



*sustantiva*

C O N • S E N T I D O

# DESARROLLO DE LA INTERFAZ DE DISEÑO WEB

Lección 05. Utilizando Bootstrap como Framework CSS

Implementar una interfaz de usuario web utilizando el framework CSS Bootstrap 4 para simplificar el desarrollo de layouts, contenidos y componentes.



## **A. Que es un framework CSS**

1. Frameworks CSS más utilizados
2. Conociendo Bootstrap 4 y las ventajas de utilizarlo.
3. Modificar y extender funcionalidad de Bootstrap con Sass
4. Conociendo layouts, contenidos y componentes de Bootstrap.
5. Grillas
6. Botones
7. Paneles
8. Tablas
9. Formularios
10. Navegación y menús





- 11. Conociendo componentes JavaScript
- 12. Alertas
- 13. Breadcrumb
- 14. Card
- 15. Carrusel
- 16. Tooltip
- 17. Modal
- 18. Popover



# Qué es un framework CSS

Un Framework CSS es una biblioteca de estilos genéricos que puede ser usada para implementar diseños web. Aportan una serie de utilidades que pueden ser aprovechadas frecuentemente en los distintos diseños web.

Entre las ventajas, podemos nombrar:

- Proporcionar una forma fácil y por tanto rápida de implementar diseños web.
- Nos aseguran que el diseño va a funcionar en una amplia gama de navegadores
- Nos aseguran que su código cumple ciertas normas estándar.
- Nos aseguran cierto grado de fiabilidad en la eficacia de las utilidades que nos aportan. El framework se supone que está bien probado para asegurarnos que no hay errores.

## Frameworks CSS más utilizados

1. Bootstrap
2. Bulma
3. Buefy
4. Foundation
5. Semantic



Foundation  
Start here, build everywhere.



# Conociendo Bootstrap 4 y las ventajas de utilizarlo

Bootstrap es un **framework CSS** utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces



## Ventajas de utilizar Bootstrap

1. Permite simplificar el proceso de maquetación, sirviéndonos de guía para aplicar las buenas prácticas y los diferentes estándares.
2. Puedes tener una web bien organizada de forma visual rápidamente: la curva de aprendizaje hace que su manejo sea asequible y rápido si ya sabes maquetar.
3. Permite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y Javascript.
4. Sea lo que sea que creemos, el diseño será adaptable, no importa el dispositivo, la escala o resolución.



## Ventajas de utilizar Bootstrap

5. El grid system: maquetar por columnas nunca fue tan fácil. Además, son muy configurables.
6. Se integra muy bien con las principales librerías Javascript.
7. El haber sido creado por Twitter nos da ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una comunidad muy activa creando, arreglando cosas, ofreciendo plugins y mucho más.
8. Cuenta con implementaciones externas para WordPress, Drupal, etc.
9. Nos permite usar LESS y SASS, para enriquecer aún más los estilos de la web.

## Creando temas para Bootstrap con Sass

Los temas en Bootstrap 4 se crean con variables Sass, mapas Sass y código CSS propio. No existe una hoja de estilos especial para crear temas, como sucedía en Bootstrap 3. Se recomienda utilizar los archivos Sass incluidos en Bootstrap para reutilizar todas las variables, mapas y *mixins* que puedas.

1. Estructura de archivos
2. Importando los estilos de Bootstrap
3. Valores por defecto de las variables de Bootstrap
4. Mapas
5. Funciones
6. Opciones Sass
7. Color
8. Componentes
9. Variables CSS

## Layout de Bootstrap

Presentamos aquí algunos componentes y opciones para diseñar su proyecto Bootstrap, incluidos contenedores envolventes, un potente sistema de cuadrícula, un objeto multimedia flexible y clases de utilidad receptivas.

### Contenedores

Son el elemento de diseño más básico en Bootstrap y son necesarios cuando se usa nuestro sistema de cuadrícula predeterminado. Elija entre un contenedor receptivo de ancho fijo (lo que significa que su ancho máximo cambia en cada punto de interrupción) o un ancho fluido (lo que significa que es 100% ancho todo el tiempo).

Si bien los contenedores se pueden anidar, la mayoría de los diseños no requieren un contenedor anidado.

