



DGTIC UNAM

DIRECCIÓN GENERAL DE CÓMPUTO Y
DE TECNOLOGÍAS DE INFORMACIÓN
Y COMUNICACIÓN

Taller de Bootstrap



Edgar Vargas Zermeño
Octubre 2022



Visibilidad Web
UNAM

Bootstrap

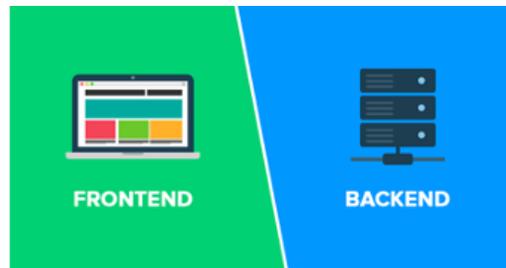


Conociendo Bootstrap

Como sabemos en el diseño de software se separan dos tipos de abstracción para una aplicación web, front-end y back-end.

El front-end es la parte que interactúa con el usuario, es donde se permitirá la entrada de información mientras que el back-end procesará dicha información devolviendo alguna respuesta de nuevo al front-end.

Diseño web
HTML
CSS
JAVASCRIPT



Servidor y Base de Datos
PHP
SQL

Ahora bien, si te dijéramos que **Bootstrap** es un framework de diseño web, ¿a qué parte lo asociarías, front-end o back-end?

Bootstrap es un framework que inicio como un proyecto de twitter en respuesta diferentes problemas de diseño web, las cuales implicaban tiempo y esfuerzo en solucionar. Fue liberado al público por el año 2011 como un proyecto Open Source en GitHub, convirtiéndose actualmente en el framework de front-end más popular.



Cuenta con elementos preconfigurados para tipografía, formularios, botones, tablas y otros [componentes](#) de interfaz como menús, ventanas, alertas, etc.

¿Qué es un framework?

Conocemos como framework o marco de trabajo dentro del desarrollo de software, al entorno que nos hará más sencilla la programación y el diseño. En este último caso las actividades como maquetar y dar una mejor vista a la interfaz y a sus elementos resultará más rápido de realizar sin hacer mucho esfuerzo, pero a costa de seguir las reglas del framework.

Se podría decir que pierdes ciertas “libertades” al quedar “sometido” a su diseño para ganar eficiencia en el desarrollo. Finalmente depende de cada uno saber como sacarle potencial a este framework, pues siempre será posible modificar la apariencia si sabes que partes modificar.



Diseño responsivo

Bootstrap está enfocado en dos paradigmas, el primero de ellos es el Diseño responsivo que trata de como se adaptará tu aplicación web no solamente en el monitor de tu computadora sino también a los diferentes dispositivos móviles que existen en el mundo, como smartphones, tabletas y todo aquellos otros móviles que puedan correr un navegador eso incluye también computadoras con monitores enormes.



Mobile First

Respecto al segundo paradigma de diseño Mobile First, como su nombre lo indica “Primero los Móviles” es una metodología de diseño que está en función de la resolución que presenta un móvil, es decir vas a maquetar tus elementos web, ya sean tablas, botones, listas, formularios, etc; pensando primero en el desarrollo de una aplicación para móviles para que posteriormente se facilite el maquetado en los demás dispositivos con mayor resolución.



Bootstrap

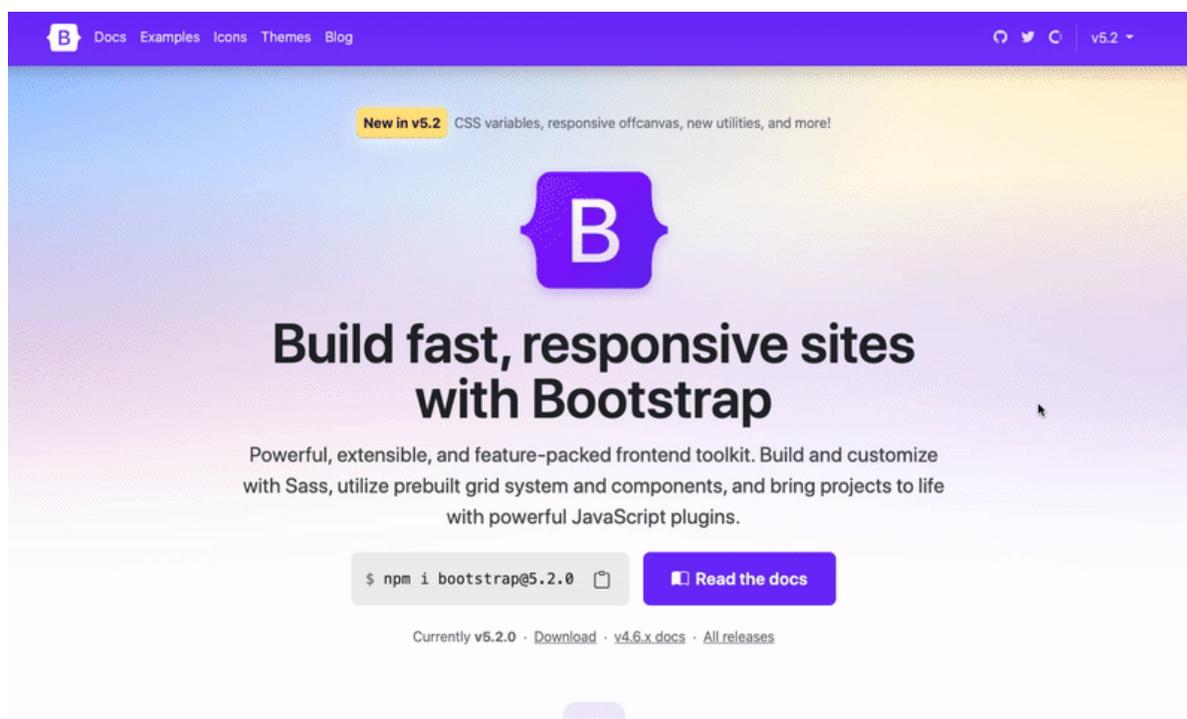


Primeros pasos con Bootstrap

Bootstrap tiene su sitio oficial donde puedes encontrar toda la documentación del framework para iniciar con él:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Existen diferentes formas de integrar **Bootstrap** en tu proyecto, en la animación se muestra una de ellas que consiste en utilizar una red de distribución de contenidos (CDN).



Bootstrap está desarrollado con los lenguajes HTML y CSS, con opción de incorporar [componentes](#) dependientes de Javascript, tecnologías que hay que referenciar desde los archivos en que queremos ocupar **Bootstrap**.

En la [Actividad 2](#) agregaremos esas referencias y la línea para integrar **Bootstrap** a un archivo html.



Bootstrap



Actividad 2

PASO 1/3

1. Descarga el archivo [01_inicio.html](#) y ábrelo en tu editor de código preferido
2. En atributo **content** de la etiqueta **meta description** escribe tu nombre
3. Ve al sitio de [Bootstrap](#) en la sección [Get started](#)
4. Pega la etiqueta **<link>** dentro de **<head>** para insertar los estilos CSS
5. Después pega la o las etiquetas **<script>** que se indican en la documentación antes del cierre **</body>**
6. Salva y prueba los cambios en tu navegador.

* Las imágenes son una muestra, el sitio Bootstrap cambiará su apariencia según se use la versión 4 o 5

The screenshot shows the Bootstrap 5.2.0 documentation page. The navigation bar includes links for Docs, Examples, Icons, Themes, and Blog, along with a version selector set to v5.2. The main content is divided into two columns:

- Install via package manager:** This section explains how to install Bootstrap's source files using npm, RubyGems, Composer, or Meteor. It includes code blocks for:


```
$ npm install bootstrap@5.2.0
```

```
$ gem install bootstrap -v 5.2.0
```
- Include via CDN:** This section provides instructions for including Bootstrap's compiled CSS or JS from a CDN. It includes code blocks for:


```
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/
```

```
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist
```

Below these sections, there is a section titled "Read our getting started guides" with icons for Webpack, Parcel, and Vite.

PASO 2/3

Debemos asegurarnos de que nuestro código tenga establecidos los estándares de diseño y desarrollo, esto incluye el uso de una meta etiqueta viewport para obtener un comportamiento responsivo adecuado.

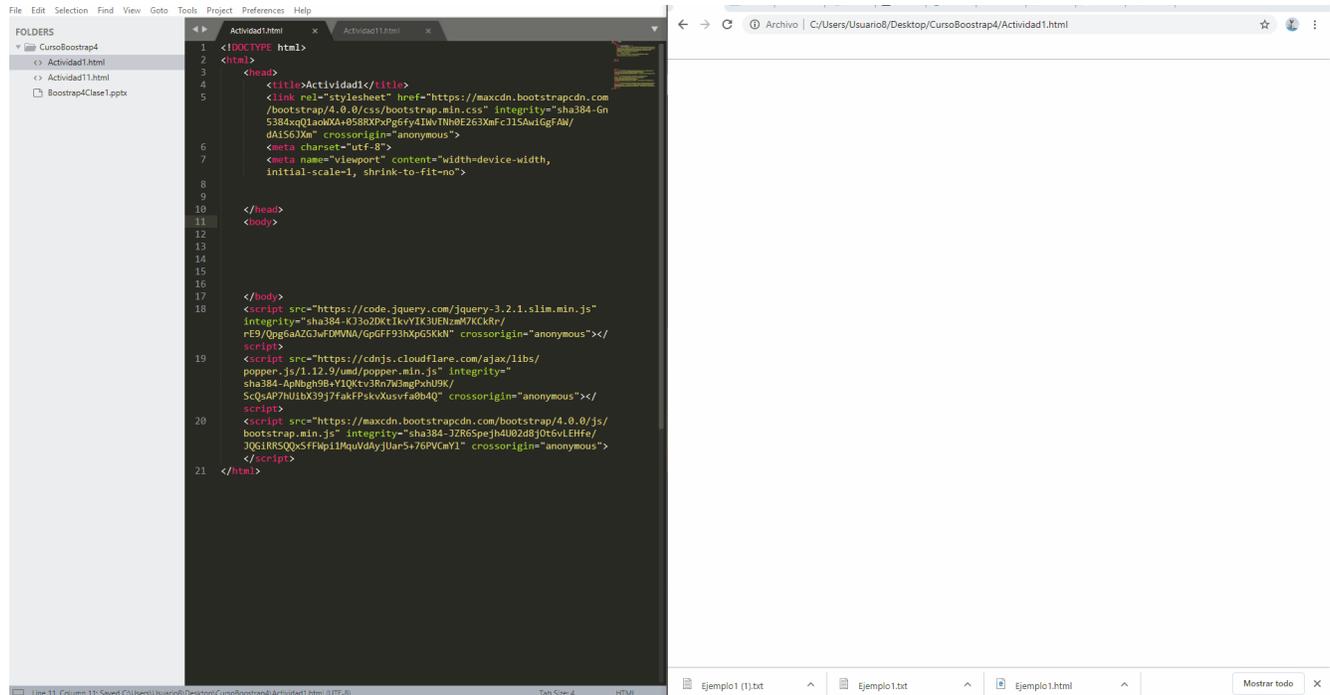
Si esta etiqueta no se agrega de forma automática en tu editor de código HTML, la puedes copiar directamente de aquí:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

PASO 3/3

Para saber si se ha integrado bien **Bootstrap** una vez hechos los pasos anteriores la fuente del texto a cambiado de la fuente de default de html al formato definido por el framework como se ve en la animación.

