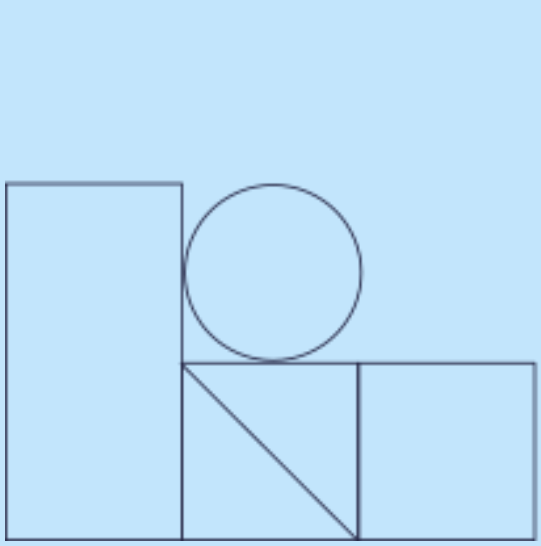
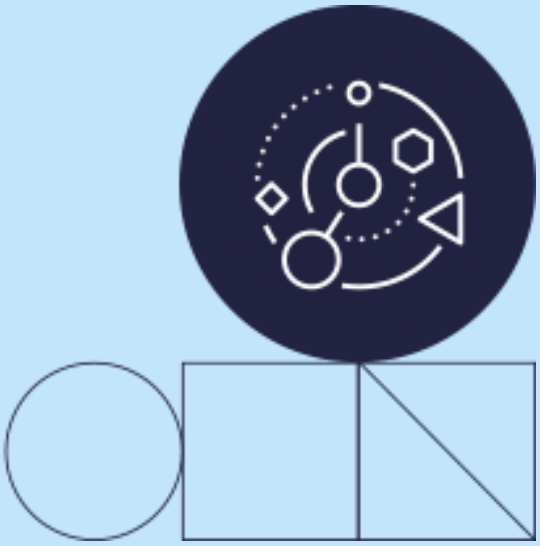
IBM SkillsBuild

IBM

Desarrollo Web

Bootstrap





Índice

Introducción

Bootstrap es un framework libre para desarrollo web.

Desarrollado inicialmente en 2011 por ingenieros de Twitter.

La versión actual, Bootstrap 5, aparece en mayo de 2021. A diferencia de las anteriores, emplea vanillla JavaScript, no jQuery.

Incluye plantillas HTML y CSS con tipografías, formas, botones, cuadros, tablas, barras de navegación, carruseles de imágenes y muchas otros elementos.

Aunque su preferencia es mobile first, permite crear diseños que se ven bien en múltiples dispositivos (responsive design) Orientado a programadores, no a diseñadores gráficos.

Características de Bootstrap 5

Ventajas

Resulta sencillo y rápido escribir páginas con muy buen aspecto.

Se adapta a distintos dispositivos (responsive design).

Proporciona un diseño consistente.

Es compatible con los navegadores modernos.

Es software libre.

Inconvenientes

Al ser una herramienta muy popular, las páginas web que no estén personalizadas quedan iguales que las de todo el mundo.

No es especialmente fácil personalizar los estilos.

Holamundo en Bootstrap

Para usar Bootstrap basta con:

- Definir el viewport.
- Incluir un elemento link apuntando al CSS de Bootstrap.
- Incluir un elemento script apuntando al código JavaScript de Bootstrap.

```
<!doctype html>
<html lang="es-ES">

<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">
  <link
href="https://cdn.jsdelivr.net/npm/bootst
rap@5.1.3/dist/css/bootstrap.min.css"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3" rel="stylesheet"
    crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH
9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous">
    </script>
  <title>Holamundo en Bootstrap 5</title>
</head>

<body>
  <div class="container">
    <h1>Holamundo en Bootstrap 5</h1>
  </div>
</body>

</html>
```

Adaptación del contenido a la pantalla

Ya desde su diseño original, un requisito importante para el web es que las páginas se puedan representar en pantallas de cualquier tamaño. Con la aparición de los smartphones, esto es aún más necesario y más complicado. A lo largo de los años se han usado varias técnicas para conseguir esto, cada vez mejores:

- Técnica inicial: Viewport. Barras de desplazamiento horizontal y vertical, recomposición de los elementos sobre el viewport.
- Primeros smartphones Viewport virtual.
- Teléfonos móviles actuales Diseño responsive basado en grid.