## Contenedor fijo

Utiliza `.container` para crear un contenedor con un **ancho máximo fijo** que **se adapta al tamaño de pantalla** según los breakpoints de Bootstrap (por ejemplo: sm, md, lg, xl).

Se utiliza cuando deseas que el contenido tenga una **estructura controlada y centrada**, como en blogs, sitios corporativos o páginas con un diseño más compacto.



HTML

COPIAR CODIGO

ABRIR EN MDB EDITOR

```
1 <div class="container">
2   <!-- Content here -->
3 </div>
```



## Contenedor fluido

Utiliza `.container-fluid` para crear un contenedor de ancho completo que se extienda a lo largo de toda la ventana del navegador (viewport).

A diferencia de `.container`, que tiene un ancho máximo fijo según el breakpoint, `.container-fluid` siempre ocupa el 100% del ancho disponible, sin importar el tamaño de pantalla.



```
HTML
```

```
1 <div class="container-fluid">
2   <!-- Content here -->
3 </div>
```

COPIAR CODIGO

ABRIR EN MDB EDITOR



**Bootstrap proporciona tres tipos de contenedores, que permiten definir el ancho del contenido:**

Contenedor	Descripción
<b>.container</b>	Contenedor fijo con un max-width que varía en cada punto de quiebre ( <i>breakpoint</i> ).
<b>.container-fluid</b>	Contenedor fluido que siempre ocupa el <b>100% del ancho</b> de la ventana, sin importar el tamaño.
<b>.container-{breakpoint}</b>	Contenedor que ocupa el 100% del ancho <b>hasta alcanzar el breakpoint especificado</b> . A partir de ese punto, se comporta como .container.

## Componentes de Bootstrap

### Layout y estructura

- Sistema de grillas (.row, .col)
- Contenedores (.container, .container-fluid)
- Navegación y menús (.navbar, .nav)

### Contenido y formularios

- Botones (.btn)
- Tablas (.table)
- Formularios (.form-group, .form-control)
- Cards (.card)
- Breadcrumb (.breadcrumb)

### Componentes interactivos (JavaScript)

- Carrusel (.carousel)
- Tooltip (.tooltip)
- Modal (.modal)
- Popover (.popover)
- Alertas (.alert)

## Grillas en Bootstrap

La **grilla** de Bootstrap es un modelo de **12 columnas idénticas** (invisibles para el usuario) que facilita el diseño y desarrollo de aplicaciones y sitios web. Las 12 columnas son un punto de partida, que permiten al diseñador diagramar la página con el uso de módulos (div/contenedores) que pueden ocupar desde 1, hasta 12 columnas de ancho.