Al terminar sube el archivo de tu **Actividad 2**



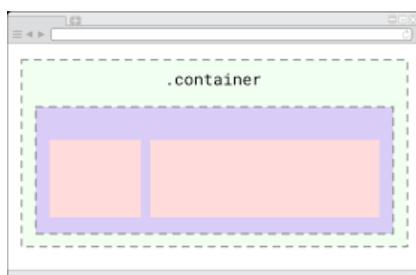
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Actividad1</title>
5 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com
  /bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn
  5384xqQ1aoWA+05SRXp66fy4Iw7TNh0E263Xmfc115AwigGfAM/
  dA1567Xm" crossorigin="anonymous">
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width,
  initial-scale=1, shrink-to-fit=no">
8
9
10 </head>
11 <body>
12
13
14
15
16
17 </body>
18 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
  integrity="sha384-k1kyy7QQZyrlrozy58mLs48ZrS7QP06QC28W64ZK763GhX
  rE9/0p66AZG1wFDMVA/0p6FF93hXpG5KkN" crossorigin="anonymous"></
  script>
19 <script src="https://cdnjs.cloudflare.com/ajax/libs/
  popper.js/1.12.0/umd/popper.min.js" integrity="
  sha384-qpph98419kr3607739AgP29BJR/
  SQ0AP7N11bX39j7fakFpskXusvfa0b40" crossorigin="anonymous"></
  script>
20 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/
  bootstrap.min.js" integrity="sha384-3265pe3jh40020b100v1Ehr/
  JQ6iRR5QxSFQp11MquVdAyjUar5+76PVCaY1" crossorigin="anonymous">
  </script>
21 </html>
```

Bootstrap



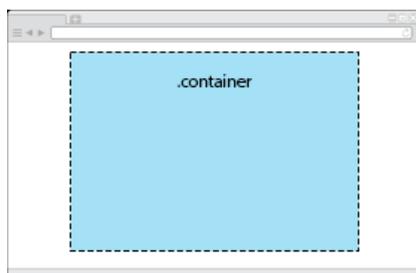
Containers

Bootstrap tiene elementos y opciones para empezar a construir tu diseño de interfaz, uno de esos elementos son los contenedores. Estos te permiten generar la o las "cajas" que anidarán los contenidos de nuestras páginas. Los contenedores son los elementos más básicos en el framework y serán requeridos cuando usemos el sistema de grid.

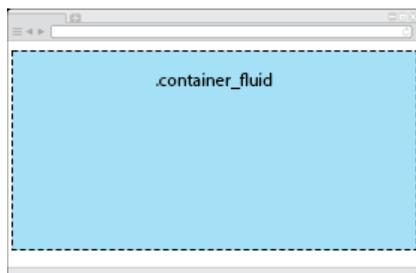


Contenedores básicos:

- **.container** provee un contenedor que permite tener el contenido de la pantalla de un ancho fijo (preestablecido por default).



- **.container-fluid** provee un contenedor que permite tener el contenido extendido al máximo del tamaño de la pantalla.



Estructuración de contenedores

Bootstrap usa selectores de clase como medios para transmitir sus estilos, en este ejemplo: `class="container"`.

La etiqueta `<div>` (división) se utiliza de forma habitual para estructurar el diseño y para aplicarles posteriormente una de las dos clases antes mencionadas.

En este ejemplo también se usó el atributo `style` para pintar el contenedor de color tomato y de esta manera visualizar mejor su alcance.

The image shows a code editor on the left and a browser window on the right. The code editor displays HTML code for a Bootstrap page. A `<div style='background-color: tomato; padding: 10px; text-align: center; color: white; font-weight: bold; margin: 10px 0; min-height: 100px; min-width: 100%;'>` container is used to wrap the main content. The browser window shows the rendered page with the text 'Hola mundo' and two paragraphs of Lorem Ipsum text, all contained within the pink container.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Actividad1</title>
5 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0/css/bootstrap.min.css" integrity="
sha384-I4q80U48LHYj9zfke1cnh017782e9662373b0FFdI14b9z3f9ae0eb9f66275630" crossorigin="anonymous">
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
8
9
10 </head>
11 <body>
12
13 <div style='background-color: tomato; padding: 10px; text-align: center; color: white; font-weight: bold; margin: 10px 0; min-height: 100px; min-width: 100%;'>
14 <h2>Hola mundo </h2></div>
15
16 <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">
17 <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.</p>
18
19 <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.</p>
20
21 </div>
22 </body>
23 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-kIX3026Ycr6KH8KHdPV6dQKAzoN5O40q3l7Yg99Lp08g97H8178zgk2z73e" crossorigin="anonymous"></script>
24 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.5/umd/popper.min.js" integrity="
sha384-ggkYqnDIDMGT8T7YYF3UKYJM30xKRKO" crossorigin="anonymous"></script>
25 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/
bootstrap.min.js" integrity="sha384-J8X4vInw31d7q87l28c88C40v7/08/O784oMnVVk8zruC6010g6617782e9662373b0FFdI14b9z3f9ae0eb9f66275630" crossorigin="anonymous"></script>

```

!REGLA IMPORTANTE!

Los contenedores NO deben estar anidados, es decir:

NO se debe anidar un `.container` o `.container-fluid` en otro `.container` o en otro `.container-fluid`.

Bootstrap



Actividad 3

Creas un archivo con el nombre **contenedores.html**, integra **Bootstrap** adecuadamente y realiza lo siguiente:

- Crea cuatro contenedores que estructuren de forma general e inicial tu página web, utiliza las etiquetas semánticas para estructurar tu página: **<main>**, **<article>**, **<footer>**, **<header>**, etc;
- Escoge una paleta de cuatro colores para dicha tarea, por ejemplo: un color para el body, otro para el header, etc; para esto deberás crear tu propio archivo "estilos.css" donde especificarás los selectores para cada elemento.
- No olvides poner tu nombre en el **content** de la etiqueta **meta description**

Ejemplo:

```

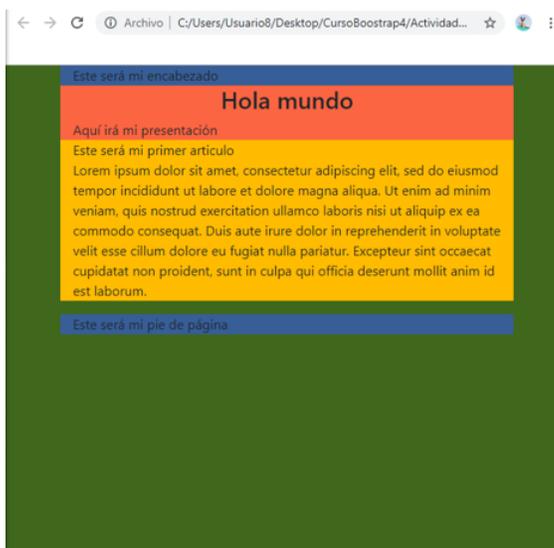
1  article {
2      background: #FFBB00
3  }
4  main {
5      background: #FB6542
6  }
7
8  footer{
9      background: #375E97
10 }
11 header{
12     background: #375E97
13 }
14
15 body{
16     background: #3F681C
17 }
18

```

```

<!DOCTYPE html>
<html>
  <head>
    <title>Actividad1</title>
    <link rel="stylesheet" href="
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/
bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA
+058RXPxPg6fy4IWvTNh0E263XmFcJlSAw1GgFAW/dAiS6JXm"
crossorigin="anonymous">
    <link rel="stylesheet" href="miEstilo.css">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
  </head>
  <body>
    <header class="container">
      Este será mi encabezado
    </header>
    <main class="container">
      <center><h3>Hola mundo </h3></center>
      Aquí irá mi presentación
    </main>

```

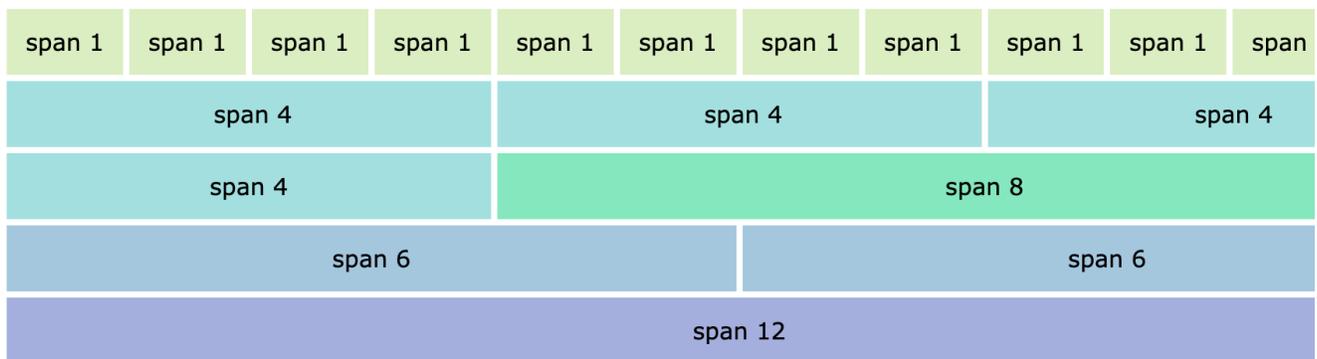


Bootstrap



Sistema de rejilla de Bootstrap

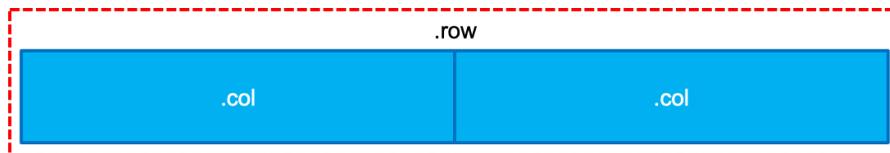
Bootstrap usa un sistema de rejillas flexibles o grids que se pueden utilizar para posicionar las áreas o secciones de la página y todo elemento de interfaz de usuario. Es posible disponer desde una columna que abarque 12 fragmentos del 100% de ancho y todas sus combinaciones en fragmentos de 12.



