



Emprendimiento digital con tecnologías Web

Módulo 5: Introducción al desarrollo web para emprendimiento digital

Aprendizaje Esperado

4. Construir una página web utilizando componentes de la biblioteca Bootstrap 4 para la implementación de una interfaz web responsiva.

Bootstrap

Qué son y para qué sirven los frameworks CSS.

Un Framework es un conjunto premeditado de conceptos, módulos y criterios estándares que ayudan en la tarea de desarrollar sitios y aplicaciones web de manera más fácil. Proporciona funcionalidad genérica con módulos ya escritos y componentes personalizados creados de manera estándar. En resumen, un framework es un entorno reutilizable de software que permite a los diseñadores y desarrolladores web acceder fácilmente a la construcción de sus proyectos y soluciones con una codificación mínima y sin preocupándose por los detalles de bajo nivel. Esto reduce el tiempo de desarrollo y proporciona mantenimiento y adaptaciones con mayor facilidad.

Usualmente, podemos encontrar dos tipos de frameworks:

- Frameworks de Front-end (CSS and JavaScript frameworks)
- Frameworks de programación de Back-end or Lado-Servidor

Mientras que los desarrolladores y programadores web usan los marcos de back-end para crear aplicaciones en el lado del servidor, los diseñadores y desarrolladores de web usan los frameworks de front-end para implementar el lenguaje de hojas de estilo en cascada CSS.

Los frameworks permiten usar una estructura estandarizada común, que reduce mucho la escritura de código desde cero y ayuda a reutilizar componentes, módulos y bibliotecas. Esto permite concentrarse en las tareas principales de alto nivel.

COMPONENTES DE UN FRAMEWORK CSS

Los componentes de un Framework CSS son los siguientes:

- Grillas (Grids) (Estructuras que ayudan a organizar el contenido y a diseñar el layout)
- Elementos Tipográficos
- Compatibilidad Cross-browser
- Clases de apoyo para posicionar elementos
- Clases de utilidad
- Elementos de navegación
- Código fuente escrito en preprocesadores, tales como Sass y LESS
- Elementos de medios (Insignias o badges, descripciones emergentes o tooltips, comentarios, y otros)

VENTAJAS DE UN FRAMEWORK CSS

Algunos desarrolladores prefieren no utilizar frameworks CSS, principalmente por aludiendo a los innecesarios tamaños de los proyectos al incluirlos, un intervención invasiva de los códigos HTML, y la inevitable estética similar con otros sitios que utilizan el mismo framework. Sin embargo, el uso de un framework CSS provee múltiples beneficios que hacen atractiva su adopción, tales como^[1]:

- Código limpio y consistente.
- Compatibilidad entre navegadores.
- Diseño basado en cuadrícula (Grid-based).
- Fácil construcción de prototipos.
- Fácil mantenimiento.
- Reutilización de código.
- Fácil expansión y modificación.
- Documentación sólida
- Accesibilidad

El framework CSS ofrece la posibilidad de evitar el uso de HTML, CSS y Javascript puro para el desarrollo de sistemas basados en tecnologías Web. Además, los diseños basados en cuadrícula ayudan a posicionar, estructurar y diseñar un layout de sitio web de manera relativamente fácil. La adopción de un framework provee también la aborción por parte del desarrollador de la consistencia de un sistema ya maduro en términos de código y prácticas. Además, es fácil incorporar nuevos colaboradores en el

proyecto, dado que tendrán una base común que es el dominio del framework.

Qué es Bootstrap y cuáles son sus ventajas.

Bootstrap es un framework creado en 2011, por Mark Otto y Jacob Thornton del equipo de desarrollo de Twitter, para realizar interfaces web responsivas en base a CSS3 y Javascript, con el objetivo de facilitar la adaptación automática a cualquier tamaño y resolución de pantalla. Anteriormente a Bootstrap se utilizaban conjuntos de librerías que resultaban inconsistentes y cuyo mantenimiento costaba mucho trabajo a los desarrolladores. En 2012, Bootstrap se convirtió en el proyecto más popular de desarrollo web.

Bootstrap utiliza hojas de estilo CSS3, combinadas con el lenguaje Javascript y además es compatible con la mayoría de los navegadores web. Es un software libre, por lo que los usuarios tienen la libertad de usarlo, mejorarlo y distribuirlo libremente. Bootstrap es el framework de filosofía Mobile First más popular en diseños web. Es utilizado extensivamente por desarrolladores del mundo. Permite construir fácilmente sitios web responsivos e incorpora componentes, módulos, funciones Javascript, y Media Queries que ayudan al desarrollador a construir sitios web de alto impacto.

Bootstrap está disponible como proyecto en GitHub y puede ser descargado fácilmente en getbootstrap.com

VENTAJAS DE LA LIBRERÍA BOOTSTRAP

1. Cuadrícula (Grid) Responsiva.

Bootstrap incluye su propio sistema de grilla predefinido. De esta forma sólo es necesario comenzar a incorporar contenido en contenedores.

La definición de puntos de quiebre responsivos para cada columna es muy sencilla utilizando con granularidades extra pequeñas, pequeñas, medianas, grandes y extragrandes.

2. Imágenes Responsivas.

Bootstrap viene con su propio código para cambiar automáticamente el tamaño de las imágenes en función del tamaño

de pantalla actual. Simplemente agregando la clase `.img-responsive` a nuestras imágenes, las reglas CSS predefinidas se encargarán de las adaptaciones responsivas.

Bootstrap contiene funcionalidades que permiten cambiar la forma de imágenes con la adición de clases como `img-circle` y `img-rounded`, con la consiguiente reducción en la necesidad de llevar todos esos requerimientos a softwares de edición de imágenes.

3. Componentes.

Bootstrap incluye una gran cantidad de componentes que pueden ser agregados fácilmente a nuestra página web, tales como:

Diálogos modales, Barras de progreso, Barras de navegación, Información emergente, Elementos emergentes, Carrusel, Alertas, Menús desplegables, Grupos de entrada, Pestañas, Paginación, Tarjetas, Formularios, Grupos de listas, Jumbotron, y otros.

Todos estos componentes están prediseñados responsivamente para adaptarse a cualquier condición de pantalla de dispositivo.

4. Javascript.

Muchos de los componentes de Bootstrap están potenciados en cuanto a su comportamiento dinámico a través de JQuery, Popper.js y muchos plugins javascript propios de Bootstrap. Es posible utilizar Bootstrap sólo basado en CSS, sin embargo, esto es posible sólo cuando no se estén utilizando los componentes que dependen de las librerías js mencionadas.

5. Documentación.

Bootstrap posee una documentación muy completa y clara. Cada pieza de código se describe y explica en detalle explícito en su sitio web. Las explicaciones también incluyen ejemplos de código para la implementación básica, lo que simplifica el proceso incluso para principiantes.

6. Personalización.

Una de las principales críticas cuando se trata de frameworks como Bootstrap es su tamaño: el peso que arrojan realmente puede ralentizar su aplicación en la primera carga. La versión 4.5.0 del archivo CSS de Bootstrap, por ejemplo, es de 160 KB. Si bien esto puede no parecer especialmente grande en comparación con los archivos de imagen y video, para un archivo CSS, eso es enorme. Sin embargo, por su modularidad y por ser código abierto, Bootstrap permite incluso personalizar editando códigos, las funcionalidades se desean incluir el código a utilizar en nuestro desarrollo.

7. Comunidad.

Al igual que con muchos proyectos de código abierto, Bootstrap tiene una gran comunidad de diseñadores y desarrolladores detrás. Estar alojado en GitHub facilita a los desarrolladores modificar y contribuir a la base de código de Bootstrap. También facilita que las personas colaboren, presten sus consejos e interactúen con colegas y usuarios.

Bootstrap tiene una página de Twitter activa, un blog de Bootstrap e incluso una sala dedicada de Slack. Por último, es importante también considerar todos los desarrolladores dispuestos a ayudar con problemas técnicos en Stack Overflow, donde todas las preguntas se pueden encontrar bajo la etiqueta bootstrap-4.

8. Plantillas.

A medida que creció la popularidad de Bootstrap, los desarrolladores comenzaron a crear plantillas basadas en éste para acelerar aún más el proceso de desarrollo web. Existen muchos sitios web dedicados a compartir y comprar plantillas personalizadas basadas en Bootstrap. Sólo basta con buscar "Bootstrap Templates" o "Bootstrap Themes" en nuestro buscador preferido. Bootstrap mismo ofrece una colección de temas premium, tales como:

- Tema Dashboard: Con componentes premium para sitios de administración técnica y analítica de datos.
- Tema Marketing: Para sitios de E-commerce y Marketing en general.
- Tema Aplicación: Con componentes especiales para aplicaciones de software a medida.

Implementación de Bootstrap en una página web.

Bootstrap es un software de código abierto, publicado bajo licencia MIT, por lo cual el código fuente completo está disponible de manera libre. Sabemos que Bootstrap está compuesto por CSS y Javascript.

En esta sección explicaremos cómo incluir Bootstrap en nuestros desarrollos, de dos formas: Usándolo desde un servidor de una red de entrega de contenidos (CDN) y incluyendo los archivos directamente en nuestra estructura de archivos de proyecto.

Usaremos en este documento la versión 4.5.0 de Bootstrap. Existen muchos ejemplos en la web que están basados en la versión 3.x de Bootstrap. En general, se recomienda tener conocimiento de Bootstrap3

para ser capaces de mantener sitios existentes, de los cuales muchos están diseñados en esa versión. Si nos orientamos a comenzar nuevos proyectos la recomendación es utilizar Bootstrap4.

USAR BOOTSTRAP DESDE CDN (Content Delivery Network)

La forma más rápida de agregar Bootstrap a nuestro proyecto es a través del uso de CDN, Red de Entrega de Contenidos en español, concepto que corresponde a una red de servidores distribuida en internet que distribuye contenido estático particular (Archivos, imágenes, aplicaciones web, entre otros), permitiendo una actualización permanente y una reducción del tiempo de respuestas entre el cliente y el servidor.

Este modo implica que nuestro código HTML posee referencias a archivos de hojas de estilo CSS y código Javascript que no se cargan desde el mismo servidor donde reside nuestra aplicación. Esto provee la ventaja de que podemos descargar nuestro servidor para servir sólo el contenido propio de nuestra aplicación, confiando en la alta disponibilidad de los servidores CDN que generalmente son muy confiables. El uso de CDN también puede implicar mejoras en el desempeño del navegador web al cargar nuestro sitio, pues es muy probable que los recursos alojados en CDN estén en caché o disponibles con mayor rapidez por cercanía virtual en la red.