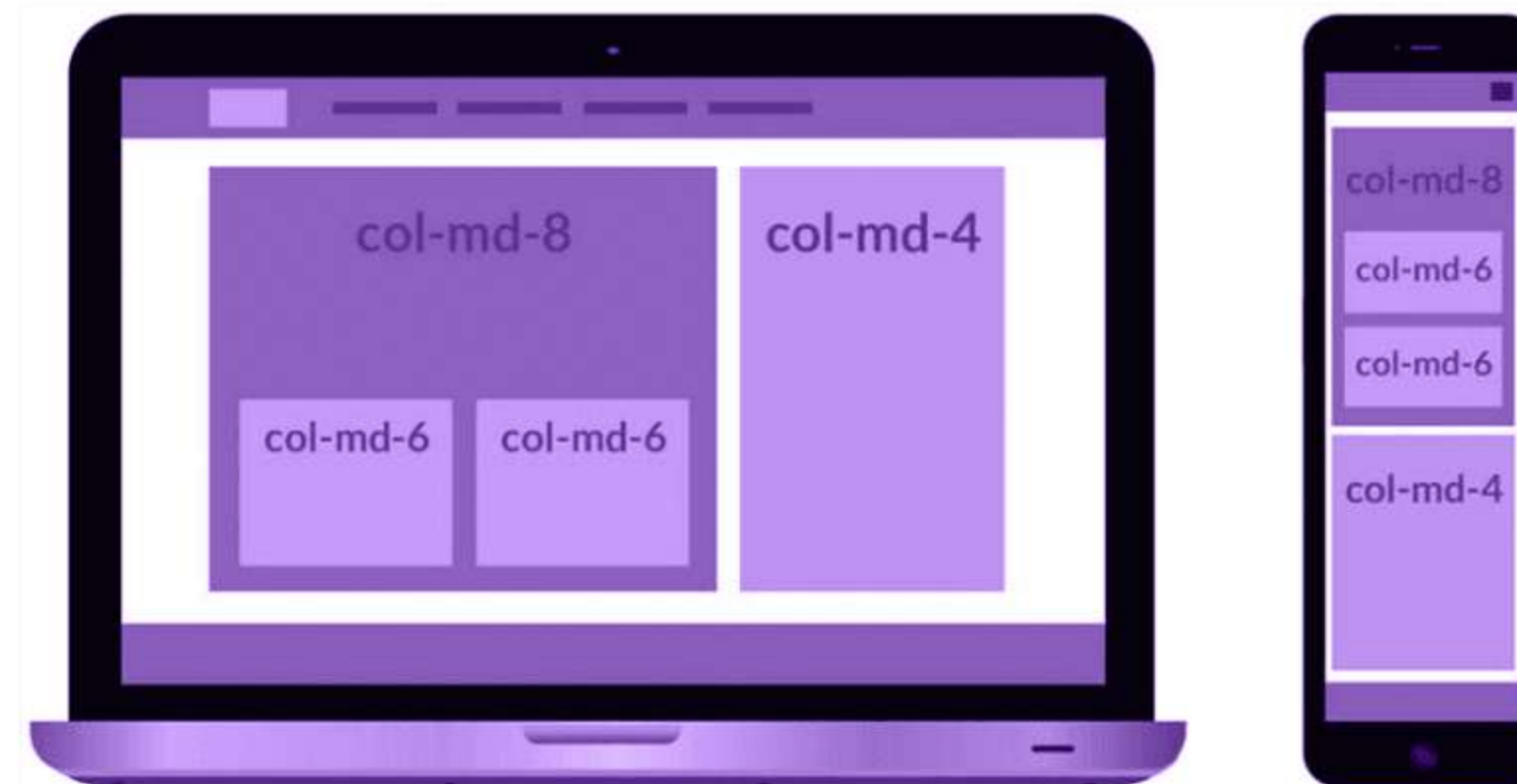
El diseñador puede combinar libremente módulos de distinto ancho en cada fila, siempre y cuando el total no supere las 12 columnas permitidas por la grilla. Cada fila (row) nueva, permite una nueva configuración de módulos.



## Rejilla De 8 Columnas + 1 Rejilla De 4 Columnas

```
<div class="col-md-8"></div>
```

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





## Botones en Bootstrap

Bootstrap incluye varios estilos de botones predefinidos, cada uno con su propio propósito semántico, con algunos extras para un mayor control.



```
<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-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```



## Paneles en Bootstrap

Un panel en bootstrap es un cuadro bordeado con algo de relleno alrededor de su contenido.

Los paneles se crean con la **.panel** clase y el contenido dentro del panel tiene una **.panel-body** clase:

Ejemplo:

```
<div class="panel panel-default">  
  <div class="panel-body">A Basic Panel</div>  
</div>
```



### Basic Panel

A Basic Panel

## Tarjetas Bootstrap

La tarjeta Bootstrap es un contenedor de contenido flexible y extensible. Incluye opciones para encabezados y pies de página, una amplia variedad de contenido, colores de fondo contextuales y potentes opciones de visualización.

A partir de **Bootstrap 4**, los componentes `.panel`, `.well` y `.thumbnail` fueron eliminados y reemplazados por un único componente flexible: `.card`.





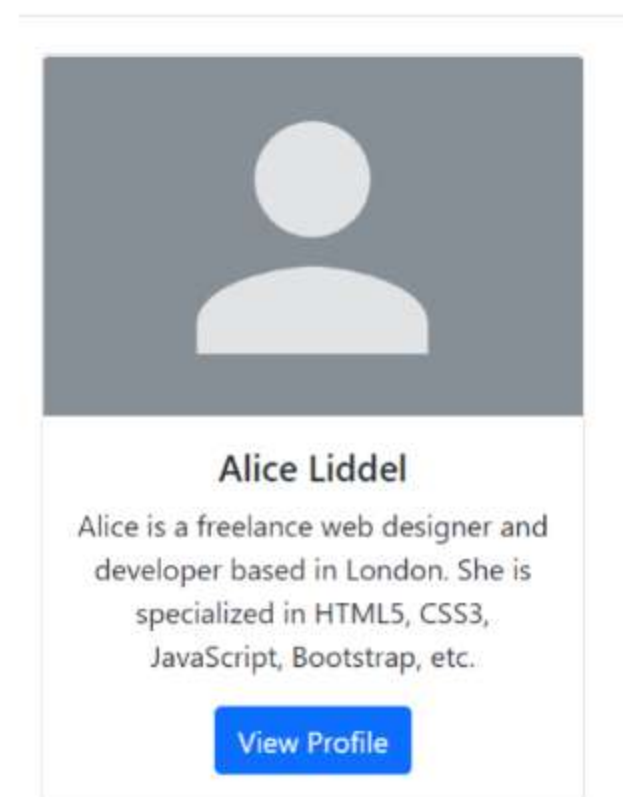
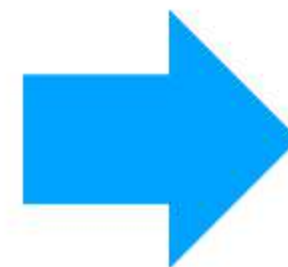
## Creación de una tarjeta básica

El marcado de la tarjeta es bastante sencillo. El envoltorio exterior requiere la clase base, mientras que el contenido se puede colocar dentro del elemento. El siguiente ejemplo le mostrará cómo crear una tarjeta con una imagen, mezclada con contenido de texto y un botón.

*.card .card-body*

Ejemplo	Pruebe este código
<pre>1 &lt;div class="card" style="width: 300px;"&gt; 2   &lt;img src="images/sample.svg" class="card-img-top" alt="..."&gt; 3   &lt;div class="card-body text-center"&gt; 4     &lt;h5 class="card-title"&gt;Alice Liddel&lt;/h5&gt; 5     &lt;p class="card-text"&gt;Alice is a freelance web designer and developer based in London. She is specialized in HTML5, CSS3, JavaScript, Bootstrap, etc.&lt;/p&gt; 6     &lt;a href="#" class="btn btn-primary"&gt;View Profile&lt;/a&gt; 7   &lt;/div&gt; 8 &lt;/div&gt;</pre>	

— El resultado del ejemplo anterior se verá algo como esto:



Una **tabla** básica Bootstrap 4 tiene un relleno y divisiones horizontales. Mediante la clase **.table** le agregamos un estilo básico a nuestra tabla.

HTML

Result

