

**Técnico en Ingeniería en Ciencias de la Computación**

Facultad de Ingeniería

**“El uso de Bootstrap como framework de HTML”**

**Integrantes:**

* Daniela Nicole García Guillén GG251303
* Rosa Lucia Villalobos Flores VF250110

**Docente**: Ing. Edwin Bonilla

**Asignatura**: Lenguaje de Marcado y Estilo Web

**Fecha**: 09/05/2025

**Índice:**

[**Introducción:** 4](#_Toc197639656)

[**Definición y conceptos básicos de Bootstrap** 5](#_Toc197639657)

[**Características principales** 5](#_Toc197639658)

[**¿Cómo funciona Bootstrap?** 5](#_Toc197639659)

[***Ejemplo básico de uso:*** 6](#_Toc197639660)

[**Ventajas de usar Bootstrap** 6](#_Toc197639661)

[**Desventajas de usar Bootstrap** 6](#_Toc197639662)

[**Diseñando con rejilla en Bootstrap** 8](#_Toc197639663)

[**Estructura básica de la rejilla** 8](#_Toc197639664)

[**Funcionamiento y reglas del sistema de rejilla** 8](#_Toc197639665)

[***Ejemplo de estructura básica*** 9](#_Toc197639666)

[**Diseño responsivo** 9](#_Toc197639667)

[**Anidamiento de rejillas** 9](#_Toc197639668)

[**Tipos de rejillas en Bootstrap** 10](#_Toc197639669)

[**Sistema de columnas y breakpoints** 10](#_Toc197639670)

[**Manejo de filas y columnas** 11](#_Toc197639671)

[**Filas (.row)** 11](#_Toc197639672)

[**Columnas (.col-\*)** 11](#_Toc197639673)

[**Breakpoints** 12](#_Toc197639674)

[***Ejemplo*** 12](#_Toc197639675)

[**Consideraciones adicionales** 12](#_Toc197639676)

[**Tipos de contenedores** 13](#_Toc197639677)

[**Contenedor fijo y responsive (.container):** 13](#_Toc197639678)

[**Contenedor fluido (.container-fluid):** 13](#_Toc197639679)

[**Contenedores responsivos con breakpoint específico (.container-{breakpoint}):** 14](#_Toc197639680)

[***Tabla de anchos máximos para .container:*** 14](#_Toc197639681)

[**Imágenes responsivas en Bootstrap** 15](#_Toc197639682)

[**¿Qué hace .img-fluid?** 15](#_Toc197639683)

[**Cómo usar .img-fluid** 15](#_Toc197639684)

[**Ventajas** 15](#_Toc197639685)

[**Clases adicionales para imágenes en Bootstrap** 16](#_Toc197639686)

[**Recomendaciones** 16](#_Toc197639687)

[**Tablas responsivas en Bootstrap** 17](#_Toc197639688)

[**Cómo hacer una tabla responsiva con Bootstrap** 17](#_Toc197639689)

[**Clases útiles para mejorar la tabla** 18](#_Toc197639690)

[***Ejemplo combinando clases:*** 18](#_Toc197639691)

[**Adaptación a diferentes tamaños de pantalla** 18](#_Toc197639692)

[**Ventajas de las tablas responsivas en Bootstrap** 18](#_Toc197639693)

[**Formularios responsivos con Bootstrap** 19](#_Toc197639694)

[**Claves para crear formularios responsivos** 19](#_Toc197639695)

[***Ejemplo básico de formulario responsivo*** 20](#_Toc197639696)

[**Ventajas adicionales** 20](#_Toc197639697)

[**Conclusión:** 21](#_Toc197639698)

[**Bibliografía:** 22](#_Toc197639699)

# **Introducción:**

En el desarrollo de webs moderno, la creación de interfaces atractivas, funcionales y adaptables a cualquier dispositivo es una necesidad fundamental. Para facilitar este proceso y optimizar el tiempo de trabajo, existen sitios que ofrecen las herramientas para brindar soluciones prácticas y eficientes. Uno de los más populares y utilizados a nivel mundial es **Bootstrap**.

