

Construir un sitio web reactivo con plantillas prediseñadas

Jorge Franganillo

Facultad de Información y Medios Audiovisuales

Universidad de Barcelona

<https://franganillo.es>

octubre de 2024

1. Plataformas y metodologías de publicación

Al afrontar un proyecto de web, una de las primeras decisiones a tomar consiste en determinar con qué tecnología se abordará el desarrollo. Es una decisión necesaria porque no hay un único paradigma. Los **gestores de contenido** (CMS, por las siglas de *content management systems*) no son la única herramienta ni metodología de desarrollo. Son una opción más, con sus ventajas e inconvenientes.

En general, se está haciendo un uso intensivo de los CMS, que en muchos casos aportan beneficios. No obstante, usarlos de forma indiscriminada para cualquier proyecto, sin ninguna reflexión, puede hacer desaprovechar las ventajas de utilizar un **framework de diseño web**, una alternativa que implementa buenas prácticas en el desarrollo de sitios web de las que los CMS no siempre permiten beneficiarse.

1.1. Ventajas e inconvenientes de los CMS

En el desarrollo web es frecuente emplear algún CMS por la facilidad con que, una vez instalado y configurado, se pueden organizar y actualizar los contenidos. De ahí que muchos de los portales web más importantes se basen en algún CMS. Pero también hay sitios muy solventes que no se basan en ese paradigma porque en ciertos contextos los CMS no son la herramienta más adecuada.

Los CMS presentan ciertas **ventajas**. Al estar orientados al desarrollo web, ya llevan integradas muchas funcionalidades comunes de un web (p. ej., *sitemap*, aspectos de optimización, etc.) y ello conduce, en un principio, a un coste de desarrollo menor porque la herramienta de gestión ya está elaborada. Entre otras ventajas, los CMS también permiten incorporar extensiones con relativa facilidad, favorecen la autonomía del usuario final, facilitan el desarrollo de contenidos dinámicos y garantizan cierto nivel de seguridad.

Sin embargo, este último aspecto es un arma de doble filo. Precisamente porque los CMS son herramientas de uso muy extendido, la seguridad es su «talón de Aquiles».

Según un informe reciente de la empresa de seguridad Sucuri (2022), *WordPress*, el sistema más usado del mundo, con una cuota de mercado del 43% (W3Techs, 2022), ha sido también el más pirateado; ha recibido el 95,62% de los ataques.

Ello ilustra que el hecho de que los CMS sean herramientas facilitadoras no significa que sean fáciles de manejar. Estas plataformas requieren igualmente cierto dominio técnico. El despliegue, la configuración y el mantenimiento son tareas que a menudo, por mero desconocimiento, se gestionan de manera pobre o con errores.

Este es, pues, uno de los **inconvenientes** de los CMS, a los que se suman otros bien conocidos, relacionados con el modelo de datos, las vulnerabilidades y la sobredimensión de la herramienta para necesidades más bien modestas. Es razonable pensar, pues, que si la potencia, la complejidad y las prestaciones de un CMS sobrepasan la naturaleza y las necesidades de un proyecto, puede ser que realmente, en definitiva, no se necesite un CMS.

1.2. La alternativa: los *frameworks* de diseño web

Las desventajas descritas aconsejan concebir los CMS como una opción. En efecto, los CMS son una posibilidad más a considerar en un ecosistema que contempla otro paradigma que en ciertos contextos resulta muy provechoso: el desarrollo a medida por medio de un ***framework de diseño web***, que ofrece varias **ventajas**:

- Al ser desarrollos a medida, permiten elegir con qué tecnología se van a desarrollar y se puede emplear el *framework* que mejor se adapte al proyecto.
- Son más ligeros que un CMS y suelen generar menos dependencias.
- Se manejan con un código fuente más limpio y optimizado, que incorpora buenas prácticas de desarrollo. Los sitios desarrollados con esta tecnología destacan por su mejor rendimiento: tienden a obtener puntuaciones más altas en las pruebas de *PageSpeed Insights*, lo que revierte en un mejor posicionamiento.

Prueba de las bondades de este modelo es que grandes marcas e instituciones como la Obama Foundation, Microsoft, la NASA, ITV News, Creative Commons y Australia Post, entre otras, han construido su web con éxito por medio de un *framework* de diseño web, según informa el sitio del proyecto HTML5 Boilerplate.

En este paradigma, el desarrollo de webs se nutre de todo un sistema de plataformas, bibliotecas y editores. En términos generales, un *framework* es una estructura conceptual y tecnológica que sustenta el desarrollo de *software*. En el entorno particular del web, un *framework* de diseño web es una aplicación o un conjunto de módulos que permiten el desarrollo ágil de sitios mediante bibliotecas o funcionalidades preexistentes, aplicando una metodología de trabajo propia.

Un *framework* lo forma un conjunto estandarizado de recursos —bibliotecas, aplicaciones y plantillas— destinados a hacer desarrollos a medida:

- **Bibliotecas *frontend*** (orientadas a la interfaz). Proyectos de código abierto destinados al diseño de sitios y aplicaciones web. Contienen plantillas predefinidas de todo tipo de componentes: formularios, botones, cuadros y menús de navegación, entre muchos otros elementos de diseño. El material está basado en HTML5 y CSS3 e incluye extensiones de JavaScript. Son ejemplos de bibliotecas *frontend* los proyectos [Bootstrap](#), [Foundation](#), [Materialize](#) y [HTML5 Boilerplate](#).
- **Constructores** (*website builders*). Aplicaciones web o de escritorio orientadas a crear un sitio en apenas unos minutos. Algunos de los más populares son [Mobirise](#), [Pingendo](#), [Carrd](#), [VvwebJs](#) y [Layoutit](#).
- **Bibliotecas de plantillas**. Colecciones descargables de plantillas de diseño web completamente funcionales. Suelen basarse en alguna biblioteca *frontend* (en su mayoría, Bootstrap), aunque hay desarrollos independientes (p. ej., [HTML5 Up](#)) optimizados para favorecer el rendimiento. Tienen una licencia Creative Commons Attribution, lo que implica que pueden usarse de forma gratuita si se conserva la mención de autoría y el enlace a la fuente original, o bien, para usar el sitio sin esta atribución, se puede adquirir una licencia.

2. Cómo funciona un *framework* de diseño web

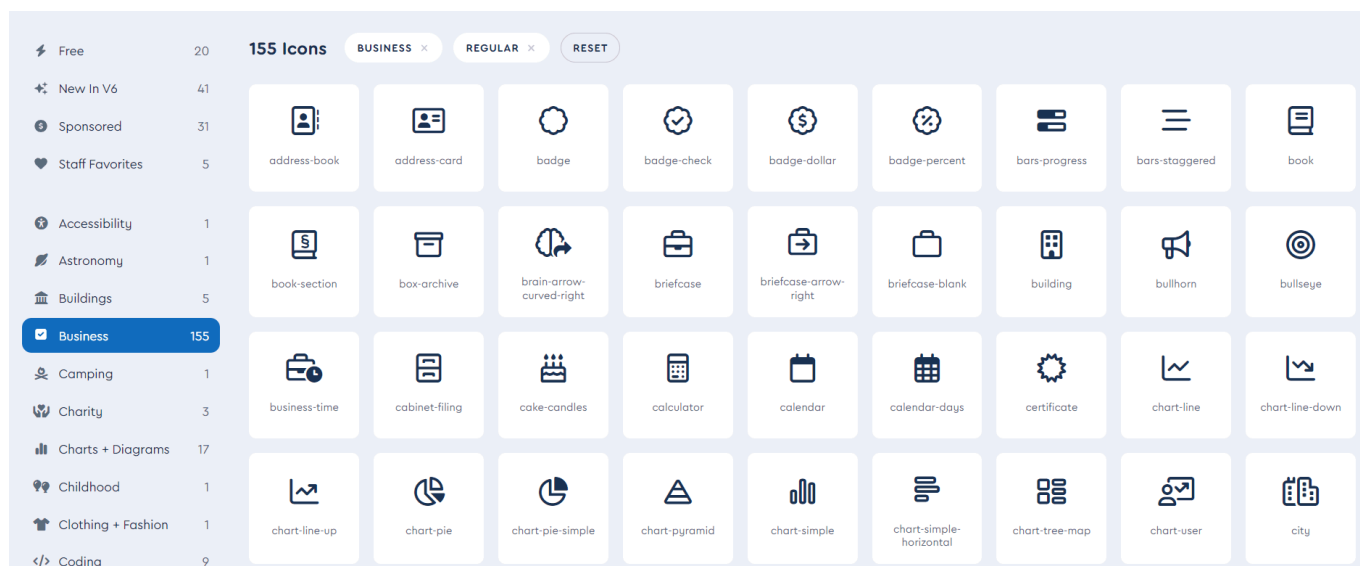
Los *frameworks* de diseño web ofrecen un esqueleto que facilita la tarea de construir **sitios reactivos** (*responsive*) que se adapten a cualquier dispositivo.



