

# Fundamentos, Componentes y Implementación Responsiva

## Sección 1: Características y Beneficios del Framework Bootstrap

Esta sección aborda la naturaleza fundamental del framework Bootstrap, analizando sus características principales, los beneficios estratégicos de su adopción y los métodos de implementación para su incorporación en un proyecto de desarrollo web.

### 1.1. ¿Qué es Bootstrap? Un Análisis Fundamental

Bootstrap es un framework de desarrollo front-end de código abierto, potente y de gran popularidad, diseñado para agilizar y estandarizar el proceso de creación de sitios web y aplicaciones web. Su propósito fundamental es proporcionar una base sólida de componentes de interfaz de usuario (UI) y un sistema de diseño (layout) que es, por defecto, responsivo y sigue el principio de *mobile-first*.<sup>1</sup>

En esencia, Bootstrap es una colección de archivos CSS y JavaScript listos para usar.<sup>2</sup>

- **Archivos CSS:** El CSS del framework proporciona un sistema de estilos global. Incluye `bootstrap.css` (el framework completo), `reboot.css` (una normalización de los estilos de los navegadores para corregir inconsistencias), `grid.css` (el sistema de rejilla flexbox) y `utilities.css` (una extensa biblioteca de clases de utilidad).<sup>2</sup>
- **Archivos JavaScript:** Los plugins de JavaScript añaden interactividad a componentes complejos como menús desplegados, carruseles y barras de navegación colapsables.

Una de las evoluciones estratégicas más significativas en Bootstrap 5 (lanzado en 2020) fue la eliminación de jQuery como dependencia.<sup>3</sup> Esta decisión representa un cambio fundamental en la filosofía del framework. jQuery fue creado originalmente para abstraer y simplificar las inconsistencias de la manipulación del DOM (Document Object Model) en navegadores antiguos, especialmente Internet Explorer. Dado que Bootstrap 5 eliminó el soporte para Internet Explorer<sup>3</sup>, esta capa de abstracción ya no era necesaria. Al eliminar jQuery, Bootstrap 5 se vuelve significativamente más ligero, rápido y se alinea con el ecosistema de desarrollo moderno, que depende de JavaScript "vanilla" (puro) y de frameworks como React, Angular y Vue, que gestionan el DOM de forma independiente.

Esta modernización también es evidente en su utillaje de compilación. Bootstrap 5 migró de Libsass a Dart Sass.<sup>4</sup> Para los desarrolladores, esto significa que el framework utiliza la implementación de referencia de Sass, que es más rápida y la primera en recibir nuevas características del lenguaje.

Versiones más recientes, como Bootstrap 5.3, han continuado esta evolución introduciendo características como el soporte nativo para modos de color (modo oscuro y claro). Esto se implementa de manera elegante a través de un atributo de datos, como `data-bs-theme="dark"` en la etiqueta `<html>`, permitiendo que toda la paleta de colores del sitio se adapte dinámicamente<sup>2</sup>, `.navbar`<sup>6</sup> o `.alert`<sup>7</sup> — los desarrolladores pueden ensamblar interfaces funcionales en una fracción del tiempo que llevaría escribir todo el CSS personalizado desde cero.

- **Responsividad "Mobile-First" Integrada:** El diseño responsivo no es una ocurrencia tardía en Bootstrap; es su núcleo. El sistema de grillas (Grid System) está diseñado bajo el principio *mobile-first*.<sup>1</sup> Esto significa que los estilos base están orientados a pantallas pequeñas (móviles), y las clases de *breakpoint* (puntos de interrupción) se utilizan para adaptar el diseño a pantallas más grandes. Esto garantiza que un sitio sea legible y funcional en un dispositivo móvil por defecto.
- **Consistencia entre Navegadores:** Uno de los mayores desafíos en el desarrollo front-end es la inconsistencia en cómo los diferentes navegadores (Chrome, Firefox, Safari) renderizan los elementos HTML. Bootstrap soluciona esto con `reboot.css`, un archivo que "resetea" y normaliza los estilos predefinidos de los navegadores, proporcionando un lienzo consistente sobre el cual construir.<sup>2</sup>
- **Mantenibilidad y Lenguaje Compartido:** En un entorno de equipo, el mayor beneficio de Bootstrap es que proporciona un "lenguaje de diseño" compartido y bien documentado. Un desarrollador puede entender inmediatamente el propósito de una clase como `.btn-primary` o `.alert-danger`.<sup>5</sup> Esto reduce la "deuda de diseño" (código CSS personalizado, anárquico y duplicado), facilita la incorporación de nuevos miembros al equipo (que solo necesitan saber Bootstrap, no un sistema interno) y hace que la base de código sea inherentemente más legible y mantenible.