```
</html>
<body>
<div class="container">
  <div class="row">
    <div class="col">
      <table class="table">
        <thead>
          <tr>
            <th>Producto</th>
            <th>Cantidad</th>
            <th>Precio</th>
          </tr>
        </thead>
```

Producto	Cantidad	Precio
Ratón	15	100
Teclado	34	340
Pantalla	10	400

Resources

1x 0.5x 0.25x

Rerun

**Bootstrap** nos facilita la creación de un formulario con diferentes opciones de diseño y componentes personalizados para crear una amplia variedad de formularios.

Dirección de correo electrónico

Nunca compartiremos su correo electrónico con nadie más.

Contraseña

☐ Me echa un vistazo

Enviar

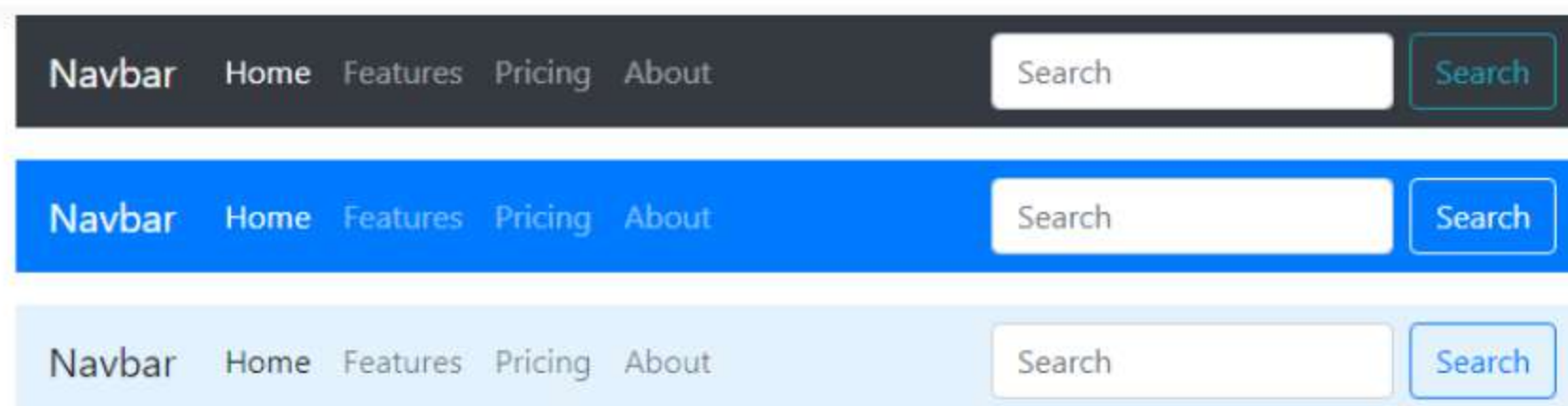


# Navegación y menús

**Bootstrap** nos facilita la creación de la barra principal de navegación de nuestra web mediante la clase **.navbar**. Esta barra se adaptará al tamaño de pantalla, mostrando los elementos colapsados en un botón en pantallas pequeñas y de forma normal para pantallas más grandes.

Una barra de navegación requiere:

- Un contenedor con la clase **.navbar**
- Una clase de expansión como **.navbar-expand-{sm|md|lg|xl}** para definir en qué tamaño dejará de colapsarse
- Clases de color como **.navbar-light** o **.navbar-dark** combinadas con **.bg-light**, **.bg-dark**, etc.



```
<nav class="navbar navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-dark bg-primary">
  <!-- Navbar content -->
</nav>

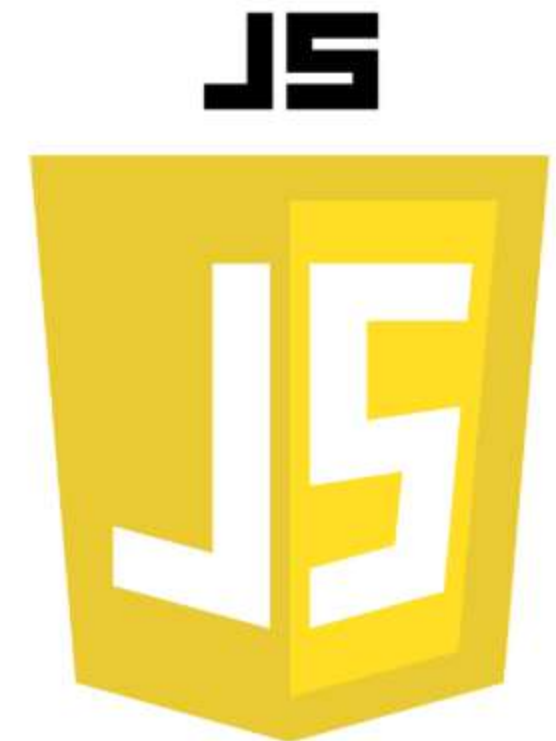
<nav class="navbar navbar-light" style="background-color: #e3f2fd;">
  <!-- Navbar content -->
</nav>
```

Copy

## Uso de Javascript en Bootstrap

Los plugins de Bootstrap se pueden incluir individualmente (usando los archivos *js/dist/\*.js* de cada uno de ellos) o todos juntos mediante el archivo *bootstrap.js* o su versión minimizada *bootstrap.min.js* (incluye solo uno de ellos, no los dos)

- Dependencias
- Atributos data
- Eventos
- API Programática
  - Funciones asíncronas y transiciones
  - Valores por defecto
- Evitando conflictos con otros plugins
- Números de versión
- Ejecutando Bootstrap con JavaScript desactivado
- La librería util.js





Las **alertas** son muy útiles para mostrar al usuario un mensaje informativo, un suceso correcto, un suceso incorrecto o erróneo y/o una **alerta** entre otros.

<b>Success!</b> This alert box indicates a successful or positive action.	×
<b>Info!</b> This alert box indicates a neutral informative change or action.	×
<b>Warning!</b> This alert box indicates a warning that might need attention.	×
<b>Danger!</b> This alert box indicates a dangerous or potentially negative action.	×
<b>Primary!</b> This alert box indicates an important action.	×
<b>Secondary!</b> This alert box indicates a less important action.	×
<b>Dark!</b> Dark grey alert box.	×
<b>Light!</b> Light grey alert box.	×

## Creación de breadcrumbs con Bootstrap

Una ruta de navegación es un esquema de navegación que indica la ubicación de la página actual al usuario dentro de un sitio web o aplicación. La navegación de *breadcrumb* puede mejorar en gran medida la accesibilidad de un sitio web que tiene una gran cantidad de páginas o una jerarquía de navegación compleja.

Puede crear diseños estáticos de migas de pan con Bootstrap simplemente usando la clase en las listas ordenadas, como se muestra en el siguiente ejemplo: `.breadcrumb`

## Creación de breadcrumbs con Bootstrap

Ejemplo

Pruebe este código

