

Bootstrap (Parte I)

Introducción al Proyecto

Bienvenidos a esta nueva unidad denominada "Mejorando la experiencia y diseño de los sitios web con **Bootstrap**".

En esta unidad crearemos un sitio web responsive usando todo lo que hemos aprendido en la unidad anterior de HTML y CSS.

Lo que haremos será:

- Conocer el framework de CSS Bootstrap
- La filosofía mobile first
- Técnicas de Bootstrap para realizar páginas responsivas

El proyecto que haremos será el sitio web de Olivia Ros.



Acerca de mí

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus. Nunc at enim et lacus porttitor tincidunt. Donec imperdiet augue nec aliquet sagittis.



FOTÓGRAFA DE COMIDA



CRÍTICA CULINARIA



BLOGGERA DE VIAJES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

Mis trabajos

Asado en la Patagonia
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

[Ver más](#)

Papas del fin del mundo
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

[Ver más](#)

Salmones de Alaska
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam a mauris id purus blandit dapibus.

[Ver más](#)

Contáctame

Nombre

Correo electrónico

Mensaje

[Enviar](#)

Olivia Ros 2019. Todos los derechos reservados.

Imagen 1: Proyecto a desarrollar

Este sitio esta pensado en dar énfasis a la marca personal de su dueña llamada Olivia Ros. En ella podremos encontrar una barra de navegación, cuatro secciones que presentan un header donde aparecen dos botones, una sección que presenta lo que hace, sus últimos trabajos y una sección de contacto, junto con footer.

Para crear esta página usaremos distintos conceptos que nos servirán para hacer de esta página una página responsive, es decir, que busca la correcta visualización de una misma página en distintos dispositivos.

- Comenzaremos por conocer Bootstrap, un framework CSS que nos ayudará a hacer páginas responsivas de manera fácil y rápida.
- También aprenderemos a trabajar usando la filosofía mobile first.
- Integraremos Bootstrap y comenzaremos a construir el proyecto usando HTML y CSS.

Al finalizar tendremos todas las competencias suficientes para crear una página web responsive.

Introducción a Bootstrap

En este capítulo entraremos en el gran mundo de los frameworks CSS.

Conociendo los frameworks CSS

Comencemos por aprender qué es un framework. En desarrollo nos referimos a un framework como un conjunto de conceptos, prácticas y criterios enfocados en solventar una problemática particular.

Un framework CSS provee una estructura básica que nos ayudará a resolver problemas típicos de estilo dentro del desarrollo web.

Dentro de los Frameworks CSS existe una gran cantidad de opciones para elegir como:

- Bootstrap
- Foundation
- Bulma
- Semantic UI
- UI kit

La mejor opción para elegir dependerá de lo que queramos hacer y lo que ya conozcamos preliminarmente.

En nuestro caso, usaremos uno de los framework CSS más completos para crear páginas web responsivas llamado Bootstrap, que además hoy en día es el más popular.

¿Qué es Bootstrap?

Bootstrap es un framework CSS, desarrollado y liberado por Twitter, que sirve para maquetar páginas web de manera rápida y sencilla. Es de código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.

¿Qué hace tan popular a Bootstrap?

Las características que hacen de Bootstrap el framework CSS más popular de la web son:

- **Fácil de integrar:** Para integrar bootstrap a nuestro proyecto necesitaremos ingresar a [Get Bootstrap](#), copiar el CDN, pegarlo dentro de proyecto y ¡listo!
- **Sistema de grillas:** Bootstrap también incluye un sistema de grillas muy completo con el cual podremos crear layout de manera muy fácil.
- **Estilos en la mayoría de los elementos HTML:** Los elementos HTML como listas, botones, formularios, entre otros., vienen estilizados y mejorados con clases. De manera que nosotros no debemos escribir mucho CSS para estilizar los elementos.
- **Una gran lista de componentes:** Contiene una gran cantidad componentes que podremos usar en nuestra página web.
- **Gran documentación:** Dentro de la página de [Bootstrap](#) podremos encontrar una excelente documentación, donde nos enseñan con ejemplos a utilizar Bootstrap. Ésta es mantenida por su gran comunidad y desarrolladores.