Bootstrap es un framework front-end de código abierto que proporciona un conjunto de estilos, componentes y utilidades predefinidas, permitiendo a los desarrolladores construir sitios web y aplicaciones responsivas de manera rápida y sencilla. Gracias a su enfoque modular y a su amplia documentación, Bootstrap se ha convertido en la opción preferida tanto para principiantes como para profesionales del desarrollo web.

En este documento se presentan algunos de los conceptos básicos de Bootstrap, enfocándonos en las funciones responsivas, sus principales características, ventajas y desventajas, así como una guía práctica sobre el uso de su sistema de rejilla, contenedores, imágenes y tablas responsivas; mostrando la interfaz básica de cada uno de ellos.

El objetivo de esta investigación es ofrecer una visión clara y accesible que facilite la comprensión y aplicación de Bootstrap en futuros proyectos web. De esta manera, se busca apoyar a programadores en formación y a personas interesadas en la tecnología, ayudándoles a profundizar en este ámbito y a optimizar tanto el diseño como la experiencia de usuario en dispositivos de diversas características.

# **Definición y conceptos básicos de Bootstrap**

Bootstrap es un framework front-end gratuito de código abierto, que facilita la creación de sitios web y aplicaciones web responsivas y adaptables a dispositivos móviles. Su principal objetivo es simplificar y acelerar el proceso de desarrollo web, permitiendo a los desarrolladores construir interfaces atractivas y funcionales sin necesidad de escribir grandes cantidades de código CSS y JavaScript desde cero.

## **Características principales**

* **Componentes listos para usar:** Incluye una amplia variedad de componentes como botones, menús de navegación, formularios, tarjetas, modales, barras de progreso y más, todos diseñados para ser fácilmente personalizables y reutilizables.
* **Consistencia visual:** Ofrece un conjunto coherente de estilos predeterminados, asegurando una apariencia uniforme en todas las partes de una aplicación o sitio web.
* **Clases utilitarias:** Proporciona numerosas clases para ajustar rápidamente márgenes, alineaciones, colores, tamaños y otros aspectos visuales sin necesidad de escribir CSS personalizado.
* **Compatibilidad y extensibilidad:** Bootstrap es compatible con los navegadores más populares y permite personalizar sus estilos mediante preprocesadores como Sass.
* **Documentación completa:** Cuenta con una documentación extensa y bien organizada, con ejemplos y guías para facilitar su aprendizaje y uso.

## **¿Cómo funciona Bootstrap?**

Bootstrap se compone principalmente de archivos CSS y JavaScript. Para utilizarlo, basta con incluir estos archivos en el proyecto web, ya sea descargándolos o a través de un CDN. Los estilos y funcionalidades se aplican mediante clases específicas que se agregan a los elementos HTML.

### ***Ejemplo básico de uso:***

<!doctype html>

<html lang="en">

  <head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

    <title>Hello, world!</title>

  </head>

  <body>

    <h1>Hello, world!</h1>

  </body>

</html>

## **Ventajas de usar Bootstrap**

* **Desarrollo rápido y eficiente:** Permite implementar interfaces y diseños complejos en menos tiempo.
* **Diseño responsive garantizado:** Todos los componentes y layouts se adaptan automáticamente a diferentes dispositivos y resoluciones.
* **Facilidad de uso:** No requiere conocimientos avanzados de CSS o JavaScript para comenzar a crear sitios web atractivos y funcionales.

## **Desventajas de usar Bootstrap**

Aunque Bootstrap es un framework muy popular y útil para el desarrollo web, presenta algunas desventajas que conviene considerar:

* **Tamaño y rendimiento:** Bootstrap incluye muchos estilos y componentes que pueden no ser necesarios en todos los proyectos, lo que puede hacer que los archivos CSS y JavaScript sean pesados y ralenticen la carga del sitio si no se optimizan adecuadamente.
* **Dificultad para personalizar y sobrescribir estilos:** Bootstrap tiene estilos predeterminados muy marcados, y modificar o sobrescribir estos estilos para adaptarlos a un diseño específico puede ser tedioso y consumir tiempo, especialmente cuando se trabaja con diseños ya establecidos o clientes que requieren una apariencia muy particular.
* **Limitaciones en diseño flexible:** El sistema de rejilla está basado en un grid de 12 columnas con anchos, márgenes y alturas predeterminadas, lo que puede dificultar cambios muy específicos o diseños que no encajen en esta estructura.

