

UNIVERSIDAD NACIONAL DEL CENTRO DEL PERÚ

FACULTAD DE INGENIERÍA DE SISTEMAS



MONOGRAFIA:

CSS CON BOOTSTRAP

INTEGRANTES:

CAPCHA CARHUACHIN, Dilhan Marcio

FLORES ILDEFONSO, Carlos Andre Johan

GOMEZ HUAMANLAZO, Ivan Paul

HOSPINA LORENZO, Juan Rusell

JULCA ARZAPALO, Anthoane Camila

JUNCO CHANCO, Anthony Nezaeth

DOCENTE:

Mg. JAIME SUASNABAR TERREL

HUANCAYO - PERÚ

2025

ÍNDICE

ÍNDICE.....	2
INTRODUCCIÓN.....	5
RESUMEN	6
ABSTRACT	6
PRESENTACION DEL TEMA.....	7
OBJETIVO GENERAL	8
OBJETIVOS ESPECÍFICOS	8
JUSTIFICACIÓN.....	9
METODOLOGÍA UTILIZADA	10
DESARROLLO.....	11
1. Sistema de Grid (Diseño responsivo con columnas).....	11
1.1. Concepto de Grid en Bootstrap	11
1.2. Estructura básica del Grid	11
1.3. Sistema de 12 columnas	12
1.4. Diseño responsivo con breakpoints	12
1.5. Alineación y espaciado en el Grid.....	13
1.6. Ventajas del sistema de Grid.....	13
2. Componentes predefinidos (Botones, tarjetas, modales, alertas)	13
2.1. Botones	13
2.2. Tarjetas (Cards).....	14
2.3. Modales	15
2.4. Alertas.....	15
2.5. Barras de navegación.....	16
3. Tipografía y estilos de texto personalizables.....	17
3.1. Encabezados (.h1 a .h6).....	17

3.2.	Títulos destacados (.display-1 a .display-6)	18
3.3.	Párrafos destacados (.lead)	18
3.4.	Peso de fuente (.fw-*).....	18
3.5.	Transformaciones de texto.....	19
3.6.	Alineación del texto.....	19
3.7.	Colores del texto (.text-*).....	19
3.8.	Tamaños de fuente personalizados (.fs-*)	20
3.9.	Importancia de la tipografía en el diseño web	20
4.	Formularios y validaciones integradas	21
4.1.	form-label	21
4.2.	form-control.....	21
4.3.	form-check.....	22
4.4.	form-floating.....	22
4.6.	Validación de formularios (Clases is-valid e is-invalid).....	23
5.	Clases de utilidad para espaciado, alineación y visibilidad.....	24
5.1.	Utilidad para espaciado	24
5.2.	Alineación de Texto.....	28
5.3.	Alineación vertical y horizontal con Flexbox	28
5.4.	Clases de visibilidad	32
6.	Personalización con variables CSS y SASS	35
6.1.	Variables CSS	35
6.2.	Variables SASS.....	35
6.3.	Estructura de un proyecto con SASS.....	36
6.4.	Importación.....	36
6.5.	Compilación	37
6.6.	Inclusión	38
6.7.	Variable defaults	38

6.8.	Maps	38
6.9.	Functions	40
6.10.	Mixins.....	42
7.	Animaciones y transiciones integradas.....	43
7.1.	Modales (Modal)	44
7.2.	Clase fade	44
7.3.	Clase show.....	45
7.4.	Clase collapse	45
7.5.	Clase collapsing.....	46
8.	Modificadores de color y estilos temáticos	47
8.1.	Colores contextuales.....	47
8.2.	Variantes de estilo.....	48
8.3.	Utilidades.....	49
9.	Diseño responsivo avanzado con media queries	50
9.1.	Breakpoints oficiales de Bootstrap 5	50
10.	Integración con JavaScript (Tooltips, Dropdowns, Carousels)	55
10.1.	Activación de componentes.....	56
10.2.	Componente Tooltip	56
10.3.	Componente Dropdown.....	57
10.4.	Componente Carousel	59
CONCLUSIÓN		61
REFERENCIA BIBLIOGRÁFICA.....		63

INTRODUCCIÓN

En el mundo actual del desarrollo web, donde la diversidad de dispositivos y tamaños de pantalla es cada vez mayor, se vuelve esencial contar con herramientas que permitan construir interfaces adaptables, visualmente atractivas y funcionales. En este contexto, Bootstrap 5, el popular framework de código abierto desarrollado originalmente por Twitter, se posiciona como una de las soluciones más efectivas y utilizadas por desarrolladores de todo el mundo.

Bootstrap no solo proporciona un sistema de diseño basado en Grid, sino que también integra una amplia variedad de componentes predefinidos, clases de utilidad, tipografías personalizables, formularios con validaciones, animaciones y compatibilidad con JavaScript. Esto permite una aceleración considerable en el proceso de diseño y desarrollo frontend, mejorando la experiencia del usuario y la eficiencia del trabajo del desarrollador.

La presente monografía tiene como objetivo analizar en profundidad los principales elementos que conforman Bootstrap 5, desde su sistema de columnas hasta su capacidad de personalización mediante SASS, con el fin de comprender su utilidad y aplicabilidad en proyectos reales de diseño web moderno.

RESUMEN

Este trabajo presenta un estudio detallado de las principales características del framework Bootstrap 5, centrando la atención en su sistema de diseño responsivo, los componentes UI, la personalización mediante SASS y la integración con JavaScript. A través del análisis teórico y práctico de temas como el sistema Grid de 12 columnas, botones, tarjetas, formularios, tipografía, animaciones y herramientas de personalización, se demuestra cómo Bootstrap permite crear interfaces web adaptables y modernas con mayor eficiencia. El documento también presenta ejemplos de código y buenas prácticas para el uso profesional del framework en entornos de desarrollo frontend.

ABSTRACT

This monograph presents a comprehensive study of the main features of the Bootstrap 5 framework, focusing on its responsive design system, UI components, SASS customization, and JavaScript integration. Through a theoretical and practical analysis of topics such as the 12-column Grid system, buttons, cards, forms, typography, animations, and customization tools, this paper shows how Bootstrap enables the efficient creation of modern and responsive web interfaces. It also includes code examples and best practices for professional usage in frontend development environments.

PRESENTACION DEL TEMA

El desarrollo web moderno requiere herramientas que permitan crear interfaces adaptables, rápidas y consistentes entre diferentes dispositivos y navegadores. Bootstrap, desde su lanzamiento, ha revolucionado la manera en que los desarrolladores frontend construyen sus sitios y aplicaciones, ofreciendo una solución integral que combina facilidad de uso, flexibilidad y potencia visual.

En su versión 5, Bootstrap introduce mejoras significativas como la eliminación de dependencias de jQuery, una mejor personalización con SASS, mayor control sobre breakpoints y una estructura más modular. Este documento presenta un recorrido por las principales secciones que conforman Bootstrap 5, enfocándose no solo en su uso técnico, sino también en las ventajas que ofrece en términos de diseño, productividad y experiencia de usuario.

OBJETIVO GENERAL

Investigar y comprender las funcionalidades esenciales del framework Bootstrap 5, explorando su sistema de diseño responsivo, sus componentes reutilizables, capacidades de personalización y su integración con tecnologías como JavaScript.

OBJETIVOS ESPECÍFICOS

- Analizar el sistema de Grid de 12 columnas como base para un diseño adaptable.
- Estudiar los componentes visuales predefinidos como botones, tarjetas, modales y alertas.
- Explorar las opciones de personalización de tipografía y estilos de texto.
- Implementar formularios con controles estandarizados y validaciones integradas.
- Aplicar clases de utilidad para espaciado, alineación y visibilidad.
- Personalizar estilos mediante variables CSS y SASS.
- Usar animaciones y transiciones para mejorar la experiencia de usuario.
- Usar modificadores de color y estilos temáticos.
- Utilizar media queries para lograr un diseño más avanzado y adaptativo.
- Integrar componentes dinámicos como tooltips, dropdowns y carruseles con JavaScript.

JUSTIFICACIÓN

La elección de Bootstrap 5 como objeto de estudio en esta monografía se fundamenta en su relevancia actual en el mundo del desarrollo web. Gracias a su arquitectura modular, su sistema de diseño responsivo y su colección de componentes reutilizables, Bootstrap permite acelerar significativamente el proceso de desarrollo y mejorar la calidad de los productos digitales.

Comprender a fondo su estructura y funcionamiento no solo facilita la creación de proyectos web profesionales, sino que también brinda al desarrollador la capacidad de adaptar y escalar sus soluciones según las necesidades del cliente o del mercado. Asimismo, el uso de buenas prácticas con Bootstrap promueve la accesibilidad, la compatibilidad cross-browser y la experiencia de usuario coherente en distintas plataformas.

METODOLOGÍA UTILIZADA

Para la elaboración de esta monografía se adoptó una metodología descriptiva y aplicada, combinando el análisis teórico con la implementación práctica de ejemplos funcionales.

1. Revisión documental: Se consultaron documentos oficiales de Bootstrap 5, como la documentación técnica.
2. Estudio práctico: Se implementaron ejemplos funcionales en proyectos web reales y de prueba, con el objetivo de validar cada concepto y funcionalidad explicada.

DESARROLLO

1. Sistema de Grid (Diseño responsivo con columnas)

Uno de los pilares fundamentales del framework Bootstrap es su sistema de **Grid** o **rejilla**, el cual permite estructurar el contenido de una página de manera organizada, flexible y adaptable a diferentes tamaños de pantalla. Este sistema facilita el diseño de interfaces responsivas, permitiendo que los elementos se reorganicen automáticamente según el dispositivo desde el cual se accede.

1.1. Concepto de Grid en Bootstrap

El **Grid System** de Bootstrap está basado en una estructura de **filas y columnas**, utilizando un diseño de **12 columnas** por fila como base. Esta organización permite dividir el ancho de una página en partes proporcionales y posicionar elementos dentro de estas divisiones, ajustándose automáticamente a tamaños como móviles, tablets, laptops o pantallas grandes.

1.2. Estructura básica del Grid

El sistema se basa en tres elementos principales:

- **.container** o **.container-fluid**: Envuelve todo el contenido del grid y define sus márgenes.
 - **.container** tiene un ancho fijo adaptativo.
 - **.container-fluid** se extiende al 100% del ancho de la pantalla.
- **.row**: Representa una fila que agrupa columnas. Siempre debe estar contenida dentro de un **.container**.
- **.col**: Define una columna. Puede usarse sola o con variantes como **.col-6**, **.col-md-4**, etc.

```
<div class="container">
  <div class="row">
    <div class="col">Columna 1</div>
    <div class="col">Columna 2</div>
    <div class="col">Columna 3</div>
  </div>
</div>
```

1.3. Sistema de 12 columnas

Cada fila en Bootstrap se divide virtualmente en **12 columnas**. Los desarrolladores pueden asignar cuántas columnas ocupa cada elemento utilizando clases como:

- `.col-4`: El elemento ocupará 4 de las 12 columnas disponibles.
- `.col-6`: Ocupará la mitad del ancho total.
- `.col-12`: Ocupará el 100% del ancho de la fila.

Esto permite diseñar estructuras como mitades, tercios, cuartos, etc.

```
<div class="row">
  <div class="col-4">1/3 del espacio</div>
  <div class="col-8">2/3 del espacio</div>
</div>
```

1.4. Diseño responsivo con breakpoints

Bootstrap integra puntos de quiebre (*breakpoints*) para adaptar las columnas según el tamaño de pantalla:

Tamaño de pantalla	Prefijo usado	Ancho mínimo
Extra pequeño	Col-	< 576 px
Pequeño	Col-sm-	≥ 576 px
Mediano	Col-md-	≥ 786 px
Grande	Col-lg-	≥ 992 px
Extra grande	Col-xl-	≥ 1200 px
XXL	Col-xxl-	≥ 1400 px

1.5. Alineación y espaciado en el Grid

Además de la estructura básica, Bootstrap permite **alinear vertical u horizontalmente** el contenido dentro del grid y añadir **espaciado** usando clases como:

- `.justify-content-center`, `.justify-content-end`: Alineación horizontal.
- `.align-items-center`, `.align-items-start`: Alineación vertical.
- `.g-0`, `.g-1`, ..., `.g-5`: Control del espacio entre columnas (gap).

1.6. Ventajas del sistema de Grid

- **Responsividad automática:** Se adapta a todo tipo de dispositivos.
- **Flexibilidad total:** Permite dividir el espacio en proporciones exactas.
- **Facilidad de uso:** No requiere conocimientos avanzados de CSS.
- **Consistencia:** Mantiene el diseño organizado y alineado.

2. Componentes predefinidos (Botones, tarjetas, modales, alertas)

Bootstrap ofrece una amplia gama de componentes visuales predefinidos que facilitan la creación de interfaces modernas y responsivas. Dichos componentes se les da estilos con CSS y pueden personalizarse (Bootstrap, 2025). Algunos de los más utilizados son:

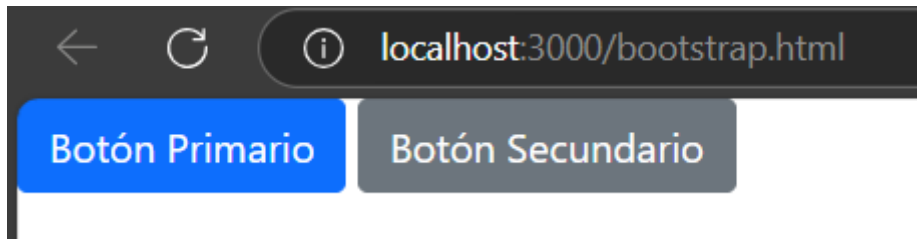
2.1. Botones

Son elementos que permiten realizar al usuario acciones. Bootstrap proporciona clases como `.btn`, `.btn-primary`, `.btn-secondary`, etc., para crear botones con diferentes estilos y tamaños.