Para diseñar webs en dispositivos móviles, es importante tener claro qué es el viewport y cómo se comporta.

Viewport es la zona visible de una página web. En los navegadores tradicionales de escritorio, coincide con la ventana del navegador.

Supongamos una página web grande y compleja, como la portada de un periódico. La página no cabría en la ventana del navegador, el usuario usará las barras de scroll para mover el viewport sobre el documento. Al redimensionar la ventana, cambiará el tamaño del viewport.

Cambiar el tamaño del viewport reposiciona el texto y todos los elementos: las líneas se truncan, las imáges se recolocan, etc.

Viewport es un rectángulo que se compone un fragmento (tal vez completo) de la página web para presentarla al usuario.

Viewport virtual

Con la aparición de los navegadores en teléfonos móviles, los cambios del tamaño de la pantalla son mucho más drásticos. La técnicas tradicionales siguen funcionando, pero proporcionan una experiencia de uso muy poco satisfactoria.

El área visible de un móvil es demasiado pequeña, componer una página web tradicional en ese viewport normalmente no queda atractiva.

Además, en un navegador para móvil no hay barras de scroll, ocuparían un espacio demasiado valioso. Ni ventanas, serían demasiado pequeñas.

Para solucionar este problema, aparece el concepto del viewport virtual, mayor que el viewport ordinario (la pantalla).

Inicialmente lo introdujo Apple en su navegador Safari en iOS, luego pasó a ser estándar.

El ancho del viewport virtual es razonablemente grande, por ejemplo 980 pixeles en el navegador safari para iPhone.

El navegador compone la página sobre este viewport virtual, ya no hacen falta barras desplazamiento horizontal.

El usuario arrastra el viewport (la pantalla, más pequeña) sobre el viewport virtual, para que le muestre una zona u otra del documento. También se le puede permitir hacer zoom. Redimensionar este viewport ya no provoca la recomposición de la página.

Páginas responsive

Una página web moderna con un mínimo de calidad se entiende que tiene que ser responsive. Es decir, la página se adaptará al tamaño de la pantalla (escritorio, tablet, móvil, etc.), sin usar la barra de desplazamiento horizontal, que es muy incómoda. La barra de desplazamiento vertical sí se sigue usando.

El diseño responsive tal y como lo conocemos en la actualidad se basa en el uso de un grid. En español se traduce por cuadrícula, rejilla o casilla.

En estas páginas ya no hace falta un viewport virtual, porque la página está diseñada para adaptarse al viewport ordinario (la pantalla pequeña).

Las mismas 12 casillas se presenta de forma distinta en:

Un ordenador	**XXXXXXXXXXXX**
En una Tablet	XXXXXX XXXXXX
En un móvil	XXXX XXXX XXXX

Movile first

Con una propiedad de etiqueta meta, podemos indicar la escala inicial del viewport.

Como las páginas con bootstrap son responsive, especificamos que el viewport virtual coincida con el ancho de la pantalla, esto es, con el viewport ordinario.

En otras palabras: que no haya un viewport virtual.

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

También se puede inhabilitar el zoom en dispositivos móviles con user-scalable=no

Los usuarios solo podrían hacer scroll y tendrá una apariencia nativa.

```
  <meta name="viewport"
content="width=device-width, initial-
scale=1, maximum-scale=1, user-
scalable=no">
```

Contenedores

Para que sean responsivos, todos los elementos de Bootstrap deben estar dentro de un elemento contenedor.

Los contenedores no se pueden anidar.

Debemos asegurarnos de cerrar correctamente cada contenedor. Si alguna fila queda fuera, sus columnas quedarían mal alineadas. Y este error no lo detecta el W3C validator.