# **Diseñando con rejilla en Bootstrap**

El sistema de rejilla (grid) de Bootstrap es una herramienta fundamental para crear diseños web responsivos y adaptativos de manera sencilla y eficiente. Este sistema permite organizar el contenido en filas y columnas, facilitando la disposición y alineación de los elementos en cualquier tamaño de pantalla.

## **Estructura básica de la rejilla**

* **Contenedor:** Todo diseño con rejilla comienza con un contenedor, que puede ser .container (ancho fijo y centrado) o .container-fluid (ocupa el 100% del ancho de la pantalla).
* **Filas:** Dentro del contenedor se crean filas usando la clase .row. Las filas agrupan horizontalmente las columnas y gestionan el espaciado y alineación.
* **Columnas:** Las columnas se definen dentro de las filas usando clases como .col, .col-6, .col-md-4, etc. Cada fila puede dividirse hasta en 12 columnas, y el contenido debe estar siempre dentro de una columna.

## **Funcionamiento y reglas del sistema de rejilla**

* El sistema está basado en 12 columnas por fila. Puedes decidir cuántas columnas ocupa cada elemento sumando hasta 12 por fila. Si el total supera 12, el contenido sobrante pasa a la siguiente fila automáticamente.
* El tamaño de las columnas se controla mediante clases CSS específicas para cada tamaño de pantalla (breakpoints), como .col-sm-6, .col-md-4, etc. Esto permite que el diseño se adapte automáticamente a dispositivos móviles, tabletas y escritorios.
* Las clases de columnas aplican sus estilos a partir del tamaño de pantalla definido y todos los tamaños superiores, a menos que se especifique una clase diferente para esos tamaños.

### ***Ejemplo de estructura básica***

<div class="container">

<div class="row">

<div class="col-4">Columna 1</div>

<div class="col-4">Columna 2</div>

<div class="col-4">Columna 3</div>

</div>

</div>

En este ejemplo, cada columna ocupa 4 de las 12 columnas disponibles en la fila, dividiendo el espacio en partes iguales.

## **Diseño responsivo**

El sistema de rejilla de Bootstrap permite crear diseños que se adaptan automáticamente a cualquier dispositivo. Utilizando los diferentes breakpoints (.col-, .col-sm-, .col-md-, .col-lg-, .col-xl-), puedes definir cómo se comportan las columnas en función del ancho de la pantalla.

## **Anidamiento de rejillas**

Para diseños más complejos, puedes anidar filas y columnas dentro de otras columnas, siguiendo la misma estructura básica. Esto te permite crear layouts avanzados y personalizados.

En resumen, diseñar con rejilla en Bootstrap consiste en estructurar el contenido dentro de contenedores, filas y columnas, aprovechando las clases y reglas del sistema para lograr diseños flexibles, ordenados y completamente adaptativos a cualquier dispositivo.

# **Tipos de rejillas en Bootstrap**

Bootstrap utiliza principalmente dos tipos de contenedores para estructurar su sistema de rejilla, que determinan cómo se comporta el ancho y la disposición del contenido en la página:

* **.container**: Es un contenedor con un ancho fijo y centrado que se adapta a diferentes tamaños de pantalla hasta un máximo predefinido. Si el ancho de la pantalla es mayor que este máximo, el contenido permanece centrado con márgenes a los lados. Es ideal para diseños con un ancho limitado y centrado en la pantalla.
* **.container-fluid**: Este contenedor ocupa el 100% del ancho disponible de la pantalla, ajustándose siempre al tamaño completo, sin importar si la pantalla es pequeña o muy grande. Es útil cuando se desea que el contenido abarque todo el ancho del dispositivo.

Dentro de estos contenedores se crean las filas (.row) y columnas (.col-\*), que forman la rejilla flexible basada en 12 columnas.

