

Desenvolvimento Web

Bootstrap



Índice

Introdução	3
Características da Bootstrap 5	4
Vantagens	4
Desvantagens	4
Holamundo em Bootstrap	4
Adaptação do conteúdo à tela	5
Viewport virtual	5
Páginas responsive	6
Mobile first	6
Contêineres	6
Sistema de grade Bootstrap	7
Alinhamento horizontal das grades	9
Componentes do Bootstrap	9
Cores contextuais	10
Botões	10
Imagens	11
Esboço	11
Alinhamento	12
Tabelas	12
Cards	13
Formulários	14
Carousel	16
Os botões	16
Elementos incapacitantes	18
Depuração	18

Introdução

Bootstrap é um framework gratuito para o desenvolvimento da web.

Desenvolvido inicialmente em 2011 pelos engenheiros do Twitter.

A versão atual, Bootstrap 5, será lançada em maio de 2021. Ao contrário das versões anteriores, ele usa o JavaScript de baunilha, não o jQuery.

Ele inclui modelos HTML e CSS com fontes, formas, botões, caixas, tabelas, barras de navegação, carrosséis de imagens e muitos outros elementos.

Embora sua preferência seja a mobilidade em primeiro lugar, permite criar projetos que tenham boa aparência em múltiplos dispositivos (responsive design) Orientado para programadores, não para designers gráficos.

Características da Bootstrap 5

Vantagens

É fácil e rápido escrever páginas que parecem ótimas.

Ela se adapta a diferentes dispositivos (responsive design).

Ele proporciona um projeto consistente.

É compatível com os navegadores modernos.

É um software livre.

Desvantagens

Como uma ferramenta muito popular, as páginas web que não são personalizadas permanecem as mesmas que as de todos os outros.

Não é particularmente fácil personalizar os estilos.

Holamundo em Bootstrap

Para usar o Bootstrap você só precisa fazê-lo:

- Defina o viewport.
- Incluir um elemento link apontando para o Bootstrap CSS.
- Incluir um elemento de script apontando para o código JavaScript do 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-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjD
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-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYsOg+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>
```

Adaptação do conteúdo à tela

Desde seu projeto original, um requisito importante para a web é que as páginas possam ser renderizadas em telas de qualquer tamanho. Com o advento dos smartphones, isto se tornou ainda mais necessário e mais complicado. Várias técnicas têm sido utilizadas ao longo dos anos para alcançar isto, cada vez melhor e melhor:

- Técnica inicial: **Viewport**. Barras de rolagem horizontais e verticais, recomposição de elementos no **viewport**.
- Primeiros smartphones **Viewport virtual**.
- Telefones celulares atuais Design responsivo baseado em **grid**.

Para projetar websites para dispositivos móveis, é importante ser claro sobre o que é o **viewport** e como ele se comporta.

O **Viewport** é a área visível de uma página da web. Em navegadores de mesa tradicionais, ele coincide com a janela do navegador.

Vamos assumir uma página web grande e complexa, tal como a primeira página de um jornal. A página não caberia na janela do navegador, o usuário usará as barras de scroll para mover o **viewport** sobre o documento. O redimensionamento da janela mudará o tamanho do **viewport**.

Redimensionamento do **viewport** reposiciona o texto e todos os elementos: as linhas são truncadas, as imagens são reposicionadas...etc.

Viewport é um retângulo que compõe um fragmento (talvez um completo) da página web para apresentá-lo ao usuário.

Viewport virtual

Com o advento dos navegadores nos telefones celulares, as mudanças no tamanho da tela são muito mais drásticas. As técnicas tradicionais ainda funcionam, mas proporcionam uma experiência de usuário muito insatisfatória.

A área visível de um telefone celular é muito pequena, compondo uma página da web tradicional nesse **viewport** é geralmente pouco atraente.

Além disso, em um navegador móvel não há barras de rolagem, elas ocupariam um espaço muito valioso. Nem janelas, elas seriam muito pequenas.

Para resolver este problema, surgiu o conceito de **viewport** virtual, maior do que o viewport comum (a tela).

Inicialmente introduzido pela Apple em seu navegador Safari no iOS, mais tarde tornou-se padrão.

A largura do **viewport virtual** é razoavelmente grande, por exemplo 980 pixels no navegador Safari para iPhone.

O navegador compõe a página nesta janela de **viewport virtual**, as barras de rolagem horizontais não são mais necessárias.

O usuário arrasta o **viewport** (a tela menor) sobre o **viewport virtual** para exibir uma área do documento ou outra. Ele também pode ser autorizado a aumentar e diminuir o zoom. O redimensionamento deste **viewport** já não faz com que a página seja recomposta.

Páginas responsive

Um site moderno com um mínimo de qualidade é entendido como sendo responsivo. Ou seja, a página se adaptará ao tamanho da tela (desktop, tablet, móvel, etc.), sem utilizar a barra de rolagem horizontal, o que é muito desconfortável. A barra de rolagem vertical ainda é utilizada.

O projeto responsivo como o conhecemos hoje é baseado no uso de uma [grid](#). Em brasileiro, isto se traduz como caixa.

Um [viewport virtual](#) não é mais necessário nestas páginas, porque a página foi projetada para se adaptar ao [viewport](#) comum (a pequena tela).

Os mesmos 12 quadrados são apresentados de forma diferente em:

Um computador	XXXXXXXXXXXX
Em uma Tablet	XXXXXX XXXXXX
Em um telefone celular	XXXX XXXX XXXX

Mobile first

Com uma propriedade meta tag, podemos especificar a escala inicial do [viewport](#).

Como as páginas de [bootstrap](#) são responsivo, especificamos que o [viewport virtual](#) deve corresponder à largura da tela, ou seja, o [viewport](#) comum.

Em outras palavras: nenhum [viewport virtual](#).

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

Também é possível desativar o zoom em dispositivos móveis com [user-scalable=no](#)

Os usuários só poderão fazer scroll e terão um visual e uma sensação nativa.

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

Contêineres

Para ser responsivo, todos os elementos do Bootstrap devem estar dentro de um elemento de contêiner.

Os contêineres não podem ser aninhados.

Devemos nos certificar de fechar cada contêiner corretamente. Se alguma fila for deixada de fora, suas colunas serão desalinhadas. E este erro não é detectado pelo W3C validator.

Para um recipiente de tamanho fixo, usamos "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-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

  <script
src="https://cdn.jsdelivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYS0g+OMhuP+IlRH
9sENB00LRn5q+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>

</body>

</html>
```