Para un contenedor responsivo de tamaño fijo, se usa "container":

```
  <div class="container">
    ...
  </div>
```

```
<!doctype html>
<html lang="es-ES">

<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">

  <link
href="https://cdn.jsdelivr.net/npm/bootst
rap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

  <script
src="https://cdn.jsdelivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH
9sENBO0LRn5q+8nbTov4+1p"
    crossorigin="anonymous"></script>
  <title>Ejemplo de container.</title>
</head>

<body>
  <div class="container">
    <div class="row">
      <div class="col-xl-2  bg-primary
text-white">.col</div>
      <div class="col-xl-10  bg-dark
text-white">.col</div>
    </div>

    <div class="row">
      <div class="col-xl-6  bg-primary
text-white">.col</div>
      <div class="col-xl-6  bg-dark text-
white">.col</div>
    </div>
  </div>

</body>

</html>
```

Si se desea un contenedor con el ancho total (del viewport), se ha de usar la expresión container-fluid:

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

Sistema de cuadrículas de Bootstrap

La pantalla se divide en filas y columnas. Llamaremos celda a la intersección entre una fila y una columna.

El contenido se coloca dentro de una celda, y siempre se mostrará dentro de esa misma celda. El ancho de cada celda se mide en casillas.

En cada fila hay hasta 12 casillas, que el diseñador decide cómo repartir entre celdas.

Cuando la pantalla tiene la suficiente resolución, las celdas de la misma fila se ven unas al lado de otras (disposición normal).

Cuando la resolución disminuye, las celdas que originalmente estaban en la misma fila pasan a verse unas encima de otras (disposición apilada).

Cada fila es un elemento div de HTML con la clase row. Observa que empleando notación de los selectores de CSS (donde el punto significa clase), podemos llamarle .row

Dentro de la fila hay elementos a los que en esta asignatura llamamos celdas, que pueden ser de los tipos .col-N, .col-sm-N, .col-md-N, .col-lg-N, .col-xl-N o .col-xxl-N

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

Estos 6 tipos de celdas dependen del ancho de viewport (pantalla) en el que queramos que las celdas se muestren en disposición normal, no apilada.

- .col-N: Pantallas muy pequeñas, de menos de 576px.
- .col-sm-N: Pantallas pequeñas de al menos 576px.
- .col-md-N: Pantallas medianas de al menos 768px.
- .col-lg-N: Pantallas grandes de al menos 992px.
- .col-xl-N: Pantallas muy grandes de al menos 1200px.
- .col-xxl-N: Pantallas extragrandes de al menos 1400px.

Donde N es un número entre 1 y 12, que indica el ancho de cada columna. El total de las columnas de cada fila puede sumar un máximo de 12. La frontera entre cada uno de estos tamaños se denomina breackpoint.

Columnas .col-xxl-N: Disposición normal en pantallas extragrandes.

Se apilan en pantallas muy grandes, grandes, medianas, pequeñas o muy pequeñas

- Columnas .col-xl-N: Disposición normal en pantallas muy grandes o extragrandes. Se apilan en pantallas grandes, medianas, pequeñas o muy pequeñas
- Columnas .col-lg-N: Disposición normal en pantallas grandes, muy grandes o extragrandes.

Se apilan en medianas, pequeñas o muy pequeñas.

- Columnas .col-md-N: Disposición normal en pantallas medianas, grandes, muy grandes o extragrandes.

Se apilan en: pequeñas o muy pequeñas

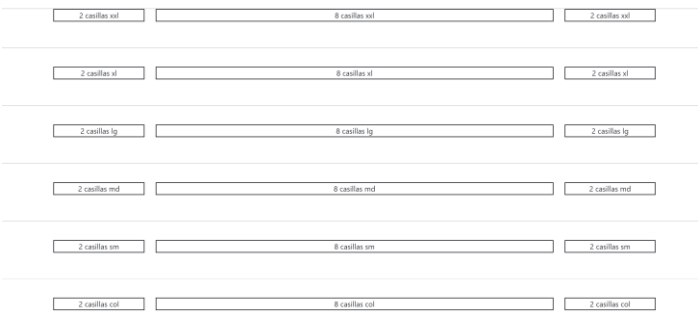
- Columnas .col-sm-N: Disposición normal en pantallas pequeñas, medianas, grandes, muy grandes o extragrandes.