Ejemplo:

```
bootstrap.html X
bootstrap.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Botones Bootstrap</title>
7   <!-- Vincula Bootstrap CSS -->
8   <link href="https://stackpath.bootstrapcdn.com/bootstrap/5.1.0/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <button type="button" class="btn btn-primary">Botón Primario</button>
12   <button type="button" class="btn btn-secondary">Botón Secundario</button>
13
14   <!-- Vincula Bootstrap JS (opcional para interacciones) -->
15   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
16   <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.1.0/js/bootstrap.bundle.min.js"></script>
17 </body>
18 </html>
19
```

Resultado:



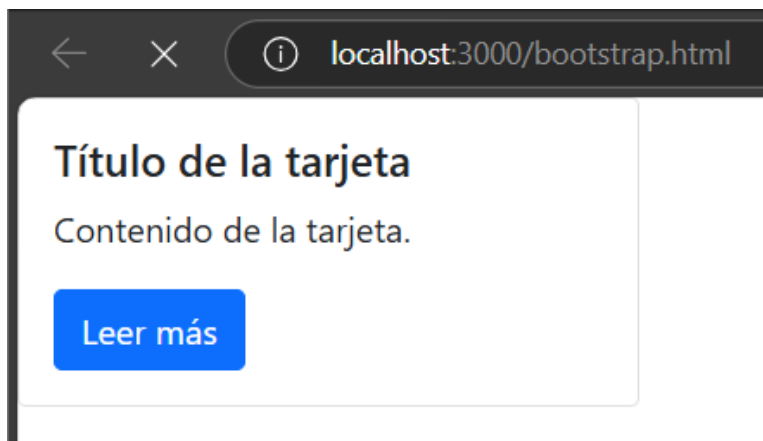
2.2. Tarjetas (Cards)

Son los que muestran contenido organizado, como imágenes, texto y botones. Se usan clases como `.card`, `.card-body`, `.card-title`.

Ejemplo:

```
bootstrap.html X
bootstrap.html > html > body
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta Bootstrap</title>
7   <!-- Vincula Bootstrap CSS-->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <div class="card" style="width: 18rem;">
12     
13     <div class="card-body">
14       <h5 class="card-title">Título de la tarjeta</h5>
15       <p class="card-text">Contenido de la tarjeta.</p>
16       <a href="#" class="btn btn-primary">Leer más</a>
17     </div>
18   </div>
19
20   <!-- Vincula Bootstrap JS-->
21   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
22   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
23 </body>
24 </html>
25
```

Resultado:



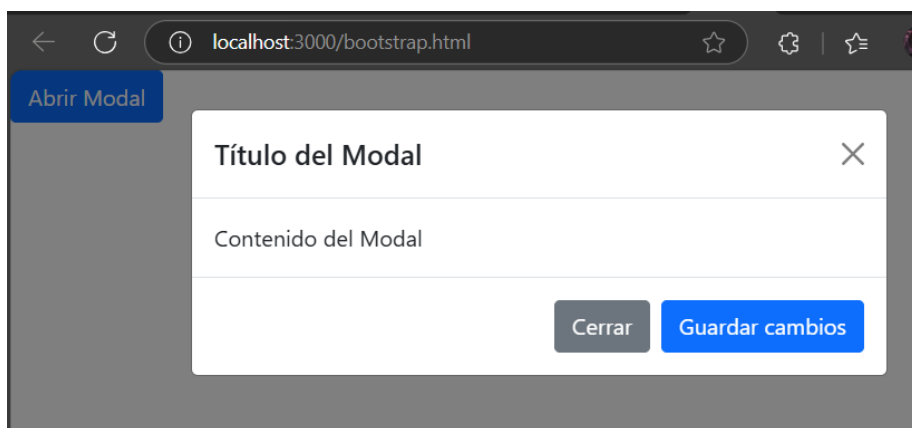
2.3. Modales

Ventanas emergentes para interacciones como formularios o alertas. Se crean con `.modal` y se activan mediante JavaScript o atributos `data`.

Ejemplo:

```
bootstrap.html X
bootstrap.html > html > head
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Modal Bootstrap</title>
7   <!-- Vincula Bootstrap CSS-->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <!-- Botón para abrir modal -->
12   <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">
13     Abrir Modal
14   </button>
15   <!-- Modal -->
16   <div class="modal fade" id="myModal" tabindex="-1" aria-labelledby="myModallabel" aria-hidden="true">
17     <div class="modal-dialog">
18       <div class="modal-content">
19         <div class="modal-header">
20           <h5 class="modal-title" id="myModallabel">Título del Modal</h5>
21           <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
22         </div>
23         <div class="modal-body">
24           Contenido del Modal
25         </div>
26         <div class="modal-footer">
27           <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
28           <button type="button" class="btn btn-primary">Guardar cambios</button>
29         </div>
30       </div>
31     </div>
32   </div>
33   <!-- Vincula Bootstrap JS-->
34   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
35   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
36 </body>
37 </html>
```

Resultado:



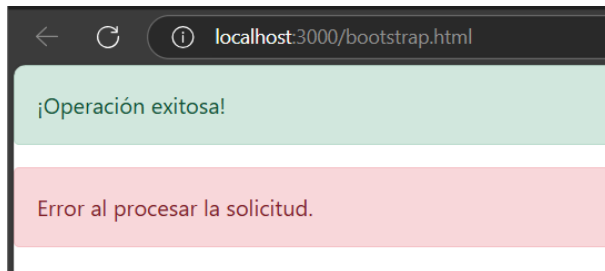
2.4. Alertas

Mensajes informativos o de advertencia con clases como `.alert`, `.alert-success`, `.alert-danger`.

Ejemplo:

```
bootstrap.html X
bootstrap.html > html > body
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Alertas Bootstrap</title>
7   <!-- Vincula Bootstrap CSS -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <!-- Alerta de éxito -->
12   <div class="alert alert-success" role="alert">
13     ¡Operación exitosa!
14   </div>
15
16   <!-- Alerta de error -->
17   <div class="alert alert-danger" role="alert">
18     Error al procesar la solicitud.
19   </div>
20   <!-- Vincula Bootstrap JS -->
21   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
22   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
23 </body>
24 </html>
25
```

Resultado:



2.5. Barras de navegación

Son componentes usados normalmente en la parte superior de la página.

Ejemplo:

```
bootstrap.html X
bootstrap.html > html > head
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Navegación Bootstrap</title>
7   <!-- Vincula Bootstrap CSS -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <nav class="navbar navbar-expand-lg navbar-light bg-light">
12     <a class="navbar-brand" href="#">Mi Marca</a>
13     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" ar
14       <span class="navbar-toggler-icon"></span>
15     </button>
16     <div class="collapse navbar-collapse" id="navbarNav">
17       <ul class="navbar-nav">
18         <li class="nav-item active">
19           <a class="nav-link" href="#">Inicio <span class="sr-only">(actual)</span></a>
20         </li>
21         <li class="nav-item">
22           <a class="nav-link" href="#">Enlace</a>
23         </li>
24       </ul>
25     </div>
26   </nav>
27   <!-- Vincula Bootstrap JS -->
28   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
29   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
30 </body>
31 </html>
32
```


Resultado:



Estos componentes son responsivos y se adaptan a diferentes dispositivos gracias a la cuadrícula de Bootstrap.

3. Tipografía y estilos de texto personalizables

La tipografía es un elemento esencial en el diseño de interfaces web, ya que influye directamente en la legibilidad, la jerarquía visual y la estética general de un sitio. Bootstrap, como framework frontend, incorpora un conjunto de clases tipográficas predefinidas que permiten aplicar estilos de forma rápida, coherente y personalizable, sin necesidad de escribir reglas CSS complejas.

Estas clases permiten no solo definir tamaños y pesos tipográficos, sino también alinear textos, aplicar transformaciones, controlar colores temáticos y adaptar los estilos de acuerdo con el diseño responsivo. Esto permite a los desarrolladores crear páginas web más accesibles, estéticas y estructuradas.

3.1. Encabezados (.h1 a .h6)

Bootstrap permite aplicar clases .h1 hasta .h6 a cualquier elemento HTML para que adopte visualmente el tamaño de un encabezado correspondiente, sin alterar la semántica del HTML.

Ejemplo:

```
<!-- Encabezados h1 a h6 -->
<h1 class="h1">Esto es un .h1</h1>
<h2 class="h2">Esto es un .h2</h2>
<h3 class="h3">Esto es un .h3</h3>
<h4 class="h4">Esto es un .h4</h4>
<h5 class="h5">Esto es un .h5</h5>
<h6 class="h6 mb-4">Esto es un .h6</h6>
```

Estas clases son útiles para mantener una jerarquía visual adecuada sin necesidad de utilizar siempre las etiquetas `<h1>` a `<h6>`, lo cual es especialmente útil en componentes personalizados.

3.2. Títulos destacados (`.display-1` a `.display-6`)

Bootstrap también ofrece una familia de clases `display`, las cuales son útiles para títulos más llamativos que se utilizan en páginas de presentación, encabezados principales o secciones destacadas.

Ejemplo:

```
<!-- Display titles -->
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
<h1 class="display-5">Display 5</h1>
<h1 class="display-6 mb-4">Display 6</h1>
```

Estas clases aplican un tamaño de fuente considerablemente más grande y un espaciado más amplio que los encabezados tradicionales.

3.3. Párrafos destacados (`.lead`)

La clase `lead` se utiliza para resaltar párrafos que cumplen una función introductoria. Se aplican estilos que aumentan el tamaño de fuente y el espaciado entre líneas, mejorando la lectura en bloques de texto iniciales.

Ejemplo:

```
<!-- Lead text -->
<p class="lead">
  Este es un párrafo con la clase .lead. Se usa para introducir contenido importante o destacado.
</p>
```

3.4. Peso de fuente (`.fw-*`)

Bootstrap incluye clases utilitarias para controlar el peso (grosor) del texto, sin tener que usar etiquetas como ``. Estas clases son:

- `.fw-bold` → texto en negrita.
- `.fw-light` → peso ligero.

Ejemplo:

```
<!-- Font weight -->
<p class="fw-bold">Este texto es <strong>negrita</strong> con <code>.fw-bold</code>.</p>
<p class="fw-light">Este texto es <em>liviano</em> con <code>.fw-light</code>.</p>
```

3.5. Transformaciones de texto

Bootstrap permite transformar el texto directamente desde el HTML con las siguientes clases:

- `.text-uppercase`: Convierte todo a mayúsculas.
- `.text-lowercase`: Convierte todo a minúsculas.
- `.text-capitalize`: Convierte la primera letra de cada palabra a mayúscula.

Ejemplo:

```
<!-- Mayúsculas -->
<p class="text-uppercase">Este texto está en mayúsculas con <code>.text-uppercase</code>.</p>
```

3.6. Alineación del texto

El framework también facilita la alineación del contenido textual mediante clases como:

- `.text-start`: Alineación a la izquierda (predeterminado).
- `.text-center`: Alineación centrada.
- `.text-end`: Alineación a la derecha.

Ejemplo:

```
<!-- Alineación -->
<p class="text-center">Este texto está <strong>centrado</strong> con <code>.text-center</code>.</p>
<p class="text-end">Este texto está <strong>alineado a la derecha</strong> con <code>.text-end</code>.</p>
```

Estas clases son responsivas y pueden adaptarse a diferentes tamaños de pantalla usando variantes como `.text-md-center`, `.text-lg-end`, etc.

3.7. Colores del texto (`.text-*`)

Bootstrap ofrece una paleta de colores basada en su sistema temático, que puede aplicarse al texto mediante clases como:

- `.text-primary` (azul)
- `.text-success` (verde)

- `.text-danger` (rojo)
- `.text-warning` (amarillo)
- `.text-muted` (gris claro)

Ejemplo:

```
<!-- Colores temáticos -->
<p class="text-primary">Texto en azul: <code>.text-primary</code></p>
<p class="text-success">Texto en verde: <code>.text-success</code></p>
<p class="text-danger">Texto en rojo: <code>.text-danger</code></p>
<p class="text-warning">Texto en amarillo: <code>.text-warning</code></p>
<p class="text-muted">Texto gris suave: <code>.text-muted</code></p>
```

Esto permite destacar o categorizar visualmente la información sin tener que escribir reglas de estilo personalizadas.

3.8. Tamaños de fuente personalizados (`.fs-*`)

Bootstrap proporciona clases `.fs-1` a `.fs-6` que permiten ajustar el tamaño de la fuente sin necesidad de usar encabezados o estilos CSS. La clase `.fs-1` es la más grande, y `.fs-6` la más pequeña.