## **Sistema de columnas y breakpoints**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ***Muy pequeño*** *<576px* | ***Pequeño*** *≥576px* | ***Medio*** *≥768px* | ***Grande*** *≥992px* | ***Extra grande*** *≥1200px* |
| **Ancho máximo del contenedor** | Ninguno (automático) | 540px | 720px | 960px | 1140px |
| **Prefijo de clase** | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| **Numero de columnas** | 12 | | | | |
| **Ancho del canalón** | 30px (15px en cada lado de una columna) | | | | |
| **Encajable** | Sí | | | | |
| **Ordenación de columnas** | Sí | | | | |

Las columnas se definen con clases que indican cuántas de las 12 columnas deben ocupar, y pueden variar según el tamaño de pantalla utilizando prefijos específicos (breakpoints):

# **Manejo de filas y columnas**

El manejo de filas y columnas en Bootstrap es fundamental para crear diseños web responsivos y adaptables. Bootstrap utiliza un sistema de rejilla flexible basado en Flexbox, que facilita la disposición y alineación del contenido en diferentes dispositivos.

## **Filas (.row)**

* Las filas se utilizan para agrupar horizontalmente las columnas.
* Deben estar dentro de un contenedor (.container o .container-fluid).
* Admiten clases modificadoras para aplicar uniformemente el tamaño de columna y clases de gutter.
* Cada fila puede contener hasta 12 columnas como suma total. Si se excede, el contenido se desborda a una nueva fila automáticamente.

## **Columnas (.col-\*)**

* Las columnas se definen dentro de las filas.
* Se pueden crear columnas usando la clase .col-\*, donde \* es el tamaño de la columna (hasta 12 columnas en una fila).
* La clase .col (sin valor) permite incluir todas las columnas que se deseen.
* El tamaño de las columnas se especifica con clases CSS que Bootstrap define para cada tamaño de pantalla, por ejemplo .col-md-XX donde XX es el tamaño de la columna, que podrá tomar valores entre 1 y 12.
* Al especificar la clase del tamaño de las columnas (ejemplo: .col-md-XX) no sólo se aplica al tamaño definido, sino que para los superiores también. Para los valores inferiores cada columna ocupará el ancho total del contenedor padre.
* Se pueden modificar columnas individuales y grupos de columnas en el nivel de fila gracias a Flexbox.

## **Breakpoints**

Bootstrap utiliza breakpoints para adaptar el diseño a diferentes tamaños de pantalla. Los breakpoints más comunes son:

* sm (≥ 576px)
* md (≥ 768px)
* lg (≥ 992px)
* xl (≥ 1200px)
* xxl (≥ 1400px)

Se utilizan las clases .col-xx-\* para definir el ancho de la columna en dispositivos de diferentes tamaños, donde xx representa el breakpoint (sm, md, lg, xl, y xxl).

### ***Ejemplo***

<div class="container">

<div class="row">

<div class="col-md-4">Columna 1</div>

<div class="col-md-4">Columna 2</div>

<div class="col-md-4">Columna 3</div>

</div>

</div>

En este ejemplo, en pantallas medianas y superiores, cada columna ocupa 4 de las 12 columnas disponibles en la fila, dividiendo el espacio en partes iguales.

## **Consideraciones adicionales**

* Al crear diseños de rejilla, todo el contenido va en columnas.
* Bootstrap incluye clases predefinidas para crear diseños rápidos y responsivos.
* Si el tamaño total de las columnas de una fila excede de 12, el sobrante se colocará en la siguiente fila.

# **Tipos de contenedores**

En Bootstrap, un contenedor es un elemento estructural que se utiliza para limitar, centrar y organizar el contenido dentro de una página web. Actúa como un "marco" que define los márgenes laterales del contenido y permite que otros elementos, como filas (.row) y columnas (.col), se alineen correctamente dentro del sistema de rejilla (grid) de Bootstrap.

Los contenedores son el punto de partida para crear una maquetación responsive. Casi todos los componentes (tarjetas, formularios, encabezados, etc.) deben colocarse dentro de un contenedor para que se comporten correctamente en distintos tamaños de pantalla.

Bootstrap viene con tres contenedores diferentes:

**Contenedor fijo y responsive (.container):**

Es el contenedor más común. Tiene un ancho máximo fijo que cambia según el punto de interrupción (breakpoint) del dispositivo, y se centra horizontalmente en la pantalla. Por ejemplo, su ancho máximo es 540px en pantallas pequeñas (≥576px), 720px en medianas (≥768px), hasta 1320px en pantallas muy grandes (≥1400px). En pantallas muy pequeñas (<576px) ocupa el 100% del ancho.