Ofrecen hojas de estilo CSS y funciones de JavaScript que pueden usarse directamente y ahorrar tiempo a la hora de generar código. La mayoría de constructores y colecciones de plantillas se basan en los materiales que ofrece la biblioteca del proyecto Bootstrap, un *framework* gratuito que agiliza y facilita el desarrollo de sitios web.

En cualquier caso, los *frameworks* reúnen unas **características** comunes:

- Incluyen **plantillas de diseño** que manejan todo tipo de componentes: la tipografía, formularios, tablas, elementos de navegación, carruseles de imágenes y elementos emergentes, entre muchos otros.
- Permiten crear **diseños reactivos** que se ajustan de manera automática a las dimensiones de la pantalla para que el contenido se pueda consumir acorde con las características físicas de cada dispositivo.
- Para aligerar el peso de los iconos, en lugar de utilizar imágenes, proponen el uso de **bibliotecas de glifos** basadas en CSS, tales como [FontAwesome](#), [Bootstrap Icons](#), [Boxicons](#), [IcoFont](#) o [Remix Icon](#).



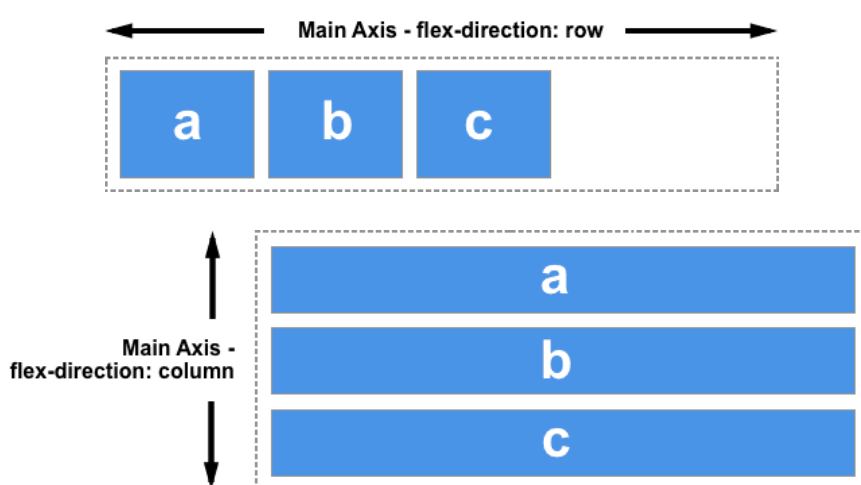
El contenido de las páginas se articula en un **sistema de retícula** construido con el módulo de caja flexible, conocido como *flexbox*, una recomendación del W3C que permite que los elementos adaptables dentro de un contenedor se organicen automáticamente según el tamaño de la pantalla o del dispositivo.

El método *flexbox* define cómo se organiza una serie de elementos que deben fluir en una dirección. Generalmente, los elementos se distribuyen en una retícula de 12 columnas de contenido, de forma que se pueden agrupar varias columnas en celdas más anchas de medidas proporcionales. Así, los elementos contenidos en una sección se pueden distribuir, p. ej., en 2, 3, 4 o 6 columnas de igual anchura, o bien en 2 columnas a una anchura desigual de 1/3 y 2/3, respectivamente, o bien en una columna única, entre otras posibilidades. Y esa colocación puede variar de manera automática según las dimensiones de la pantalla.

Es importante observar que la retícula se articula en 12 columnas porque ese es el menor número natural con 6 divisores, lo que permite muchas combinaciones armónicas.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Al ser un sistema reactivo, los bloques de contenido se organizan según el tamaño de la pantalla y se distribuyen de manera ordenada en filas o columnas.



Atrás queda, pues, la mala práctica de diseñar con tablas, que no se adaptan a los dispositivos móviles y, además, empeoran la accesibilidad del contenido.

2.1. Vincular una biblioteca *frontend* a un proyecto

Las bibliotecas *frontend* no requieren instalación. El código fuente se entrega listo para usar. En el caso particular de Bootstrap, para colocar esta biblioteca en un proyecto, basta con incluir en la cabecera de cada página un enlace a una **red de distribución de contenidos** (CDN, por las siglas de *content distribution network*), un servidor que contiene una copia de esos ficheros para que cualquiera pueda utilizarlos en su propio sitio sin necesidad de alojarlos en él. La biblioteca Bootstrap (v5.2.1), por ejemplo, se vincula colocando este fragmento de código en la sección *head* de cada página:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous" />
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u10knCvxwVY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTP00mMi466C8"
crossorigin="anonymous"></script>
```

Enlazar una biblioteca a través de una CDN tiene dos **beneficios** añadidos:

- Por definición, las CDN sirven el contenido desde el servidor más cercano, lo cual **aceleran la descarga**.
- Dado que muchos usuarios ya habrán visitado otros sitios basados en Bootstrap, esta biblioteca se mantendrá en su memoria caché y así **se reducirá el tiempo de descarga** de cualquier otro sitio basado en Bootstrap.

3. Plantillas prediseñadas

Quien no quiera desarrollar un sitio desde cero, puede agilizar la labor mediante una **plantilla prediseñada**. Las bibliotecas de plantillas, tales como *HTML5 Up* o *BootstrapMade*, ofrecen varias ventajas: son fáciles de manejar, compatibles con todos los navegadores, presentan un diseño reactivo y están orientadas al *mobile first*, lo que, en conjunto, mejora la experiencia de uso porque hace las interfaces fáciles de manejar, facilita la navegación y mejora el posicionamiento en buscadores porque sitúa el contenido importante en la parte superior del código fuente.

En estas colecciones, las plantillas incluyen todo un conjunto de componentes que se presentan a modo de muestra, como ejemplos de uso. En el desarrollo de un sitio no es necesario, pues, incluir todos los materiales o emplearlos tal como se entregan. Todo depende de la naturaleza del propio proyecto. La filosofía del desarrollo web a partir de plantillas se basa en la modificación de esos materiales con el fin de personalizarlos y adaptarlos a una necesidad particular. Los diversos componentes se pueden eliminar, cambiar, combinar, ampliar y elaborar en cualquier dirección.

La tarea es fácil y cuenta con un aliado fundamental: las herramientas de desarrollo (*DevTools*) del navegador. La funcionalidad de inspección (Ctrl+Mayús+I) que incorporan navegadores como *Chrome* o *Firefox* permiten, no solo examinar el código HTML y CSS, sino también modificarlo y probarlo en tiempo real.

All Web Design Branding Photography



