

Introducción

El presente trabajo es una guía para dar a conocer el uso, funcionamiento e implementación del framework bootstrap en un sitio web. Con el objetivo de exponer desde las definiciones y los conceptos básicos, los tipos de rejillas y el diseño que obtenemos haciendo uso de esta, manejo de filas y columnas, así como también la utilización de los contenedores, y no olvidando los elementos responsivos como lo son las imágenes, tablas y formularios. Cada uno de estos elementos nos ayudaran en el camino de nuestro aprendizaje, dándonos así una herramienta más para trabajar nuestras páginas y sitios web.

Marco Teórico

BOOTSTRAP



Definición y conceptos básicos de Bootstrap.

Aquí tienes algunos conceptos básicos relacionados con Bootstrap:

Grid System (Sistema de Cuadrícula): Bootstrap utiliza un sistema de cuadrícula de 12 columnas para organizar el diseño de la página. Esto facilita la creación de diseños flexibles y responsivos, que se adaptan a diferentes tamaños de pantalla.

Componentes:

Tipografía: Bootstrap incluye estilos prediseñados para encabezados, párrafos, listas, etc.

Botones: Estilos predefinidos para botones que se pueden personalizar fácilmente.

Formularios: Estilos y componentes predefinidos para la creación de formularios.

Navegación: Barras de navegación responsivas y otros elementos de navegación.

Carruseles, modalidades y más: Componentes adicionales que facilitan la implementación de características comunes.

JavaScript Incorporado: Bootstrap incluye bibliotecas de JavaScript para agregar funcionalidades interactivas a los elementos de la interfaz de usuario. Esto incluye cosas como modales, carruseles y plegado de navegación (navbar).

Clases de Utilidad: Bootstrap proporciona un conjunto de clases de utilidad que pueden aplicarse directamente en el marcado HTML para realizar acciones específicas, como ocultar o mostrar elementos en diferentes tamaños de pantalla.

Diseño Responsivo: Uno de los aspectos más destacados de Bootstrap es su enfoque en el diseño responsivo, lo que significa que los sitios web y las aplicaciones creadas con Bootstrap se adaptan y funcionan bien en una variedad de dispositivos y tamaños de pantalla.

Tema Personalizable: Bootstrap permite a los desarrolladores personalizar fácilmente el aspecto y la sensación de sus proyectos utilizando variables de Sass o Less, lo que facilita la adaptación del diseño a las necesidades específicas del proyecto.

Compatibilidad con Navegadores: Bootstrap está diseñado para ser compatible con los navegadores más populares, lo que garantiza una experiencia coherente para los usuarios independientemente del navegador que utilicen.

Diseñando con rejilla.

Utiliza nuestra potente cuadrícula flexbox mobile-first para crear diseños de todas las formas y tamaños gracias a un sistema de doce columnas, seis niveles responsive (breakpoints) predeterminados, variables Sass y mixins, y docenas de clases predefinidas.

Ejemplo

El sistema de cuadrícula (grid) de Bootstrap utiliza una serie de contenedores, filas y columnas para diseñar y alinear el contenido. Está construido con [flexbox](#) y es completamente responsive. A continuación se muestra un ejemplo y una explicación detallada de cómo se compone el sistema de cuadrícula.

Columna	Columna	Columna
---------	---------	---------

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

El ejemplo anterior crea tres columnas de igual ancho en todos los dispositivos y viewports utilizando nuestras clases de cuadrícula predefinidas. Esas columnas están centradas en la página con el `.container` principal.

¿Cómo funciona el sistema de cuadrícula (Grid)?

Desglosándolo, así es como se compone el sistema de cuadrícula:

Nuestra cuadrícula admite [seis breakpoints responsive](#). Los breakpoints se basan en media queries min-width, lo que significa que afectan ese breakpoint y todos los anteriores (por ejemplo, `.col-sm-4` se aplica a sm, md, lg, xl y xxl). Esto significa que puedes controlar el tamaño y el comportamiento del contenedor y la columna en cada breakpoint.