<div class="container">

<!-- Contenido centrado y con ancho máximo variable -->

</div>

**Contenedor fluido (.container-fluid):**

Ocupa el 100% del ancho del viewport en todos los tamaños de pantalla, sin límites máximos de ancho. Es ideal cuando se desea que el contenido abarque todo el ancho disponible.

<div class="container-fluid">

<!-- Contenido que ocupa todo el ancho -->

</div>

**Contenedores responsivos con breakpoint específico (.container-{breakpoint}):**

Son contenedores que tienen un ancho del 100% hasta alcanzar un punto de interrupción específico, a partir del cual aplican un ancho máximo fijo como .container. Por ejemplo, .container-sm es ancho completo hasta 576px, y luego tiene un ancho máximo fijo; .container-md es ancho completo hasta 768px, y así sucesivamente.

<div class="container-sm">Ancho 100% hasta ≥576px, luego ancho fijo</div>

<div class="container-md">Ancho 100% hasta ≥768px, luego ancho fijo</div>

<div class="container-lg">Ancho 100% hasta ≥992px, luego ancho fijo</div>

<div class="container-xl">Ancho 100% hasta ≥1200px, luego ancho fijo</div>

<div class="container-xxl">Ancho 100% hasta ≥1400px, luego ancho fijo</div>

### ***Tabla de anchos máximos para .container:***

|  |  |
| --- | --- |
| **Punto de interrupción** | **Ancho máximo .container** |
| *Extra pequeño <576px* | 100% (ancho completo) |
| *Pequeño ≥576px* | 540px |
| *Mediano ≥768px* | 720px |
| *Grande ≥992px* | 960px |
| *Extra grande ≥1200px* | 1140px |
| *Extra extra grande ≥1400px* | 1320px |

# **Imágenes responsivas en Bootstrap**

Bootstrap permite que las imágenes se adapten automáticamente al tamaño del contenedor utilizando la clase .img-fluid. Esta clase aplica las propiedades:

* **max-width:** 100%
* **height:** auto

Estas reglas aseguran que las imágenes mantengan su proporción original y se escalen correctamente dentro de cualquier contenedor. Esto es especialmente útil para evitar que las imágenes se desborden o se deformen en pantallas pequeñas, garantizando una visualización adecuada en dispositivos móviles, tablets y computadoras de escritorio.

Para lograr que las imágenes se adapten correctamente a diferentes tamaños de pantalla y contenedores, Bootstrap ofrece la clase principal que es *.img-fluid.*

## **¿Qué hace .img-fluid?**

* Aplica las propiedades CSS max-width: 100%; y height: auto; a la imagen.
* Esto garantiza que la imagen escale proporcionalmente y nunca exceda el ancho del contenedor padre.
* Así, la imagen se adapta automáticamente a cualquier dispositivo, desde móviles hasta pantallas grandes, manteniendo su proporción original y evitando distorsiones.

## **Cómo usar .img-fluid**

Solo debes agregar la clase a la etiqueta <img> de la siguiente manera:

<img src="ruta/de/la/imagen.jpg" class="img-fluid" alt="Descripción de la imagen">

## **Ventajas**

* Mejora la experiencia de usuario al asegurar que las imágenes no se desborden ni se vean cortadas.
* Facilita la creación de diseños responsivos sin necesidad de escribir CSS personalizado.
* Compatible con todos los navegadores modernos.

## **Clases adicionales para imágenes en Bootstrap**

* .rounded: para bordes redondeados en la imagen.
* .img-thumbnail: para agregar un borde y sombra estilo miniatura.
* Clases de relación de aspecto (.ratio) para mantener proporciones específicas, por ejemplo:

<div class="ratio ratio-16x9">

<img src="imagen.jpg" class="img-fluid" alt="Descripción">

</div>

## **Recomendaciones**

* Siempre incluye el atributo alt para mejorar la accesibilidad.
* Coloca las imágenes dentro de contenedores adecuados (.container, .row, .col) para un mejor control del diseño.
* Optimiza las imágenes para web para mejorar tiempos de carga y rendimiento.