```

</div>
<div class="col-md-12 col-lg-6 text-left
text-lg-right aos-init aos-animate" data-aos=
"fade-up" data-aos-delay="100">
  <div id="filters" class="filters">
    <a href="#" data-filter="*" class=
    "active">_</a>
    <a href="#" data-filter=".web">Web</a>
    <a href="#" data-filter=".design">Design
    </a>
    <a href="#" data-filter=".branding">
    Branding</a>
    ...
    <a href="#" data-filter=".photography">
    Photography</a> == $0
  </div>
</div>
</div>
<div id="portfolio-grid" class="row no-gutter
aos-init aos-animate" data-aos="fade-up" data-
aos-delay="200" style="position: relative;
height: 748px;">_</div>
</div>
</section>

```

xt-left.text-lg-right.aos-init.aos-animate div#filters.filters a ...

Styles Computed Event Listeners DOM Breakpoints >>

font-family :hov .cls + <

Inherited from body

body { style.css:11

font-family: "Inconsolata", monospace;

color: #0d1e2d;

body { _reboot.scss:47

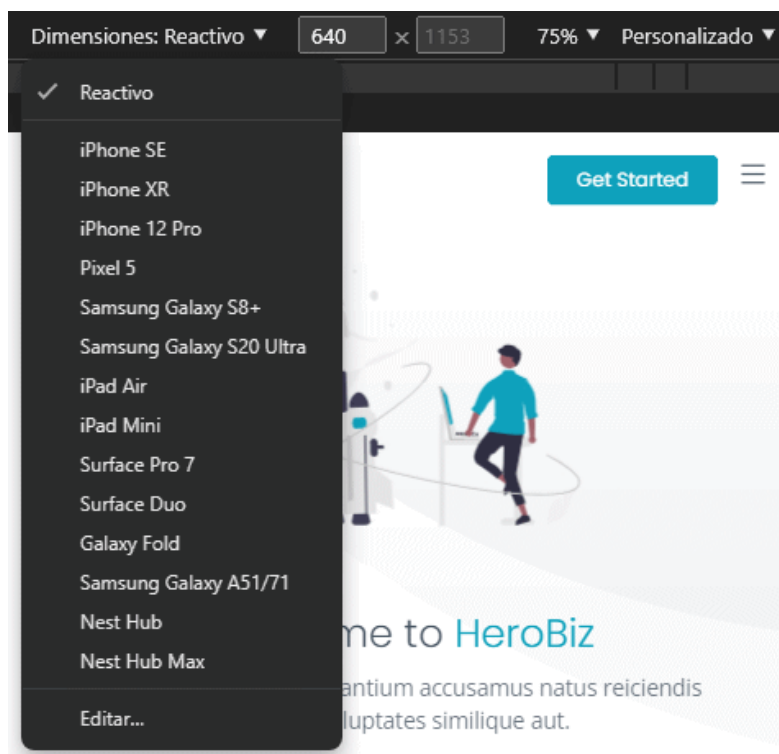
margin: 0;

font-family: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";

font-size: 1rem;

font-weight: 400;

Además, permiten probar el contenido en varias configuraciones de pantalla predefinidas. Así se puede verificar cómo el contenido se acomoda a diferentes tamaños de pantalla y en dispositivos comunes como un iPhone, un iPad o un teléfono móvil cualquiera.



4. Descargar y adaptar una plantilla

Existen varios depósitos gratuitos (y semigratuitos) de plantillas prediseñadas, tales como [HTML5 Up](#) y las colecciones basadas en el proyecto Bootstrap: [BootstrapMade](#), [TemplateMo](#), [W3Layouts](#), [Start Bootstrap](#) o [Theme Wagon](#). Las plantillas, basadas en HTML5 optimizado y CSS3, suelen estar disponibles bajo licencias Creative Commons y se caracterizan por ser totalmente reactivas y ampliamente personalizables.

Se pueden examinar a través del botón «Demo» o «Live Demo». La descarga gratuita es posible en la opción «Free Download», aunque conviene prestar atención a las condiciones de uso. La licencia Creative Commons exige respetar el reconocimiento de autoría y prohíbe eliminar, en consecuencia, el enlace de crédito que se encuentra a pie de página.

	OnePage Free Limited	OnePage Pro Extra Features	Membership Get 90 templates
	\$0	\$19 One time payment Lifetime access	\$49 For 3 months access No auto renewal
Use in personal projects	✓	✓	✓
Number of websites/domains	1	1	Unlimited
Create websites for clients	✗	✓	✓
Remove footer credit link	✗	✓	✓
Working PHP/AJAX contact form	✗	✓	✓
Premium support via Email	✗	✓	✓
Sass/SCSS Files	✗	✓	✓
Access to all our 90 pro templates	✗	✗	✓
	Free Download	Buy Now	Buy Now

A continuación se muestran algunas de las muchas posibilidades de personalización, tomando como ejemplo, de la colección *BootstrapMade*, la plantilla denominada *Impact*. Cada plantilla incluye los componentes básicos: varios modelos de página (documentos HTML), hojas de estilo, secuencias JavaScript e imágenes de recurso.

blog.html	Modelo de página de navegación de blog
blog-details.html	Modelo de página informativa de blog
index.html	Modelo de página inicial
portfolio-details.html	Modelo de página de producto
readme.txt	Información sobre la plantilla
sample-inner-page.html	Modelo de página informativa
assets	Complementos
css	Hojas de estilo
main.css	Hoja de estilo principal
img	Imágenes
js	Secuencias de JavaScript
main.js	Funciones para mejorar la interactividad
scss	Hojas de estilo SCSS
vendor	Materiales de apoyo
forms	Formulario (solo funcional en la versión Pro)

4.1. Nociones de HTML y CSS

Para obtener el máximo provecho de las plantillas prediseñadas se necesita al menos un conocimiento básico de HTML y CSS. Quien no haya manejado antes estos lenguajes, encontrará ayuda en estos recursos:

- [Shay Howe: Learn to Code HTML & CSS](#)
- [Codecademy: HTML & CSS](#)
- [Channel 9: HTML5 & CSS3 Fundamentals](#)

A quien los conozca y solo necesite una puesta al día le serán útiles:

- [Shay Howe: Learn to Code Advanced HTML & CSS](#)
- [CSS-Tricks](#)

4.2. Herramientas de edición

Cualquier editor de texto llano es válido para trabajar con código HTML y CSS. Sin embargo, hará la tarea más fácil un editor dedicado:

- [Sublime Text](#) (Windows, macOS, Linux)
- [Atom](#) (Windows, macOS, Linux)
- [Notepad++](#) (Windows)

4.3. Estructura de un modelo de página

Cada modelo de página de una plantilla se articula en diversas secciones, cada una con una función distinta. Así está organizada la página inicial (*index.html*) de la plantilla *Impact*:

```
<body>
  <section id="topbar"></section> ..... Barra superior: datos de contacto
  <header> ... <nav></nav> ... </header> ..... Cabecera y barra de navegación
  <section id="hero"></section> ..... Sección hero
  <main> ..... Contenido principal, abarca:
    <section id="about"></section> ..... Sección "Acerca de"
    <section id="clients"></section> ..... Carrusel de clientes
    <section id="stats-counter"></section> ..... Datos cuantitativos
    <section id="call-to-action"></section> ..... Llamada a la acción destacada
    <section id="services"></section> ..... Carta de servicios
    <section id="testimonials"></section> ..... Recomendaciones de clientes
    <section id="portfolio"></section> ..... Muestrario
    <section id="team"></section> ..... Orla
    <section id="pricing"></section> ..... Planes y precios
    <section id="faq"></section> ..... Preguntas frecuentes
    <section id="recent-posts"></section> ..... Publicaciones recientes
    <section id="contact"></section> ..... Formas de contacto
  </main>
  <footer></footer> ..... Pie de página o faldón
</body>
```

4.4. Manos a la obra

4.4.1. Primeros pasos

El documento *index.html* contiene un modelo de página inicial, que se puede usar como único elemento para un sitio de página única o bien como página principal de un sitio multipágina. Al abrirlo, en la sección *head* se encuentra el título de la página y los metadatos descriptivos:

```
<title>Impact Bootstrap Template - Index</title>
<meta content="" name="description">
<meta content="" name="keywords">
```

Estos elementos se pueden adaptar a las características del proyecto propio. Solo hay que sustituir el título y rellenar los metadatos *description* y *keywords* como corresponda.