Los contenedores centran y rellenan horizontalmente su contenido. Usa `.container` para un ancho de píxel responsive, `.container-fluid` para `width: 100%` en todos los viewports y dispositivos, o un contenedor responsive (p. ej., `.container-md`) para una combinación entre fluido y anchos en píxeles.

Las filas son envoltorios para las columnas. Cada columna tiene un padding horizontal (llamado gutter) para controlar el espacio entre ellas. Este padding luego se contrarresta en las filas con márgenes negativos para garantizar que el contenido de tus columnas esté alineado visualmente en el lado izquierdo. Las filas también admiten clases modificadoras para [aplicar uniformemente el tamaño de columna](#) y [clases de gutter](#) para cambiar el espaciado de tu contenido.

Las columnas son increíblemente flexibles. Hay 12 columnas de plantilla disponibles por fila, lo que te permite crear diferentes combinaciones de elementos que abarcan cualquier número de columnas. Las clases de columna indican el número de columnas de plantilla que abarcan (por ejemplo, `col-4` abarca cuatro). widths se establecen en porcentajes para que siempre tengas el mismo tamaño relativo.

Los gutters también son responsive y personalizables. [Las clases Gutter están disponibles](#) en todos los breakpoints, con los mismos tamaños que nuestro [espaciado de margin y padding](#). Cambia los gutters horizontales con las clases `.gx-*`, los gutters verticales con `.gy-*` o todos los gutters con las clases `.g-*`. `.g-0` también está disponible para eliminar gutters.

Las variables Sass, los mapas y los mixins alimentan la cuadrícula. Si no deseas usar las clases de cuadrícula predefinidas en Bootstrap, puedes usar nuestra [código fuente Sass de cuadrículas](#) para crear el tuyo propio con más marcado semántico. También incluimos algunas propiedades personalizadas de CSS para consumir estas variables de Sass para una mayor flexibilidad.

Ten en cuenta las limitaciones y los [errores relacionados con flexbox](#), como la [incapacidad de usar algunos elementos HTML como contenedores flexibles](#).

Opciones de cuadrícula

El sistema de cuadrícula de Bootstrap puede adaptarse a los seis breakpoints predeterminados y a cualquier breakpoint que personalices. Los seis niveles de cuadrícula predeterminados son los siguientes:

Extra small (xs)

Small (sm)

Medium (md)

Large (lg)

Extra large (xl)

Extra extra large (xxl)

Como se indicó anteriormente, cada uno de estos breakpoints tienen su propio contenedor, prefijo de clase único y modificadores. Así es como cambia la cuadrícula en estos breakpoints:

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Contenedor <code>max-width</code>	None (auto)	540px	720px	960px	1140px	1320px
Prefijo de clase	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>
# de columnas	12					
Ancho de gutter	1.5rem (.75rem a izquierda y derecha)					
Gutters personalizados	Si					
Anidable	Si					
Ordenamiento de columnas	Si					

Tipos de rejillas.

El sistema de rejilla de Bootstrap se basa en la creación o disposición del contenido de nuestra web dentro de rejillas flexibles, las cuales se escalarán al tamaño y posición adecuada de forma automática dependiendo del tamaño de la pantalla en la que se rendericen.

Elemento contenedor

El sistema de rejilla tiene que ser utilizado dentro de uno de los dos elementos contenedores que provee Bootstrap: `container` ó `container-fluid`. Es importante tener en cuenta que estos elementos se utilizan como raíz de la rejilla y no se podrán anidar unos dentro de otros.