Para incluir el framework en un nuestro desarrollo web, debemos hacer intervenciones en nuestro código HTML, incorporando el siguiente código HTML^[2]:

Se agrega el siguiente elemento `<link>` dentro de la sección `<head>` de nuestro documento HTML. Con ésto incluimos la hoja de estilos CSS estándar de Bootstrap, en este caso versión 4.5.0 como mencionamos más arriba:

```
<link
  rel="stylesheet"

  href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bo
otstrap.min.css"

  integrity="sha384-9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAO
H8WgZI5MYYxMvc+NcPb1dKGj7Sk"
  crossorigin="anonymous"
>
```

Por otro lado, incluimos los siguientes elementos `<script>` con códigos Javascript al final de la sección `<body>`. Bootstrap depende de las librerías JQuery y Popper.js y debemos cargar los archivos en el orden indicado:

```
<script
  src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

  integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MV
  BnE+lbbVYUew+OrCXaRkfj"
  crossorigin="anonymous">
</script>

<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.mi
n.js"

integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEwIdlvI9IOYy5n3zV9zzT
tml3UksdQVRVvoxMfooAo"
  crossorigin="anonymous">
</script>

<script

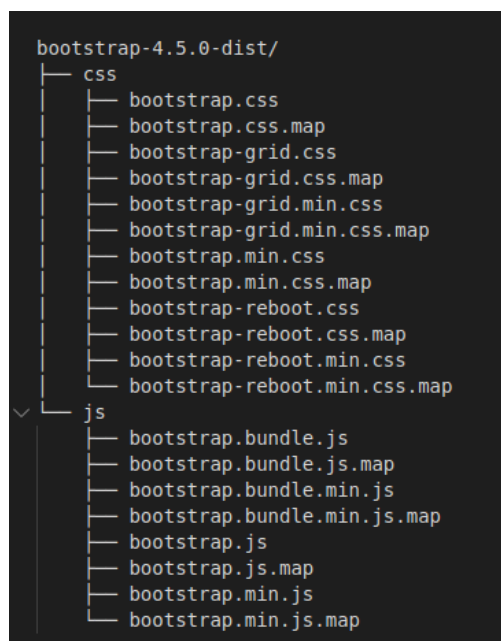
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.
min.js"

integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYj
ZE3lpu6Tp75j7Bh/kR0JKI"
  crossorigin="anonymous">
</script>
```

Es importante tener en cuenta la compatibilidad entre las versiones de las distintas librerías y los órdenes de carga de los archivos. Las versiones y disposición utilizada en este caso corresponde a la sugerida por la página <https://getbootstrap.com/> para Bootstrap en su versión 4.5.0. Es probable encontrar incompatibilidades y errores de funcionamiento del framework si combinamos versiones no recomendadas.

DESCARGANDO ARCHIVOS BOOTSTRAP

Para descargar e instalar Bootstrap, debemos acceder a su página web oficial, <https://getbootstrap.com/>, y procedemos a descargar el framework con el botón "download". Éste nos entrega la opción de bajar el código fuente completo del proyecto Bootstrap para desarrolladores que contribuyen al proyecto, o solamente una versión precompilada que incluye, entre otros, los archivos compilados listos para usar en producción. Escogeremos esta última opción y descargaremos el archivo **bootstrap-4.5.0-dist.zip**. En general, Bootstrap mantiene la convención de denominar los archivos como **bootstrap-x.y.z-dist.zip**, donde podemos reemplazar x.y.z con los números correspondientes a la versión que estamos utilizando. La palabra **dist** en el nombre del archivo que alude a "distribución", que quiere decir que son los archivos compilados, listos para usar en producción. De esta forma obtenemos los siguientes archivos:



De estos archivos los importantes, para nuestra misión de poner en funcionamiento Bootstrap, son las versiones **bootstrap.min.css** y **bootstrap.min.js**

Además, bajamos desde los sitios de Popper y JQuery los archivos Javascript de las mismas versiones utilizadas en el caso de CDN.

Ponemos todo esto en una estructura básica de nuestro proyecto incluyendo lo siguiente:

```
bootstrap_example_local/  
├── bootstrap-4.5.0-dist  
│   ├── css  
│   │   ├── bootstrap.css  
│   │   └── bootstrap.min.css  
│   └── js  
│       ├── bootstrap.js  
│       └── bootstrap.min.js  
├── bootstrap_example_local.html  
├── jquery-3.5.1  
│   └── jquery-3.5.1.slim.min.js  
├── popper-1.16.0  
│   └── popper.min.js
```

De esta forma quedamos con nuestro proyecto inicial de la siguiente manera:

```
<!doctype html>  
<html lang="es">  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1,  
shrink-to-fit=no">  
    <!-- Bootstrap CSS -->  
    <link rel="stylesheet"  
href="bootstrap-4.5.0-dist/css/bootstrap.min.css">  
    <title>Ejemplo Bootstrap CDN</title>  
  </head>  
  <body>  
    <div class="card">  
      <h1>Ejemplo Bootstrap CDN</h1>  
    </div>  
    <!-- jQuery, Popper.js, Bootstrap JS -->  
    <script src="jquery-3.5.1/jquery-3.5.1.slim.min.js"></script>  
    <script src="popper-1.16.0/popper.min.js"></script>  
    <script src="bootstrap-4.5.0-dist/js/bootstrap.min.js"></script>  
  </body>  
</html>
```

Creando páginas responsive con bootstrap.

En la sección anterior, dimos un punta pie inicial a un documento HTML donde cargamos de dos formas alternativas el framework Bootstrap para quedar en condiciones de comenzar a desarrollar nuestro ejemplo responsivo que nos llevará a través de las funcionalidades principales de Bootstrap.

Como se ha señalado previamente, Bootstrap 4 se basa en el paradigma **Mobile First**. Como tal, no debería sorprendernos que el sistema de cuadrícula esté optimizado para **viewports** pequeños y se amplíe para adaptarse a viewports de mayores tamaños. Para lograr lo mencionado, Bootstrap consta de un **Sistema de Cuadrícula o Grilla**, que es una de sus piedras angulares y revisaremos a continuación.

SISTEMA DE CUADRÍCULA O GRILLA

El Sistema de Cuadrícula o Grilla (Utilizaremos la denominación de Grilla de aquí en adelante) de Bootstrap es posiblemente su característica más impresionante y más utilizada, ya que resuelve la importante tarea de posicionamiento horizontal y vertical del contenido de una página, permitiendo que el contenido se estructura en varios anchos de visualización. Por lo tanto, dominar esta funcionalidad central es esencial para cualquier desarrollador de Bootstrap.

La grilla que compone ésta sistema es una estructura que consiste en tres partes distintas, pero fundamentalmente relacionadas. La unidad mayor se denomina contenedor (**container**) y se divide en filas (**row**) que a su vez están divididas en 12 **columnas (column)** de igual dimensión.

CONTAINERS

Los contenedores son el núcleo del sistema de grilla de Bootstrap. Un contenedor es exactamente lo que insinúa su nombre: Un elemento para contener el resto del contenido dentro de una sección de una página. Podemos hacernos la idea de un contenedor como un área en la ventana del navegador donde podemos emplazar nuestro contenido. Todo el contenido se mostrará en esa área, que a su vez puede transformarse en función de su contexto. A menos que se especifique explícitamente, el contenido nunca se sobrepasará los límites de esta área, independientemente de las dimensiones de la ventana de navegación. Un contenedor puede aplicarse a todo el contenido de una página, caso en el cual correspondería a un elemento contenedor raíz; o a diferentes

secciones de una página, que en este caso tendría múltiples contenedores para diversas secciones de la página web.

Bootstrap posee dos tipos de clases para contenedores: **container** y **container-fluid**.

container

Uso típico: `<div class="container">...</div>`

La clase container despliega sus contenidos en un ancho fijo determinado. Típicamente este ancho está determinado por el ancho de la ventana de navegación, y toma ventaja de Media Queries (Que ya hemos visto anteriormente en este documento), para definir cuál es el ancho más adecuado. Bootstrap define 5 reglas de Media Query con sus correspondientes breakpoints:

- Extra small (xs): Normalmente se refiere a las pantallas extra pequeñas que se encuentran en los dispositivos móviles (por ejemplo, pantallas verticales de teléfonos móviles).
- Small (sm): Se refiere a la pantalla promedio de su dispositivo móvil (por ejemplo, pantallas horizontales de teléfonos móviles)
- Medium (md): Se refiere a tamaños de pantalla medianos, como los que se encuentran en tablets.
- Large (lg): Se refiere a pantallas grandes, como las que se encuentran en dispositivos portátiles y de escritorio
- Extra large (xl): Se refiere a pantallas extragrandes, como las de los monitores de escritorio de pantalla ancha

Estos breakpoints gatillarán los cambios de ancho de container, el cual quedará en un ancho fijo durante todo el rango entre dos breakpoints. Los valores en pixeles para cada breakpoint y la propiedad max-width de container en cada tramo se muestran a continuación:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
max-width	100%	540px	720px	960px	1140px

container-fluid

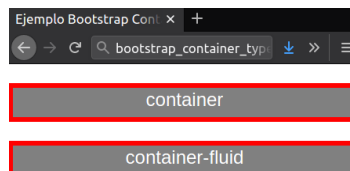
Uso típico: `<div class="container-fluid">...</div>`

La clase container-fluid no posee los puntos de quiebre para su comportamiento como la clase container. En el caso de container-fluid su

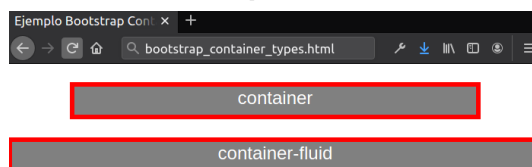
ancho se ajusta continuamente a un ancho de pantalla de navegación cambiante, siempre cubriendo un 100% del ancho de ésta.

A continuación mostramos una secuencia ilustrativa, para los distintos breakpoints, del comportamiento de dos divs; uno configurado con la clase container y otro con la clase container-fluid:

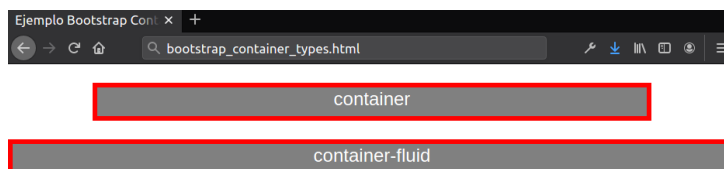
breakpoint: xs



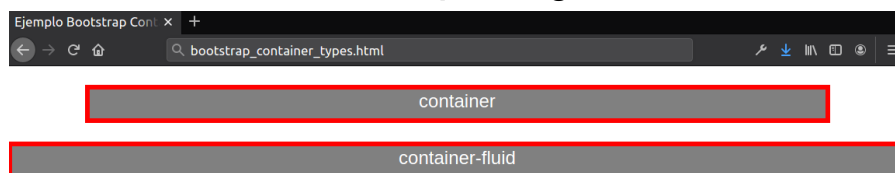
breakpoint: sm



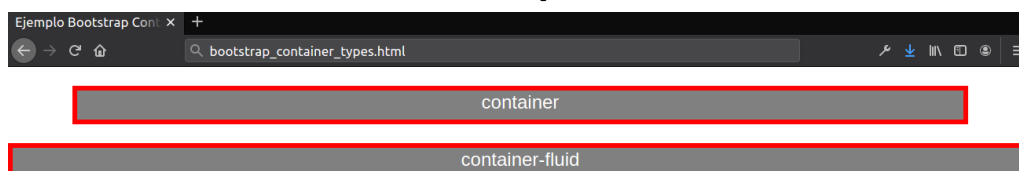
breakpoint: md



breakpoint: lg



breakpoint: xl



row

Uso típico: `<div class="row">...</div>`

Una fila o row es utilizada para definir una selección de elementos que debe ser manejado como un grupo horizontal de elementos. La facilidad de estas filas consiste en que se ordenan verticalmente, recurriéndose a ellas usualmente para definir secciones de una página. Crear una fila es tan simple como asignar la clase row al elemento particular. Las facilidades más interesantes de Bootstrap se comienzan a sentir cuando este tipo de filas se combinan con la clase de columnas del framework, la cuales tratamos a continuación.

col

Uso típico: `<div class="col-1">...</div>`

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

Podría decirse que las columnas son la pieza más importante del sistema de grilla de Bootstrap. Existen filas dentro de los contenedores, y esas filas tienen la posibilidad de dividirse hasta en 12 partes. Esas unidades columnares podemos agruparlas de diversas formas para tener columnas de distinto ancho, siempre manteniendo la suma total de 12 unidades columnares totales en una fila.