Ejemplos de sitios realizados con Bootstrap

<https://www.licornpublishing.com/>

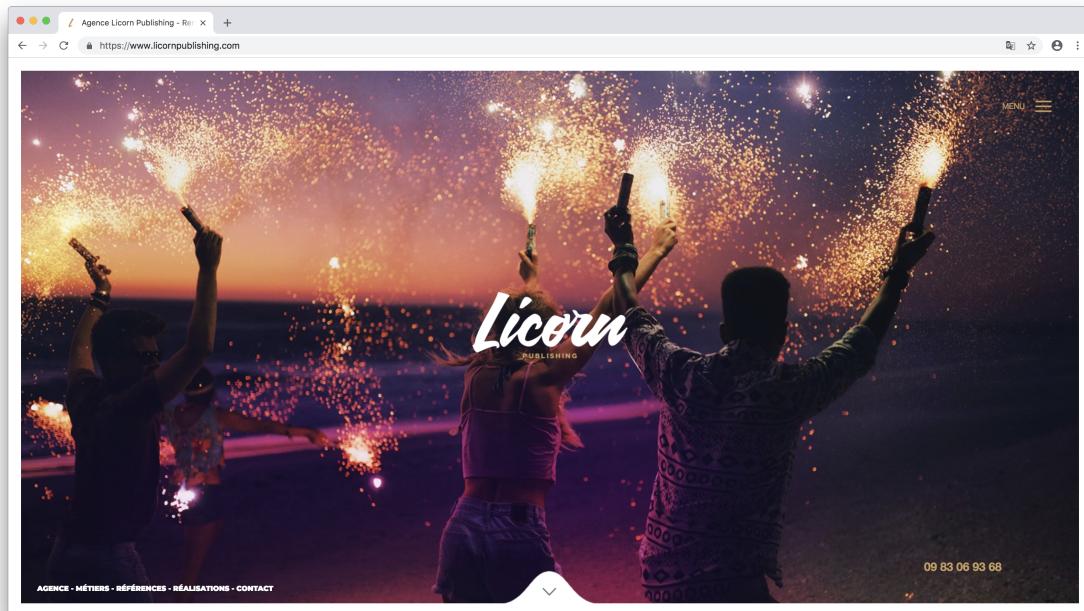


Imagen 2: Sitio desarrollado con Bootstrap - Licorn Publishing

<https://www.1bite2go.com/en/>

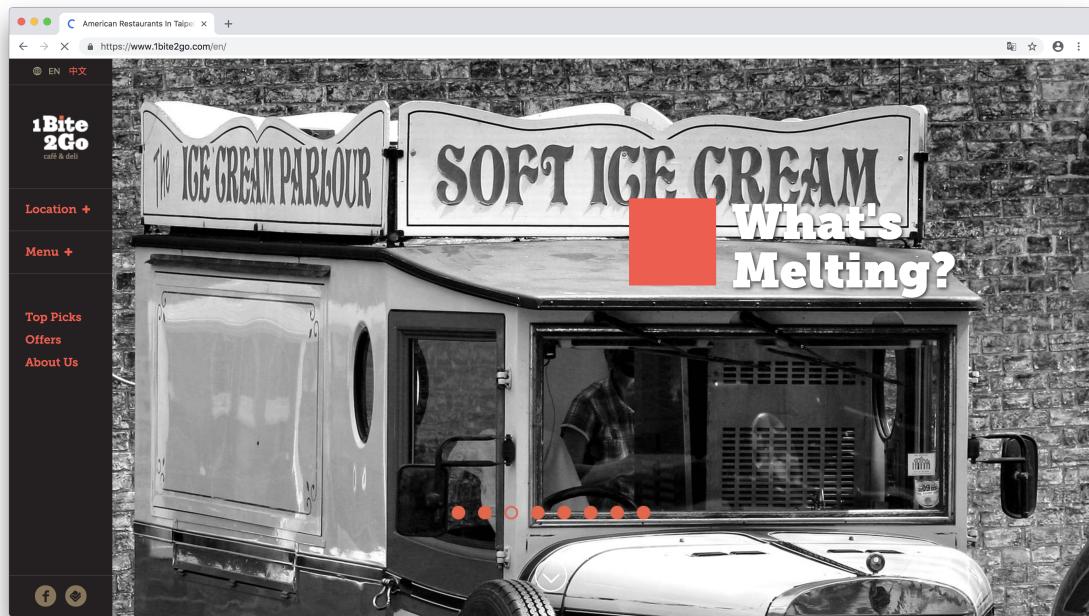


Imagen 3: Sitio desarrollado con Bootstrap - 1Bite2Go

<https://thisisneat.com.au>

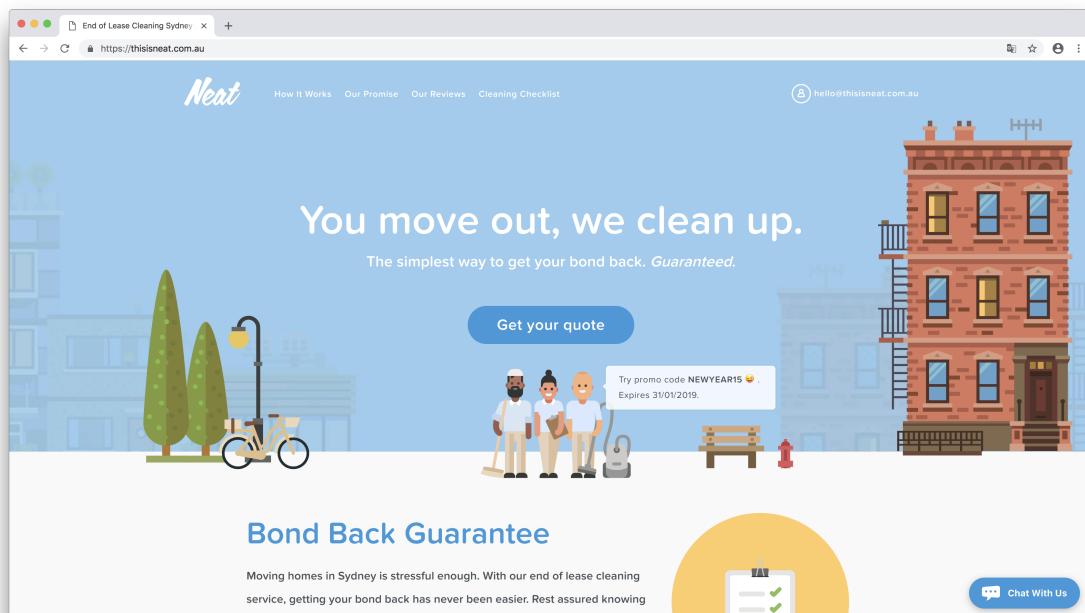


Imagen 4: Sitio desarrollado con Bootstrap - Neat

<http://kingdomrush.com>

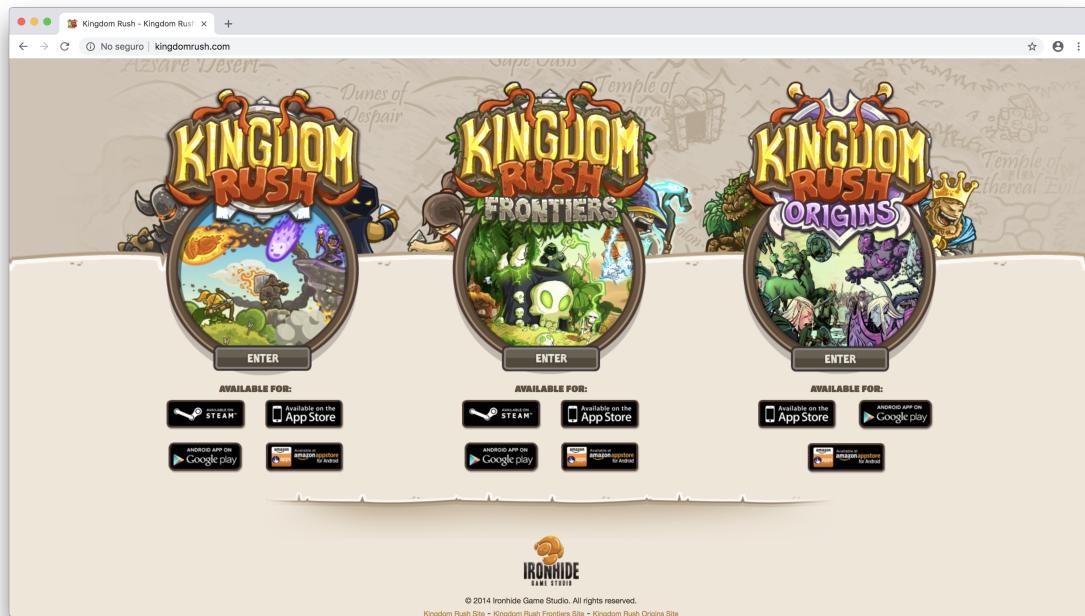


Imagen 5: Sitio desarrollado con Bootstrap - Kingdom Rush

Como hemos visto, Bootstrap es una gran herramienta que con la que podremos hacer atractivos sitios y no necesariamente todos se van a ver igual.

En el siguiente video conoceremos un método de trabajo muy importante para crear sitios web responsivos, que de hecho será la base de todo lo que haremos en el proyecto.

Conociendo Mobile First

Conoceremos uno de los conceptos más importantes a la hora de crear páginas web responsivas, o pensadas para ser adaptadas a los distintos tamaños de dispositivo.

De hecho, es tan reconocida que Bootstrap se basó en esta filosofía para crear todo sus componentes y características. Conocerla nos dará todas las bases para poder trabajar de manera óptima con Bootstrap.

¿Qué es mobile first?