Se apilan en muy pequeñas.

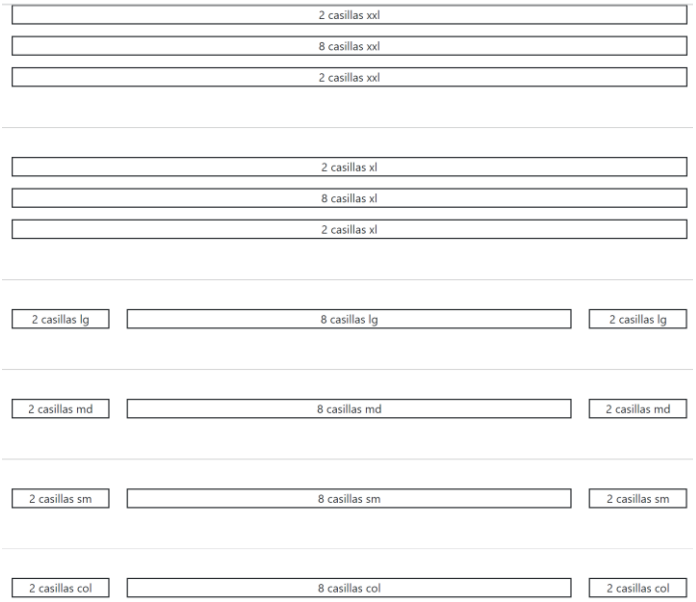
- Columnas .col-N: Disposición normal en cualquier pantalla: muy pequeñas, pequeñas, medianas, grandes, muy grandes o extragrandes. Nunca se apilan.

Dicho de otro modo, cada tipo de columna se muestra en su disposición normal, esto es, horizontalmente, si la pantalla es de su tipo o de un tipo mejor. En otro caso, las casillas se apilan verticalmente.

Esto parece un poco complicado, pero con el siguiente ejemplo veremos que no:



Si vamos reduciendo el ancho gradualmente, esto es equivalente a tener una pantalla menor. Veremos que según vayamos reduciendo, las cuadrículas que originalmente están en disposición normal (horizontal) se van apilando (verticalmente):



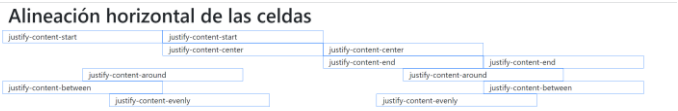
Alineación horizontal de las cuadrículas

Las celdas (que formarán columnas cuando sean varias a la misma distancia del eje vertical) se pueden alinear de diversa forma en horizontal añadiendo a la fila (el div de clase row) las clases:

- justify-content-start
- justify-content-center
- justify-content-end
- justify-content-around
- justify-content-between
- justify-content-evenly

En el código fuente de este ejemplo podremos ver:

- El resultado de usar las distintas clases de alineamiento horizontal. En este caso con dos columnas de 3 casillas cada una
- El uso de la clase border con el color border-primary.



Componentes de Boostrap

Bootstrap viene con una serie de estilos (generalmente en formato de clase CSS) y componentes en JavaScript:

- btn
- table
- card
- carousel

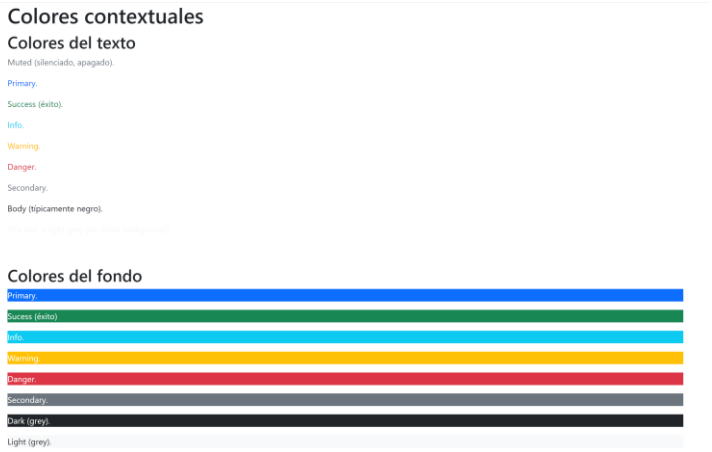
y otras utilidades responsive.