Si lo que queremos es que el contenido de nuestra web aparezca centrado y con un ancho fijo entonces podemos utilizar la etiqueta `.container`, de la forma:

```
<div class="container">
  ...
</div>
```

Por el contrario, si queremos que el contenido de nuestra web pueda ocupar todo el ancho disponible (hay que tener en mente todos los tamaños de pantalla, incluso las muy grandes), podemos usar la etiqueta `.container-fluid`:

```
<div class="container-fluid">
  ...
</div>
```

Funcionamiento del sistema de rejilla

El sistema de rejilla está pensado para ayudarnos en la disposición de los contenidos de nuestra web y su adaptación a los diferentes tamaños de pantalla de forma automática. Para ello tenemos que poner el contenido dentro de celdas o columnas que irán dentro de filas.

Cada fila se puede dividir hasta en 12 columnas, pero seremos nosotros los que definiremos el número de columnas deseado para cada tamaño de pantalla.

A continuación se detalla el funcionamiento de este sistema:

Las columnas irán agrupadas dentro de filas (.row).

Las filas (.row) se deben colocar dentro de una etiqueta contenedora: .container (para ancho fijo) o .container-fluid (para poder ocupar todo el ancho), esto permitirá alinear las celdas y asignarles el espaciado correcto.

El contenido se debe disponer dentro de columnas o celdas, las cuales deben de ser el único hijo posible de las filas (.row), las cuales, a su vez, serán el único hijo posible del contenedor (.container o .container-fluid).

Al seguir este orden el sistema de rejilla funcionará correctamente, creando el espaciado interior y los márgenes apropiados dependiendo de las dimensiones de la pantalla.

Cada fila se puede dividir hasta un máximo de 12 columnas, pero somos nosotros los que tendremos que definir el número de columnas en el que queremos dividir cada fila y su ancho para cada tamaño de pantalla. Por ejemplo: 3 columnas de igual ancho.

Si el tamaño total de las columnas de una fila excede de 12 el tamaño sobrante se colocará en la siguiente fila.

El tamaño de las columnas se especificará 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.

En la siguiente tabla se muestra un resumen del sistema de rejilla de Bootstrap, su comportamiento según el tamaño del dispositivo y las clases CSS que nos permiten controlarlo:

Pantalla	Dimensiones	Prefijo de la clase	Ancho del contenido
Tamaño extra pequeño	< 576 px	.col-	Ninguno (automático)
Tamaño pequeño	≥ 576 px	.col-sm-	540px
Tamaño medio	≥ 768 px	.col-md-	720px
Tamaño grande	≥ 992 px	.col-lg-	960px
Tamaño extra grande	≥ 1200 px	.col-xl-	1140px

Es importante destacar que al definir estas clases no solo se aplican para ese tamaño de pantalla sino para los superiores también. Por ejemplo, al indicar el tamaño de las columnas con las clases para tablets (.col-sm-), también se aplicará para los tamaños de pantalla medianos y grandes (si no hubieran otras clases para estos tamaños que los sobrescribieran). Es decir, nos tenemos que fijar que en la tabla anterior el tamaño se indica con el símbolo de mayor o igual (≥) (o de menor para el caso de xs) a un tamaño dado, y por lo tanto se aplicará esa disposición a partir de ese tamaño, a no ser que se indique otra cosa.

Bootstrap está diseñado pensando en los dispositivos móviles primero (o como ellos indican: siguiendo la estrategia mobile first). Por lo tanto todos los tamaños y dimensiones están

pensadas para los dispositivos móviles, y para tamaños más grandes lo que hacen es adaptar o escalar estos tamaños.

Manejo de filas y columnas.

Este sistema, creado con [flexbox](#), consiste en dividir la pantalla en filas, donde cada fila puede dividirse en 12 columnas o espacios (también permite diseños de más columnas, pero como base se tiene en cuenta una división estándar de 12 columnas). De esta forma podemos, por ejemplo, crear un diseño con dos columnas, una que ocupe 5 espacios y otra que ocupe 7 espacios, tal y como se muestra a continuación:

```
1. <div class="container">
2.   <div class="row">
3.     <div class="col-5">Contenedor de 5 columnas</div>
4.     <div class="col-7">Contenedor de 7 columna</div>
5.   </div>
6. </div>
```

Los siguientes puntos son importante:

Las columnas se basan en la arquitectura flexbox de la cuadrícula. Flexbox significa que tenemos opciones para cambiar columnas individuales y [modificar grupos de columnas en el nivel de fila](#). Tú eliges cómo crecen, se reducen o cambian las columnas.