Mobile first es una filosofía de desarrollo que apunta a diseñar prioritariamente la visualización y estructura del sitio en dispositivos móviles.

Luego de tener el diseño establecido para las pantallas reducidas, se irá escalando hacia pantallas de mayor tamaño.



Imagen 6: Mobile First

Desarrollar una página pensando en dispositivos móviles es una gran estrategia, debido al gran uso que han alcanzado estos dispositivos el último tiempo.

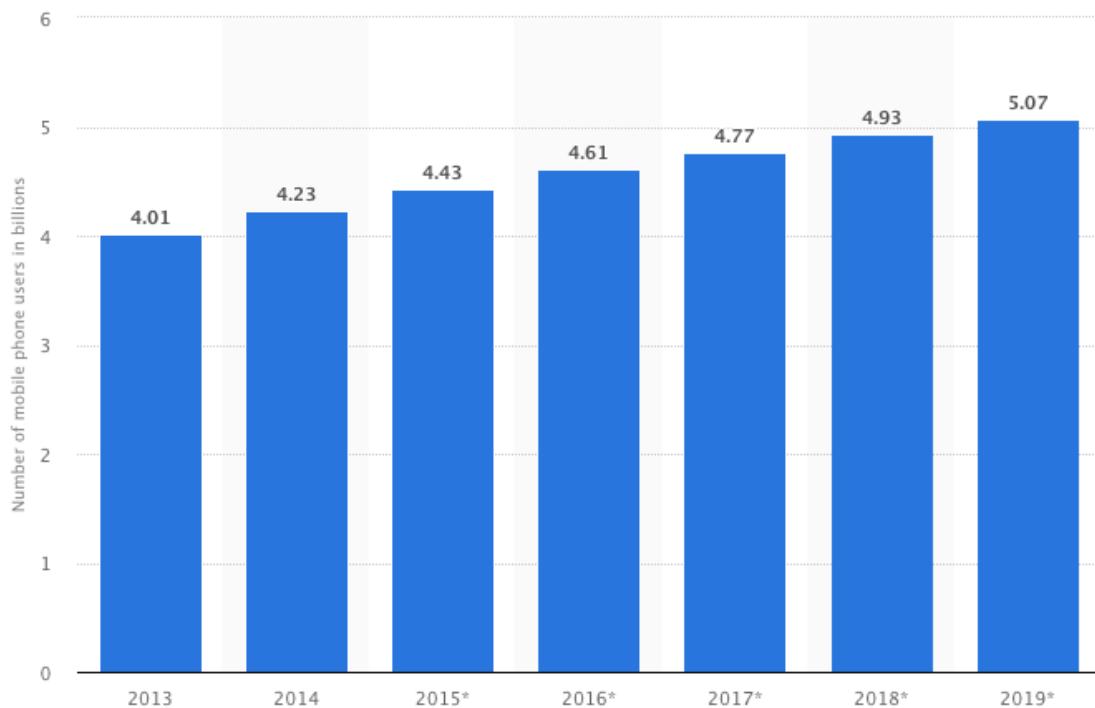


Imagen 7: [Number of mobile phone users worldwide from 2015 to 2020 \(in billions\)](#)

De hecho, se han vuelto tan esenciales para nuestra vida diaria que la cantidad de personas que tienen un dispositivo móvil es, a la fecha, de 4.93 billones, esto es casi el 63% de la población mundial.

Entender estos números es importante ya que las tecnologías están apuntando hacia esa dirección.

¿Cómo podemos comenzar a trabajar con mobile first?

Para comenzar a trabajar con mobile first en nuestro código debemos **entender cómo trabajar con diferentes resoluciones de pantalla**. Esto conlleva que debemos aprender a cómo normalizar los tamaños sobre los cuáles trabajaremos mientras desarrollamos la página.

Trabajar en un mundo donde las resoluciones no tienen estándar es todo un desafío.