Por lo tanto, el diagrama esquemático de cómo podríamos potencialmente dividir un contenedor es como se muestra a continuación:

container

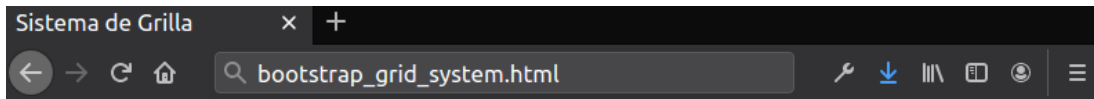
row 1	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9	col 10	col 11	col 12
row 2												
row 3												
row 4												

Podemos utilizar la clase **col** (`<div class="col">...</div>`) para indicar que deseamos que Bootstrap distribuya automáticamente. En este caso incluimos tantos elementos dentro de una fila como columnas queramos tener. Bootstrap dividirá el ancho total disponible para asignar anchos iguales a todas las columnas.

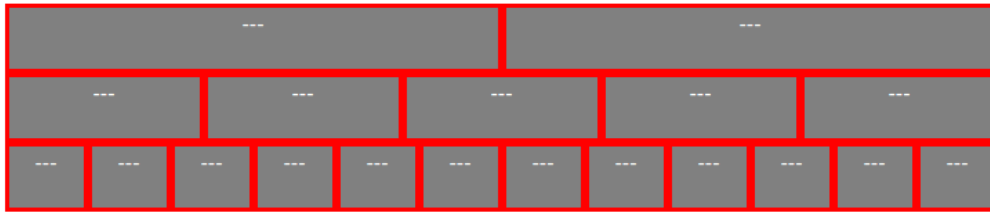
Los conceptos de **container**, **row** y **col** vistos hasta el momento, se traducen en el siguiente código HTML que implementa a modo de ejemplo una distribución de elementos <div> basado en la grilla de Bootstrap, utilizando la clase **col**. De esta forma solicitamos a Bootstrap distribuir uniformemente en columnas iguales los elementos que incluyamos en cada fila:

```
<div class="container">
  <h4>Grilla Bootstrap</h4>
  <div class="row">
    <div class="col">----</div>
    <div class="col">----</div>
  </div>
  <div class="row">
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
  </div>
  <div class="row">
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
    <div class="col">----</div>
  </div>
</div>
```

El resultado de esto, al aplicarlo al último código de la sección 1.4.1.3., con algunos estilos para visualización más didáctica es:



Grilla Bootstrap



En este caso, no hemos solicitado a Bootstrap ninguna función de responsividad, más que ajustar nuestra página al ancho del dispositivo por lo que no habrá diferencia en la disposición de los elementos `<div>` para distintos viewports, y en un dispositivo móvil se verá como se muestra en la figura del costado. Hemos utilizado las herramientas de desarrollador de nuestro navegador para simular la vista de un dispositivo Galaxy S9 con Android 7.0 y navegador Chrome67. Verificamos de esta forma que nuestro diseño se ajusta a la pantalla. Sin embargo, no tenemos ninguna redistribución de columnas para hacernos más fácil la visualización en dispositivo móvil.



Con el fin de tener más control sobre el ancho de columnas y la responsividad de nuestra página, el sistema de grilla de Bootstrap4 provee cinco clases más específicas. De esta forma, podemos tener control del número de columnas a mostrar en los distintos tamaños de dispositivos según los diversos breakpoints que ya hemos venido mencionando:

`.col-*` (dispositivos extra small - viewport menor a 576px)

.col-sm-* (dispositivos small – viewport mayor o igual a 576px)
.col-md-* (dispositivos medium – viewport mayor o igual a 768px)
.col-lg-* (dispositivos large – viewport mayor o igual a 992px)
.col-xl-* (dispositivos extra large – viewport mayor o igual a 1200px)

Debemos reemplazar el símbolo * indicado en esta lista de clases, con el número de unidades columnares de Bootstrap (12 en total como hemos mencionado) que deseemos que posea esa columna en particular.

Para nuestro último ejemplo podemos hacer algunas modificaciones a las clases asignadas a los elemento <div> dentro de cada fila. Es posible asignar múltiples clases a un mismo elemento de forma tal que controlemos cómo se verá en dispositivos que estén dentro de los distintos rangos de breakpoints.

Debemos tener presente que, al momento de definir clases de este último listado, que hagan referencia a los breakpoints de las Media Queries para manejo de responsividad mayores a xs, Bootstrap asumirá que en todos los tamaños menores éste deberá mostrar cada elemento <div> con un ancho de 12 unidades columnares Bootstrap. Así al comenzar visualizando en tamaños xs, las columnas ocuparan el ancho completo de la pantalla. Al probar con una herramienta de desarrollo incrementando el ancho de nuestra pantalla, el ajuste a la nueva distribución de número y ancho de columnas ocurrirá en el breakpoint que hayamos definido. La disposición de esos elementos se mantendrá a medida que vayamos aumentando el ancho de nuestra pantalla, hasta el próximo breakpoint donde hayamos definido una nueva disposición de columnas, si es que lo hemos hecho. De lo contrario continuará hasta xl con la misma disposición.

Nuestro ejemplo anterior, lo modificamos de la siguiente forma para ilustrar lo que hemos expresado en el último párrafo:

```
<div class="container">
  <h4>Grilla Bootstrap</h4>
  <div class="row" style="color:white;">
    <div class="col-4">---</div>
    <div class="col-8">---</div>
  </div>
  <div class="row" style="color:blue;">
    <div class="col-md-4 col-lg-2">---</div>
    <div class="col-md-4 col-lg-2">---</div>
```

```

<div class="col-md-4 col-lg-6">---</div>
<div class="col-md-6 col-lg-1">---</div>
<div class="col-md-6 col-lg-1">---</div>
</div>
<div class="row" style="color:yellow;">
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
  <div class="col-sm-6 col-lg-1">---</div>
</div>
</div>

```

Hemos agregado algunos estilos de color de fuente en modo “inline” para diferenciar visualmente los resultados de lo que hemos modificado.

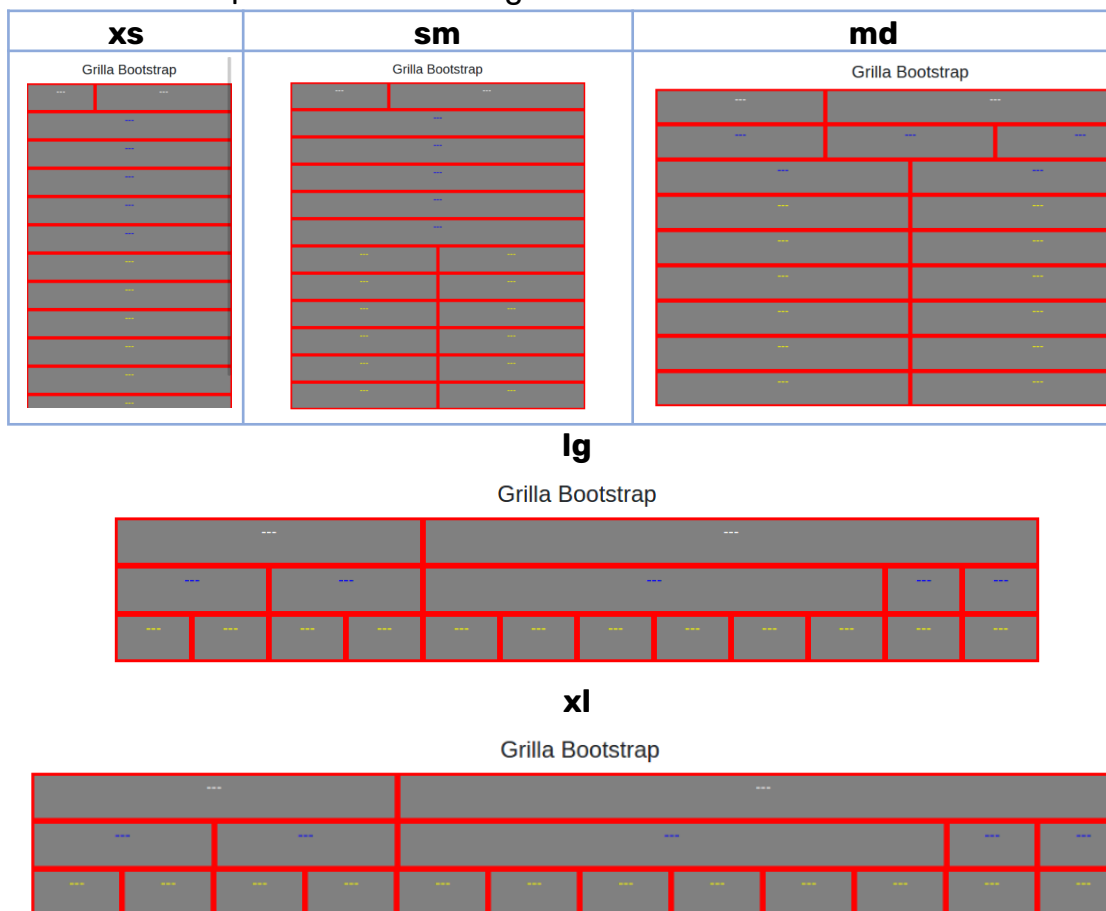
En resumen, bajo el paradigma Mobile First, partimos del supuesto que, en un dispositivo muy pequeño (**xs**), todas las columnas se visualizarán ordenadas verticalmente ocupando cada uno 100% del ancho de la pantalla. Bajo ese supuesto inicial de diseño, nos preguntamos si realmente es lo que deseamos.

En nuestro caso, decidimos que el primer elemento **<div class="row">** siempre se muestre en 2 columnas, donde la primera ocupe 4 y la segunda 8 unidades columnares, independiente del ancho de la pantalla del dispositivo.