### 1.3. Dónde Obtenerlo y Cómo Incorporarlo a un Proyecto HTML

Existen dos métodos principales para incluir Bootstrap en un documento HTML, cada uno con sus propias ventajas estratégicas.<sup>8</sup>

#### Método 1: Inicio Rápido con CDN (Content Delivery Network)

Este es el método más rápido y sencillo para empezar a usar Bootstrap<sup>8</sup>, ideal para prototipos, proyectos pequeños o sitios públicos. Utiliza un CDN, como jsDelivr, para enlazar los archivos compilados directamente.<sup>9</sup>

Para implementarlo, se debe utilizar una plantilla HTML5 estándar e insertar las siguientes líneas:

1. **Enlace CSS:** El archivo CSS de Bootstrap debe colocarse dentro de la etiqueta `<head>` del documento, antes de cualquier otra hoja de estilos personalizada.
2. **Enlace JavaScript:** El archivo JavaScript debe colocarse justo antes del cierre de la etiqueta `</body>`.

### Plantilla de Inicio Rápido (CDN):

HTML

```
<!doctype html>

<html lang="es">

  <head>

    <meta charset="utf-8">

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

    <title>Sitio Básico con Bootstrap</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnD
JzQlu9" crossorigin="anonymous">

  </head>

  <body>

    <h1>¡Hola, mundo!</h1>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-HwwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgr
km" crossorigin="anonymous"></script>

  </body>
```

</html>

- **Ventajas del CDN:**

- **Menor Carga del Servidor:** Los archivos se sirven desde servidores optimizados y distribuidos globalmente, no desde el servidor propio.<sup>10</sup>
- **Reducción de Latencia:** Los usuarios reciben los archivos desde el servidor geográficamente más cercano a ellos.<sup>10</sup>
- **Caching del Navegador:** Si un usuario ha visitado previamente otro sitio que utiliza la misma versión de Bootstrap desde el mismo CDN, el archivo ya estará en la caché de su navegador, lo que resulta en tiempos de carga casi instantáneos.<sup>10</sup>

- **Desventajas del CDN:**

- **Dependencia Externa:** Si el CDN falla o está inaccesible, los estilos y la funcionalidad de Bootstrap no se cargarán.<sup>10</sup>
- **Requiere Conexión:** No es posible realizar pruebas o desarrollo en un entorno local sin conexión a Internet.<sup>10</sup>

## Método 2: Instalación de Archivos Locales Compilados

Este método implica descargar los archivos CSS y JavaScript compilados de Bootstrap y alojarlos en el propio servidor del proyecto.<sup>8</sup> Es el método preferido para aplicaciones empresariales, sitios de intranet o cualquier proyecto donde la fiabilidad y el control total de las dependencias sean críticos.

El proceso implica:

1. Visitar la página de descargas de Bootstrap.<sup>12</sup>
2. Descargar el archivo ZIP de "Compiled CSS and JS" (CSS y JS Compilados).<sup>12</sup>
3. Descomprimirlo y colocar las carpetas `css/` y `js/` en el directorio del proyecto.
4. Enlazar los archivos localmente en el HTML:

HTML

<!doctype html>

<html lang="es">

```
<head>

<meta charset="utf-8">

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

<title>Sitio Local con Bootstrap</title>

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

</head>

<body>

<h1>¡Hola, mundo!</h1>

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

</body>

</html>
```

Es importante notar que esta descarga de archivos compilados no incluye la documentación ni los archivos fuente de Sass.<sup>12</sup>

### La Importancia del "Bundle" (Paquete)

En ambos métodos (CDN y local), se enlaza `bootstrap.bundle.min.js`. Este "bundle" (paquete) es crucial porque incluye la biblioteca **Popper.js**.<sup>2</sup> Popper.js es una pequeña biblioteca de posicionamiento que Bootstrap *requiere* para el funcionamiento de componentes dinámicos como menús desplegados (Dropdowns), información emergente (Popovers) y descripciones emergentes (Tooltips).<sup>9</sup> Si un desarrollador enlaza erróneamente solo `bootstrap.min.js` (sin el "bundle"), descubrirá que estos componentes no funcionan correctamente. Por lo tanto, el "bundle" es la opción recomendada y a prueba de fallos.

## Sección 2: Elementos y Estilos Principales de Bootstrap

Esta sección cataloga y analiza los componentes fundamentales, los sistemas de layout y los estilos básicos que constituyen el núcleo del framework Bootstrap 5.