Ejemplo:

```
<!-- Tamaños de fuente con .fs-* -->
<p class="fs-1">Texto tamaño .fs-1</p>
<p class="fs-2">Texto tamaño .fs-2</p>
<p class="fs-3">Texto tamaño .fs-3</p>
<p class="fs-4">Texto tamaño .fs-4</p>
<p class="fs-5">Texto tamaño .fs-5</p>
<p class="fs-6">Texto tamaño .fs-6</p>
```

3.9. Importancia de la tipografía en el diseño web

El uso adecuado de las clases tipográficas en Bootstrap permite mantener la coherencia visual y mejorar la experiencia del usuario. Al establecer jerarquías claras, estilos llamativos y estructuras legibles, se facilita la navegación y comprensión del contenido. Además, al estar integradas dentro del sistema responsivo de Bootstrap, estas clases garantizan que los estilos se adapten adecuadamente a cualquier dispositivo, lo que es esencial en el diseño moderno.

4. Formularios y validaciones integradas

Los formularios en Bootstrap, son ampliamente conocido por facilitar el diseño web responsivo y moderno. Las clases que exploraremos son: form-label, form-control, form-check, form-floating y form-select. Estas clases permiten estructurar, estilizar y mejorar la experiencia del usuario al interactuar con formularios en páginas web. Asimismo, también trataremos el tema de las **validaciones visuales** en Bootstrap, usando únicamente las clases que este ofrece por defecto, como is-valid e is-invalid, sin recurrir a validaciones con JavaScript personalizado. Con esta revisión, buscamos entender cómo Bootstrap mejora la usabilidad y accesibilidad de los formularios web con una mínima cantidad de código.

4.1. form-label

La clase form-label se utiliza para estilizar las etiquetas (<label>) de los formularios. Su propósito principal es asegurar una presentación coherente, alineada y con la tipografía adecuada según el diseño general del sitio.

Características:

- Se aplica directamente sobre el elemento <label>.
- Proporciona el estilo adecuado en términos de fuente, tamaño y espaciado.
- Suele ir acompañada de un for que referencia el id del input asociado, lo que mejora la accesibilidad.

Ejemplo:

```
<label for="email" class="form-label">Correo electrónico</label>  
<input type="email" class="form-control" id="email">
```

4.2. form-control

La clase form-control se aplica a campos de entrada como <input>, <textarea> y <select> (aunque para <select> es más común form-select). Sirve para darles un estilo uniforme, moderno y adaptativo.

Características:

- Estiliza los inputs con bordes redondeados, padding y fuentes legibles.

- Se adapta automáticamente al ancho del contenedor padre.
- Permite una experiencia de usuario consistente en todos los navegadores.

Ejemplo:

```
<input type="text" class="form-control" placeholder="Ingrese su nombre">
```

4.3. form-check

form-check es una clase de utilidad que se usa para organizar y presentar casillas de verificación (<input type="checkbox">) y botones de opción (<input type="radio">).

Características:

- Se aplica a un contenedor <div> que agrupa el input y su label.
- Ayuda a espaciar correctamente y alinear los elementos.
- Puede combinarse con form-check-inline para mostrar los elementos en línea.

Ejemplo (checkbox):

```
<div class="form-check">
  <input class="form-check-input" type="checkbox" id="flexCheckDefault">
  <label class="form-check-label" for="flexCheckDefault">
    Acepto los términos y condiciones
  </label>
</div>
```

4.4. form-floating

La clase form-floating permite crear campos flotantes donde la etiqueta (<label>) se posiciona dentro del campo de entrada y luego se eleva automáticamente cuando el usuario comienza a escribir o el campo tiene contenido.

Características:

- Mejora el diseño estético de los formularios.
- Reduce el espacio ocupado por las etiquetas externas.
- Requiere que el campo de entrada tenga un placeholder.

Ejemplo:

```
<div class="form-floating">
  <input type="email" class="form-control" id="floatingInput" placeholder="nombre@ejemplo.com">
  <label for="floatingInput">Correo electrónico</label>
</div>
```

4.5. form-select

La clase form-select se usa exclusivamente para estilizar listas desplegables (<select>). Reemplaza al uso de form-control en estos casos y les da un diseño más moderno y coherente.

Características:

- Mejora la apariencia de los desplegables nativos.
- Permite un diseño adaptativo.
- Compatible con múltiples opciones y con la propiedad multiple.

Ejemplo:

```
<select class="form-select" aria-label="Seleccione un país">
  <option selected>Elija un país</option>
  <option value="1">Perú</option>
  <option value="2">México</option>
  <option value="3">Argentina</option>
</select>
```

4.6. Validación de formularios (Clases is-valid e is-invalid)

Bootstrap incluye clases para proporcionar retroalimentación visual sobre la validez de los campos del formulario sin necesidad de JavaScript adicional.

- **is-valid:** indica que el campo es válido. Se muestra con un borde verde y puede ir acompañado de un mensaje positivo.
- **is-invalid:** indica que el campo es inválido. Se muestra con un borde rojo y puede acompañarse de un mensaje de error.

Estas clases se aplican directamente sobre los inputs, y los mensajes se colocan generalmente en elementos <div> con clases auxiliares como valid-feedback e invalid-feedback.

Ejemplo:

```
<div class="mb-3">
  <label for="username" class="form-label">Usuario</label>
  <input type="text" class="form-control is-invalid" id="username">
  <div class="invalid-feedback">
    Este campo es obligatorio.
  </div>
</div>

<div class="mb-3">
  <label for="email" class="form-label">Correo</label>
  <input type="email" class="form-control is-valid" id="email">
  <div class="valid-feedback">
    ¡Se ve bien!
  </div>
</div>
```

5. Clases de utilidad para espaciado, alineación y visibilidad

Bootstrap no solo proporciona componentes prediseñados, sino que también incluye un conjunto robusto de clases de utilidad que permiten un control preciso sobre el diseño, espaciado, alineación, visibilidad y comportamiento adaptativo de los elementos. Estas clases son fundamentales para mejorar la productividad del desarrollador y garantizar una presentación consistente en distintos dispositivos.

5.1. Utilidad para espaciado

El espaciado es esencial para una buena experiencia de usuario. Bootstrap facilita el ajuste de márgenes (m) y relleno (p) mediante clases predefinidas, eliminando la necesidad de escribir CSS personalizado.

Sintaxis General

$\{m|p\}\{lado\}-\{valor\}$

- m = margen, p = padding
- lado puede ser:
 - t (top), b (bottom), l (left), r (right),
 - x (horizontal: left + right), y (vertical: top + bottom),

- sin especificador = aplica a todos los lados
- valor: de 0 a 5 (multiplicadores de 0.25rem)
 - Ejemplo: p-3 equivale a padding: 1rem;

La alineación es un aspecto clave del diseño web, ya que permite estructurar visualmente los contenidos de forma clara, atractiva y funcional. Bootstrap ofrece clases para alinear tanto texto como elementos dentro de contenedores, utilizando tanto CSS tradicional como Flexbox.

Ejemplo:

```
01espaciado.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Espaciado</title>
7  |   <link rel="stylesheet" href="css/bootstrap.min.css">
8  </head>
9  <body>
10 |   <script src="js/bootstrap.bundle.min.js"></script>
11
12 |   <div class="m-0 p-0">Sin espaciado</div>
13 |   <div class="mt-3 mb-3 px-4 py-2 bg-light border shadow-sm">
14 | |   Espaciado vertical y horizontal personalizado
15 | </div>
16 |   <div class="p-3 bg-dark text-white">
17 | |   Otro bloque debajo
18 | </div>
19
20 </body>
21 </html>
```

ESTRUCTURA GENERAL

<!DOCTYPE html>

<html lang="en">

- Esto indica que el documento es HTML5 y que el idioma principal es inglés.

<head> SECCIÓN

<head>

<meta charset="UTF-8">

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

`<title>Espaciado</title>`

`<link rel="stylesheet" href="css/bootstrap.min.css">`

`</head>`

- `<meta charset="UTF-8">`: Define la codificación de caracteres como UTF-8 (soporta casi todos los caracteres).
- `<meta name="viewport"...>`: Hace que el diseño sea responsivo en dispositivos móviles.
- `<title>`: Define el título de la pestaña del navegador.
- `<link rel="stylesheet"...>`: Carga el archivo de estilos de Bootstrap para aplicar sus clases CSS.

`<body>` SECCIÓN

`<body>`

`<script src="js/bootstrap.bundle.min.js"></script>`

- Carga el archivo JavaScript de Bootstrap, que incluye componentes interactivos como modales, tooltips, etc.

DIV 1: Sin espaciado

`<div class="m-0 p-0">Sin espaciado</div>`

- `m-0`: Elimina **todos los márgenes**.
- `p-0`: Elimina **todo el padding** (relleno interno).
- Resultado: Este div no tiene separación con otros elementos ni margen interno.

♦ **DIV 2: Espaciado personalizado**

`<div class="mt-3 mb-3 px-4 py-2 bg-light border shadow-sm">`

Espaciado vertical y horizontal personalizado

`</div>`

- mt-3: Margen superior (margin-top) de tamaño 3.
- mb-3: Margen inferior (margin-bottom) de tamaño 3.
- px-4: Padding horizontal (izquierda y derecha) de tamaño 4.
- py-2: Padding vertical (arriba y abajo) de tamaño 2.
- bg-light: Fondo claro.
- border: Añade un borde.
- shadow-sm: Sombra ligera alrededor del bloque.
- Resultado: Un bloque estilizado con espaciado interno y externo, fondo claro, borde y sombra.

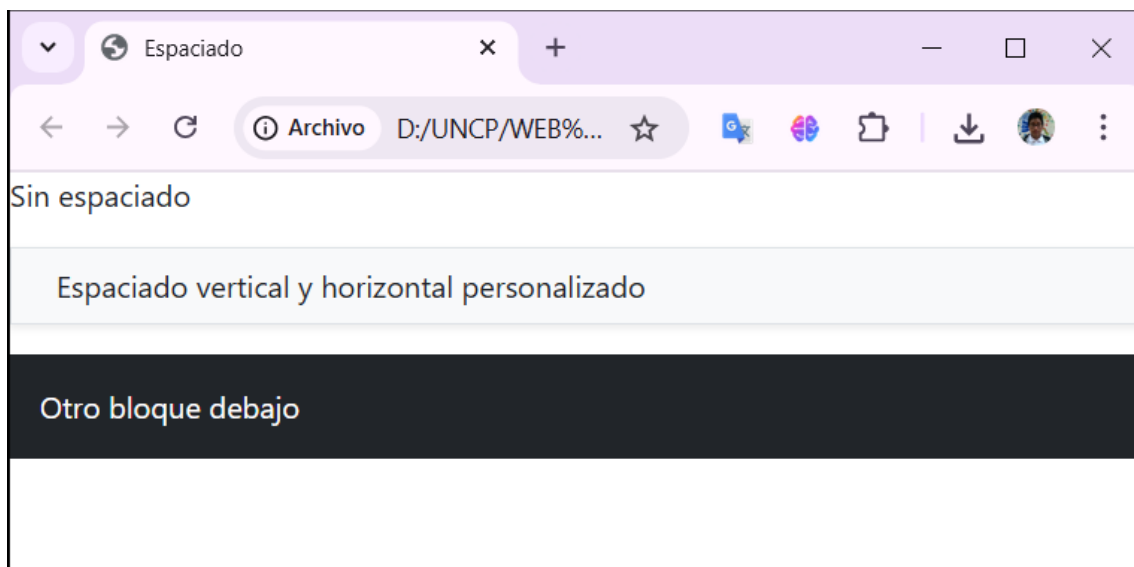
DIV 3: Otro bloque debajo

```
<div class="p-3 bg-dark text-white">
```

Otro bloque debajo

```
</div>
```

- p-3: Padding de tamaño 3 por todos lados.
- bg-dark: Fondo oscuro.
- text-white: Texto blanco.
- Resultado: Un bloque con fondo oscuro y texto blanco, con relleno.



5.2. Alineación de Texto

Bootstrap simplifica la alineación de texto mediante las siguientes clases:

Clase	Función
text-start	Alinea el texto a la izquierda
text-center	Centra el texto
text-end	Alinea el texto a la derecha

Estas clases pueden combinarse con *breakpoints* para que la alineación cambie según el tamaño de la pantalla:

```
<p class="text-center text-md-start">
```

Este párrafo está centrado en pantallas pequeñas y alineado a la izquierda en pantallas medianas o mayores.

```
</p>
```

5.3. Alineación vertical y horizontal con Flexbox

Uno de los puntos fuertes de Bootstrap es su sistema basado en **Flexbox**, que permite controlar cómo se distribuyen los elementos dentro de un contenedor.