Para el segundo **<div class="row">** de nuestro ejemplo, decidimos que desde al superar el breakpoint **md** las primeras tres columnas se adapten a 4 unidades columnares cada una y las dos siguientes a 6 unidades columnares cada una. Así, en ese contexto, el elemento **<div>** será desplegado visualmente en dos filas. Posteriormente, superando el breakpoint **lg**, todas las columnas se presentarán visualmente en una misma fila, con anchos en unidades columnares de: 2, 2, 6, 1 y 1.

Para el tercer elemento `<div class="row">`, estaremos de acuerdo en mantener todas las columnas desplegadas verticalmente hasta superar el breakpoint **sm**, momento en el cual deseamos que cada una comience a tener un ancho de 6 unidades columnares. De esta forma los 12 elementos serán mostrados visualmente en 6 filas. Esto se mantendrá hasta el breakpoint **lg**, donde comenzaremos a observar 12 columnas desplegadas visualmente en una fila.

Gráficamente podemos ver los siguientes casos:



Son muchas las facilidades que provee el método revisado hasta el momento para diagramar y distribuir contenidos en un esquema responsivo dentro de nuestra página web.

Es posible incorporar elementos anidados, tales como containers dentro de otros containers. Se recomienda al lector experimentar alterando los códigos que hemos revisado hasta el momento. Asimismo, es de utilidad buscar plantillas para sitios web, basadas en HTML5 y Bootstrap4, y explorar la forma en que han sido diseñadas. Existen muchísimos más detalles sobre las posibilidades que ofrece Bootstrap. Por razones de

espacio y tiempo, acá no es posible hacer una revisión taxativa de todos los detalles del framework y su Sistema de Grilla. Es hábito del desarrollador aprender los conceptos básicos y posteriormente expandir su conocimiento a través de estudio de documentación del framework y “disección” de código de proyectos existentes.

Componentes principales de Bootstrap 4.

Adicionalmente al Sistema de Grilla, Bootstrap tiene muchas prestaciones para confeccionar sitios con componentes prediseñados, de interfaz de usuario UI, que facilitan nuestra tarea. Podemos centrarnos en el concepto de nuestras secciones y contenidos, en lugar de desgastarnos diseñando visualmente cada elemento de nuestra página. Entre los componentes que ofrece Bootstrap se encuentran:

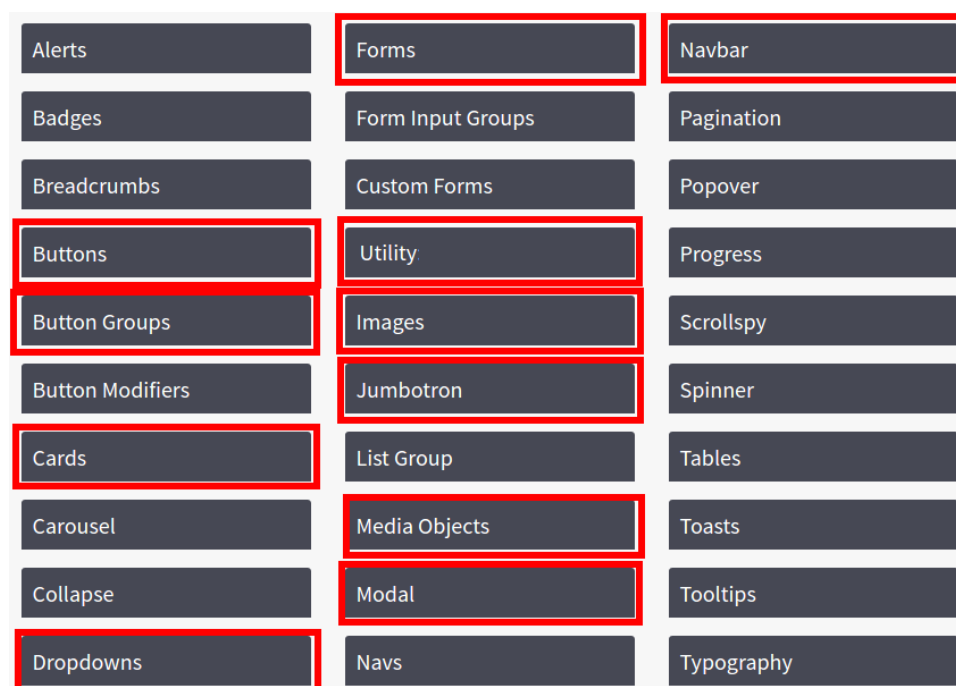
Alerts	Forms	Navbar
Badges	Form Input Groups	Pagination
Breadcrumbs	Custom Forms	Popover
Buttons	Utility	Progress
Button Groups	Images	Scrollspy
Button Modifiers	Jumbotron	Spinner
Cards	List Group	Tables
Carousel	Media Objects	Toasts
Collapse	Modal	Tooltips
Dropdowns	Navs	Typography

Fuente: <https://hackerthemes.com/bootstrap-cheatsheet/>

Más que revisar de manera aislada cada uno de estos componentes, abordaremos algunos de ellos bajo una perspectiva práctica. Iremos acumulando los componentes que revisemos con una estrategia acumulativa. Construiremos una base de proyecto que permitirá al lector contar con un punto de partida para otros proyectos, y además para

comprender la estructura de plantillas disponibles en internet que están basadas en conjuntos similares de componentes Bootstrap.

Partiremos desde la estructura de proyecto **HTML + JQuery + Popper.js + Bootstrap4** que vimos en la sección 1.4.1.3. y agregaremos progresivamente distintos elementos para dar una visión de cual es el proceso a seguir en un proyecto web que utilice las facilidades de Bootstrap. A continuación destacamos en color rojo los principales componentes que incorporaremos en este ejemplo:



Daremos inicio a nuestro código HTML incorporando a nuestro proyecto base las distintas secciones, que estarán basadas en elementos **<div>** a los que asignaremos la clase **container-fluid**. Además, con el fin de diferenciar visualmente estas secciones, asignamos algunas utilidades de color de Bootstrap para asignarles un color de fondo alternado. Usamos las utilidades **bg-primary** para la primera sección y alternamos entre **bg-light** y **bg-dark** en las siguientes secciones. Por último, agregamos un id para referenciar cada una de estas secciones, cuya utilidad iremos apreciando a medida que avancemos en el proyecto. Así, nuestro código inicial queda como se indica a continuación (Se destaca en color rojo el código agregado a nuestro código base de la sección 1.4.1.3):

Archivo: [index.html](#)

```

<!doctype html>
<html lang="es">
  <head>
    <title>Awakelab</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Loading Bootstrap CSS -->
    <link
      rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity="sha384-9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8WgZl
5MYYYxYFfc+NcPb1dKGj7Sk"
      crossorigin="anonymous">
    </head>
    <body>
      <div class="container-fluid bg-primary" id="principal">
        <div class="row">
          <h1>Grupo Full Stack Python de Awakelab</h1>
        </div>
      </div>
      <div class="container-fluid bg-dark" id="competencias">
        <div class="row">
          <h3>Nuestras Competencias</h3>
        </div>
      </div>
      <div class="container-fluid bg-light" id="equipo">
        <div class="row">
          <h3>Nuestro Equipo</h3>
        </div>
      </div>
      <div class="container-fluid bg-dark" id="enlaces">
        <div class="row">
          <h3>Enlaces de Interés</h3>
        </div>
      </div>
      <div class="container-fluid bg-light" id="contacto">
        <div class="row">
          <h2>Contáctanos</h2>
        </div>
      </div>
      <footer class="footer bg-dark">
        <p class="text-muted">&copy; Awakelab.</p>

```



```

<p class="text-muted">Python Developers</p>
<p class="text-muted">Link</p>
</footer>
<!-- jQuery, Popper.js, Bootstrap JS -->
<script
  src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVB
nE+IbbVYUew+OrCXaRkfj"
  crossorigin="anonymous">
</script>
<script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.
js"

integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTt
mI3UksdQRVvoxMfooAo"
  crossorigin="anonymous">
</script>
<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.
min.js"

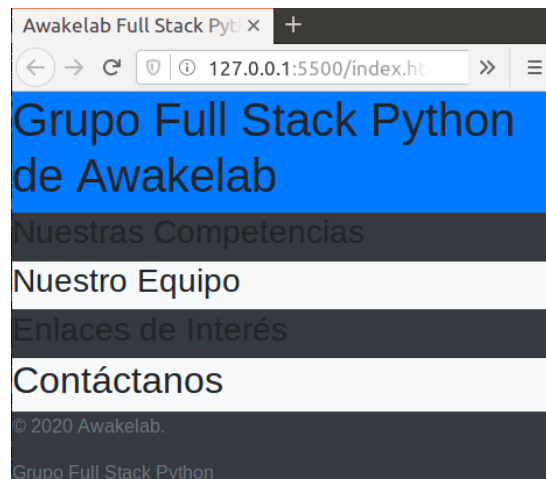
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZ
E3Ipu6Tp75j7Bh/kR0JKI"
  crossorigin="anonymous">
</script>
</body>
</html>

```

Nos detenemos un momento para sugerir al lector que instale la Extensión de **Visual Studio Code** llamada **Live Server 5.6.1**. Esta extensión nos permitirá servir nuestra página web desde la dirección local de nuestro PC de desarrollo. De esta forma, no necesitamos abrir el archivo desde una larga ruta hasta el archivo index.html, sino que podemos directamente verlo en operación a través del puerto 5500 en nuestra dirección IP de localhost, es decir, <http://127.0.0.1:5500>. Esto nos da la interesante facilidad adicional de poder acceder a nuestra página desde otros dispositivos que puedan conectarse a nuestra máquina en la red local. Por ejemplo, si nuestro PC está conectado a una red WiFi con la dirección IP 192.168.1.43, otros dispositivos tales como teléfonos y otros

computadores podrán acceder a nuestra página escribiendo en su navegador la URL <http://193.168.1.43:5500>

De esta forma visualmente obtenemos lo siguiente:



Dado que estaremos agregando código y preocupándonos de mejorar el aspecto visual de nuestro sitio, crearemos inmediatamente una hoja de estilos CSS en un archivo externo y que contendrá todas las personalizaciones que hagamos de nuestra página y de los componentes Bootstrap que utilicemos. Como se observa en el código, nuestra implementación utiliza el framework Bootstrap bajo modalidad CDN, por lo que sólo tenemos el archivo **index.html** en la carpeta del proyecto. Crearemos el directorio **css** y ubicaremos en su interior nuestro archivo **style.css**. En nuestro archivo HTML agregaremos, dentro de la sección **<head>**, el enlace **<link rel="stylesheet" href="css/style.css">**. El contenido inicial de nuestro archivo CSS será principalmente especificar **min-height** de nuestras secciones y la propiedad **padding-top** de las mismas. Utilizaremos selectores individuales para cada sección, a modo de práctica de utilización de distintos tipos de selectores CSS revisados en contenidos anteriores.