Imagen 8: Trabajar con Mobile First

Afortunadamente, existen los frameworks CSS, como Bootstrap, que nos pueden ayudar a lidiar con esto. Dentro de su página existen media queries que contienen tamaños específicos para diferentes anchos de pantalla y ventanas de visualización.

```
// Extra small devices (portrait phones, less than 576px)
// No media query since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

Trabajar usando características de los diseños responsivos

Se debe trabajar con funcionalidades usadas en los diseños responsivos, esto incluye:

- Usar la etiqueta meta viewport: El meta viewport es una etiqueta HTML usada para darle al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página. Ésta se pone dentro de la etiqueta `<head>`.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

`<width=device-width>`: Establece el ancho de la página para seguir el ancho de pantalla del dispositivo (que variará dependiendo del tamaño del dispositivo y de el ancho de la ventana de visualización del navegador).

`<initial-scale=1.0>`: Establece el nivel de zoom inicial, cuando el navegador carga la página por primera vez.

Esta etiqueta nos puede ayudar a mejorar drásticamente la usabilidad de una página web, puesto que el contenido de la página se escalará dependiendo del tamaño.

En el siguiente ejemplo podemos ver una página no utilizando la etiqueta meta viewport y otra que sí.



Imagen 9: Ejemplo sin utilizar viewport



Etiqueta meta viewport:
viewport="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no";
Este punto debe ser el menor, concretamente dejándolo así, ya que nos permite que el contenido se adapte al tamaño de la pantalla del dispositivo móvil y no se corten imágenes o textos. De esta forma el contenido se verá bien en todos los dispositivos móviles.

Imagen 10: Ejemplo utilizando viewport

- Usar grillas: Las grillas son un componente de estructura que nos ayudará a estructurar y disponer los elementos de una página web de forma óptima, sencilla y rápida.

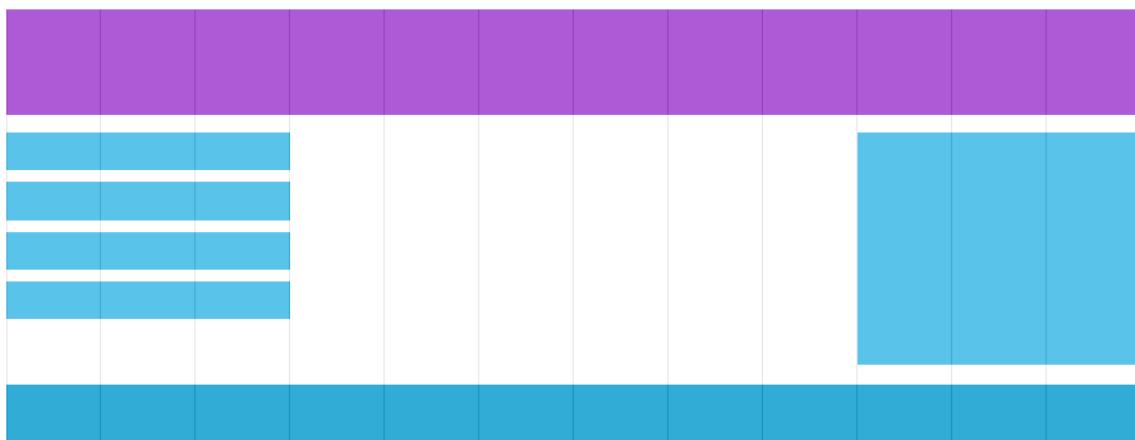


Imagen 11: Grilla

Para hacerlo tenemos dos opciones:

- Nosotros creamos la grilla usando CSS
- O usamos una grilla pre-fabricada

En el caso de este proyecto usaremos la grilla que incluye Bootstrap, debido a que es fácil de implementar en la diagramación del sitio web.

- Usar imágenes responsivas: El siguiente punto es simple, puesto que para crear imágenes responsivas sólo debemos usar unidades de medida relativas.

Esto quiere decir que estas medidas estarán basadas en tamaños relativos al tamaño del viewport o su contenedor. A través de las clases que nos entrega Bootstrap podremos darle esa propiedad.

Considerar la experiencia que tendrá el usuario con la página web

El siguiente paso es reconocer la experiencia que tendrá el usuario con nuestro sitio web en diversos dispositivos.

Reconocer el cómo interactúa el usuario con nuestro sitio es esencial, ya que al saberlo podremos desarrollar una mejor experiencia al usar características específicas de un dispositivo en concreto.

Por ejemplo: Podemos aprovechar los gestos de un teléfono para hacer scroll más rápido, o usar un atajo de teclado para abrir un link en nuestro computador. Tener en consideración esto hará de nuestras páginas sean más cómodas, demostrando que fueron hechas pensando en los usuarios.

Priorizar el contenido de la página web

El último punto importante para trabajar con mobile first, es el priorizar el contenido basado en la importancia que tiene dentro de la página web.

Esto quiere decir, que nosotros debemos ser capaces reconocer cuándo es mejor mostrar o no un contenido, dependiendo de la resolución de pantalla del dispositivo. Por lo general los dispositivos móviles, al tener menor tamaño, deben tener un contenido simplificado, que se enfoque en el objetivo principal del sitio web.

Integrando Bootstrap

Nuevo Proyecto

Comencemos a construir el proyecto.

Antes de comenzar, debemos descargar los archivos que se encuentra dentro de los **assets** que usaremos en la página web. Con eso listo, comenzemos a crear la estructura de directorios del proyecto.

Creando directorios y archivos del proyecto

1. Crear un nuevo directorio al cual llamaremos `olivia-ros`, podemos crearlo donde más nos acomode, en nuestro caso lo pondremos dentro del Escritorio.
2. Entrar a Atom.
3. Crear el `index.html`, en la raíz de la carpeta `olivia-ros`.
4. Agregar todos los directorios que falten en el proyecto. En este caso sería la carpeta `assets`, que contendrá a todos diferentes archivos de nuestro proyecto como las imágenes, los archivos CSS, entre otras cosas.
5. Crear la carpeta `assets` presionando "botón derecho" y luego nueva carpeta.
6. Crear dentro de la carpeta `assets` una carpeta llamada `img`, donde irán todas las imágenes del proyecto (descargadas desde `assets`).
7. Mover las imágenes a la carpeta `img`.
8. Crear la carpeta `css`.
9. Crear el archivo `style.css` presionando nuevo archivo.

Ya hemos terminado de crear todas las carpetas y archivos necesarios para comenzar con nuestro proyecto.

Creando la estructura HTML base del proyecto

Lo siguiente que haremos será agregar la estructura HTML de nuestro proyecto, para hacerlo agregaremos el doctype de la versión HTML5.

```
<!DOCTYPE html>
```

Luego irá la etiqueta HTML.

```
<html>  
</html>
```

Y dentro de la etiqueta HTML agregaremos el atributo de idioma llamado `lang`. Esta etiqueta ayudará al navegador a reconocer el idioma de nuestra página web.

```
<html lang="es">  
</html>
```

Lo siguiente como sabemos es la etiqueta `<head>`.

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  
</head>  
</html>
```

Dentro pondremos:

- El meta `charset="utf-8"` para no tener problemas con los caracteres especiales.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8">
</head>
</html>
```

- Debajo agregaremos el meta que vimos en el video anterior llamado meta viewport. Este meta, como sabemos, nos ayudará a escalar el tamaño del viewport dependiendo de espacio que dispongamos dentro la ventana del navegador.

La etiqueta meta viewport contiene en su interior un atributo `width=device-width` que establece el ancho de la página basado en el ancho del dispositivo, además de `initial-scale=1.0`, que establece el zoom inicial de la página web al ser cargada la primera vez por el navegador.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
</html>
```

- Lo último que nos falta es agregar el título de la página usando la etiqueta `<title>`. Escribamos como título `Olivia Ros | Food Blogger`

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Olivia Ros | Food Blogger</title>
</head>
</html>
```

Entremos a la carpeta donde se aloja `index.html`.

Integrando el CDN de Bootstrap

En este video integraremos todos los archivos externos que usaremos en el proyecto.

Integrando el CDN de Bootstrap (con sus scripts necesarios)

Lo primero que haremos será integrar Bootstrap 4 dentro de nuestra página web.

- Debemos ir a getbootstrap.com.
- Dentro de la página revisaremos si nos encontramos en la versión actual de Bootstrap.

Si miramos debajo del botón "get started" podremos ver cuál es la versión más reciente de este framework. También lo podemos hacer buscando en la barra de navegación.

Bootstrap ha tenido varios cambios de versiones recientemente, de modo que es importante conocer sus últimas novedades visitando el [blog](#).

- Volvamos al home del sitio de Bootstrap, clickemos "get started", ahí nos darán el paso a paso para incorporar Bootstrap.

Dentro de esta sección encontraremos dos subtítulos: el primero dice "CSS only" y es un snippet que contiene el código CSS de Bootstrap, mientras que el segundo contiene los archivos "jQuery, popper.js y Bootstrap JS", que analizaremos a continuación.

jQuery es necesario para el funcionamiento de muchos de los componentes de Bootstrap que están desarrollados con esta librería de JavaScript. Luego viene el archivo popper.js, que es un plugin de JavaScript muy necesario para algunos componentes incluidos en Bootstrap como los dropdown, tooltips y popovers. Y finalmente, el último archivo es el código JavaScript propio de Bootstrap.

- Copiemos la primera parte y peguémolas debajo del meta viewport.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Olivia Ros | Food Blogger</title>
  <!-- Bootstrap 4.1.3 CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
        integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
        crossorigin="anonymous">
</head>
```

Agreguemos arriba un comentario con la versión de Bootstrap. El hacer esto nos ayudará a reconocer de manera más rápida el código incluido en nuestro HTML. Para incluir un comentario debemos escribir:

```
<!--El comentario-->
```

El siguiente paso será copiar jQuery, Popper.js y el JavaScript de Bootstrap para pegarlo en index.html, pero esta vez lo pegaremos debajo, justo antes del cierre de la etiqueta `<body>`. La razón del por qué dejaremos estos archivos ahí es porque los navegadores leen los scripts de forma secuencial, bloqueando el renderizado de la página hasta leerlos completamente.