Para activar Flexbox se usa la clase d-flex. Luego, se aplican las siguientes clases para controlar la alineación:

- ✓ **Alineación horizontal** (justify-content-*):

Clase	Resultado
justify-content-start	Alinea los elementos a la izquierda
justify-content-center	Centra horizontalmente
justify-content-end	Alinea a la derecha
justify-content-between	Distribuye con espacio entre medios

Clase	Resultado
justify-content-around	Distribuye con espacio alrededor
justify-content-evenly	Espacio uniforme entre todos

✓ **Alineación vertical (align-items-*):**

Clase	Resultado
align-items-start	Alinea al inicio (arriba)
align-items-center	Centra verticalmente
align-items-end	Alinea al final (abajo)

```
<div class="d-flex justify-content-center align-items-center" style="height: 200px; background-color: #f8f9fa;">
```

```
<div class="bg-primary text-white p-3 rounded">
```

Elemento alineado en el centro

```
</div>
```

```
</div>
```

Ejemplo:

```
02alineaciontexto.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Alineacion</title>
7  |   <link rel="stylesheet" href="css/bootstrap.min.css">
8  </head>
9  <body>
10 |   <script src="js/bootstrap.bundle.min.js"></script>
11 |
12 |   <p class="text-start">Texto alineado a la izquierda</p>
13 |   <p class="text-center">Texto centrado</p>
14 |   <p class="text-end">Texto alineado a la derecha</p>
15 |
16 |   <div class="d-flex justify-content-center align-items-center" style="height: 200px; background-color: #0eacb7;">
17 |     <div class="bg-primary text-white p-3 rounded">
18 |       Elemento alineado en el centro
19 |     </div>
20 |   </div>
21 |
22 </body>
23 </html>
```

ESTRUCTURA BÁSICA

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

Esto define un documento HTML estándar. Ahora vamos con lo importante: lo que hay **dentro del <body>**.

LÍNEAS DE TEXTO CON ALINEACIÓN

```
<p class="text-start">Texto alineado a la izquierda</p>
```

```
<p class="text-center">Texto centrado</p>
```

```
<p class="text-end">Texto alineado a la derecha</p>
```

- **text-start:** Alinea el texto a la izquierda (por defecto en muchos idiomas).
- **text-center:** Centra horizontalmente el texto.
- **text-end:** Alinea el texto a la derecha.

Estas clases son de Bootstrap y son útiles para aplicar alineación de texto sin CSS personalizado.

```
<p class="text-start">Texto alineado a la izquierda</p>
```

CONTENEDOR FLEX PARA CENTRAR ELEMENTO

```
<div class="d-flex justify-content-center align-items-center" style="height: 200px; background-color: #0eacb7;">
```

```
<div class="bg-primary text-white p-3 rounded">
```

Elemento alineado en el centro

```
</div>
```

```
</div>
```

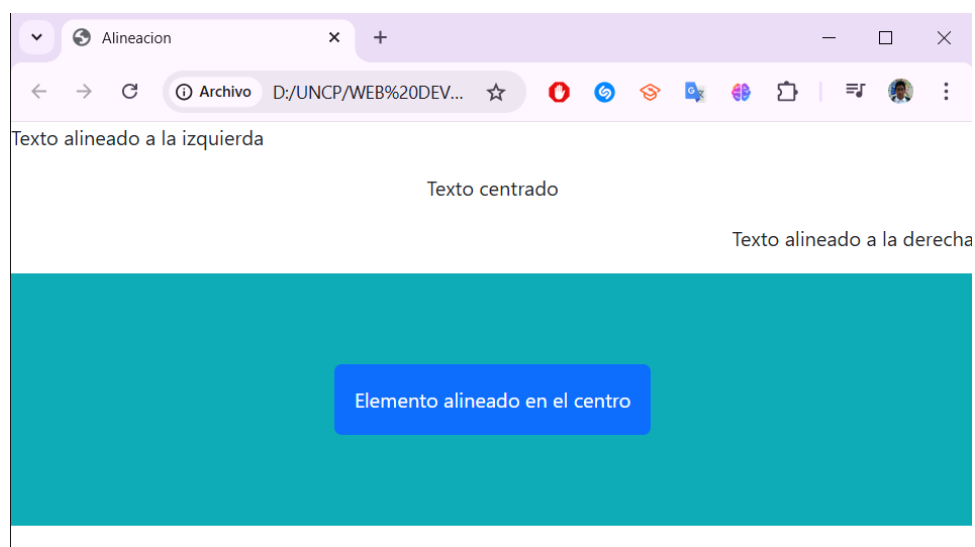
Este bloque es muy interesante. Vamos línea por línea:

- **d-flex:** Convierte el contenedor en un *flexbox*, lo que permite usar alineaciones modernas.
- **justify-content-center:** Centra el contenido **horizontalmente**.
- **align-items-center:** Centra el contenido **verticalmente**.
- **style="height: 200px; background-color: #0eacb7;"**: Le da altura al contenedor y le pone un color de fondo turquesa oscuro.

Dentro de este div, hay un bloque con estas clases:

- **bg-primary:** Fondo azul Bootstrap.
- **text-white:** Texto blanco.
- **p-3:** Padding (espaciado interno) tamaño 3.
- **rounded:** Bordes redondeados.

Resultado: Un bloque azul con texto blanco, redondeado y centrado vertical y horizontalmente en su contenedor de 200px de alto.



5.4. Clases de visibilidad

Bootstrap ofrece utilidades para controlar la visibilidad de los elementos en una página web, lo cual es fundamental para la creación de interfaces responsivas, limpias y dinámicas. Con estas clases, el desarrollador puede decidir si un elemento debe mostrarse u ocultarse según el tamaño de pantalla o de forma general, sin escribir reglas CSS personalizadas.

Estas utilidades se dividen principalmente en dos categorías:

- Cambiar el tipo de visualización (display).
- Controlar la visibilidad visual sin afectar el espacio ocupado.

❖ **Mostrar y ocultar elementos con clases display**

Las clases de display (d-*) permiten mostrar u ocultar elementos directamente en el HTML. Estas son útiles cuando se quiere que ciertos elementos aparezcan solo en dispositivos específicos o según el diseño deseado.

Clases básicas:

Clase	Descripción
d-none	Oculto el elemento completamente
d-block	Muestra el elemento como bloque
d-inline	Muestra el elemento en línea
d-flex	Muestra el elemento como flexbox
d-grid	Muestra el elemento como grid

❖ **Control de visibilidad sin alterar el espacio:** visible e invisible

A diferencia de las clases d-none, que eliminan completamente un elemento del flujo de la página, la clase invisible simplemente lo oculta **sin quitarle el espacio que ocupa**.

Clase	Función
visible	Hace que el elemento sea visible
invisible	Oculta el elemento visualmente, pero mantiene su espacio en el layout

Esto puede ser útil en situaciones donde el diseño no debe moverse, aunque un contenido esté oculto, por Ejemplo, al mostrar mensajes temporales o estados cargando (loading...).

Diferencia clave entre d-none e invisible

Clase	¿Elimina del DOM visual?	¿Ocupa espacio?
d-none	Sí	No
invisible	No (solo oculta visual)	Sí

Por eso, d-none se usa para ocultar contenido por completo (por Ejemplo, en dispositivos donde no se necesita), mientras que invisible es más útil para manejar transiciones o mantener la estructura visual.

Ejemplo:

```

03visibilidad.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Visibilidad</title>
7  |   <link rel="stylesheet" href="css/bootstrap.min.css">
8  </head>
9  <body>
10 |   <script src="js/bootstrap.bundle.min.js"></script>
11 |
12 |   <div class="d-block d-md-none bg-warning p-3 text-white">
13 |     Solo visible en pantallas pequeñas (móviles)
14 |   </div>
15 |
16 |   <div class="d-none d-md-block bg-success p-3 text-white">
17 |     Solo visible en pantallas medianas o mayores
18 |   </div>
19 |
20 |   <p class="invisible">Este texto está oculto, pero su espacio aún se reserva.</p>
21 |   <p class="visible">Este texto se muestra normalmente.</p>
22 |
23 |
24 </body>
25 </html>

```

Primer div: visible solo en móviles

```
<div class="d-block d-md-none bg-warning p-3 text-white">
```

Solo visible en pantallas pequeñas (móviles)

```
</div>
```

- d-block: Muestra el div como un bloque.
- d-md-none: A partir del tamaño **"md" (768px)** hacia arriba, lo oculta.
- bg-warning: Fondo amarillo (Bootstrap).
- p-3: Padding (espaciado interno).
- text-white: Texto blanco.

Resultado: Este div **solo se muestra en pantallas pequeñas** como móviles.

2. Segundo div: visible solo en pantallas medianas o grandes

```
<div class="d-none d-md-block bg-success p-3 text-white">
```

Solo visible en pantallas medianas o mayores

```
</div>
```

- d-none: Oculta el div en todos los tamaños por defecto.
- d-md-block: A partir del tamaño **"md" (768px)**, lo muestra como bloque.
- bg-success: Fondo verde.
- p-3: Padding.
- text-white: Texto blanco.

Resultado: Este div **solo se muestra en pantallas medianas o más grandes** (tablets, laptops, etc).

3. Texto invisible pero con espacio reservado

```
<p class="invisible">Este texto está oculto, pero su espacio aún se reserva.</p>
```

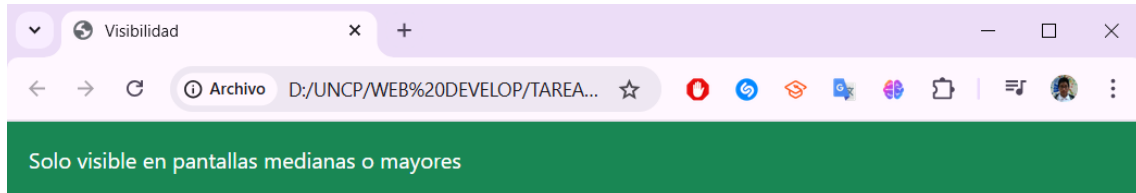
- invisible: El texto **no se ve**, pero **ocupa espacio** (como visibility: hidden en CSS).

Resultado: El texto no se muestra visualmente, pero sigue dejando un hueco en el diseño.

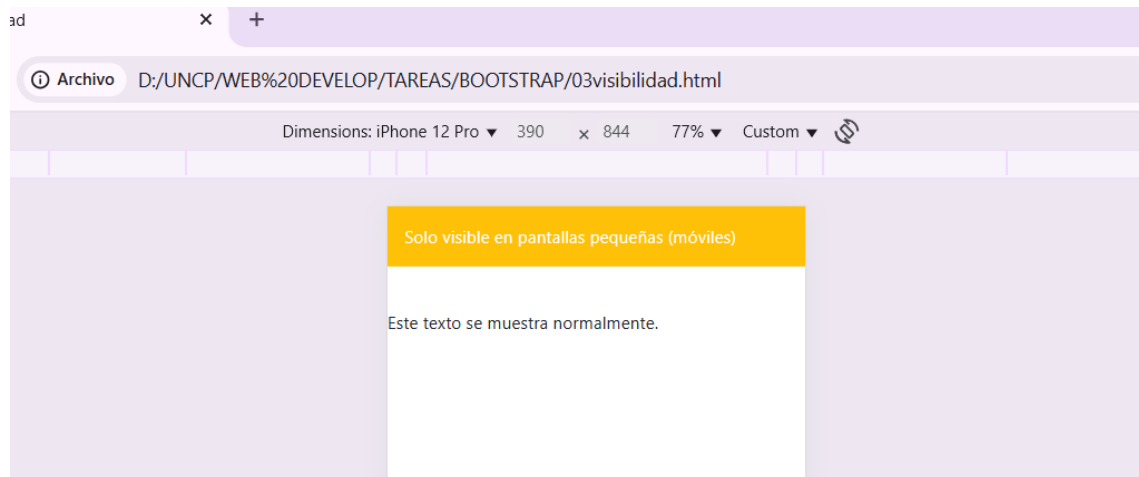
4. Texto visible normalmente

`<p class="visible">Este texto se muestra normalmente.</p>`

- visible: Clase para asegurar que el contenido **sea visible** (aunque en la mayoría de los casos no es necesario usarla, ya que es el estado por defecto).



Este texto se muestra normalmente.



6. Personalización con variables CSS y SASS

6.1. Variables CSS

La forma de crear una variable es con “—“

```
:root {  
  --bs-primary: #ff6600;  
}
```

6.2. Variables SASS

Sass (*Syntactically Awesome Stylesheets*) es un **preprocesador de CSS** que extiende las capacidades del CSS tradicional con características como:

- **Variables**
- **Anidamiento**
- **Mixins**
- **Funciones**
- **Mapas y bucles**

Estas herramientas permiten escribir código CSS **más limpio, modular, reutilizable y fácil de mantener.**

La forma de crear una variable es con “\$”

```
$primary: #ff6600; // Personaliza el color primario
@import "bootstrap";
```

6.3. Estructura de un proyecto con SASS

Una vez creado el proyecto de Bootstrap, se crea la carpeta scss con el archivo custom.css

```
your-project/
├── scss/
│   └── custom.scss
├── bootstrap/
│   ├── js/
│   └── scss/
└── index.html
```

6.4. Importación

En tu archivo custom.scss, importarás los archivos fuente Sass de Bootstrap. Tienes dos opciones: incluir todo Bootstrap o seleccionar las partes que necesitas. Recomendamos esta última opción, aunque ten en cuenta que existen algunos requisitos y dependencias entre nuestros componentes. También deberás incluir JavaScript para nuestros plugins.