En un sistema de grid no necesitas pensar en el ancho que ocupa un elemento respecto a los demás. Sólo debes decidir "cuántas columnas del grid ocupa dicho elemento".

Reglas para generar columnas

Para crear grupos de columnas horizontales utilizamos filas mediante la clase `.row`. Para crear columnas sencillas y del mismo tamaño utilizamos la clase `.col`



Los contenidos deben estar dentro de columnas y sólo columnas pueden ser hijos inmediatos de filas. No se recomienda tener un `.col` directamente dentro de otro `.col`, ni tampoco tener un `.row` directamente dentro de otro `.row`. Eso causaría resultados inesperados que desbordarían el ancho de pantalla.

```

<div class="container">
  <div class="row">
    <div class="row">
      <div class="col">
        <div class="col">
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

✗

```

<div class="container">
  <div class="row">
    <div class="col">
      <div class="row">
        <div class="col">
          <div class="col">
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

✓

Para hacerlo de forma correcta solamente tenemos que crear un nuevo .row dentro de un .col, y dentro de esta nueva fila podremos subdividirla usando también hasta 12 columnas.

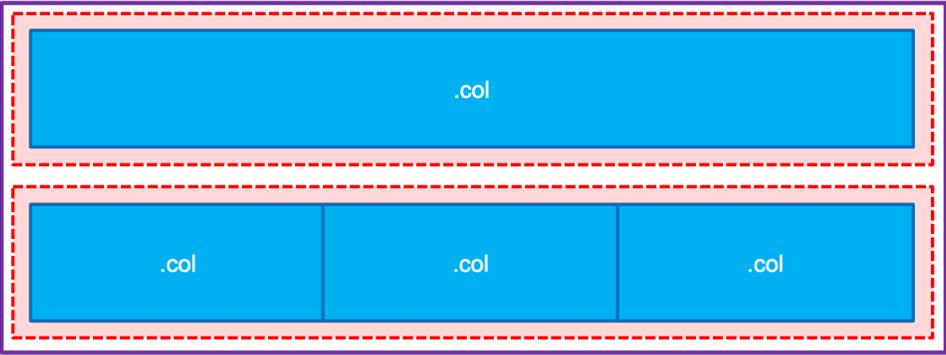
Estructuración básica con columnas

Cuando usamos la clase .col, las columnas que estén dentro del .row siempre tendrán el mismo ancho porcentual, es decir que si en una fila hay 2 columnas cada una ocupará el 50%, si ponemos 3 ocuparan el 33.33% y así sucesivamente según fragmentemos.

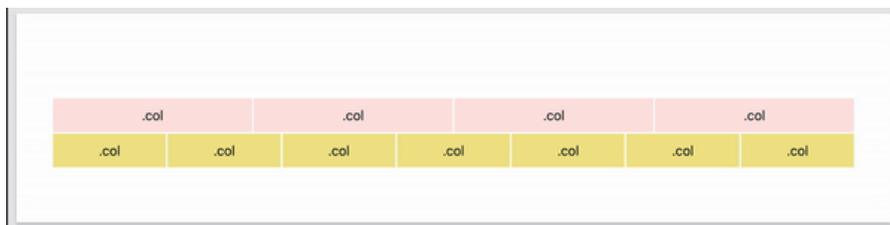
```

<div class="container">
  <div class="row">
    <div class="col"></div>
  </div>
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
</div>

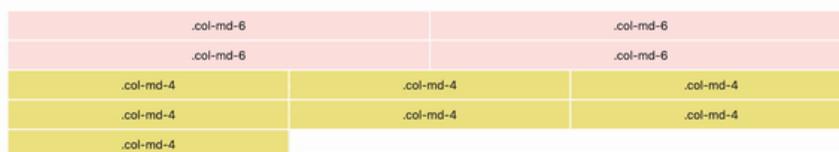
```



Al usar un solo elemento con la clase .col se ocupará el 100% de ancho, para indicar otros anchos deseados en nuestra distribución usamos .col-[ancho de columna]. El número de ancho de columna indica cuántos fragmentos respecto de 12 fragmentos abarcará la columna, donde asumiremos que al usar .col-6 estamos usando un ancho de 6 fragmentos de 12, es decir el 50% de ancho. Si solo usamos esta fórmula, las columnas siempre conservarán su estructura sin importar el tamaño del dispositivo en el que se visualizan.



Para modificar el comportamiento de acuerdo a la resolución del dispositivo en el que visualizamos las columnas utilizamos los infijos de clase. .col-[infijo]-[ancho de columna]



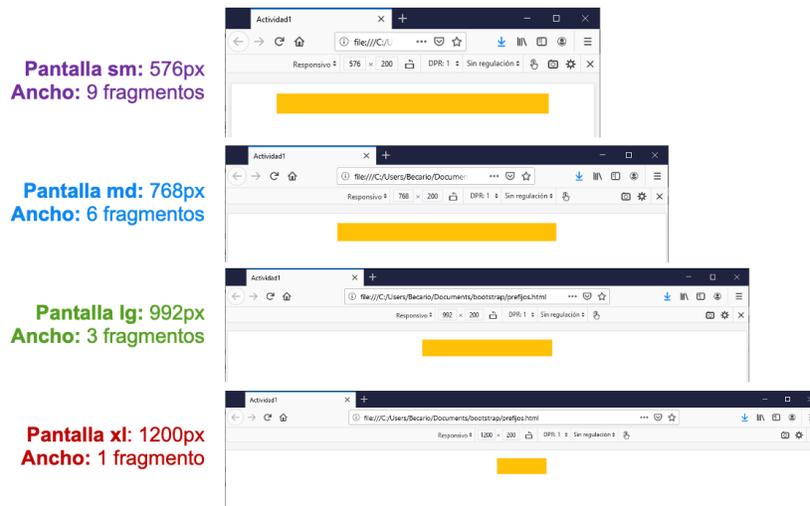
Dependiendo de la resolución en la que queramos trabajar, utilizaremos los siguientes infijos:

Punto de ruptura	Infijo de clase	Dimensiones
X-Small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Se pueden asignar todos los infijos a una sola columna, para que tenga un tamaño diferente en cada resolución.

```
<div class="col-sm-9 col-md-6 col-lg-3 col-xl-1"></div>
```

Se obtendría el siguiente resultado:



Pero NO es necesario asignar todos los infijos a cada columna.

Si a una columna se le asigna una medida con un infijo (por ejemplo: .col-md-6, donde el infijo es -md y la medida es -6), la columna tomará esa medida en todas las resoluciones mayores a la del infijo asignado.

Ejemplo:

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

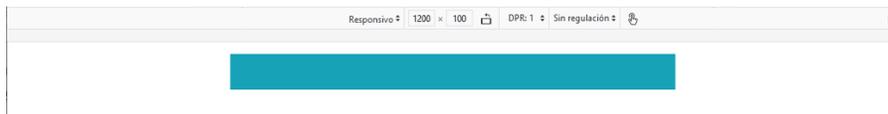
Sin indicar nada, en pantallas menores a 750px, 12 fragmentos de ancho



Indicando el infijo md: a partir de 768px, 6 fragmentos de ancho



Aún a los 1200px, de no indicar otro infijo, heredará lo indicado por md



Al modificar el ejemplo anterior, si se agrega otra clase con infijo a la columna, el infijo menor indicado abarcará desde su resolución hasta la resolución del nuevo infijo.

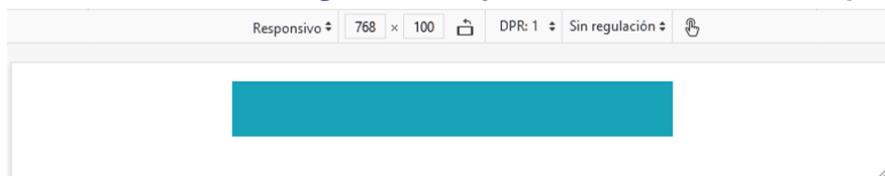
Ejemplo:

```
<div class="col-md-6 col-xl-10"> </div>
```

Sin indicar nada 12 fragmentos en pantallas de ancho < 768px



Indicando en md 6 fragmentos en pantallas con ancho >= 768px



Indicando xl/ 10 fragmentos en pantallas con ancho >= 1200px



Un ejemplo más:

Tenemos dos resoluciones con dos indicaciones diferentes: x-small (.col) y large (.col-lg-).

En anchos < 992px, la columna 1 y 2 tendrán tamaño de 6, pues se les aplica col-6, mientras que las columnas 3 y 4 tendrán tamaño de 12, pues se les ha indicado col-12.