### 2.1. El Fundamento del Layout: Contenedores (Containers)

Los contenedores son el bloque de construcción más fundamental en Bootstrap. Son un elemento de envoltura (wrapper) obligatorio para el contenido del sitio y son necesarios para albergar el sistema de grillas.<sup>1</sup> Su propósito principal es contener, aplicar relleno (*padding*) y alinear el contenido dentro de un *viewport* (la ventana visible del navegador).<sup>14</sup>

Bootstrap 5 ofrece tres tipos principales de contenedores:

1. **.container** (Contenedor por defecto):  
Esta es la clase de contenedor más común. Es un contenedor responsivo de ancho fijo. Esto significa que su propiedad *max-width* (ancho máximo) cambia y se ajusta en cada uno de los breakpoints (puntos de interrupción) de pantalla.<sup>13</sup> Esto crea un layout que se centra en la página con márgenes a los lados en pantallas grandes, pero ocupa todo el ancho en pantallas móviles pequeñas.
2. **.container-fluid** (Contenedor fluido):  
Esta clase se utiliza para crear un contenedor de ancho completo, que abarca el 100% del ancho del *viewport* en todos los tamaños de pantalla, desde el más pequeño hasta el más grande.<sup>13</sup> Su estilo es, simplemente, *width: 100%*.
3. **.container-{breakpoint}** (Contenedores responsivos híbridos):  
Estas clases (ej. **.container-md**, **.container-lg**) ofrecen un comportamiento híbrido. Un contenedor con **.container-md** se comportará como un **.container-fluid** (ocupando el *width: 100%*) en pantallas más pequeñas que el breakpoint md (Medium, 768px). Una vez que el *viewport* alcanza o supera ese breakpoint, el contenedor se comportará como un **.container** de ancho fijo.<sup>14</sup>

La siguiente tabla resume cómo cambian los anchos máximos (*max-width*) de los contenedores en los diferentes *breakpoints* responsivos.

**Tabla 1: Comparativa de *max-width* de Contenedores por Breakpoint**<sup>14</sup>

Clase	Extra small (<576px)	Small (≥576px)	Medium (≥768px)	Large (≥992px)	X-Large (≥1200px)	XX-Large (≥1400px)
<b>.container</b>	100%	540px	720px	960px	1140px	1320px

<b>.container-sm</b>	100%	540px	720px	960px	1140px	1320px
<b>.container-md</b>	100%	100%	720px	960px	1140px	1320px
<b>.container-lg</b>	100%	100%	100%	960px	1140px	1320px
<b>.container-xl</b>	100%	100%	100%	100%	1140px	1320px
<b>.container-xxl</b>	100%	100%	100%	100%	100%	1320px
<b>.container-fluid</b>	100%	100%	100%	100%	100%	100%

## 2.2. El Sistema de Grillas (Grid System)

El sistema de grillas de Bootstrap es la herramienta principal para crear layouts responsivos. Es un potente sistema *mobile-first* construido con **flexbox**<sup>1</sup> que permite organizar el contenido en una serie de filas y columnas.

### Principios de Funcionamiento (12 Columnas)

El sistema se basa en una cuadrícula conceptual de **12 columnas** por cada fila.<sup>1</sup> Esta elección de 12 es intencional: es un número altamente divisible (por 2, 3, 4, 6), lo que permite una gran flexibilidad para crear diferentes divisiones de layout (ej. dos columnas de 6, tres de 4, cuatro de 3, etc.).

La estructura jerárquica del sistema de grillas es estricta y debe seguirse para un funcionamiento correcto<sup>1</sup>:

1. **Contenedor (Container):** Se necesita un `.container` (fijo) o `.container-fluid` (fluido) para envolver todo el sistema.<sup>13</sup>
2. **Fila (Row):** Dentro del contenedor, se usa la clase `.row` para crear un grupo horizontal de columnas. Las filas son *wrappers* (envoltorios) para las columnas.<sup>1</sup>
3. **Columnas (Column):** El contenido visual debe colocarse dentro de las columnas (ej. `.col-6`). Solo las columnas pueden ser hijos inmediatos de una `.row`.<sup>13</sup>

Las filas (`.row`) utilizan márgenes horizontales negativos para contrarrestar el *padding* (relleno) horizontal que tienen las columnas (conocido como *gutter* o canal). Esto asegura que el contenido dentro de las columnas se alinee perfectamente con el borde del contenedor padre.<sup>1</sup>

### Clases de Breakpoint: El Núcleo de la Responsividad

Bootstrap implementa su filosofía *mobile-first* a través de seis niveles (o *tiers*) de *breakpoints* responsivos. Un *breakpoint* es el punto de ancho del *viewport* en el que el layout puede cambiar.