Forma 1

```
// Custom.scss
// Option A: Include all of Bootstrap

// Include any default variable overrides here (though functions won't be available)

@import "../node_modules/bootstrap/scss/bootstrap";

// Then add additional custom code here
```

Forma 2

```
// Custom.scss
// Option B: Include parts of Bootstrap

// 1. Include functions first (so you can manipulate colors, SVGs, calc, etc)
@import "../node_modules/bootstrap/scss/functions";

// 2. Include any default variable overrides here

// 3. Include remainder of required Bootstrap stylesheets (including any separate color
@import "../node_modules/bootstrap/scss/variables";
@import "../node_modules/bootstrap/scss/variables-dark";

// 4. Include any default map overrides here

// 5. Include remainder of required parts
@import "../node_modules/bootstrap/scss/maps";
@import "../node_modules/bootstrap/scss/mixins";
@import "../node_modules/bootstrap/scss/root";

// 6. Include any other optional stylesheet partials as desired; list below is not inclu
@import "../node_modules/bootstrap/scss/utilities";
@import "../node_modules/bootstrap/scss/reboot";
@import "../node_modules/bootstrap/scss/type";
```

6.5. Compilación

Para usar tu código Sass personalizado como CSS en el navegador, se necesita un compilador Sass. Uno de ellos viene en el paquete de node

`npm install -g sass`

La forma de ejecutar y poder ver los cambios en tiempo real es con el siguiente comando

`sass --watch ./scss/custom.scss ./css/custom.css`

El cual verá reflejado todos los cambios realizados en el custom.scss y los irá convirtiendo en clases de CSS

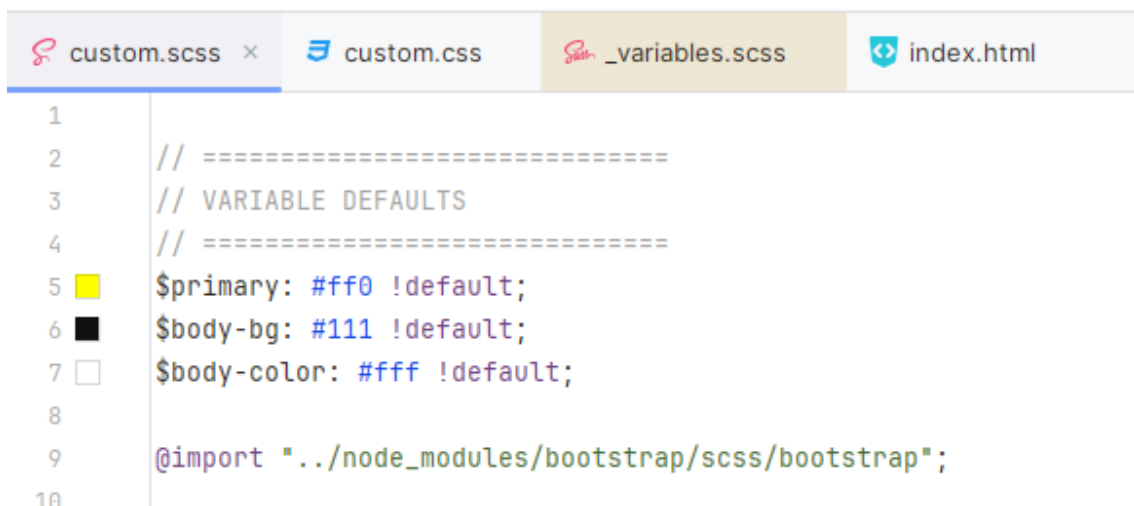
6.6. Inclusión

Una vez compilado el CSS, puedes incluirlo en tus archivos HTML. Dentro del archivo index.html, deberás incluir el archivo CSS compilado.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Custom Bootstrap</title>
    <link href="/css/custom.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

6.7. Variable defaults

Cada variable Sass en Bootstrap incluye el indicador !default, que permite sobrescribir el valor predeterminado de la variable en tu propio Sass sin modificar el código fuente de Bootstrap.



6.8. Maps

Los "maps" son estructuras clave-valor que agrupan datos relacionados:

- Add to map: Agregar nuevos elementos al mapa

```
custom.scss x custom.css _variables.scss index.html
16
17 @import "../node_modules/bootstrap/scss/functions";
18 @import "../node_modules/bootstrap/scss/variables";
19
20 $custom-colors: (
21   "verde-UNCP": #3F7C3A,
22 );
23
24
25 $theme-colors: map-merge($theme-colors, $custom-colors);
26
```

```
custom.scss custom.css x _variables.scss index.html
3069 .btn-dark {
3084 }
3085
3086 .btn-verde-UNCP {
3087   --bs-btn-color: #fff;
3088   --bs-btn-bg: #3F7C3A;
3089   --bs-btn-border-color: #3F7C3A;
3090   --bs-btn-hover-color: #fff;
3091   --bs-btn-hover-bg: rgb(53.55, 105.4, 49.3);
3092   --bs-btn-hover-border-color: rgb(50.4, 99.2, 46.4);
3093   --bs-btn-focus-shadow-rgb: 92, 144, 88;
3094   --bs-btn-active-color: #fff;
3095   --bs-btn-active-bg: rgb(50.4, 99.2, 46.4);
3096   --bs-btn-active-border-color: rgb(47.25, 93, 43.5);
3097   --bs-btn-active-shadow: inset 0 3px 5px rgba(0, 0, 0, 0.125);
3098   --bs-btn-disabled-color: #fff;
3099   --bs-btn-disabled-bg: #3F7C3A;
3100   --bs-btn-disabled-border-color: #3F7C3A;
3101 }
3102
```

- Modify map: Modificar un mapa existente, por ejemplo cambiar un color del tema

```
custom.scss x custom.css _variables.scss index.html
16
17 @import "../node_modules/bootstrap/scss/functions";
18 @import "../node_modules/bootstrap/scss/variables";
19
20
21
22 $theme-colors: map-merge($theme-colors, ("primary": #00F));
23
24 @import "../node_modules/bootstrap/scss/bootstrap";
25
```

```

5
7 .btn-primary {
8   --bs-btn-color: #fff;
9   --bs-btn-bg: #00F;
10  --bs-btn-border-color: #00F;
11  --bs-btn-hover-color: #fff;
12  --bs-btn-hover-bg: rgb(0, 0, 216.75);
13  --bs-btn-hover-border-color: #0000cc;
14  --bs-btn-focus-shadow-rgb: 38, 38, 255;
15  --bs-btn-active-color: #fff;
16  --bs-btn-active-bg: #0000cc;
17  --bs-btn-active-border-color: rgb(0, 0, 191.25);
18  --bs-btn-active-shadow: inset 0 3px 5px rgba(0, 0, 0, 0.125);
19  --bs-btn-disabled-color: #fff;
20  --bs-btn-disabled-bg: #00F;
21  --bs-btn-disabled-border-color: #00F;
22 }

```

- Remove from map: Eliminar claves de un mapa con map-remove

```

17 @import "../node_modules/bootstrap/scss/functions";
18 @import "../node_modules/bootstrap/scss/variables";
19
20
21
22 $theme-colors: map-remove($theme-colors, "danger");
23
24 @import "../node_modules/bootstrap/scss/bootstrap";

```

6.9. Functions

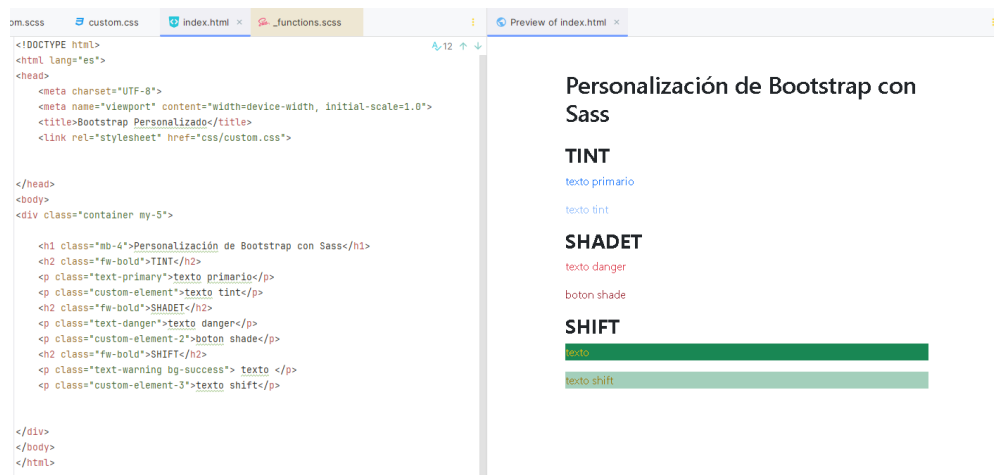
Bootstrap incluye funciones útiles para manipular colores.

Ejemplo:


```
custom.scss  _functions.scss  custom.css  index.html
> 🔍 0 results
203
204 // scss-docs-start color-functions
205 // Tint a color: mix a color with white
206 @function tint-color($color, $weight) {
207   @return mix(white, $color, $weight);
208 }
209
210 // Shade a color: mix a color with black
211 @function shade-color($color, $weight) {
212   @return mix(black, $color, $weight);
213 }
214
215 // Shade the color if the weight is positive, else tint it
216 @function shift-color($color, $weight) {
217   @return if($weight > 0, shade-color($color, $weight), tint-color($color,
218     -$weight));
219 }
220 // scss-docs-end color-functions
...
```

```
custom.scss  custom.css  index.html  _functions.scss
13
14 @import "../node_modules/bootstrap/scss/functions";
15 @import "../node_modules/bootstrap/scss/variables";
16
17 .custom-element {
18   color: tint-color($primary, 50%);
19 }
20
21 .custom-element-2 {
22   color: shade-color($danger, 30%);
23 }
24
25 .custom-element-3 {
26   color: shift-color($warning, 40%);
27   background-color: shift-color($success, -60%);
28 }
29 @import "../node_modules/bootstrap/scss/bootstrap";
30
```

Resultado

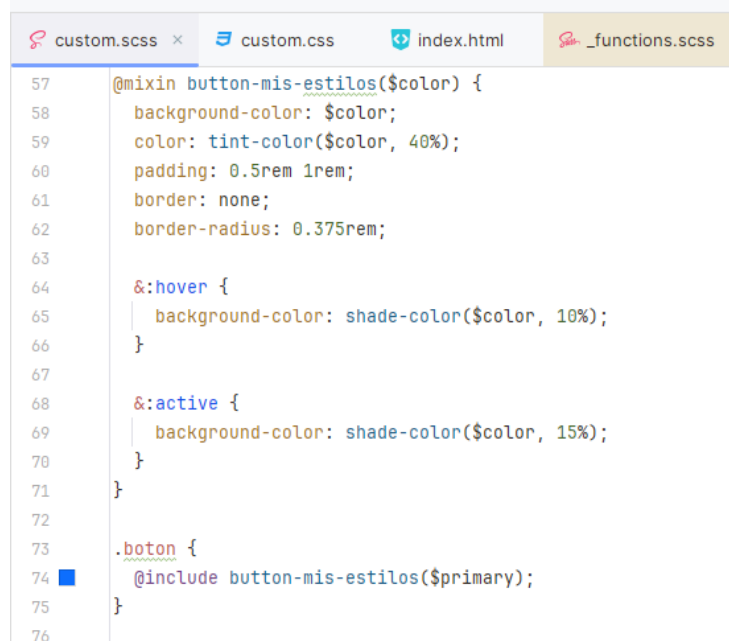


6.10. Mixins

Un mixin puede incluir cualquier código Sass o CSS válido: estilos, estructuras de control, parámetros, otros mixins, etc. Son ideales para mantener tu código **modular**, **DRY (Don't Repeat Yourself)** y fácil de mantener.

- Propiedades CSS comunes
- Parámetros personalizados
- Condicionales (@if, @else)
- Bucles (@for, @each, @while)
- Anidamiento de selectores
- Inclusión de otros mixins

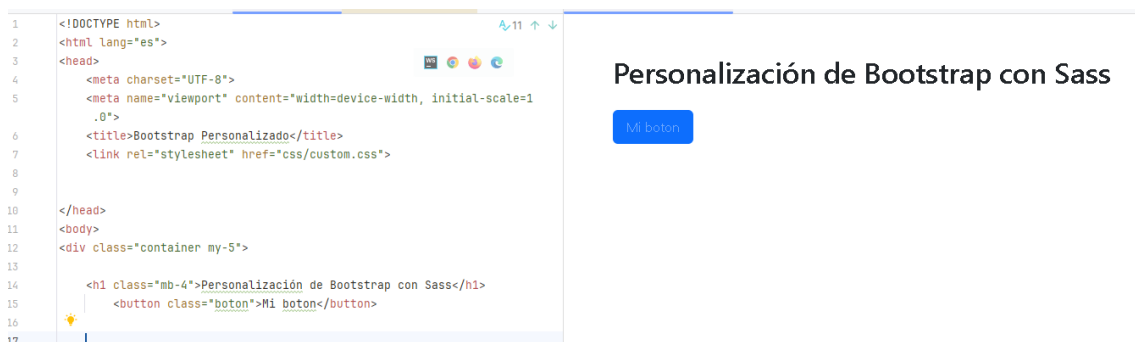
Ejemplo



```
custom.scss | custom.css | index.html | _functions.scss

1  @charset "UTF-8";
2  .boton {
3      background-color: #0d6efd;
4      color: rgb(109.8, 168, 253.8);
5      padding: 0.5rem 1rem;
6      border: none;
7      border-radius: 0.375rem;
8  }
9  .boton:hover {
10     background-color: rgb(11.7, 99, 227.7);
11 }
12 .boton:active {
13     background-color: rgb(11.05, 93.5, 215.05);
14 }
```

Resultado:



7. Animaciones y transiciones integradas

En este apartado nos enfocaremos en las animaciones y transiciones que ofrece Bootstrap de manera predeterminada, sin recurrir al uso de JavaScript personalizado. Específicamente, analizaremos el comportamiento de los **modales** y el funcionamiento de las clases utilitarias como `fade`, `collapse`, `show` y `collapsing`, que permiten agregar efectos de transición suave al mostrar u ocultar componentes. Estas clases forman parte de la estructura básica de Bootstrap para lograr animaciones visuales controladas y responsivas. Dejaremos fuera, por ahora, otros componentes interactivos como `tooltips`, `popovers` o `carruseles`, los cuales requieren mayor intervención con JavaScript, y serán tratados más adelante.