Colores contextuales

La gama concreta de colores se decidirá en el CSS. Aquí pondremos clases con valor semántico.

Con alguna excepción como light o white, puesto que, al elegir el color del fondo, puede ser necesario indicar también el color del texto (en este ejemplo, el texto blanco sobre fondo blanco no se ve):

```
  <h2>Colores del texto</h2>
  <p class="text-muted">Muted
(silenciado, apagado).</p>
  <p  class="text-primary">Primary.</p>
  <p class="text-success">Success
(´exito).</p>
  <p class="text-info">Info.</p>
  <p  class="text-warning">Warning.</p>
  <p  class="text-danger">Danger.</p>
  <p  class="text-
secondary">Secondary.</p>
  <p class="text-body">Body (típicamente
negro).</p>
  <p class="text-light">Light grey .</p>
  <p  class="text-white">White.</p>
  <h2>Colores del fondo</h2>
  <p class="bg-primary text-
white">Primary.</p>
  <p class="bg-success text-white">Sucess
(´exito)</p>
  <p class="bg-info text-white">Info.</p>
  <p class="bg-warning text-
white">Warning.</p>
  <p class="bg-danger text-
white">Danger.</p>
  <p class="bg-secondary text-
white">Secondary.</p>
  <p class="bg-dark text-white">Dark
(grey).</p>
  <p class="bg-light text-dark">Light
(grey).</p>
```



Botones

La clase btn de Bootstrap puede anñdirse a los elementos HTML:

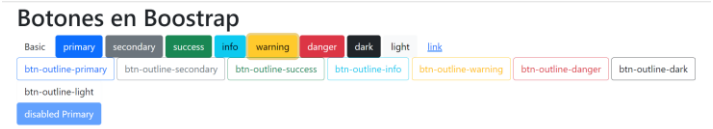
<button>, <input> y <a>

Tienen efecto hover: destacan un botón cuando se posiciona el ratón encima.

```
  <button type="button"
class="btn">Basic</button>
  <button type="button" class="btn btn-
primary">primary</button>
  <button type="button" class="btn btn-
secondary">secondary</button>
  <button type="button" class="btn btn-
success">success</button>
  <button type="button" class="btn btn-
info">info</button>
  <button type="button" class="btn btn-
warning">warning</button>
  <button type="button" class="btn btn-
danger">danger</button>
  <button type="button" class="btn btn-
dark">dark</button>
  <button type="button" class="btn btn-
light">light</button>
  <button type="button" class="btn btn-
link">link</button>
  <button type="button" class="btn btn-outline-
primary">btn-outline-primary</button>
  <button type="button" class="btn btn-outline-
secondary">btn-outline-secondary</button>
  <button type="button" class="btn btn-outline-
success">btn-outline-success</button>
  <button type="button" class="btn btn-outline-
info">btn-outline-info</button>
  <button type="button" class="btn btn-outline-
warning">btn-outline-warning</button>
  <button type="button" class="btn btn-outline-
danger">btn-outline-danger</button>
  <button type="button" class="btn btn-outline-
dark">btn-outline-dark</button>
  <button type="button" class="btn btn-outline-light
text-dark">btn-outline-light</button>
```

Con el atributo disabled (atributo, no clase), el botón queda inhabilitado.

```
  <button type="button" class="btn btn-
primary" disabled> disabled Primary
  </button>
```



Imágenes

Para modificar el aspecto de una imagen, Bootstrap, nos permite añadir clases al elemento <img>

Contorno:

rounded: Esquinas redondeadas

rounded-circle: Circular

img-thumbnail: Miniatura (reborde blanco)

Alineación:

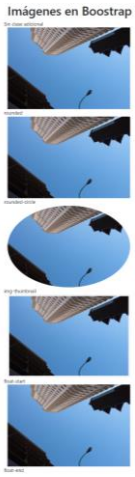
float-start: Izquierda

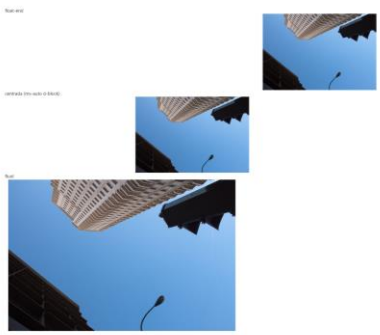
float-end: Derecha

mx-auto d-block: centrada

fluid: Todo el espacio disponible

```
  <div class="row"> rounded
    <div class="col-xl-12">
    <img src="images/plaza_espana.jpg"
alt="Plaza de Espa˜na, Madrid"
width="400" class="rounded">
    </div>
  </div>
```



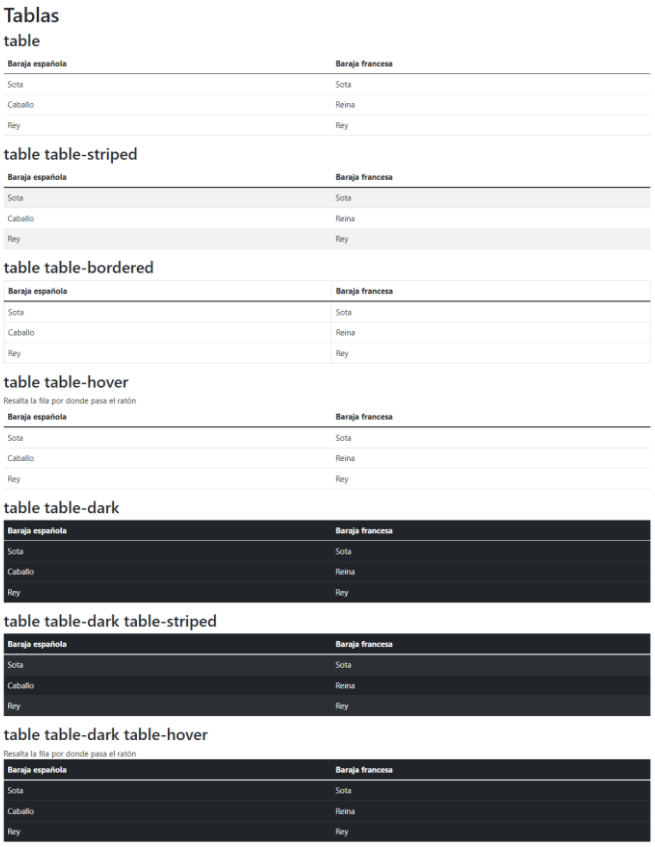


Tablas

Para dar formato a un elemento <table>, Bootstrap 5 nos ofrece las clases:

.table, .table-bordered, .table-hover, .table-dark y .table-striped

```
<table class="table table-striped">
  <thead>
  <tr>
  <th>Baraja española</th>
  <th>Baraja francesa</th>
  </tr>
  </thead>
  <tbody>
  <tr>
  <td>Caballo</td>
  <td>Reina</td>
  </tr>
  <tr>
  <td>Rey</td>
  <td>Rey</td>
  </tr>
  </tbody>
  </table>
```



Se les puede poner un color contextual de fondo añadiendo las clases que ya conocemos:

.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark y .bg-light

```
  <div class="card" style="width:400px">
    <div class="card-header">
    <h4 class="card-title">Gato
Panchi</h4>
    </div>
    <div  class="card-body">
    <img src="images/gato.jpg"
alt="Fotograf´ıa de un gato llamado
Panchi" width="300" class="mx-auto d-
block">
    </div>
    <div class="card-footer">
    <a href="#" class="btn btn-primary
float-end">M´as informaci´on</a>
    </div>
  </div>
```

Cards

Una tarjeta (card) es una caja redondeada dividida en cabecera, cuerpo y pie.

Es útil para agrupar otros elementos como botones, formularios, imágenes, etc.

Sucesor de los antiguos panels en las versiones anteriores de Bootstrap.