Tabla 2: Opciones del Sistema de Grillas por Breakpoint<sup>1</sup>

Nivel de Grilla	Prefijo de Clase	Ancho Mínimo del Viewport
Extra small (xs)	<code>.col-</code>	< 576px (Implicito)
Small (sm)	<code>.col-sm-</code>	≥ 576px
Medium (md)	<code>.col-md-</code>	≥ 768px
Large (lg)	<code>.col-lg-</code>	≥ 992px
Extra large (xl)	<code>.col-xl-</code>	≥ 1200px



Extra extra large (xxl)	<code>.col-xxl-</code>	$\geq 1400\text{px}$
-------------------------	------------------------	----------------------

La lógica *mobile-first* significa que las clases se aplican desde el *breakpoint* especificado **hacia arriba** (es decir, se basan en `min-width`).<sup>1</sup> Por ejemplo, si se aplica la clase `.col-md-6` a un elemento:

- En pantallas *pequeñas* (menos de 768px), la clase no se aplica, y el elemento se comportará como un bloque (generalmente ocupando el 100% del ancho, apilándose).
- En pantallas *medianas* (768px) y *todas las superiores* (lg, xl, xxl), la clase `.col-md-6` se activará, y el elemento ocupará 6 de las 12 columnas (es decir, el 50% del ancho de la fila).<sup>1</sup>

Este enfoque permite el patrón de diseño "apilado a horizontal" (*stacked-to-horizontal*). Un desarrollador define el layout apilado para móviles (a menudo usando `.col-12` o dejando el comportamiento por defecto) y luego usa clases de *breakpoint* más grandes (como `.col-lg-4`) para definir cómo se reorganizan esas columnas en pantallas de escritorio.<sup>1</sup>

### Columnas de Auto-Layout (Flexbox)

Gracias a flexbox, no siempre es necesario especificar un número.<sup>16</sup>

- **.col:** Si se colocan múltiples `divs` con solo la clase `.col` dentro de una `.row`, se distribuirán automáticamente el espacio disponible por igual. Tres `div.col` crearán un layout de tres columnas (33.3% cada una).<sup>1</sup>
- **.col-auto:** La columna ocupará únicamente el ancho que necesite su contenido interno.<sup>1</sup>

## 2.3. Componentes de Contenido y Estilos Básicos

Bootstrap proporciona un conjunto robusto de estilos para elementos HTML comunes.

### Tablas (Tables)

Los estilos de tabla en Bootstrap son "opt-in", lo que significa que no se aplican automáticamente a las etiquetas `<table>`. Para aplicar el estilo base de Bootstrap, se debe añadir la clase `.table`.<sup>17</sup>

Estilos modificadores clave<sup>17</sup>:

- **.table-striped**: Añade un rayado de cebra (filas alternas con un fondo ligeramente más oscuro) a las filas dentro del `<tbody>`.
- **.table-hover**: Habilita un efecto de resaltado (hover) en las filas del `<tbody>` cuando el cursor pasa por encima.
- **.table-bordered**: Añade bordes a todos los lados de la tabla y sus celdas.
- **.table-sm**: Reduce el *padding* (relleno) de las celdas a la mitad para una tabla más compacta.

**Responsividad de Tablas:** Las tablas HTML, por naturaleza, pueden ser muy anchas y romper el layout en dispositivos móviles. Bootstrap soluciona esto requiriendo que la `<table>` sea envuelta en un `<div>` con la clase `.table-responsive`.<sup>17</sup> Esto no encoge la tabla, sino que permite que la tabla se desborde horizontalmente y añade una barra de scroll horizontal solo a la tabla, salvaguardando el layout de la página. También existen variantes como `.table-responsive-md`, que hacen que la tabla sea responsiva solo *por debajo* del *breakpoint* md.<sup>17</sup>

### Jumbotron (Deprecado y Reemplazado)

Un cambio notable en Bootstrap 5 es la **eliminación** (deprecación) del componente `.jumbotron`.<sup>3</sup> En versiones anteriores, `.jumbotron` era un gran contenedor gris usado para destacar contenido clave.

Esta eliminación fue una decisión filosófica deliberada, que favorece el uso de **clases de utilidad (utility classes)** para lograr el mismo efecto con mayor flexibilidad.<sup>19</sup> En lugar de estar limitados por los estilos de `.jumbotron`, los desarrolladores ahora *componen* su propio "héroe" (como se les llama en los ejemplos de B5<sup>21</sup>) combinando utilidades.