Al crear diseños de cuadrícula, todo el contenido va en columnas. La jerarquía de la cuadrícula de Bootstrap va desde [container](#) de fila a columna hasta tu contenido. En raras ocasiones, puedes combinar contenido y columna, pero ten en cuenta que puede haber consecuencias no deseadas.

Bootstrap incluye clases predefinidas para crear diseños rápidos y responsive. Con [seis breakpoints](#) y una docena de columnas en cada nivel de cuadrícula, tenemos docenas de clases ya construidas para que puedas crear tus diseños deseados. Esto se puede desactivar a través de Sass si lo deseas.

Debemos pensar para armar nuestro esquema de la página que tenemos la posibilidad de definir filas y en cada fila definir de 1 hasta 12 columnas. Cada columna con un ancho relativo a ese número 12.

Luego Bootstrap se encarga de colapsar las columnas cuando se accede al sitio desde un dispositivo con una capacidad limitada en cuanto al ancho en píxeles (esto permite una experiencia más placentera al visitante que accede a nuestra página desde un celular con capacidades limitadas)

Para ver y entender como creamos las columnas en cada fila y como colapsan según el ancho del dispositivo podemos analizar el siguiente código:

```
<!doctype html>
<html>
<head>
  <title>Prueba de Bootstrap 5</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
wEmelV1mKuiNpC+IOBjl7aAzPcEZeedi5yW5f2yOq55WWLwNGMvxx4Um1vskeMj0"
crossorigin="anonymous">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col-xxl-4" style="background-color:#aaa">
```

```
<h1>Columna 1</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
<div class="col-xxl-4" style="background-color:#bbb">
```

```
<h1>Columna 2</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
<div class="col-xxl-4" style="background-color:#ccc">
```

```
<h1>Columna 3</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
</div>
```

```
<hr>
```

```
<div class="row">
```

```
<div class="col-xl-4" style="background-color:#aaa">
```

```
<h1>Columna 1</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
<div class="col-xl-4" style="background-color:#bbb">
```

```
<h1>Columna 2</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
<div class="col-xl-4" style="background-color:#ccc">
```

```
<h1>Columna 3</h1>
```

```
<p>Esto es una prueba de bootstrap.</p>
```

```
</div>
```

```
</div>
```

```
<hr>
```

```
<div class="row">
```

```
<div class="col-lg-4" style="background-color:#aaa">
```

```
<h1>Columna 1</h1>
```



```
<p>Esto es una prueba de bootstrap.</p>
</div>
<div class="col-lg-4" style="background-color:#bbb">
  <h1>Columna 2</h1>
  <p>Esto es una prueba de bootstrap.</p>
</div>
<div class="col-lg-4" style="background-color:#ccc">
  <h1>Columna 3</h1>
  <p>Esto es una prueba de bootstrap.</p>
</div>
</div>

<hr>

<div class="row">
  <div class="col-md-4" style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-md-4" style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-md-4" style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
</div>

<hr>

<div class="row">
  <div class="col-sm-4" style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-sm-4" style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-sm-4" style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
</div>
```

```

</div>

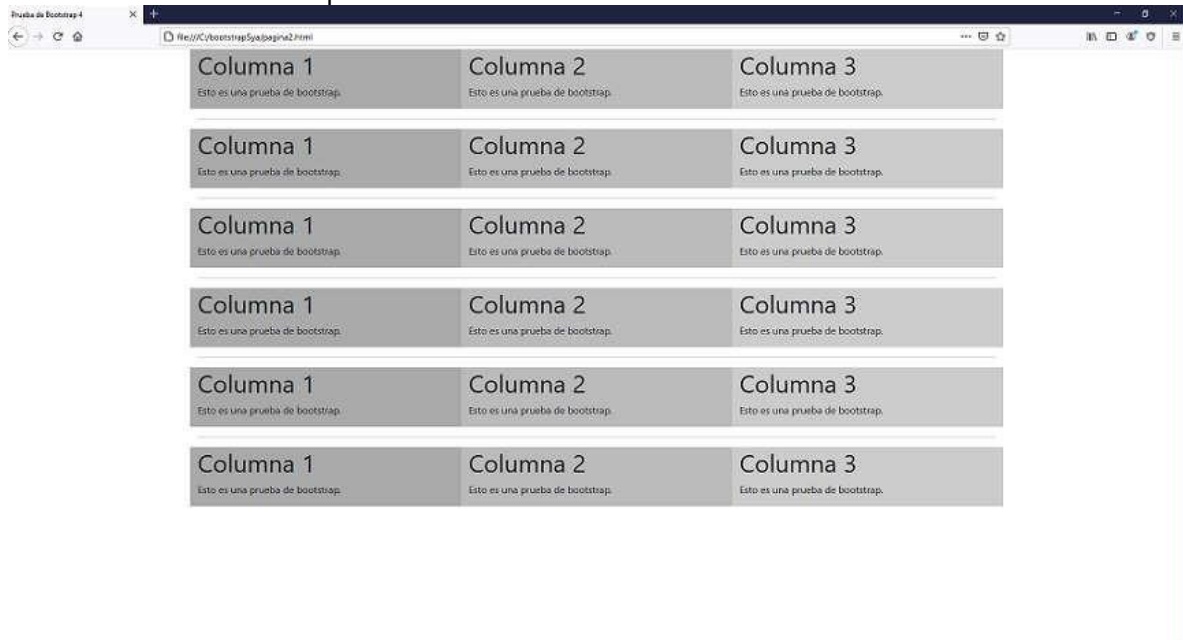
<hr>

<div class="row">
  <div class="col-4" style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-4" style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-4" style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
</div>

</div>
</body>
</html>