```
1 <nav>
2   <ol class="breadcrumb">
3     <li class="breadcrumb-item"><a href="#">Home</a></li>
4     <li class="breadcrumb-item"><a href="#">Products</a></li>
5     <li class="breadcrumb-item active">Accessories</li>
6   </ol>
7 </nav>
```



[Home](#) / [Products](#) / Accessories



## Creación de carruseles con Bootstrap

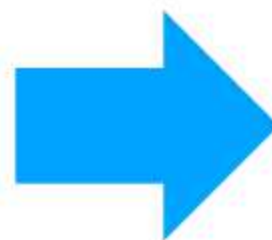
El carrusel también conocido como presentación de diapositivas o control deslizante de imágenes es una de las mejores maneras de mostrar la gran cantidad de contenidos dentro de un pequeño espacio en las páginas web. Es una presentación dinámica de contenidos donde el texto y las imágenes se hacen visibles o accesibles para el usuario recorriendo varios elementos.

El siguiente ejemplo le mostrará cómo crear un carrusel simple o una presentación de diapositivas con controles e indicadores anteriores / siguientes utilizando el complemento de carrusel Bootstrap.

```
<div id="myCarousel" class="carousel slide" data-bs-ride="carousel">
  <!-- Carousel indicators -->
  <ol class="carousel-indicators">
    <li data-bs-target="#myCarousel" data-bs-slide-to="0" class="active"></li>
    <li data-bs-target="#myCarousel" data-bs-slide-to="1"></li>
    <li data-bs-target="#myCarousel" data-bs-slide-to="2"></li>
  </ol>

  <!-- Wrapper for carousel items -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>

  <!-- Carousel controls -->
  <a class="carousel-control-prev" href="#myCarousel" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </a>
  <a class="carousel-control-next" href="#myCarousel" data-bs-slide="next">
    <span class="carousel-control-next-icon"></span>
  </a>
</div>
```



– El resultado del ejemplo anterior se verá algo como esto:



## Tooltip en Bootstrap

Un **tooltip** es una pequeña ventana emergente que aparece al **pasar el puntero** sobre un elemento (como un botón o un enlace), con el fin de mostrar **información adicional o una sugerencia contextual**.

La información sobre herramientas puede ser muy útil para los nuevos visitantes de su sitio web porque permiten al usuario conocer el propósito de los iconos y enlaces colocando el puntero del mouse sobre ellos.

## Paso 1: Agregar el marcado HTML

Para crear un tooltip, añade el atributo *data-bs-toggle="tooltip"* a cualquier elemento HTML. El texto del tooltip se define en el atributo *title*.

Ejemplo:

```
<a href="#" data-bs-toggle="tooltip" title="Algún texto"> Pasa el cursor sobremí</a>
```

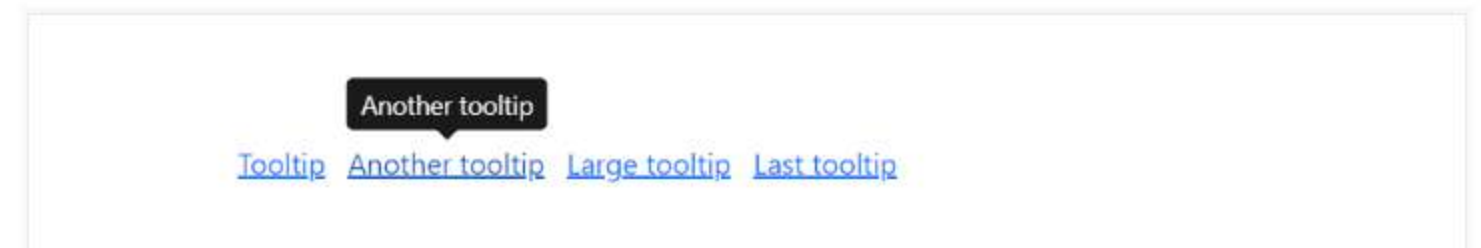


## Paso 2: Activar los tooltips con JavaScript

Los tooltips no se activan automáticamente. Debes inicializarlos con JavaScript, ya sea individualmente o de forma masiva.

Ejemplo jQuery JavaScript Prueba este código

```
1 <script>
2 document.addEventListener("DOMContentLoaded", function(){
3     var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-
4 toggle="tooltip"]'));
5     var tooltipList = tooltipTriggerList.map(function(element){
6         return new bootstrap.Tooltip(element);
7     });
8 </script>
```



## Creación de Modales con Bootstrap

Modal es básicamente un cuadro de diálogo o ventana emergente que se utiliza para proporcionar información importante al usuario o solicitar al usuario que tome las medidas necesarias antes de continuar. Los modales se utilizan ampliamente para advertir a los usuarios sobre situaciones como el tiempo de espera de la sesión o para recibir su confirmación final antes de realizar cualquier acción crítica, como guardar o eliminar datos importantes.

Puede crear fácilmente cuadros de diálogo muy inteligentes y flexibles con el complemento modal Bootstrap. En el ejemplo siguiente se utiliza la estructura básica para crear un modal simple con un encabezado, el cuerpo del mensaje y el pie de página que contiene botones de acción para el usuario.

<!-- JavaScript Code (to Show Modal on Page Load) -->

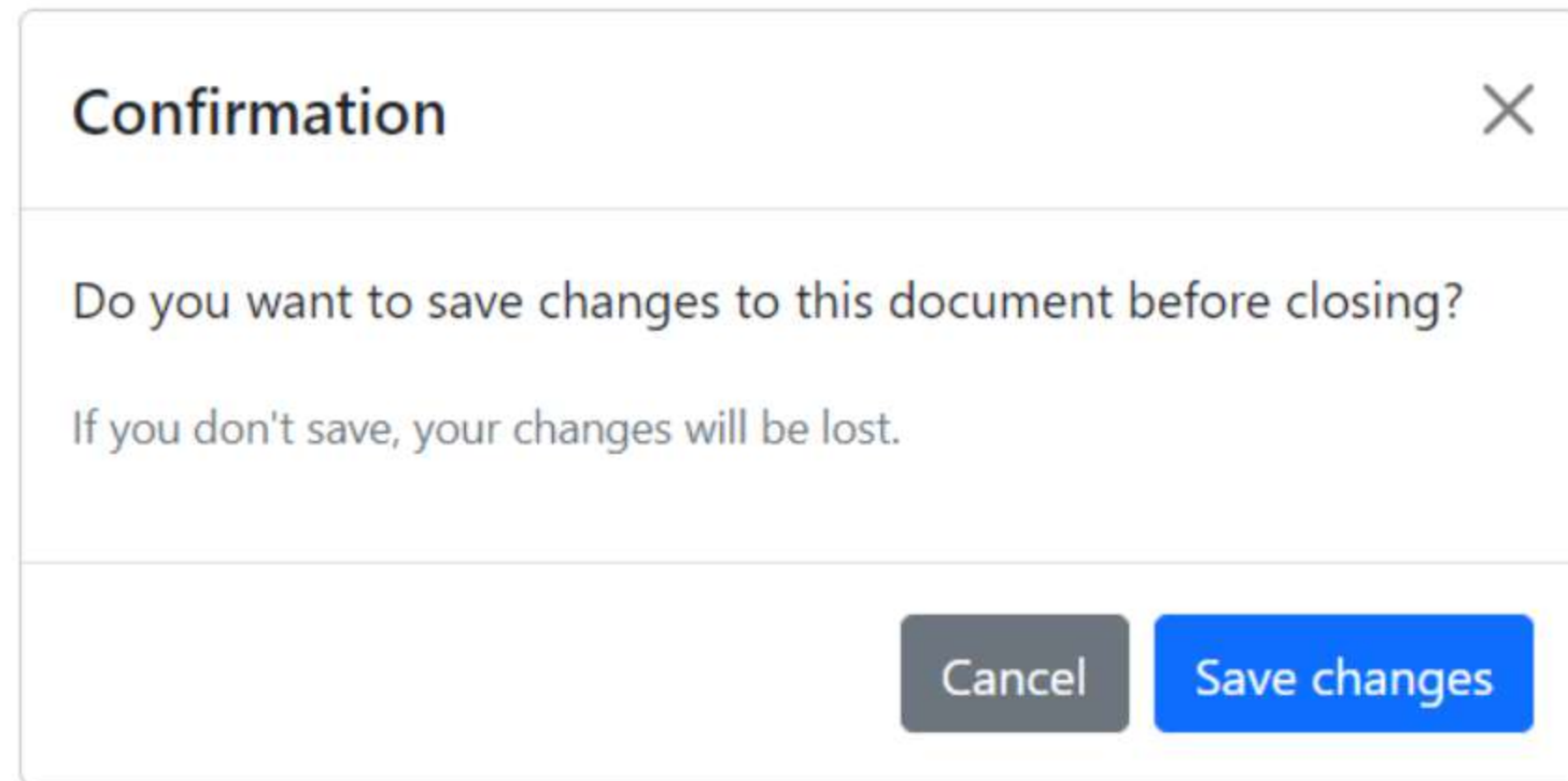
```
<script>
document.addEventListener("DOMContentLoaded", function() {
  var myModal = new bootstrap.Modal(document.getElementById("myModal"));
  myModal.show();
});
</script>
```

<!-- Modal HTML -->

