

Guía de estudio - Introducción a Bootstrap



¡Hola! Te damos la bienvenida a esta nueva guía de ejercicios.

¿En qué consiste esta guía?

Hasta ahora hemos aprendido cómo crear páginas web utilizando HTML y CSS; estos conocimientos son la base que nos permitirá construir sitios cada vez más profesionales.

Para todo desarrollador es importante optimizar el tiempo en aquellos aspectos que resultan recurrentes y dedicar sus esfuerzos en dar valor a las características particulares de cada proyecto. Aquí es donde aparecen los frameworks, que vienen a apoyarnos a resolver estas tareas repetitivas.

En esta guía conoceremos qué son los frameworks CSS, cuáles son las ventajas de su utilización y tendremos un primer acercamiento a Bootstrap y uno de sus usos más comunes, la distribución de elementos basados en su grilla.

¡Vamos con todo!



Tabla de contenidos

| | |
|--|----|
| Conociendo los frameworks CSS | 3 |
| ¿Qué es Bootstrap? | 3 |
| Integrando el CDN de Bootstrap | 6 |
| Integrando Bootstrap mediante descarga | 6 |
| Modificación de textos con Bootstrap | 7 |
| Alineamiento del texto | 9 |
| Tamaño del texto | 9 |
| Colores de fondo | 10 |
| Márgenes y paddings | 11 |
| Actividad 1 | 12 |
| Bootstrap es un framework responsive | 13 |
| ¿Qué es un diseño responsive? | 13 |
| Contenedores de Bootstrap | 14 |
| Tipos de contenedores | 15 |
| Actividad 2: Jugando con containers | 16 |
| Actividad 3 | 16 |
| Sistema de Grillas de Bootstrap | 17 |
| Actividad 4 | 20 |
| Controlando el ancho de las columnas | 20 |
| Gutters | 22 |
| Actividad 5 | 23 |
| Grillas responsivas | 24 |
| Actividad 6 | 27 |
| Resumen | 28 |



¡Comencemos!

Conociendo los frameworks CSS

Comencemos por aprender qué es un **framework**:



"Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar." (Coco Solution. Retrieved 2021, from <https://cocosolution.com/que-es-un-framework/>, 2020)

En otras palabras, un framework CSS provee una estructura básica que nos ayudará a resolver problemas típicos de estilo dentro del desarrollo web.

Dentro de los Frameworks CSS, existe una gran cantidad de opciones para elegir como:

- Bootstrap
- Foundation
- Bulma
- Semantic UI
- UI kit
- Tailwind

La mejor opción para elegir dependerá de lo que queramos hacer y lo que ya conozcamos preliminarmente. En nuestro caso, usaremos uno de los framework CSS más completos para crear páginas web responsivas llamado Bootstrap, que además hoy en día es el más popular.

¿Qué es Bootstrap?

Bootstrap es un framework CSS, desarrollado y liberado por Twitter, que sirve para maquetar páginas web de manera rápida y sencilla. Es de código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.

Las características que hacen de Bootstrap el framework CSS más popular de la web son:

- **Fácil de integrar:** para integrar bootstrap a nuestros proyectos necesitaremos ingresar a [Get Bootstrap](#), copiar el CDN, pegarlo dentro del proyecto y ¡listo!
- **Sistema de grillas:** Bootstrap también incluye un sistema de grillas muy completo con el cual podremos crear layout de manera muy fácil.

- **Estilos en la mayoría de los elementos HTML:** los elementos HTML, como listas, botones, formularios, entre otros, vienen estilizados y mejorados con clases. De manera que nosotros no debemos escribir mucho CSS para estilizar los elementos.
- **Una gran lista de componentes:** contiene una gran cantidad de componentes que podremos usar en nuestra página web.
- **Gran documentación:** dentro de la página de [Bootstrap](#) podremos encontrar una excelente documentación, donde nos enseñan con ejemplos a utilizar la herramienta.

A continuación, te presentamos algunos ejemplos de sitios que utilizan Bootstrap:

1. Agence Lincorn Publishing:

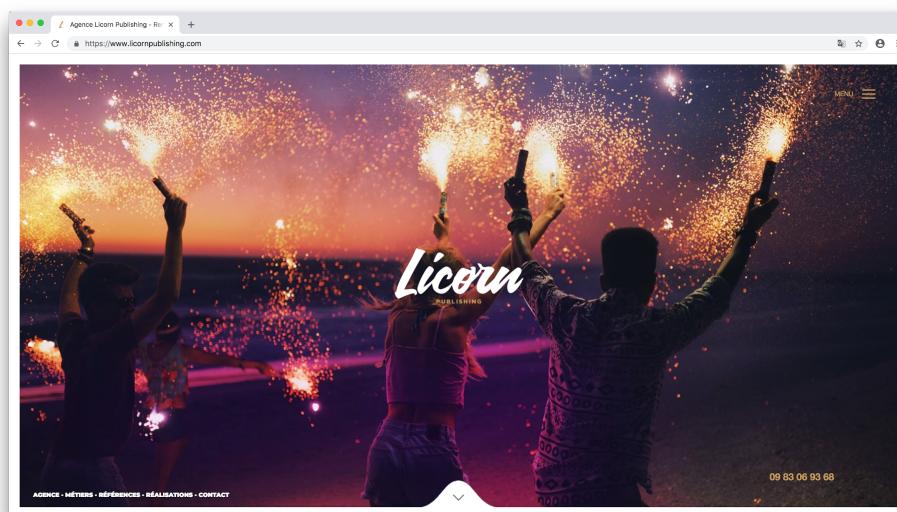


Imagen 1. Sitio desarrollado con Bootstrap - Licorn Publishing.

Fuente: licornpublishing.com.

