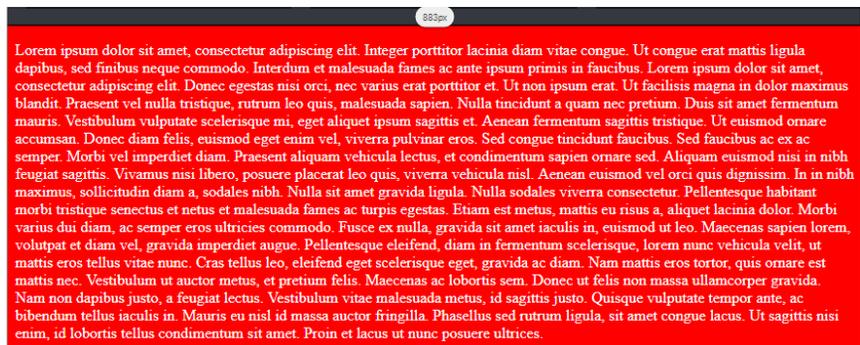
@media only screen and (max-width: 800px) {
  body {
    background-color: LightGray;
    color: #000;
  }
}
```

La regla definida para el body (fondo rojo y texto blanco) se aplicará por defecto, salvo que haya otra regla más específica (y posterior) que la sobrescriba **[F8.4a]**. Es lo que ocurre con la regla definida en la media query, que se aplicará cuando el ancho de pantalla sea menor o igual que 800 px (max-width: 800px).

F8.4a

Media Queries

Estilos por defecto cuando no se cumple la condición de la media query.

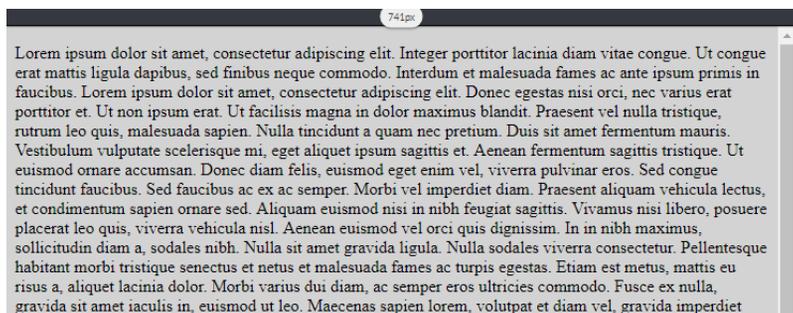


Cuando el ancho sea menor o igual de 800px, se aplicarán los estilos definidos en la media query (fondo gris y texto negro) **[F8.4b]**.

F8.4a

Media Queries

Estilos aplicados al cumplirse la condición de la media query.



Nota: Es importante tener en cuenta el orden en que se definen y ejecutan las reglas. El último valor que se aplica a una propiedad es el último escrito en el CSS (sobrescribe al resto), independientemente de si está definido o no dentro de una media query, entendiéndose que la condición se está cumpliendo en ese momento.

Sintaxis

La estructura general de una Media query es la siguiente:

```
@media not|only media-type and (media-feature and|or|not media-feature) {
  Código CSS;
}
```

Media-type. Hace referencia al tipo de elemento o dispositivo. Los valores posibles son:

- all. Se aplica para cualquier dispositivo.
- print. Se aplica en impresoras (versión para imprimir del documento).
- screen. Se aplica en pantallas en general (ordenador, tablet, móvil, etc.).
- speech. Se aplica en lectores de pantalla.

not|only. Se utilizan junto con el media-type. El valor **only** indica que sólo se debe aplicar la regla al media-type especificado (se utiliza por compatibilidad con navegadores antiguos). El valor **not** invierte el valor de la regla, de forma que se aplicará en el resto de casos.

Media features

La segunda parte de la media query son las media-features. Se trata de características o propiedades que se deben cumplir en la regla. Algunas de las características más utilizadas son:

- **min-width, max-width.** Ancho mínimo y máximo de la ventana donde se está abriendo la página. Por ejemplo, el tamaño de la ventana del navegador.
- **min-height, max-height.** Alto mínimo y máximo de la ventana donde se está abriendo la página (navegador).
- **orientation.** Orientación de la pantalla, pudiendo tener los valores landscape (horizontal) o portrait (vertical). La orientación se aplica en función del ancho y alto de la ventana del navegador, que no tiene que coincidir estrictamente con la orientación de pantalla del dispositivo.
- **min-resolution, max-resolution.** Resolución mínima y máxima del dispositivo.

Ejemplos

La siguiente regla se aplica cuando el ancho de pantalla está entre 600px y 900px (ambos inclusive).

```
@media only screen and (min-width: 600px) and (max-width: 900px) {
  div.example {
    background: yellow;
  }
}
```

En la siguiente regla definimos varias condiciones separadas por coma (,), lo que equivale a un OR. La regla se aplica si el ancho de pantalla está entre 600 y 900px o si el ancho de pantalla tiene un ancho mínimo de 1100px.

```
@media only screen and
(min-width: 600px) and (max-width: 900px), (min-width: 1100px) {
  div.example{
    background: yellow;
  }
}
```

En la siguiente regla se comprueba el ancho de pantalla (entre 768 y 1024px), pero también que el dispositivo esté en posición horizontal (landscape).

```
@media only screen and (min-width: 768px)
and (max-width: 1024px)
and (orientation: landscape) {
  div.example{
    background: yellow;
  }
}
```

Guía de formato de Media Queries en Drupal

Según la Guía de formato de CSS de Drupal, se debe aplicar el formato del resto de reglas CSS. Además de la indentación con dos espacios, tenemos que tener en cuenta estras otras reglas:

- Dejar un espacio entre la media-feature y su valor, después de los dos puntos (orientation: landscape)
- Los valores relativos a anchos y altos deberían especificarse en "rems". Aunque esto está especificado en la guía de estilos de Drupal, veremos que raramente se utilizan en este formato, siendo más habitual el uso directo en px.
- Por aclaración, el valor correspondiente en px se especificará como comentario tras la llave de apertura de la regla.

```
@media screen and (min-width: 28.125rem) { /* 450px */
  #page {
    margin-left: 20px;
    margin-left: 1.25rem;
  }
}
```

- **Guía de formato CSS en Drupal:**
<https://www.drupal.org/docs/develop/standards/css/css-formatting-guidelines>

Breakpoints típicos

Los breakpoints (o puntos de ruptura) son los anchos de pantalla donde se produce un cambio de estilo. Básicamente, son las cantidades que se expresarán en las Media queries a través de min-width y max-width).

Como es imposible tener en cuenta las resoluciones de todos los dispositivos que hay en el mercado, se suelen definir unos breakpoints que encajen en la mayoría de los casos.

Un ejemplo simple de declaración de breakpoints podría tener estos 5 grupos:

```
/* <600px. Para móviles */
@media only screen and (max-width: 600px) {...}

/* Entre 600px y 768px: Algunos móviles y tabletas en vertical */
@media only screen and (min-width: 600px) {...}

/* De 768px a 992px: Tabletás en horizontal */
@media only screen and (min-width: 768px) {...}

/* De 992px a 1200px: Portátiles y ordenadores con resolución baja */
@media only screen and (min-width: 992px) {...}

/* >1200px: Portátiles y ordenadores con resolución alta */
@media only screen and (min-width: 1200px) {...}
```

Ten en cuenta que no siempre es necesario definir los tramos indicando el ancho mínimo y máximo, ya que, por el orden establecido, se aplicará siempre la última media query que cumpla con las condiciones.

Carga de archivos css específicos

Una forma más cómoda de aplicar estilos según el dispositivo es cargando archivos de estilos diferentes en función del dispositivo. Lo haremos incluyendo una media query dentro del atributo media del elemento <link>:

```
<link rel="stylesheet" media="screen and (min-width: 900px)"
href="widescreen.css">
<link rel="stylesheet" media="screen and (max-width: 600px)"
href="smallscreen.css">
```

Ocultar elementos

La información que se muestra en un dispositivo puede no ser relevante en otros dispositivos. Cuando necesitamos eliminar elementos podemos utilizar la propiedad **display: none;** en combinación con media queries.

En el siguiente ejemplo el contenedor **div.example** se oculta cuando el ancho es menor de 600px:

```
@media only screen and (max-width: 600px) {  
  div.example {  
    display: none;  
  }  
}
```

Enlaces de ampliación:

- Responsive Web Design - Media Queries (w3schools.com)
https://www.w3schools.com/css/css_rwd_mediaqueries.asp
- CSS @media Rule (w3schools.com)
https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Breakpoints en Drupal

8.5

Definición de breakpoints en el archivo `.breakpoints.yml`

En Drupal 8, los temas y módulos pueden definir breakpoints que se almacenan en configuración (archivo `tema.breakpoints.yml`). Internamente utiliza el módulo **Breakpoint** (incluido en el módulo y activado por defecto), para que otros módulos puedan utilizar los breakpoints declarados. Como ejemplo mostramos el contenido del archivo `bartik.breakpoints.yml`, del tema Bartik:

YML: `bartik.breakpoints.yml`

```
bartik.mobile:
  label: mobile
  mediaQuery: ''
  weight: 0
  multipliers:
    - 1x
bartik.narrow:
  label: narrow
  mediaQuery: 'all and (min-width: 560px) and (max-width: 850px)'
  weight: 1
  multipliers:
    - 1x
bartik.wide:
  label: wide
  mediaQuery: 'all and (min-width: 851px)'
  weight: 2
  multipliers:
    - 1x
```

En Bartik se declaran tres breakpoints:

- **mobile**. Para anchos menores de 559px (inclusive). Nótese que en este breakpoint no se ha definido una media query específica, lo que implica que se aplicará por descarte o por defecto, cuando no se cumplan las siguientes reglas.
- **narrow**. Para anchos entre 560px y 850px (ambos inclusive)
- **wide**. Para anchos mayores de 851px (inclusive).

Cada breakpoint se define con un nombre de sistema único, añadiendo como prefijo el nombre del tema (o módulo). Las propiedades que debemos especificar son:

- **label**. Etiqueta para identificar al breakpoint. Esta etiqueta se mostrará en la interfaz de Drupal, en los puntos donde se haga uso de estos breakpoints.
- **mediaQuery**. Es la Media Query que define las condiciones de aplicación del breakpoint.
- **weight**. Peso u orden de ejecución del breakpoint. Generalmente siguen el mismo orden en que se han colocado en el archivo de configuración.
- **multipliers**. Los multiplicadores permiten la compatibilidad con dispositivos de mayor resolución (o densidad de pixels), adaptando las medidas a la resolución del dispositivo. Los valores posibles son: 1x (normal), 1.5x (Android) y 2x (Retina display en dispositivos Mac).

Grupos de breakpoints

En el archivo de configuración también podemos definir grupos de breakpoints, siguiendo esta nomenclatura:

- tema.grupo.nombre_breakpoint

Por ejemplo:

- bartik.group1.mobile
- bartik.group1.narrow
- ...
- bartik.group2.mobile
- bartik.group2.narrow

Como ejemplo puedes consultar la definición de breakpoints en el módulo AdaptiveTheme (/themes/adaptivetheme/at_core/at_core.breakpoints.yml), donde se definen los grupos at_core.simple, at_core.advanced y at_core.responsive_menus con diferentes finalidades.

Enlaces de ampliación:

- **Working with breakpoints in Drupal 8**
<https://www.drupal.org/docs/8/theming-drupal-8/working-with-breakpoints-in-drupal-8>

Uso de breakpoints en CSS

No hay una forma directa de utilizar los breakpoints definidos en la configuración de Drupal en las reglas CSS, lo que nos obliga a repetir las mismas Media queries para aplicar estilos en función de esos breakpoints.

En la próxima unidad veremos un método para vincular los archivos Sass a los breakpoints de configuración, de forma que podremos hacer referencia directa a los breakpoints sin tener que repetir las Media queries.

Por ahora, nos quedaremos con la idea de que los breakpoints definidos en el archivo de configuración sólo pueden ser usados por otros módulos, como **Responsive Image** (núcleo) e **Inline Responsive Images** (contribuido). Estudiaremos estos módulos en el próximo apartado.

Breakpoints desde consola

Hay disponibles dos comandos de Drupal Console relacionados con los breakpoints:

drupal debug:breakpoints

Muestra los breakpoints disponibles en el sistema.

Si no especificamos ningún argumento, se muestran todos los grupos de breakpoints existentes:

```
$ drupal debug:breakpoints
Extensions with breakpoints
bartik
responsive_image
seven
toolbar
at core.advanced
at core.responsive menus
at core.simple
```

Si especificamos como argumento el nombre de uno de los grupos, se mostrarán los breakpoints correspondientes:

```
$ drupal debug:breakpoints bartik
bartik.mobile

label: mobile
mediaQuery: ''
weight: 0
multipliers:
  - 1x

bartik.narrow

label: narrow
mediaQuery: 'all and (min-width: 560px) and (max-width: 850px)'
weight: 1
multipliers:
  - 1x

bartik.wide

label: wide
mediaQuery: 'all and (min-width: 851px)'
weight: 2
multipliers:
  - 1x
```

<https://hechoendrupal.gitbooks.io/drupal-console/content/en/commands/debug-breakpoints.html>

drupal generate:breakpoint

Genera un breakpoint especificando los siguientes parámetros:

- Nombre del breakpoint
- Etiqueta
- Media query
- Peso
- Multiplicadores

Si no especificamos estos valores como opciones del comando, el sistema los irá solicitando uno a uno:

```
$ drupal generate:breakpoint

// Welcome to the Drupal breakpoint generator
Enter the theme name (i.e. classy, stable) [classy]:
> b_forcontu

Enter breakpoint name [narrow]:
> narrow

Enter breakpoint label [narrow]:
>

Enter breakpoint media query [all and (min-width: 560px) and
(max-width: 850px)]:
> all and (min-width: 560px) and (max-width: 850px)

Enter breakpoint weight [1]:
> 1

Enter breakpoint multipliers [1x]:
>

Do you want to add another breakpoint? (yes/no) [yes]:
> no

Do you want proceed with the operation? (yes/no) [yes]:
>

Generated or updated files
Generation path: /home/desa1/public_html/fe/fe1/
1 - themes/b_forcontu/b_forcontu.breakpoints.yml
```

El comando crea el archivo `.breakpoints.yml` si éste no existe en el tema. Si el archivo de configuración sí existe, añadirá el nuevo breakpoint sin modificar los ya existentes.

<https://hechoendrupal.gitbooks.io/drupal-console/content/en/commands/generate-breakpoint.html>

Configuración de breakpoints en AdaptiveTheme

Al generar un subtema de AdaptiveTheme no se añade el archivo de configuración breakpoints.yml por defecto, pero podemos utilizar los grupos de breakpoints definidos en at_core.

En at_core se definen tres grupos de breakpoints (at_core.simple, at_core.advanced y at_core.responsive_menus) con diferentes finalidades. En los comentarios se ha añadido una representación gráfica del ancho de cada breakpoint.

Archivo: /themes/adaptivetheme/at_core/at_core.breakpoints.yml

```
#
# Breakpoints - Simple
#
# Pixel values based on 1em = 16px.
#
# mobile - 0 to 45em (0 to 720px), small screens only.
# tablet - 45.0625em and max-width 60em (721px to 960px), isolate tablets.
# wide - 60.0625em (961px) and up.
#
#           45em           60em
# | mobile ----->|
#           | tablet ----->|
#           |           | wide ----->
#
at_core.simple.mobile:
at_core.simple.tablet-isolate:
at_core.simple.wide:

# Breakpoints Advanced
# -----
# Pixel values based on 1em = 16px.
#
# mobile-isolate - 0 to 45em (0 to 720px), small screens only.
# tablet-isolate - 45.0625em and max-width 65em (721px to 960px), isolate tablets.
# wide-isolate - 60.0625em and max-width 120em (961px to 1920px), isolate tablets.
# tablet - 45.0625em (721px) and up.
# wide - 60.0625em (961px) and up.
# huge - 120.0625em (1921px) and up, for big resolution.
#
# Here you can see how each breakpoint cascades or isolates. Most of the time
# a layout will use tablet-isolate and widescreen OR tablet-isolate, widescreen-isolate
# and huge.
#
# | $mobile-isolate -->|
#           | $tablet-isolate -->|
#           |           | $wide-isolate ----->|
#           | $tablet ----->|
#           |           | $wide ----->
#           |           | $huge ----->
#
at_core.advanced.mobile-isolate:
at_core.advanced.tablet-isolate:
at_core.advanced.wide-isolate:
at_core.advanced.tablet:
at_core.advanced.wide:
at_core.advanced.huge:

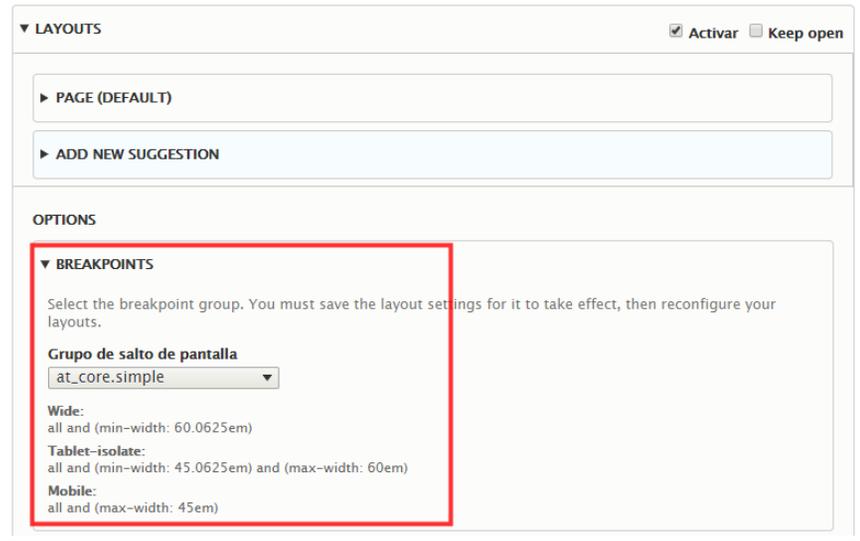
# Breakpoints - Responsive Menu
#
# Pixel values based on 1em = 16px.
#
# mobile - 0 to 60em (0 to 960px)
# maxwidth - 0 to 75em (0 to 1200px)
#
# | mobile ----->|
# | maxwidth ----->|
#
at_core.responsive_menus.mobile:
at_core.responsive_menus.max-width:
```

Por defecto nuestro tema utilizará el grupo `at_core.simple`. Podemos modificar este comportamiento desde la configuración del tema, en selección Layouts **[F8.5a]**.

F8.5a

Breakpoints

Selección del grupo de breakpoints en AdaptiveTheme.



The screenshot shows the AdaptiveTheme configuration interface. At the top, there is a 'LAYOUTS' section with a '▼' icon, a 'Activar' checkbox (checked), and a 'Keep open' checkbox (unchecked). Below this are two buttons: 'PAGE (DEFAULT)' and 'ADD NEW SUGGESTION'. The 'OPTIONS' section is below, with a '▼ BREAKPOINTS' sub-section highlighted by a red box. Inside the red box, there is a text instruction: 'Select the breakpoint group. You must save the layout settings for it to take effect, then reconfigure your layouts.' Below this is a dropdown menu labeled 'Grupo de salto de pantalla' with 'at_core.simple' selected. Underneath, three breakpoint configurations are listed: 'Wide: all and (min-width: 60.0625em)', 'Tablet-isolate: all and (min-width: 45.0625em) and (max-width: 60em)', and 'Mobile: all and (max-width: 45em)'.

Imágenes responsive

8.6

Estilos de imagen responsive

En la unidad 15.4 del libro de Experto en Drupal 8 Site Building I ya vimos cómo crear **estilos de imagen responsive** desde la interfaz. Volveremos a este punto sin entrar en mucho detalle, pero analizando el uso de los breakpoints definidos en configuración.

Drupal 8 incluye en el núcleo el **módulo Responsive Image**, inicialmente desactivado, con el que podemos crear estilos de imagen responsive. De esta forma, el estilo que se aplica a la imagen dependerá del ancho del dispositivo (PC, tablet, móvil, etc.), optimizando tanto la presentación de la imagen como el **tamaño real** del archivo cargado.

El módulo Responsive Image utiliza los breakpoints definidos en el tema a través del archivo de configuración breakpoints.yml, tal y como vimos en el apartado anterior.

Lo que permite el módulo Responsive Image es establecer un estilo de imagen específico para cada punto de ruptura. Por ejemplo, teniendo en cuenta los breakpoints del tema Bartik, podríamos crear el estilo **Portada publicación responsive**, que en realidad engloba los siguientes estilos:

- **Portada publicación pequeño.** Se aplica cuando el ancho es menor de 560px.
- **Portada publicación mediano.** Se aplica para un ancho entre 560px y 850px.
- **Portada publicación grande.** Se aplica para un ancho superior a 851px.

Crearemos primero estos tres estilos de imagen, como estilos de imagen "normales". Una vez creados los estilos de imagen, el siguiente paso es crear el estilo de imagen responsive que los englobará. Lo haremos desde **[F8.6a]**:

Administración ⇒ Configuración ⇒ Multimedia
⇒ Estilos de imagen responsive (o adaptables)

Recuerda que para que esta opción esté disponible primero tienes que activar el módulo **Responsive images** incluido en el núcleo.

Responsive image styles ☆

Inicio » Administración » Configuración » Multimedia

A responsive image style associates an image style with each breakpoint defined by your theme.

+ Add responsive image style

ETIQUETA	NOMBRE DE SISTEMA	OPERACIONES
Todavía no hay Responsive image style.		

F8.6a

Estilos de imagen responsive

Listado de estilos y botón de agregar nuevo estilo de imagen responsive.

Al añadir un nuevo **Estilo de imagen responsive**, indicaremos la etiqueta del estilo, el tema que establece los grupos de breakpoint y el estilo predeterminado en caso de que el sistema no pudiera determinar el ancho de pantalla adecuado. Para cada punto de ruptura indicaremos el estilo simple que se aplicará **[F8.6b]**.

F8.6b

Estilo de imagen responsive

Asignación de los estilos de imagen según los puntos de ruptura definidos en el tema.

Añadir el estilo adaptable a la imagen ☆

Inicio » Administración » Configuración » Multimedia » Estilos adaptables de la imagen.

Etiqueta *

Portada publicación responsive

Nombre de sistema:

portada_publicacion_responsive [Editar]

Por ejemplo: "Imagen del héroe" o "Imagen del autor".

Grupo de salto de pantalla *

Bartik

Seleccionar un grupo de puntos de ruptura desde los temas y módulos instalados.

▼ 1X WIDE [ALL AND (MIN-WIDTH: 851PX)]

Tipo

- Seleccionar múltiples estilos de imágenes y use el atributo de tamaños
- Seleccione un estilo de imagen simple.
- No utilice este punto de ruptura.

Ver la [página de ayuda de Imágenes Adaptables](#) para información acerca del atributo de tamaño.

Estilo de imagen

Portada publicación grande

Seleccione un estilo de imagen para este punto de ruptura.

▼ 1X NARROW [ALL AND (MIN-WIDTH: 560PX) AND (MAX-WIDTH: 850PX)]

Tipo

- Seleccionar múltiples estilos de imágenes y use el atributo de tamaños
- Seleccione un estilo de imagen simple.
- No utilice este punto de ruptura.

Ver la [página de ayuda de Imágenes Adaptables](#) para información acerca del atributo de tamaño.

Estilo de imagen

Portada publicación mediano

Seleccione un estilo de imagen para este punto de ruptura.

▼ 1X MOBILE []

Tipo

- Seleccionar múltiples estilos de imágenes y use el atributo de tamaños
- Seleccione un estilo de imagen simple.
- No utilice este punto de ruptura.

Ver la [página de ayuda de Imágenes Adaptables](#) para información acerca del atributo de tamaño.

Estilo de imagen

Portada publicación pequeño

Seleccione un estilo de imagen para este punto de ruptura.

Estilo de imagen en caso de fallo *

Portada publicación pequeño

Seleccione el estilo de imagen más pequeño que espera que aparezca en este espacio. El estilo de imagen de respaldo solo aparecerá en el sitio si ocurre un error.

Guardar

Y listo, en el listado de estilos de imagen responsive podremos ver el estilo creado **[F8.6c]**:

Responsive image styles ☆

Inicio » Administración » Configuración » Multimedia

A responsive image style associates an image style with each breakpoint defined by your theme.

+ Add responsive image style

ETIQUETA	NOMBRE DE SISTEMA	OPERACIONES
Portada publicación responsive	portada_publicacion_responsive	Editar ▼

F8.6c

Estilo de imagen responsive

Nuevo estilo de imagen responsive. Puede utilizarse en campos como cualquier otro estilo de imagen.

Repasa las unidades **16** y **20** en los libros de Experto en Drupal 8 Site Building I y II, respectivamente, para recordar cómo aplicar estos estilos en campos y vistas.

¿Cómo funciona a nivel de HTML y CSS?

Este es el código completo generado a nivel de HTML para un campo de tipo imagen con el estilo responsive creado anteriormente:

```
<picture>
  <source
srcset="/sites/default/files/styles/portada_publicacion_grande/public/2018-10/sample.jpg?itok=y9WuQ_3Q 1x" media="all and (min-width: 851px)"
type="image/jpeg">
  <source
srcset="/sites/default/files/styles/portada_publicacion_mediano/public/2018-10/sample.jpg?itok=Hd3UnIrC 1x" media="all and (min-width: 560px)
and (max-width: 850px)" type="image/jpeg">
  <source
srcset="/sites/default/files/styles/portada_publicacion_pequeno/public/2018-10/sample.jpg?itok=9jyowjan 1x" type="image/jpeg">
  
</picture>
```

Para entenderlo mejor vamos a analizar una versión simplificada:

```
<picture>
  <source srcset="grande.jpg 1x"
media="all and (min-width: 851px)" type="image/jpeg">
  <source srcset="mediano.jpg 1x"
media="all and (min-width: 560px) and (max-width: 850px)"
type="image/jpeg">
  <source srcset="pequeno.jpg 1x" type="image/jpeg">

  
</picture>
```

La clave para lograr un comportamiento realmente responsive está en el elemento **<picture>**, que puede contener varios **<source>** o rutas diferentes para una misma imagen, que se indican en la propiedad **srcset**. En función de la condición incluida en **media** (Media query generada a partir del breakpoint), el navegador cargará únicamente la imagen correspondiente.

El elemento **** contenido en **<picture>** es obligatorio. Servirá como base para mostrar la imagen y como imagen por defecto en caso de que no se cumpla ninguna de las condiciones definidas.

Imágenes responsive desde el editor de texto

El módulo anterior nos permite crear estilos de imagen responsive y aplicarlos en los elementos de Drupal que aceptan estilos de imagen, como los campos o las vistas. Pero también puede surgir la necesidad de aplicar estilos responsive en imágenes insertadas directamente a través del editor de texto (wysiwyg). Para este propósito podemos instalar el módulo **Inline Responsive Images**, disponible en:

https://www.drupal.org/project/inline_responsive_images

El módulo aún tiene algunos errores importantes que han sido reportados en la cola de issues. Para probar su funcionamiento sigue estos pasos:

- Edita el formato de texto Full HTML (HTML completo) **[F8.6d]**.
- Activa el filtro **Display responsive images**. Es importante que no esté activado al mismo tiempo el filtro **Display image styles** (error conocido).

F8.6d

Módulo Inline responsive images

Configuración del formato de texto. Filtro: Display responsive images.

Filtros activos

- Display image styles
Uses the data-image-style attribute on tags to display image styles.
- Alinear imágenes
Utiliza un atributo data-align en las etiquetas para alinear las imágenes.
- Subtítulos de imágenes
Use un atributo data-caption en las etiquetas para subtitar imágenes.
- Corregir HTML defectuoso o incompleto
- Rastrear las imágenes subidas a través de un editor de texto
Se asegura de que se muestren las últimas versiones de las imágenes subidas mediante un editor de texto.
- Display responsive images
Uses the data-responsive-image-style attribute on tags to display responsive images.
- Limita las etiquetas HTML permitidas y corrige el HTML incorrecto

- Cambia el orden del filtro **Display responsive images**, colocándolo al final de la lista.
- En las opciones del filtro, selecciona los estilos de imagen que se utilizarán **[F8.6e]**.

F8.6e

Módulo Inline responsive images

Configuración del formato de texto. Filtro: Display responsive images.

Orden de procesamiento de los filtros

[Mostrar pesos de la fila](#)

- ⊕ Alinear imágenes
- ⊕ Subtítulos de imágenes
- ⊕ Corregir HTML defectuoso o incompleto
- ⊕ Rastrear las imágenes subidas a través de un editor de texto
- ⊕ Display responsive images

Opciones de filtro

- Display responsive images**
Activado
- Select the responsive styles that are available in the editor
- Narrow
 - Portada publicación responsive
 - Wide

Guardar configuración

Edita cualquier contenido que tenga asociado un cuadro de texto (Cuerpo o similar). Añade una imagen desde el editor y selecciona un estilo de imagen responsive (adaptable) **[F8.6f]**.

**F8.6f****Módulo Inline responsive images**

Configuración del formato de texto. Filtro: Display responsive images.

El HTML generado será similar al que vimos anteriormente, generado por los campos de tipo imagen con estilos de imagen responsive.

Enlaces de ampliación:

- **Responsive Web Design – Images (w3schools.com)**
https://www.w3schools.com/css/css_rwd_images.asp
- **Imágenes adaptables (MDN web docs)**
https://developer.mozilla.org/es/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images

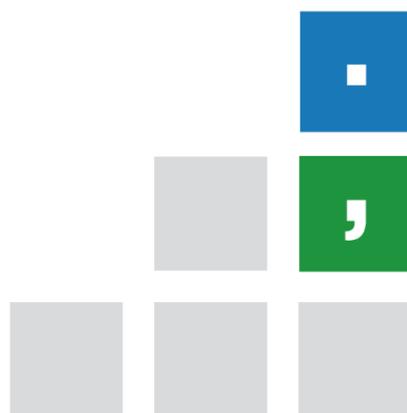
Fran Gil

Front-End Development I

9

Sass

Experto en
DRUPAL 8



Experto en Drupal 8 Front-End Development

Este documento forma parte del material didáctico del curso **Experto en Drupal 8 Front-End Development**. Si deseas adquirir otros materiales complementarios o realizar el curso online y certificarte como **Experto en Drupal 8 Front-End Development**, visita www.forcontu.com o ponte en contacto con nosotros a través del correo info@forcontu.com.

Actualizaciones

Este material será actualizado frecuentemente para nuevas versiones. Si has adquirido la versión digital en www.forcontu.com, tendrás acceso permanente y gratuito a las nuevas versiones.

Reporte de errores

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Licencia de uso exclusivo

Esta copia es de uso exclusivo y particular para el usuario especificado, que podrá almacenarlo, imprimirlo y consultarlo en cuantos dispositivos requiera.

Se prohíbe expresamente el uso compartido del documento, la transferencia a otras personas y la publicación en páginas web o aplicaciones que favorezcan el libre acceso al mismo, ya sean abiertos al público o privados.

Este documento tampoco podrá ser utilizado en acciones formativas, ya sean online o presenciales, que no cuenten con la autorización y reconocimiento de Forcontu S.L.

Copyright © 2018 Forcontu S.L.

Todos los derechos reservados. El contenido de este documento, tanto en su totalidad como parcialmente no puede ser reproducido, almacenado o transmitido de cualquier forma o por cualquier medio ya sea electrónico, mecánico, fotocopiado, grabado o de otra forma, sin la previa autorización expresa y escrita por parte de Forcontu S.L. Se incluye, en particular, su mera reproducción y/o puesta a disposición como resúmenes, reseñas o revistas de prensa, fines para los que también será necesario contar con la correspondiente autorización de Forcontu S.L. Para obtener más información, póngase en contacto a través de info@forcontu.com.

9 Sass

Sass es una ampliación de CSS que permite utilizar, dentro de las hojas de estilos, estructuras adicionales cercanas a las de un lenguaje de programación. De esta forma se pueden construir hojas de estilo mejor organizadas y optimizadas, evitándose el código repetitivo.

El código Sass necesita ser compilado para generar el código CSS final. Los navegadores no interpretan el código Sass. Lo que llega al navegador es el código CSS que ya conocemos, independientemente de cómo haya sido generado en el lado del servidor.

En esta unidad no sólo estudiaremos la sintaxis de Sass (SCSS), sino que veremos otras librerías y herramientas de ampliación, como Compass, Grunt y Gulp. También veremos cómo utilizar estas herramientas en el tema AdaptiveTheme.

Comparativa D8/D7

Todo lo estudiado en esta unidad se pueden aplicar a cualquier versión de Drupal.

Contenidos de la Unidad

- 9.1 Introducción a Sass
- 9.2 Sintaxis de Sass (SCSS)
- 9.3 Instalación y configuración de Sass
- 9.4 Archivos de mapeo en Sass
- 9.5 Compass
- 9.6 Herramientas de automatización: Grunt
- 9.7 Herramientas de automatización: Gulp
- 9.8 Configuración de Sass en AdaptiveTheme

9

9.1 Introducción a Sass

Qué es Sass

Sass es una ampliación de CSS que permite utilizar, dentro de las hojas de estilos, estructuras adicionales cercanas a las de un lenguaje de programación, como variables, reglas anidadas, mixins, funciones, etc. De esta forma se pueden construir hojas de estilo mejor organizadas y optimizadas, evitándose el código repetitivo.

El código Sass necesita ser **compilado** para generar el código CSS final. Los navegadores no interpretan el código Sass. Lo que llega al navegador es el código CSS que ya conocemos, independientemente de cómo haya sido generado en el lado del servidor.

Las hojas de estilo Sass **no contienen contenido dinámico**, así que sólo necesitan compilarse para generar el CSS correspondiente cuando el código Sass es modificado.

Sass y SCSS

En Sass podemos utilizar una de estas dos sintaxis **[F9.1a]**:

- **Sass** (también conocida como sintaxis indentada de Sass). Es la versión más antigua y también la más alejada de CSS. Utiliza indentación para anidar propiedades (en lugar de llaves) y saltos de línea para separar propiedades (en lugar de punto y coma). Los archivos con sintaxis Sass utilizan la extensión **.sass**.

La sintaxis indentada **no es compatible con CSS**, de forma que no podemos incluir CSS directamente dentro de un archivo Sass, sin adaptarlo previamente a su sintaxis.

- **SCSS**. Es la versión más actualizada de Sass. Se asemeja más a CSS, utilizando llaves para las reglas y los anidamientos, y punto y coma para separar las propiedades. Al utilizar la misma sintaxis que CSS, es totalmente compatible con las reglas CSS. Los archivos con sintaxis SCSS utilizan la extensión **.scss**.

F9.1a

Comparativa Sass/SCSS

Comparativa entre las dos sintaxis de Sass disponibles.

Sintaxis Sass	Sintaxis SCSS
<pre>#main p color: #fff width: 95% .box background-color: Red color: #000</pre>	<pre>#main p { color: #fff; width: 95%; .box { background-color: Red; color: #000; }</pre>

En el código Sass/SCSS de ejemplo estamos anidando una regla dentro de otra. En ambos casos, una vez compilado el código Sass, el CSS resultante será el mismo:

CSS:

```
#main p {
  color: #fff;
  width: 95%;
}
#main p .box {
  background-color: Red;
  color: #000;
}
```

En el código CSS generado vemos que las reglas anidadas se han separado en dos reglas independientes. Esto es así porque en CSS no se pueden anidar reglas y los navegadores sólo entienden el código CSS.

¿Qué sintaxis escoger?

La sintaxis SCSS es la más aceptada por su compatibilidad con CSS, aunque muchos desarrolladores prefieren usar Sass indentado por su legibilidad y simplificación del código. Aunque **en este curso utilizaremos la sintaxis SCSS**, siéntente libre de escoger para tus proyectos la sintaxis con la que te sientas más cómodo.

En este enlace puedes ampliar información sobre la utilización del Sass indentado (indentado):

http://sass-lang.com/documentation/file.INDENTED_SYNTAX.html

Probando el código Sass

Como ya hicimos con HTML y CSS, para probar el código Sass/SCSS fuera de Drupal, también podemos utilizar herramientas online.

SassMeister

Para una conversión directa entre Sass/SCSS y CSS, podemos utilizar SassMeister:

<https://www.sassmeister.com/>

F9.1b SassMeister

SassMeister es una herramienta online que nos permite compilar código Sass para obtener el código CSS correspondiente.

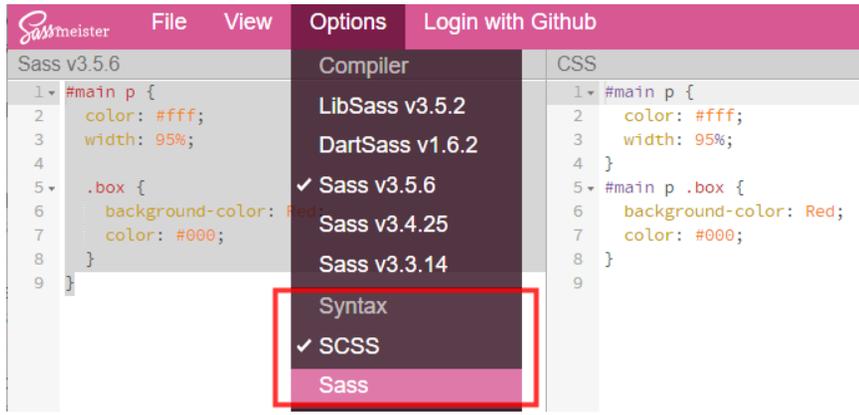
Desde Opciones podemos seleccionar la versión de Sass y la sintaxis a utilizar (SCSS o Sass). Cada vez que hagamos un cambio en la ventana de Sass (izquierda), se realizará la compilación para obtener el CSS resultante (derecha).

Una funcionalidad interesante de esta web es que nos permite también realizar **conversión entre las sintaxis de Sass** (desde SCSS a Sass y viceversa). Para ello seleccionaremos en Opciones la sintaxis de origen (por ejemplo, Sass). Ahora podemos añadir en la ventana de la izquierda código con sintaxis Sass (indentado). Una vez tengamos el código a convertir, cambiaremos en Opciones la sintaxis (a SCSS), y el código de la izquierda se transformará automáticamente a la nueva sintaxis.

F9.1c

SassMeister

Selección de sintaxis de Sass en SassMeister.



CodePen

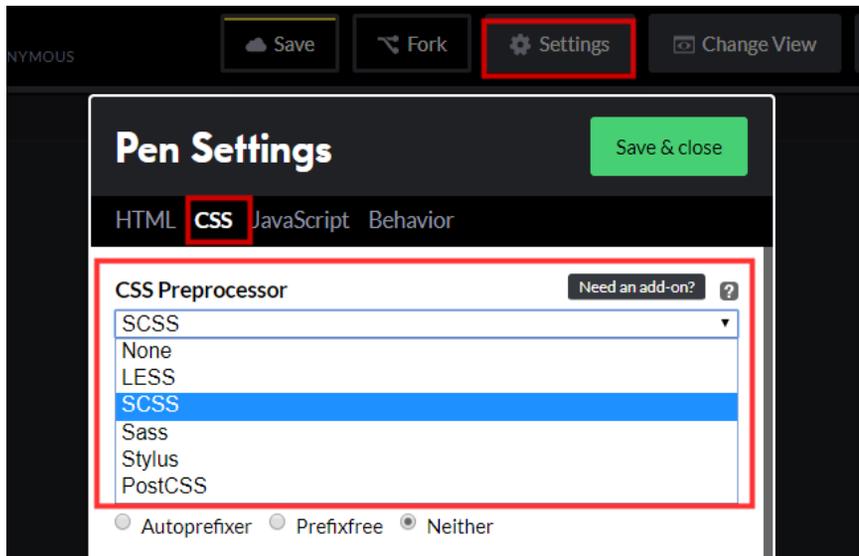
Si queremos una herramienta más completa donde ver el resultado final de aplicar HTML, CSS (a partir de Sass) y JavaScript, podemos volver a utilizar CodePen.

Si vamos a escribir código SCSS (o Sass) en lugar de CSS, cambiaremos el preprocesador CSS desde la pestaña Settings (CSS) [F9.1d].

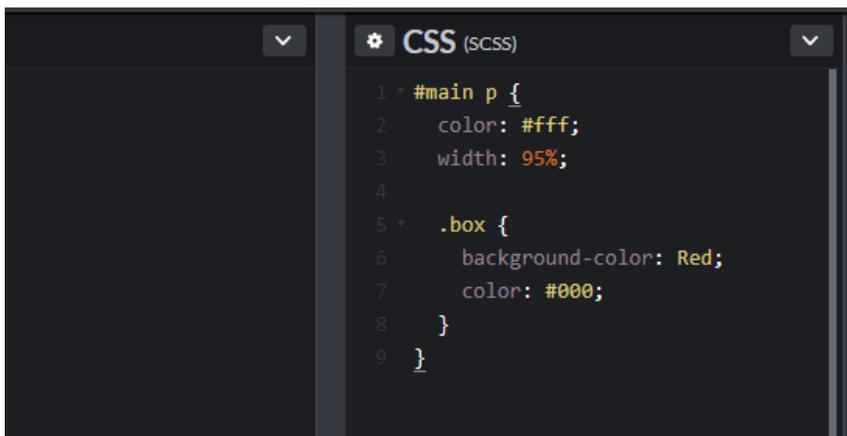
F9.1d

CodePen

Desde CodePen también podemos utilizar Sass/SCSS como alternativa a CSS.



Una vez seleccionado el preprocesador CSS, en la pestaña CSS se mostrará entre paréntesis la sintaxis esperada. Por ejemplo CSS (SCSS) [F9.1e].



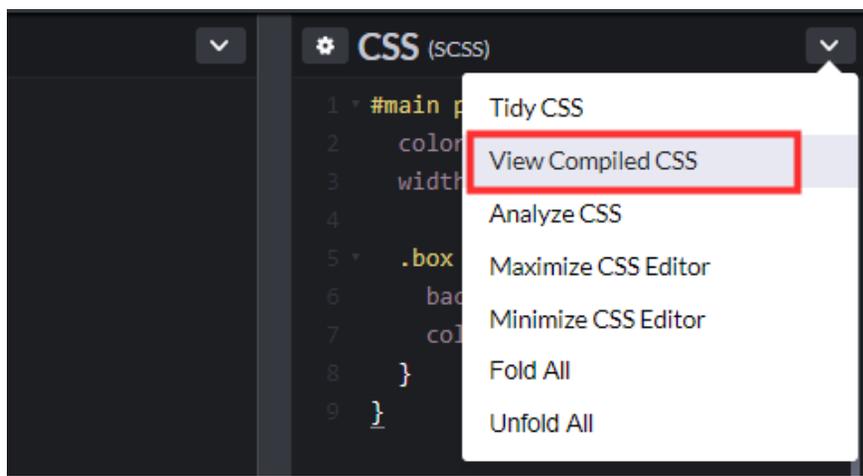
```

1 #main p {
2   color: #fff;
3   width: 95%;
4
5 .box {
6   background-color: Red;
7   color: #000;
8 }
9 }

```

F9.1e
CodePen
Selección de SCSS.

Si queremos ver el CSS resultante una vez compilado el código, podemos cambiar a CSS desde la opción (**View Compiled CSS**). Para volver al código Sass/SCSS, usaremos la opción **View Uncompiled SCSS** [F9.1f].



```

1 #main p
2 color
3 width
4
5 .box
6 bac
7 col
8 }
9 }

```

- Tidy CSS
- View Compiled CSS**
- Analyze CSS
- Maximize CSS Editor
- Minimize CSS Editor
- Fold All
- Unfold All

F9.1f
CodePen
Ver CSS compilado.

Convertir de CSS a Sass/SCSS

También hay aplicaciones para convertir de CSS y Sass/SCSS, como por ejemplo:

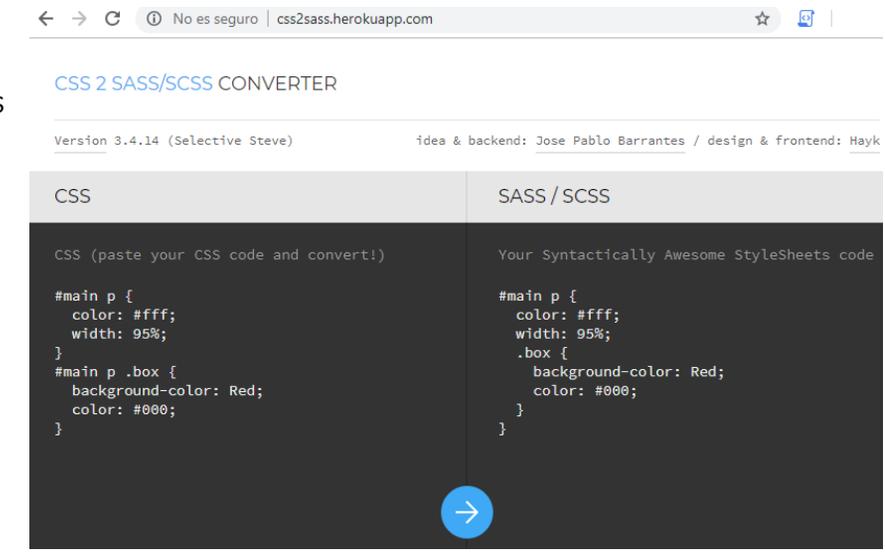
<http://css2sass.herokuapp.com/>

Esta opción puede ser útil para empezar a usar Sass/SCSS en proyectos que vienen de CSS directo.

F9.1g

Convertir CSS a Sass

Conversión inversa, de CSS a Sass/SCSS.



The screenshot shows a web browser window with the URL `css2sass.herokuapp.com`. The page title is "CSS 2 SASS/SCSS CONVERTER". Below the title, it says "Version 3.4.14 (Selective Steve)" and "idea & backend: Jose Pablo Barrantes / design & frontend: Hayk". The main content area is split into two columns: "CSS" and "SASS / SCSS". The "CSS" column contains the text "CSS (paste your CSS code and convert!)" and a code block with the following CSS:

```
#main p {
  color: #fff;
  width: 95%;
}
#main p .box {
  background-color: Red;
  color: #000;
}
```

The "SASS / SCSS" column contains the text "Your Syntactically Awesome StyleSheets code" and a code block with the following SASS/SCSS:

```
#main p {
  color: #fff;
  width: 95%;
}
.box {
  background-color: Red;
  color: #000;
}
```

A blue arrow button is positioned at the bottom center of the code blocks.

Sintaxis de Sass (SCSS)

9.2

En este apartado vamos a introducir los diferentes elementos del código Sass, utilizando la sintaxis SCSS.

Selectores anidados

Sass permite anidar los selectores CSS siguiendo el orden natural de los elementos tal y como los encontramos en el documento HTML.

Ejemplo 1

Comenzamos mostrando una lista de enlaces dentro de un elemento contenedor `<nav>`.

HTML