Para recrear un Jumbotron en Bootstrap 5, se usaría un `<div>` y se le aplicarían clases de utilidad para el fondo, el espaciado y la tipografía.<sup>19</sup>

### Ejemplo de Reemplazo de Jumbotron:

HTML

```
<div class="container-fluid p-5 bg-light rounded-3">

  <div class="container">

    <h1 class="display-4">¡Componente Héroe!</h1>

    <p class="lead">
```

Este es un reemplazo moderno del jumbotron, construido con clases de utilidad para espaciado (p-5), color de fondo (bg-light) y tipografía (display-4, lead).

</p>

</div>

</div>

## Alertas (Alerts)

Las alertas se utilizan para proporcionar mensajes de retroalimentación contextual. Se construyen con una clase base `.alert` y una clase contextual que define el color y el propósito semántico.<sup>7</sup>

- Clases contextuales: `.alert-primary`, `.alert-secondary`, `.alert-success` (éxito), `.alert-danger` (peligro), `.alert-warning` (advertencia), `.alert-info` (información), `.alert-light`, `.alert-dark`.<sup>25</sup>
- **Enlaces en Alertas:** Para que los enlaces coincidan con el color de la alerta, se debe usar la clase de utilidad `.alert-link`.<sup>7</sup>
- **Alertas Descartables (Requiere JavaScript):** Para permitir que un usuario cierre una alerta, se requiere una estructura específica<sup>25</sup>:
  - Añadir la clase `.alert-dismissible` al `<div>` de la alerta.
  - Añadir las clases `.fade` y `.show` para permitir la animación de desvanecimiento.
  - Añadir un botón de cierre: `<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>`.
  - El atributo `data-bs-dismiss="alert"` es el gancho de JavaScript que activa la funcionalidad de descarte.<sup>25</sup>

**Nota de Accesibilidad:** La documentación de Bootstrap advierte que usar *solo* el color (ej. `.alert-danger`) para transmitir significado no es accesible para usuarios con lectores de pantalla o discapacidades visuales. El texto dentro de la alerta debe comunicar explícitamente el estado (ej. "Error: El campo es obligatorio").<sup>7</sup>

## Botones (Buttons)

Bootstrap proporciona una amplia gama de estilos de botones. Se requiere una clase base `.btn` aplicada a un elemento `<button>`, `<a>` o `<input>`.<sup>5</sup>

- **Clases Contextuales:** Se aplican junto con `.btn` para definir la apariencia: `.btn-primary` (acción principal), `.btn-secondary`, `.btn-success`, `.btn-danger`, `.btn-warning`, `.btn-info`, `.btn-light`, `.btn-dark`, `.btn-link` (apariencia de enlace).<sup>5</sup>
- **Estilos de Contorno (Outline):** Para botones con borde pero sin fondo sólido, se usan las clases `.btn-outline-*` (ej. `.btn-outline-primary`).<sup>5</sup>
- **Tamaños:** Se puede controlar el tamaño con `.btn-lg` (grande) y `.btn-sm` (pequeño).<sup>5</sup>

**Accesibilidad de Roles:** Cuando se utiliza una etiqueta `<a>` (que semánticamente es un enlace de navegación) para que *funcione* como un botón (es decir, para activar una funcionalidad en la página como un colapso, en lugar de navegar a una nueva URL), es crucial añadir el atributo `role="button"`. Esto informa a las tecnologías de asistencia sobre el propósito real del elemento.<sup>5</sup>

## Imágenes (Images)

Bootstrap ofrece clases de utilidad para controlar la apariencia y responsividad de las imágenes.

- **Imágenes Responsivas:** La clase más importante es `.img-fluid`. Esta clase aplica `max-width: 100%;` y `height: auto;` a la imagen. Esto permite que la imagen escale proporcionalmente para ajustarse al ancho de su contenedor padre, evitando que se desborde en dispositivos móviles.<sup>26</sup>
- **Estilos:**
  - `.img-thumbnail:` Aplica un *padding* ligero y un borde redondeado de 1px.<sup>26</sup>
  - `.rounded:` Aplica un `border-radius` sutil a la imagen.<sup>26</sup>
- **Alineación:** Las imágenes se alinean usando clases de utilidad. Para centrar una imagen de nivel de bloque, se usan `.mx-auto` (margen horizontal automático) y `.d-block` (display block).<sup>26</sup> Para flotar, se usan `.float-start` y `.float-end`.<sup>26</sup>

Al utilizar el elemento HTML5 `<picture>` (para arte responsivo), las clases de estilo e imagen (como `.img-fluid`) deben aplicarse a la etiqueta `<img>` que se encuentra *dentro* del `<picture>`, no al propio elemento `<picture>`.<sup>26</sup>

## 2.4. Componentes Complejos de Interfaz (Composites)

Estos componentes combinan múltiples elementos y, a menudo, dependen de los plugins de JavaScript de Bootstrap para su funcionalidad completa.