La solución a esto es poner estos archivos abajo de todos los scripts, puesto que el navegador cargará primero el HTML, luego el CSS y finalmente los archivos JavaScript. Suena muy técnico, pero aplicarlo nos ayudará a cargar nuestra página web de manera más rápida.

Sabiendo esto, es momento de pegar los scripts antes del cierre de la etiqueta `<body>`.

```
<body>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
  crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity="sha384-ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
  crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqxuZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin="anonymous"></script>
</body>
```

Se debe tener en cuenta es la etiqueta usada para los integrar archivos JavaScript.

Esta etiqueta se llama `<script>` y es la encargada de contener los scripts de JavaScript. Se puede usar para llamar un archivo externo o escribir directamente los script dentro de la etiqueta. Con esto interiorizado, sigamos mejorando nuestro código.

Agreguemos un comentario arriba de cada script mencionando qué contiene cada archivo.

- Sobre el primero escribiremos: `<!--jQuery 3.3.1-->`
- Sobre el segundo escribiremos: `<!--Popper JS 1.14.3-->`
- Sobre el último escribiremos: `<!--Bootstrap JS 4.1.3-->`

```
<!--jQuery 3.3.1-->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
  crossorigin="anonymous"></script>
<!--Popper JS 1.14.3-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity="sha384-ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
  crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqxuZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin="anonymous"></script>
```

Integrando fuentes e íconos para el proyecto

Bien, ya solo nos falta agregar los CDN de las fuentes e íconos que usaremos dentro del proyecto.

En la [maqueta del proyecto](#) podemos ver que se ocupa la tipografía:

- Raleway Regular (400)
- Raleway Bold (700)

Así que iremos a [Google Fonts](#) y haremos los pasos ya conocidos para vincular la tipografía con nuestro proyecto.

Buscaremos y seleccionaremos la tipografía "Raleway". Con la fuente seleccionada, debemos customizar los estilos que necesitamos descargar. En customiz marcamos la opción de los pesos que requerimos: Regular y Bold

En embed copiaremos la URL que necesitamos añadir a nuestro archivo HTML:

```
<link href="https://fonts.googleapis.com/css?family=Raleway:400,700" rel="stylesheet">
```

También iremos a [Font Awesome](#) a copiar el código para llamarlo en nuestro archivo y así tener a disposición los íconos de Font Awesome.

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHrtZL1+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8IWUE00s/" crossorigin="anonymous">
```

Lo último que haremos antes de probar si funciona Bootstrap es agregar el archivo con nuestros estilos.

Agreguemos la etiqueta `<link>` y luego agreguemos la ruta de nuestro archivo CSS. La ruta es la siguiente:

```
<link rel="stylesheet" href="assets/css/style.css">
```

Con esto hemos terminado con la integración de todo lo que necesitaremos en este proyecto, ahora vamos a probar si está todo OK para comenzar.

Resultado

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Olivia Ros | Food Blogger</title>
    <!-- Bootstrap 4.1.3 CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
      <!--Google Font: Raleway (Regular, Bold)-->
      <link href="https://fonts.googleapis.com/css?family=Raleway:400,700" rel="stylesheet">
      <!--Font Awesome 5.6.3-->
      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8IWUE00s/" crossorigin="anonymous">
        <!--Mi estilo-->
        <link rel="stylesheet" href="assets/css/style.css">
    </head>
    <body>
      <!--jQuery 3.3.1-->
      <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abTE1Pi6jizo" crossorigin="anonymous"></script>
      <!--Popper JS 1.14.3-->
      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>
      <!--Bootstrap JS 4.1.3-->
      <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqxuZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>
    </body>
</html>
```

Probando la integración de Bootstrap

Es muy necesario que entendamos qué hicimos anteriormente.

CSS de Bootstrap

Al poner el código:

```
<!-- Bootstrap 4.1.3 CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
```

Lo que hicimos fue llamar, desde CDN, el código CSS de Bootstrap minificado, o sea, condensado para reducir su tamaño. Esto se logra eliminando los caracteres superfluos, sin que deje de funcionar.

Este código minificado tiene sólo 7 líneas de código, ya que al estar minificado evita caracteres como espacios adicionales, saltos de línea, sangrías y comentarios. Con el objetivo de que pese menos.

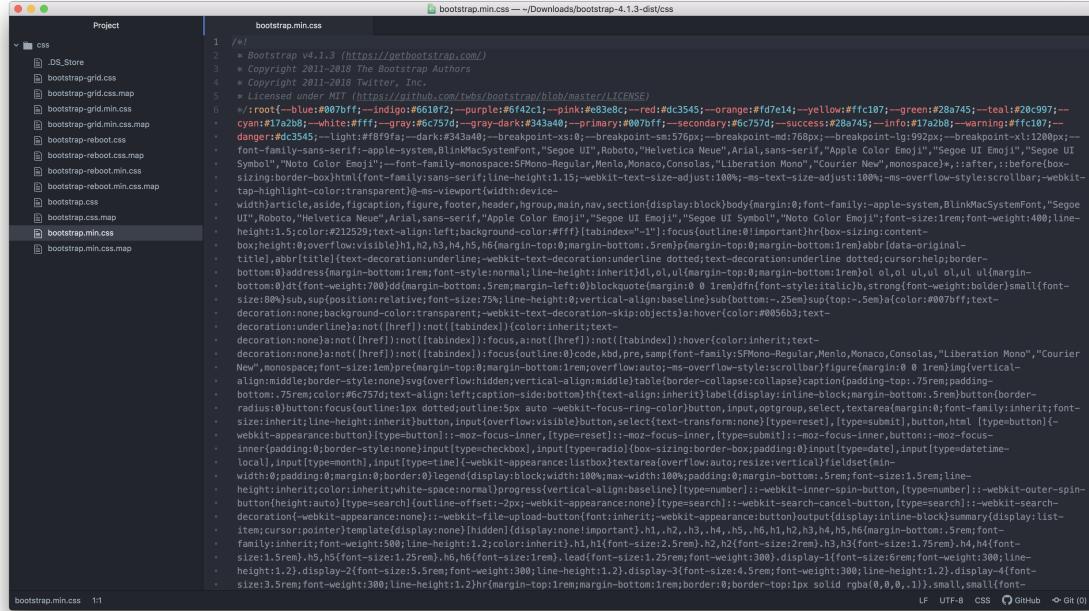
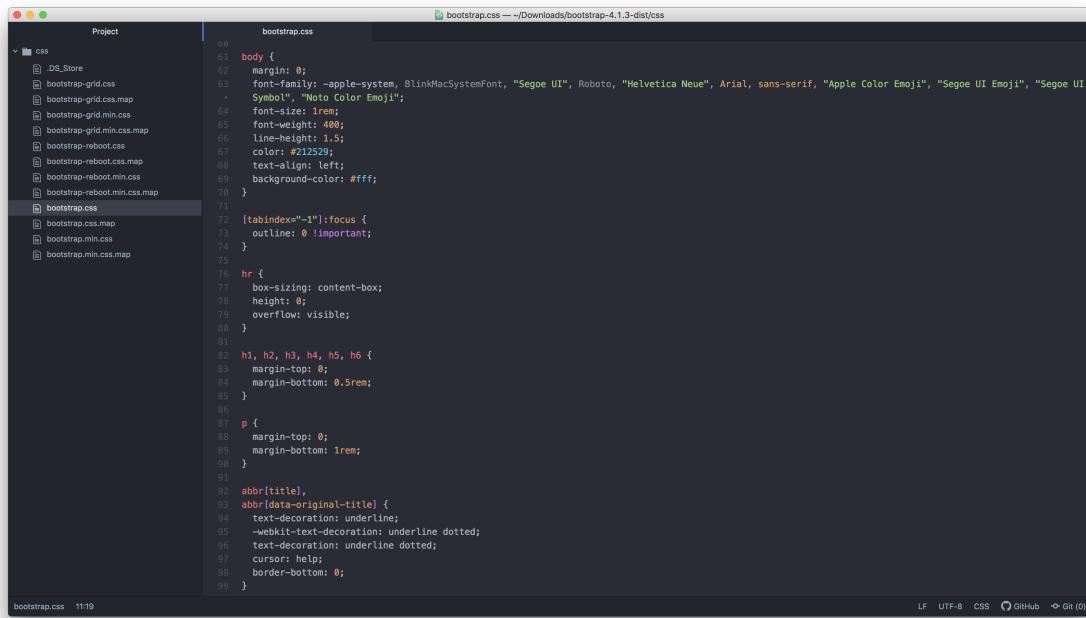