```
<title>Eventos Cuatro Vientos</title>
<meta content="Agencia de organización de eventos en Barcelona. Creamos y
organizamos eventos para empresas. Todo lo que necesitas, sin complicaciones."
name="description">
<meta content="eventos, celebraciones, turismo de negocios" name="keywords">
```

A continuación aparecen los elementos *link* que referencian al icono de página (*favicon*), una pequeña imagen que los navegadores mostrarán en la barra de direcciones y en el encabezado de la pestaña correspondiente.

```
<!-- Favicons -->
<link href="assets/img/favicon.png" rel="icon">
<link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">
```

Por defecto, la plantilla muestra el logotipo de Bootstrap, que se puede sustituir por cualquier otro si se reemplazan los ficheros *favicon.png* y *apple-touch-icon.png* de la subcarpeta *assets/img* por los que se quieran emplear en su lugar.

Ya en la sección *body*, donde se encuentra el contenido visible, se pueden modificar aspectos del contenido. Esta sección de la plantilla se articula en módulos cuyo código fuente se encuentra identificado con una línea de comentario. La cabecera comienza en:

```
<!-- ===== Header ===== -->
```

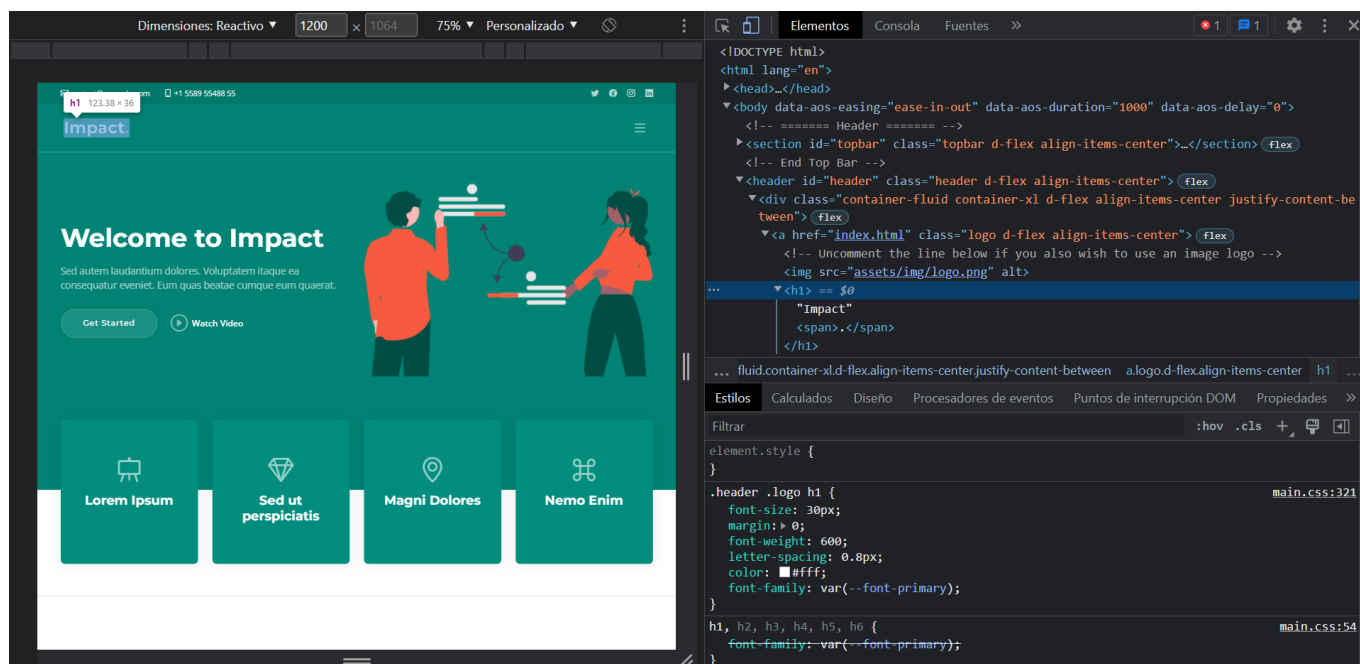
El título y el logotipo de la página se pueden modificar en la sección *header*.

```
<header id="header" class="header d-flex align-items-center">
  <div class="container-fluid container-xl d-flex align-items-center
  justify-content-between">
    <a href="index.html" class="logo d-flex align-items-center">
<!-- Uncomment the line below if you also wish to use an image logo -->
<!--  -->
      <h1>Impact<span>.</span></h1>
    </a>
```

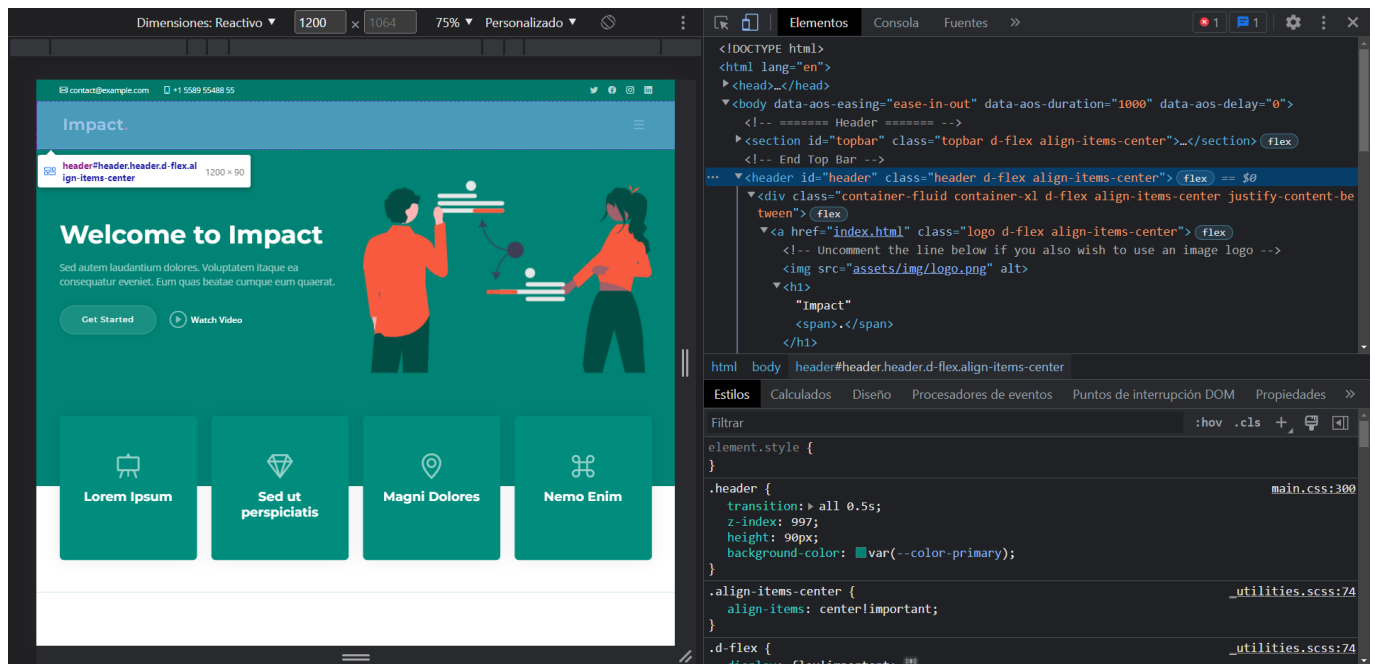
El título visible de la página está contenido en el elemento *h1*, pero, tal como indica el comentario precedente, se puede sustituir o complementar con una imagen si se habilita el elemento *img* que, por defecto, aparece deshabilitado como un comentario. Para sustituirlo, se habilita esa línea y se inhabilita el elemento *h1*. El elemento desechado se podría borrar, pero conviene ser conservadores y, en lugar de ello, se convierte en un comentario cuyo contenido será ignorado por los navegadores.