### Barras de Navegación (Navbars)

Las Navbars son componentes compuestos y responsivos que sirven como la cabecera de navegación principal de un sitio. Su estructura es compleja y depende tanto de clases CSS como de atributos de JavaScript.<sup>6</sup>

- **Estructura Base:** El contenedor principal utiliza la clase `.navbar`.
- **Comportamiento Responsivo (El "Hamburger"):**
  - **Clase de Expansión:** La clase `.navbar-expand-{breakpoint}` (ej. `.navbar-expand-lg`) es el control de responsividad más importante.<sup>6</sup>  
Determina en qué *breakpoint* (lg en este caso) la barra de navegación pasará de su estado colapsado (móvil, mostrando el botón "hamburger") a su estado expandido (escritorio, mostrando los enlaces horizontalmente).<sup>6</sup>
  - **El Botón "Hamburger":** Este es un `<button>` con la clase `.navbar-toggler`. Es visible solo por debajo del *breakpoint* de expansión.<sup>6</sup>
  - **Contenido Colapsable:** Los enlaces de navegación deben estar envueltos en un `<div>` con las clases `.collapse` y `.navbar-collapse`.<sup>6</sup>
- **Dependencia de JavaScript:** Para que el botón *toggler* funcione, debe estar vinculado al `div` colapsable. Esto se logra mediante atributos `data-bs-*` que dependen del plugin de colapso de JavaScript de Bootstrap.<sup>6</sup>
  - El `<button class="navbar-toggler">` debe tener `data-bs-toggle="collapse"`.
  - El `<button>` debe apuntar al `div` colapsable mediante `data-bs-target="#idDelContenido"`.<sup>6</sup>
  - El `<div>` colapsable debe tener el `id` correspondiente (ej. `id="idDelContenido"`).
  - Si el archivo `bootstrap.bundle.min.js` no está enlazado, el botón "hamburger" aparecerá en móviles pero no funcionará al hacer clic.<sup>9</sup>
- **Contenido Interno:**
  - `.navbar-brand`: Se usa para el logo o nombre del sitio.<sup>6</sup>
  - `.navbar-nav`: Se aplica a un `<ul>` para la lista principal de enlaces.<sup>6</sup>
  - `.nav-item` y `.nav-link`: Se aplican a los `<li>` y `<a>` dentro de la lista de navegación, respectivamente.

## Formularios (Forms)

Bootstrap rediseña los controles de formulario HTML estándar para una apariencia moderna y responsiva.<sup>30</sup>

- **Clases de Control Principales:**

- **.form-control:** Esta es la clase más importante para los formularios. Se aplica a `<input>`, `<textarea>`, y `<select>`. Les da un estilo responsivo, aplicando `width: 100%` y normalizando la apariencia.<sup>30</sup>
- **.form-label:** Se aplica a las etiquetas `<label>` para un espaciado y ponderación de fuente adecuados.<sup>30</sup>

- **Estructura y Agrupación:**

- Bootstrap 5 eliminó la antigua clase `.form-group`. En su lugar, fomenta el uso de **clases de utilidad de espaciado** para agrupar etiquetas y controles.
- La práctica estándar es envolver un par de `.form-label` y `.form-control` en un `<div>` y aplicarle una utilidad de margen, como `.mb-3` (que significa `margin-bottom: 1rem`).<sup>30</sup> Este enfoque de composición de utilidades es central en Bootstrap 5.

- **Otros Controles:**

- **.form-check:** Se utiliza como un *wrapper* (`div`) para agrupar `<input type="checkbox">` o `<input type="radio">` y sus respectivas etiquetas (`.form-check-label`, `.form-check-input`).<sup>30</sup>
- **.form-text:** Se usa para texto de ayuda debajo de un campo de entrada.<sup>30</sup>
- **Tamaños:** Se pueden usar clases como `.form-control-lg` o `.form-control-sm` en los *inputs*.<sup>30</sup>

## Sección 3: Aplicación Práctica: Implementación de un Sitio Web Básico Responsivo

Esta sección sintetiza los elementos de las secciones 4.1 y 4.2 en un tutorial práctico. El objetivo es construir un documento HTML único que demuestre cómo utilizar los estilos y componentes de Bootstrap para organizar visualmente una página de perfil de usuario básica y resolver el problema de la maquetación responsiva (correspondiente a la consulta 4.3).