Archivo: `css/style.css`

```
#principal,  
#competencias,
```

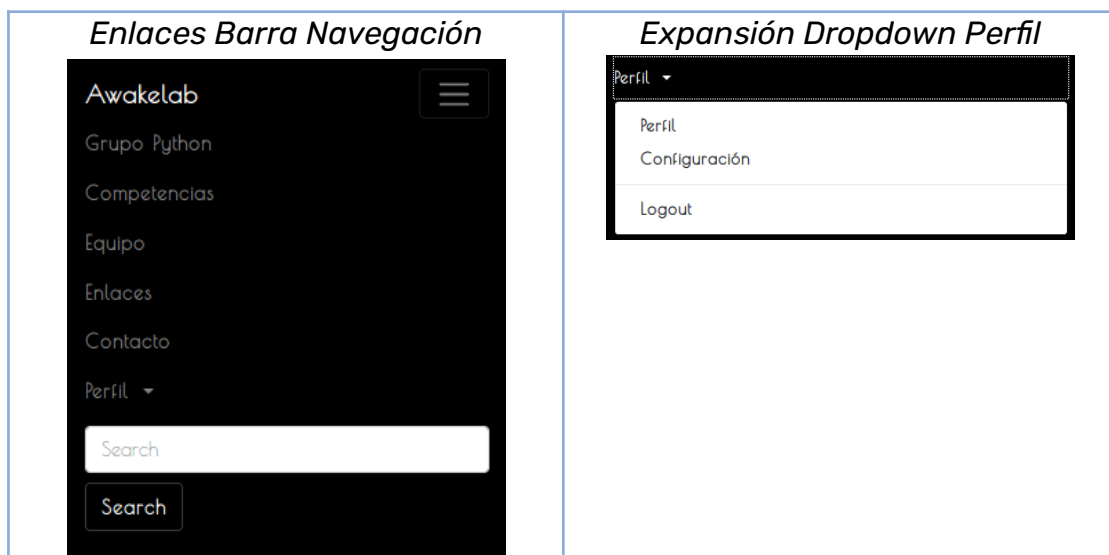
```
#equipo,
#enlaces,
#contacto {
  padding-top: 60px;
  min-height: 700px;
}
```

NAVBAR, DROPDOWN, FORM, MODAL

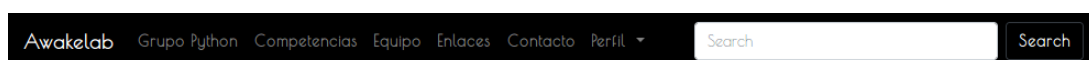
Agregamos a continuación el código relacionado con la incorporación de una barra de navegación para poder acceder a través de links a cada una de las secciones, junto con agregar dos funcionalidades más a modo ilustrativo: un componente **dropdown** para perfil de usuario, y un componente de **formulario**. El perfil de usuario a su vez gatillará dos componentes **modales**, uno para perfil de usuario y otro de configuración de cuenta.

Desde el punto de vista estético, queremos lograr el siguiente diseño gráfico, para la barra de navegación móvil, y su equivalente en formato de escritorio:

Vista Móvil Barra de Navegación Objetivo



Vista Escritorio



Cada componente de Bootstrap contiene un conjunto de elementos HTML y asignación de clases que en algunos casos no son simples de comprender como principiante. Sin embargo, en el desarrollo e incorporación de estos componentes es usual recurrir a referencias en las que se pueden obtener los códigos correspondientes a cada uno de éstos. Sugerimos considerar la utilización de los mencionados códigos de componente como un bloque predefinido, los que serán progresivamente entendidos más profundamente con la práctica, y a través de la experiencia que entrega involucrarse en intentos de personalización de los mismos.

Como ejemplo de estos recursos online para obtención de plantillas de código para todos los componentes de Bootstrap, están:

- Documentación de Bootstrap:
<https://getbootstrap.com/docs/4.5/getting-started/introduction/>
- W3Schools:
<https://www.w3schools.com/bootstrap4/>
- Bootstrap4 Cheatsheet:
<https://hackerthemes.com/bootstrap-cheatsheet/>

De esta forma podemos implementar un código inicial, a ser incorporado dentro de **<body>** que implemente una primera versión de la barra de navegación que nos hemos puesto como objetivo:

```
<!-- Navigation Bar -->
<nav class="navbar navbar-expand-lg navbar-dark fixed-top">
  <a class="navbar-brand" href="#">Awakelab</a>

  <!-- Navigation Bar Hamburger-->
  <button
    class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navigation"
    aria-expanded="false"
    aria-controls="navigation"
    aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <!-- End Navigation Bar Hamburger -->

  <!-- Navigation Links -->
```

```
<div class="collapse navbar-collapse" id="navigation">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item">
      <a
        class="nav-link scroll-trigger"
        href="#principal">
        Welcome
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link scroll-trigger"
        href="#servicios">
        Servicios
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link scroll-trigger"
        href="#galeria">
        Galería
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link scroll-trigger"
        href="#acercade">
        Acerca de
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link scroll-trigger"
        href="#contacto">
        Contacto
      </a>
    </li>
    <li class="nav-item dropdown">
      <a
        href="#"
        class="nav-link dropdown-toggle"
        data-toggle="dropdown"
        role="button"
        aria-haspopup="true"
        aria-expanded="false">
        Perfil <span class="caret"></span>
      </a>
```

```

<div class="dropdown-menu dropdown-menu-right">
  <a
    class="dropdown-item scroll-trigger"
    href="#"
    data-toggle="modal"
    data-target="#profile-modal">
    Perfil
  </a>
  <a class="dropdown-item scroll-trigger"
    data-toggle="modal"
    data-target="#settings-modal">
    Configuración
  </a>
  <div class="dropdown-divider"></div>
  <a class="dropdown-item scroll-trigger" href="">
    Logout
  </a>
</div>
</li>
</ul>
<form class="form-inline my-2 my-lg-0">
  <input class="form-control mr-sm-2"
    type="text"
    placeholder="Search"
    aria-label="Search">
  <button class="btn btn-outline-success border-dark
    font-weight-bold text-white my-2 my-sm-0"
    type="submit">Search</button>
</form>
</div>
<!-- Navigation Links -->

</nav>
<!-- End Navigation Bar -->

```

Todo el código anterior corresponde a un elemento HTML **<nav>** que es nuestra barra de navegación para el sitio. Desde el punto de vista móvil, agregamos primeramente el elemento **<button>** de clase **navbar-toggler** de aspecto “Hamburguesa” ☰, el cual se alimenta de los datos que están en el **<div>** que posee asignado **id="navigation"**. Para esto, se especifica el atributo **data-target="#navigation"** que apunta hacia dicho **<div>**. Este botón se expande para mostrar todas las opciones de enlaces de

clase a las secciones del sitio, que se encuentran especificados en los elementos **<a>** de clase **nav-link**. Debemos hacer notar que si no incluimos el element **<button>**, la barra de navegación se implementará sólo con el **<div>** de **id="navigation"** y tendrá disponible solamente el aspecto extendido, apropiado para pantallas grandes.

Además, como puede verse en el código mostrado, hemos agregado tres componentes adicionales en la barra de navegación:

- Un componente Dropdown (**dropdown-menu**) para configuraciones de perfil de usuario.
- Dos componentes Modales, que son llamados desde los links del componente Dropdown; para Perfil y Configuración.
- Un componentes Formulario (**form-inline**) para búsqueda en nuestro sitio.

Un formulario modal es un código HTML que es mostrado por el navegador como una ventana emergente. Ésta puede ser sólo informativa, contener un pequeño formulario u opciones que el usuario deba definir.

Los modales son llamados con **data-target="#profile-modal"** y **data-target="#settings-modal"** desde los elementos de clase **dropdown-item** del componente **dropdown-menu**. Así, los modales deberán tener definidos los id **profile-modal** y **settings-modal**.

Como convención para ordenar nuestro código, los elementos correspondientes a componentes modales de perfil y configuración los ubicaremos en la parte final del **<body>** de nuestra página, justo antes de los elementos **<script>**. En este caso los códigos que agregaremos son:

```
<!-- MODAL FORMS -->
<div class="modal" id="profile-modal" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
          aria-label="Close"><span aria-hidden="true">&times;</span></button>
        <p class="modal-title" id="profile-modal-label">Perfil</p>
      </div>
      <div class="modal-body">
```



```

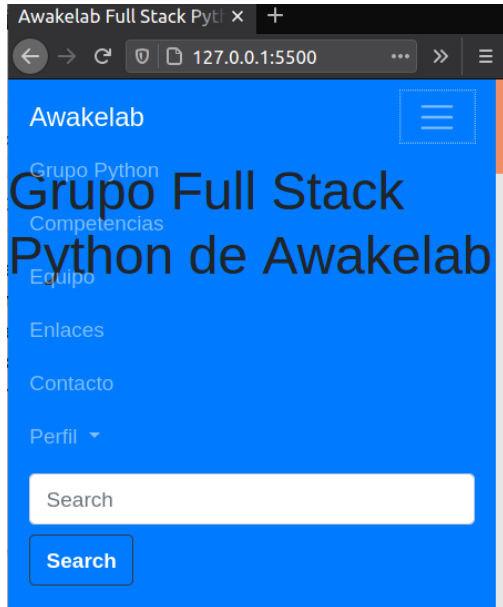
        Perfil
    </div>
    <div class="modal-footer">
        <button type="button" class="btn
btn-success">Guardar</button>
        <button type="button"
        class="btn btn-secondary"
        data-dismiss="modal">Cancelar</button>
    </div>
</div>
</div>
</div>
<div class="modal" id="settings-modal" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal"
                aria-label="Close"><span aria-hidden="true">&times;</span>
            </button>
                <p class="modal-title" id="settings-modal-label">
                    Configuración</p>
            </div>
            <div class="modal-body">
                Configuración
            </div>
            <div class="modal-footer">
                <button type="button"
                class="btn btn-success">Guardar</button>
                <button type="button"
                class="btn btn-secondary"
                data-dismiss="modal">Cancelar</button>
            </div>
        </div>
    </div>
</div>
</div>
<!-- END MODAL FORMS -->

```

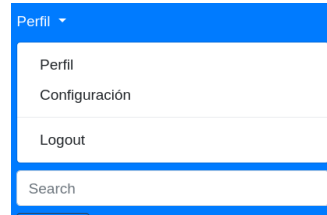
Cada uno de los modales contienen las secciones **<header>**, **<body>** y **<footer>**. En **<header>** incluimos un botón de cerrar y el nombre del formulario modal, en **<body>** un texto alusivo al tema del modal, y en footer dos botones: uno para guardar y otro para cancelar.