```
<nav>
  <ul>
    <li><a href="/link1">Link 1</a></li>
    <li><a href="/link2">Link 2</a></li>
    <li><a href="/link3">Link 3</a></li>
  </ul>
</nav>
```

En el código SCSS hemos anidado las etiquetas ``, `` y `<a>` dentro del contenedor `<nav>`.

SCSS	CSS compilado
<pre>nav { ul { margin: 0; padding: 0; list-style: none; } li { display: inline-block; } a { display: block; padding: 6px 12px; text-decoration: none; background-color: #000; color: #fff; } }</pre>	<pre>nav ul { margin: 0; padding: 0; list-style: none; } nav li { display: inline-block; } nav a { display: block; padding: 6px 12px; text-decoration: none; background-color: #000; color: #fff; }</pre>

Ejemplo 2

En el ejemplo anterior era suficiente con anidar los elementos bajo la etiqueta `<nav>`, sin necesidad de llegar a una anidación estricta de todos los elementos. Veamos cómo quedaría el código SCSS si anidamos todas las etiquetas según la anidación en HTML:

SCSS	CSS compilado
<pre> nav { ul { margin: 0; padding: 0; list-style: none; li { display: inline-block; a { display: block; padding: 6px 12px; text-decoration: none; background-color: #000; color: #fff; } } } } </pre>	<pre> nav ul { margin: 0; padding: 0; list-style: none; } nav ul li { display: inline-block; } nav ul li a { display: block; padding: 6px 12px; text-decoration: none; background-color: #000; color: #fff; } </pre>

Ejemplo 3

Otro ejemplo de anidación de selectores. Analiza siempre el CSS resultante tras la compilación del código.

SCSS	CSS compilado
<pre> .box { width: 50%; p, div { font-size: 2em; a { font-weight: bold; } } pre, span { font-size: 3em; } } </pre>	<pre> .box { width: 50%; } .box p, .box div { font-size: 2em; } .box p a, .box div a { font-weight: bold; } .box pre, .box span { font-size: 3em; } </pre>

Referencia al selector padre &

El operador & hace referencia al elemento padre. En el siguiente ejemplo, & hace referencia al selector **a**, por lo que **&:hover** equivale a **a:hover**.

SCSS	CSS compilado
<pre> a { font-weight: bold; text-decoration: none; &:hover { text-decoration: underline; } .special & { font-weight: normal; } } </pre>	<pre> a { font-weight: bold; text-decoration: none; } a:hover { text-decoration: underline; } .special a { font-weight: normal; } </pre>

Cuando el elemento **padre es una clase o un id**, podemos utilizarlo como **prefijo** usando el operador de referencia **&**. En el siguiente ejemplo, **&-first** se sustituirá por **.sidebar-first**:

SCSS	CSS compilado
<pre>.sidebar { color: black; &-first { width: 20%; } &-second { width: 30%; } }</pre>	<pre>.sidebar { color: black; } .sidebar-first { width: 20%; } .sidebar-second { width: 30%; }</pre>

Propiedades anidadas

Sass permite anidar las propiedades CSS de la misma "familia", que son aquellas propiedades que comparten prefijo. Por ejemplo:

- **font:** font-size, font-style, font-weight, font-family, font-variant
- **text:** text-align, text-decoration, text-transform, text-indent, text-shadow, text-overflow.

SCSS	CSS compilado
<pre>.foo { font: { family: Arial; size: 10em; weight: bold; } } .bar { text: { align: left; decoration: underline; transform: uppercase; } }</pre>	<pre>.foo { font-family: Arial; font-size: 10em; font-weight: bold; } .bar { text-align: left; text-decoration: underline; text-transform: uppercase; }</pre>

Comentarios

El código Sass podemos incluir comentarios de varias líneas (*/* */*) o de una línea (*//*). En el código CSS compilado se mantendrán únicamente los comentarios de varias líneas.

SCSS	CSS compilado
<pre>/* Comentario * en varias líneas */ body { width: 100%; } // Comentarios</pre>	<pre>@charset "UTF-8"; /* Comentario * en varias líneas */ body { width: 100%; }</pre>

<pre>// En líneas individuales div { color: Red; }</pre>	<pre>} div { color: Red; }</pre>
--	------------------------------------

Nota: En este caso, como nuestro comentario tiene caracteres especiales (tildes), el sistema ha añadido la directiva **@charset "UTF-8"**; para evitar confusión en la selección del juego de caracteres por defecto.

Variables

En Sass podemos definir variables, que luego podemos reutilizar en el código. Los nombres de variables empiezan por \$.

SCSS	CSS compilado
<pre>\$color: #fe527a; p { color: \$color; } div { background-color: \$color; }</pre>	<pre>p { color: #fe527a; } div { background-color: #fe527a; }</pre>

Nota: Las variables sólo están disponibles en el nivel donde se definen y niveles anidados, pero no en los niveles superiores.

Nombres de selectores en variables #{}

El operador **#{}** nos permite utilizar nombres de selectores y propiedades almacenados en variables.

SCSS	CSS compilado
<pre>\$name: foo; \$attr: border; p.#{ \$name } { #{ \$attr }-color: blue; #{ \$attr }-style: solid; }</pre>	<pre>p.foo { border-color: blue; border-style: solid; }</pre>

Operaciones

Se pueden realizar operaciones de suma (+), resta (-), multiplicación (*), división (/) y módulo o resto (%).

SCSS	CSS compilado
<pre> \$size: 14px; p { font-size: \$size + 2; width: \$size * 10; padding: \$size - 10; } </pre>	<pre> p { font-size: 16px; width: 140px; padding: 4px; } </pre>

División

El operador de división / no siempre es tratado como tal. Los casos en que sí se realiza la división son:

- Si alguno de los valores es una variable o es devuelto por una función.
- Si la operación está dentro de paréntesis (salvo que sean paréntesis para englobar una lista de valores).
- Si el valor se utiliza como parte de otra operación.

SCSS	CSS compilado
<pre> \$width: 1000px; p { width: \$width/2; //a max-width: round(4.1)/2; //a height: (500px/2); //b margin-left: 10px + 8px/2px; //c } </pre>	<pre> p { width: 500px; max-width: 2; height: 250px; margin-left: 14px; } </pre>

En los siguientes ejemplos la división no se realiza y es tratada como texto plano:

SCSS	CSS compilado
<pre> p { font: 10px/5px; } span { font: (italic bold 10px/5px); } </pre>	<pre> p { font: 10px/5px; } span { font: italic bold 10px/5px; } </pre>

Mixins

Los mixins son grupos de declaraciones CSS que podemos reutilizar. Lo podemos comparar con una función, pudiendo incluso aceptar parámetros de entrada para hacer los mixins más flexibles.

```
@mixin nombre(argumentos) {
  declaraciones CSS;
}
```

La forma de utilizar un mixin es mediante la directiva **@include**. El contenido del mixin se colocará justo en el punto donde se ha realizado la llamada:

SCSS	CSS compilado
<pre>@mixin headline (\$color, \$size) { color: \$color; font-size: \$size; } h1 { @include headline(Red, 12px); }</pre>	<pre>h1 { color: Red; font-size: 12px; }</pre>

Argumentos variables ...

Cuando un mixin tiene varios parámetros de entrada, podemos indicar los argumentos en forma de lista (array) o de mapeo de valores, utilizando el operador ...

El mixin espera varios parámetros de entrada, así que con el operador ... lo que le estamos diciendo es que siga asignando valores a esos parámetros a partir de la lista o mapeo de valores facilitada. Cuando se trata de un mapeo de valores, estos pueden estar en un orden diferente al de los parámetros en el mixin.

Veamos un ejemplo con dos variables que utilizaremos como argumentos del mixin:

- \$box-style1 -> lista
- \$box-style2 -> mapeo de valores (clave: valor)

SCSS	CSS compilado
<pre>\$box-style1: 100%, 80px, #ccc; \$box-style2: (background: Red, width: 100%, height: 100px); @mixin box(\$width, \$height, \$background) { width: \$width; height: \$height; background-color: \$background; } .foo { @include box(\$box-style1...); } .bar { @include box(\$box-style2...); }</pre>	<pre>.foo { width: 100%; height: 80px; background-color: #ccc; } .bar { width: 100%; height: 100px; background-color: Red; }</pre>

Al pasar `$box-style1` como lista de parámetros, nos tenemos que asegurar de que los elementos en la variable están definidos en el mismo orden que los parámetros del mixin `box($width, $height, $background)`.

Al pasar `$box-style2`, como se trata de un mapeo de valores, no es necesario seguir el orden.

Valores por defecto en mixins

Al igual que las funciones, los parámetros en los mixins pueden tener valores por defecto, que se utilizarán si no se pasa ese argumento en la llamada al mixin.

En el siguiente ejemplo realizamos dos llamadas, una con los dos parámetros y otra sólo con el parámetro obligatorio:

SCSS	CSS compilado
<pre> @mixin headline (\$color, \$size: 10px) { color: \$color; font-size: \$size; } h1 { @include headline(Red, 12px); } h2 { @include headline(Green); } </pre>	<pre> h1 { color: Red; font-size: 12px; } h2 { color: Green; font-size: 10px; } </pre>

@content

El operador `@content` nos permite añadir contenido o propiedades adicionales al mixin. De esta forma podemos definir un mixin más "abierto" o dinámico.

En el siguiente ejemplo el mixin `media()` añade una media-query con las propiedades adicionales que queramos añadir, y que incluiremos en el momento de la llamada, dentro del `@include`:

SCSS	CSS compilado
<pre> @mixin media(\$width) { @media only screen and (max-width: \$width) { @content; } } @include media(320px) { background: red; } </pre>	<pre> @media only screen and (max- width: 320px) { background: red; } </pre>

Herencia (@extend)

Las clases definidas con **%** (%name) son un conjunto de propiedades CSS que podemos reutilizar mediante la cláusula **@extend**. Si definimos una clase que no se llega a utilizar (con @extend), no se mostrará en el código CSS compilado.

Como ejemplo hemos definido la clase **%message**, que extendemos en varias ocasiones:

SCSS	CSS compilado
<pre> message { border: 1px solid #ccc; padding: 10px; color: #333; } .status { @extend message; } .success { @extend message; border-color: green; } .error { @extend message; border-color: red; } .warning { @extend message; border-color: yellow; } </pre>	<pre> .status, .success, .error, .warning { border: 1px solid #ccc; padding: 10px; color: #333; } .success { border-color: green; } .error { border-color: red; } .warning { border-color: yellow; } </pre>

clearfix hack con Sass

En el **apartado 8.3** vimos el uso del **clearfix hack**, que en CSS se generaba con este código:

```

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

```

Vamos a analizar varios métodos con Sass para generar el mismo clearfix hack, pero de una forma más eficiente y, por supuesto, reutilizable.

Método 1: Definiendo un mixin.

Definimos el mixin con las propiedades anteriores. Ten en cuenta que el selector `::after` debe referenciar al elemento padre o contenedor desde donde hacemos la llamada del mixin. Por ello usamos el operador `&`. En el siguiente ejemplo hacemos el `@include` del mixin dentro del contenedor `.wrap`, que será el elemento padre:

SCSS	CSS compilado
<pre> @mixin clearfix { &::after { content: ""; display: table; clear: both; } } .wrap { border: 1px solid #ccc; @include clearfix; } </pre>	<pre> .wrap { border: 1px solid #ccc; } .wrap::after { content: ""; display: table; clear: both; } </pre>

Método 2: Usando `@extend` con una regla "normal"

En este ejemplo, en lugar de un mixin definimos una regla CSS normal, que luego extenderemos con `@extend`. También utilizamos el operador `&` para referenciar al elemento padre.

En este caso podemos utilizar directamente la clase **clearfix** dentro del código HTML, o extenderla para asociar su contenido a cualquier otro contenedor.

SCSS	CSS compilado
<pre> .clearfix { &::after { content: ""; display: table; clear: both; } } .wrap { border: 1px solid #ccc; @extend .clearfix; } .main-header { border: none; @extend .clearfix; } </pre>	<pre> .clearfix::after, .wrap::after, .main- header::after { content: ""; display: table; clear: both; } .wrap { border: 1px solid #ccc; } .main-header { border: none; } </pre>

Método 3: Usando @extend con una clase %

La ventaja con respecto al método 2 es que la clase definida no aparece en el CSS si no se extiende en el código. El CSS resultante también es más limpio, al no generarse la clase `.clearfix`. Sólo se incluye contenido de la regla en los contenedores donde se vaya a utilizar.

SCSS	CSS compilado
<pre> %clearfix { &::after { content: ""; display: table; clear: both; } } .wrap { border: 1px solid #ccc; @extend %clearfix; } .main-header { border: none; @extend %clearfix; } </pre>	<pre> .wrap::after, .main-header::after { content: ""; display: table; clear: both; } .wrap { border: 1px solid #ccc; } .main-header { border: none; } </pre>

Funciones

Funciones personalizadas

Podemos definir funciones personalizadas mediante la cláusula `@function`. La función devuelve el resultado con `@return`. En el código CSS compilado sólo se mostrará el resultado de la función.

SCSS	CSS compilado
<pre> @function column-width(\$col-span) { @return 100% / \$col-span; } .column-2 { width: column-width(2); } .column-3 { width: column-width(3); } .column-4 { width: column-width(4); } </pre>	<pre> .column-2 { width: 50%; } .column-3 { width: 33.3333333333%; } .column-4 { width: 25%; } </pre>

Las funciones devuelven un valor calculado, mientras que los mixins devuelven un conjunto de propiedades.

Librería de funciones

Sass cuenta con una librería de funciones propias, recogidas en su API:

<http://sass-lang.com/documentation/Sass/Script/Functions.html>

Algunas de estas funciones, agrupadas por funcionalidad, son:

- **Tratamiento de colores:** hsl(), hue(), saturation(), lightness(), darken(), etc.
- **Funciones de cadenas:** str-length(), to-upper-case(), etc.
- **Funciones de números:** round(), ceil(), floor(), random(), etc.
- **Funciones de listas:** length(), nth(), join(), append(), etc.

Ejemplo 1

La función `darken()` oscurece el color dado como parámetro. Podemos indicar el porcentaje en que se reducirá la luminosidad.

SCSS	CSS compilado
<pre> \$color: #3d42ff; .color { background: \$color; } .color-dark { background: darken(\$color, 20%); } </pre>	<pre> .color { background: #3d42ff; } .color-dark { background: #0006d6; } </pre>

Ejemplo 2

Las funciones `min()` y `max()` devuelven los valores mínimo y máximo de una lista. Recuerda utilizar el operador **puntos suspensivos (...)** para que la lista `$numbers` se acepte correctamente como argumento de la función.

SCSS	CSS compilado
<pre> \$numbers: 10px, 20px, 50px, 100px; .box { width: min(\$numbers...); height: max(\$numbers...); } </pre>	<pre> .box { width: 10px; height: 100px; } </pre>

Condicional @if

En Sass también podemos utilizar condiciones y bucles.

La cláusula @if comprueba si se cumple una condición, realizándose las acciones incluidas entre llaves. En el siguiente ejemplo damos un valor por defecto a la variable \$resultado. Si la condición del @if se cumple, el valor se sobrescribe, siendo ese el resultado devuelto por la función.

SCSS	CSS compilado
<pre> @function column-width(\$col-span) { \$resultado: 5%; @if \$col-span < 4 { \$resultado: 100% / \$col-span; } @return \$resultado; } .colum-2 { Width: column-width(2); } .colum-10 { Width: column-width(10); } </pre>	<pre> .colum-2 { Width: 50%; } .colum-10 { Width: 5%; } </pre>

Recuerda que se trata de código "estático", que se calculará en el momento de compilar el código Sass para obtener un código CSS fijo.

Bucles @for y @while

Los bucles @for y @while nos permiten generar de una forma más optimizada código CSS repetitivo. Por ejemplo:

SCSS	CSS compilado
<pre> @for \$i from 1 through 4 { .grid-#{ \$i } { width: 2em * \$i; } } </pre>	<pre> .grid-1 { width: 2em; } .grid-2 { width: 4em; } .grid-3 { width: 6em; } .grid-4 { width: 8em; } </pre>

Existen dos variantes de @for:

- @for \$var from <start> **through** <end>. **Incluye** el valor final.
- @for \$var from <start> **to** <end>. **No incluye** el valor final.

La estructura @while necesita una variable contador (\$i) que servirá para salir del bucle:

SCSS	CSS compilado
<pre> \$i: 1; @while \$i < 4 { .block-#{ \$i } { width: 2em * \$i; } \$i: \$i + 1; } </pre>	<pre> .block-1 { width: 2em; } .block-2 { width: 4em; } .block-3 { width: 6em; } </pre>

Depuración

@debug

La directiva **@debug** nos permite conocer el valor de una variable, función o expresión. Esta información no se mostrará en el CSS generado, sino en la consola desde donde se realiza la compilación de Sass.

Para probar su funcionamiento, abre la consola de CodePen (enlace a Console en la esquina inferior izquierda). En el ejemplo de la Figura **[F9.2a]** comprobamos el valor devuelto por una función dado un parámetro determinado:

```
@debug column-width(8);
```

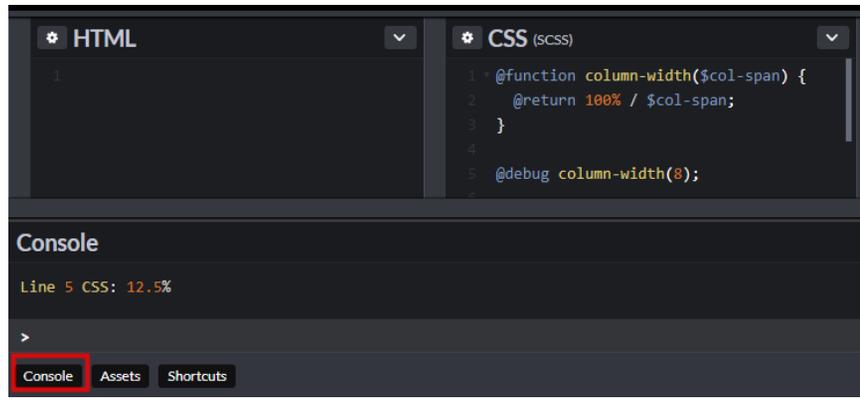
El resultado se muestra en la consola, indicando la línea donde se devuelve el mensaje de depuración:

```
Line 5 CSS: 12.5%;
```

En la función de depuración también podemos añadir texto para facilitar la depuración:

F9.2a**Depuración en Sass**

Podemos conocer el valor de una variable o expresión mediante la directiva `@debug`.

**SCSS:**

```

@function column-width($col-span) {
  @return 100% / $col-span;
}

@debug "column-width(8): " + column-width(8);
@debug "column-width(10): " + column-width(10);

```

Siendo el resultado devuelto por consola:

Consola:

```

Line 5 CSS: column-width(8): 12.5%
Line 6 CSS: column-width(10): 10%

```

@warn

La directiva `@warn` nos permite lanzar advertencias por consola. Los warning no afectan a la compilación, generándose el CSS con normalidad.

SCSS:

```

@function column-width($col-span) {
  @if ($col-span > 8) {
    @warn "No se recomienda usar más de 8 columnas";
  }
  @return 100% / $col-span;
}

.nine-col {
  width: column-width(9);
}

```

Consola:

```

Line 4 CSS: No se recomienda usar más de 8 columnas

```

@error

La directiva @error nos permite lanzar un error y evitar que el código Sass se compile.

SCSS:

```
@function column-width($col-span) {  
  @if ($col-span == 0) {  
    @error "No puede tener 0 columnas";  
  }  
  @return 100% / $col-span  
}  
  
.two-col {  
  width: column-width(0);  
}
```

Devolverá el error:

No puede tener 0 columnas

Enlaces de ampliación:

- Sass Basics (página oficial)
<http://sass-lang.com/guide>
- Sass Reference (página oficial)
http://sass-lang.com/documentation/file.SASS_REFERENCE.html
- Nine Ways to Write the Best SASS Code (Meenakshi Gupta, 2018)
<https://opensenselabs.com/blog/tech/nine-ways-write-best-sass-code>

9.3 Instalación y configuración de Sass

Una vez estudiada la sintaxis de Sass, vamos a ver cómo aplicarlo en un proyecto real. Nos centraremos en la instalación y utilización de Sass desde la línea de comandos.

Instalación de Sass

Consulta aquí los pasos a seguir para instalar Sass en tu equipo o servidor:

<http://sass-lang.com/install>

En los servidores de Forcontu ya encontrarás Sass instalado. Puedes comprobarlo entrando por consola al servidor (SSH) y ejecutando el comando:

```
sass --version
```

Compilación de archivos Sass

El comando `sass` también nos permite compilar un archivo sass para obtener el archivo CSS resultante. Sólo tenemos que indicar la ruta de origen y la ruta de destino:

```
sass input.scss output.css
```

Como estamos trabajando con la sintaxis SCSS, también tenemos disponible el comando `scss`. Es exactamente igual al comando `sass` pero asumiendo por defecto que la sintaxis es SCSS.

```
scss input.scss output.css
```

Por otra parte, también podemos utilizar `--scss` con el comando `sass` para indicar de forma explícita que la sintaxis a usar es SCSS. Si los archivos de origen tienen extensión `.scss`, `sass` determina automáticamente la sintaxis a utilizar.

```
sass --scss input.scss output.css
```

watch

La opción `--watch` mantiene a `sass` a la espera de cambios en los archivos `scss`. Cuando detecta un cambio, realiza la compilación automáticamente. Se puede "vigilar" un único archivo:

```
sass --watch input.scss output.css
```

O especificar una carpeta completa (origen y destino):

```
sass --watch ruta/sass:ruta/css
```

Al lanzar este comando desde consola, el sistema localizará los archivos de origen y destino y se quedará esperando por posibles cambios en el SCSS. Podemos parar de vigilar cambios con Control-C.

```
sass --watch scss:css
>>> Sass is watching for changes. Press Ctrl-C to stop.
      write css/style.css
      write css/style.css.map
```

Si realizamos algún cambio en los archivos SCSS, el comando lo detectará y mostrará un mensaje indicando que los archivos CSS han sido modificados. El comando seguirá activo, a la espera de nuevos cambios.

```
sass --watch scss:css
>>> Sass is watching for changes. Press Ctrl-C to stop.
      write css/style.css
      write css/style.css.map
>>> New template detected: scss/style.scss
      write css/style.css
      write css/style.css.map
```

Aunque este comando permite agilizar la compilación de archivos Sass a CSS, nos obliga a abrir la consola y ejecutarlo cada vez que vayamos a comenzar una sesión de trabajo editando archivos Sass. En el **apartado 9.6** veremos cómo automatizar la compilación y otras tareas con herramientas adicionales como Grunt y Gulp.

Convertir archivos CSS a SCSS

Si partimos de un portal (o tema de Drupal) implementado exclusivamente con CSS, pero queremos empezar a trabajar con Sass, tendremos que convertir primero los archivos CSS a Sass/SCSS.

Podemos realizar esta conversión con el comando sass-convert:

<https://www.npmjs.com/package/sass-convert>

Algunas de las opciones que utilizaremos son:

- **--from css**
Indica el formato de origen, css. También acepta los formatos scss y sass.
- **--to scss**
Indica el formato de destino, scss. También acepta el formato sass.
- **-R origen destino**
--recursive, para convertir todos los archivos de un directorio. A continuación debemos especificar la carpeta de origen, con los archivos CSS a convertir, y la carpeta de destino, donde se almacenarán los archivos SCSS generados.

```
sass-convert --from css --to scss -R css scss
  directory scss
  convert css/style.css
  create scss/style.scss
```

Una vez generados los archivos SCSS, ya no volveremos a tocar directamente los archivos CSS del tema. Los cambios siempre se realizarán en Sass y la compilación regenerará los archivos CSS del tema.

Nota: El comando `sass-convert` también puede ser utilizado para convertir entre sintaxis de Sass. Es especialmente útil si nos encontramos un proyecto donde se ha utilizado la sintaxis indentada de Sass y queremos empezar a utilizar la sintaxis SCSS (o viceversa).

Estructura de archivos y partials

El código Sass se puede dividir en archivos parciales llamados **partials**, que podemos incluir desde otros archivos Sass. Es una forma de mantener el código separado y ordenado, evitando archivos con demasiado grandes.

Los `partials` se diferencian de otros archivos Sass porque tienen como prefijo un guión bajo (`_nombre.scss`). Es la forma de comunicarle a Sass que estos archivos no deben compilarse directamente y, por tanto, no se generarán los archivos CSS equivalentes.

Podemos tener uno o varios archivos principales que importarán los archivos `partials` definidos mediante la directiva `@import`. Sass sólo compilará estos archivos principales, generando los archivos CSS correspondientes.

Volviendo a la metodología SMACSS estudiada, podemos organizar los `partials` en las carpetas:

- scss/base
- scss/components
- scss/layout
- scss/theme

Supongamos, por ejemplo, la siguiente estructura de archivos y carpetas:

```
scss/base/                                # Base
  _reset.scss
  _typography.scss
  ...

scss/components/                          # Components
  _buttons.scss
  _breadcrumb.scss
  _dialog.scss
  ...

scss/...                                  # etc.

scss/main.scss                            # Archivo SCSS principal
```

En el archivo principal `main.scss` llamaremos a los `partials` usando la directiva **@import**. No es necesario indicar ni el prefijo `_` ni la extensión de los `partials`:

```
//main.scss

/* base */
@import "base/reset";
@import "base/typography";

/* components */
@import "components/buttons";
@import "components/breadcrumb";
@import "components/dialog";
```

En este ejemplo sólo se generará un archivo **main.css** que incluirá todo el código compilado.

9.4 Archivos de mapeo en Sass

Los archivos de mapeo (source maps) guardan la relación entre cada archivo CSS generado y su archivo Sass de origen. Es la forma de saber desde qué parte del código Sass se ha generado cada regla del CSS.

En cada archivo CSS generado desde Sass, se añade una línea indicando la ubicación del archivo de mapeo:

```
/*# sourceMappingURL=style.css.map */
```

Los archivos de mapeo tienen el nombre del archivo CSS más la extensión .map.

```
style.css => style.css.map
base.css => base.css.map
```

Estos archivos de almacenan por defecto en la misma carpeta que los archivos CSS y su contenido tiene este formato:

```
{
  "version": 3,
  "mappings":
    "AAAA,OAAQ;EACN,gBAAgB,EAAE,OAAO;EACzB,gBAAgB,EAAE,IAAI;;AAGxB,sBA
    AuB;EACrB,KAAK,EAAE,IAAI;;AAGb,sBAAuB;EACrB,KAAK,EAAE,IAAI;;AAGb,W
    AAY;EACV,gBAAgB,EAAE,OAAO;;AAG3B,UAAW;EACT,MAAM,EAAE,cAAc;EACTB,OA
    AO,EAAE,GAAG;;AAMZ,YAAiB;EAAE,KAAK,EAAE,GAAG;;AAALC,YAAiB;EAAE,KAA
    K,EAAE,GAAG;;AAALC,YAAiB;EAAE,KAAK,EAAE,GAAG",
  "sources": ["../scss/style.scss"],
  "names": [],
  "file": "style.css"
}
```

Si el archivo de mapeo no se crea automáticamente, se puede añadir la directiva `--sourcemap` al comando sass:

```
sass --sourcemap sass/styles.scss public/styles.css
```

Depuración Sass desde el navegador

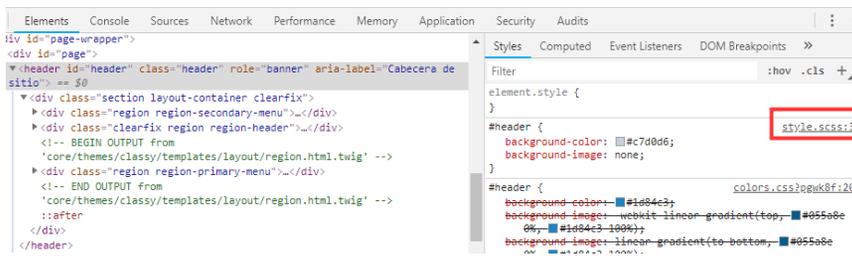
En el apartado 3.3 vimos que con el **inspector de elementos en Chrome** (y otros navegadores), es posible averiguar qué reglas de estilos CSS están actuando sobre cada elemento HTML.

Cuando trabajamos con Sass, también necesitamos conocer qué regla de Sass es la que ha generado ese CSS, ya que las modificaciones las haremos a nivel de Sass. Recuerda que el CSS se genera automáticamente tras compilar el código Sass.

Los navegadores actuales utilizan directamente los archivos de mapeo para mostrarnos esta información.

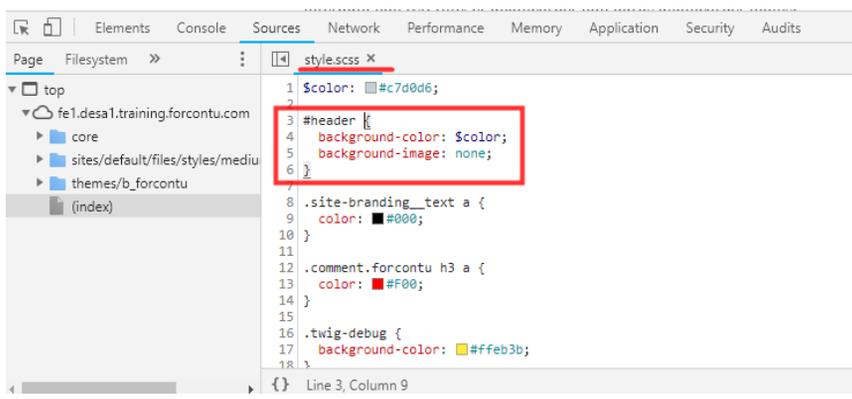
Desde Chrome

En Chrome, el uso de archivos de mapeo de Sass está activado por defecto. Cuando una regla de estilos está definida en un archivo Sass, se mostrará el archivo y la línea donde se encuentra **[F9.4a]**.



F9.4a
Sass en Chrome
 Inspección de estilos CSS con acceso al archivo fuente en Sass.

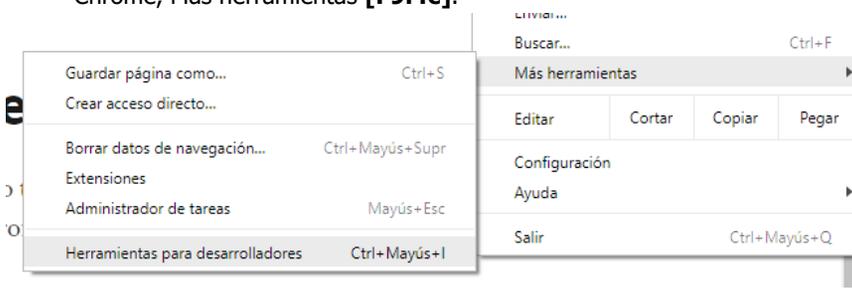
Haciendo clic sobre el nombre del archivo, se mostrará el código Sass (o SCSS) fuente **[F9.4b]**.



F9.4b
Sass en Chrome
 Inspección de estilos CSS con acceso al archivo fuente en Sass.

Si la opción de trabajar con archivos de mapeo no está activada, podemos activarla siguiendo estos pasos:

1. Abre las herramientas para desarrolladores desde el menú principal de Chrome, Más herramientas **[F9.4c]**.



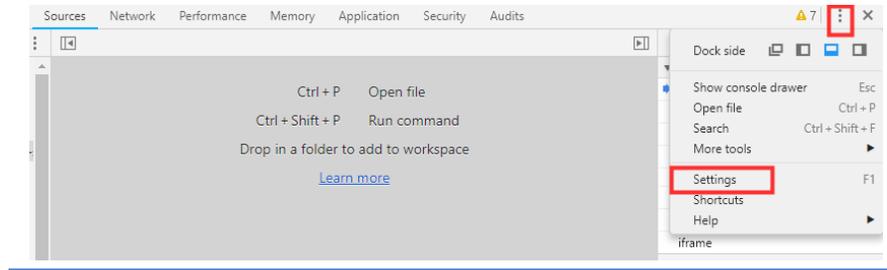
F9.4c
Sass en Chrome
 Configuración de archivos de mapeo CSS/Sass.

2. Despliega el menú en la esquina superior derecha (tres puntos verticales), y accede a **Settings [F9.4d]**.

F9.4d

Sass en Chrome

Configuración de archivos de mapeo CSS/Sass.

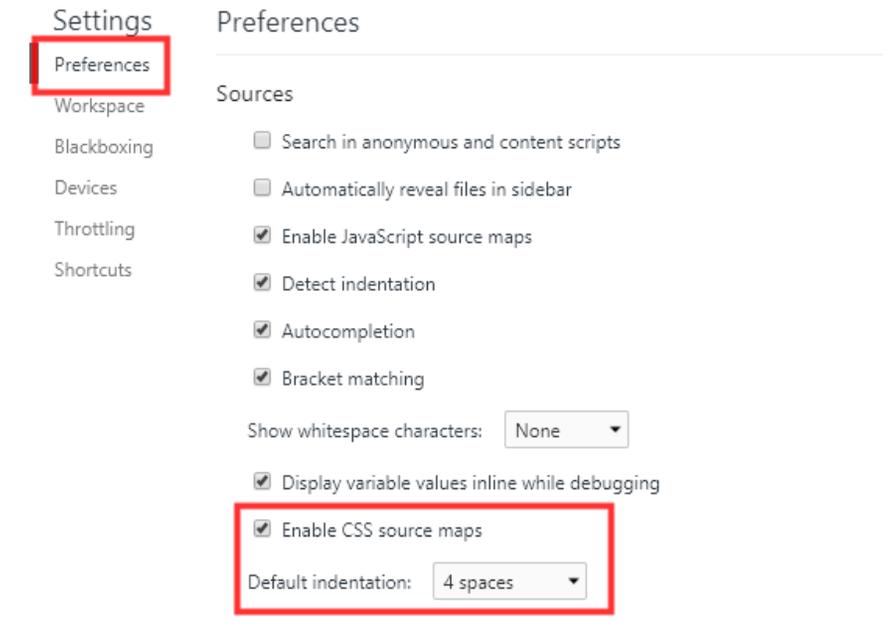


3. Desde **Preferences**, activa la opción "**Enable CSS source maps**", dentro del grupo de opciones **Sources** [F9.4e].

F9.4e

Sass en Chrome

Configuración de archivos de mapeo CSS/Sass.



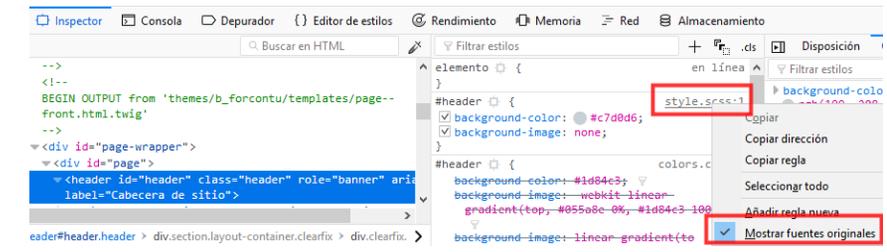
Desde Firefox

En Firefox, el uso de archivos de mapeo Sass también está activado por defecto. Al mostrarse el nombre del archivo Sass, desplegando el menú con botón derecho, comprueba que esté activada la opción **Mostrar fuentes originales** [F9.4f].

F9.4f

Sass en Firefox

Inspección de estilos CSS con acceso al archivo fuente en Sass.



Compass

9.5

Qué es Compass

Compass es una extensión de Sass que añade funcionalidad adicional para automatizar tareas repetitivas. Compass se puede considerar un framework con una librería propia de **Variables configurables**, **Constantes**, **Funciones** y **Mixins**, que podremos utilizar directamente dentro de Sass para ahorrar código y evitar duplicidades.

La página oficial de Compass es: <http://compass-style.org/>

Desde la opción de menú **Code Reference** podemos consultar toda la librería:

<http://compass-style.org/reference/compass/>

Podemos seguir uno de estos dos caminos:

- 1) Funcionalidades de Compass organizadas por categorías (Browser Support, CSS3, Helpers, Layout, Reset, Typography y Utilities). Utiliza el menú horizontal o lateral Compass Core.
- 2) Listado completo en orden alfabético de Variables, Mixins y Funciones. Sigue los enlaces en la sección **Alphabetical index of all...**

Instalación y configuración de Compass

Consulta aquí los pasos a seguir para instalar Compass en tu equipo o servidor:

<http://compass-style.org/install/>

En los servidores de Forcontu ya encontrarás Compass instalado. Puedes comprobarlo entrando por consola al servidor (SSH) y ejecutando el comando:

```
compass version
```

Crear o inicializar el proyecto

Una vez instalado, el primer paso es crear un proyecto de Compass. Se puede usar dos comandos:

- **compass create**, para crear un proyecto nuevo.
- **compass init**, para inicializar un proyecto existente.

Como estamos trabajando sobre un tema existente, utilizaremos este último comando. Lo ejecutaremos **desde la carpeta del tema**:

```
compass init --css-dir=css --sass-dir=scss --javascript-dir=js
           --syntax=scss --prepare

  create config.rb
  modified config.rb
  clean css
  delete css/style.css
  write css/style.css

*****
Congratulations! Your compass project has been created.
```

En los parámetros adicionales del comando hemos indicado las carpetas donde se almacenan los diferentes tipos de archivos:

```
--css-dir=css, ubicación de archivos CSS
--sass-dir=scss, ubicación de archivos SCSS
--javascript-dir=js, ubicación de archivos JavaScript
```

También podemos indicar la sintaxis, scss o sass, aunque scss es la sintaxis por defecto:

```
--syntax=scss
```

Por último, indicamos que se cree únicamente el archivo de configuración de compass con la directiva:

```
--prepare
```

De no incluirla, dentro de las carpetas css y scss se crearán archivos adicionales para utilizar de base (ie.css, print.css y screen.css). Ten en cuenta que estos archivos adicionales no se tendrán en cuenta en Drupal si no se registran en la librería del tema (archivo .libraries.yml).

Archivo de configuración

Al iniciar o crear un proyecto se creará el archivo de configuración de Compass, llamado **config.rb**. Es un archivo de texto que podemos editar si necesitamos modificar algún parámetro de configuración (carpetas donde se ubican los archivos css, sass, imágenes y javascript, librerías adicionales de Compass, etc.)

```
require 'compass/import-once/activate'
# Require any additional compass plugins here.

# Set this to the root of your project when deployed:
http_path = "/"
css_dir = "css"
sass_dir = "scss"
images_dir = "images"
javascripts_dir = "js"

# You can select your preferred output style here (can be
# overridden via the command line):
# output_style = :expanded or :nested or :compact or :compressed

# To enable relative paths to assets via compass helper functions.
# Uncomment:
```

```
# relative_assets = true

# To disable debugging comments that display the original location
of your selectors. Uncomment:
# line_comments = false

preferred_syntax = :scss
```

Archivos de mapeo

Para que se generen los archivos de mapeo entre Sass y CSS, tenemos que añadir al archivo de configuración la directiva:

```
sourcemap = true
```

Compilar archivos de Sass a CSS

Para compilar los archivos Sass con Compass, usaremos el comando:

```
compass compile ruta
```

Si ejecutamos el comando desde el directorio principal del proyecto Compass, no es necesario indicar la ruta:

```
compass compile
```

El comando indicará los archivos modificados, incluyendo los archivos de mapeo si se ha establecido así en la configuración:

```
compass compile
clean css
delete css/style.css
delete css/style.css.map
write css/style.css
write css/style.css.map
```

Con la configuración por defecto, en los archivos CSS generados veremos estas líneas de comentarios que apuntan al código Sass encargado de generar cada regla CSS:

```
/* line 5, ../scss/style.scss */
#header {
  background-color: #c7d0d6;
  background-image: none;
}

/* line 10, ../scss/style.scss */
.site-branding__text a {
  color: #000;
}
```

Si no necesitas que se muestre esta información, descomenta la siguiente línea del archivo de configuración, eliminando la #:

```
# To disable debugging comments that display the original location
of your selectors. Uncomment:
# line_comments = false
```

La línea quedará de esta forma:

```
# To disable debugging comments that display the original location
of your selectors. Uncomment:
line_comments = false
```

Tras compilar nuevamente, no se añadirán las líneas de comentarios anteriores:

```
#header {
  background-color: #c7d0d6;
  background-image: none;
}

.site-branding__text a {
  color: #000;
}
```

Watch

La opción **watch** mantiene a compass a la espera de cambios en los archivos scss. Cuando detecta un cambio, realiza la compilación automáticamente. Compass vigila la carpeta completa de archivos scss, definida en la configuración:

```
compass watch
>>> Compass is watching for changes. Press Ctrl-C to Stop.
```

Cuando detecta un cambio, compila los archivos Sass modificados y regenera los archivos CSS, mostrando un mensaje con los archivos afectados:

```
compass watch
>>> Compass is watching for changes. Press Ctrl-C to Stop.
  created scss/style.scss
  write css/style.css
```

El comando seguirá activo, a la espera de nuevos cambios, hasta que lo detengamos pulsando Control-C.

Otros comandos

Puedes consultar otros comandos de consola de Compass en:

<http://compass-style.org/help/documentation/command-line/>

Probando Compass

Podemos probar Compass desde **CodePen**, utilizando la misma configuración para introducir SCSS que ya estudiamos en esta unidad. Para que las funcionalidades de Compass estén disponibles es necesario importar la librerías correspondientes, añadiendo esta línea al principio del código SCSS:

```
@import "compass";
```

Como ejemplo vamos a utilizar el mixin **border-radius** incluido en Compass:

SCSS (+ Compass)

```
@import "compass";

.simple {
  @include border-radius(4px, 4px);
}
```

Cambiando a **View Compiled CSS** veremos el CSS resultante:

CSS compilado

```
.simple {
  -moz-border-radius: 4px / 4px;
  -webkit-border-radius: 4px 4px;
  border-radius: 4px / 4px;
}
```

También podemos **añadir librerías más específicas**, en función de los componentes de Compass que vayamos a utilizar. Para ello consultaremos en la página de la API de Compass la librería necesaria. Por ejemplo, la descripción del mixin border-radius se encuentra en:

http://compass-style.org/reference/compass/css3/border_radius/

En esta página encontraremos la librería necesaria para este componente **[F9.5a]**:

Compass Border Radius [Source on Github](#)

The border-radius mixin is used to give a block element rounded corners. It automatically outputs the correct vendor specific syntax for each browser (e.g. `-webkit-border-radius` and `-moz-border-radius`). See [CSS3 spec: border-radius](#).

This file can be imported using: `@import "compass/css3/border-radius"`

F9.5a

API de Compass

Localización de la librería necesaria para utilizar un componente específico de la API de Compass.

SCSS (+ Compass)

```
@import "compass/css3/border-radius";

.simple {
  @include border-radius(4px, 4px);
}
```

Variables configurables

Según la librería y componentes de Compass que estemos utilizando, podemos hacer uso de determinadas **variables configurables**. Se trata de variables con valores por defecto que podemos sobrescribir en nuestro código. Generalmente estas variables son utilizadas por mixins o funciones de Compass.

Como ejemplo vamos a analizar las variables configurables asociadas al mixin **border-radius**:

http://compass-style.org/reference/compass/css3/border_radius/

La variable **\$default-border-radius** permite almacenar una longitud de radio por defecto, que es de **5px**. Cuando llamamos al mixin `border-radius()` sin pasar ningún parámetro, se usará este valor por defecto.

SCSS (+ Compass)

```
@import "compass/css3/border-radius";

.simple {
  @include border-radius;
}
```

CSS compilado

```
.simple {
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  border-radius: 5px;
}
```

Para sobrescribir el valor de la variable, tenemos que **asignarle un valor antes de importar la librería**:

SCSS (+ Compass):

```
$default-border-radius: 10px;
@import "compass/css3/border-radius";

.simple {
  @include border-radius;
}
```

Ahora, siempre que el sistema utilice esa variable, utilizará el nuevo valor sobrescrito.

CSS compilado

```
.simple {
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;
  border-radius: 10px;
}
```

Enlaces de ampliación:

- Working with Configurable Variables (Documentación Compass)
<http://compass-style.org/help/tutorials/configurable-variables/>
- Compass Variables (Documentación Compass)
<http://compass-style.org/index/variables/>

Mixins

Compass incluye una amplia librería de Mixins listos para ser utilizados y ahorrarnos mucho código CSS. Además, como en el ejemplo anterior, destacan por añadir el código CSS necesario para asegurar la compatibilidad con la mayoría de navegadores.

Ejemplo 1:

Como ejemplo vamos a aplicar estilos a una lista de enlaces utilizando los siguientes mixins de Compass:

- **pretty-bullets**(\$bullet-icon, \$width, \$height, \$line-height, \$padding). Añade un icono a los elementos de la lista.
<http://compass-style.org/reference/compass/typography/lists/bullets/>
- **link-colors**(\$normal, \$hover, \$active, \$visited, \$focus). Asigna todos los colores de un enlace en una única llamada.
http://compass-style.org/reference/compass/typography/links/link_colors/

No olvides importar las librerías correspondientes antes de utilizar los mixins.

SCSS (+ Compass)

```
@import "compass/typography/lists/bullets";
@import "compass/typography/links/link-colors";

$normal: #0c1b33;
$hover: #fff;
$active: #03b5aa;
$visited: #03b5aa;
$focus: #dbfe87;

nav {
  ul {
    @include pretty-bullets('icon.gif', 12px, 12px);
  }

  a {
    @include link-colors($normal, $hover, $active, $visited, $focus);
    font-family: Arial, Verdana;
    background-color: #ddd;
    &:hover {
      background-color: #000;
    }
  }
}
```

CSS compilado

```

nav ul {
  margin-left: 0;
}
nav ul li {
  padding-left: 14px;
  background: url(icon.gif) no-repeat 1px 3px;
  list-style-type: none;
}
nav a {
  color: #0c1b33;
  font-family: Arial, Verdana;
  background-color: #ddd;
}
nav a:visited {
  color: #03b5aa;
}
nav a:focus {
  color: #dbfe87;
}
nav a:hover {
  color: #fff;
}
nav a:active {
  color: #03b5aa;
}
nav a:hover {
  background-color: #000;
}

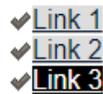
```

El resultado se muestra en la Figura [F9.5b]:

F9.5b

Mixins de Compass

Ejemplo de lista de enlaces con estilos aplicados a partir de mixins de Compass.



Ejemplo 2:

En este ejemplo utilizaremos los siguientes mixins para maquetar una caja leading and trailing contenedora con bordes redondeados:

- **border-radius**(\$radius, \$vertical-radius). Redondea las esquinas según el valor de radio especificado.
http://compass-style.org/reference/compass/css3/border_radius/
- **horizontal-borders**(\$width, \$lines, \$font-size, \$border-style). Aplica estilo de borde a ambos lados de la caja.
http://compass-style.org/reference/compass/typography/vertical_rhythm/#mixin-horizontal-borders
- **rhythm-borders**(\$width, \$lines, \$font-size, \$border-style). Aplica bordes y espacios (padding) iguales en todos los lados del contenedor.
http://compass-style.org/reference/compass/typography/vertical_rhythm/#mixin-rhythm-borders

SCSS (+ Compass)

```
@import "compass/css3/border-radius";
@import "compass/typography/vertical_rhythm";

.box {
  @include border-radius(10px);
  @include horizontal-borders(2px, .5);
  @include rhythm-borders(2px, .5);
  background-color: #ddd;
}
```

CSS compilado

