

FLEXBOX / GRID / BOOTSTRAP / ACCESIBILIDAD

▼ FLEXBOX

CSS Flexbox:

1. Introducción a Flexbox

- **Concepto:** Flexbox (Flexible Box Layout) es un sistema de diseño que facilita la distribución y alineación de elementos en contenedores unidimensionales (fila o columna).
- **Ventajas principales:**
 - Alineación y distribución sencillas de elementos.
 - Reducción del uso de posicionamiento o flotación.
 - Soporte amplio en navegadores modernos.

2. Terminología Básica

1. Contenedor flexible (Flex Container)

Elemento padre que utiliza

`display: flex` o `display: inline-flex` para activar el contexto Flexbox.

2. Elementos flexibles (Flex Items)

Hijos directos del contenedor flexible que obedecen las reglas de Flexbox.

Recuerda: inline-flex funciona igual que flex, pero el contenedor se comporta como un elemento en línea en lugar de uno de bloque.

3. Propiedades del Contenedor Flexible

Propiedad	Descripción	Valores más comunes
-----------	-------------	---------------------

<code>flex-direction</code>	Define la dirección del eje principal (main axis).	<code>row</code> (default), <code>row-reverse</code> , <code>column</code> , <code>column-reverse</code>
<code>flex-wrap</code>	Controla si los elementos se ajustan (wrap) o no (nowrap) al no haber espacio.	<code>nowrap</code> (default), <code>wrap</code> , <code>wrap-reverse</code>
<code>flex-flow</code>	Propiedad abreviada que combina <code>flex-direction</code> y <code>flex-wrap</code> .	Ej.: <code>flex-flow: row wrap;</code>
<code>justify-content</code>	Alinea los elementos en el eje principal.	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> , <code>space-evenly</code>
<code>align-items</code>	Alinea los elementos en el eje transversal (cross axis).	<code>stretch</code> (default), <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>baseline</code>
<code>align-content</code>	Alinea las líneas flexibles cuando hay varias filas/columnas (con <code>flex-wrap</code>).	<code>stretch</code> (default), <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code>
<code>gap</code> , <code>row-gap</code> , <code>column-gap</code>	Espacio entre elementos (similares a Grid).	Valores en px, em, rem, etc. (Ej.: <code>gap: 20px;</code>)

Punto clave en un examen:

- `flex-direction: row` es el valor predeterminado.
- `flex-wrap` por defecto es `nowrap`.
- `justify-content` y `align-items` alinean en ejes distintos.

4. Propiedades de los Elementos Flexibles

Propiedad	Descripción	Valores más comunes
<code>order</code>	Cambia visualmente el orden de los elementos sin alterar el DOM.	Números enteros (default: <code>0</code>)
<code>flex-grow</code>	Define cuánto crece un elemento respecto de los demás si sobra espacio.	Número (default: <code>0</code>)

<code>flex-shrink</code>	Define cuánto se encoge un elemento respecto de los demás si falta espacio.	Número (default: <code>1</code>)
<code>flex-basis</code>	Tamaño inicial del elemento antes de distribuir el espacio sobrante.	Unidad (px, em, %) o <code>auto</code>
<code>flex</code>	Abreviatura de <code>flex-grow</code> , <code>flex-shrink</code> y <code>flex-basis</code> .	Ejemplo: <code>flex: 1 0 auto;</code>
<code>align-self</code>	Sobrescribe la alineación del elemento individual (anula <code>align-items</code>).	<code>auto</code> , <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>baseline</code> , <code>stretch</code>

Punto clave:

- `flex: 1` es equivalente a `flex: 1 1 0;`
- `flex: none` es equivalente a `flex: 0 0 auto;`

5. Ejes en Flexbox

- **Eje principal (Main Axis):** Determinado por `flex-direction`.
 - `row` / `row-reverse`: eje principal horizontal.
 - `column` / `column-reverse`: eje principal vertical.
- **Eje transversal (Cross Axis):** Perpendicular al eje principal.
 - Controlado principalmente por `align-items` (para los elementos) y `align-content` (para el contenedor cuando hay varias líneas).

Pregunta típica: ¿Qué ocurre si cambias `flex-direction: row` a `column`?

- El eje principal pasa de horizontal a vertical, alterando la forma en que se aplica `justify-content` (ahora vertical) y `align-items` (ahora horizontal).

6. Ejemplos Prácticos

6.1 Centrado perfecto (horizontal y vertical)

```
.container {
  display: flex;
  justify-content: center; /* Eje principal */
  align-items: center;     /* Eje transversal */
  height: 100vh;
}
```

Clásico en un examen: Cómo centrar horizontal y verticalmente un elemento con Flexbox.

6.2 Distribución responsiva

```
.container {
  display: flex;
  flex-wrap: wrap;
  gap: 10px; /* Espacio entre elementos */
}

.item {
  flex: 1 1 200px; /* Crece y encoge en base a 200px */
}

@media (max-width: 600px) {
  .item {
    flex: 1 1 100%; /* Ocupa toda la línea en pantallas pequeñas */
  }
}
```

7. Valores Clave y Usos en Alineación

7.1 Justificación (eje principal) – `justify-content`

Valor	Descripción
<code>flex-start</code>	Alinea al inicio del contenedor. (Por defecto)
<code>flex-end</code>	Alinea al final del contenedor.