Imagen 12: CSS Bootstrap

Pero si este código no estuviese minificado sería muchísimo más largo, en específico tendría 9030 líneas de código.



The screenshot shows a code editor window with the file 'bootstrap.css' open. The left sidebar shows a project structure with files like 'DB_Store', 'bootstrap.css', 'bootstrap-grid.css', etc. The main pane displays the CSS code for Bootstrap 4.1.3. The code includes styles for the body, focus states, horizontal rules (hr), and various heading and paragraph elements. It also includes specific styles for abbr[title] and abbr[data-original-title] elements. The code is well-organized with comments and specific vendor prefixes for certain properties. The status bar at the bottom indicates the file is 1119 lines long and is saved in UTF-8 format.

```
body { margin: 0; font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol"; font-size: 1rem; font-weight: 400; line-height: 1.5; color: #212529; text-align: left; background-color: #fff; } [tabindex="-1"]:focus { outline: 0 !important; } hr { box-sizing: content-box; height: 0; overflow: visible; } h1, h2, h3, h4, h5, h6 { margin-top: 0; margin-bottom: 0.5rem; } p { margin-top: 0; margin-bottom: 1rem; } abbr[title], abbr[data-original-title] { text-decoration: underline; -webkit-text-decoration: underline dotted; text-decoration: underline dotted; cursor: help; border-bottom: 0; }
```

Imagen 13: CSS Bootstrap 2

Como pueden ver, el CSS que hemos integrado de Bootstrap es nada más ni nada menos que una gran hoja de estilo. Lo que hace Bootstrap es entregarnos propiedades definidas para las distintas etiquetas HTML, además de propiedades definidas para diversas clases creadas que se pueden utilizar en distintos elementos desde el HTML.

Desde ahora en adelante debemos plantearnos la idea de siempre leer la documentación de lo que estemos utilizando. En el caso de Bootstrap también debemos leer la [documentación](#) para sacarle el mayor partido.

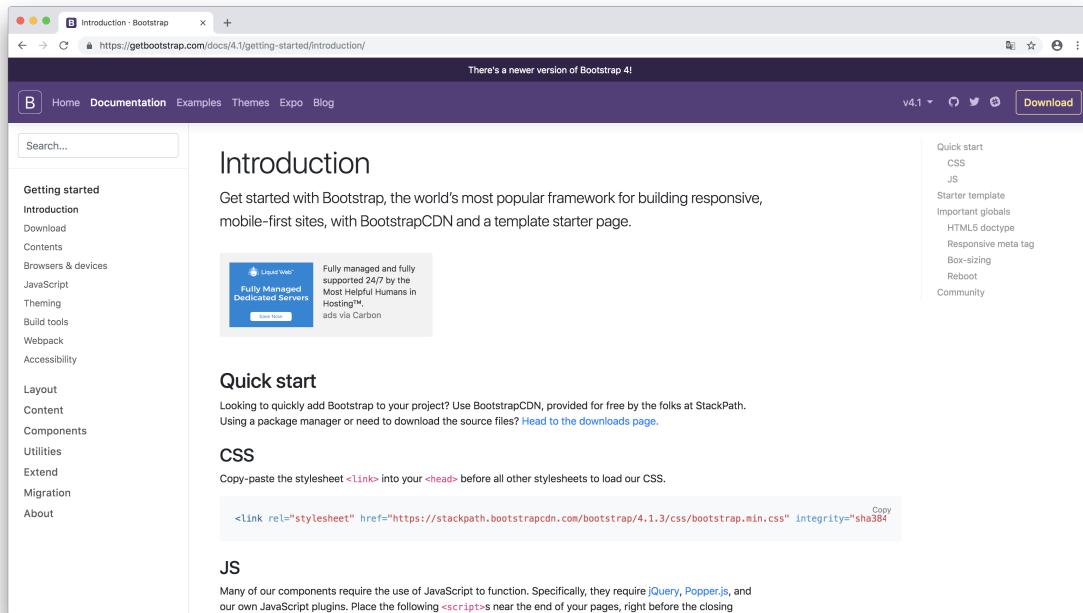


Imagen 14: Documentación Bootstrap

No vaya a ser que reescribamos algo que ya nos entrega Bootstrap.