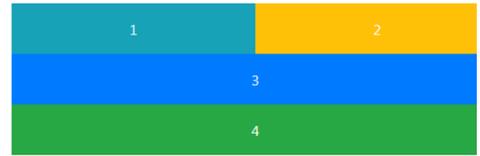
Cuando el ancho sea >=992px, todas las columnas tendrán un tamaño de 3 fragmentos, pues se aplica col-lg-3.

```

<div class="container">
  <div class="row">
    <div class="col-6 col-lg-3">1</div>
    <div class="col-6 col-lg-3">2</div>
    <div class="col-12 col-lg-3">3</div>
    <div class="col-12 col-lg-3">4</div>
  </div>
</div>

```

< 992px



>= 992px (lg)

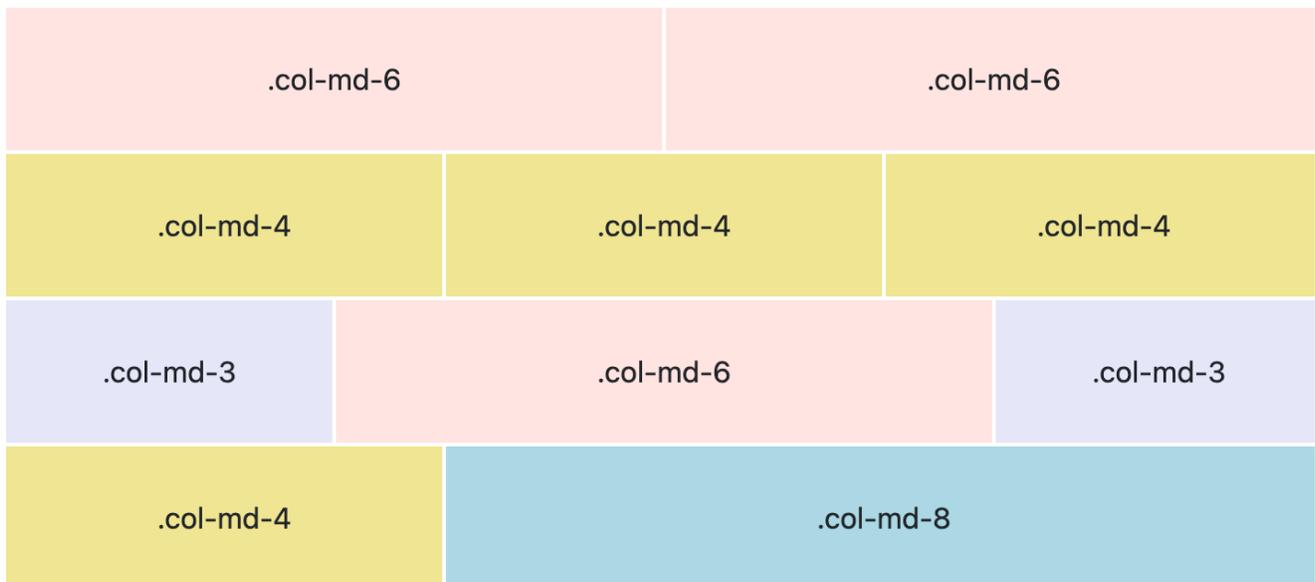


Bootstrap



Actividad 4

1. Genera esta estructura con el sistema de rejillas de **Bootstrap**.
2. Coloca tu nombre en el *content* de la etiqueta *meta description*
3. Sube tu archivo.



Bootstrap

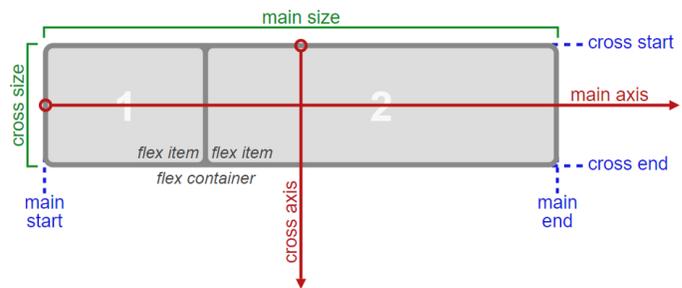


Flexbox en Bootstrap: maquetación básica

Una de las características de **Bootstrap** es que puede implementar maquetación utilizando Flexible Box, el módulo CSS que nos permite crear estructuras responsivas sin usar las propiedades *position*, *float* y *clear*. Para entender la maquetación en **Bootstrap** lo primero será analizar varios conceptos de Flexbox y sus relaciones:

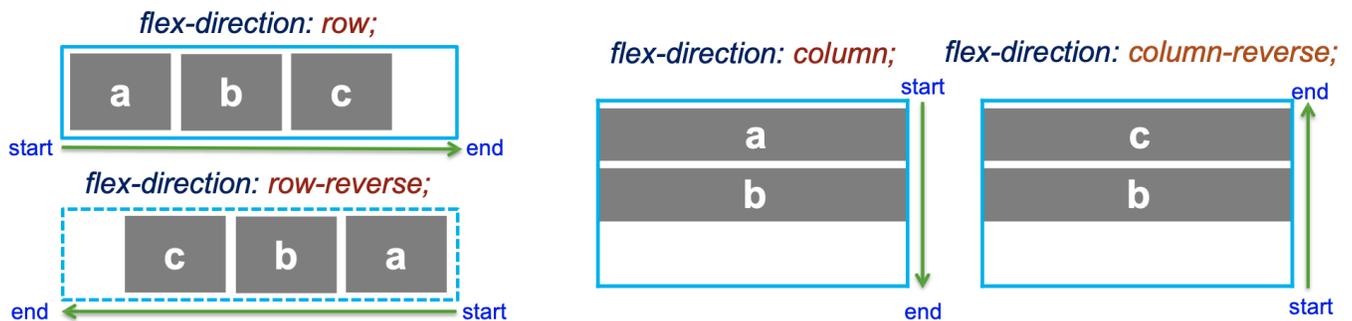
Una vez declarado un *flex-container* los objetos contenidos serán *flex-items* que se organizarán sobre dos ejes *main-axis* y *cross-axis*.

Cada eje tiene de forma natural un inicio y fin (*start* y *end*) que se corresponde a los límites de ancho y alto del contenedor.



En **Flexbox** la indicación de filas y columnas se realiza con la propiedad *flex-direction* sobre el *main axis* o eje principal. A la vez que se indica acomodo en filas (*row*) o en columnas (*column*), se puede invertir el sentido del direccionamiento del eje principal.

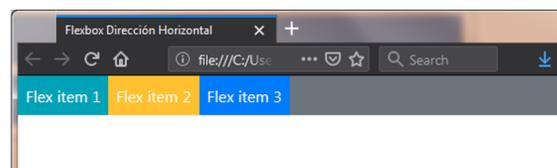
Se muestran a continuación los cuatro valores para la propiedad *flex-direction*:



Contenedor flex en Bootstrap

De forma implícita todos los *.row* son contenedores flex, pero si queremos hacer otra implementación usamos la clase *.d-flex* en un cualquier elemento para volverlo *flex-container*, automáticamente convierte en *flex-items* a los elementos que se encuentren dentro de él como sucede con los *.col* en un *.row*.

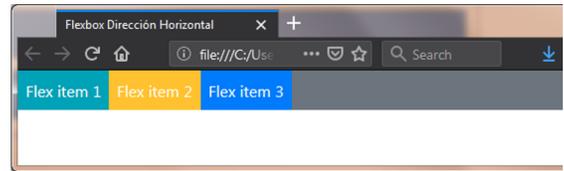
```
<div class="d-flex bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
```



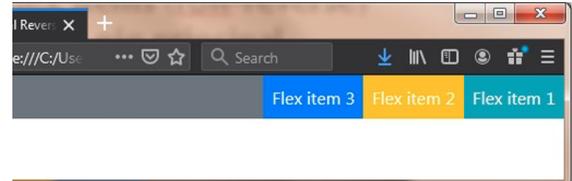
Dirección horizontal en Bootstrap

Para mostrar los **flex ítems** en dirección horizontal sobre el **main axis** se utiliza la clase **.flex-row** (comportamiento por defecto). Para invertir el orden de los **flex ítems** se usa la clase **.flex-row-reverse**, lo cual invertirá el sentido horizontal de lectura y colocará el **start** a la derecha y el **end** a la izquierda.

```
<div class="d-flex flex-row bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



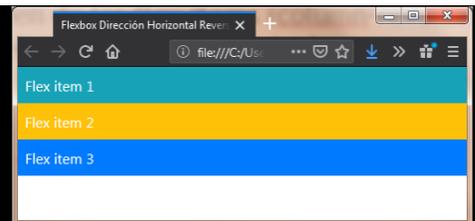
```
<div class="d-flex flex-row-reverse bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



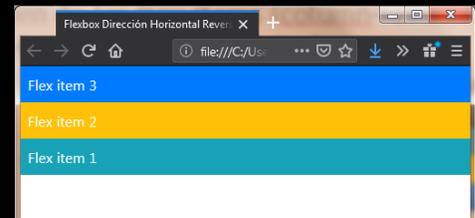
Dirección vertical en Bootstrap

Para mostrar los **flex ítems** en dirección vertical se utiliza la clase **.flex-column**. El contenedor flex pasará el **main axis** a posición vertical y el **cross axis** a horizontal. Para invertir el orden de los flex ítems se usa la clase **.flex-column-reverse**, lo cual invertirá el sentido vertical de lectura.

```
<div class="d-flex flex-column bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



```
<div class="d-flex flex-column-reverse bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



Bootstrap



Flexbox en Bootstrap: márgenes y alineación

Alineación horizontal de elementos

Hemos logrado direccionar los elementos de forma horizontal y vertical, ahora vamos a alinear los elementos en el espacio horizontal disponible. Se utilizan clase `.justify-content-[sufijo]` con una indicación de cómo distribuir los elementos.

`.justify-content-start` alinea los elementos al inicio del eje principal.

`.justify-content-end` alinea los elementos al final del eje principal.

`.justify-content-center` alinea los elementos al centro del eje principal.

`.justify-content-between` alinea los elementos ocupando el total de ancho del contenedor.

`.justify-content-around` alinea los elementos dejando el mismo espacio junto a cada elemento.

```
<div class="d-flex justify-content-start bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
```



```
<div class="d-flex justify-content-end bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
```



```
<div class="d-flex justify-content-center bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
```



```
<div class="d-flex justify-content-between bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
```



```
<div class="d-flex justify-content-around bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



End y Start en el eje principal

Es importante tomar en cuenta que los elementos se van a alinear con respecto al eje principal y no al contenedor, por lo que si aplicamos `row-reverse`, el valor `justify-content-start` alinearán los ítems al lado derecho puesto que nosotros indicamos con `reverse` que en el eje principal `start` es a la derecha y `end` a la izquierda.

```

<div class="d-flex flex-row-reverse justify-content-start
bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
    
```

Justify content en flex-column

También es importante tomar en cuenta que en el caso de *.flex-column*, se alinearán de manera vertical, pues ha cambiado la dirección del eje principal a vertical.