Se você quiser um container com a largura total (do [viewport](#)), você tem que usar a expressão [container-fluid](#):

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

Sistema de grade Bootstrap

A tela é dividida em filas e colunas. A interseção entre uma linha e uma coluna é chamada de célula.

O conteúdo é colocado dentro de uma célula, e será sempre exibido dentro dessa célula. A largura de cada célula é medida em células.

Em cada fila há até 12 células, que o projetista decide como distribuir entre as células.

Quando a tela tem resolução suficiente, as células da mesma linha são exibidas uma ao lado da outra (layout normal).

Quando a resolução diminui, as células que estavam originalmente na mesma fila são exibidas umas em cima das outras (layout empilhado).

Cada linha é um elemento [HTML div](#) com classe [row](#). Note que usando a notação seletora CSS (onde o ponto significa classe), podemos chamá-lo de [.row](#)

Dentro da fila há elementos que neste assunto chamamos de células, que podem ser dos 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>
```

Estes 6 tipos de células dependem da largura do [viewport](#) (tela) no qual queremos que as células sejam exibidas em um arranjo normal, não empilhadas.

- [.col-N](#): Telas muito pequenas, menos do que 576px.
- [.col-sm-N](#): Pequenas telas de pelo menos 576px.
- [.col-md-N](#): Telas médias de pelo menos 768px.
- [.col-lg-N](#): Telas grandes de pelo menos 992px.
- [.col-xl-N](#): Telas muito grandes de pelo menos 1200px.
- [.col-xxl-N](#): Telas extra grandes de pelo menos 1400px.

Onde N é um número entre 1 e 12, indicando a largura de cada coluna. O total das colunas em cada linha pode somar até um máximo de 12. A borda entre cada um desses tamanhos é designada como [breakpoint](#).

[Columnas .col-xxl-N](#): Layout normal em telas extra grandes.

Empilhados em telas muito grandes, grandes, médias, pequenas ou muito pequenas.

- [Columnas .col-xl-N](#): Layout normal em telas muito grandes ou extra grandes. Empilhados em telas grandes, médias, pequenas ou muito pequenas.
- [Columnas .col-lg-N](#): Layout normal em displays grandes, muito grandes ou extra grandes.

Empilhamento em tamanhos médios, pequenos ou muito pequenos.

- [Columnas .col-md-N](#): Layout normal em displays médios, grandes, muito grandes ou extra grandes.

Empilhamento: pequeno ou muito pequeno.

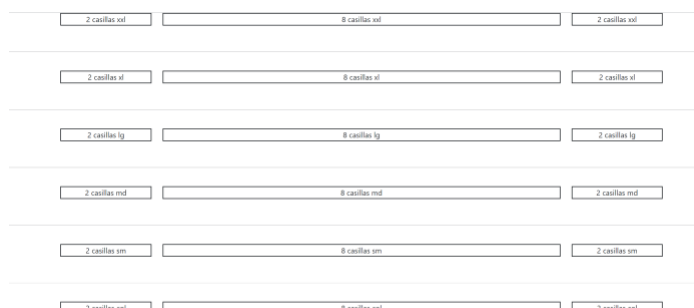
- [Columnas .col-sm-N](#): Layout normal em displays pequenos, médios, grandes, muito grandes ou extra-grandes.

Empilhado em muito pequenos.

- [Columnas .col-N](#): Arranjo normal em qualquer tela: muito pequena, pequena, média, média, grande, muito grande ou extra grande. Nunca empilhados.

Em outras palavras, cada tipo de coluna é exibido em sua disposição normal, ou seja, horizontalmente, se a exibição for do seu tipo ou de um tipo melhor. Caso contrário, as células são empilhadas verticalmente.

Isto parece um pouco complicado, mas com o seguinte exemplo, veremos que não é:



Se reduzirmos gradualmente a largura, isto equivale a ter uma tela menor. Você verá que ao reduzir a

largura, as grades que estavam originalmente em um layout normal (horizontal) se empilharão (verticalmente):