2. Neat:

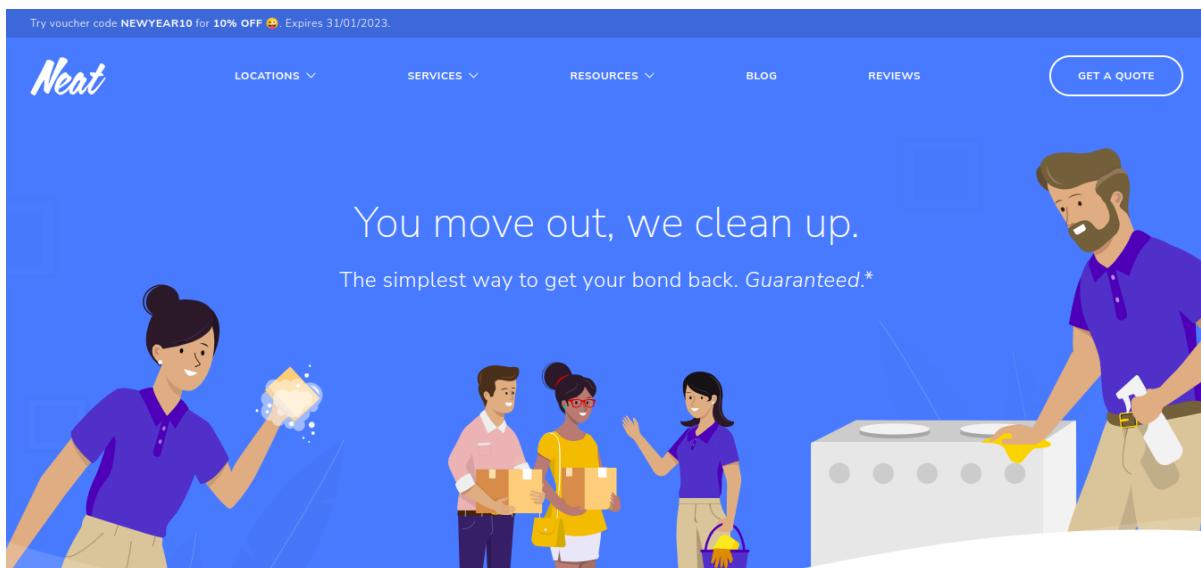


Imagen 2. Sitio desarrollado con Bootstrap - Neat.

Fuente: thisisneat.com.

3. Kingdom Rush:

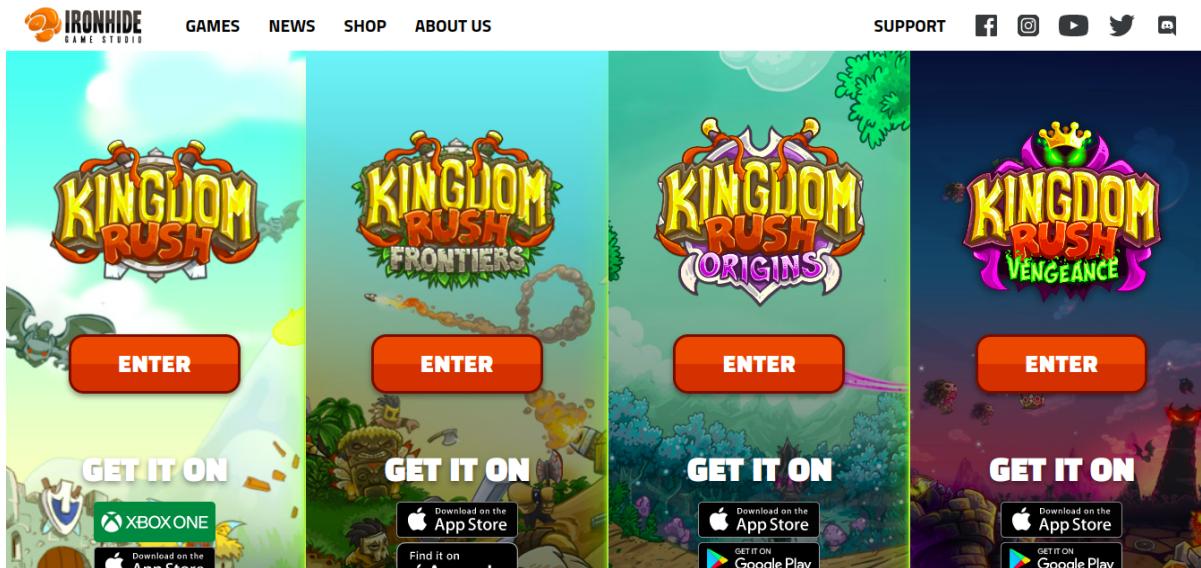


Imagen 3. Sitio desarrollado con Bootstrap - Kingdom Rush.

Fuente: kingdomrush.com.



¡Importante! Bootstrap ha tenido varios cambios de versiones recientemente, de modo que es importante conocer sus últimas novedades visitando el [blog](#). En este ejercicio estamos ocupando la versión 5.1.

Integrando el CDN de Bootstrap

Para integrar Bootstrap a nuestro proyecto, debemos incluir algunos recursos, como el CSS, que aparecen en la documentación del sitio, para esto iremos a la página oficial: <https://getbootstrap.com/> Aquí, seleccionaremos la opción download, y buscaremos el código que aparece bajo CDN vía JSDelivr.

Otra opción es partir con una plantilla inicial, eso lo podemos hacer utilizando la que aparece en <https://getbootstrap.com/> > seleccionando la opción Getting started > y luego, buscar starter template, para después copiar y pegarlo en un archivo .html

Integrando Bootstrap mediante descarga

Otra forma de integrar Bootstrap a nuestro proyecto, es descargando los archivos y agregándolos a nuestras carpetas. Para hacer esto, debemos ir a la opción “[download](#)” dentro del menú derecho de la documentación y descargar los archivos bajo la opción de “compiled CSS and JS”:

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.1.1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

[Download](#)

Imagen 4. Screenshot página descarga de archivos de bootstrap.

Fuente: [Getbootstrap.com](https://getbootstrap.com/).

Una vez que hemos descargado los archivos, vemos que contiene dos carpetas: “js” y “CSS”. Copiaremos ambas y la llevaremos a la carpeta assets del proyecto en el que estemos trabajando:

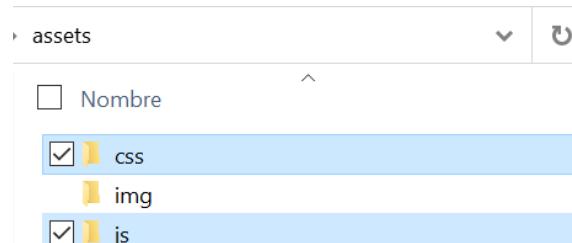


Imagen 5. Descarga de archivos de bootstrap.

Fuente: Desafío Latam.

La forma de integrarlo al proyecto es referenciando estos archivos a la ruta local de nuestro proyecto.

- bootstrap.min.css
- bootstrap.bundle.min.js

Ejemplo:

```
<head>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">
</head>
```

```
<body>
  <!-- Bootstrap JS -->
  <script src="assets/js/bootstrap.bundle.min.js"></script>
</body>
```

Modificación de textos con Bootstrap

Para definir los estilos del texto: color, tamaño y tipo, utilizábamos código en CSS. Para aprovechar Bootstrap al máximo, primero debemos buscar si lo que queremos hacer existe entre lo que ofrece, así que iremos a la [documentación](#) a buscar lo que necesitemos.

La documentación para las fuentes, imagen, tamaño y posición, las encontraremos en el apartado [utilidades](#) de Bootstrap.



Analicemos en detalle algunas de las opciones de estilo que tenemos disponibles, comenzando por el color de la letra.