```
<header id="header" class="header d-flex align-items-center">
  <div class="container-fluid container-xl d-flex align-items-center justify-content-between">
    <a href="index.html" class="logo d-flex align-items-center">
<!-- Uncomment the line below if you also wish to use an image logo -->
    
<!-- <h1>Impact<span>.</span></h1> -->
    </a>
```

Si el logotipo se viera demasiado pequeño, el tamaño se puede adaptar simplemente ajustando el espacio reservado por defecto. Para averiguar de qué elemento conviene modificar las propiedades de estilo, se puede hacer clic sobre él con el botón derecho y seleccionar la opción «Inspeccionar». Al hacer esta operación sobre el logotipo, se abrirán las herramientas de desarrollo, que además de mostrar el código fuente de la página y de las hojas de estilo activas, permitirán probar la página con distintas configuraciones de pantalla.



Según se mueva el puntero sobre el código fuente, la herramienta de desarrollador resaltará a la izquierda, sobre la página, el área afectada. Al ascender hasta el elemento *header* se observa lo siguiente:



Según se observa entre los estilos, este elemento, con la clase `.header`, tiene asignada una altura de 90 píxeles.

```
.header {
  transition: all 0.5s;
  z-index: 997;
  height: 90px;
  background-color: var(--color-primary);
}
```

Para aumentar su altura, basta con modificar el valor de la propiedad `height` (por ejemplo, a 140 píxeles).

```
.header {
  transition: all 0.5s;
  z-index: 997;
  height: 140px;
  background-color: var(--color-primary);
}
```

El cambio se puede previsualizar sobre la página si se modifica ese valor en la herramienta de desarrollo. Este cambio será temporal. Para hacerlo definitivo, habrá que introducirlo en la hoja de estilo y guardarla de nuevo. La ubicación exacta de cada selector la señala la herramienta a su derecha. El selector `.header` se encuentra en:

`main.css:300`

Es decir, en la línea 300 del fichero `main.css`.

La misma sección `header` contiene la barra de navegación, dentro del elemento `nav`, que también se puede modificar. Las diversas opciones y subopciones del menú están articuladas como listas anidadas y se pueden adaptar a las características del propio proyecto.

```

<nav id="navbar" class="navbar">
  <ul>
    <li><a href="#hero">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#services">Services</a></li>
    <li><a href="#portfolio">Portfolio</a></li>
    <li><a href="#team">Team</a></li>
    <li><a href="blog.html">Blog</a></li>
    <li class="dropdown"><a href="#"><span>Drop Down</span>
    <i class="bi bi-chevron-down dropdown-indicator"></i></a>
    <ul>
      <li><a href="#">Drop Down 1</a></li>
      ...
      <li><a href="#">Drop Down 2</a></li>
      <li><a href="#">Drop Down 3</a></li>
      <li><a href="#">Drop Down 4</a></li>
    </ul>
    </li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>

```

En la plantilla, algunas opciones apuntan a secciones de la misma página definidas mediante el atributo *id* y referenciadas aquí mediante marcadores (*#hero*, *#about*, *#services*, etc.) y otras apuntan a otras páginas del sitio indicando el nombre del fichero que debe ser el destino del enlace (*blog.html*, etc.). Esta navegación es adaptable a las características del propio proyecto. Solo hay que eliminar o ampliar partes de esta estructura y adaptar el destino y el texto de anclaje de los enlaces.

Tras la cabecera de la página se encuentra el área *hero*, destinada a destacar contenido orientado al posicionamiento de marca.

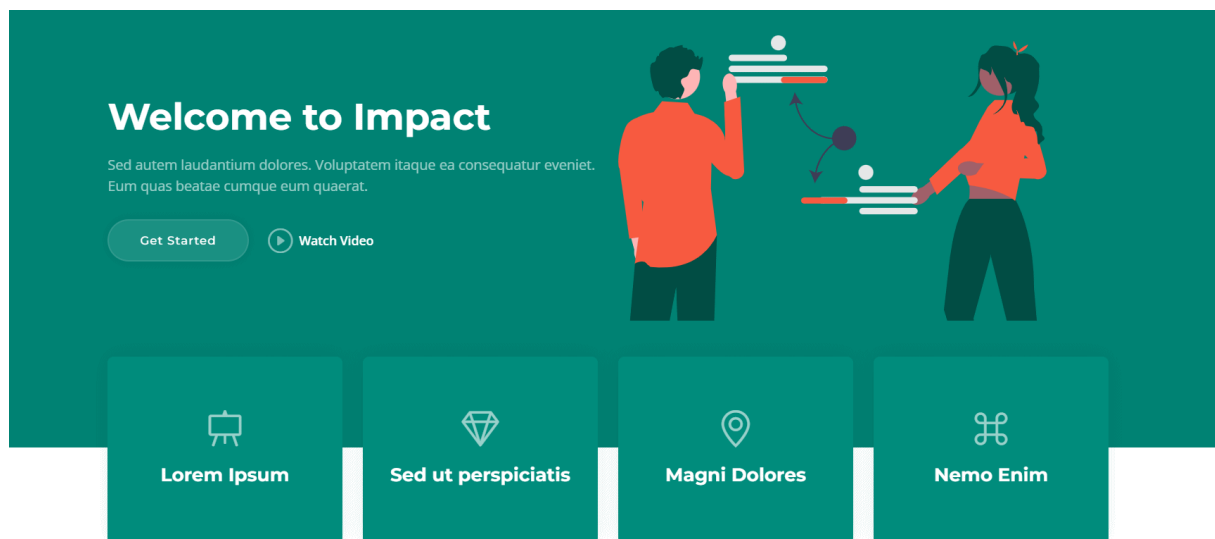
```