2 casillas xxl	
8 casillas xxl	
2 casillas xxl	

2 casillas xl	
8 casillas xl	
2 casillas xl	

2 casillas lg	8 casillas lg	2 casillas lg
---------------	---------------	---------------

2 casillas md	8 casillas md	2 casillas md
---------------	---------------	---------------

2 casillas sm	8 casillas sm	2 casillas sm
---------------	---------------	---------------

2 casillas col	8 casillas col	2 casillas col
----------------	----------------	----------------

Alinhamento horizontal das grades

As células (que formarão colunas quando houver várias células à mesma distância do eixo vertical) podem ser alinhadas horizontalmente de várias maneiras, acrescentando à linha (a classe `row div`) as classes:

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

No código fonte deste exemplo, podemos ver:

- O resultado da utilização das diferentes classes de alinhamento horizontal. Neste caso, com duas colunas de 3 células cada.
- O uso da classe de borda com a cor de `border-primary`.

Alineación horizontal de las celdas



Componentes do Bootstrap

Bootstrap vem com um conjunto de estilos (geralmente em formato de classe CSS) e componentes JavaScript.:

- `btn`
- `table`
- `card`
- `carousel`

e outros responsive utilities.

Cores contextuais

A gama de cores do concreto será decidida no CSS. Aqui colocaremos classes com valor semântico.

Com algumas exceções, como `light` ou `white`, já que, ao escolher a cor do fundo, também pode ser necessário indicar a cor do texto (neste exemplo, o texto branco sobre fundo branco não é visível):

```
<h2>Colores del texto</h2>
<p class="text-muted">Muted
(silenciado, apagado).</p>
<p class="text-primary">Primary.</p>
<p class="text-success">Success
('éxito').</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
('éxito')</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>
```

Colores contextuales

Colores del texto

Muted (silenciado, apagado).

Primary.

Success (éxito).

Info.

Warning.

Danger.

Secondary.

Body (típicamente negro).

Light grey (light grey con fondo background).

Colores del fondo

Primary.

Success (éxito).

Info.

Warning.

Danger.

Secondary.

Dark (grey).

Botões

A classe `btn` do Bootstrap pode ser adicionada aos elementos HTML:

`<button>`, `<input>` e `<a>`

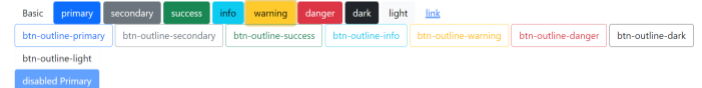
Eles têm um efeito **hover**: eles destacam um botão quando o mouse está pairando sobre ele.

```
<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>
```

Com o atributo disabled (atributo, não classe), o botão é desativado.

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

Botones en Bootstrap



Imagens

Para modificar a aparência de uma imagem, Bootstrap nos permite acrescentar classes ao elemento ``

Esboço

rounded: Cantos arredondados

rounded-circle: Circular

img-thumbnail: Miniatura (flange branca)

Alinhamento

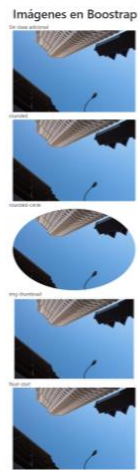
`float-start`: Esquerda

`float-end`: Direita

`mx-auto` d-block: centrada

`fluid`: Todo o espaço disponível

```
<div class="row"> rounded
  <div class="col-xl-12">
    
  </div>
</div>
```



Tabelas

Para formatar um elemento `<table>`, o Bootstrap 5 nos oferece as classes:

`.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>
```