```

<div class="d-flex flex-column justify-content-end
bg-secondary text-white" style="height:200px">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div><br>
    
```

Fill y Grow

Para indicar de forma explícita el mismo ancho de todos los ítems se usa en cada ítem la clase *.flex-fill* (comportamiento por defecto).

```

<div class="d-flex bg-secondary text-white">
  <div class="p-2 flex-fill bg-info">Flex item 1</div>
  <div class="p-2 flex-fill bg-warning">Flex item 2</div>
  <div class="p-2 flex-fill bg-primary">Flex item 3</div>
</div><br>
    
```

También podemos hacer que un ítem ocupe el espacio sobrante de un contenedor, utilizamos la clase *.flex-grow-1* en el ítem que deseamos que incremente su tamaño.

```

<div class="d-flex mb-3">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 flex-grow-1 bg-primary">Flex item 3</div>
</div>
    
```

Uso de márgenes automáticos

Se pueden agregar márgenes automáticos mediante las clases *.mr-auto* (para empujar los ítems hacia la derecha) y *.ml-auto* (para empujar los ítems hacia la izquierda). Como podemos observar en el ejemplo, los empuja desde el ítem en el que se coloca la clase.

```

<div class="d-flex mb-3 bg-secondary text-white">
  <div class="p-2 mr-auto bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
    
```

```

<div class="d-flex mb-3 bg-secondary text-white">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning">Flex item 2</div>
  <div class="p-2 ml-auto bg-primary">Flex item 3</div>
</div>
    
```

Alineación en cross axis o eje secundario

Se utiliza la clase `.align-items-[sufijo]` para alinear los **flex items** de una fila simple (row) con respecto al eje secundario.

Sufijos posibles para el eje secundario:

- **start**: Alinea todos los flex ítems al inicio del eje secundario.
- **end**: Alinea todos los flex ítems al final del eje secundario.
- **center**: Alinea todos los flex ítems al centro del eje secundario.
- **baseline**: Ajusta todos los flex ítems de manera que su líneas base queden alineadas.
- **stretch (default)**: Alinea todos los flex ítems incrementando su largo para que ocupen todo el espacio que tengan dentro del contenedor.

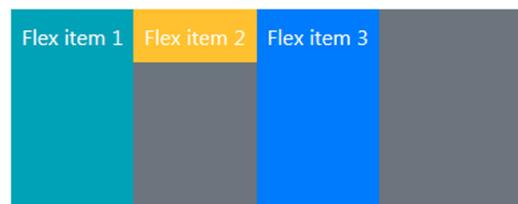
Es importante mencionar que el espacio vertical del contenedor flex debe ser suficientemente mayor a la altura de los elementos para notar los cambios con claridad.

<pre><div class="d-flex flex-row align-items-start bg-secondary text-white" style="height:150px"> <div class="p-2 bg-info">Flex item 1</div> <div class="p-2 bg-warning">Flex item 2</div> <div class="p-2 bg-primary">Flex item 3</div> </div></pre>	→	
<pre><div class="d-flex flex-row align-items-end bg-secondary text-white" style="height:150px"> <div class="p-2 bg-info">Flex item 1</div> <div class="p-2 bg-warning">Flex item 2</div> <div class="p-2 bg-primary">Flex item 3</div> </div></pre>	→	
<pre><div class="d-flex flex-row align-items-center bg-secondary text-white" style="height:150px"> <div class="p-2 bg-info">Flex item 1</div> <div class="p-2 bg-warning">Flex item 2</div> <div class="p-2 bg-primary">Flex item 3</div> </div></pre>	→	
<pre><div class="d-flex flex-row align-items-baseline bg-secondary text-white" style="height:150px"> <div class="p-2 bg-info">Flex item 1</div> <div class="p-2 bg-warning">Flex item 2</div> <div class="p-2 bg-primary">Flex item 3</div> </div></pre>	→	
<pre><div class="d-flex flex-row align-items-stretch bg-secondary text-white" style="height:150px"> <div class="p-2 bg-info">Flex item 1</div> <div class="p-2 bg-warning">Flex item 2</div> <div class="p-2 bg-primary">Flex item 3</div> </div></pre>	→	

Autoalineación

Se puede controlar la alineación de un ítem específico con respecto al eje secundario con la clase `.align-self-[sufijo]`. La clase se agrega en el ítem que se desea alinear. Los valores de sufijo para esta clase son los mismos que para `.align-items-[sufijo]`

```
<div class="d-flex bg-secondary text-white" style="height:150px">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning align-self-start">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



```
<div class="d-flex bg-secondary text-white" style="height:150px">
  <div class="p-2 bg-info">Flex item 1</div>
  <div class="p-2 bg-warning align-self-center">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```



Infijos responsivos

En **Bootstrap**, todas las propiedades que se utilizan para **FlexBox**, tienen sus respectivas clases responsivas, con las cuales podemos modificar el comportamiento de las cajas flexibles en diferentes tamaños de pantalla.

Los asteriscos pueden sustituirse con los infijos **sm**, **md**, **lg**, **xl** o **xxl**:

- **Contenedores:** *.d-*-flex*
- **Dirección:** *.flex-*-row*, *.flex-*-row-reverse*, *.flex-*-column*, *.flex-*-column-reverse*
- **Alineación main axis:** *.justify-content-*-start*, *.justify-content-*-end*
- **Alineación cross axis:** *.align-items-*-start*, *.align-items-*-end*, *.align-items-*-center*
- **Autoalineación:** *.align-self-*-start*, *.align-self-*-end*, *.align-self-*-center*

Bootstrap

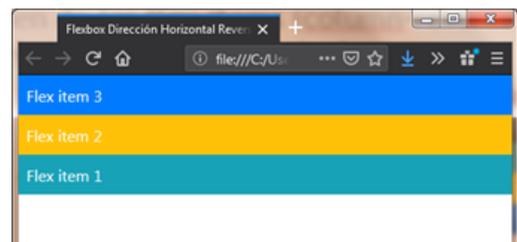
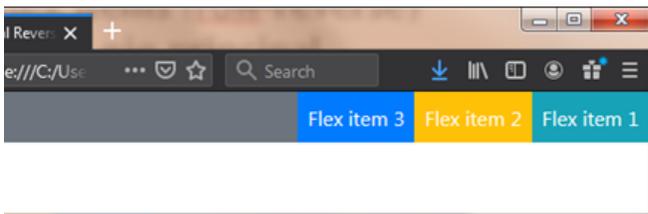
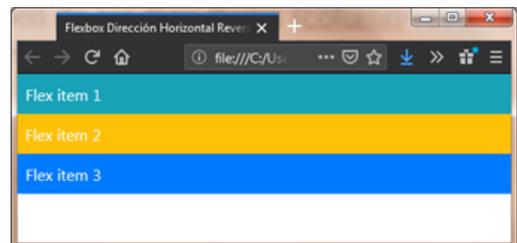
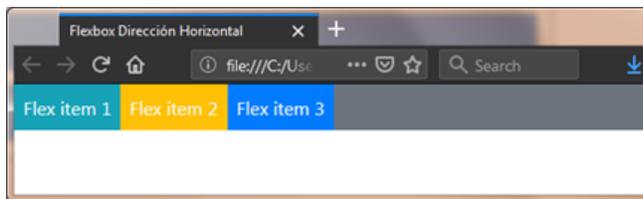


Actividad 5

1. Descarga los archivos [Flexbox.zip](#) y realiza los 3 ejercicios.
2. Anota tu nombre en el **content** la etiqueta **meta description** para cada uno.
3. Comprime y sube el archivo.

EJERCICIO 1/3

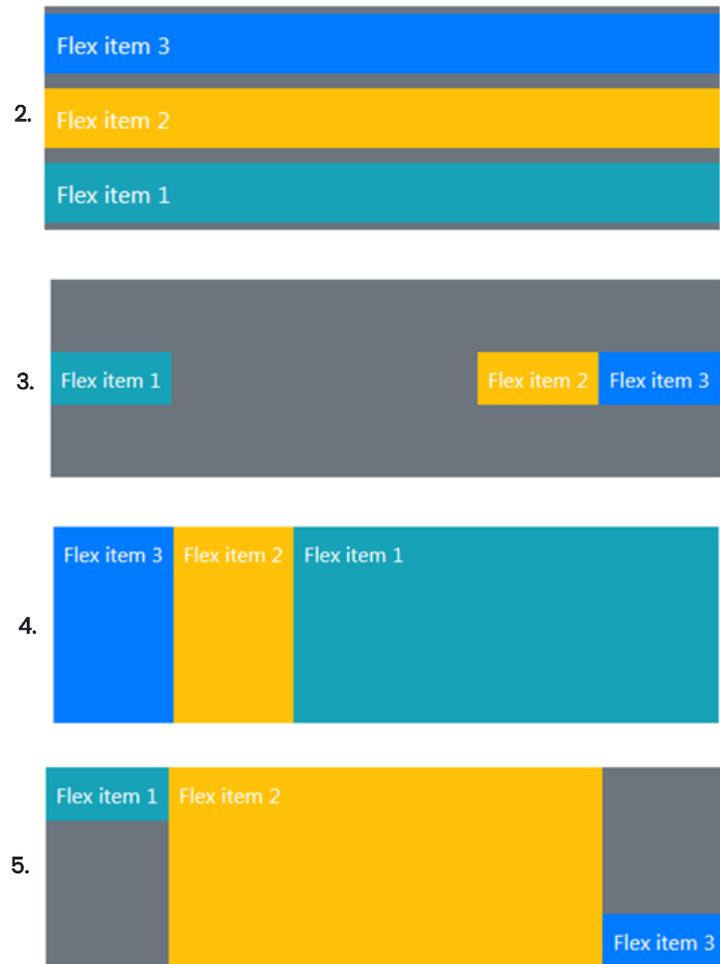
Utiliza las clases que has aprendido (`.d-flex`, `.flex-row*`, `.flex-column*`) en el archivo `FlexBox1.html` y modifícalo para obtener contenedores flex como los que se muestran a continuación:



EJERCICIO 2/3

Modifica el archivo `FlexBox2.html` utilizando las clases de **Bootstrap** para Flex Box (`.flex-row*`, `.flex-column*`, `.justify-content-*`, `.align-items-*`, `.flex-grow-1`, `.ml-auto`, `.mr-auto`) para obtener los siguientes resultados :

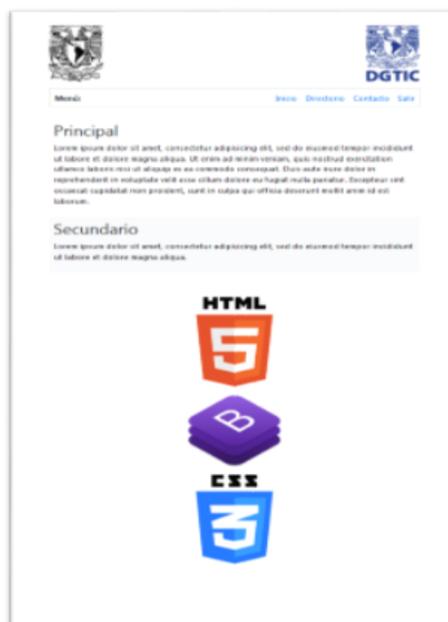
- 1.



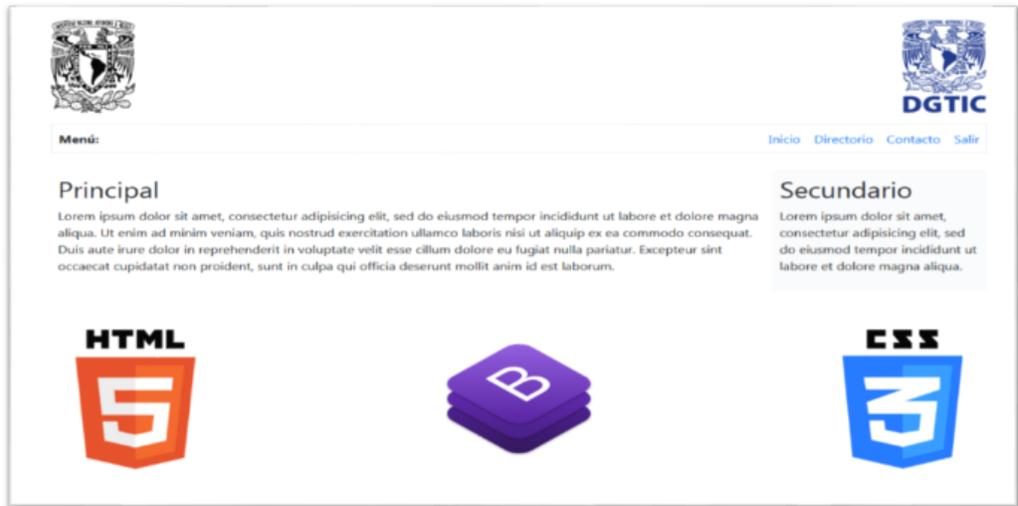
ARCHIVO 3/3

Modifica el archivo **FlexBox3.html** utiliza las clases para **Flex Box** de **Bootstrap** para obtener el siguiente maquetado:

Pantalla pequeña:



Pantalla grande (lg):



Bootstrap



Formato de texto, tablas y botones

Manejo de texto

Bootstrap prevé el uso de texto como se hace en cualquier entorno HTML, pero ha intervenido la apariencia de las etiquetas de manejo de texto para dar un diseño integral desde su uso básico. Además pone a nuestra disposición varias clases que nos ayudaran a su manejo.

TÍTULOS CON BOOTSTRAP

Título `<h1>`

Título `<h2>`

Título `<h3>`

Título `<h4>`

Título `<h5>`

Título `<h6>`

TÍTULOS CON HTML SIN BOOTSTRAP

Título `<h1>`

Título `<h2>`

Título `<h3>`

Título `<h4>`

Título `<h5>`

Título `<h6>`

Alineación de textos

Lo primero que consideraremos es la alineación de textos. Podemos utilizar clases correspondientes a las 4 alineaciones: left, right, center y justify; en combinación con los condicionamientos que **Bootstrap** emplea según el infijo de tamaño de pantalla: sm, md, lg, xl y xxl.

- Uso con prefijo **text** seguido de un guión, más el sufijo de la alineación deseada

text-center

- Infijo según tamaño de pantalla actual

text-md-center

Uso de **text-left**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sequi id minima omnis vero rerum illo blanditiis sed itaque asperiores aperiam reiciendis aliquid.

Uso de **text-justify**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sequi id minima omnis vero rerum illo blanditiis sed itaque asperiores aperiam reiciendis aliquid.

Uso de **text-center**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sequi id minima omnis vero rerum illo blanditiis sed itaque asperiores aperiam reiciendis aliquid.

Uso de **text-right**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sequi id minima omnis vero rerum illo blanditiis sed itaque asperiores aperiam reiciendis aliquid.

Flujo de texto

La cantidad de texto que manejamos no siempre se ajusta al tamaño de su contenedor y requerimos tomar decisiones. Mostramos 3 clases que ayudan controlar el comportamiento del texto.

El uso de word-break divide palabras largas

Parrafo con palabra larga supercalifragilisticospiralidoso que se divide

Al emplear text-wrap el texto fluye dentro del contenedor

Parrafo que ajusta el texto en el contenedor.

Al usar text-unwrap el texto desborda fuera del contenedor

Parrafo que desborda el texto

Manejo de tablas

Bootstrap proveerá un diseño básico que integra visualmente las tablas al entorno. Esta apariencia se implementa al incluir la clase **table** para cada uno de los elementos `<table>` que usemos. Puede ser modificada en 4 aspectos: color general, colores alternos, resaltado al paso del puntero y bordes. Cada característica debe añadir (además de la clase **table**) la palabra **table** más la **NuevaCaracteristica** unida por un guión intermedio tal como muestra este ejemplo:

```
<table class="table table-hover">
```

APARIENCIA DE TABLA USANDO **table**

#	Nombre	Apellido	Alias
1	Juan	Pérez	Jhon
2	Pedro	López	Pet
3	Roberto	Gomez	Bob

De igual manera para emplear colores predefinidos en una tabla usamos después de indicar la clase **table**, el prefijo **table** seguido de un guión más el sufijo de la apariencia deseada.

table-primary

APARIENCIA DE TABLA CON INDICACIÓN DE COLOR

#	Nombre	Apellido	Alias
1	Juan	Pérez	Jhon
2	Pedro	López	Pet
3	Roberto	Gomez	Bob

Definir colores alternados ayudará a la lectura de los datos por parte del usuario, se usa la clase **table-striped** Para que el usuario vea un resaltado al pasar el puntero por las filas de la tabla usamos **table-hover**

Ejemplo de tabla con resaltado y colores alternados:

#	Nombre
1	Juan
2	Pedro
3	Roberto

Las tablas en **Bootstrap** presentan bordes en horizontal que pueden ser alterados ya sea agregando todo borde posible con *table-bordered*, o por el contrario indicar que no haya ningún borde *table-borderless*.

TABLA SIN INDICACIONES DE BORDES

#	Nombre
1	Juan
2	Pedro
3	Roberto

TABLA CON INDICACIÓN DE BORDES

#	Nombre
1	Juan
2	Pedro
3	Roberto

TABLA CON INDICACIÓN DE ELIMINAR BORDES

#	Nombre
1	Juan
2	Pedro
3	Roberto

Uso de botones

Se ha previsto el uso de botones con estilos propios de **Bootstrap** para acciones y diálogos, con soporte para múltiples colores, tamaños y estados. Las indicaciones van como en otros elementos la clase **btn** para todo botón, más su modificación agregando enseguida el prefijo **btn** seguido de un guión y el sufijo de la apariencia que desea darse.

BOTON SIMPLE *btn*

Aceptar

CON COLOR *btn-success*

Success

CON BORDE *btn-outline-primary*

Primary

TAMAÑOS *btn-lg* Y *btn-sm*

Grande

Pequeño

DESHABILITADO *disabled*

Success

EN BLOQUE *btn-block*

BOTON EN BLOQUE

Para conocer las posibilidades de apariencia toma ideas de la referencia oficial de [Buttons Bootstrap](#)

Bootstrap



Actividad 6

Ahora es el momento de dar formato a la información al contenido de una página.

1. Descarga y abre el archivo [formateo.html](#) en un navegador y luego revisa el código en tu editor
2. Agrega tu nombre al **content** de la etiqueta **meta description**
3. Marca el texto "Alineación de texto en Bootstrap" como **<h1>**
4. Indica en las celdas del primer contenedor la clase que nos permita obtener 3 columnas en vez de celdas.
5. Indica **<h2>** para el título de cada columna, y **<p>** para el texto del párrafo
6. Escribe la clase correspondiente a la alineación de párrafo según menciona cada título
7. Para la tabla indica la clase que modifica y le da apariencia en **Bootstrap**, comprueba el cambio de apariencia
8. Agrega ahora la clase **table-dark** y observa los cambios
9. Copia el código de la tabla, pégalo debajo y modifica la clase usada para que aparezca de algún otro color
10. Agrega las clases correspondientes a color alternado y resaltada al pasar el puntero
11. Por último da un formato distinto cada uno de los botones. Recuerda que puedes usar la guía oficial de [botones de Bootstrap](#)
12. Sube la actividad

Bootstrap



Manejo de formularios

¿Cómo se manejan los forms en Bootstrap?

Los formularios son parte esencial del desarrollo de aplicaciones web, dentro de **Bootstrap** se tienen diferentes formas de diseñar y desplegar uno. **Bootstrap** (como en casi todos sus [componentes](#)) usa selectores de clases para lograr una presentación adecuada de formularios y sus elementos en navegadores y dispositivos.

Estaremos revisando las reglas que deben seguirse para los diferentes tipos de formularios para obtener una adecuada presentación. Lo primero que debes saber es que en **Bootstrap** existen diferentes tipos de formularios y para obtener el despliegue adecuada a nuestra necesidad deben seguirse distintas reglas según el tipo de formulario que deseemos utilizar.

Vertical form (formulario por default en Bootstrap)

Horizontal form **.form-horizontal**

Inline form **.form-inline**

Orden de clases para formularios

Para administrar medidas apropiadas de tamaño y espacio para etiquetas y controles, se deben aplicar dos estilos básicos:

.form-group sirve para agrupar en un conjunto el control y su etiqueta.