<!-- ===== Hero Section ===== -->
<section id="hero" class="hero">

```

Debe contener mensajes cortos, concisos y relevantes, y también imágenes o vídeos, que en conjunto ofrezcan una idea clara de cuál es la oferta (productos o servicios) de la marca. A veces pueden incluir una llamada a la acción.

En la plantilla *Impact*, la sección *hero* presenta este aspecto.



Buena parte del contenido textual es *lorem ipsum*, o sea, texto que se usa habitualmente en borradores de diseño para probar los aspectos gráficos de un documento, tales como la tipografía y la composición, antes de insertar el texto definitivo. Igual que todo el contenido de la plantilla, los textos se deberán sustituir por el contenido que corresponda.

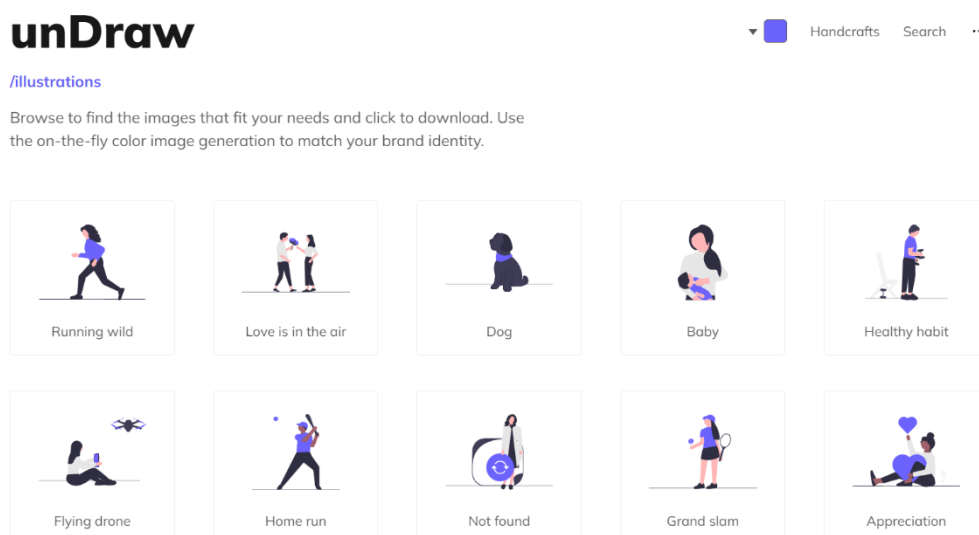
Adaptemos primero la imagen *hero*. Para averiguar qué elemento se debe modificar, siguiendo las indicaciones anteriores, bastará con hacer clic derecho sobre ella y seleccionar «Inspeccionar». Esta acción abrirá de nuevo la herramienta de desarrollo y apuntará a este elemento *img*:

```

```

La extensión del fichero *hero-img.svg* denota que es una imagen vectorial en formato SVG. Es un formato ideal para este tipo de usos gráficos porque es escalable y ligero, y además admite transparencias.

Para nutrir esa sección con otras imágenes se puede acudir al sitio [unDraw](https://unDraw.co/), que ofrece ilustraciones gratuitas en este mismo formato. Por defecto, las imágenes tienen el violeta como color dominante, pero se puede elegir otro.

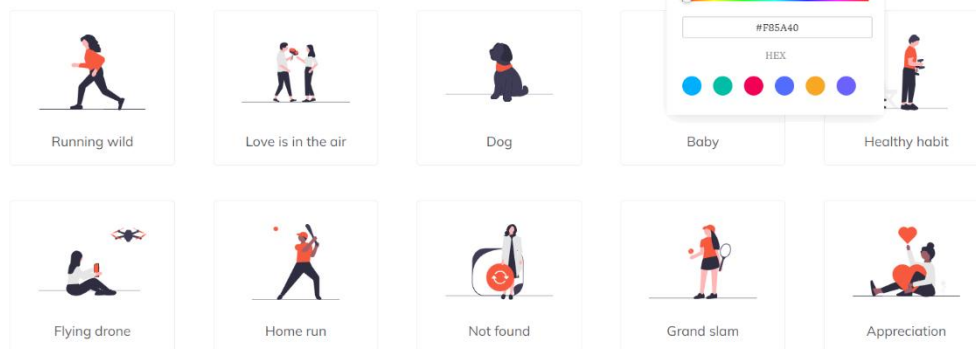


La barra de navegación ofrece una paleta para escoger cualquier otro color, del que se puede proporcionar un valor RGB exacto (en notación hexadecimal). El gráfico incluido en la plantilla *Impact* tiene como color dominante el naranja codificado como #F85A40.

unDraw

/illustrations

Browse to find the images that fit your needs and click to download. Use the on-the-fly color image generation to match your brand identity.



Tomemos, por ejemplo, el grafismo «People», disponible para la descarga en los formatos SVG (vectorial) y PNG (matricial).

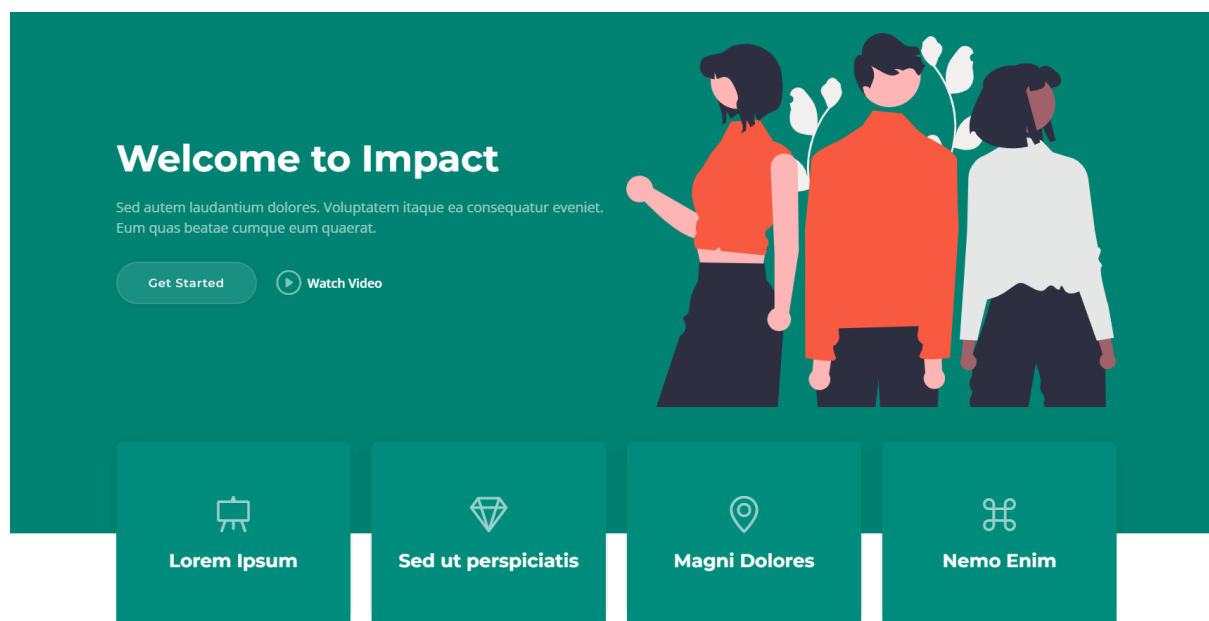


People

Download SVG for your projects

Download PNG for your blog / social

En formato SVG se obtendrá un fichero llamado *undraw_people_re_8spw.svg*, que se puede renombrar como *hero-img.svg* para sustituirlo por el incluido en la plantilla. Aplicado este cambio, así lucirá ahora el área *hero*.



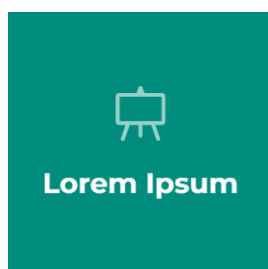
Los iconos que aparecen en las cajas (y en cualquier otra sección) también se pueden personalizar. Proceden de una biblioteca de código abierto de glifos. Los glifos representan iconos comunes de una forma muy ligera. En lugar de emplear ficheros de imagen, que suelen ser pesados y ralentizan la descarga, se utiliza un único fichero de tipografía, del cual se extraen los glifos mediante simples clases CSS.

De bibliotecas de glifos hay varias: *FontAwesome* (la más completa y popular), *IcoFont*, *BoxIcons*, *Remix Icon* y *Bootstrap Icons*, entre otras. Una misma plantilla puede nutrirse de más de una biblioteca de glifos, aunque por razones de rendimiento y eficiencia es recomendable usar solo una. El modo más sencillo de averiguar a qué biblioteca pertenecen los iconos de la plantilla consiste en inspeccionar un icono (elemento *i*) y deducirlo según la clase que se usa en el código fuente. La clave está en el prefijo:

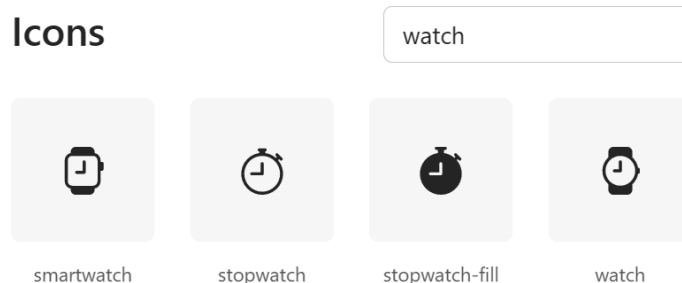
- fa = [FontAwesome](#)
- bx = [BoxIcons](#)
- rx = [Remix Icon](#)
- bi = [Bootstrap Icons](#)

La primera caja de la sección *hero* incluye el glifo de un caballete (*easel*), extraído de *Bootstrap Icons*, la biblioteca nativa de Bootstrap.