Scripts para Bootstrap

Lo segundo que escribimos -al final de la etiqueta `<body>` - fue colocar los scripts necesarios para que Bootstrap pueda funcionar:

```
<!--jQuery 3.3.1-->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<!--Popper JS 1.14.3-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqxuZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

- jQuery es una librería de JavaScript que permite agregar interactividad y efectos visuales en un sitio web.
- Popper JS: es una librería de JavaScript que permite añadir tooltips y popovers en elementos HTML.
- Boostrap JS: es una serie de componentes de JavaScript para Bootstrap. Necesita de jQuery y Popper JS para funcionar.

Éstos scripts, según la propia [documentación de Bootstrap](#) deben ir antes del cierre de la etiqueta `<body>` y en ese orden, ya que va leyéndolos desde arriba hacia abajo.

Google Fonts y Font Awesome

El llamado a Google Fonts y Font Awesome ya lo conocemos, respectivamente nos dan la tipografía y los íconos que vamos a usar.

Añadiendo Navbar

Para probar si funcionó la integración de Bootstrap, debemos copiar algún componente de Bootstrap y ver si se están respetando los estilos que le entrega a las distintas clases del contenido en sí, es decir, un contenido que en HTML no debería tener formato, con Bootstrap se le atribuye.

Un componente interesante que podemos copiar será un Navbar, ya que tenemos uno en la [maqueta del proyecto](#).

Entonces dentro de [GetBootstrap.com](#) vamos a ir a [Documentation](#) y luego a [Components](#).

Ahora, en el menú que apareció busquemos la opción Navbar.

Allí dentro, copiemos todo el snippet de código que se encuentra bajo el encabezado **Nav**.

The screenshot shows a browser window with the URL <https://getbootstrap.com/docs/4.1/components/navbar/>. The page has a dark purple header with a logo and navigation links for Home, Documentation, Examples, Themes, Expo, and Blog. On the left, there's a sidebar with categories like Getting started, Components (which is expanded), and various UI components like Alerts, Buttons, and Cards. The main content area is titled 'Nav' and contains a snippet of HTML code for a responsive navbar. The code includes classes like 'navbar navbar-expand-lg navbar-light bg-light', 'navbar-brand', 'navbar-toggler', and 'nav-item active'. A 'Copy' button is visible next to the code. To the right, there's a sidebar with links to 'How it works', 'Supported content', and other Bootstrap sections like Brand, Nav, Forms, etc.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
  </div>
</nav>
```