```
.box {
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;
  border-radius: 10px;
  border-top-width: 0.125em;
  border-top-style: solid;
  padding-top: 0.625em;
  border-bottom-width: 0.125em;
  border-bottom-style: solid;
  padding-bottom: 0.625em;
  border-width: 0.125em;
  border-style: solid;
  padding: 0.625em;
  background-color: #ddd;
}
```

El resultado se muestra en la Figura [F9.5c]:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin mauris leo, dapibus non orci a, porta porttitor justo. Sed at vestibulum tortor, eu porta mauris. Nam iaculis tellus at turpis faucibus, ac tincidunt odio posuere. Vivamus tincidunt lorem id consectetur pharetra. Cras nulla eros, gravida vitae placerat et, aliquet sed dui. Vestibulum cursus condimentum augue, vitae auctor felis porta in. Etiam malesuada ornare arcu id sagittis. Curabitur ac odio sem. Nam eu tempus augue. Nam elementum dolor sed justo efficitur consequat. Suspendisse non turpis convallis, porta neque a, feugiat lectus.

F9.5c

Mixins de Compass

Ejemplo de estilos de borde usando mixins de Compass.

Enlaces de ampliación:

- Compass Mixins (Documentación Compass)
<http://compass-style.org/index/mixins/>

Funciones

Como ejemplo de función de Compass aplicaremos:

- **contrast-color**(\$color, \$dark, \$light, \$threshold). Devuelve el color que más contrasta con el color facilitado.

<http://compass-style.org/reference/compass/utilities/color/contrast/#function-contrast-color>

SCSS (+ Compass)

```
@import "compass/utilities/color/contrast";

$color: #eef99a;

.box-light {
  background-color: $color;
  color: contrast-color($color);
}

.box-dark {
  background-color: contrast-color($color);
  color: $color;
}
```

El resultado se muestra en la Figura **[F9.5d]**:

F9.5d

Funciones de Compass

Ejemplo de colores contrastados usando funciones de Compass.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla condimentum justo leo, quis dignissim massa tristique id. Sed blandit nulla a egestas consectetur. Cras erat dolor, fermentum consectetur venenatis ut, egestas nec sem. Morbi molestie at nunc in rutrum. Aliquam facilisis nunc quis dapibus tincidunt. Nulla lacinia interdum ligula. Nulla mattis tortor massa, pharetra porta quam ullamcorper a. Aliquam malesuada libero a elementum mattis. Suspendisse finibus eget sem nec finibus.

Quisque volutpat, quam ac posuere mattis, lorem felis accumsan elit, sit amet rutrum arcu lectus et neque. Praesent volutpat, leo ac imperdiet imperdiet, dolor ipsum dapibus arcu, volutpat rhoncus enim ligula vel velit. In ultricies malesuada sapien, ac luctus tortor finibus nec. Etiam sit amet mauris at mi scelerisque vestibulum quis sed tortor. Sed venenatis nunc eu tellus malesuada viverra. Cras lobortis felis lectus, at eleifend diam posuere gravida. Suspendisse potenti.

Enlaces de ampliación:

- Sass Based Functions (Documentación Compass)
<http://compass-style.org/index/functions/>

Otras librerías: Bourbon

Existen otros frameworks o librerías similares a Compass, que extienden Sass aportando mixins, funciones y variables adicionales. Un ejemplo es **Bourbon** (<https://www.bourbon.io/>), una librería muy ligera, cuyos componentes puedes consultar en: <https://www.bourbon.io/docs/latest/>

Para instalar Bourbon, puedes seguir estos pasos (recuerda que en Forcontu ya lo encontrarás instalado): <https://github.com/thoughtbot/bourbon#installation>

En el siguiente ejemplo aplicamos diferentes estilos a un contenedor, utilizando algunos mixins de Bourbon:

SCSS (+ Bourbon)

```
@import "bourbon";

.box {
  @include border-style(solid);
  @include border-color(#a60b55 #76cd9c null #e8aela);
  @include padding(1rem 2rem);
  @include border-top-radius(10px);
  @include border-bottom-radius(10px);
  @include clearfix;
}
```

CSS compilado

```
.box {
  border-style: solid;
  border-top-color: #a60b55;
  border-right-color: #76cd9c;
  border-left-color: #e8aela;
  padding: 1rem 2rem;
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
  border-bottom-left-radius: 10px;
  border-bottom-right-radius: 10px;
}
.box::after {
  clear: both;
  content: "";
  display: table;
}
```

El resultado se muestra en la Figura [F9.5e]:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec dictum auctor cursus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed sem et metus fringilla ultricies. Proin augue metus, viverra id tempus vitae, fringilla eu lacus. Donec porta urna id metus aliquet, sed sodales nunc rhoncus. Aenean et augue sed nulla gravida lacinia. Mauris et purus rhoncus, feugiat mauris eget, vestibulum mi. Donec bibendum lectus id libero rhoncus dapibus. Quisque faucibus porttitor eros, at ornare nisi maximus aliquam.

F9.5e

Bourbon

Estilos aplicados con Bourbon.

Utilización de Bourbon en Drupal

Sigue los siguientes pasos para utilizar Bourbon en tu tema de Drupal:

1. Instala Bourbon dentro de la carpeta que contiene los archivos Sass (scss).

```
cd themes/b_forcontu/scss
```

```
bourbon install
```

2. Importa la librería en los archivos SCSS con:

```
@import "bourbon/bourbon";
```

3. Compila normalmente, con Sass o Compass.

Herramientas de automatización: Grunt

9.6

Los **task runners** son aplicaciones encargadas de automatizar tareas repetitivas relacionadas con el flujo de trabajo. Algunas de las tareas que podemos realizar con un task runner son:

- **Compilación de Sass** para obtener el CSS.
- **Minificación** de archivos JavaScript y CSS. Compresión del código para ahorrar ancho de banda.
- **Unit testing**. Pruebas unitarias de código JavaScript.
- **Linting**. Verificar la sintaxis del código (Sass, CSS, JavaScript).

En esta unidad nos centraremos en la **tarea de compilación de Sass** para obtener los archivos CSS.

Grunt (<https://gruntjs.com>) es un **task runner** de JavaScript que cuenta con un gran número de plugins para automatizar todo tipo de tareas.

Instalación y configuración

La línea de comandos de Grunt (grunt-cli) ya está instalada a nivel global en el servidor de Forcontu, así que este paso no será necesario. Si necesitas instalarlo a nivel local o en tu propio servidor, consulta las instrucciones en:

<https://gruntjs.com/getting-started> (apartado **Installing the CLI**).

1. Inicializar el proyecto

Crearemos el proyecto en la carpeta del tema donde vayamos a utilizarlo (Por ejemplo, themes/b_forcontu).

```
cd themes/b_forcontu
npm init
```

El sistema nos guiará para crear el archivo package.json, donde se registrarán las dependencias o paquetes adicionales que requieran nuestras tareas.

```
package name: (b_forcontu)
version: (1.0.0)
description: b forcontu grunt
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
```

El archivo `package.json` tendrá el siguiente contenido:

```
{
  "name": "b_forcontu",
  "version": "1.0.0",
  "description": "b_forcontu grunt",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

2. Instalación de Grunt y plugins adicionales

Seguimos en el directorio raíz del tema e instalamos Grunt con el siguiente comando: `npm install <package> --save-dev`

```
npm install grunt --save-dev
```

A continuación, instalaremos los paquetes adicionales que necesitamos para trabajar con Sass:

- **grunt-contrib-sass.** Realiza la compilación de archivos Sass (SCSS) para obtener el CSS.
<https://github.com/gruntjs/grunt-contrib-sass>
- **grunt-contrib-watch.** Vigila los cambios en archivos para proceder a la compilación, que se realizará con la tarea anterior.
<https://github.com/gruntjs/grunt-contrib-watch>

```
npm install grunt-contrib-sass --save-dev
npm install grunt-contrib-watch --save-dev
```

El archivo `package.json` recogerá ahora la información de los paquetes instalados dentro de la sección **"devDependencies"**. La versión de cada paquete se corresponde con la última versión estable disponible:

`package.json` (fragmento)

```
"devDependencies": {
  "grunt": "^1.0.3",
  "grunt-contrib-sass": "^1.0.0",
  "grunt-contrib-watch": "^1.1.0"
}
```

3. Archivo Gruntfile.js

En el archivo **Gruntfile.js** se definen las tareas que se realizarán al ejecutar Grunt. Aunque es un archivo JavaScript (lenguaje que estudiaremos más adelante), es suficiente con respetar su estructura:

Gruntfile.js

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    sass: {
      dist: {
        files: {
          'css/style.css' : 'scss/style.scss'
        }
      }
    },
    watch: {
      css: {
        files: '**/*.scss',
        tasks: ['sass']
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-sass');
  grunt.loadNpmTasks('grunt-contrib-watch');
  grunt.registerTask('default', ['watch']);
}
```

Dentro del método **initConfig()** de Grunt se definen las tareas y las opciones de configuración de cada tarea, que encontraremos en la página de ayuda y descarga de cada plugin.

Por ejemplo, dentro de la tarea **sass** (<https://github.com/gruntjs/grunt-contrib-sass>), podemos añadir otras opciones de configuración, como:

- **update: true.** Sólo compila los archivos que hayan cambiado.
- **sourcemap: 'auto'.** Genera los archivos de mapeo (es el valor por defecto). Para desactivar los archivos de mapeo podemos utilizar la opción 'none'.

Gruntfile.js (fragmento)

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    sass: {
      dist: {
        options: {
          update: true,
          sourcemap: 'auto'
        },
        files: {
          'css/style.css' : 'scss/style.scss'
        }
      }
    },
    ...
  });
}
```

En **files** podemos registrar todos los archivos que van a ser compilados, siguiendo el formato:

```
'destination': 'source'
```

Gruntfile.js (fragmento)

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    sass: {
      dist: {
        files: {
          'css/style.css' : 'scss/style.scss',
          'css/main.css' : 'scss/main.scss',
        }
      }
    },
  },
  ...
}
```

La tarea se carga llamando al método `loadNpmTasks()`. Por ejemplo:

```
grunt.loadNpmTasks('grunt-contrib-sass');
```

Y el método `registerTask()` nos permite definir tareas adicionales agrupando las tareas definidas.

Por ejemplo, podemos crear la tarea 'dev' que llamará a las tareas sass y watch:

```
grunt.registerTask('dev', ['sass', 'watch']);
```

En nuestro archivo se cargan las tareas y se define la tarea 'default', que se usará cuando se llame al comando `grunt` sin parámetros:

Gruntfile.js (fragmento)

```
...
grunt.loadNpmTasks('grunt-contrib-sass');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.registerTask('default', ['watch']);
}
```

Ejecución de tareas

Las tareas se ejecutan desde la carpeta donde está instalado Grunt, que en nuestro caso se corresponde con la carpeta raíz del tema.

Podemos ejecutar las tareas de forma individual, indicando el nombre de la tarea:

```
grunt sass
grunt watch
```

También podemos ejecutar un conjunto de tarea definido con `registerTask()`:

```
grunt dev
```

Integración con Compass

El plugin **grump-contrib-compass** permite compilar archivos Sass con Compass.

<https://github.com/gruntjs/grunt-contrib-compass>

Instalaremos primero el plugin y comprobaremos que se ha añadido al archivo `package.json`:

```
npm install grunt-contrib-compass --save-dev
```

`package.json` (fragmento)

```
"devDependencies": {
  "grunt": "^1.0.3",
  "grunt-contrib-compass": "^1.1.1",
  "grunt-contrib-sass": "^1.0.0",
  "grunt-contrib-watch": "^1.1.0"
}
```

Ahora ya podemos crear la nueva tarea dentro del archivo `Grumpfile.js`. En nuestro caso eliminaremos la tarea **sass** para dejar que **Compass** controle la compilación de Sass.

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    compass: {
      dist: {
        options: {
          sassDir: 'scss',
          cssDir: 'css',
          sourcemap: true
        }
      }
    },
    watch: {
      css: {
        files: '**/*.scss',
        tasks: ['compass']
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-compass');
  grunt.loadNpmTasks('grunt-contrib-watch');
  grunt.registerTask('default', ['watch']);
}
```

Las opciones **sassDir** y **cssDir** permiten indicar las carpetas con los archivos Sass (SCSS) y CSS, respectivamente. Tenemos que incluir la opción **sourcemap** a true para que se generen los archivos de mapeo, ya que el valor por defecto es false.

Si el archivo de configuración de Compass (`config.rb`) no se encuentra en el mismo directorio que el archivo `Grumpfile.js`, podemos especificar la ruta añadiendo la opción **config**.

Probamos nuevamente el comando grunt y realizamos alguna modificación en los archivos SCSS, esta vez incluyendo algún componente de Compass:

```
# grunt
Running "watch" task
Waiting...
>> File "scss/style.scss" changed.
Running "compass:dist" (compass) task
   write css/style.css (0.973s)
   write css/style.css.map

Done.
Completed in 4.952s at Mon Oct 22 2018 19:23:23 GMT+0200
(GMT+02:00) - Waiting...
```

Desinstalar un plugin

Al haber cambiado a Compass, ya no necesitamos el plugin grunt-contrib-sass. Lo podemos desinstalar con el siguiente comando, que además elimina la dependencia con el paquete del archivo package.json:

```
npm uninstall grunt-contrib-sass --save-dev
```

Enlaces de ampliación (Grunt)

- Getting started with Grunt and Sass
<http://ryanchristiani.com/getting-started-with-grunt-and-sass/>
- Add Compass to your Grunt task
<http://ryanchristiani.com/add-compass-to-your-grunt-task/>

Herramientas de automatización: Gulp

9.7

Continuamos el estudio de herramientas de automatización con Gulp. Al igual que Grunt, **Gulp** (<https://gulpjs.com/>) es un **task runner** de JavaScript que cuenta con un gran número de plugins para automatizar todo tipo de tareas. Las posibilidades de ambos son prácticamente las mismas, variando la sintaxis en la creación de tareas y la forma en que estas se ejecutan. Veremos más sobre las diferencias entre Grunt y Gulp al final de este apartado.

Instalación y configuración

La línea de comandos de Gulp (gulp-cli) ya está instalada a nivel global en el servidor de Forcontu, así que este paso no será necesario. Si necesitas instalarlo a nivel local o en tu propio servidor, consulta las instrucciones en:

- Install gulp globally:
<https://github.com/gulpjs/gulp/blob/v3.9.1/docs/getting-started.md>

1. Inicializar el proyecto

Crearemos el proyecto en la carpeta del tema donde vayamos a utilizarlo. Para este ejemplo hemos replicado el tema `b_forcontu` renombrándolo a **g_forcontu**, así que usaremos Gulp dentro de **themes/g_forcontu**:

```
cd themes/g_forcontu
npm init
```

El sistema nos guiará para crear el archivo `package.json`, donde se registrarán las dependencias o paquetes adicionales que requieran nuestras tareas.

```
package name: (g_forcontu)
version: (1.0.0)
description: g_forcontu gulp
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
```

El archivo `package.json` tendrá el siguiente contenido:

```
{
  "name": "g_forcontu",
  "version": "1.0.0",
  "description": "g_forcontu gulp",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

2. Instalación de Gulp y plugins adicionales

Seguimos en el directorio raíz del tema e instalamos Gulp con el siguiente comando:
npm install <package> --save-dev

```
npm install gulp --save-dev
```

A continuación, instalaremos los paquetes adicionales que necesitamos para trabajar con Sass:

- **gulp-sas**. Realiza la compilación de archivos Sass (SCSS) para obtener el CSS.

<https://www.npmjs.com/package/gulp-sass>

```
npm install node-sass gulp-sass --save-dev
```

Nota: también hemos instalado el paquete node-sass, que es el vínculo entre Node.js y la librería Sass.

El archivo package.json recogerá ahora la información de los paquetes instalados dentro de la sección "**devDependencies**". La versión de cada paquete se corresponde con la última versión estable disponible:

package.json (fragmento)

```
"devDependencies": {  
  "gulp": "^3.9.1",  
  "gulp-sass": "^4.0.2",  
  "node-sass": "^4.9.4"  
}
```

3. Archivo gulpfile.js

En el archivo **gulpfile.js** se definen las tareas que se realizarán al ejecutar Gulp. Mostramos a continuación la estructura general del archivo:

gulpfile.js