<code>center</code>	Centra los elementos en el eje principal.
<code>space-between</code>	Distribuye el espacio entre elementos (no hay espacio al borde).
<code>space-around</code>	Deja espacio igual alrededor de los elementos (bordes incluidos).
<code>space-evenly</code>	Espacio uniforme entre y alrededor de los elementos.

7.2 Alineación (eje transversal) – `align-items`

Valor	Descripción
<code>stretch</code>	Estira los elementos para que llenen el contenedor (valor por defecto).
<code>flex-start</code>	Al inicio del eje transversal (parte superior si <code>row</code>).
<code>flex-end</code>	Al final del eje transversal (parte inferior si <code>row</code>).
<code>center</code>	Centra los elementos en el eje transversal.
<code>baseline</code>	Alinea los elementos en base a la línea base del texto.

Recordatorio: `align-content` solo se aprecia cuando hay múltiples líneas de flex items (i.e., con `flex-wrap` activo).

8. Resumen de Defaults en Flexbox

1. `display: flex;`
2. `flex-direction: row;`
3. `flex-wrap: nowrap;`
4. `justify-content: flex-start;`
5. `align-items: stretch;`
6. `align-content: stretch;`
7. `flex-grow: 0;`
8. `flex-shrink: 1;`
9. `flex-basis: auto;`

▼ GRID

CSS Grid Layout:

1. Introducción a Grid

- **Concepto:** Grid es un sistema de diseño bidimensional (organiza elementos tanto en filas como en columnas).
- **Ventajas principales:**
 - Permite crear diseños complejos y responsivos sin necesidad de flotaciones o posicionamientos complicados.
 - Proporciona un control total sobre filas y columnas (bidimensional).
 - Soportado en navegadores modernos.

2. Diferencia principal: Grid vs Flexbox

Característica	CSS Grid	Flexbox
Dimensión	Bidimensional (filas y columnas)	Unidimensional (fila o columna)
Uso típico	Maquetación global de la página	Distribución interna de elementos en un eje (menús, barras, etc.)
Ejemplo de uso	Layout principal, cabecera, sidebar, contenido	Alineación de elementos en un navbar

Punto clave en exámenes: Saber que Grid se usa para estructuras complejas (bidimensional) y Flexbox para distribuir elementos en un solo eje (unidimensional).

3. Estructura Básica de CSS Grid

3.1 Contenedor de la cuadrícula (Grid Container)

Se activa con:

```
.grid-container {  
  display: grid; /* o inline-grid */  
}
```

- Este contenedor define el **contexto de cuadrícula**.
- Sus hijos se convierten en **elementos de grid (Grid Items)**.

3.2 Elementos de la cuadrícula (Grid Items)

- Hijos directos del contenedor con `display: grid`.
- Se organizan automáticamente según las reglas y propiedades de Grid.

Recordatorio: Para trabajar con subgrid, el elemento hijo también podría tener `display: subgrid` (característica todavía con soporte limitado).

4. Propiedades del Contenedor Grid

Propiedad	Descripción	Valores / Ejemplos
<code>display</code>	Activa el modo Grid.	<code>grid</code> , <code>inline-grid</code>
<code>grid-template-columns</code>	Define el número y tamaño de columnas.	Ej.: <code>100px 200px auto</code> ; <code>repeat(3, 1fr)</code>
<code>grid-template-rows</code>	Define el número y tamaño de filas.	Ej.: <code>50px auto 2fr</code>
<code>grid-template-areas</code>	Permite asignar nombres a secciones de la cuadrícula para un layout más legible.	Ej.: <code>"header header" "nav main" "footer footer"</code>
<code>gap</code> (o <code>row-gap</code> , <code>column-gap</code>)	Establece la separación entre filas y columnas.	Ej.: <code>gap: 20px;</code>
<code>justify-content</code>	Alinea el contenido total de la cuadrícula en el eje horizontal.	<code>start</code> , <code>end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> , etc.
<code>align-content</code>	Alinea el contenido total de la cuadrícula en el eje vertical.	<code>start</code> , <code>end</code> , <code>center</code> , <code>stretch</code>
<code>justify-items</code>	Alinea los elementos dentro de cada celda horizontalmente.	<code>start</code> , <code>end</code> , <code>center</code> , <code>stretch</code>
<code>align-items</code>	Alinea los elementos dentro de cada celda verticalmente.	<code>start</code> , <code>end</code> , <code>center</code> , <code>stretch</code>



Nota: `justify-items` y `align-items` afectan la distribución de cada ítem dentro de su celda, mientras que `justify-content` y `align-content` distribuyen toda la cuadrícula en su contenedor.