```

Si abrimos esta página en un monitor con un ancho superior a 1400 px podemos ver que se muestran tres columnas por fila:



Debemos utilizar la clase "row" para indicar el comienzo de una fila. Dentro de la fila dispondremos tantos div como columnas tenga dicha fila. Para indicar cada columna debemos utilizar la siguiente sintaxis para cada columna:

```
col-xxl-*
```

Donde aparece el asterisco lo remplazamos por un valor entre 1 y 12 (en nuestro ejemplo hemos dispuesto un 4, esto quiere decir que quedan 8 unidades de columna para repartir entre las otras columnas)

En nuestro ejemplo cada una de las columnas tiene un valor 4 (siempre la suma de dichos valores de una fila debe sumar 12 o quedarán columnas vacías en dicho caso) Como asignamos cuatro a cada columna las tres columnas tienen el mismo ancho:

```
<div class="row">
  <div class="col-xxl-4" style="background-color:#aaa">
    <h1>Columna 1</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-xxl-4" style="background-color:#bbb">
    <h1>Columna 2</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
  <div class="col-xxl-4" style="background-color:#ccc">
    <h1>Columna 3</h1>
    <p>Esto es una prueba de bootstrap.</p>
  </div>
</div>
```

Ahora depende del tipo de página que uno tiene que implementar el definir en que momento queremos que nuestra estructura de página colapse sus columnas. Si no queremos que colapse luego empleamos col-* para definir las distintas columnas.

Tipos de contenedores en Bootstrap.

Los contenedores son el elemento de diseño más básico en Bootstrap y son requeridos cuando se usa un sistema de cuadrícula predeterminado. Los contenedores se usan para contener rellenar y centrar el contenido dentro de ellos. Si bien los contenedores pueden anidarse, la mayoría de los diseños no requieren de un contenedor anidado.