```
<div id="myModal" class="modal fade" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Confirmation</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body">
        <p>Do you want to save changes to this document before closing?</p>
        <p class="text-secondary"><small>If you don't save, your changes will be lost.</small></p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

## Creación de Modales con Bootstrap: Resultado

Si prueba el ejemplo anterior, se iniciará la ventana modal automáticamente cuando el DOM esté completamente cargado a través de JavaScript. La salida se verá algo como esto:





## Popovers en Bootstrap

**Popover** es una pequeña superposición de contenido que se utiliza para mostrar información secundaria de cualquier elemento cuando un usuario hace clic en él, como los del iPad.

### Paso 1: Agregar el marcado popover

Para crear una ventana emergente, debe agregar el atributo a un elemento. Mientras que, el título de la ventana emergente y su contenido que se mostraría al desencadenar o activar el atributo se pueden especificar utilizando el atributo y respectivamente. *data-bs-toggle="popover"*  
*title**data-bs-content*

```
<tipo de botón = "botón" data-bs-toggle="popover" title="Popover title" data-bs-content="Aquí hay un contenido increíble." >Haga clic para alternar popover</button>
```

## Paso 2: Habilitar las ventanas emergentes

Las ventanas emergentes se pueden activar a través de JavaScript: simplemente llame al método Bootstrap con el `,` o cualquier selector CSS del elemento requerido en su código JavaScript. *popover()**idclass*

Puede inicializar popovers individualmente o todo de una sola vez. El siguiente código de ejemplo inicializará todas las ventanas emergentes de la página seleccionandolas por su atributo. *data-bs-toggle*

## Paso 2: Habilitar las ventanas emergentes

Ejemplo	<a href="#">jQuery</a>	<a href="#">JavaScript</a>	Pruebe este código
1	<script>		
2	\$(document).ready(function(){		
3	\$('[data-bs-toggle="popover"]').popover();		
4	});		
5	</script>		

— El resultado del ejemplo anterior se verá algo como esto:



Popover

Large popover

Popover title

A larger popover to demonstrate the max-width of the Bootstrap popover.



*substantiva*

CON • SENTIDO