7.1. Modales (Modal)

Un **modal** en Bootstrap es una ventana emergente centrada en la pantalla que se superpone sobre el contenido principal. Se utiliza comúnmente para mostrar formularios, alertas o confirmaciones sin cambiar de página.

Ejemplo:

```
<!-- Botón para abrir el modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#miModal">
  Abrir Modal
</button>

<!-- Modal -->
<div class="modal fade" id="miModal" tabindex="-1" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Título del Modal</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>
      </div>
      <div class="modal-body">
        Contenido del modal.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
      </div>
    </div>
  </div>
</div>
```

Animaciones involucradas:

- fade: hace que el modal aparezca y desaparezca con una transición suave de opacidad.
- show: se aplica automáticamente mediante JavaScript de Bootstrap cuando el modal está visible.

7.2. Clase fade

La clase fade se utiliza para aplicar una animación de desvanecimiento (transición de opacidad) a los componentes cuando se muestran u ocultan.

Características:

- Se usa junto con componentes como modales o alertas.
- Funciona en combinación con JavaScript de Bootstrap para activar la animación.
- Solo agrega la transición, no muestra el componente por sí sola.

Ejemplo:

```

<!-- Modal -->
<div class="modal fade" id="miModal" tabindex="-1" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Título del Modal</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>
      </div>
      <div class="modal-body">
        Contenido del modal.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>
      </div>
    </div>
  </div>
</div>

```

7.3. Clase show

La clase show es utilizada por Bootstrap para indicar que un componente (como un modal o un elemento colapsable) está visible. Generalmente se aplica dinámicamente por el propio Bootstrap con JavaScript, aunque también puede colocarse manualmente para pruebas.

Características:

- Hace visible un elemento con animación si está combinado con fade o collapse.
- Utiliza transiciones CSS.

Ejemplo:

```

<div class="alert alert-success fade show">
  Esta alerta se muestra con animación.
</div>

```

7.4. Clase collapse

La clase collapse se utiliza para ocultar o mostrar un elemento con una transición de altura. Se aplica a elementos que deben plegarse (como menús o secciones de texto).

Características:

- Por defecto, oculta el contenido colapsado.
- Al hacer clic en un disparador, el elemento cambia entre oculto y visible, con una animación suave.

Ejemplo:

```
<!-- Botón -->
<button class="btn btn-primary" type="button" data-bs-toggle="collapse" data-bs-target="#contenido">
  Mostrar contenido
</button>

<!-- Contenido oculto -->
<div class="collapse" id="contenido">
  <div class="card card-body">
    Este es el contenido colapsable.
  </div>
</div>
```

Funcionamiento:

- collapse: oculta el contenido inicialmente.
- Al activarse, Bootstrap le agrega collapsing y luego show.

7.5. Clase collapsing

La clase collapsing se aplica temporalmente por Bootstrap cuando el contenido está en transición (es decir, expandiéndose o colapsándose).

Características:

- Se usa internamente durante la animación.
- Dura solo mientras se completa la transición CSS.
- Indica que el contenido está en movimiento (ni completamente visible ni oculto).

No se utiliza manualmente, pero es útil conocerla para comprender cómo Bootstrap gestiona el ciclo de la animación:

Ciclo del colapso:

1. Estado inicial: collapse
2. Al activar: se quita collapse, se agrega collapsing
3. Final de transición: se remueve collapsing, se agrega collapse show

```

<!--
Estado inicial:
<div class="collapse"> → Oculto

Al activarse:
→ Se elimina 'collapse'
→ Se añade 'collapsing' → Transición (altura)

Después de la transición:
→ Se elimina 'collapsing'
→ Se añade 'collapse show' → Visible
-->

```

8. Modificadores de color y estilos temáticos

Bootstrap utiliza un sistema de clases para personalizar colores y estilos, basado en su paleta predeterminada y temas. Los modificadores permiten ajustar la apariencia sin escribir CSS adicional.

8.1. Colores contextuales

Son colores predefinidos que se puede aplicar para indicar un estado específico, se usan principalmente para resaltar información. Clases como `.bg-primary`, `.text-success`, `.btn-danger` aplican colores predefinidos (primario, secundario, éxito, peligro, advertencia, etc.).

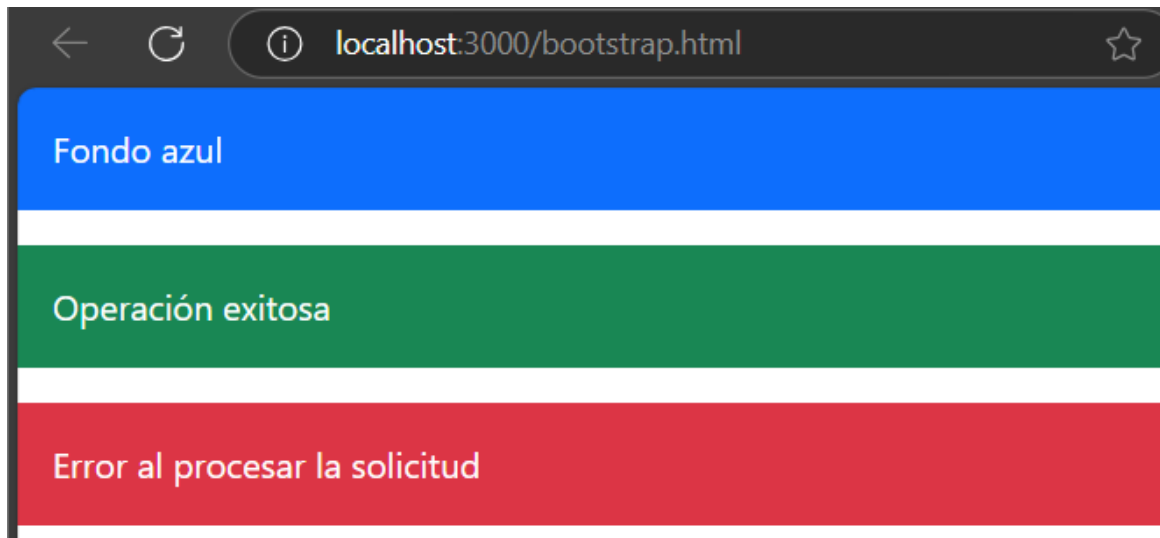
Ejemplo:

```

bootstrap.html X
bootstrap.html > html
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Colores Contextuales</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body>
10
11   <!-- Fondo azul -->
12   <div class="bg-primary text-white p-3">
13     Fondo azul
14   </div>
15
16   <!-- Fondo verde para éxito -->
17   <div class="bg-success text-white p-3 mt-3">
18     Operación exitosa
19   </div>
20
21   <!-- Fondo rojo para peligro -->
22   <div class="bg-danger text-white p-3 mt-3">
23     Error al procesar la solicitud
24   </div>
25
26   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
27 </body>
28 </html>

```

Resultado:



8.2. Variantes de estilo

Permite la personalización de la apariencia de los componentes. Modificadores como `.btn-outline-*` (para botones con bordes) o `.shadow` (para sombras) permiten personalizar la estética.

Ejemplo:

```
bootstrap.html X
bootstrap.html > html > body > div.container.mt-5
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Variantes de Estilo</title>
7    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
8  </head>
9  <body>
10
11    <div class="container mt-5">
12      <h2>Ejemplos de Botones y Elementos con Variantes de Estilo</h2>
13
14      <!-- Botón con borde (Outline) -->
15      <button class="btn btn-outline-secondary">Botón Outline</button>
16
17      <!-- Botón con sombra -->
18      <button class="btn btn-primary shadow mt-3">Botón con Sombra</button>
19
20      <!-- Botón con borde y sombra -->
21      <button class="btn btn-outline-success shadow mt-3">Botón Outline con Sombra</button>
22
23      <!-- Botón con borde (Outline) y tamaño grande -->
24      <button class="btn btn-outline-danger btn-lg mt-3">Botón Outline Grande</button>
25
26      <!-- Botón con borde (Outline) y tamaño pequeño -->
27      <button class="btn btn-outline-info btn-sm mt-3">Botón Outline Pequeño</button>
28    </div>
29
30    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
31  </body>
32 </html>
```


Resultado:



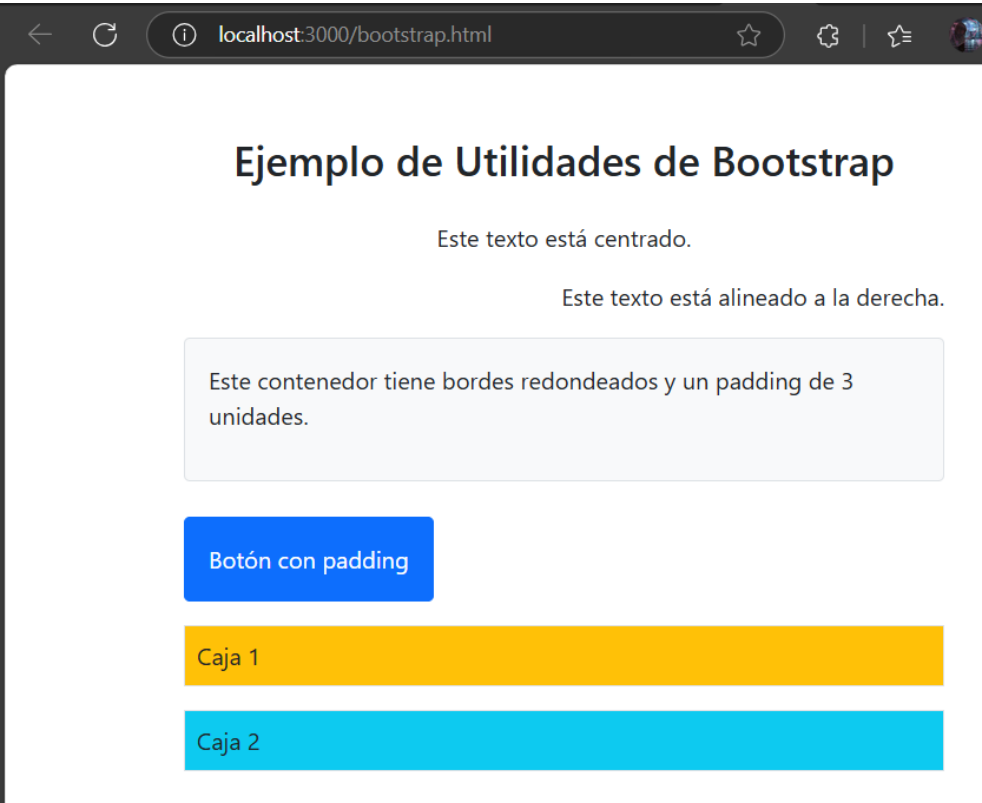
8.3. Utilidades

Son clases predefinidas que permiten realizar ajustes rápidos y precisos. Clases como `.text-center`, `.rounded`, `.p-3` (padding) ofrecen ajustes rápidos para alineación, bordes y espaciado.

Ejemplo:

```
bootstrap.html x
bootstrap.html > html > body > div.container.mt-5
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Utilidades de Bootstrap</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body>
10  <div class="container mt-5">
11    <h2 class="text-center mb-4">Ejemplo de Utilidades de Bootstrap</h2>
12    <!-- Alineación de texto al centro -->
13    <p class="text-center">Este texto está centrado.</p>
14    <!-- Alineación de texto a la derecha -->
15    <p class="text-end">Este texto está alineado a la derecha.</p>
16    <!-- Contenedor con borde redondeado y padding -->
17    <div class="rounded p-3 border bg-light">
18      <p>Este contenedor tiene bordes redondeados y un padding de 3 unidades.</p>
19    </div>
20    <!-- Botón con padding personalizado -->
21    <button class="btn btn-primary p-3 mt-4">Botón con padding</button>
22    <!-- Contenedor con espaciado entre elementos -->
23    <div class="mt-3">
24      <div class="p-2 border bg-warning mb-3">Caja 1</div>
25      <div class="p-2 border bg-info">Caja 2</div>
26    </div>
27  </div>
28  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"></script>
29 </body>
30 </html>
31
```

Resultado:



9. Diseño responsivo avanzado con media queries

Bootstrap implementa el diseño web responsivo utilizando media queries preconfiguradas, lo que permite adaptar el diseño y comportamiento visual de los elementos según el tamaño del dispositivo. Esta capacidad es fundamental para garantizar una experiencia de usuario óptima en móviles, tablets, laptops y pantallas grandes.