Formularios

Bootstrap incluye clases para mejorar el aspecto y usabilidad de los formularios:

- El uso de <label> es necesario, no es válido escribir texto HTML para identificar los elementos del formulario.
- Los distintos elementos de un formulario aparecen unos debajo de otros. Si los queremos unos al lado de otros, usaremos las fils y columnas de la rejilla.
- A los <label> les anñdimos class="form-label"
- A los elementos de entrada de texto, <input> y <textarea> les añadimos class="form-control"
- A los <checkbox> los metemos en un al que añadimos class="form-check"
- A los <input type="radio"> <input type="checkbox"> les añadimos class="form-check-input"

```
<!doctype html>
<html lang="es-ES">

<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">

  <link
href="https://cdn.jsdelivr.net/npm/bootst
rap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH
9sENBO0LRn5q+8nbTov4+1p"
    crossorigin="anonymous"></script>

  <title>Formularios en Bootstrap
5.</title>
</head>

<body>
  <div class="container">

    <h1>Formularios en Bootstrap 5</h1>

    <form action="/action_page.html">
      <div class="form-group">
        <label for="usuario" class="form-
label"> Nombre de usuario:</label>
        <input type="text" id="usuario"
class="form-control" name="usuario">
      </div>

      <div class="form-group">
        <label for="contrasenya"
class="form-label"> Contraseña:</label>
        <input type="password"
name="contrasenya" id="contrasenya"
class="form-control">
      </div>


      <div class="form-group">
        <label for="pais" class="form-
label">País</label>
        <input type="text" name="pais"
id="pais" value="España" class="form-
control"><br><br>
      </div>

      <input type="submit">
    </form>

    <hr>

    <form>
      <input type="radio" name="os"
value="Linux" class="form-check-input"
checked>Linux<br>
      <input type="radio" name="os"
value="MacOS" class="form-check-
input">MacOS<br>
```

```
      <input type="radio" name="os"
value="Windows" class="form-check-
input">Windows<br>
      <input type="radio" name="os"
value="other" class="form-check-
input">Otro<br>
      <br>
      <div class="form-check">
        <input type="checkbox"
name="terminos" value="si" class="form-
check-input">He leido los términos y
        condiciones<br>
        <input type="checkbox"
name="publicidad" value="si" class="form-
check-input">Deseo recibir comunicaciones
        comerciales<br>
      </div>
      <br>
      <input type="submit">
    </form>

    <hr>

    <form class="form">
      <fieldset>
        <legend>
          Datos personales
        </legend>

        <div class="form-group">
          <label for="favcolor"
class="form-label">Elija un
color:</label>
          <br>
          <input type="color"
name="favcolor" id="favcolor">

        </div>

        <div class="form-group">
          <label for="nacimiento"
class="form-label"> Fecha de
nacimiento:</label>
          <input type="date"
name="nacimiento" id="nacimiento">
        </div>

        <div class="form-group">
          <label for="nacimiento-hora"
class="form-label">Fecha y hora de
nacimiento:</label>
          <input type="datetime-local"
name="nacimiento-hora" id="nacimiento-
hora">
        </div>

        <div class="form-group">
          <label for="email" class="form-
label"> E-mail:</label>
          <input type="email"
name="email" id="email" class="form-
control">
        </div>

        <div class="form-group">
          <label for="numero"
class="form-label"> Indica un número del
1 al 10:</label>
          <input type="number"
name="numero" min="1" max="10"
id="numero" class="form-control">
        </div>

        <br>
        <input type="submit">
      </fieldset>
    </form>

    <br>
    <hr>

    <form class="form">
      <div class="form-group">
        <label for="departamento"
class="form-label">Indique el
departamento:</label>
        <select name="departamento"
id="departamento">
          <option
value="Comercial">Comercial</option>
          <option
value="Técnico">Técnico</option>
          <option
value="Webmaster">Webmaster</option>
        </select>
      </div>
```

```
      <div class="form-group">
        <label for="mensaje" class="form-
label">Mensaje:</label>
        <textarea name="mensaje"
id="mensaje" rows="10" cols="30"
          class="form-control">Escriba
aquí su mensaje.</textarea>
      </div>
      <input type="submit">
    </form>
  </div>
</body>

</html>
```

Carousel

El componente carousel muestra fotografías que se desplazan horizontalmente, como un pase de diapositivas. Se les puede añadir título o cualquier otro texto.