# **Tablas responsivas en Bootstrap**

Las tablas adaptables permiten desplazarse horizontalmente con facilidad. Para que cualquier tabla sea adaptable en todas las ventanas gráficas, encierra un elemento .tablecon .table-responsive. Las tablas adaptables utilizan overflow-y: hidden, que recorta cualquier contenido que sobrepase los bordes superior e inferior de la tabla. En particular, esto puede recortar menús desplegables y otros widgets de terceros.

## **Cómo hacer una tabla responsiva con Bootstrap**

Para lograr que una tabla sea responsiva, simplemente se debe envolver la tabla con un contenedor que tenga la clase .table-responsive. Esto hace que la tabla pueda desplazarse horizontalmente cuando el espacio no es suficiente, mostrando una barra de scroll para acceder a toda la información sin romper el diseño.

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th>Nombre</th>

<th>Apellido</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>Juan</td>

<td>Pérez</td>

<td>juan@example.com</td>

</tr>

<tr>

<td>Ana</td>

<td>García</td>

<td>ana@example.com</td>

</tr>

</tbody>

</table>

</div>

## **Clases útiles para mejorar la tabla**

* .table-striped: filas con colores alternos para mejorar la legibilidad.
* .table-hover: efecto hover al pasar el cursor sobre las filas.
* .table-bordered: agrega bordes a las celdas.
* .table-sm: tabla más compacta con menos padding.

### ***Ejemplo combinando clases:***

<div class="table-responsive">

<table class="table table-striped table-hover table-bordered table-sm">

<!-- contenido -->

</table>

</div>

## **Adaptación a diferentes tamaños de pantalla**

Bootstrap también permite usar variantes como .table-responsive-sm, .table-responsive-md, .table-responsive-lg, .table-responsive-xl y .table-responsive-xxl para que la tabla sea responsiva solo en ciertos puntos de interrupción, optimizando la experiencia según el dispositivo.

## **Ventajas de las tablas responsivas en Bootstrap**

* Evitan que la tabla se desborde o se corte en pantallas pequeñas.
* Mantienen la integridad de la información sin perder columnas.
* Mejoran la usabilidad y accesibilidad en dispositivos móviles.
* Son fáciles de implementar con pocas líneas de código.

# **Formularios responsivos con Bootstrap**

Bootstrap facilita la creación de formularios que se ajustan automáticamente a distintas resoluciones de pantalla gracias a su sistema de rejilla (grid). Utilizando clases como row y col, es posible distribuir los elementos del formulario en columnas que se reordenan dinámicamente según el tamaño de la pantalla.

De esta manera, en pantallas pequeñas los campos pueden mostrarse en una sola columna, mientras que en pantallas grandes pueden organizarse horizontalmente en múltiples columnas, mejorando la usabilidad y estética del formulario en todo tipo de dispositivos.

## **Claves para crear formularios responsivos**

* **Uso del sistema de rejilla (grid):** Organiza los campos del formulario dentro de filas *(.row)* y columnas *(.col-),* permitiendo distribuir y alinear los controles de manera flexible según el tamaño de pantalla. Por ejemplo, puedes hacer que un campo ocupe toda la fila en móviles (col-12) y solo una parte en pantallas más grandes (col-md-6).
* **Clases de formulario de Bootstrap:** Utiliza clases como *.form-control* para inputs, *selects y textareas*, que aplican estilos consistentes y responsivos automáticamente.
* **Etiquetas y controles alineados:** Bootstrap permite crear formularios horizontales y verticales con alineación adecuada usando clases como *.form-label, .form-check, y .form-floating.*
* **Validación integrada:** Bootstrap incluye clases para mostrar estados de *validación (.is-valid, .is-invalid)* y mensajes de error, mejorando la interacción y guía al usuario.
* **Componentes interactivos:** Incluye botones, casillas, radios y menús desplegables accesibles y adaptativos.