9.1. Breakpoints oficiales de Bootstrap 5

Los breakpoints definen los anchos mínimos para que ciertas clases surtan efecto. Bootstrap los nombra con prefijos específicos:

Tamaño	Abreviatura	Ancho mínimo	Dispositivos comunes
Extra small (xs)	<i>ninguno</i>	0px	Teléfonos pequeños
Small (sm)	sm	576px	Teléfonos grandes (iPhone Plus)
Medium (md)	md	768px	Tablets
Large (lg)	lg	992px	Laptops

Tamaño	Abreviatura	Ancho mínimo	Dispositivos comunes
Extra large (xl)	xl	1200px	Monitores de escritorio
Extra extra large (xxl)	xxl	1400px	Pantallas muy grandes (4K, etc.)

Las clases responsivas siguen la forma:

{propiedad}-{breakpoint}-{valor}

Si se tiene, text-md-start significa "alinear el texto a la izquierda a partir de pantallas medianas (768px)".

`<h2 class="text-center text-md-start">`

Centrado en móviles, alineado a la izquierda en pantallas medianas o más grandes.

`</h2>`

Ejemplo

```

04ejemplo.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>Ejemplo combinado Bootstrap</title>
7    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8  </head>
9  <body class="bg-light">
10
11    <div class="container py-5">
12      <div class="row align-items-center">
13
14        <!-- Imagen visible solo en md o más -->
15        <div class="col-md-4 d-none d-md-block">
16          
18        </div>
19
20        <!-- Contenido principal -->
21        <div class="col-md-8 p-4 bg-white rounded shadow">
22
23          <h2 class="text-center text-md-start mb-4">Bienvenido a Nuestra Página</h2>
24
25          <p class="text-center text-md-start">
26            | Este contenido está alineado al centro en móviles y a la izquierda en pantallas medianas y mayores. Se adapta automáticamente.
27          </p>
28
29          <!-- Botones alineados y con espacio -->
30          <div class="d-flex flex-column flex-sm-row justify-content-center justify-content-md-start gap-3 mt-4">
31            <button class="btn btn-primary px-4 py-2">Ver más</button>
32            <button class="btn btn-outline-secondary px-4 py-2">Contacto</button>
33          </div>
34
35          <!-- Mensaje oculto solo visualmente -->
36          <div class="mt-3 invisible">
37            | Este mensaje está oculto visualmente pero ocupa espacio.
38          </div>
39
40        </div>
41      </div>
42    </div>
43
44  </body>
45  </html>

```

1. ESTRUCTURA BÁSICA

<!DOCTYPE html>

<html lang="es">

<head>...</head>

<body class="bg-light">...</body>

</html>

- El idioma está definido como **español** (lang="es").
- El body tiene un fondo gris claro (bg-light) para una apariencia suave.
- Carga **Bootstrap 5.3** desde CDN (sin necesidad de archivos locales).

2. CONTENEDOR PRINCIPAL

<div class="container py-5">

- container: Crea un contenedor con márgenes automáticos.
- py-5: Padding vertical (arriba y abajo) de tamaño 5 → da espacio alrededor del contenido.

3. FILA Y COLUMNAS RESPONSIVAS

<div class="row align-items-center">

- row: Inicia una fila del grid de Bootstrap.
- align-items-center: Centra verticalmente los elementos dentro de la fila (útil si la imagen y el texto no tienen la misma altura).

4. COLUMNA CON IMAGEN

<div class="col-md-4 d-none d-md-block">

</div>

- col-md-4: Ocupa 4/12 columnas en pantallas **medianas o mayores**.

- `d-none d-md-block`: Oculta la imagen en móviles. Solo se muestra en pantallas medianas hacia arriba.
- `img-fluid`: Hace que la imagen sea responsiva (se escala con su contenedor).
- `rounded`: Bordes redondeados.
- `shadow`: Agrega una sombra sutil.

En móvil: La imagen desaparece → se gana espacio.

En desktop/tablet: La imagen aparece al lado del texto.

5. COLUMNA CON CONTENIDO PRINCIPAL

```
<div class="col-md-8 p-4 bg-white rounded shadow">
```

- `col-md-8`: Ocupa el resto de la fila (8 columnas) desde pantallas medianas.
- `p-4`: Padding interno.
- `bg-white`: Fondo blanco (resalta sobre el `bg-light`).
- `rounded, shadow`: Bordes redondeados y sombra para resaltar el bloque.

Título y Párrafo

```
<h2 class="text-center text-md-start">...</h2>
```

```
<p class="text-center text-md-start">...</p>
```

- **Responsive alignment:**
 - `text-center`: En pantallas pequeñas (móvil), el texto está centrado.
 - `text-md-start`: A partir de pantallas medianas, se alinea a la izquierda.

→ centrado → alineado a la izquierda

6. BOTONES CON FLEXBOX

```
<div class="d-flex flex-column flex-sm-row justify-content-center justify-content-md-start gap-3 mt-4">
```

- `d-flex`: Usa Flexbox para alinear los botones.

- flex-column: En pantallas pequeñas (por defecto), los botones van **uno debajo del otro**.
- flex-sm-row: Desde pantallas **pequeñas ($\geq 576\text{px}$)** se alinean **en fila**.
- justify-content-center: Centra horizontalmente los botones en móviles.
- justify-content-md-start: Alinea a la izquierda desde md en adelante.
- gap-3: Espacio uniforme entre botones.
- mt-4: Espacio arriba del grupo de botones.

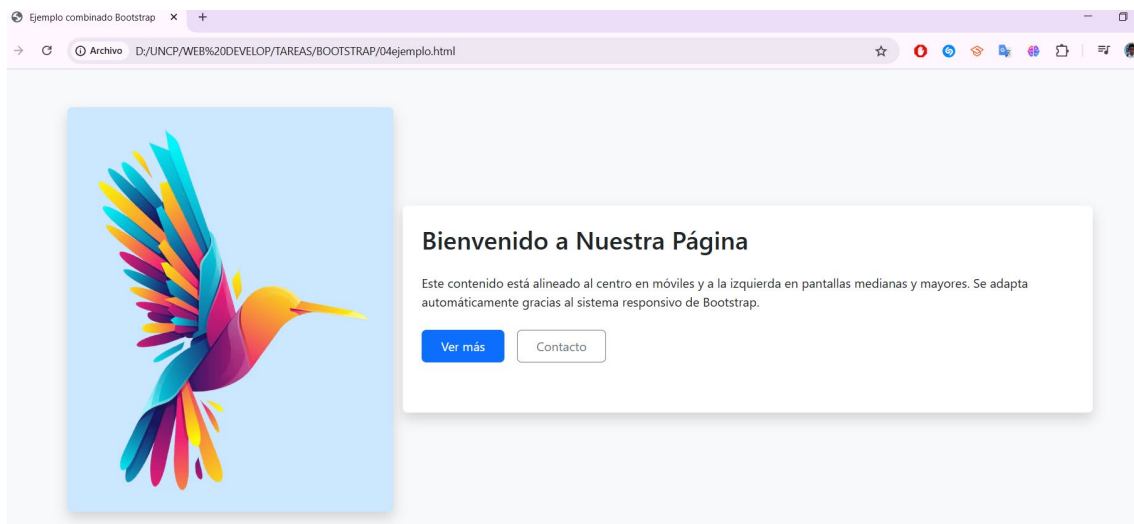
7. MENSAJE INVISIBLE (ESPACIO RESERVADO)

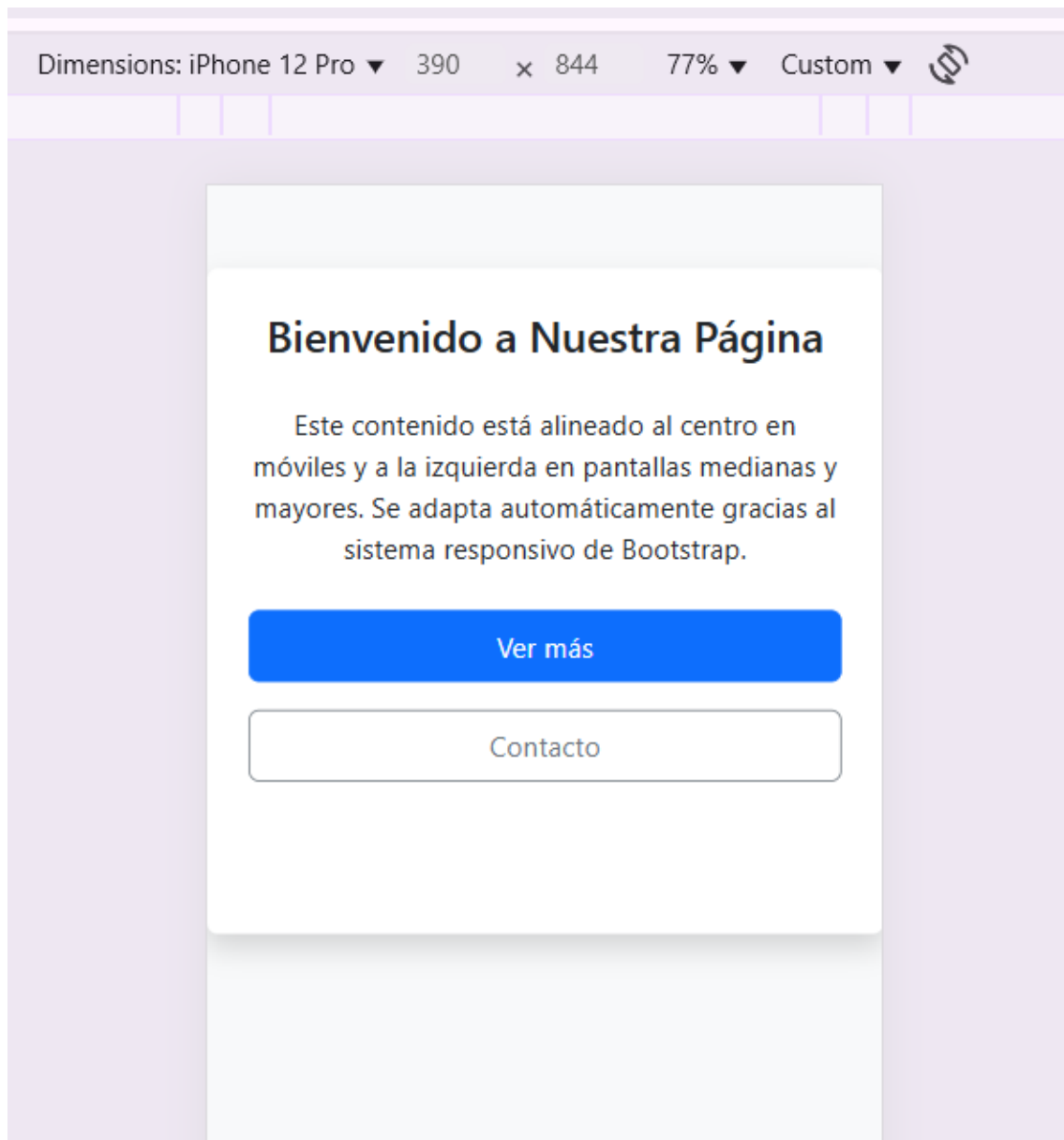
```
<div class="mt-3 invisible">
```

Este mensaje está oculto visualmente pero ocupa espacio.

```
</div>
```

- invisible: Oculta el contenido **visualmente**, pero **reserva el espacio** (como visibility: hidden en CSS).
- mt-3: Añade un margen arriba.





10. Integración con JavaScript (Tooltips, Dropdowns, Carousels)

En el mundo actual del desarrollo web, la interactividad ya no es opcional, es una expectativa. Cuando un usuario entra a una página, no solo quiere ver información, quiere interactuar con ella, entenderla fácilmente y sentir que la interfaz responde a sus acciones de manera rápida y natural. En ese escenario, la integración de JavaScript dentro de Bootstrap se convierte en una herramienta clave.

Bootstrap no es solo una colección de estilos visuales elegantes; es un ecosistema que ofrece comportamientos dinámicos preconstruidos que hacen que cualquier sitio o aplicación web se sienta viva y moderna. Componentes como tooltips (mensajes flotantes de ayuda), dropdowns (menús desplegables) o carousels (galerías deslizantes) son más

que adornos: son soluciones prácticas que mejoran la experiencia del usuario sin requerir programación avanzada.

Gracias a esta integración, es posible transformar una página estática en cuestión de minutos, haciendo que pequeños detalles como un mensaje emergente o una transición suave marquen una gran diferencia en cómo el usuario percibe y utiliza el sitio. Y lo mejor de todo: Bootstrap permite lograr esto con unas pocas líneas de código, muchas veces incluso sin escribir JavaScript directamente.

Esta sección no solo busca explicar cómo funcionan estos componentes, sino mostrar su valor real: cómo convierten una web funcional en una experiencia que se siente fluida, profesional y pensada para el usuario.

10.1. Activación de componentes

Existen dos formas principales de activar los componentes dinámicos en Bootstrap:

- **HTML (Data API):** Se usa directamente en el HTML mediante atributos como `data-bs-toggle`, `data-bs-target`, `data-bs-ride`, etc. Esta opción es muy práctica porque no requiere escribir ninguna línea de JavaScript.
- **JavaScript manual:** Es ideal cuando se necesita más control. Por Ejemplo, se puede usar `new bootstrap.Tooltip(elemento)` para activar un tooltip de forma personalizada, controlar eventos o destruirlo si ya no se necesita.