5. Propiedades de los Elementos (Grid Items)

Propiedad	Descripción	Valores / Ejemplos
<code>grid-column</code>	Determina la posición y extensión del elemento en columnas.	Ej.: <code>grid-column: 1 / 3;</code> (de col 1 a col 3)
<code>grid-row</code>	Determina la posición y extensión del elemento en filas.	Ej.: <code>grid-row: 2 / span 2;</code> (empieza en fila 2 y abarca 2 filas)
<code>grid-area</code>	Atajo para especificar simultáneamente <code>grid-row-start</code> , <code>grid-column-start</code> , etc.	Ej.: <code>grid-area: 1 / 2 / 3 / 4;</code>
<code>align-self</code>	Alinea el elemento verticalmente dentro de su propia celda, sobrescribiendo <code>align-items</code> .	<code>start</code> , <code>end</code> , <code>center</code> , <code>stretch</code>
<code>justify-self</code>	Alinea el elemento horizontalmente dentro de su propia celda, sobrescribiendo <code>justify-items</code> .	<code>start</code> , <code>end</code> , <code>center</code> , <code>stretch</code>
<code>order</code> *	Al igual que en Flexbox, cambia visualmente el orden de los ítems sin modificar el DOM.	Números enteros (poco habitual en Grid)

- * El uso de `order` en Grid no está tan extendido, pero existe a nivel de especificación.

6. Propiedades Abreviadas y Extras

6.1 `grid-template`

Unifica la definición de:

- `grid-template-rows`
- `grid-template-columns`
- `grid-template-areas`

Ejemplo:

```
.grid-container {
  grid-template:
    "header header" 50px
    "sidebar content" 1fr
    "footer footer" 30px
    / 200px 1fr;
}
```

Se definen simultáneamente las áreas, el tamaño de cada fila y el tamaño de cada columna.

6.2 `place-content`

Combina `align-content` y `justify-content` en una sola línea:

```
.grid-container {
  place-content: center center; /* Equivale a: align-content: center; justify-content: center; */
}
```

6.3 `place-items`

Combina `align-items` y `justify-items`:

```
.grid-container {
  place-items: center center; /* Equivale a: align-items: center; justify-items: center; */
}
```

7. Unidades y Funciones Comunes en Grid

1. **fr (Fractional Unit)**: Representa una fracción del espacio disponible.

```
grid-template-columns: 1fr 2fr; /* La primera columna
recibe 1/3 y la segunda 2/3 del espacio. */
```

2. **auto**: Tamaño automático según el contenido.
3. **minmax(min, max)**: Define un rango entre un tamaño mínimo y uno máximo.

```
grid-template-columns: minmax(100px, 1fr);
```

4. **repeat(count, size)**: Repite un patrón de columnas o filas.

```
grid-template-columns: repeat(3, 1fr); /* 3 columnas,
cada una 1fr. */
```

5. **auto-fit** vs **auto-fill** (en **repeat()**):

- **auto-fit**: Intenta ajustar cuantas columnas quepan, *colapsando las celdas vacías* si hay más espacio.
- **auto-fill**: Llena el contenedor con el máximo número de columnas de ese tamaño, incluso si algunas quedan vacías.

Pregunta típica: Diferencia entre auto-fill y auto-fit.

Respuesta: **auto-fill** crea tantas columnas como quepan sin recolocar las celdas vacías, mientras que **auto-fit** ajusta o "colapsa" el sobrante para que las celdas se adapten.

8. Ejemplos Prácticos

8.1 Estructura responsiva básica con **auto-fit**

```
.grid-container {
  display: grid;
  gap: 20px;
```

```
grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
}
```

- Se generan automáticamente tantas columnas como quepan, cada una mínimo de `150px` y máximo de `1fr`.

8.2 Layout con áreas nombradas

```
.grid-container {
  display: grid;
  grid-template-areas:
    "header header"
    "menu   content"
    "footer footer";
  grid-template-rows: 80px 1fr 40px;
  grid-template-columns: 200px 1fr;
}

.header {
  grid-area: header;
}

.menu {
  grid-area: menu;
}

.content {
  grid-area: content;
}

.footer {
  grid-area: footer;
}
```

- Facilita la lectura y modificación del layout.

▼ BOOTSTRAP

1. Instalación y Uso de Bootstrap 5

1.1 Usar Bootstrap via CDN

La forma más sencilla e inmediata de incluir Bootstrap en tu proyecto. Requiere conexión a internet para funcionar.

1. Enlazar el CSS (en `<head>`):

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
```

2. Enlazar el JavaScript (justo antes de `</body>`):

- Archivos separados (Popper + Bootstrap):

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js"></script>
```

- Archivo combinado (Bootstrap incluye Popper):

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

3. Estructura básica:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ejemplo con Bootstrap 5 (CDN)</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1 class="text-center">Hello, Bootstrap!</h1>
    <p class="text-muted">Texto con estilo Bootstrap</p>
```