```
<i class="bi bi-easel"></i>
```

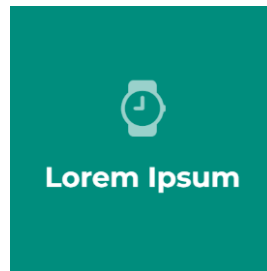


¿Cómo sustituirlo, por ejemplo, por el glifo de un reloj de muñeca? Solo hay que acceder a [Bootstrap Icons](#) y lanzar una búsqueda (en inglés):



Se obtienen cuatro glifos, con los nombres *smartwatch*, *stopwatch*, *stopwatch-fill* y *watch*, respectivamente. Para incorporar este último, sustituimos el valor *bi-easel* por *bi-watch* entre las clases del elemento *i*.

```
<i class="bi bi-watch"></i>
```



Y es así como se pueden personalizar todos los iconos de cualquier plantilla.

El texto destacado se puede modificar, simplemente reescribiendo el contenido del elemento que lo contiene.

```
<h4 class="title"><a href="" class="stretched-link">Lorem Ipsum</a></h4>
```

Supongamos que queremos destacar la puntualidad del servicio como rasgo diferencial. Para ello, basta con sustituir «Lorem Ipsum» por «Rapidez».

```
<h4 class="title"><a href="" class="stretched-link">Rapidez</a></h4>
```

Si, además, se quiere añadir una frase explicativa, se puede insertar un elemento de párrafo (*p*).

```
<h4 class="title"><a href="" class="stretched-link">Rapidez</a></h4>
<p>Garantizamos una respuesta inmediata a cualquier solicitud.</p>
```

He aquí el resultado:



4.4.2. Retícula y *flexbox*

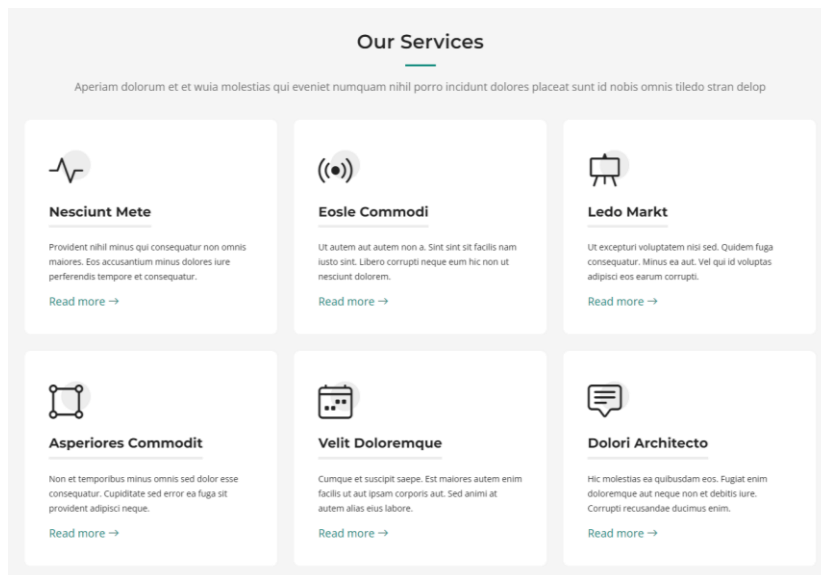
Los *frameworks* de diseño web suelen incluir **clases utilitarias** para controlar la distribución de elementos en el espacio. Se puede observar un ejemplo en la sección *services*.

```
<section id="services" class="services sections-bg">
```

En esta área hay 6 cajas (elementos *div*) organizadas con dos clases utilitarias cada una:

```
<div class="col-lg-4 col-md-6">
...
</div>
```

El valor *col-lg-4* se aplica en pantallas de gran anchura, donde cada caja ha de ocupar el espacio equivalente a 4 celdas de las 12 que abarca el sistema de retícula, lo que significa que en cada fila cabrán 3 cajas ($4 \times 3 = 12$).



El valor *col-md-6*, en cambio, se aplica en pantallas de anchura mediana, donde cada caja ha de ocupar 6 celdas de las 12 del sistema de retícula, lo que significa que en cada fila cabrán 2 cajas ($6 \times 2 = 12$).

La definición de lo que es grande, mediano o pequeño la establecen los **puntos de ruptura** (*breakpoints*) especificados en la hoja de estilo. En el diseño reactivo, los puntos de ruptura son medidas de anchura donde la página cambia la disposición de los elementos y aplica propiedades de estilo concretas.

Los valores de distribución de cajas se pueden cambiar a voluntad para que en cada configuración de pantalla se acomoden al espacio disponible de una forma u otra.

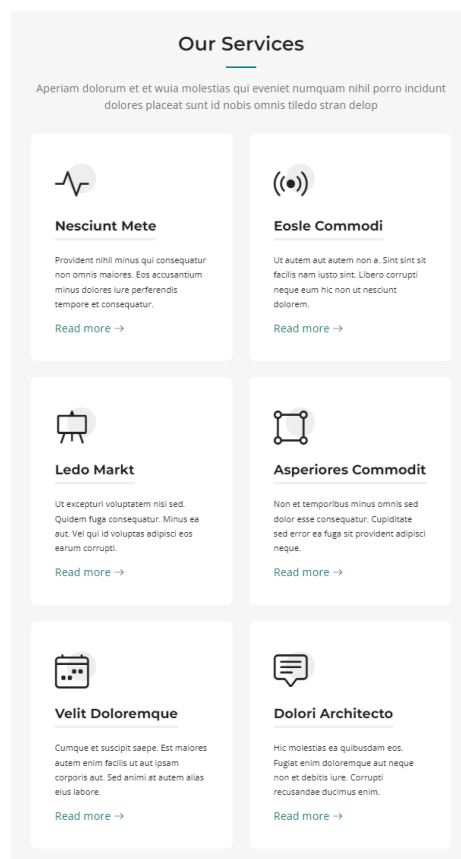
Si se sustituyen estas clases...