```

<body>
  <div class="container">
    <form>
      <div class="form-group">
        <label for="exampleInputEmail">Correo electrónico: </label>
        <input type="email" class="form-control" id="exampleInputEmail" aria-describedby="emailHelp" placeholder="Ingresa correo">
        <small id="emailHelp" class="form-text text-muted">No compartiremos tu correo con nadie.</small>
      </div>
      <div class="form-group">
        <label for="exampleInputPassword1">Contraseña: </label>
        <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Contraseña">
      </div>
    </form>
  </div>

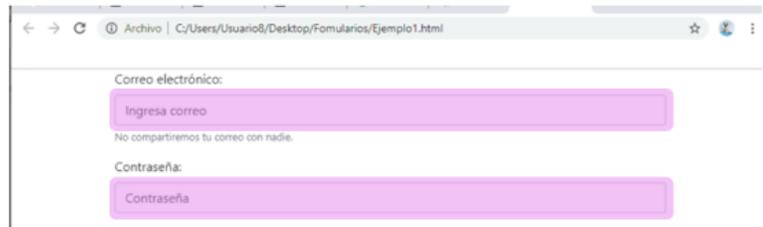
```

.form-control sirve para agregar el formato definido de Bootstrap para el control usado.

```

<body>
  <div class="container">
    <form>
      <div class="form-group">
        <label for="exampleInputEmail1">Correo electrónico: </label>
        <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Ingresa correo">
        <small id="emailHelp" class="form-text text-muted">No compartiremos tu correo con nadie.</small>
      </div>
      <div class="form-group">
        <label for="exampleInputPassword1">Contraseña: </label>
        <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Contraseña">
      </div>
    </form>
  </div>
</body>

```



Los estilos de **Bootstrap** permiten los formularios sean totalmente responsivos.

Formulario de orden vertical

De la misma manera que actúa en el sistema de rejilla el selector de clase `.row`, actúa el selector de clase `.form-row` que simplifica la generación de formularios con formato predefinido. La clase `form-row` es, por tanto, la combinación del uso del sistema grid con los estilos de formulario, es por ello que se logra el acomoda los elementos del formulario en columnas con un despliegue predefinido por nosotros.

Este ejemplo muestra el uso de `.form-row` y columnas en el código:

```

<body>
  <div class="container">
    <form>
      <div class="form-row">
        <div class="form-group col-sm-5">
          <label for="exampleInputEmail1">Email: </label>
          <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Email">
        </div>
        <div class="form-group col-sm-3">
          <label for="exampleInputPassword1">Password: </label>
          <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
        </div>
      </div>
      <center>
        <button type="submit" class="btn btn-primary">Enviar</button>
      </center>
    </form>
  </div>
</body>

```

Deja en un mismo renglón del formulario los grupos de etiqueta y control que están dentro del div

Número de columnas para cada grupo de etiqueta y control

En resumen:

- Un `.form-row` debe ir asignado a un elemento `<div>`.
- Dentro de ese divisor los `.form-group` necesarios en sus respectivos `<div>`.
- En ese mismo `<div>` se colocará el estilo correspondiente al ancho de columna, por ejemplo: `.col-md-3`.
- Por último, cada control debe tener asignada la clase `.form-control`.

Formulario de orden horizontal

Se pueden crear formularios horizontales también usando el sistema grid, solo añadiendo el estilo `.row` ahora dentro del divisor que llevará también el estilo `.form-group` e indicando que los elementos (tanto label como inputs) serán las columnas.

Para indicar un formulario horizontal es necesario tener en cuenta que:

- La clase `.row` debe ir en el mismo `<div>` que el `.form-group`, no se usa `form-row` en un `<div>` envolvente .
- La etiqueta `label` llevará 2 clases: `.col-form-label` y la indicación de ancho de columna, por ejemplo `.col-md-3`.
- Cada control debe ir dentro de un `<div>` con la indicación de ancho de columna requerido
- Al igual que en el formulario vertical la clase `.form-control` debe estar presente en cada control del formulario

```
<form>
  <div class="form-group row">
    <label for="exampleInputEmail1" class="col-sm-3 col-form-label">Email: </label>
    <div class="col-sm-9">
      <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Email">
    </div>
  </div>
  <div class="form-group row">
    <label for="exampleInputPassword1" class="col-sm-3 col-form-label">Password: </label>
    <div class="col-sm-9">
      <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
    </div>
  </div>
</form>
```

Formulario en línea (inline)

Los formularios en línea (inline), permiten desplegar los elementos del formulario en una sola fila horizontal:

Importante: Los controles solo aparecen con el estilo inline en pantallas que son al menos de 576px de ancho, debido a que las pantallas en los dispositivos móviles no pueden soportar el estilo.

Para generar un formulario inline necesitamos considerar:

- La etiqueta `form` llevará la clase `.form-inline`
- Los elementos de control deben tener la clase `.form-control`
- Los elementos `label` no llevan ninguna clase de **Bootstrap**
- No se usa `.form-group`. Al ser un formulario en línea sobrescribiría los estilos de espaciado con un resultado inesperado.

```
<body>
  <div class="container">
    <form class="form-inline" action="/action_page.php">
      <label for="nombre">Nombre:</label>
      <input type="text" class="form-control" placeholder="Ingrese nombre" id="nom">
    </form>
  </div>
</body>
```

Checkbox y radiobuttons

Existen clases que permiten crear grupos de elementos **checkbox** y **radio**. Similar a como ocurre con el selector de clase **.form-group** existe el selector de clase **.form-check**, que envuelve grupos de control y etiqueta para elementos checkbox y radio. Por otro lado, para los controles y etiquetas existen las clases **.form-check-input** y **.form-check-label**.

Para crear un grupo de elementos check se debe seguir:

- Agregar la clase **.form-check** al div que envolverá a los elementos de control y etiqueta.
- Los elementos de tipo control deben tener la clase **.form-check-input**
- Los elementos **label** deben tener la clase **.form-check-label**

```
<form>
  <div class="form-check">
    <input class="form-check-input" type="radio" name="radioEjemplo" id="radioEjemplo1">
    <label class="form-check-label" for="radioEjemplo1">
      Ejemplo 1
    </label>
  </div>
```

Por defecto, **.form-check** despliega los elementos verticalmente. Similar a como ocurre con el selector de clase **.form-inline** existe el selector de clase **.form-check-inline** para desplegar los elementos en una sola fila horizontal.

.form-check

- Ejemplo 1
- Ejemplo 2
- Ejemplo 3
- Ejemplo 4
- Ejemplo 5

.form-check-inline

- Ejemplo 1
- Ejemplo 2
- Ejemplo 3
- Ejemplo 4
- Ejemplo 5

Para crear un grupo de elementos check en línea, se debe agregar la clase **.form-check-inline** a cada **<div>** que envolverá a los elementos de control y su etiqueta.