El resultado de lo anterior en el navegador es como se muestra a continuación:

Vista de barra de navegación expandida en vista móvil



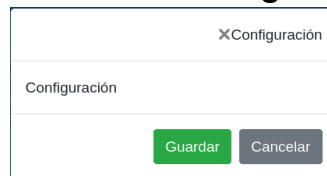
dropdown-menu



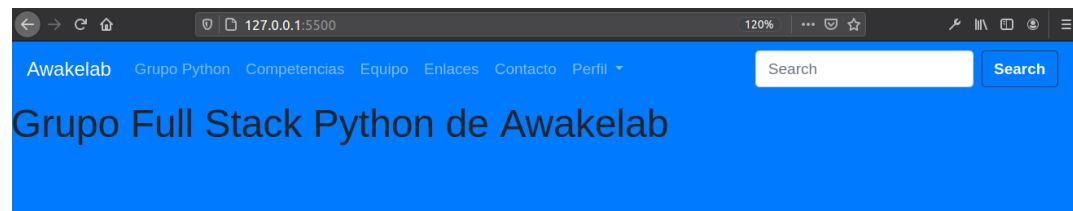
modal-profile



modal-settings



Vista de barra de navegación expandida en vista desktop



Según esto, aún nos quedan algunos detalles de estilo para lograr el objetivo visual que nos propusimos inicialmente.

En nuestro archivo **css/style.css**, agregaremos las siguientes reglas que nos permitirán tener fondo negro en la barra de navegación y textos en negrita:

```
.navbar a {  
  font-weight: bold;  
}  
  
.navbar-dark {
```

```
background-color:black;
}
```

Por último, con el fin de tener una tipografía especial como la que nos propusimos en el diseño gráfico inicial, importaremos la fuente requerida desde CDN de google, agregando el siguiente elemento en la sección **<head>** de nuestro documento HTML:

```
<link
  rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstr
ap.min.css"

integrity="sha384-9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8Wg
ZI5MYYYxFfc+NcPb1dKGj7Sk"
  crossorigin="anonymous">
```

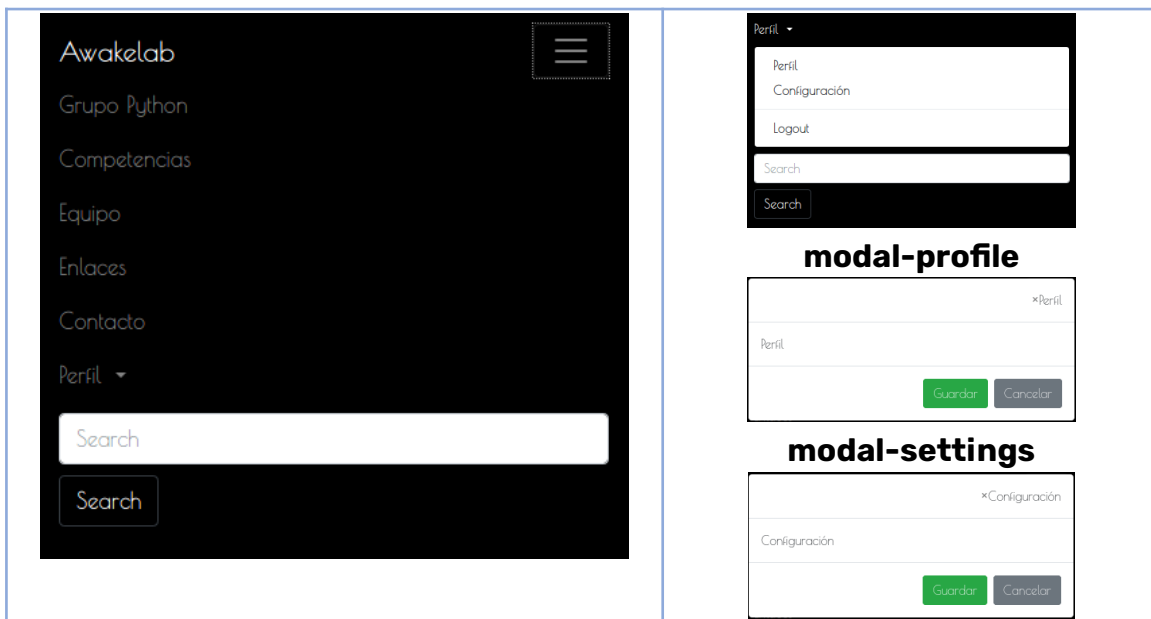
Y agregamos en **css/style.css** el uso de esta fuente en los textos de las clases **navbar** y **modal**. Agregamos también otros selectores a esta regla para homogeneizar tipografías posteriores (h1, h2, h3, h4, h5, h6 para homogeneizar):

```
h1, h2, h3, h4, h5, h6, .navbar, .modal {
  font-family: 'Poiret One', cursive;
}
```

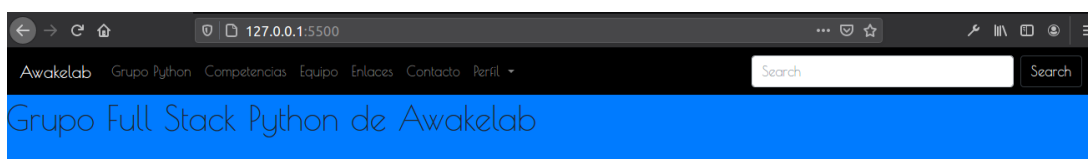
De esta forma obtenemos el diseño objetivo de nuestro diseño:

**Vista de barra de navegación
expandida en vista móvil**

dropdown-menu



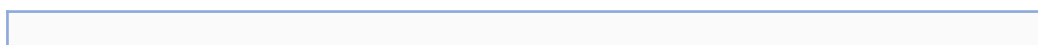
Vista de barra de navegación expandida en vista desktop



JUMBOTRON, IMÁGENES

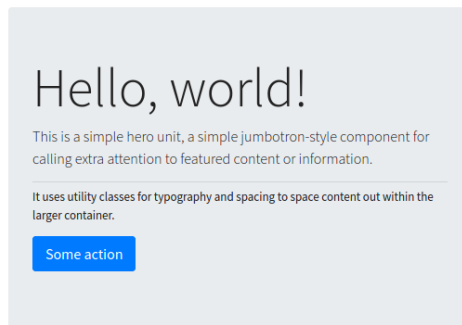
Ahora nos enfocaremos en las diferentes secciones de nuestra página. En la zona superior queremos incluir una imagen llamativa y una zona de texto destacado que presente el título principal de nuestra página.

Para la imagen, crearemos en **css/style.css** una clase **bg-principal** con la que reemplazaremos la clase **bg-primary** que hemos utilizado hasta el momento para el **container-fluid** de la sección principal. Crearemos un directorio **img/** al mismo nivel jerárquico que el directorio **css/** en nuestro directorio de proyecto y colocaremos dentro una imagen denominada **bg-wind-generators.jpg**. Luego en nuestro archivo CSS agregaremos la siguiente regla, que agrega la imagen mencionada como fondo del container. Se incluye la propiedad **background-size:cover**, con la que la imagen se configura para cubrir la totalidad del área del container.



```
.bg-principal {  
  background-image: url('../img/bg-wind-generators.jpg');  
  background-size: cover;  
}
```

Para el texto destacado, recurriremos al componente **jumbotron** que incluiremos en nuestro código HTML para potenciar el elemento `<h1>` actual. Un componente **jumbotron** luce como el siguiente ejemplo:



El código para **jumbotron** es simple y equivale a lo siguiente (Hemos dividido el texto de `<h1>` en dos elementos: `<h2>` y `<h3>`):

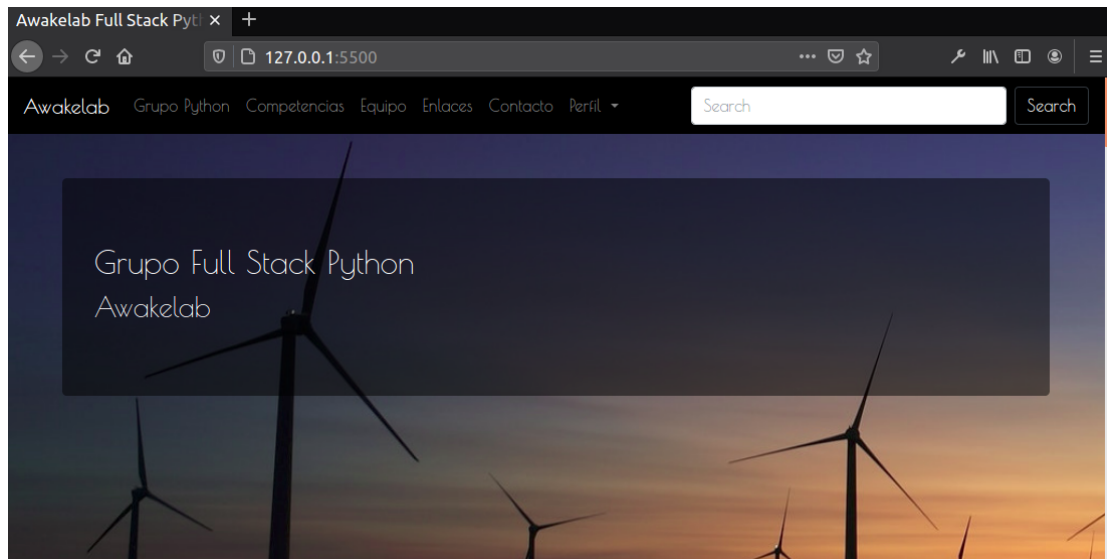
```
<div class="col-sm-12">  
  <div class="jumbotron jumbotron-main">  
    <h2>Grupo Full Stack Python</h2>  
    <h3>Awakelab</h3>  
  </div>  
</div>
```

Este código HTML lo acompañamos con la siguiente regla para estilos en nuestro archivo **css/style.css**, que define color blanco para los textos y fondo negro con un 50% de transparencia; además de padding separando el estilo para el **id principal**:

```
.jumbotron {  
  color: white;  
  background: rgb(0, 0, 0); /* Para navegadores antiguos */  
  background: rgba(0, 0, 0, 0.5);  
}  
  
#principal {  
  padding: 100px 5% 20px;
```

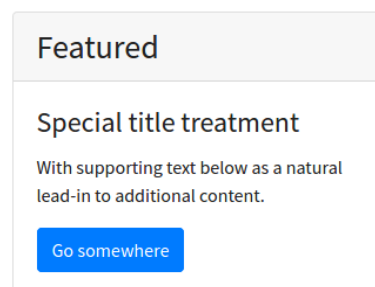
```
min-height: 700px;
}
```

Con todos estos cambios logramos tener la siguiente sección **principal**, que es la bienvenida a nuestro sitio:



CARDS

Continuamos revisando componentes de Bootstrap4 con la siguiente sección de nuestro sitio. Hemos denominado “Competencias” a esta sección haciendo alusión a las habilidades que tiene el Grupo Full Stack Python de Awakelab. Deseamos ordenar la información de esta sección en paneles o tarjetas que en nuestra vista Mobile First estén ordenadas verticalmente, y cada una aborde una competencia del grupo, conteniendo un título, un subtítulo, un texto principal y un botón de “ver más” para revisar más detalles sobre ese tópico. Una **card** estándar de Bootstrap se ve más o menos de la siguiente forma:



Utilizaremos el Sistema de Grilla para ordenar nuestras tarjetas y que éstas se adapten en los casos que requieran ser mostradas en una pantalla más ancha que la de un dispositivo móvil. Para esto tomaremos la decisión de tener dos grupos de tarjetas: Un primer grupo con 4 tarjetas y otro con 3 tarjetas, y deseamos que estos grupos se comporten de manera distinta al sufrir redistribuciones por cambio de dimensiones de la ventana de navegación.