Si buscamos "colors" en el buscador, nos aparecerán los [colores de Bootstrap](#):

The screenshot shows a list of Bootstrap text color classes with their corresponding visual swatches. The classes listed are: .text-primary (blue), .text-secondary (light blue), .text-success (green), .text-danger (red), .text-warning (yellow), .text-info (teal), .text-light (light gray), .text-dark (dark gray), .text-body (black), .text-muted (gray), .text-white (white), .text-black-50 (light gray), and .text-white-50 (white). The .text-light, .text-white, and .text-white-50 classes have their respective swatches highlighted with a dark gray background.

```
<p class="text-primary">.text-primary</p>
<p class="text-secondary">.text-secondary</p>
<p class="text-success">.text-success</p>
<p class="text-danger">.text-danger</p>
<p class="text-warning">.text-warning</p>
<p class="text-info">.text-info</p>
<p class="text-light bg-dark">.text-light</p>
<p class="text-dark">.text-dark</p>
<p class="text-body">.text-body</p>
<p class="text-muted">.text-muted</p>
<p class="text-white bg-dark">.text-white</p>
<p class="text-black-50">.text-black-50</p>
<p class="text-white-50 bg-dark">.text-white-50</p>
```

Imagen 6. Colores.

Fuente: Getbootstrap.com.

Ahí podemos ver que para asignarle un color claro, basta con utilizar la clase `text-light`. Por lo tanto, si queremos agregar un texto claro a toda la sección del `<header>`, lo haremos de la siguiente manera:

```
<header class="text-light">
  <!-- Contenido HTML -->
</header>
```

Alineamiento del texto

Para el alineamiento del texto haremos lo mismo, buscaremos en Bootstrap "[text-alignment](#)", que nos dará los distintos tipos de alineación. En estos casos, podremos elegir el comportamiento que tenga frente a distintos viewport:

Start aligned text on all viewport sizes.

Center aligned text on all viewport sizes.

End aligned text on all viewport sizes.

Start aligned text on viewports sized SM (small) or wider.

Start aligned text on viewports sized MD (medium) or wider.

Start aligned text on viewports sized LG (large) or wider.

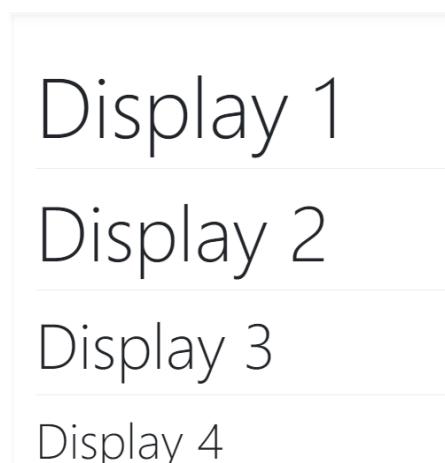
Start aligned text on viewports sized XL (extra-large) or wider.

Imagen 7. Alineamiento de texto.

Fuente: Getbootstrap.com.

Tamaño del texto

Bootstrap permite una mayor versatilidad en el uso del tamaño de los headings. Si bien podemos utilizar las etiquetas `<h1>..<h6>` existen formas de aumentar el tamaño más allá de `<h1>`. Para esto iremos a la documentación, en "Content", luego "Typography" y buscamos la personalización de los headings.



Display 1
Display 2
Display 3
Display 4

Imagen 8. Tamaño de texto.

Fuente: Getbootstrap.com.

Colores de fondo

Dentro de las clases utilitarias de bootstrap también podemos encontrar colores de fondo incluidos, estos son muy útiles para botones o incluso para la barra de navegación, pero los podemos utilizar donde estimemos conveniente.

Background color

Similar to the contextual text color classes, set the background of an element to any contextual class.

Background utilities **do not set color**, so in some cases you'll want to use `.text-* color utilities`.

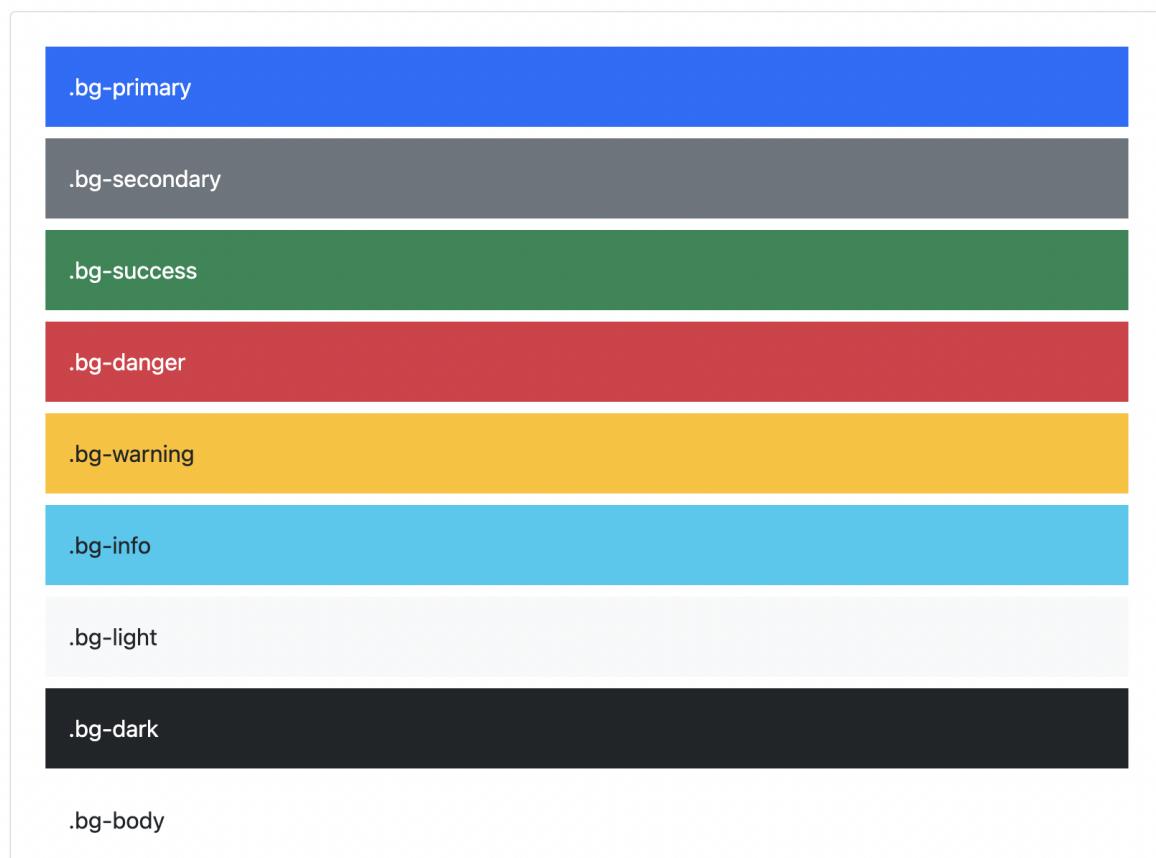


Imagen 9. Colores de fondo.