```
<form>
  <div class="form-check-inline"> ☒
```

Bootstrap



Actividad 7

1. Descarga los el archivo [formularios.zip](#)
2. Descomprímelo y mira su contenido
3. Agrega en cada archivo tu nombre en el **content** de la etiqueta **meta description**
4. Sigue los pasos para hacer los 3 forms que se solicitan
5. Una vez terminado comprímelos y súbelos.

FORMULARIO 1

PASO 1/4

Haz lo necesario en Formularios1.html para que los 4 primeros controles del form del archivo original luzcan como el resultado esperado.

Archivo original

Table Color Themes x
localhost:48920/registration.html

Fill in this form to register to Visual Studio Smarts Conference

Personal Data

Salutation

First Name:

Last Name:

Email:

Your Smarts Membership status

Resultado esperado

Conference Registration x
localhost:49126/registration.html

Fill in this form to register to Visual Studio Smarts Conference

Personal Data

Salutation:

First Name:

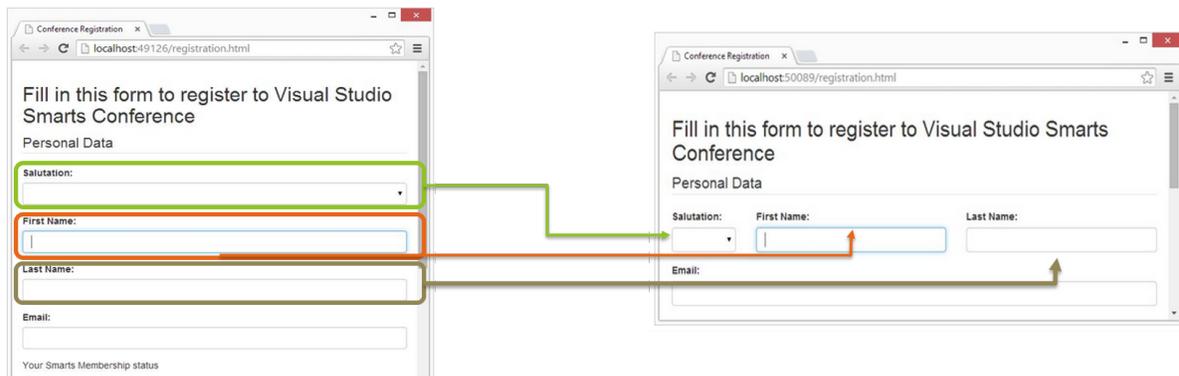
Last Name:

Email:

Your Smarts Membership status

PASO 2/4

Ahora haz lo necesario para que se desplieguen los 3 primeros campos en un mismo renglón.



PASO 3/4

Aplica los estilos necesarios a los controles de **Your Conference Account** para que se despliegan de la siguiente manera:

Formulario (Bootstrap)

file:///C:/Users/Becario/Documents/boots...

Email:

Your Smarts Membership status

Not a member Silver member Gold a member

[Read more about Smarts membership](#)

Your Conference Account

Login name:

Password: Confirm Password:

Sessions

What are the tracks you are interested in?

ALM ASP.NET C# Visual Basic SQL

What do you expect from sessions?

PASO 4/4

Posteriormente aplica los estilos necesarios para que los **checkbox** y **radios** se despliegan de manera horizontal en una sola fila, tal como se muestra a continuación:

Formulario (Bootstrap)

file:///C:/Users/Becario/Docu

Email:

Your Smarts Membership status

Not a member Silver member Gold a member

[read more about smarts membership](#)

Your Conference Account

Login name:

Password: Confirm Password:

Sessions

What are the tracks you are interested in?

ALM ASP.NET C# Visual Basic SQL

What do you expect from sessions?

FORMULARIO 2

Aplicar los estilos de maquetado necesarios para que el archivo Formularios2.html luzca como en la animación

Archivo | C:/Users/Usuario8/Desktop/Formularios/FormularioEjeHrzntl.html

Ingrese información al formulario

Dive site: (e.g. Al Arouk)

Region: (e.g. Egypt)

Date: dd/mm/a

Max depth (feet): 0

Bottom time (minutes):

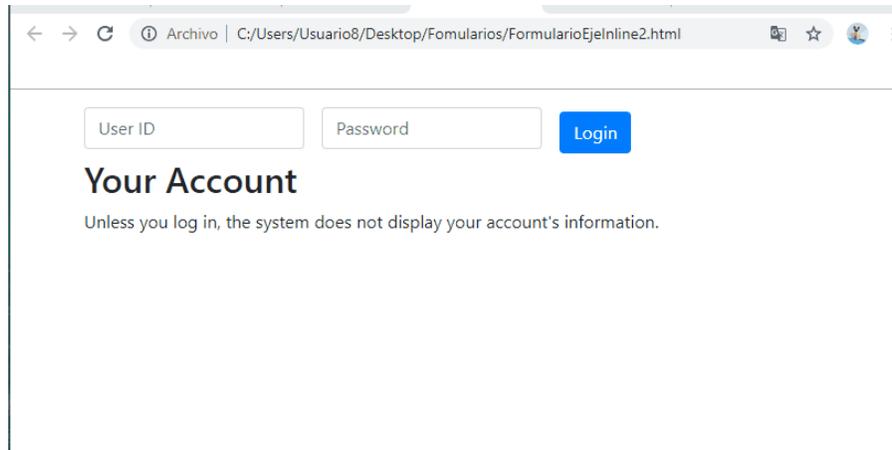
Nightdive

Comment:

Save

FORMULARIO 3

Modifica el archivo Formularios3.html, usa los estilos de **Bootstrap** para generar un formulario in-line tal como se muestra:



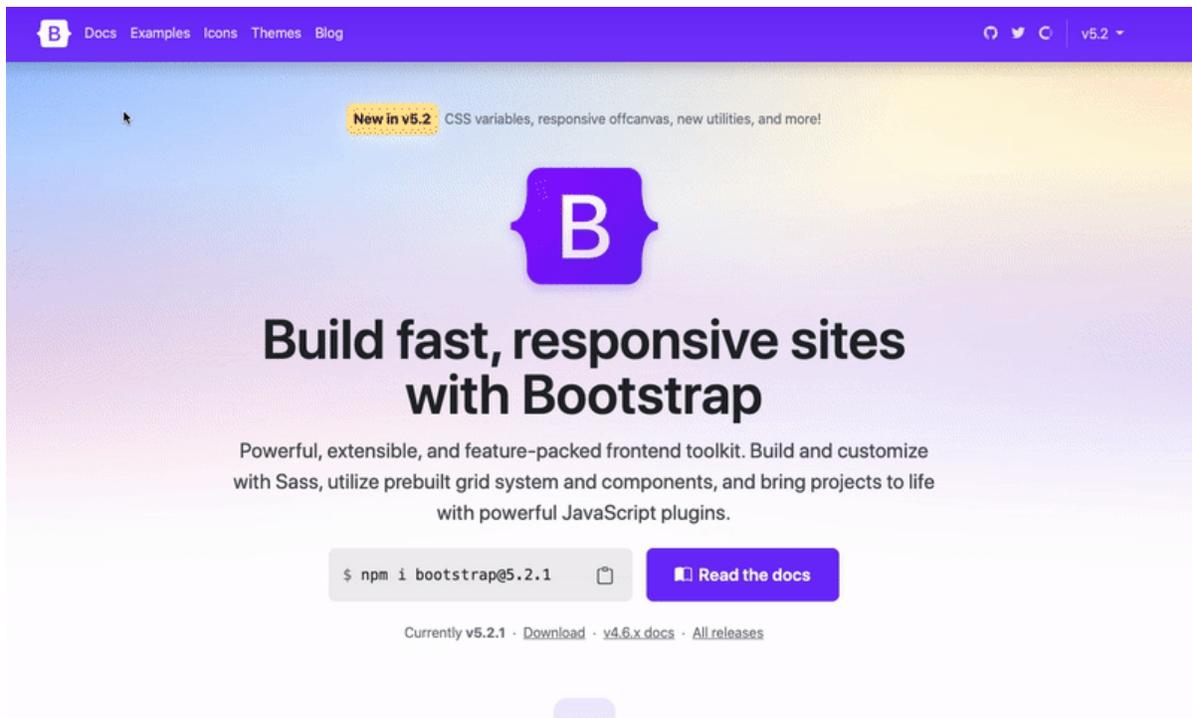
The screenshot shows a web browser window with the address bar displaying "Archivo | C:/Users/Usuario8/Desktop/Fomularios/FormularioEjeInLine2.html". The page content includes a login form with two input fields: "User ID" and "Password", followed by a blue "Login" button. Below the form is a section titled "Your Account" with the text "Unless you log in, the system does not display your account's information."

Bootstrap



Componentes

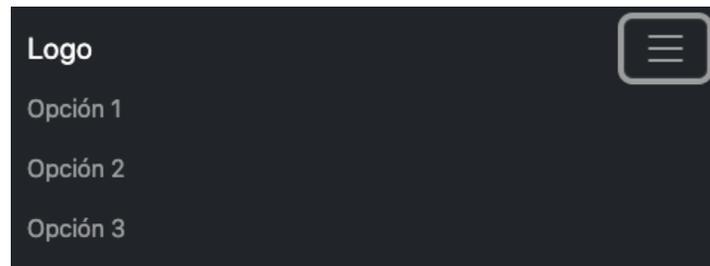
Para cerrar el curso integremos algunos componentes de Bootstrap, son elementos de interfaz de usuario preconstruidos que utilizan clases de estilo y scripts de Bootstrap para ahorrarnos tiempo en el desarrollo. Lo primero que debes saber es que cada componente está descrito en la [documentación oficial de Bootstrap](#), basta con localizar el componente que se acomoda a la solución de nuestras necesidades y copiar alguno de los ejemplos disponibles del componente que nos interesa.



Todo proyecto necesita un menú que permita al usuario navegar en nuestras páginas o secciones. Bootstrap tiene solucionado el menú mediante una barra de navegación provista por el componente *navbar*

Logo Opción 1 Opción 2 Opción 3

Este componente de navegación se adaptará de manera automática a pantallas de menor tamaño sin que tengamos que hacer más.



Es tan simple de implementar que basta con copiar el código y pegarlo en nuestro proyecto. Aquí puedes ver el código del ejemplo anterior

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Logo</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="#">Opción 1</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Opción 2</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Opción 3</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Una alternativa muy útil para probar componentes la encontramos en la plataforma gratuita [W3schools](https://www.w3schools.com/), en ella podrás practicar el uso de los componentes sin tener que abrir un proyecto en tu editor, pegar código y corregir errores, pues W3schools nos facilita un espacio de ensayo para nuestra posible solución. Cada componente cuenta con explicaciones, muestras, código y el botón **Try it yourself** que abre una nueva ventana con un editor de código listo para usar y probar cada componente.

The screenshot shows the W3Schools Bootstrap 5 Tutorial page. At the top, there is a navigation bar with the W3Schools logo and links for Tutorials, References, Exercises, and Videos. There are also buttons for 'Pro', 'Get Certified', and 'Free Website'. Below this is a dark navigation bar with links for HTML, CSS, JAVASCRIPT, SQL, PYTHON, JAVA, PHP, BOOTSTRAP (highlighted), HOW TO, and W3.CSS. A left sidebar contains a list of Bootstrap 5 components, with 'BS5 HOME' highlighted. The main content area features the title 'Bootstrap 5 Tutorial', navigation buttons for '< Home' and 'Next >', a green introductory text block stating that Bootstrap 5 is the newest version of the framework, and a 'Start learning Bootstrap 5 now »' button. Below this is a section titled 'Try it Yourself Examples' with a description of the tutorial's content and an online editor feature. At the bottom, there is a section titled 'Bootstrap 5 Example' with a blue rectangular placeholder.

W3schools

Tutorials ▾ References ▾ Exercises ▾ Videos

Pro Get Certified Free Website

HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP **BOOTSTRAP** HOW TO W3.CSS

Bootstrap 5 Tutorial

BS5 HOME

- BS5 Get Started
- BS5 Containers
- BS5 Grid Basic
- BS5 Typography
- BS5 Colors
- BS5 Tables
- BS5 Images
- BS5 Jumbotron
- BS5 Alerts
- BS5 Buttons
- BS5 Button Groups
- BS5 Badges
- BS5 Progress Bars
- BS5 Spinners
- BS5 Pagination
- BS5 List Groups
- BS5 Cards
- BS5 Dropdowns
- BS5 Collapse
- BS5 Navs
- BS5 Navbar
- BS5 Carousel
- BS5 Modal
- BS5 Tooltip
- BS5 Popover

Bootstrap 5 Tutorial

[< Home](#) [Next >](#)

Bootstrap 5 is the newest version of [Bootstrap](#), which is the most popular HTML, CSS, and JavaScript framework for creating responsive, mobile-first websites.

Bootstrap 5 is completely free to download and use!

[Start learning Bootstrap 5 now »](#)

Try it Yourself Examples

This tutorial contains hundreds of Bootstrap 5 examples.

With our online editor, you can edit the code, and click on a button to view the result.

Bootstrap 5 Example

Bootstrap



Actividad 8

Integración

Como integración final has una página Web que use 3 [componentes](#): *navbar*, *carousel* y *cards*. Debe quedarte como se ve el ejemplo, la imagen que ves fue construida desde el *Try it yourself editor* de *W3schools*, si lo deseas puedes usarlo también para ensayar, solo ten en cuenta que el vínculo de las imágenes se perderá si sacas el código de la plataforma, para realizar el ejercicio con las mismas imágenes puedes obtener las rutas absolutas de las imágenes con el botón derecho de tu mouse e integrarlas al código final probando que no falle nada.

Otra cosa por destacar es que los *cards* están anidados en columnas que usan la [clase p-2](#) (*padding: .5rem;*), para provocar que haya espacio entre nuestros *cards*.

Dónde utilizar los [containers](#) lo descubrirás por tu cuenta ¡Suerte! y gracias por participar en el curso. ¡Hasta pronto!