### 3.1. Paso 1: El Documento HTML Base y la Configuración de Bootstrap

Se comenzará con una plantilla HTML5 estándar. Para esta implementación, se utilizará el método **CDN** (Content Delivery Network) por su rapidez.<sup>9</sup>

Se añade el enlace al archivo CSS (`bootstrap.min.css`) dentro de la etiqueta `<head>` y el enlace al archivo JavaScript (`bootstrap.bundle.min.js`) justo antes del cierre de la etiqueta `</body>`.<sup>9</sup> El "bundle" se utiliza para asegurar que componentes como el *toggler* de la barra de navegación funcionen correctamente.<sup>9</sup>

### Código HTML Base:

HTML

```
<!doctype html>

<html lang="es">

  <head>

    <meta charset="utf-8">

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

    <title>Perfil de Usuario - Bootstrap</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXhOJMHjY6hW+ALE
wIH" crossorigin="anonymous">

  </head>

  <body>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB6ONNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jleH
z" crossorigin="anonymous"></script>

  </body>

</html>
```

## 3.2. Paso 2: Creación de la Navegación Principal

Se implementará una barra de navegación responsiva (.navbar) en la parte superior de la página. Se utilizará .navbar-expand-lg para que la navegación se colapse en un botón "hamburger" en pantallas medianas y pequeñas (por debajo de 992px).<sup>6</sup> Se incluye un .navbar-brand para el título del sitio y una lista .navbar-nav para los enlaces.<sup>6</sup>

### Código de Navbar (Insertado dentro de <body>):

#### HTML

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

  <div class="container-fluid">

    <a class="navbar-brand" href="#">PerfilPro</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNav">

      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">

        <li class="nav-item">

          <a class="nav-link active" aria-current="page" href="#">Inicio</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="#">Portafolio</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="#">Contacto</a>

        </li>

      </ul>

    </div>

  </div>
```



</div>

</nav>

### 3.3. Paso 3: Diseño del Encabezado "Hero" (Reemplazo de Jumbotron)

Dado que `.jumbotron` está obsoleto<sup>4</sup>, se creará un encabezado de "héroe" moderno utilizando únicamente clases de utilidad. Se usará un `.container-fluid` con utilidades de *padding* (`.p-5`), color de fondo (`.bg-light`) y centrado de texto (`.text-center`).<sup>19</sup>

**Código de Héroe (Insertado después de `<nav>`):**

HTML

```
<header class="container-fluid p-5 bg-light text-center">

  <div class="container">

    <h1 class="display-4">Ana García</h1>

    <p class="lead">Desarrolladora Web Front-End y Diseñadora UX/UI</p>

  </div>

</header>
```

### 3.4. Paso 4: Implementación del Layout de Contenido con Grillas

Ahora se organizará el contenido principal. Se utilizará un `.container` estándar (de ancho fijo en escritorio) y dentro de él, una `.row`.<sup>1</sup>

El layout será de dos columnas que se apilarán en dispositivos móviles.

1. Una columna principal para el contenido (8/12 de ancho en escritorio).
2. Una barra lateral (sidebar) para un formulario (4/12 de ancho en escritorio).

Esto se logra con la estrategia *mobile-first* "apilado a horizontal".<sup>1</sup> Las clases `.col-12` se aplican a ambas columnas (diseño apilado por defecto). Las clases `.col-lg-8` y `.col-lg-4` se añaden para *modificar* ese comportamiento en el *breakpoint* `lg` ( $\geq 992\text{px}$ ), creando el layout de dos columnas.<sup>1</sup>

Código de Grilla (Insertado después de `<header>`):

HTML

```
<main class="container mt-4 mb-5"> <div class="row">
```

```
  <div class="col-12 col-lg-8">
```

```
    <h2>Acerca de Mí</h2>
```

```
    <hr>
```

```
  </div>
```

```
  <div class="col-12 col-lg-4">
```

```
    <h3>Contactar</h3>
```

```
    <hr>
```

```
  </div>
```

```
</div>
```

```
</main>
```



### 3.5. Paso 5: Estilizado de Elementos de Contenido

Finalmente, se poblarán las columnas creadas con los componentes de Bootstrap.

En la Columna Principal (`.col-lg-8`):

Se añadirá una imagen de perfil, una alerta de notificación y una tabla de habilidades.

- La imagen usará `.img-fluid` para ser responsiva y `.rounded-circle` para darle forma circular.<sup>28</sup>

- La alerta usará `.alert` y `.alert-info`.<sup>25</sup>
- La tabla usará `.table` y `.table-striped` para su estilo.<sup>17</sup>

En la Barra Lateral (`.col-lg-4`):

Se añadirá un formulario de contacto.

- Cada grupo de campo usará un `div.mb-3`.<sup>30</sup>
- Las etiquetas usarán `.form-label`.<sup>30</sup>
- Los campos de entrada usarán `.form-control`.<sup>30</sup>
- El botón de envío usará `.btn` y `.btn-primary`.<sup>5</sup>

#### Código de Contenido Final (Reemplazando los comentarios del Paso 4):

HTML