Fuente: Getbootstrap.com.

Márgenes y paddings

Ahora, trabajaremos con los márgenes entre elementos, para esto, revisemos la [documentación](#) de cómo notar márgenes y paddings.

Para escribir una clase para margin se debe comenzar con la letra **m**, y para escribir una clase para padding se debe comenzar con la letra **p**.

Luego, se debe aplicar otra letra que indica para qué lado lo quieras aplicar:

- t: para top.
- b: para bottom.
- s: para left o start.
- e: para right o end.
- x: para lo horizontal, o sea, left y right.
- y: para lo vertical, o sea, top y bottom.
- : sin especificar para todos lados.

Posteriormente, seguido de un **-** se debe asignar la cantidad de margen o padding (los tamaños van desde 0.25 rem a 3 rem):

- 0: elimina el margen o padding.
- 1.
- 2.
- 3.
- 4.
- 5.
- auto.

Por ejemplo, para aplicar un margen vertical al elemento **<p>**, haremos lo siguiente:

```
<p class="my-3">  
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud  
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.  
</p>
```

Como puedes ver, el código se ve bastante limpio al agregar las clases de Bootstrap para añadir los estilos y es sencillo de utilizar, si consultamos su documentación y seguimos las instrucciones y ejemplos de código que se nos presentan.



Actividad 1

- En una página web con Bootstrap cargado copia y pega el siguiente código:

```
<div class="">
  <p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea nemo voluptate, molestias autem quae suscipit pariatur quas culpa! Laborum cum ea expedita animi optio officiis pariatur eaque commodi nihil necessitatibus.</p>
</div>
```

- Agrega las clases necesarias para que la página web se vea como la siguiente imagen:

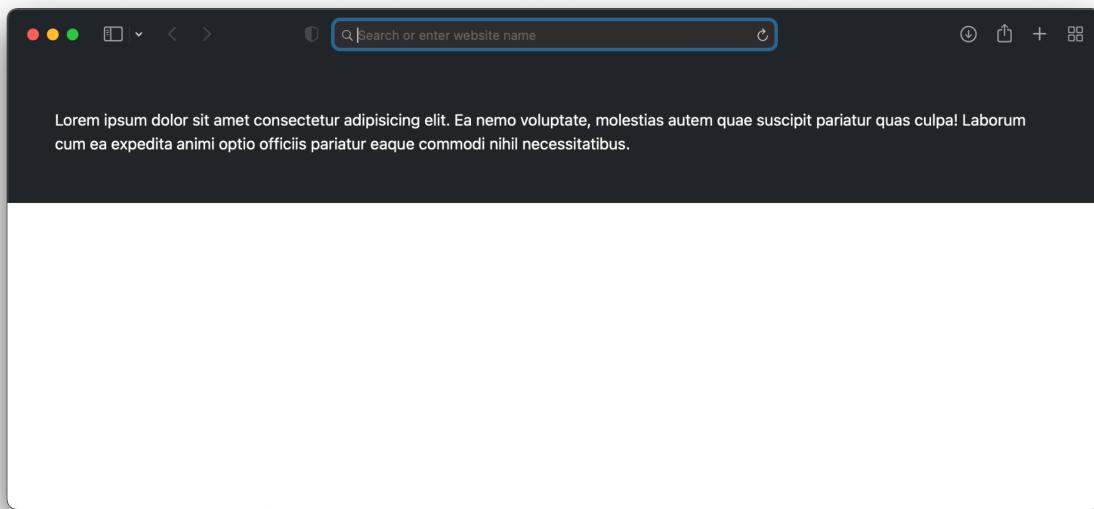


Imagen 10. Screenshot ejemplo de uso de clases de bootstrap

Fuente: Desafío Latam

Tip para resolver el ejercicio:



1. ¿Tiene background? ¿A cuál background de la lista de colores de fondo se parece?
2. ¿El texto está alineado de alguna forma particular?
3. ¿El color de texto es parte de alguno de los estudiados?
4. ¿Hay margin o padding? Utiliza la clase respectiva para lograr el efecto.

Bootstrap es un framework responsive

¿Qué es un diseño responsive?

Informalmente, hablamos de **diseño responsive** cuando la misma página web es posible verla tanto de un smartphone, tablet, notebook, computador de escritorio o incluso una televisión y se acomode para mostrarse bien en el espacio disponible.

Antes de precisar en la definición tenemos que introducir los conceptos de **viewport** y **breakpoints**.

Una página web se muestra dentro de una ventana, a veces esta ocupa toda la pantalla del dispositivo, pero en otras ocasiones tenemos múltiples ventanas abiertas o una sola ocupando parte de la pantalla, para evitar estas ambigüedades y referirnos específicamente al área visible de una página web en una ventana utilizaremos el término **viewport**.

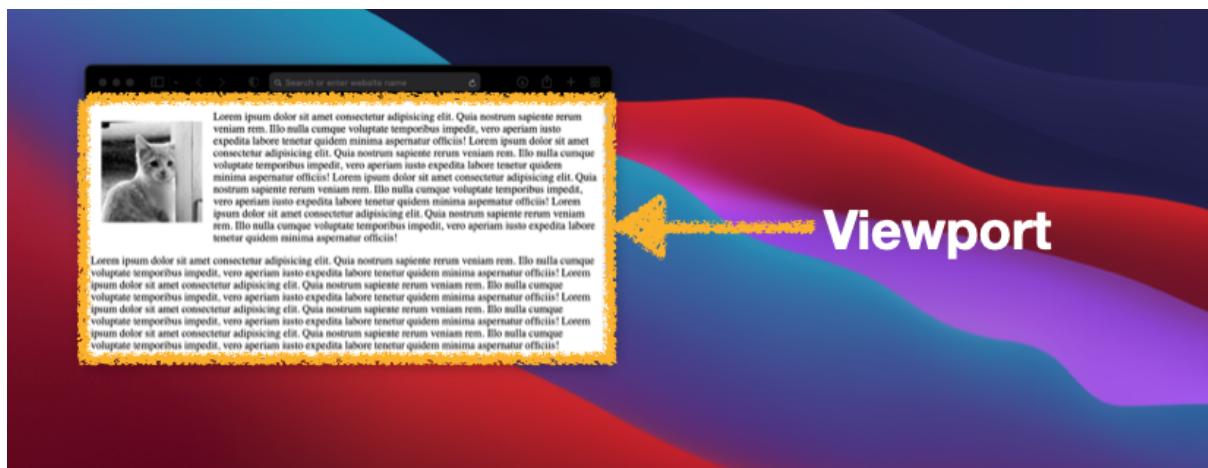


Imagen 11. Viewport
Fuente: Desafío Latam.