```

        <button class="btn btn-primary">Botón Primario</button>

        <!-- JavaScript de Bootstrap -->
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
    </body>
</html>

```

Ventaja de la CDN: Se beneficia del caché del navegador si el usuario ya la ha descargado en otros sitios.

2. Sistema de Grid (Rejilla) en Bootstrap 5

Bootstrap aprovecha Flexbox para su **sistema de rejilla**, el cual facilita la creación de diseños responsivos. La estructura básica es:

1. **Contenedor** (`.container` , `.container-fluid` o `.container-{breakpoint}`)
2. **Fila** (`.row`)
3. **Columnas** (`.col` , `.col-*` con distintos tamaños)

2.1 Contenedores

Clases principales de contenedor:

Clase	Descripción
<code>.container</code>	Ancho fijo y centrado en anchos de pantalla mayores que 576px.
<code>.container-fluid</code>	Ocupa el 100% del ancho de la pantalla, siempre.
<code>.container-{bp}</code>	Cambia de ancho fluido a fijo a partir de un breakpoint (<code>sm</code> , <code>md</code> , <code>lg</code> , <code>xl</code> , <code>xxl</code>).

Ejemplo de uso:

```

<div class="container">Contenido centrado con ancho fijo
</div>

```

```
<div class="container-fluid">Ocupa todo el ancho disponible</div>
<div class="container-sm">Fijo a partir de 576px de ancho</div>
```

2.2 Filas y Columnas

- `.row`: Agrupa columnas, aplica margen negativo a los lados para acomodarlas correctamente.
- `.col`: Cada columna se comporta como un elemento flex dentro de la fila.

Ejemplo básico:

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

Por defecto, cada `.col` ocupará el mismo espacio disponible en la fila.

Columnas con tamaño específico

Puedes especificar columnas más concretas mediante `.col-1` hasta `.col-12`:

```
<div class="row">
  <div class="col-4">Col (4/12)</div>
  <div class="col-8">Col (8/12)</div>
</div>
```

Si la suma de las columnas excede las 12 unidades, se hace salto de línea.

3. Alineación en el Sistema de Grid

Bootstrap hereda las utilidades de **Flexbox** para alinear horizontal y verticalmente.

3.1 Alineación Horizontal

Se aplica a la fila (`.row`) mediante clases `justify-content-*`.

Clase	Descripción
<code>.justify-content-start</code>	Alinea las columnas al inicio (left).
<code>.justify-content-center</code>	Centra las columnas horizontalmente.
<code>.justify-content-end</code>	Alinea las columnas al final (right).
<code>.justify-content-around</code>	Espacio alrededor de las columnas (espacios a los lados y entre ellas).
<code>.justify-content-between</code>	Espacio uniforme entre columnas, sin margen lateral.
<code>.justify-content-evenly</code>	Espacio uniforme entre y alrededor de todas las columnas.

Ejemplo:

```
<div class="row justify-content-center">
  <div class="col-4">Centered Column</div>
</div>
```

3.2 Alineación Vertical

Se aplica también en la fila (`.row`) con las clases `align-items-*`.

Clase	Descripción
<code>.align-items-start</code>	Alinea los contenidos arriba.
<code>.align-items-center</code>	Centra verticalmente los contenidos.
<code>.align-items-end</code>	Alinea los contenidos abajo.

Ejemplo:

```
<div class="row align-items-center" style="height: 200px;">
  <div class="col-6">Columna centrada verticalmente</div>
</div>
```

4. Breakpoints y Responsive Design

Bootstrap 5 define **breakpoints** por defecto para pantallas de diferentes anchos:

Breakpoint	Tamaño en px	Clases comunes
xs	< 576px	(mobile first, sin prefijo)
sm	≥ 576px	<code>.col-sm-*</code> , <code>.d-sm-none</code> etc.
md	≥ 768px	<code>.col-md-*</code> , <code>.d-md-none</code> etc.
lg	≥ 992px	<code>.col-lg-*</code> , <code>.d-lg-none</code> etc.
xl	≥ 1200px	<code>.col-xl-*</code> , <code>.d-xl-none</code> etc.
xxl	≥ 1400px	<code>.col-xxl-*</code> , <code>.d-xxl-none</code> etc.

Ejemplo de columnas responsivas:

```
class="col-12 col-sm-6 col-md-4"
```

- En móviles (<576px): ocupa 12 columnas (toda la fila).
- En tablets (≥576px): ocupa 6 columnas (media fila).
- En pantallas medianas (≥768px): ocupa 4 columnas (un tercio de la fila).

5. Utilidades de Colores

Bootstrap incluye una paleta de colores denominada **Theme Colors**:

Nombre	Clase Texto	Clase Fondo	Color (aprox.)
Primary	<code>.text-primary</code>	<code>.bg-primary</code>	#0d6efd

Secondary	<code>.text-secondary</code>	<code>.bg-secondary</code>	#6c757d
Success	<code>.text-success</code>	<code>.bg-success</code>	#198754
Danger	<code>.text-danger</code>	<code>.bg-danger</code>	#dc3545
Warning	<code>.text-warning</code>	<code>.bg-warning</code>	#ffc107
Info	<code>.text-info</code>	<code>.bg-info</code>	#0dcaf0
Light	<code>.text-light</code>	<code>.bg-light</code>	#f8f9fa
Dark	<code>.text-dark</code>	<code>.bg-dark</code>	#212529

Aplicaciones comunes:

- Texto: `<p class="text-info">Info Text</p>`
- Fondo: `<div class="bg-warning">Fondo Amarillo</div>`
- Botones: `<button class="btn btn-primary">Botón Azul</button>`

6. Botones y Sus Variantes

Los botones en Bootstrap se definen con la clase `.btn` más la variante de color:

- `.btn-primary`
- `.btn-secondary`
- `.btn-success`
- `.btn-danger`
- `.btn-warning`
- `.btn-info`
- `.btn-light`
- `.btn-dark`

Ejemplo:

```
<button class="btn btn-primary">Primario</button>
<button class="btn btn-success">Éxito</button>
```

7. Mostrar/Ocultar Elementos

Mediante utilidades de display (`.d-*`) y visibilidad (`.visible-*` , `.invisible`) se controla cómo se muestran u ocultan elementos en distintos breakpoints.

- `.d-none` : Ningún display.
- `.d-sm-block` : Visible como bloque a partir de `sm` .
- `.d-md-inline` : Se muestra como inline a partir de `md` .
- `.d-lg-none` : Oculto en `lg` o mayor.

Ejemplo: `<div class="d-none d-sm-block">Solo se muestra en sm o mayor</div>`

8. Clases para Dar Estilo a Textos

8.1 Encabezados `<h1>` – `<h6>`

- **Estilos Predefinidos:** Los encabezados (`<h1>` a `<h6>`) vienen con tamaños y espaciados por defecto, además de usar la fuente del sistema.
- **Etiqueta `<small>`** : Permite crear texto secundario más claro en un encabezado.

```
<h1>Encabezado Principal <small>texto secundario</small></h1>
```

- **Etiqueta `<mark>`** : Resalta el fondo en amarillo con algo de padding.

```
<p>Párrafo con <mark>texto resaltado</mark> en Bootstrap.</p>
```

- **Etiqueta `<abbr>`** : Muestra un borde inferior punteado para acrónimos o abreviaturas.

```
<p>Ejemplo de <abbr title="HyperText Markup Language">HTML</abbr> en Bootstrap.</p>
```

- **Etiqueta `<blockquote>` con `.blockquote`** : Estiliza citas:

```
<blockquote class="blockquote">
  <p>Una gran cita para inspirar a los desarrolladores.</p>
  <footer class="blockquote-footer">Autor de la cita</footer>
</blockquote>
```

8.2 Clases de Texto Principales

A continuación se muestra un resumen de las principales utilidades de texto de Bootstrap 5:

Clase	Descripción
<code>.h1</code> – <code>.h6</code>	Aplica el estilo propio de <code><h1></code> a <code><h6></code> sin usar necesariamente esas etiquetas de encabezado.
<code>.display-1</code> ... <code>.display-6</code>	Crea encabezados de gran tamaño (1 es el más grande, 6 el más pequeño).
<code>.text-start</code>	Alinea el texto al inicio (izquierda en idiomas LTR).
<code>.text-center</code>	Centra el texto.
<code>.text-end</code>	Alinea el texto al final (derecha en idiomas LTR).
<code>.text-wrap</code>	Permite que el texto se ajuste al ancho del contenedor.
<code>.text-nowrap</code>	Evita que el texto se ajuste, pudiendo desbordar.
<code>.text-break</code>	Rompe las palabras largas para evitar que distorsionen el diseño.
<code>.text-lowercase</code>	Convierte el texto a minúsculas.
<code>.text-uppercase</code>	Convierte el texto a mayúsculas.
<code>.text-capitalize</code>	Capitaliza la primera letra de cada palabra.

<code>.fs-1</code> ... <code>.fs-6</code>	Cambia el tamaño de la fuente (1 es el más grande).
<code>.fw-light</code> <code>.fw-normal</code> <code>.fw-bold</code>	Cambia el peso (grosor) de la fuente.
<code>.fst-italic</code>	Aplica estilo cursivo (itálico) al texto.
<code>.fst-normal</code>	Restablece el estilo normal si estaba en itálico.

Ejemplos Rápidos:

```
<h1 class="display-1">Encabezado Display-1</h1>
<p class="text-uppercase fs-3 fw-bold">
  Texto en mayúsculas, tamaño de fuente 3, en negrita
</p>
<p class="fst-italic">
  Texto cursivo con .fst-italic
</p>
```

8.3 Colores de Texto

Bootstrap 5 proporciona clases para colorear fácilmente el texto:

Clase	Descripción
<code>.text-primary</code>	Texto azul (color primario).
<code>.text-secondary</code>	Texto gris (color secundario).
<code>.text-success</code>	Texto verde.
<code>.text-danger</code>	Texto rojo (indicativo de error/peligro).
<code>.text-warning</code>	Texto amarillo (advertencia).
<code>.text-info</code>	Texto azul claro (informativo).
<code>.text-dark</code>	Texto oscuro (negro o casi negro).
<code>.text-light</code>	Texto claro (ideal sobre fondos oscuros).
<code>.text-white</code>	Texto blanco.
<code>.text-black-50</code>	Texto negro al 50% de opacidad.
<code>.text-white-50</code>	Texto blanco al 50% de opacidad.
<code>.text-muted</code>	Texto gris tenue (desactivado).

Enlaces: Es preferible usar `.link-primary`, `.link-success`, etc., ya que incluyen un color más oscuro al pasar el ratón (`:hover`).

Ejemplo Rápido:

```
<p class="text-success">Texto con color de éxito (verde)
</p>
<a href="#" class="link-danger">Enlace con estilo rojo</a>
```

8.4 Colores de Fondo en los Textos

- Clases como `.bg-primary`, `.bg-success`, etc., aplican un color de fondo directamente.
- Para tener **color de fondo y texto** con el contraste adecuado, se recomienda usar `.text-bg-{color}`:

```
<p class="p-2 text-bg-danger">Fondo rojo con texto adaptado</p>
```

- Disponibles variantes: `.text-bg-primary`, `.text-bg-success`, `.text-bg-warning`, `.text-bg-info`, `.text-bg-dark`, `.text-bg-light`, etc.

8.5 Badges (Insignias)

Las insignias se usan para resaltar cantidades, estados, o etiquetas:

- **Clase base:** `.badge`
- **Color de fondo:** `.bg-{color}` (por ejemplo: `.bg-primary`, `.bg-secondary`, etc.)
- **Ejemplo dentro de un encabezado:**

```
<h1>Encabezado <span class="badge bg-secondary">Nuevo
</span></h1>
```

- **Usar como contador en un botón:**

```
<button class="btn btn-primary">
  Notificaciones <span class="badge bg-light text-blac
ck">4</span>
</button>
```

- **Píldoras (forma redondeada):** `.badge-pill` (En Bootstrap 5, esta clase se simplifica, pero aún funciona el concepto de `border-radius`).

Accesibilidad: Incluye texto adicional para describir el significado de la insignia a usuarios de lectores de pantalla.

9. Clases para Dar Estilo a Tablas en Bootstrap 5

Para convertir una tabla HTML estándar en una **tabla con estilo Bootstrap**, se aplica la clase `.table`.

9.1 Clase Base y Opcionales

Clase	Descripción
<code>.table</code>	Clase base para estilizar la tabla.
<code>.table-striped</code>	Alterna colores de fila para mejor legibilidad.
<code>.table-bordered</code>	Añade bordes a las celdas de la tabla.
<code>.table-borderless</code>	Elimina los bordes de la tabla y celdas.
<code>.table-hover</code>	Resalta las filas al pasar el ratón por encima.
<code>.table-sm</code>	Reduce el tamaño de fuente y espaciado, útil en tablas densas.
<code>.table-responsive</code>	Hace que la tabla sea desplazable horizontalmente en pantallas pequeñas (<i>se aplica al contenedor que rodea la tabla</i>).
<code>.caption-top</code>	Coloca el <code><caption></code> en la parte superior de la tabla.

Ejemplo Rápido:

```
<div class="table-responsive">
  <table class="table table-striped table-hover">
```

```

<caption>Título de la Tabla</caption>
<thead>
  <tr>
    <th>Producto</th><th>Cantidad</th><th>Precio</th>
  </tr>
</thead>
<tbody>
  <tr><td>Ratón</td><td>15</td><td>100</td></tr>
  <tr><td>Teclado</td><td>34</td><td>340</td></tr>
  <tr><td>Pantalla</td><td>10</td><td>400</td></tr>
</tbody>
</table>
</div>

```

9.2 Colorear Tablas

Bootstrap 5 permite aplicar colores de fondo a toda la tabla o a filas concretas:

Clase	Descripción
<code>.table-dark</code>	Tema oscuro para toda la tabla.
<code>.table-light</code>	Tema claro.
<code>.table-primary</code>	Fondo con color primario.
<code>.table-secondary</code>	Fondo gris (color secundario).
<code>.table-success</code>	Fondo verde.
<code>.table-danger</code>	Fondo rojo (alerta/peligro).
<code>.table-warning</code>	Fondo amarillo (advertencia).
<code>.table-info</code>	Fondo azul claro (informativo).
<code>.table-active</code>	Resalta una fila o celda específica (efecto "activo").

Nota: También se pueden combinar con `.table-striped`, `.table-hover`, etc.

9.3 Alineamiento en las Tablas

Para alinear horizontal y verticalmente los contenidos de las celdas:

Clase	Descripción
<code>.align-middle</code>	Centra el contenido verticalmente.
<code>.align-top</code>	Alinea el contenido en la parte superior.
<code>.align-bottom</code>	Alinea el contenido en la parte inferior.
<code>.text-start</code>	Alinea el texto a la izquierda.
<code>.text-center</code>	Alinea el texto al centro.
<code>.text-end</code>	Alinea el texto a la derecha.

Ejemplo de uso:

```
<table class="table table-striped">
  <tbody>
    <tr>
      <td class="align-top text-start">Arriba e izquierd
a</td>
      <td class="align-middle text-center">Medio y centr
o</td>
      <td class="align-bottom text-end">Abajo y derecha
</td>
    </tr>
  </tbody>
</table>
```

10. Clases para Dar Estilo a las Imágenes en Bootstrap 5

Bootstrap 5 proporciona una serie de clases que facilitan la personalización de la apariencia y el comportamiento de las imágenes.

Clase	Descripción
<code>.img-fluid</code>	Hace que la imagen sea fluida (max-width: 100%; height: auto).
<code>.rounded</code>	Bordes ligeramente redondeados.

<code>.rounded-circle</code>	Muestra la imagen como un círculo (ideal para avatares).
<code>.rounded-0</code>	Esquinas completamente cuadradas (elimina redondeo).
<code>.img-thumbnail</code>	Estilo de miniatura con borde y sombra ligeros.
<code>.float-start</code>	Flota la imagen a la izquierda (similar a <code>float: left</code>).
<code>.float-end</code>	Flota la imagen a la derecha (similar a <code>float: right</code>).
<code>.mx-auto</code>	Centra la imagen horizontalmente (requiere <code>.d-block</code> o <code>.d-flex</code>).
<code>.my-auto</code>	Centra la imagen verticalmente (en contenedores flex).
<code>.d-block</code>	Convierte la imagen en un elemento de bloque.
<code>.d-inline-block</code>	Mantiene la imagen en línea pero permite ajustar su ancho/alto.

10.1 Imagen Responsiva

La clase `.img-fluid` garantiza que la imagen se ajuste al ancho del contenedor y conserve su proporción de aspecto.

```

```

10.2 Bordos Redondeados

Bootstrap ofrece múltiples opciones de redondeado:

```
<!-- Bordos suavemente redondeados -->


<!-- Solo en esquinas superiores -->


<!-- Forma circular completa -->

```

10.3 Imágenes Tipo Miniatura

Las miniaturas (`.img-thumbnail`) añaden borde y algo de sombra, muy útiles para galerías.

```

```

10.4 Alineación de Imágenes

Para flotar a un lado del texto:

```

<p>Texto que fluye alrededor de la imagen...</p>
```

(`.me-3` añade margen a la derecha.)

11. Clases para Dar Estilo a los Botones en Bootstrap 5

Los botones en Bootstrap se basan en la clase `.btn` y se combinan con variantes de color para transmitir distintos significados.

Clase	Significado
<code>.btn</code>	Clase base (se usa siempre)
<code>.btn-primary</code>	Botón principal (azul)
<code>.btn-secondary</code>	Botón secundario (gris)
<code>.btn-success</code>	Éxito (verde)
<code>.btn-danger</code>	Peligro/error (rojo)
<code>.btn-warning</code>	Advertencia (amarillo)
<code>.btn-info</code>	Información (celeste)
<code>.btn-light</code>	Fondo claro
<code>.btn-dark</code>	Fondo oscuro
<code>.btn-link</code>	Botón con estilo de enlace (texto subrayado)
<code>.btn-outline-*</code>	Botón de contorno (borde y texto del color)

<code>.btn-lg</code> , <code>.btn-sm</code>	Botones grande o pequeño
<code>.disabled</code>	Desactiva interacciones del botón
<code>.active</code>	Estado activo (para toggles)
<code>.btn-group</code>	Agrupar botones horizontalmente
<code>.btn-group-vertical</code>	Agrupar botones verticalmente
<code>.btn-toolbar</code>	Contenedor para múltiples grupos de botones

11.1 Botones Básicos

```
<button class="btn btn-primary">Primary</button>
<button class="btn btn-danger">Danger</button>
<a href="#" class="btn btn-success">Enlace con estilo de
botón</a>
```

11.2 Botones de Contorno

Si prefieres un estilo más ligero, usa `.btn-outline-{color}`:

```
<button class="btn btn-outline-success">Outline Success
</button>
<button class="btn btn-outline-warning">Outline Warning
</button>
```

11.3 Tamaños y Formato de Bloque

```
<button class="btn btn-sm btn-info">Pequeño</button>
<button class="btn btn-lg btn-dark">Grande</button>
<!-- Botón de ancho completo al usar un grid o "d-grid"
-->
<div class="d-grid gap-2">
  <button class="btn btn-primary btn-lg">Ancho completo
</button>
</div>
```

11.4 Estado Desactivado o Activo

```
<button class="btn btn-secondary" disabled>Desactivado</button>
<button class="btn btn-primary active" aria-pressed="true">Activo</button>
```

11.5 Grupos de Botones

- **Horizontal:**

```
<div class="btn-group" role="group">
  <button class="btn btn-outline-primary">Opción 1</button>
  <button class="btn btn-outline-primary">Opción 2</button>
</div>
```

- **Vertical:**

```
<div class="btn-group-vertical" role="group">
  <button class="btn btn-outline-secondary">Vertical 1</button>
  <button class="btn btn-outline-secondary">Vertical 2</button>
</div>
```

12. Bordes, Padding y Margin en Bootstrap 5

12.1 Bordes

Bootstrap ofrece diversas clases para manipular bordes en tus elementos:

Clase	Descripción
<code>.border</code>	Añade un borde básico alrededor del elemento.
<code>.rounded</code>	Bordes redondeados (pequeño radio).
<code>.rounded-circle</code>	Forma circular (radio al 100%).
<code>.rounded-0</code>	Desactiva cualquier redondeo de bordes.

<code>.border-{color}</code>	Borde del color primario, secundario, etc. (<code>.border-primary</code> , <code>.border-success</code> , etc.)
<code>.border-0</code>	Elimina todos los bordes.

Ejemplo:

```
<div class="border border-primary rounded p-3">
  Contenido con borde azul y esquinas redondeadas
</div>
```

12.2 Padding (Espacio Interno) y Margin (Espacio Externo)

Las clases de espaciado siguen la convención `.m|p-{lado}-{tamaño}`:

- **m** = margin, **p** = padding
- **{lado}** puede ser `t` (top), `b` (bottom), `s` (start), `e` (end), `x` (izq/der), `y` (arriba/abajo) o vacío (aplica a los cuatro lados).
- **{tamaño}** varía de 0 a 5 (incrementos de 0.25rem), `auto`, etc.

Ejemplo	Resultado
<code>.m-0</code>	Margin 0 en todos los lados
<code>.mt-3</code>	Margin-top de 1rem (aprox.)
<code>.mx-auto</code>	Margin horizontal auto (centrado)
<code>.p-2</code>	Padding de 0.5rem en todos lados
<code>.py-4</code>	Padding vertical de 1.5rem

Ejemplo de uso:

```
<div class="p-3 m-auto bg-light border">
  Elemento con padding 3 y margin auto (centrado horizontalmente)
</div>
```

▼ ACCESIBILIDAD

1. Diseño de Webs Accesibles

El diseño de webs accesibles busca garantizar que los recursos en línea sean funcionales, accesibles y de alta calidad, mejorando la experiencia para todos los usuarios, incluidas las personas con discapacidades.

Consortio World Wide Web (W3C)

El W3C desarrolla estándares abiertos para asegurar la interoperabilidad y el crecimiento de la web.

Importancia de los Estándares

1. Compatibilidad entre navegadores y dispositivos.
2. Mejora de la accesibilidad.
3. Facilidad de mantenimiento y escalabilidad.
4. Optimización del rendimiento.
5. Seguridad.
6. Consistencia en la experiencia del usuario (UX/UI).
7. Mejor posicionamiento en buscadores (SEO).
8. Adopción de tecnologías futuras.
9. Interoperabilidad con otras tecnologías y servicios.
10. Reducción de costos y tiempo de desarrollo.

2. Evolución de un Estándar

Fase	Descripción
Working Draft (WD)	Borrador preliminar que describe una propuesta técnica.
Candidate Recommendation (CR)	Fase intermedia que permite la implementación general.
Proposed Recommendation (PR)	Versión casi final, lista para ser adoptada como estándar.
W3C Recommendation (REC)	Estándar oficial listo para implementación generalizada.

3. Pautas de Accesibilidad Web (WCAG)

Las WCAG son directrices del W3C para mejorar la accesibilidad web, dirigidas a:

- Diseñadores y desarrolladores web.
- Fabricantes de herramientas de desarrollo.
- Desarrolladores de herramientas de evaluación.

Principios de Accesibilidad WCAG

1. **Perceptible:** El contenido debe ser visible y entendible.
2. **Operable:** Los usuarios deben poder interactuar con el contenido de manera sencilla.
3. **Inteligible:** El contenido debe ser claro y comprensible.
4. **Robusto:** Compatible con diferentes herramientas y navegadores.

Niveles de Prioridad WCAG

Nivel	Requisitos
A	Cumple con los puntos de verificación esenciales.
AA	Cumple con los puntos de prioridad media y esencial.
AAA	Cumple con todos los puntos de verificación, incluidos los más exigentes.

4. Herramientas de Análisis de Accesibilidad

Herramienta	Descripción
TAW	Analiza automáticamente la conformidad con WCAG.
Wallace	Audita código CSS y evalúa rendimiento y complejidad.
Cynthia Says	Evalúa accesibilidad según WCAG 1.0 y la Sección 508 (EE.UU.).
WAVE	Identifica y corrige problemas de accesibilidad en páginas web.

5. Distribución Centralizada vs. CDN

Aspecto	Distribución Centralizada	Distribución CDN
Ubicación	Un único servidor o pocos en la misma	Múltiples servidores distribuidos globalmente.

	región.	
Latencia	Alta para usuarios lejanos.	Baja al usar servidores cercanos al usuario.
Escalabilidad	Limitada con alta demanda.	Alta, al distribuir el tráfico.
Disponibilidad	Vulnerable a fallos del servidor.	Alta disponibilidad con múltiples servidores.
Costos	Económico inicialmente, pero menos eficiente.	Costos adicionales, pero mejora rendimiento.
Seguridad	Vulnerable a ataques DDoS.	Mayor seguridad con mitigación de DDoS.

6. Protocolo HTTP

Cada elemento en una página web genera una conexión separada:

- Una conexión para el HTML.
- Conexiones individuales para CSS, JavaScript, imágenes y otros recursos.