```
<div class="col-12 col-lg-8">

<h2>Acerca de Mí</h2>

<hr>



<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam scelerisque leo nec
lacus laoreet, ac egestas odio viverra. Praesent sed maximus magna. Integer ac sem eu orci
commodo tristique.</p>

<div class="alert alert-info mt-4" role="alert">

    ¡Actualmente disponible para proyectos freelance!

</div>
```

```
<h3 class="mt-4">Habilidades Técnicas</h3>
```

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

```
<thead>
```

```
<tr>
```

```
<th scope="col">Tecnología</th>
```

```
<th scope="col">Nivel de Experiencia</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>HTML5 & CSS3</td>
```

```
<td>Experto</td>
```

```
</tr>
```

```
<tr>
```

```
<td>JavaScript (ES6+)</td>
```

```
<td>Avanzado</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Bootstrap 5</td>
```

```
<td>Avanzado</td>
```

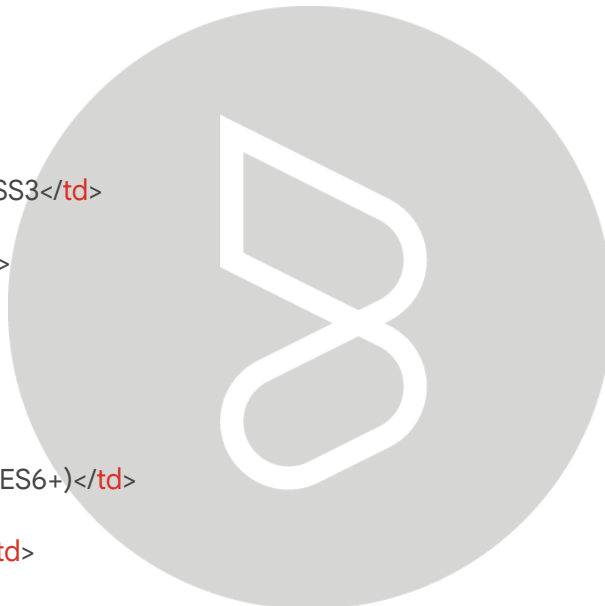
```
</tr>
```

```
<tr>
```

```
<td>React</td>
```

```
<td>Intermedio</td>
```

```
</tr>
```



```
</tbody>
```

```
</table>
```

```
</div>
```

```
<div class="col-12 col-lg-4">
```

```
<h3>Contactar</h3>
```

```
<hr>
```

```
<form>
```

```
<div class="mb-3">
```

```
<label for="emailInput" class="form-label">Dirección de Email</label>
```

```
<input type="email" class="form-control" id="emailInput"
placeholder="nombre@ejemplo.com">
```

```
</div>
```

```
<div class="mb-3">
```

```
<label for="messageTextarea" class="form-label">Mensaje</label>
```

```
<textarea class="form-control" id="messageTextarea" rows="5"></textarea>
```

```
</div>
```

```
<button type="submit" class="btn btn-primary w-100">Enviar Mensaje</button>
```

```
</form>
```

```
</div>
```

### 3.6. Conclusión de la Implementación

El documento HTML resultante utiliza un conjunto de componentes y utilidades de Bootstrap 5 para resolver el problema de organización visual (Consulta 4.3). El `Navbar`<sup>6</sup> gestiona la navegación. El "Héroe" (reemplazo de `Jumbotron`<sup>19</sup>) crea un encabezado destacado. El `Grid System` (`.container`, `.row`, `.col-12`, `.col-lg-*`)<sup>1</sup> organiza el contenido principal en un layout de dos columnas que se apila correctamente en dispositivos móviles. Finalmente, los componentes (`Image`<sup>28</sup>, `Alert`<sup>25</sup>, `Table`<sup>17</sup>, `Form`<sup>30</sup>, `Button`<sup>5</sup>) se utilizan para estilizar el contenido dentro de las columnas de la grilla. El resultado es una página web cohesiva, visualmente organizada y completamente responsiva.