Para lograr que una página web se adapte al espacio que tiene disponible hay dos estrategias.

La primera consiste en especificar el tamaño de los contenedores en términos de porcentajes, de esta forma si un contenedor mide el 80%, esto será un 80% tanto de un viewport reducido como uno con mucho espacio.

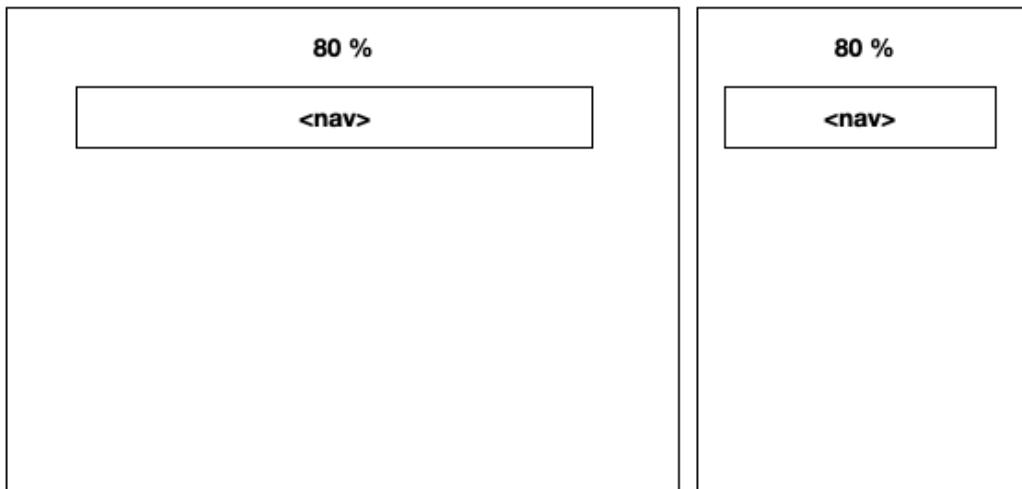


Imagen 12. Viewport en Desktop vs Viewport en Mobile.

Fuente: Desafío Latam.

La segunda estrategia consiste en utilizar breakpoints o puntos de quiebre indicando que el tamaño que debe tener un contenedor depende del viewport. Por ejemplo, podría ser que si el viewport tiene entre 800px y 1200px, el contenedor tendrá un tamaño de 600, pero si tiene un tamaño menor que 600, entonces se especifica que el contenedor tenga un ancho de 400px.

Para lograr especificar estos breakpoints se ocupan media queries, por ahora, nosotros no crearemos breakpoints, pero utilizaremos aquellos creados dentro del framework Bootstrap para crear diseños responsivos.

Contenedores de Bootstrap

Los contenedores de Bootstrap son el bloque básico de construcción y sirven para ajustar el contenido acorde al dispositivo y el viewport.

Podemos tener un único contenedor dentro de nuestro sitio o múltiples contenedores, usualmente uno por sección.

```
<!-- Opción 1 -->
<nav class="navbar"> </nav>
<div class="container"> Sección 1 </div>
<div class="container"> Sección 2 </div>
<div class="container"> Sección 3 </div>
<div class="container"> Sección 4 </div>
<footer> </footer>
```

```
<!-- Opción 2 -->
<div class="container"> Sección 1 </div>
<nav> </nav>
<section> </section>
<footer> </footer>
</div>
```

¿Cuál opción es mejor? La primera es más flexible, como aprenderemos a continuación, existen 2 tipos de contenedores y de esta forma podemos utilizar en algunos casos uno y en otros casos el otro, pero no todos los tipos de página requieren de esta flexibilidad y en ese caso sería más sencillo tener un único contenedor.

Tipos de contenedores

- **container**

Tienen un ancho fijo definido, el cual cambia a medida que cambia el tamaño del dispositivo.

- **container-fluid**

Tiene un ancho de un 100%, esto quiere decir que no está limitado por un ancho específico, sino que tiene todo el ancho de la página para funcionar. Por ejemplo, si necesitamos agregar varios párrafos, la mejor opción será un **container**, ya que tiene un ancho que limitará el texto hacia los lados.

Si lo que queremos es crear una sección que incluya un fondo con color, la mejor opción será una **container-fluid**, pues su ancho de 100% sin margen ayudará a que el fondo de color se vea a tamaño completo.



Actividad 2: Jugando con containers

- Crea la página containers.html con la estructura básica de un html5 autogenerada por vscode.
- Agrega bootstrap.
- Crea dos divs uno con clase container-fluid y el otro con clase container.
- Copia una barra de navegación dentro de cada uno de ellos.
- El código debería quedar así:

```
<div class="container-fluid mt-5">
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">...
    </nav>
</div>

<div class="container mt-5">
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">...
    </nav>
</div>
```

Imagen 13. Actividad.

Fuente: Desafío Latam.

Observarás que la barra de navegación está colapsada desde el editor para hacer más fácil de entender que está dentro del container.

Ahora:

- Abre la página.
- Reduce el viewport (el tamaño de la ventana de la página web) lentamente.
- Observa qué sucede con ambas barras de navegación.
- Vuelve a leer la diferencia entre container y container-fluid
 - Intenta identificar los breakpoints.
 - Pista: Es justo el momento en que hay un cambio abrupto en el tamaño de la barra de navegación.



Actividad 3

- En esta actividad debes clonar (duplicar) el proyecto realizado en la actividad 1, y agregar la clase container al div.
- Abrir la página.
- Probar con container-fluid.
- Observar ambas diferencias.

Sistema de Grillas de Bootstrap

El sistema de grillas en Bootstrap nos permite dividir una página en columnas:

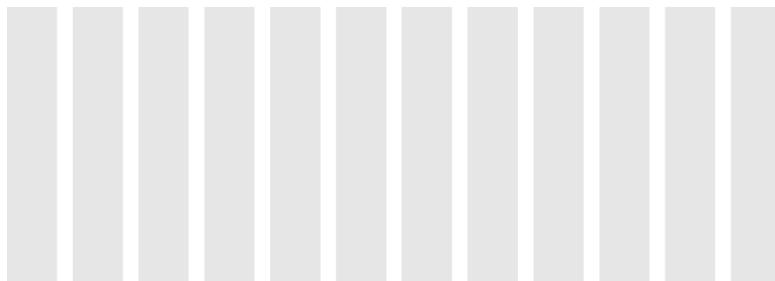


Imagen 14. Sistema de grillas.

Fuente: Desafío Latam.