Ambos grupos de tarjetas reaccionarán con una primera redistribución en el breakpoint **md**, en el que el primer grupo pasará a mostrarse en dos filas de dos tarjetas cada una, con un ancho de cada tarjeta de 6 unidades columnares según Bootstrap. El segundo grupo, al sobrepasar el mismo breakpoint, se redistribuirá en una única fila de 3 tarjetas.

Finalmente, el primer grupo tendrá una siguiente redistribución en el breakpoint **lg**, a partir del cual las 4 tarjetas serán mostradas en una única fila. El segundo grupo no sufrirá redistribuciones después de la mencionada para el breakpoint **md**.

Para esto, según lo aprendido en secciones anteriores sobre el sistema de grilla de Bootstrap, y recurriendo a código de bloques de los componentes de tipo **card**, tendremos el siguiente código que implementa lo descrito en los párrafos anteriores. Éste consiste principalmente en dos nuevos `<div>` de clase **row**, dentro del container principal:

```
<div class="row">
  <div class="col-md-6 col-lg-3">
    <div class="card">
      <h3 class="card-header">Competencia 1</h3>
      <div class="card-body">
        <h4 class="card-title">Subtítulo competencia</h4>
        <p class="card-text">Descripción de la competencia del
          grupo Full Stack Python de Awakelab.</p>
        <a href="#" class="btn btn-dark">Ver mas</a>
      </div>
    </div>
  </div>
  <div class="col-md-6 col-lg-3">
    <div class="card">
      <h3 class="card-header">Competencia 2</h3>
      <div class="card-body">
        <h4 class="card-title">Subtítulo competencia</h4>
        <p class="card-text">Descripción de la competencia del
          grupo Full Stack Python de Awakelab.</p>
      </div>
    </div>
  </div>
</div>
```



```

        <a href="#" class="btn btn-dark">Ver mas</a>
    </div>
</div>
</div>
<div class="col-md-6 col-lg-3">
    <div class="card">
        <h3 class="card-header">Competencia 3</h3>
        <div class="card-body">
            <h4 class="card-title">Subtítulo competencia</h4>
            <p class="card-text">Descripción de la competencia del
                grupo Full Stack Python de Awakelab.</p>
            <a href="#" class="btn btn-dark">Ver mas</a>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-3">
    <div class="card">
        <h3 class="card-header">Competencia 4</h3>
        <div class="card-body">
            <h4 class="card-title">Subtítulo competencia</h4>
            <p class="card-text">Descripción de la competencia del
                grupo Full Stack Python de Awakelab.</p>
            <a href="#" class="btn btn-dark">Ver mas</a>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-md-4">
        <div class="card">
            <h3 class="card-header">Competencia 5</h3>
            <div class="card-body">
                <h4 class="card-title">Subtítulo competencia</h4>
                <p class="card-text">Descripción de la competencia del
                    grupo Full Stack Python de Awakelab.</p>
                <a href="#" class="btn btn-dark">Ver mas</a>
            </div>
        </div>
    </div>
    <div class="col-md-4">
        <div class="card">
            <h3 class="card-header">Competencia 6</h3>
            <div class="card-body">
                <h4 class="card-title">Subtítulo competencia</h4>
                <p class="card-text">Descripción de la competencia del
                    grupo Full Stack Python de Awakelab.</p>
                <a href="#" class="btn btn-dark">Ver mas</a>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="card">
    <h3 class="card-header">Competencia 7</h3>
    <div class="card-body">
      <h4 class="card-title">Subtítulo competencia</h4>
      <p class="card-text">Descripción de la competencia del
        grupo Full Stack Python de Awakelab.</p>
      <a href="#" class="btn btn-dark">Ver mas</a>
    </div>
  </div>
</div>
</div>
</div>

```

Con el fin de cambiar el color del título **<h3>** de la sección “Competencias” a **white**, y a la vez aprovechar de ilustrar otra forma de usar selectores CSS, agregamos la siguiente regla en nuestra hoja de estilos:

```

#competencias > div > h3 {
  color: white;
}

```

Finalmente, agregamos algunos ajustes de estilo a las tarjetas para que tengan una separación mínima entre ellas en las distintas disposiciones:

```

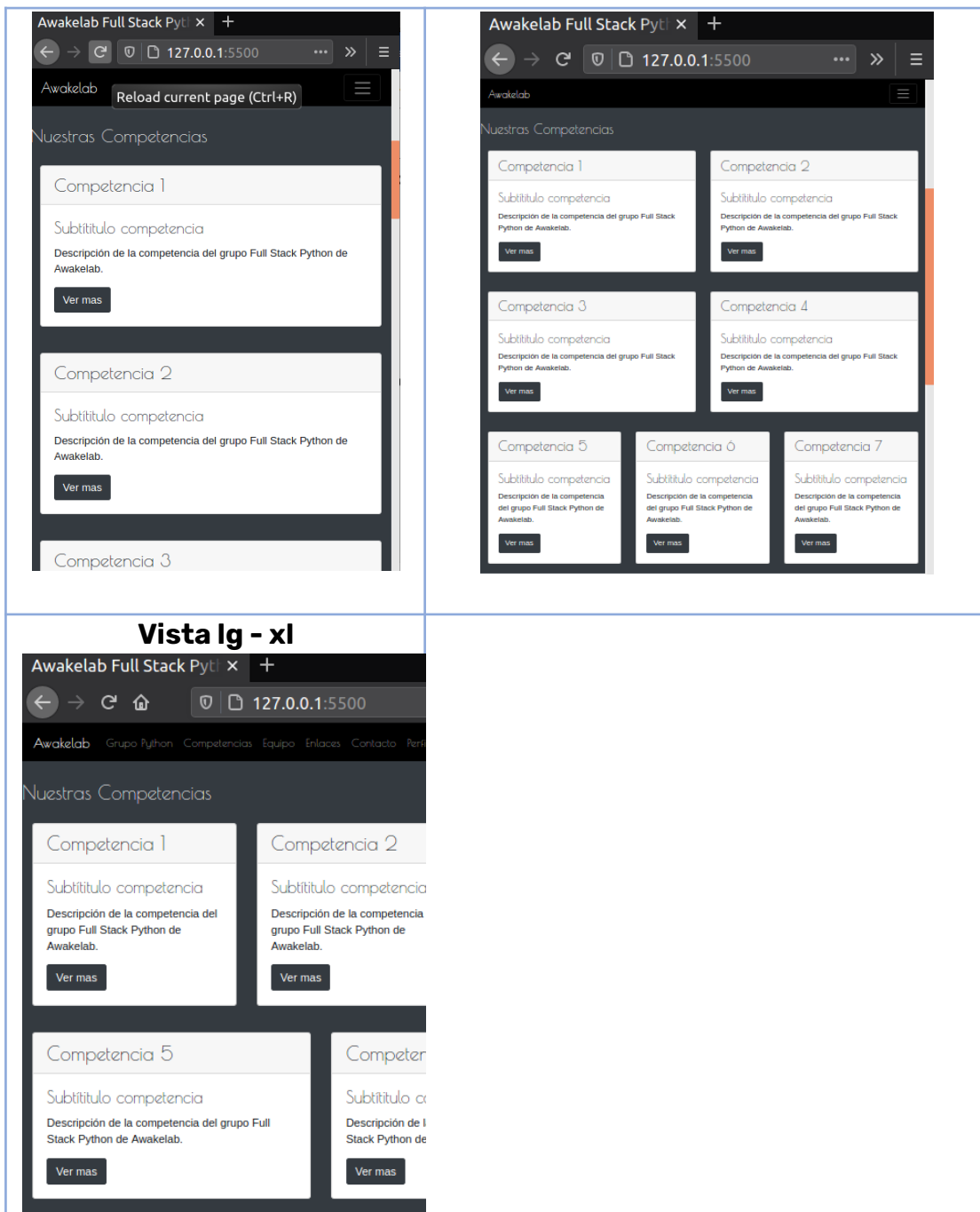
.card {
  margin-top: 20px;
  margin-bottom: 20px;
}

```

Con esto, obtenemos un aspecto satisfactorio para nuestra sección “Competencias” que resulta ser:

Vista Móvil xs - sm

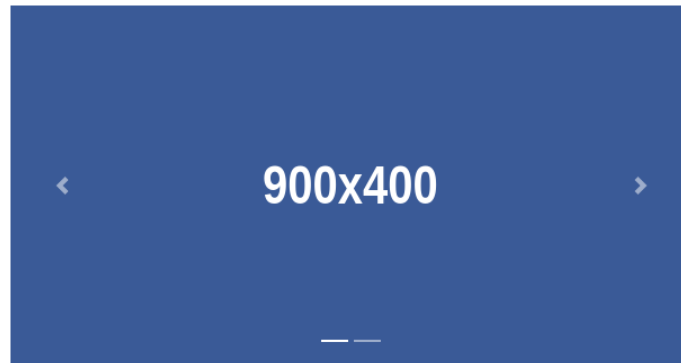
Vista md



CAROUSEL

El siguiente componente Bootstrap lo aplicaremos en la sección “Equipo”, que está orientado a mostrar el equipo humano de nuestro Grupo Full Stack Python de Awakelab. Deseamos mostrar un cuadro de imágenes

deslizantes automáticas donde podamos dar a conocer fotos de los grupos de nuestro equipo de trabajo. Un componente Bootstrap que nos provee esta facilidad es el denominado **carousel**, que hace justamente lo que hemos descrito. Esquemáticamente un componente carousel se ve como en la siguiente imagen:



Configuraremos 3 imágenes y un tiempo de transición entre ellas de 3 segundos. Agregamos, dentro del directorio **img/** de nuestro proyecto, tres archivos de imágenes: **group1.jpg**, **group2.jpg**, y **group3.jpg**. El código correspondiente para ello es como se muestra a continuación, lo agregamos como un nuevo **<div>** de clase **row** en la sección “Equipo”:

```
<div class="row">
  <div id="team-carousel"
    class="carousel slide"
    data-ride="carousel"
    data-interval="3000">
    <div class="carousel-inner" style="padding: 4rem;" role="listbox">
      <div style="height: 400px" class="carousel-item active">
        
        <div class="carousel-caption">
          Grupo Front End
        </div>
      </div>
      <div style="height: 400px" class="carousel-item">
        
        <div class="carousel-caption">
          Grupo Back End
        </div>
      </div>
    </div>
  </div>
```

```

<div style="height: 400px" class="carousel-item">
  
  <div class="carousel-caption">
    Grupo DevOps
  </div>
</div>
</div>
<a class="left carousel-control"
  href="#team-carousel"
  role="button"
  data-slide="prev">
  <span class="icon-prev" aria-hidden="true"></span>
</a>
<a class="right carousel-control"
  href="#team-carousel"
  role="button"
  data-slide="next">
  <span class="icon-next" aria-hidden="true"></span>
</a>
<ol class="carousel-indicators">
  <li data-target="#team-carousel"
    data-slide-to="0"
    class="active"></li>
  <li data-target="#team-carousel"
    data-slide-to="1"></li>
  <li data-target="#team-carousel"
    data-slide-to="2"></li>
</ol>
</div>
</div>