### ***Ejemplo básico de formulario responsivo***

<form>

<div class="row mb-3">

<div class="col-12 col-md-6">

<label for="nombre" class="form-label">Nombre</label>

<input type="text" class="form-control" id="nombre" placeholder="Tu nombre">

</div>

<div class="col-12 col-md-6">

<label for="email" class="form-label">Correo electrónico</label>

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

</div>

</div>

<div class="mb-3">

<label for="mensaje" class="form-label">Mensaje</label>

<textarea class="form-control" id="mensaje" rows="3"></textarea>

</div>

<button type="submit" class="btn btn-primary">Enviar</button>

</form>

## **Ventajas adicionales**

* **Adaptación automática:** Los formularios se ajustan sin necesidad de media queries personalizadas.
* **Accesibilidad:** Bootstrap fomenta buenas prácticas para etiquetas y controles accesibles.
* **Personalización sencilla:** Puedes modificar colores, tamaños y estilos mediante variables Sass o clases utilitarias.
* **Validación visual:** Feedback inmediato para el usuario con mensajes y estilos claros.

# **Conclusión:**

Bootstrap se presenta como una herramienta fundamental en el desarrollo web moderno, facilitando la creación de interfaces atractivas, funcionales y adaptables a múltiples dispositivos con sus herramientas constantemente actualizadas y adaptadas a las necesidades de los usuarios que las utilizan. A lo largo de este documento se han abordado conceptos básicos del framework, principalmente en el sistema de rejilla, contenedores, imágenes y tablas responsivas, desarrollándolas con un enfoque en sus funciones responsivas, así como sus principales características, ventajas y desventajas.

Además, para profundizar en el tema, se incluyeron ejemplos claros que muestran la sintaxis de las herramientas de Bootstrap, así como diversos casos que ilustran la variedad de estilos aplicables a sus funciones. Esto demuestra la amplia gama de opciones disponibles, permitiendo personalizar y adaptar los recursos para futuros proyectos de manera eficiente y creativa.

El propósito de esta investigación ha sido ofrecer una visión clara y accesible que facilite a futuros desarrolladores y personas interesadas en la tecnología la comprensión y aplicación de Bootstrap en sus proyectos web. De este modo, se contribuye a optimizar tanto el diseño como la experiencia de usuario en dispositivos con diferentes características, promoviendo el aprendizaje y la adopción de buenas prácticas en el desarrollo front-end.

En conclusión, Bootstrap es una herramienta fundamental tanto para desarrolladores como para usuarios, ya que sus aportaciones facilitan la comunicación de ideas mediante interfaces intuitivas y adaptables, mejorando la interacción y comprensión entre quienes crean y quienes utilizan las aplicaciones web.

# **Bibliografía:**

A, D., & A, D. (2023, January 11). *¿Qué es Bootstrap? - Una guía para principiantes*. ES Tutoriales. <https://www.hostinger.com/es/tutoriales/que-es-bootstrap>

Author, G. (2021, February 12). *Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo*. Rock Content - ES. <https://rockcontent.com/es/blog/bootstrap/>

Contributors, M. O. J. T. a. B. (n.d.-a). *Containers*. <https://getbootstrap.com/docs/5.0/layout/containers/>

Contributors, M. O. J. T. a. B. (n.d.-b). *Forms*. <https://getbootstrap.com/docs/5.0/forms/overview/>

Contributors, M. O. J. T. a. B. (n.d.-c). *Grid system*. <https://getbootstrap.com/docs/4.0/layout/grid/>

Contributors, M. O. J. T. a. B. (n.d.-d). *Images*. <https://getbootstrap.com/docs/4.0/content/images/>

Contributors, M. O. J. T. a. B. (n.d.-e). *Tables*. <https://getbootstrap.com/docs/4.0/content/tables/>

Factory, C. (2017a, January 8). *Bootstrap: entender el modelo de rejilla (II)*. Cubic Factory. <https://www.cubicfactory.com/bootstrap-entender-el-modelo-de-rejilla/>

Factory, C. (2017b, January 8). *Bootstrap: Tablas responsive con StackTable (IV)*. Cubic Factory. <https://www.cubicfactory.com/bootstrap-tablas-responsive-con-stacktable/>

*Sistema de rejilla en las páginas | Servicios Informáticos*. (n.d.). <https://ssii.ucm.es/sistema-de-rejilla-en-las-paginas>

*W3Schools.com*. (n.d.). <https://www.w3schools.com/bootstrap5/bootstrap_containers.php>