Tablas

table

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-striped

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-bordered

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-hover

Resalta la fila por donde pasa el ratón

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-dark

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-dark table-striped

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

table table-dark table-hover

Resalta la fila por donde pasa el ratón

Baraja española	Baraja francesa
Sota	Sota
Caballo	Reina
Rey	Rey

Eles podem receber uma cor de fundo contextual, acrescentando as classes que já conhecemos:

`.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">
    
  </div>
  <div class="card-footer">
    <a href="#" class="btn btn-primary
    float-end">Más informaci'on</a>
  </div>
</div>
```

Cards

Um cartão (**card**) é uma caixa redonda dividida em cabeçalho, corpo e rodapé.

É útil para agrupar outros elementos, como botões, formulários, imagens... etc.

Sucessor dos **panels** antigos nas versões anteriores do Bootstrap.

Cards

Gato Panchi



Más información

Formulários

O bootstrap inclui aulas para melhorar a aparência e a usabilidade dos formulários:

- O uso de `<label>` é necessário, não é válido para escrever texto HTML para identificar os elementos do formulário.
- Os diferentes elementos de uma forma aparecem um abaixo do outro. Se os quisermos um ao lado do outro, usaremos as linhas e colunas da grade.
- Ao `<label>` adicionamos `class="form-label"`
- Para os elementos da entrada de texto, `<input>` e `<textarea>` adicionamos `class="form-control"`
- Para os `<checkbox>` os colocamos em um `<div>` ao qual acrescentamos `class="form-check"`
- Para os `<input type="radio">` `<input type="checkbox">` adicionamos `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.jsdelivrivr.net/npm/bootst
rap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjD
brCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivrivr.net/npm/bootstr
ap@5.1.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH
9sENB00LRn5q+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="contrasena"
class="form-label"> Contraseña:</label>
        <input type="password"
name="contrasena" id="contrasena"
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 leído 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>

```

Formularios en Bootstrap 5

Nombre de usuario:

Contraseña:

País:

España

Enviar

☒ Linux
☐ MacOS
☐ Windows
☐ Otro

☐ He leído los términos y condiciones
☐ Deseo recibir comunicaciones comerciales

Enviar

Datos personales

Elija un color:

Fecha de nacimiento: dd/mm/aaaa

Fecha y hora de nacimiento: dd/mm/aaaa --:--

E-mail:

Indica un número del 1 al 10:

Enviar

Indique el departamento: Comercial

Mensaje:

Escriba aquí su mensaje.

Enviar

Carousel

O componente `carousel` exhibe fotografías que rolan horizontalmente, como um slideshow. Você pode adicionar legendas ou qualquer outro texto a eles.

O elemento de nível superior do `carousel` é um `div` com as classes `carousel` e `slide`. Ele tem um atributo de `id` cujo valor será referenciado pelos botões que ele contém:

```

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

```

O `.slide .carousel` conterà três `divs`:

.carousel-indicators: Os pontos ou pequenas linhas representando cada uma das fotos. Um `div` de classe `carousel-indicators`

.carousel-inner: Um `div` de classe `carousel-inner` com as imagens.

Cada imagem é um `carousel-item`, que contém a imagem e um `carousel-caption`. É recomendado que todas as imagens tenham a mesma relação altura/largura.

Os botões

```

<!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
9sENB00LRn5q+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">
          
          <div class="carousel-caption">
            <h3>Ensalada reloj</h3>
          </div>

```

```

        </div>

        <div class="carousel-item">
          
          <div class="carousel-caption">
            <h3>Espárragos con
manzana</h3>
          </div>
        </div>
        <div class="carousel-item">
          
          <div class="carousel-caption">
            <h3>Ensalada de centollo</h3>
          </div>
        </div>
        <div class="carousel-item">
          
          <div class="carousel-caption">
            <h3>Sushi suelto</h3>
          </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>
```



Elementos incapacitantes

Como vimos, muitos elementos do [bootstrap](#) apoiam a classe [disabled](#) para indicar que eles têm uma aparência gráfica diferente, desabilitada.

Mas isto na verdade não os desabilita. Para desativar um elemento completamente, usamos o atributo [disabled](#).

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

Depuração

Se a página não se parece com a que estamos procurando:

- Devemos nos certificar de que todos os elementos estejam dentro de um contêiner. Normalmente devemos usar apenas um para a página.
- Use o W3C validator. Ele detectará elementos não fechados (mas não elementos fechados no lugar errado).
- Verifique se a estrutura de div está correta, se não fechamos muito cedo ou muito tarde. Um bom editor nos ajudará com isso, mostrando o código em níveis.
- Se utilizarmos o Bootstrap, não acrescente regras de CSS diretamente. A menos que estejamos seguros do que estamos fazendo.