El elemento de mayor nivel jerárquico del carrusel es un div con las clases carousel y slide Tiene un atributo id cuyo valor será referenciado por los botones que contenga:

```
  <div id="carrusel01" class="carousel
slide" data-bs-ride="carousel">
```

El .slide .carousel contendrá tres divs:

.carousel-indicators: Los puntos o pequeñas líneas que representan cada una de las fotos. Un div de clase carousel-indicators

.carousel-inner: Un div de clase carousel-inner con las imágenes.

Cada imagen es un carousel-item, que contiene la imagen y un carousel-caption. Es recomendable que todas las imágenes tengan la misma relación alto/ancho.

Los botones

```
<!doctype html>
<html lang="es-ES">

<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">

  <link
href="https://cdn.jsdelivr.net/npm/bootst
```

```
rap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH
9sENBO0LRn5q+8nbTov4+1p"
    crossorigin="anonymous"></script>

  <title>Carrusel en Bootstrap5 .</title>
</head>

<body>
  <div class="container">
    <div id="carrusel01" class="carousel
slide" data-bs-ride="carousel">

      <!-- Indicators -->
      <div class="carousel-indicators">
        <button type="button" data-bs-
target="#carrusel01" data-bs-slide-to="0"
class="active"></button>
        <button type="button" data-bs-
target="#carrusel01" data-bs-slide-
to="1"></button>
        <button type="button" data-bs-
target="#carrusel01" data-bs-slide-
to="2"></button>
        <button type="button" data-bs-
target="#carrusel01" data-bs-slide-
to="3"></button>
      </div>

      <!-- Fotos -->
      <div class="carousel-inner">
        <div class="carousel-item
active">
          <img
src="images/ensalada_reloj.jpg"
alt="Ensalada reloj" class="d-block w-
100">
          <div class="carousel-caption">
            <h3>Ensalada reloj</h3>
          </div>
        </div>

        <div class="carousel-item">
          <img
src="images/esparragos_manzana.jpg"
alt="Esparragos con manzana" class="d-
block w-100">
          <div class="carousel-caption">
            <h3>Espárragos con
manzana</h3>
          </div>
        </div>
        <div class="carousel-item">
          <img
src="images/ensalada_centollo.jpg"
alt="Ensala de centollo" class="d-block
w-100">
          <div class="carousel-caption">
            <h3>Ensalada de centollo</h3>
          </div>
        </div>
        <div class="carousel-item">
          <img
src="images/sushi_suelto.jpg" alt="Sushi
suelto" class="d-block w-100">
          <div class="carousel-caption">
            <h3>Sushi suelto</h3>
          </div>
        </div>
      </div>

      <!-- Botones izquierda y derecha --
>
      <button class="carousel-control-
prev" type="button" data-bs-
target="#carrusel01" data-bs-
slide="prev">
        <span class="carousel-control-
prev-icon"></span>
      </button>
      <button class="carousel-control-
next" type="button" data-bs-
target="#carrusel01" data-bs-
slide="next">
        <span class="carousel-control-
next-icon"></span>
      </button>
    </div>
```

```
  </div>
</body>

</html>
```

Deshabilitar elementos

Como hemos visto, muchos elementos bootstrap admiten la clase disabled para indicar que tengan un aspecto gráfico distinto, deshabilitado.

Pero esto no los deshabilita realmente. Para deshabilitar por completo un elemento, usamos el atributo disabled.

```
  <button type="button" class="btn btn-
lg" disabled>Botón</button>
  <input type="text" name="lname"
disabled><br>
```

Depuración

Si la página no tiene el aspecto que buscamos:

- Debemos asegurarnos de que todos los elementos están dentro de un container. Normalmente solo deberíamos usar uno para la página.
- Usar el W3C validator. Detectará elementos sin cerrar (aunque no elementos cerrados en el sitio incorrecto)
- Comprobar que la estructura de los div está bien, que no hemos cerrado ninguno demasiado pronto o demasiado tarde. Un buen editor nos ayudará con esto mostrando el código por niveles.
- Si usamos Bootstrap, no añadir directamente reglas CSS. Excepto si estamos seguros de lo que hacemos.