```
var gulp = require('gulp');  
  
gulp.task('default', function() {  
  // código a ejecutar dentro de la tarea  
});  
  
gulp.task('task1', function() {  
  // código a ejecutar dentro de la tarea  
});
```

Vamos a añadir al archivo `gulpfile.js` las tareas relacionadas con el plugin `gulp-sass` (<https://www.npmjs.com/package/gulp-sass>).

gulpfile.js

```
'use strict';

var gulp = require('gulp');
var sass = require('gulp-sass');

sass.compiler = require('node-sass');

gulp.task('sass', function () {
  return gulp.src('./scss/*.scss')
    .pipe(sass().on('error', sass.logError))
    .pipe(gulp.dest('./css'));
});

gulp.task('sass:watch', function () {
  gulp.watch('./scss/*.scss', ['sass']);
});
```

Analizando el código de este archivo, destacamos:

- Cada tarea se encapsula con el método `.task()`, definiendo el nombre de la tarea y dentro de `function()` las acciones que realice.

gulpfile.js (fragmento)

```
gulp.task('sass', function () {
  //...
});
```

- NodeJs trabaja con streams y pipes. En este punto nos quedamos únicamente con la idea de que las llamadas a `pipe()` se pueden encadenar, de la forma: `a.pipe(b).pipe(c)`, donde `a`, `b` y `c` son streams o flujos de información.
- Dentro de la tarea 'sass', se asigna la carpeta de origen (`gulp.src()`), se hace una llamada a `sass()` para compilar los archivos, y se almacena el resultado en el destino (`gulp.dest()`). Son tres streams encadenados que trabajarían en ese orden, aunque no necesariamente de forma atómica.

Consulta los enlaces recomendados al final de este apartado para leer más sobre este tema.

Ejecución de tareas

Las tareas se ejecutan desde la carpeta donde está instalado Gulp, que en nuestro caso se corresponde con la carpeta raíz del tema.

Podemos ejecutar las tareas de forma individual, indicando el nombre de la tarea:

```
grunt sass
grunt watch
```

En Gulp también podemos crear tareas para agrupar otras tareas:

gulpfile.js (fragmento)

```
gulp.task('dev', ['sass', 'watch']);
```

En este ejemplo, al ejecutar la tarea dev, se ejecutarán las tareas sass y watch, en ese orden:

```
gulp dev
```

También podemos añadir código adicional a la tarea "agrupadora", añadiendo su correspondiente function:

gulpfile.js (fragmento)

```
gulp.task('dev', ['sass', 'watch'], function () {
  // gulp.src( ...
});
```

Integración con Compass

Para crear tareas Compass en Gulp, tenemos que instalar el plugin gulp-compass, disponible en: <https://www.npmjs.com/package/gulp-compass>

Instalaremos primero el plugin y comprobaremos que se ha añadido al archivo package.json:

```
npm install gulp-compass --save-dev
```

package.json (fragmento)

```
"devDependencies": {
  "gulp": "^3.9.1",
  "gulp-compass": "^2.1.0",
  "gulp-sass": "^4.0.2",
  "node-sass": "^4.9.4"
}
```

Ahora ya podemos crear la nueva tarea dentro del archivo gumpfile.js. En nuestro caso eliminaremos la tarea **sass** para dejar que **Compass** controle la compilación de Sass:

```
'use strict';

var gulp = require('gulp');
var compass = require('gulp-compass');

gulp.task('compass', function() {
  gulp.src('./scss/*.scss')
    .pipe(compass({
      config_file: './config.rb',
      css: 'css',
      sass: 'scss'
    }))
    .pipe(gulp.dest('app/assets/temp'));
});
```

En este ejemplo estamos utilizando el archivo de configuración de Compass (config.rb), colocado a la misma altura que gulpfile.js. Las propiedades css y sass, que indican las carpetas donde se ubican los archivos, deben coincidir con las del archivo de configuración config.rb.

Al ejecutar la tarea (gulp compass), se compilarán los archivos y se generarán los archivos de mapeo, si así lo hemos definido en el archivo de configuración de Compass (sourcemap = true):

```
$ gulp compass
[06:21:47] Using gulpfile ~/themes/g_forcontu/gulpfile.js
[06:21:47] Starting 'compass'...
[06:21:47] Finished 'compass' after 5.19 ms
   write css/style.css
   write css/style.css.map
```

watch

El plugin gulp-compass permite ejecutar la tarea en modo compilación o watch añadiendo el parámetro de configuración task:

- **task: 'compile'**, compila directamente.
- **task: 'watch'**, queda a la espera de cambios en los archivos Sass.

```
'use strict';

var gulp = require('gulp');
var compass = require('gulp-compass');

gulp.task('compass', function() {
  gulp.src('./scss/*.scss')
    .pipe(compass({
      config_file: './config.rb',
      css: 'css',
      sass: 'scss',
      task: 'watch'
    }))
    .pipe(gulp.dest('app/assets/temp'));
});
```

Ahora, al ejecutar la tarea (gulp compass), el sistema se quedará esperando hasta que detecte algún cambio en los archivos SCSS:

```
$ gulp compass
[06:43:32] Using gulpfile
~/public_html/fe/fel/themes/g_forcontu/gulpfile.js
[06:43:32] Starting 'compass'...
[06:43:32] Finished 'compass' after 5.59 ms
>>> Compass is watching for changes. Press Ctrl-C to Stop.

created scss/style.scss
   write css/style.css
   write css/style.css.map
```

Para dejar de vigilar los cambios, cortaremos la tarea con **Control-C**.

Desinstalar un plugin

Al haber cambiado a Compass, ya no necesitamos el plugin gulp-sass. Lo podemos desinstalar con el siguiente comando, que además elimina la dependencia con el paquete del archivo package.json:

```
npm uninstall gulp-sass --save-dev
```

Enlaces de ampliación (Gulp)

- Getting started with Gulp and Sass (Ryan Christiani)
<http://ryanchristiani.com/getting-started-with-gulp-and-sass/>
- Creating automation tools for custom themes (Gulpjs)
<https://www.drupal.org/docs/8/theming/creating-automation-tools-for-custom-themes-gulpjs>
- Getting started with Gulp.js (Aleksandar Olic)
<https://semaphoreci.com/community/tutorials/getting-started-with-gulp-js>
- El manejo de streams en NodeJS
<https://elabismodenuell.wordpress.com/2017/03/28/el-manejo-de-streams-en-nodejs/>

Diferencias entre Gulp y Grump

Gulp y Grump tienen funcionalidades muy parecidas y ambos cubren los tipos de tareas más usados en un proyecto web.

Destacamos algunas diferencias importantes que te ayudarán a la hora de decidir cuál de ellos usar:

1. **Configuración y programación.**
Grunt es más fácil de configurar. Al tener un enfoque a variables de configuración, prácticamente no hay que programar nada (aunque también es posible añadir código personalizado). Simplemente buscamos las opciones de configuración en la página del plugin, y lo adaptamos a nuestras necesidades.

Gulp está orientado a tareas que necesitan ser programadas, y por eso es **más complicado**. Suele ser la opción ideal para desarrolladores con amplia experiencia en JavaScript, ya que les otorga **más flexibilidad** a la hora de añadir código personalizado.
2. **Plugins disponibles.**
Grunt tiene más plugins que Gulp, pero esta cantidad no debería ser relevante para decantarte por uno de ellos. Busca los plugins que

necesites y comprueba que estén disponibles en tu task runner preferido. Si no es así, quizás si que necesites cambiar de aplicación. Como ya comentamos, los plugins más usados están disponibles para ambos.

3. Rendimiento.

Grunt utiliza archivos temporales para realizar sus operaciones.

Gulp permite la ejecución de tareas en modo stream de Node.js, haciendo un mayor uso de memoria. El resultado de las operaciones se vuelca en los archivos finales, pero en el procesamiento de las tareas se aprovecha más la memoria principal, consiguiendo con ello más velocidad y mejor rendimiento.

Generalmente Grump suele ser una buena opción para proyectos pequeños y desarrolladores con poca experiencia en JavaScript. Gulp lo podemos dejar para proyectos más complejos donde el rendimiento es vital.

Enlaces de ampliación

- Gulp vs. Grunt: dos task runners y sus características
<https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/gulp-vs-grunt-que-diferencia-a-estos-task-runners/>
- Grunt o Gulp: ¿qué uso?
<http://blog.koalite.com/2015/06/grunt-o-gulp-que-uso/>

9.8 Configuración de Sass en AdaptiveTheme

Aunque AdaptiveTheme incluye muchas funcionalidades que se pueden configurar directamente desde la interfaz de configuración del tema, también incluye algunas herramientas útiles para desarrolladores Front-End.

Grunt

El tema **AT Forcontu** creado a partir de AdaptiveTheme incluye Grunt como task runner, con tareas ya preparadas para compilar Sass.

El tema incluye un archivo **Gruntfile.js** preconfigurado, pero debemos instalar Grunt para poder utilizarlo. También encontraremos el archivo **package.json** con las versiones de los plugins que se deben instalar.

Desde el directorio raíz del tema, `themes/at_forcontu`, ejecuta el comando:

```
npm install grunt --save-dev
```

También instalaremos los plugins requeridos. Puedes consultar el listado completo en el archivo **package.json** (ten en cuenta que los plugins requeridos pueden cambiar en las siguientes versiones de AdaptiveTheme):

```
"devDependencies": {
  "grunt": "1.0.1",
  "grunt-sass": "2.0.0",
  "grunt-sass-globber": "1.5.1",
  "grunt-contrib-watch": "1.0.0",
  "autoprefixer": "7.1.6",
  "grunt-postcss": "0.9.0",
  "grunt-browser-sync": "2.2.0",
  "grunt-contrib-csslint": "2.0.0",
  "susy": "2.2.12",
  "modularscale-sass": "3.0.3"
},
```

En este caso, los comandos a ejecutar serán:

```
npm install grunt-sass --save-dev
npm install grunt-sass-globber --save-dev
npm install grunt-contrib-watch --save-dev
npm install autoprefixer --save-dev
npm install grunt-postcss --save-dev
npm install grunt-browser-sync --save-dev
npm install grunt-contrib-csslint --save-dev
npm install susy --save-dev
npm install modularscale-sass --save-dev
```

Una vez instalados todos los plugins, ya podemos lanzar el comando `grunt watch`, que quedará a la espera de cambios en los archivos SCSS:

```
$ grunt watch
Running "watch:uikit" (watch) task
Waiting...
```

Sass

AdaptiveTheme utilizar la sintaxis SCSS, y organiza los archivos de la siguiente forma:

- `/themes/at_forcontu/layout`. Estilos relacionados con los layouts que seleccionamos desde la interfaz. Estos estilos utilizan el framework **Susy**.
- `/themes/at_forcontu/styles/uikit`. Estilos relacionados con los componentes (botones, imágenes, fuentes, menús, etc.). Estos estilos utilizan el framework **UIkit**.

Podemos tocar cualquiera de estos estilos, pero recuerda que **sólo debemos modificar los archivos .scss** y nunca los archivos .css, ya que estos últimos se generarán automáticamente al compilar Sass (y cualquier cambio directo será eliminado).

Nota: También es importante **desactivar la agregación de archivos CSS** de Drupal, desde Rendimiento, o de lo contrario no veremos los cambios en el CSS por estar cacheados.

Los archivos CSS compilados se encuentran en:

- `/themes/at_forcontu/styles/css/components`. Estilos generados desde la carpeta uikit, separados por componentes.
- `/themes/at_forcontu/styles/css/generated`. Estilos generados desde la carpeta layout. Se genera un único archivo con los estilos relacionados con el layout.

Susy

Susy es un framework de Sass orientado a la maquetación de layouts (<http://oddbird.net/susy/>).

No hemos entrado en detalle en este curso porque desde la aparición de soluciones como Flexbox o CSS Grid, que sí hemos estudiado, ya no tiene tanto sentido en uso de Susy. Puedes leer sobre este tema en la propia página de Susy.

Hay dos versiones de Susy, 2.x y 3.x. La versión concreta que se está utilizando en AdaptiveTheme se puede consultar en el archivo package.son. Como es una versión 2.x (2.2.12), debes consultar la documentación correspondiente a esta versión:

<https://susy.readthedocs.io/>

Dentro de Toolkit encontrarás los componentes disponibles (mixins y funciones).

Para usar los componentes de Susy, necesitamos incluir la librería en los archivos Sass. Susy está disponible en `at_forcontu/node_modules/susy/sass/susy`, pero debemos indicar la ruta relativa, que depende de dónde esté situado el archivo a modificar. Por ejemplo:

Archivo: `at_forcontu/layout/page-layout/_page-layout-base.scss`

```
@import "../../node_modules/susy/sass/susy";
```

UIkit

UIkit (<https://getuikit.com/>) es un paquete que contiene un amplio catálogo de estilos preconfigurados. Generalmente se instala el paquete base y a partir de ahí se van añadiendo nuevos estilos.

Recuerda que no necesitas conocer toda la estructura archivos. Tal y como hicimos en el apartado 9.4, desde el navegador puedes inspeccionar los estilos que se aplican para cada elemento HTML, y localizar el archivo SCSS a modificar en cada caso.

PostCSS

grunt-postcss es un plugin de Grunt para incorporar la herramienta PostCSS (<https://github.com/postcss/postcss>). Lo que permite es aplicar reglas y cambios sobre los CSS ya generados, según una serie de post-procesadores.

En AdaptiveTheme se utiliza el post-procesador Autoprefixer, que adapta el código CSS en función de los navegadores (y versiones) que se deseen soportar. Lo que hace es añadir los prefijos o propiedades particulares de los navegadores más populares (`-webkit-`, `-o-`, `-moz-`, `-ms-`, etc.). Por ejemplo:

CSS

```
.box {
  background: linear-gradient(to bottom, white, black);
}
```

CSS procesado con Autoprefixer

```
.box {
  background: -webkit-gradient(linear, left top, left bottom,
  from(white), to(black));
  background: -webkit-linear-gradient(top, white, black);
  background: -o-linear-gradient(top, white, black);
  background: linear-gradient(to bottom, white, black);
}
```

Puedes leer más sobre Autoprefixer en: <https://github.com/postcss/autoprefixer>

O probarlo online en: <http://autoprefixer.github.io/>

En el archivo Gruntfile.js vemos que la condición o filtro que se añade es que se soporten las 5 últimas versiones (de los navegadores más populares).

Gruntfile.js (fragmento)

```
postcss: {
  uikit: {
    src: 'styles/css/components/**/*.css',
    options: {
      map: {
        inline: false,
        annotation: 'styles/css/components'
      },
      processors: [
        require('autoprefixer')({browsers: 'last 5 versions'})
      ]
    }
  },
  ...
},...
```

La tarea postcss está incluida dentro de las tareas de 'watch', después de la compilación de los archivos Sass. Por tanto, no es necesario ejecutar la tarea de forma independiente (seguiremos trabajando con **grunt watch**).

csslint

grunt-contrib-csslint es un plugin de Grunt que permite verificar la sintaxis del código CSS generado.

Según el archivo Gruntfile.js, sólo se verifica la sintaxis de los archivos CSS generados en la carpeta 'styles/css/components':

```
csslint: {
  options: {
    csslintrc: '.csslintrc'
  },
  strict: {
    options: {
      import: 2
    },
    src: ['styles/css/components/**/*.css']
  }
},
```

Para ejecutar esta tarea, utiliza el comando: **grunt csslint**

```
$ grunt csslint
Running "csslint:strict" (csslint) task
Linting styles/css/components/aggregator.css...ERROR
[L8:C1]
WARNING: Rule doesn't have all its properties in alphabetical
order. Assure properties are in alphabetical order (order-
alphabetical) Browsers: All
Linting styles/css/components/base.css...ERROR
[L6:C1]
WARNING: Use of !important Be careful when using !important
declaration (important) Browsers: All
[L361:C3]
```


X Diario de cambios (Change log)

Ayúdanos a mejorar los libros reportando cualquier error que encuentres. Puedes hacerlo directamente completando este formulario (sólo para usuarios registrados en forcontu.com):

<https://www.forcontu.com/libros/reportar-error>

Versión 1.0 (25/10/2018). Lanzamiento.