Utilizar la grilla es tan sencillo como hacer lo siguiente, definimos una fila usando la clase `row` y dentro de esta fila creamos 3 columnas con la clase `col`:

```
<div class="row">
  <div class="col">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Eaque repudiandae assumenda
    aliquid deserunt eos ut, id at voluptas dicta nemo ratione magnam a nulla obcaecati
    perferendis minima maxime sint. Quia.
  </div>
  <div class="col">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Eaque repudiandae assumenda
    aliquid deserunt eos ut, id at voluptas dicta nemo ratione magnam a nulla obcaecati
    perferendis minima maxime sint. Quia.
  </div>
  <div class="col">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Eaque repudiandae assumenda
    aliquid deserunt eos ut, id at voluptas dicta nemo ratione magnam a nulla obcaecati
    perferendis minima maxime sint. Quia.
  </div>
</div>
```

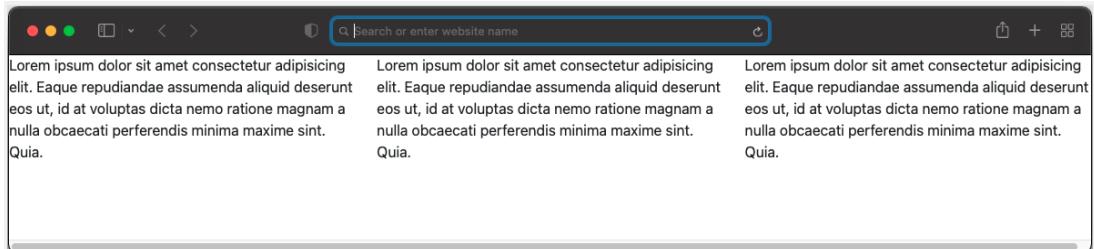


Imagen 15. Probando el sistema de grillas.

Fuente: Desafío Latam.

Combinando la grilla con contenedores podemos lograr rápidamente una sección de características.

```
<div class="container-fluid bg-dark text-light">
<div class="row">
  <div class="col p-5">
    <p class="text-center">
      <i class="bi-alarm text-center" style="font-size: 5em"></i>
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit, iure!
    </p>
  </div>
  <div class="col p-5">
    <p class="text-center">
      <i class="bi-activity text-center" style="font-size: 5em"></i>
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit, iure!
    </p>
  </div>
  <div class="col p-5">
    <p class="text-center">
      <i class="bi-bank text-center" style="font-size: 5em"></i>
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Odit, iure!
    </p>
  </div>
</div>
```

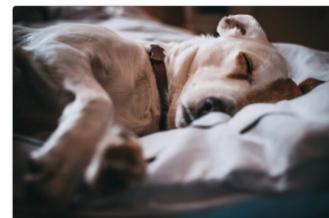
También podemos combinarlas con cards para crear una galería de imágenes como la siguiente.



Some quick example text to build on the card title and make up the bulk of the card's content.



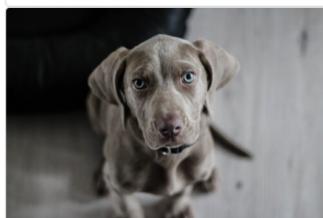
Some quick example text to build on the card title and make up the bulk of the card's content.



Some quick example text to build on the card title and make up the bulk of the card's content.



Some quick example text to build on the card title and make up the bulk of the card's content.



Some quick example text to build on the card title and make up the bulk of the card's content.



Some quick example text to build on the card title and make up the bulk of the card's content.

Imagen 16. Galería de imágenes sobre el sistema de grillas.

Fuente: Desafío Latam.



En el módulo siguiente se explica con mayor detalle el uso de Grid usando CSS, pero puedes revisar la [documentación de Mozilla](#) para comenzar a investigar. Por el momento, utilizaremos las clases que nos entrega Bootstrap.



Actividad 4

En una página web nueva con Bootstrap realiza los siguientes pasos para construir la galería de imágenes:

- Agrega un div con clase container (o container-fluid).
- Dentro del div, agrega dos div con clase row.
- Dentro de cada row, agrega 3 divs con clase col.
- Dentro de cada col, agrega un card sacado directamente de la documentación de bootstrap.
- Reemplaza el src de la img de card por <https://placelodog.net/500/300?id=1>
- Por cada imagen, cambia el número de la id para obtener imágenes distintas, en la segunda imagen la url debería ser <https://placelodog.net/500/300?id=2> y para la tercera <https://placelodog.net/500/300?id=3> y sucesivamente.
- Si ocupas la card que dice style="width: 18 rem" cámbialo por "width: 100%"
- Reduce el viewport al mínimo, ¿qué sucede con las cartas (cards)?
- Agrega una nueva columna y una card dentro del primer row, ¿cómo cambió el tamaño de las cartas?

Controlando el ancho de las columnas

La grilla de Bootstrap funciona con columnas. Cada vez que nosotros agregamos un contenedor con la clase col dentro de un contenedor con la clase row, el espacio total disponible se divide en espacios iguales.

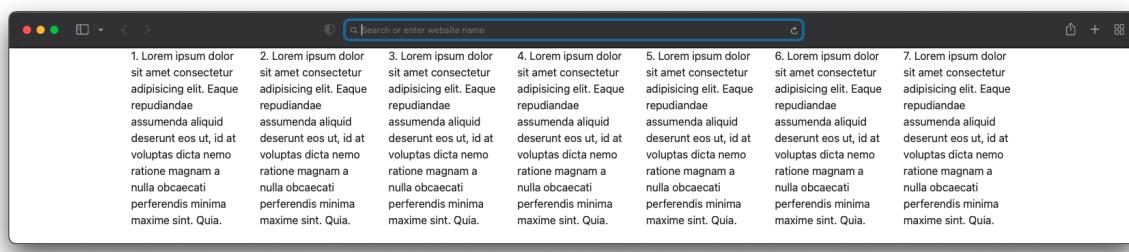


Imagen 17. Ejemplo con 7 columnas

Fuente: Desafío Latam.

También es posible controlar el ancho de una columna especificando un valor del 1 al 12 (más adelante explicaremos el motivo de este número), por ejemplo, en la actividad anterior puedes probar cambiando cualquier clase col por col-6 y obtendremos algo similar a lo que se muestra en la siguiente imagen:

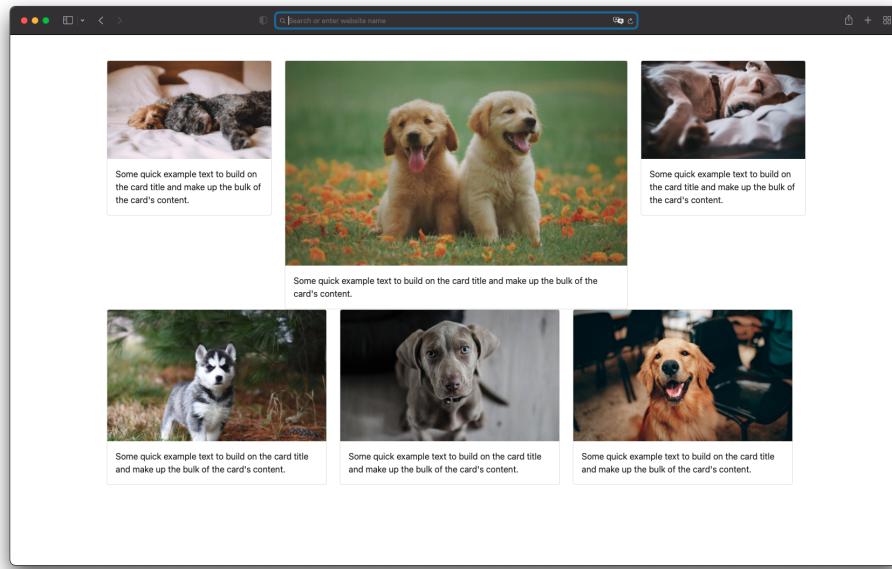


Imagen 18. Imágenes ajustadas a la grilla de Bootstrap
Fuente: Desafío Latam.