10.2. Componente Tooltip

Los tooltips son pequeños mensajes informativos que aparecen al pasar el cursor por encima de un elemento, y son útiles para aclarar funciones sin recargar visualmente la interfaz.

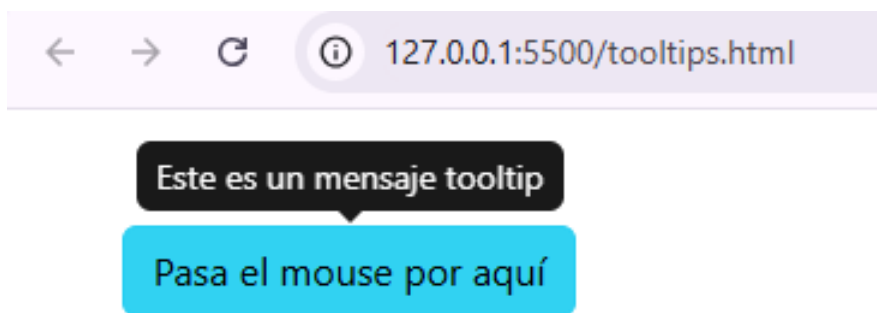
Ejemplo:


```

tooltips.html > html > head > meta
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Ejemplo de Tooltip</title>
6
7   <!-- Enlace al CSS de Bootstrap 5 desde CDN -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body class="p-5">
11   <!-- Botón con clase Bootstrap y atributo data-bs-toggle para activar el tooltip -->
12   <button type="button" class="btn btn-info" data-bs-toggle="tooltip" title="Este es un mensaje tooltip">
13     Pasa el mouse por aquí
14   </button>
15
16   <!-- Carga del JavaScript de Bootstrap (incluye Popper.js) -->
17   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
18
19   <!-- Script para inicializar manualmente todos los tooltips de la página -->
20   <script>
21     // Selecciona todos los elementos que tengan el atributo data-bs-toggle="tooltip"
22     const tooltips = document.querySelectorAll('[data-bs-toggle="tooltip"]');
23
24     // Inicializa el tooltip usando el constructor de Bootstrap
25     tooltips.forEach(el => new bootstrap.Tooltip(el));
26   </script>
27
28 </body>
29 </html>

```

Resultado:



Explicación:

En este caso, el botón tiene un mensaje oculto que se muestra cuando el usuario pasa el cursor. Gracias al atributo `data-bs-toggle` y a la pequeña función JavaScript, se logra una interacción rápida, útil y elegante.

10.3. Componente Dropdown

Los dropdowns permiten mostrar listas ocultas hasta que el usuario lo necesite. Son muy comunes en menús de navegación o formularios donde hay que elegir una opción.

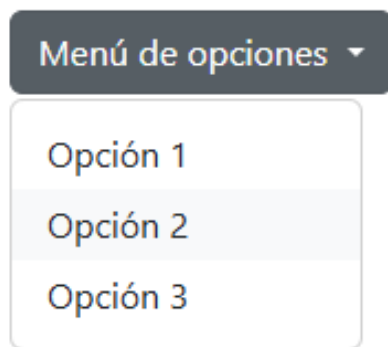
Ejemplo:

```

<? dropdowns.html > <? html > <? body.p-5 > <? script
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>Ejemplo de Dropdown</title>
6 <!-- Enlace al CSS de Bootstrap 5 desde CDN -->
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body class="p-5">
10 <!-- Contenedor del dropdown -->
11 <div class="dropdown">
12 <!-- Botón que activa el menú desplegable -->
13 <!-- La clase 'dropdown-toggle' crea el diseño del botón con flecha -->
14 <!-- El atributo 'data-bs-toggle="dropdown"' activa el dropdown dinámicamente -->
15 <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown">
16   Menú de opciones
17 </button>
18 <!-- Menú desplegable con varias opciones -->
19 <!-- La clase 'dropdown-menu' define el contenedor del menú -->
20 <!-- 'dropdown-item' da formato a cada opción del menú -->
21 <ul class="dropdown-menu">
22 <li><a class="dropdown-item" href="#">Opción 1</a></li>
23 <li><a class="dropdown-item" href="#">Opción 2</a></li>
24 <li><a class="dropdown-item" href="#">Opción 3</a></li>
25 </ul>
26 </div>
27 <!-- Carga del script JavaScript de Bootstrap (incluye Popper.js) -->
28 <!-- Este script permite que el componente funcione automáticamente -->
29 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
30 </body>
31 </html>

```

Resultado:



Explicación:

Basta con hacer clic en el botón para que aparezca un menú elegante y funcional. Todo se logra únicamente con HTML gracias a Bootstrap, sin necesidad de programar la interacción.

10.4. Componente Carousel

El carousel es una galería de imágenes o contenidos que se desplazan automáticamente o mediante botones. Ideal para mostrar promociones, portafolios o cualquier contenido visual que requiera atención.

Ejemplo:

```
carousels.html > html > body.p-5 > div#miCarousel.carousel.slide > div.carousel-inner
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Ejemplo de Carousel</title>
6
7   <!-- Enlace al archivo CSS de Bootstrap desde CDN -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body class="p-5">
11
12   <!-- Contenedor principal del carousel -->
13   <!-- 'carousel slide' activa el efecto de transición deslizable -->
14   <!-- 'data-bs-ride="carousel"' hace que inicie automáticamente -->
15   <div id="miCarousel" class="carousel slide" data-bs-ride="carousel">
16
17     <!-- Contenedor de las imágenes -->
18     <div class="carousel-inner">
19
20       <!-- Cada imagen se coloca dentro de un 'carousel-item' -->
21       <!-- La clase 'active' indica que es la imagen inicial -->
22       <div class="carousel-item active">
23         
24       </div>
25       <div class="carousel-item">
26         
27       </div>
28       <div class="carousel-item">
29         
30       </div>
31     </div>
32
33     <!-- Botón para ir a la imagen anterior -->
34     <!-- data-bs-target conecta con el ID del carousel -->
35     <!-- data-bs-slide="prev" indica dirección hacia atrás -->
36     <button class="carousel-control-prev" type="button" data-bs-target="#miCarousel" data-bs-slide="prev">
37       <span class="carousel-control-prev-icon" aria-hidden="true"></span>
38     </button>
39
40     <!-- Botón para ir a la siguiente imagen -->
41     <!-- data-bs-slide="next" indica dirección hacia adelante -->
42     <button class="carousel-control-next" type="button" data-bs-target="#miCarousel" data-bs-slide="next">
43       <span class="carousel-control-next-icon" aria-hidden="true"></span>
44     </button>
45
46   </div>
47
48   <!-- Enlace al JavaScript de Bootstrap (incluye Popper.js) -->
49   <!-- Necesario para que los botones y el auto-slide funcionen -->
50   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
51 </body>
52 </html>
```

```
carousels.html > html > body.p-5 > div#miCarousel.carousel.slide > div.carousel-inner
2 <html lang="es">
10 <body class="p-5">
15   <div id="miCarousel" class="carousel slide" data-bs-ride="carousel">
18     <div class="carousel-inner">
26       <div class="carousel-item">
27         
28       </div>
29     </div>
30
31     <!-- Botón para ir a la imagen anterior -->
32     <!-- data-bs-target conecta con el ID del carousel -->
33     <!-- data-bs-slide="prev" indica dirección hacia atrás -->
34     <button class="carousel-control-prev" type="button" data-bs-target="#miCarousel" data-bs-slide="prev">
35       <span class="carousel-control-prev-icon" aria-hidden="true"></span>
36     </button>
37
38     <!-- Botón para ir a la siguiente imagen -->
39     <!-- data-bs-slide="next" indica dirección hacia adelante -->
40     <button class="carousel-control-next" type="button" data-bs-target="#miCarousel" data-bs-slide="next">
41       <span class="carousel-control-next-icon" aria-hidden="true"></span>
42     </button>
43
44   </div>
45
46   <!-- Enlace al JavaScript de Bootstrap (incluye Popper.js) -->
47   <!-- Necesario para que los botones y el auto-slide funcionen -->
48   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
49 </body>
50 </html>
```

Resultado:

**Explicación:**

El carousel hace que una página cobre dinamismo visual sin esfuerzo. Con solo estructurar bien el HTML y aplicar unas clases de Bootstrap, el contenido se desliza de forma profesional.

En una web moderna, la interactividad es clave. Gracias a Bootstrap, no es necesario ser un experto en JavaScript para lograr resultados dinámicos y profesionales. Esta integración equilibrada entre HTML, CSS y JavaScript facilita la vida del desarrollador y mejora la experiencia de quien usa la página. Aprender a usar correctamente estas herramientas no solo es útil, sino también gratificante: cada clic, cada transición, cada detalle aporta a un diseño web más humano, funcional y agradable.

CONCLUSIÓN

El sistema de Grid de 12 columnas en Bootstrap 5 se presenta como una base fundamental para el diseño responsivo. Permite distribuir el contenido de manera eficiente en distintas resoluciones de pantalla, facilitando la creación de interfaces fluidas y ordenadas. Su estructura modular y el uso de breakpoints otorgan al desarrollador control total sobre la adaptación del diseño en múltiples dispositivos, mejorando así la experiencia del usuario y garantizando coherencia visual.

Los componentes predefinidos de Bootstrap, como botones, tarjetas, modales y alertas, aportan una gran ventaja en el desarrollo rápido de interfaces estéticas y funcionales. Estos elementos siguen estándares visuales consistentes, son fáciles de implementar y se adaptan al diseño responsivo sin complicaciones. Su reutilización permite mantener coherencia en el diseño, reducir tiempo de desarrollo y ofrecer al usuario final una experiencia clara y profesional.

La personalización de la tipografía y los estilos de texto en Bootstrap 5 es altamente flexible, permitiendo ajustar encabezados, pesos de fuente, alineaciones, tamaños y colores según las necesidades del proyecto. Estas opciones no solo enriquecen la estética del sitio web, sino que también mejoran la legibilidad y jerarquía visual de la información. Al dominar estas clases, el desarrollador puede reforzar la identidad visual del producto y optimizar la comunicación con el usuario.

Bootstrap facilita la creación de formularios estandarizados mediante clases específicas como `form-label`, `form-control` y `form-check`, lo que permite mantener una presentación clara y uniforme. Además, las validaciones integradas con clases como `is-valid` e `is-invalid` mejoran la interacción con el usuario al proporcionar retroalimentación visual inmediata, ayudando a prevenir errores y garantizar la calidad de los datos ingresados.

Las clases de utilidad de Bootstrap 5 permiten aplicar cambios rápidos y precisos en el diseño sin necesidad de escribir CSS personalizado. Gracias a ellas, se puede ajustar márgenes, rellenos, alineaciones y visibilidad de elementos de manera eficiente. Esta metodología favorece el desarrollo ágil y mantiene el código más limpio y fácil de mantener, además de fomentar buenas prácticas en la organización del diseño web.

La personalización mediante variables CSS y SASS otorga a Bootstrap un nivel avanzado de adaptabilidad. Esta funcionalidad permite modificar temas, colores, tamaños

y estructuras de manera centralizada, facilitando la creación de diseños únicos que se ajusten a la identidad visual de cada proyecto. Además, el uso de SASS permite una mayor modularidad y organización en el código, beneficiando tanto el rendimiento como la escalabilidad.

Bootstrap incluye clases específicas para animaciones y transiciones, como fade, show y collapse, que mejoran significativamente la interacción visual del usuario con el sitio. Estas animaciones aportan dinamismo y suavidad en la presentación de contenidos, ayudando a captar la atención sin distraer del contenido principal. Su fácil implementación refuerza la calidad estética y profesional del sitio.

El uso de media queries en Bootstrap 5 permite refinar aún más la adaptabilidad del diseño, ajustando estilos específicos para diferentes tamaños de pantalla. Esta técnica brinda control total sobre el comportamiento del layout en dispositivos variados, desde teléfonos móviles hasta pantallas grandes, lo que garantiza una experiencia de usuario coherente, accesible y optimizada en todo momento.

La integración de componentes dinámicos como tooltips, dropdowns y carousels a través de JavaScript permite enriquecer la interactividad del sitio sin requerir librerías externas. Estos elementos mejoran la funcionalidad y experiencia del usuario al proporcionar información adicional, navegación intuitiva o contenido visual atractivo. Bootstrap ofrece una implementación sencilla y eficaz de estos componentes, lo que facilita el desarrollo de interfaces modernas y dinámicas.

Bootstrap 5 representa una solución integral para el desarrollo frontend, combinando eficiencia, accesibilidad y estética. Su aprendizaje y aplicación no solo optimiza el tiempo de desarrollo, sino que también garantiza resultados profesionales en términos de diseño y usabilidad. Por ello, dominar este framework se convierte en una competencia fundamental para cualquier desarrollador web actual.

REFERENCIA BIBLIOGRÁFICA

Bootstrap. (2025). *Customize components*. Recuperado de <https://getbootstrap.esdocu.com/docs/5.1/customize/components/>

Bootstrap. (2025). *Sass*. Getbootstrap.com. Recuperado de <https://getbootstrap.com/docs/5.3/customize/sass/>