```

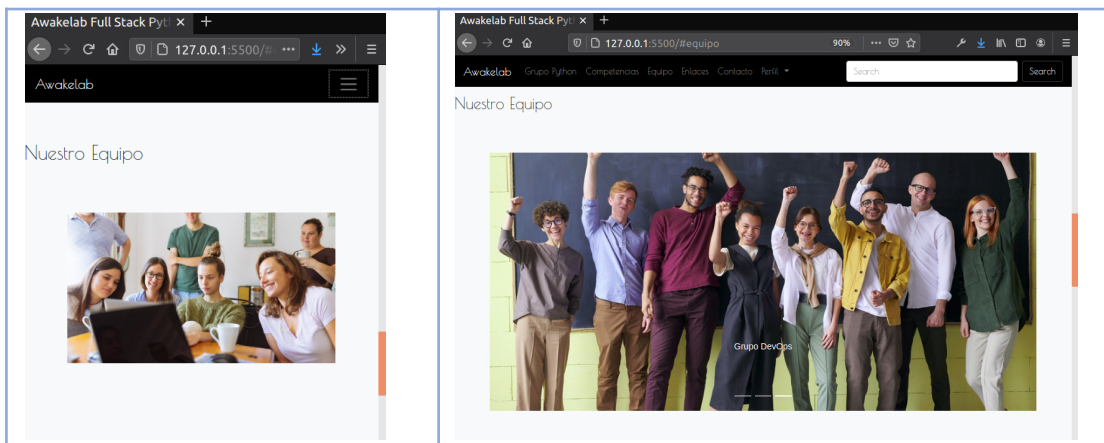
Ajustamos algunos estilos básicos del container de id="equipo" en nuestro archivo de estilos CSS, separándolo de los estilos de las otras secciones:

```

#equipo {
  padding-top: 60px;
  padding-bottom: 60px;
}

```

Con lo que obtenemos el siguiente resultado para vista móvil y desktop:



MEDIA

La siguiente sección, "Enlaces", la aprovechamos para ilustrar el uso de otro componente Bootstrap que se denomina **media**. Este tipo de componente permite incorporar fácilmente un bloque compuesto por una imagen y textos con un orden predefinido como se muestra a continuación:

64x64

Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

En nuestro caso lo utilizaremos para agrupar enlaces a sitios de interés agrupados por tema y con una imagen pequeña que represente el concepto relacionado. Implementaremos una grilla con dos elementos **<div>** de clase **row**, donde cada uno tendrá dos componentes **media**. Todos estos componentes se mostrarán ordenados verticalmente en nuestra aproximación Mobile First, y configuraremos a través de la clase **col-md-6** haremos que al visualizar en una pantalla que supere el breakpoint **md**, éstos se reordenen de manera que se muestren dos componentes por fila.

Para la implementación, agregaremos cuatro imágenes pequeñas a nuestra carpeta **img/** con nombres **flag-64.jpg**, **world-64.jpg**, **screen-64.jpg** y **gear-64.jpg**. Cada uno de los grupos de enlaces tendrá

asignada una de estas imágenes, además de un título y la lista de enlaces. El código HTML será de la siguiente forma:

```
<div class="row">
  <div class="col-md-6">
    <div class="media">
      
      <div class="media-body">
        <h5 class="mt-0">Enlaces Tipo 1</h5>
        <ul>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        </ul>
      </div>
    </div>
  </div>
  <div class="col-md-6">
    <div class="media">
      
      <div class="media-body">
        <h5 class="mt-0">Enlaces Tipo 2</h5>
        <ul>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-6">
    <div class="media">
      
      <div class="media-body">
        <h5 class="mt-0">Enlaces Tipo 3</h5>
        <ul>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
          <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

```

        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
      </ul>
    </div>
  </div>
</div>
<div class="col-md-6">
  <div class="media">
    
    <div class="media-body">
      <h5 class="mt-0">Enlaces Tipo 4</h5>
      <ul>
        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
        <li>Enlace de interés del grupo <a href="#">Ver más</a></li>
      </ul>
    </div>
  </div>
</div>
</div>

```

Además, para asegurar estilos consistentes con la estética de nuestra página, independizamos la regla del container que posee **id="enlaces"**, y agregamos algunos márgenes a los componentes **media**:

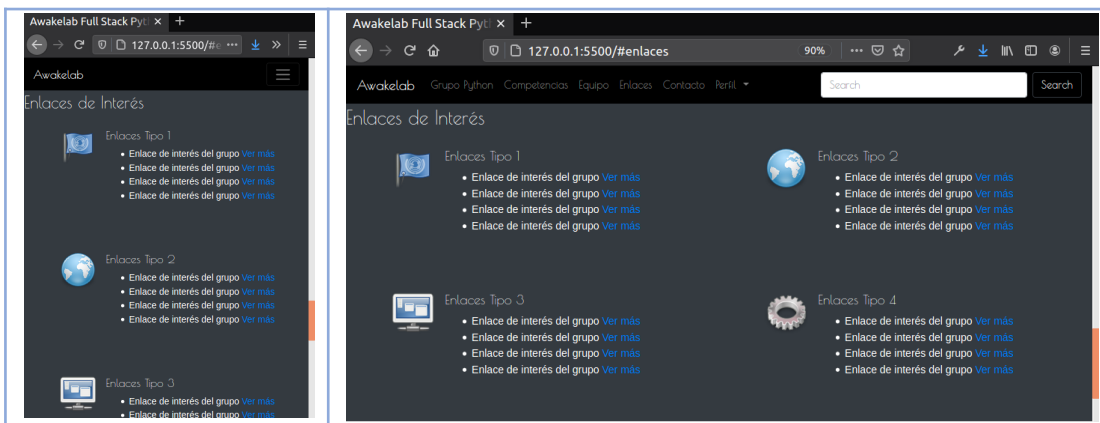
```

#enlaces {
  color:white;
  padding-top: 60px;
  min-height: 100%;
}

.media {
  margin: 20px 5% 50px 10%;
}

```

Así, obtenemos el siguiente resultado:



FORMULARIOS, BTN-GROUP, BTN

Utilizaremos la sección “Contacto” de nuestra página para abordar los componentes **form-group** y **btn**. En primera instancia podría pensarse que son una réplica de elementos **<form>** y **<button>** de HTML. Sin embargo, al igual que los otros componentes que hemos revisado, estos componentes Bootstrap permite simplificar mucho la especificación de estilo de formularios y botones. Para ilustrar su uso, implementaremos un formulario de contacto para nuestra página. Agregamos un elemento **<div>** de clase **row** en el que incorporaremos un elemento de grilla centrado de ancho 6. Dentro de éste colocaremos los componentes **form-group** y **btn-group**.

El código para esto corresponde al siguiente:

```
<div class="row justify-content-center align-items-center h-100">
  <div class="col-sm-6" id="contact-form">
    <form>
      <div class="form-group">
        <label for="contact-name">Nombre</label>
        <input type="text" class="form-control" id="contact-name"
          placeholder="Tu Nombre">
      </div>
      <div class="form-group">
        <label for="contact-email">Email</label>
        <input type="email" class="form-control" id="contact-email"
          placeholder="Tu Email">
      </div>
      <div class="form-group">
        <label for="contact-msg">Email</label>
        <textarea class="form-control" id="contact-msg"
          placeholder="Tu Mensaje" rows="4"></textarea>
      </div>
    </form>
  </div>
</div>
```

```
<div class="btn-group">
  <button type="button"
    class="btn btn-outline-secondary">Enviar</button>
  <button type="button"
    class="btn btn-outline-secondary">Limpiar</button>
</div>
</form>
</div>
</div>
```

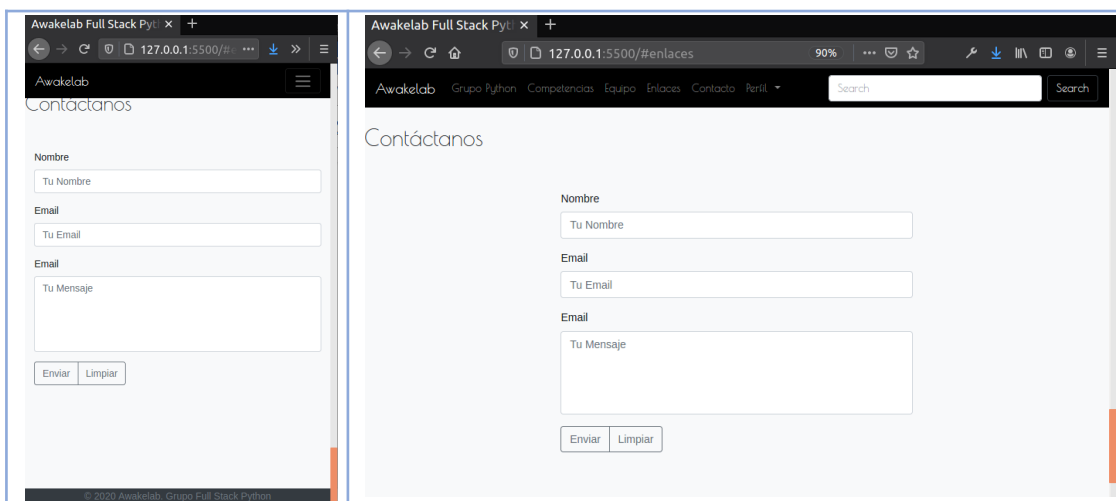
Y agregamos algunos ajustes de estilos para acomodar de mejor forma el componente a nuestra página. El área de mensaje la configuramos para que no pueda ser redimensionada por el usuario. Aprovechamos de incorporar aquí un ajuste adicional para el footer de nuestra página agregando la clase **text-center** a éste en HTML y con **display:inline** en nuestra hoja de estilos CSS. Las reglas CSS mencionadas quedarán de la siguiente manera:

```
#contact-form {
  padding-top: 3rem;
}

textarea {
  resize: none;
}

.footer p {
  display: inline;
}
```

De esta forma el resultado de todos los últimos cambios se traduce en el siguiente resultado visual:



1.4.1.6. Documentación de Bootstrap 4.

Bootstrap provee una gran documentación, y la correspondiente a la **versión 4.5.0** que hemos revisado en este documento está disponible en: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

1.4.2.- Referencias

[1] Aravind Shenoy, CSS Framework Alternatives, 2018.

[2] Build fast, responsive sites with Bootstrap.
<https://getbootstrap.com/>

[3] Benjamin Jakobus, Mastering Bootstrap 4, 2018.

[4] Bootstrap 4 Tutorial
<https://www.w3schools.com/bootstrap4/>

[5] Bootstrap 4 Cheat Sheet
<https://hackerthemes.com/bootstrap-cheatsheet/>