Bootstrap viene con tres contenedores diferentes:

Container, que establece un máx-width en cada breakpoint responsive.

Container fluid, que es width :100% en todos los breakpoints.

Container breakpoint que es width:100% hasta el breakpoint especificado.

Contenedor predeterminado:

El contenedor predeterminado es un contenedor sensible de ancho fijo, lo que significa que si max-width cambia en cada breakpoint.

```
<div class="container">
  <!-- Contenido aquí -->
</div>.
```

Contenedor responsive.

Permiten especificar una clase que tiene un 100% de ancho hasta que alcanzan el breakpoint especificado, después de lo cual se aplica un max-width para cada uno de los breakpoints más altos.

Copy

```
<div class="container-sm">100% de ancho hasta el small breakpoint</div>
<div class="container-md">100% de ancho hasta el medium breakpoint</div>
<div class="container-lg">100% de ancho hasta el large breakpoint</div>
<div class="container-xl">100% de ancho hasta el extra large breakpoint</div>
<div class="container-xxl">100% de ancho hasta el extra extra large breakpoint</div>
```

Contenedor de fluidos

Se usa para un contenedor ancho completo que abarca todo el ancho del viewport.

```
<div class="container-fluid">
...
</div>
```

Sass

Bootstrap puede generar una serie de clases de contenedores predefinidas para ayudar a crear los diseños que se requieran. Se puede personalizar estas clases de contenedor predefinidas modificando el mapa de Sass.

Copy

```
$container-max-widths: (
  sm: 540px,
  md: 720px,
  lg: 960px,
  xl: 1140px,
  xxl: 1320px
);
```

además, se pueden personalizar el sass, también se pueden crear nuestros propios contenedores con nuestro mixin Sass

```
// Source mixin
@mixin make-container($padding-x: $container-padding-x) {
  width: 100%;
  padding-right: $padding-x;
  padding-left: $padding-x;
  margin-right: auto;
```

```
margin-left: auto;
}

// Uso
.custom-container {
  @include make-container();
}
```

Imágenes responsivas.

Una imagen responsiva es una imagen que se adapta a las características de los distintos dispositivos. Cuando se hacen bien, las imágenes responsivas pueden mejorar el rendimiento y la experiencia del usuario de un sitio. Se pueden abordar las imágenes responsivas desde dos ángulos: sirviendo la misma imagen con diferentes tamaños o sirviendo diferentes imágenes según las características de la pantalla.

Cuando los ingenieros de software crearon la web, no tuvieron en cuenta cómo manejarían los navegadores las imágenes responsivas. Al fin y al cabo, los usuarios sólo accedían a la web desde ordenadores de sobremesa o portátiles. Sin embargo, esa no es la situación que se da hoy en día.

Según Statista, más del 90% de la población mundial de Internet se conecta a través de su teléfono móvil. La mayoría de las páginas web de Internet contienen imágenes y éstas son una de las métricas utilizadas para medir el rendimiento de la web. Para mejorar el rendimiento, hay que optimizar las imágenes haciéndolas responsivas.

Las imágenes en Bootstrap se hacen responsive con `.img-fluid`. Esto aplica `max-width: 100%;` y `height: auto;` a la imagen para que se escale con el ancho de su elemento padre

Miniaturas de imágenes

Además de nuestras [utilidades border-radius](#), puedes usar `.img-thumbnail` para darle a una imagen una apariencia de borde redondeado de 1px.

Alinear imágenes

Alinea las imágenes con las [clases flotantes auxiliares](#) o [clases de alineación de texto](#). Las imágenes a nivel de block se pueden centrar usando [la clase de utilidad de margen .mx-auto](#).

Picture

Si estás utilizando el elemento `<picture>` para especificar múltiples elementos `<source>` para un `` específico, asegúrate de agregar las clases `.img-*` a `` y no a la etiqueta `<picture>`.