Imagen 15: Navbar

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
  </div>
</nav>
```

Cuando este listo, lo pegaremos dentro de la etiqueta `<body>`, justo cuando empieza.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Olivia Ros | Food Blogger</title>
    <!-- Bootstrap 4.1.3 CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
    <!--Google Font: Raleway (Regular, Bold)-->
    <link href="https://fonts.googleapis.com/css?family=Raleway:400,700" rel="stylesheet">
    <!--Font Awesome 5.6.3-->
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-9eUQ5HjZM200ZIaDjBkWZlqWZnN+Lz3P0JZdPZJ/2Dc3Z9zRJjEJ0JyWUE00s/" crossorigin="anonymous">
    <!--Mi estilo-->
    <link rel="stylesheet" href="assets/css/style.css">
  </head>
  <body>
    <!--Navbar agregado-->
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="#">Navbar</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item active">
            <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Features</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Pricing</a>
          </li>
          <li class="nav-item">
            <a class="nav-link disabled" href="#">Disabled</a>
          </li>
        </ul>
      </div>
    </nav>
    <!--jQuery 3.3.1-->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abTE1Pi6jizo" crossorigin="anonymous"></script>
    <!--Popper JS 1.14.3-->
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-9eUQ5HjZM200ZIaDjBkWZlqWZnN+Lz3P0JZdPZJ/2Dc3Z9zRJjEJ0JyWUE00s/" crossorigin="anonymous"></script>
    <!--Bootstrap JS 4.1.3-->
```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"  
crossorigin="anonymous"></script>  
</body>  
</html>
```

Recordemos indentar de forma que podamos identificar cuál etiqueta está dentro de cuál.

Ahora guardamos y luego vamos a abrir `index.html` dentro de nuestro navegador.

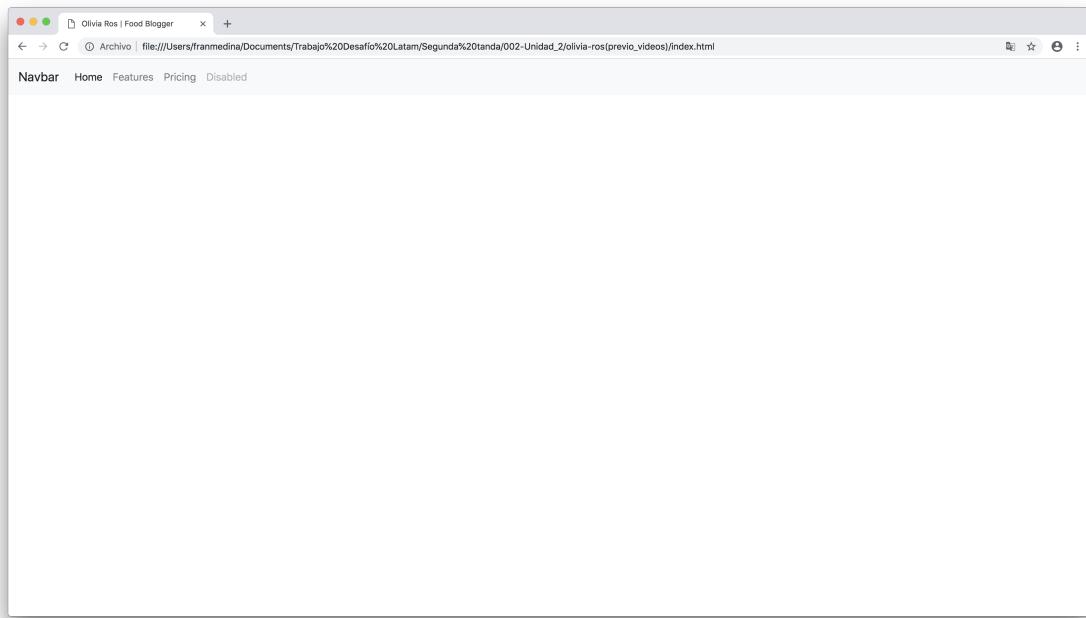


Imagen 16: Index

Podemos ver que hemos escrito un navbar con estilo, colores, tipografía y tamaños -todo gracias a Bootstrap- entonces podemos comprobar que el CSS de Bootstrap está bien incorporado.

Ahora, para comprobar que los scripts están bien incorporados, achiquemos el tamaño de resolución del navegador. Podemos ver que el menú se transformó en el llamado "menú hamburguesa".

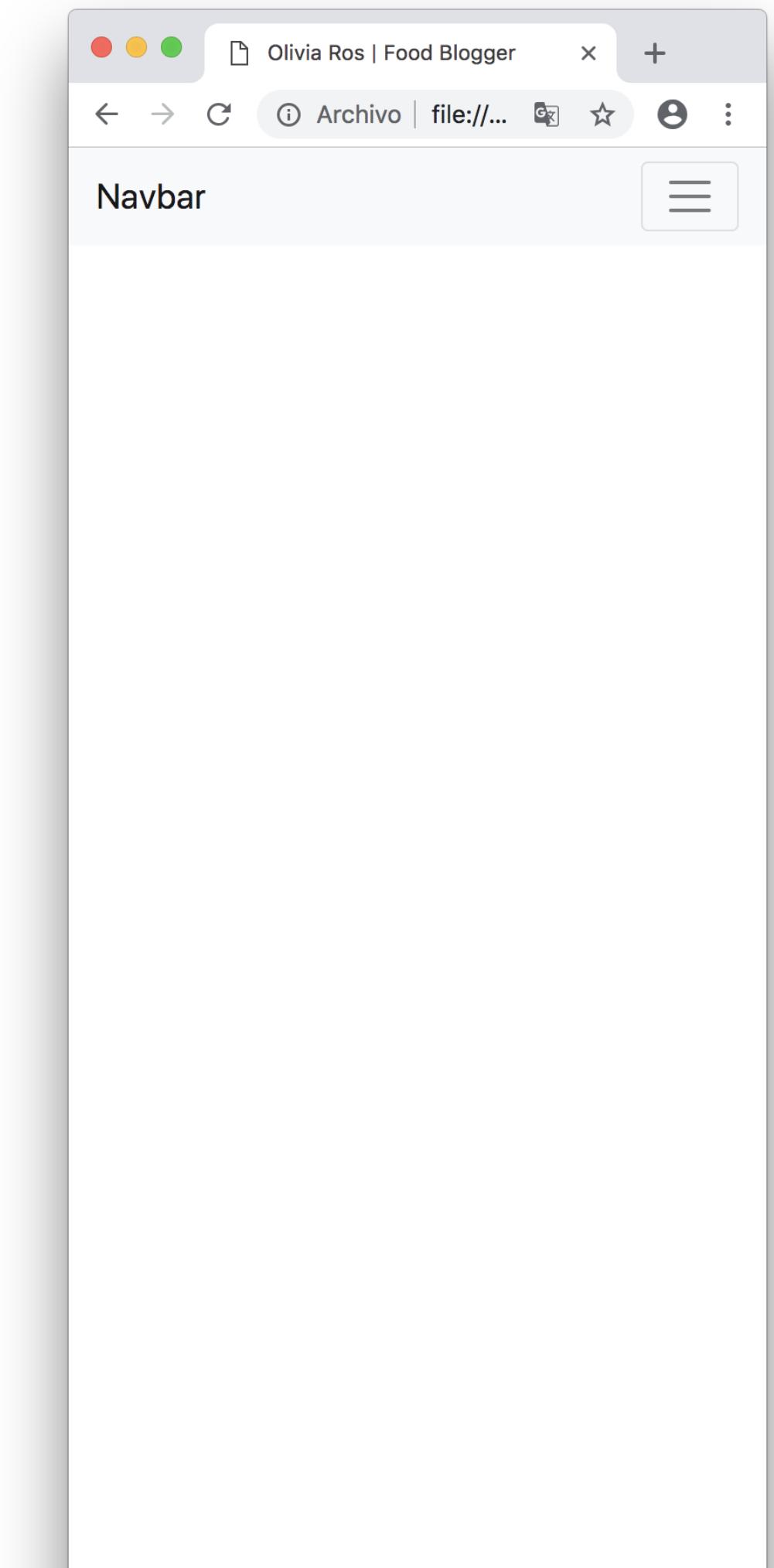




Imagen 17: Menú hamburguesa

Probemos presionando la hamburguesa del menú para ver si se despliega el menú.

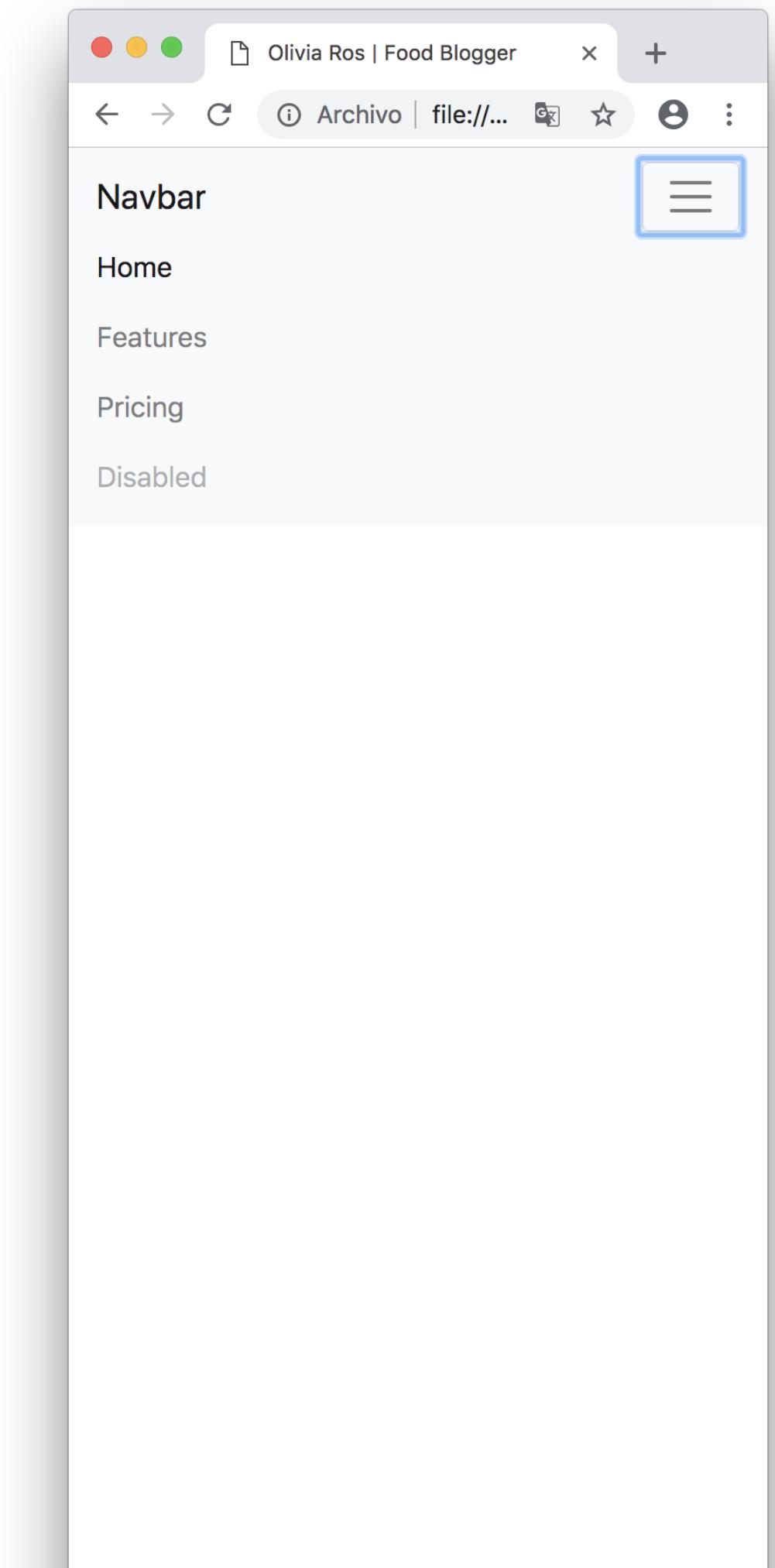




Imagen 18: Desplegar menú

Efectivamente, los ítems del menú se desplegaron. Eso sólo se puede lograr con JavaScript, por lo tanto podemos asumir que también está bien incorporado.

Con esto hemos comprobado que hemos integrado todo lo que necesitaremos en este proyecto, preliminarmente.