Bootstrap tiene una plantilla de 12 columnas que nos ayudan a disponer el contenido según nuestras necesidades.

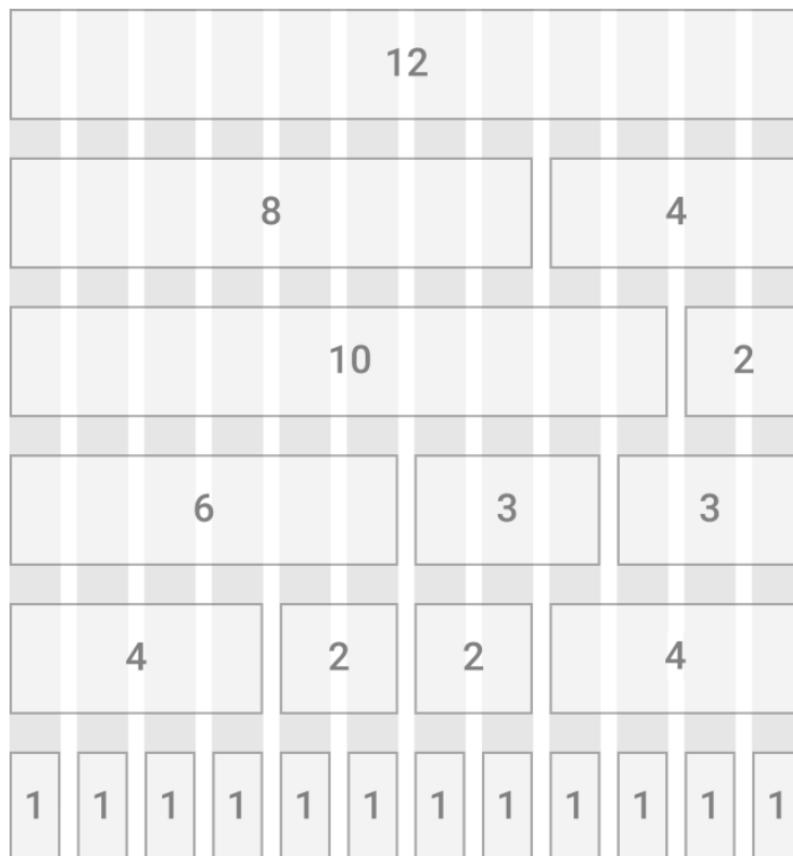


Imagen 19. Columnas.
Fuente: Desafío Latam.

Utilizando col-número podemos controlar el ancho a nuestra voluntad.

Resumen de grilla

- Cuando queramos dividir un espacio en columnas de igual tamaño utilizaremos col.
- Cuando queramos dividir el espacio acorde a necesidades específicas, emplearemos la plantilla de 12 columnas y distribuiremos el espacio según nuestras necesidades.
- Si en la plantilla de 12 columnas nuestra suma es mayor de 12, ese elemento pasará automáticamente a la siguiente fila.

Gutters

Los gutters o canaletas nos permiten generar separación vertical y horizontal en una grilla. Utilizando gutters podremos mejorar la distribución de espacio de nuestra galería de imágenes y lograr lo siguiente:

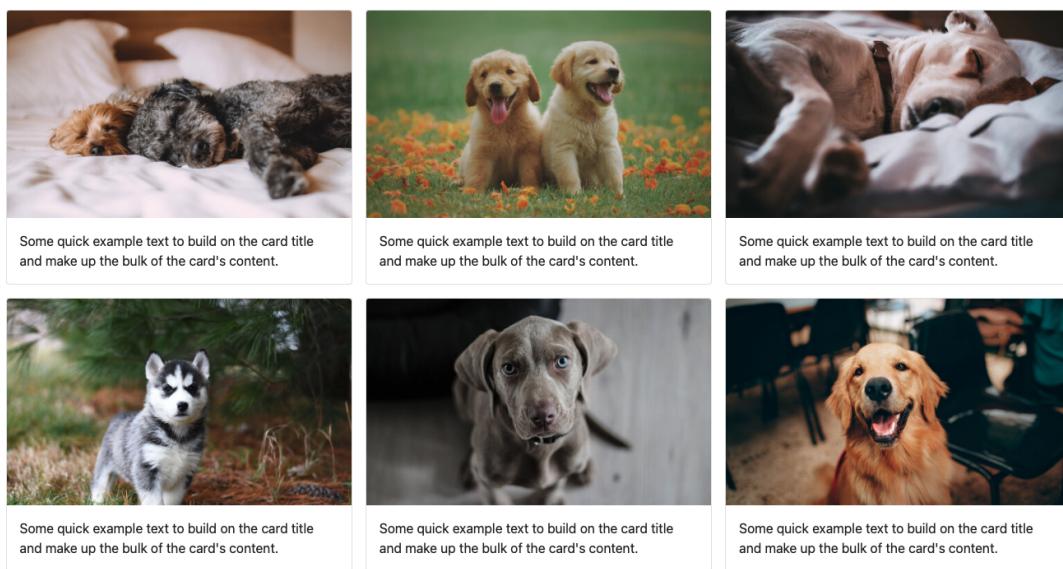


Imagen 20. Gutters

Fuente: Desafío Latam.

El cambio es un poco sutil, pero ahora hay más separación de espacio entre la primera fila de imágenes y la segunda.

Para lograr lo anterior hay una restricción muy importante, los gutters verticales funcionan cuando todos los elementos están en un único row. En la siguiente actividad haremos cambios en nuestra galería de imágenes para que cumpla la condición.



Actividad 5

- A partir del proyecto anterior, realizaremos una copia y modificaremos el html de la copia utilizando la siguiente estructura:

```
<div class="row">
    <!-- col 1 -->
    <div class="col">...
    </div>
    <!-- col 2-->
    <div class="col">...
    </div>
    <!-- col 3-->
    <div class="col">...
    </div>
    <!-- Quiebre (nueva fila)-->
    <div class="w-100"></div>
    <!-- col 4-->
    <div class="col">...
    </div>
    <!-- col 5-->
    <div class="col">...
    </div>
    <!-- col 6-->
    <div class="col">...
    </div>
</div>
```

Imagen 21. Código actividad.