Tablas responsivas.

Las tablas responsivas en Bootstrap se adaptan al tamaño de la pantalla y mejoran la experiencia del usuario en dispositivos móviles. En Bootstrap, puedes usar la clase `.table-responsive` para hacer que las tablas se desplacen horizontalmente en pantallas pequeñas y permitir que los usuarios vean todo el contenido sin que se recorte o se vuelva ilegible. Esto es especialmente útil cuando las tablas tienen muchas columnas, Esta clase permite que

la tabla tenga un comportamiento de desplazamiento horizontal en pantallas más pequeñas.

Bootstrap utiliza reglas de CSS para modificar el comportamiento estándar de las tablas en dispositivos móviles. Cuando se agrega la clase `.table-responsive`, se aplica un desplazamiento horizontal para asegurar que la tabla sea completamente visible y se pueda deslizar hacia la izquierda o la derecha en pantallas estrechas, lo que facilita la visualización de todas las columnas.

Este enfoque ayuda a mantener la estructura y legibilidad de la tabla en diferentes tamaños de pantalla, asegurando que el usuario pueda interactuar y ver fácilmente todo el contenido, incluso en dispositivos móviles.

Formularios responsivos.

Bootstrap ofrece un sistema de cuadrícula responsiva y clases predefinidas que te permiten crear formularios que se adaptan automáticamente a diferentes tamaños de pantalla y dispositivos. Esto facilita su visualización y funcionamiento en teléfonos móviles, tabletas y computadoras de escritorio. Acelera el proceso de diseño y desarrollo de formularios porque no tienes que empezar desde cero. Puedes aprovechar las clases y componentes predefinidos de Bootstrap para que tus formularios sean atractivos y funcionales.

Incluye componentes y elementos interactivos, como etiquetas de formulario, grupos de botones, casillas de verificación, botones de alternancia, entre otros. Estos te permiten añadir, de manera sencilla, funciones adicionales; por ejemplo, la selección múltiple, casillas de verificación, etc.

Conclusiones

Como equipo trabajo damos por confirmado que es muy importante saber y conocer cómo se utilizan y se implementan los framework para el diseño web, en este caso investigamos sobre bootstrap un poderoso aliado para crear páginas web de una manera eficiente y rápida. Cada uno de sus elementos son de gran beneficio para el desarrollador frontend, iniciando desde los conceptos básicos, pasando por manejo de rejillas, utilización de filas y columnas, tipos de contenedores, y la parte responsiva, desde las imágenes, tablas y formularios. Definitivamente en bootstrap podemos encontrar una herramienta de desarrollo para realizar trabajos de calidad.

Bibliografía

<https://bootstrap-esdocu.com/docs/5.1/layout/containers/>

<https://www.eniun.com/sistema-columnas-contenedores-rejillas-bootstrap/>

<https://bootstrap-esdocu.com/docs/5.1/layout/columns/>

<https://www.tutorialesprogramacionya.com/bootstrap5ya/detalleconcepto.php?punto=3&codigo=3&inicio=0>

<https://ajgallego.gitbook.io/bootstrap-4/sistema-de-rejilla>

<https://www.tutorialesprogramacionya.com/cssya/bootstrapya/detalleconcepto.php?punto=2&codigo=117&inicio=0>

[https://blog.hubspot.es/website/formulario-](https://blog.hubspot.es/website/formulario-bootstrap#:~:text=Dise%C3%B1o%20responsivo,tama%C3%B1os%20de%20pantalla%20y%20dispositivos.)

[bootstrap#:~:text=Dise%C3%B1o%20responsivo,tama%C3%B1os%20de%20pantalla%20y%20dispositivos.](https://blog.hubspot.es/website/formulario-bootstrap#:~:text=Dise%C3%B1o%20responsivo,tama%C3%B1os%20de%20pantalla%20y%20dispositivos.)