```
<div class="col-lg-4 col-md-6">
```

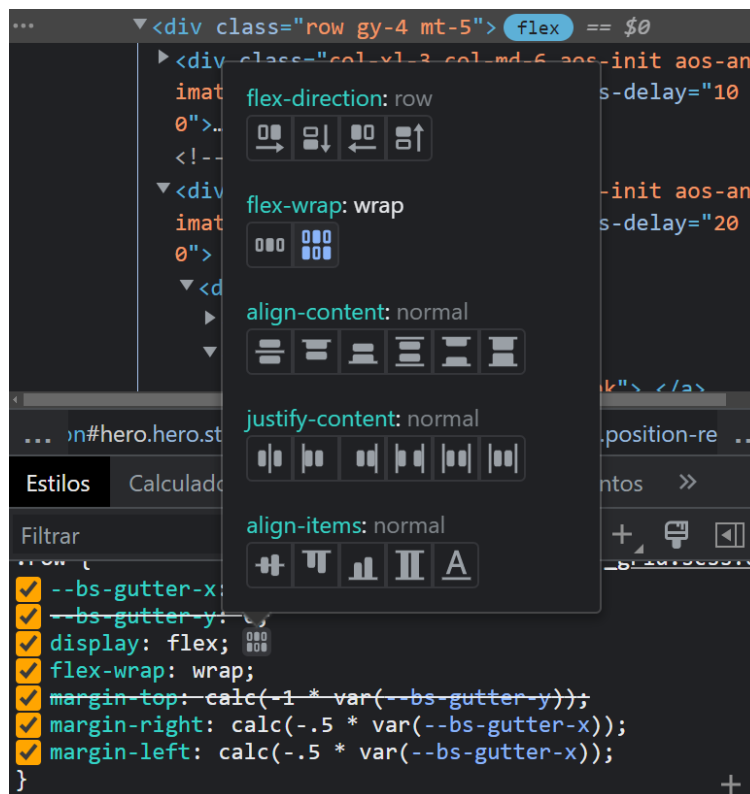
... por estas otras...

```
<div class="col-lg-3 col-md-4">
```

... el resultado será que, en pantallas grandes, las cajas se distribuirán en 4 columnas por fila (porque *col-lg-3* asignará 3 celdas de la retícula a cada una, hasta llenar las 12 celdas), mientras que, en pantallas medianas, las cajas se articularán en 3 columnas (porque *col-md-4* asignará 4 celdas a cada una).



La colocación de los elementos que forman parte de un contenedor flexible (*flexbox*) se puede modificar por medio de varias propiedades. Los contenedores flexibles aparecen identificados en la herramienta de inspección con un botón «flex». Al inspeccionar cualquiera de ellos, junto a la propiedad de CSS *display* se puede ver un botón que conduce al editor de *flexbox*.



Para cambiar el modo de presentación de los elementos, basta con elegir entre las opciones disponibles. La herramienta de inspección añadirá las propiedades y los valores correspondientes, y mostrará en tiempo real el resultado. Cuando se obtenga el resultado esperado, bastará con trasladar las nuevas propiedades a la hoja de estilo.

4.4.3. Color

Algunas plantillas asignan el color de los elementos mediante propiedades *color*. En cambio, otras, como la utilizada aquí como ejemplo, utilizan *Sass*, un procesador de CSS que ayuda a modularizar los estilos y a reducir el tiempo de mantenimiento en proyectos grandes.

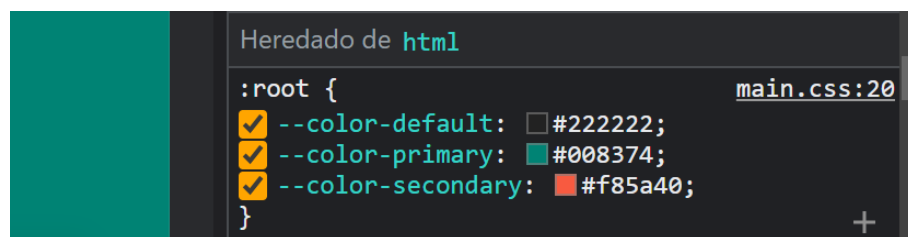
En cualquier caso, los colores se especifican mediante códigos RGB hexadecimales. El sistema RGB describe cómo se compone el color en términos de intensidad de los colores primarios de la luz: rojo (R, *red*), verde (G, *green*) y azul (B, *blue*).

El sistema hexadecimal, por su parte, es un sistema de numeración que tiene como base aritmética el número 16 (a diferencia del sistema decimal, que tiene el 10). Lo compone el conjunto de símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F, en el que las letras corresponden a los valores numéricos del 10 al 15 (A=10, B=11, etc.).

Se utiliza para simplificar expresiones binarias, tales como los 24 bits que representan el valor RGB de un color, que se expresan mediante tres pares de dígitos hexadecimales (un par para cada componente de color). Así, un color definido con el valor `#ffff00` representaría el color amarillo, dado que otorga la máxima luminosidad (`ff`) a los componentes rojo y verde, y ninguna luminosidad (`00`) al azul.

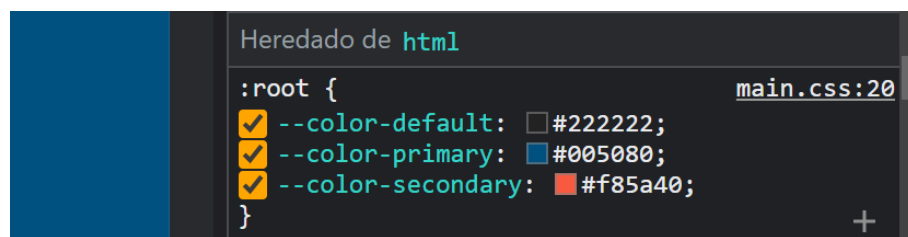
Aplicaciones web como [Adobe Color](#), [Coolors](#) o [Data Color Picker](#) permiten elegir un color y obtener el código RGB correspondiente, así como propuestas de paletas basadas en reglas de armonía cromática (analogía, complementariedad, etc.) que aseguran una combinación equilibrada.

Si se quiere cambiar el color primario de la plantilla, primero se debe inspeccionar el elemento *body* para averiguar qué propiedad asigna ese color. En la herramienta de desarrollo se observa lo siguiente:



El color primario se define en la línea 20 de la hoja de estilo *main.css*, en la propiedad personalizada `--color-primary`. El valor `#008374` expresa el tono aguamarina que aparece a la izquierda como color de fondo.

Al cambiarlo por un valor ligeramente distinto (p. ej., `#005080`) se obtiene otro color de fondo. Si el tono es satisfactorio, solo queda sustituir el original por el nuevo en esa línea 20 de la hoja de estilo *main.css*.



Otros elementos pueden tener definido el color frontal o de fondo mediante las propiedades `color` o `background`. Las cajas de la sección *hero*, p. ej., tienen, entre otras, la propiedad siguiente:

```
background:#008d7d
```

Si se le cambia el valor por este otro...

```
background:#006099
```

... se consigue colorear las cajas con un tono azulado, en armonía con el nuevo color de fondo.



El color de frente o de fondo del resto de elementos se puede cambiar siguiendo este mismo procedimiento, siempre recordando incorporar a la hoja de estilo los cambios aplicados temporalmente en la herramienta de inspección.

De igual forma, se puede modificar a voluntad el contenido, la estructura y la navegabilidad de cualquiera de las páginas incluidas en la plantilla. Basta con aplicar, adaptadas a cada contexto particular, las indicaciones expuestas aquí.

Referencias

Sucuri (2022). *2021 website threat research report*. Menifee, Calif.: Sucuri Security. <https://sucuri.net/wp-content/uploads/2022/04/22-sucuri-2021-hacked-report.pdf>

W3Techs (2022). «Usage statistics of content management systems». *W3Techs: web technology surveys*. https://w3techs.com/technologies/overview/content_management

Cómo citar este documento

Franganillo, Jorge (2024). *Construir un sitio web reactivo con plantillas prediseñadas*. Material docente. <<https://franganillo.es/plantillas.pdf>>