Fuente: Desafío Latam.

¿Qué cambió?

En esta nueva versión todos los contenedores col son hermanos y pertenecen al mismo row. El quiebre de línea está dado por <div class="w-100"></div>



Puedes aprender más sobre esto revisando la documentación [column breaks](#).

Para agregar los gutters simplemente agregaremos las clases gx-2 y gy-3 al row, al abrir el navegador deberías poder observar el espacio vertical.

Ahora, prueba cambiando los valores que acompañan a gx y gy por valores del 0 al 5, por ejemplo con gx-0 y gy-3. Finalmente, deja el resultado que más te guste.

Grillas responsivas

En algunas ocasiones vamos a querer que nuestra galería de imágenes y/o productos se muestre en 3 columnas dentro de una pantalla grande, 1 columna en una pantalla chica y 2 columnas en pantallas medianas, como un tablet.

Gracias a bootstrap eso es sencillo de lograr solo agregando un par de clases:

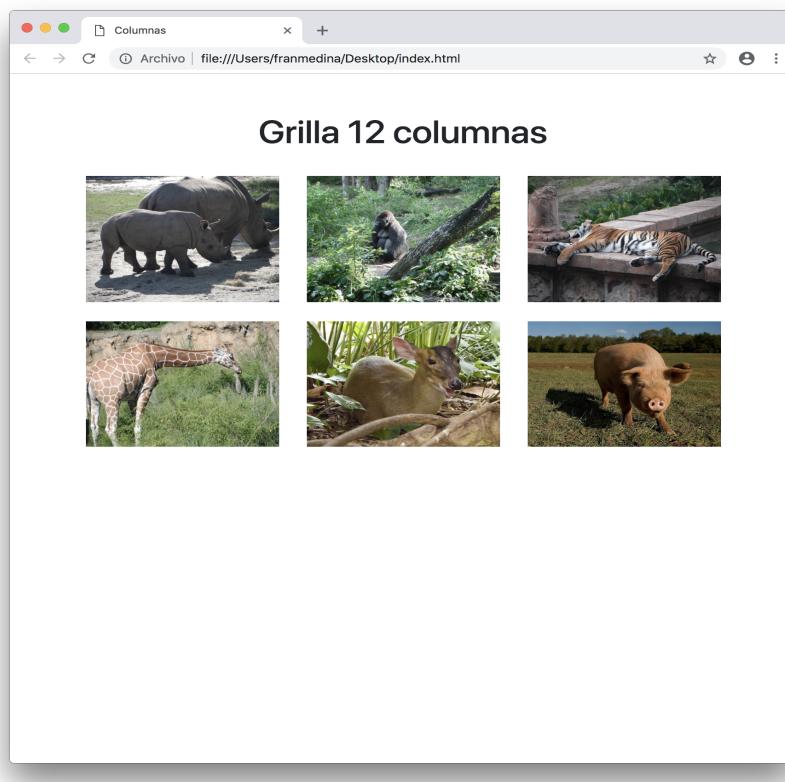


Imagen 22. Ejemplo 3 bloques de 4 columnas.

Fuente: Desafío Latam.

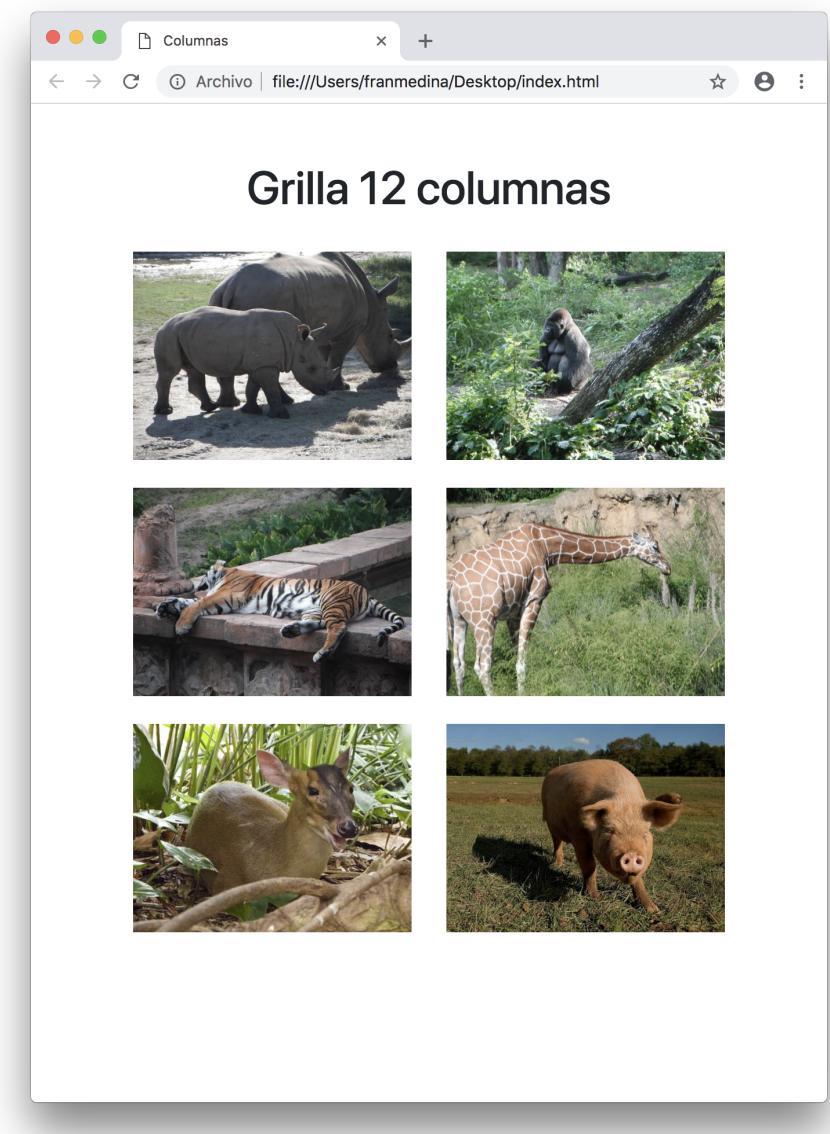


Imagen 23. Ejemplo 2 bloques de 6 columnas.

Fuente: Desafío Latam.



Imagen 24. Ejemplo 1 bloque de 12 columnas.
Fuente: Desafío Latam.

Para lograr que el número de columnas se defina en función del tamaño del dispositivo, tenemos que especificar un apellido a la clase col para que sepa cuantas columnas debe ocupar dependiendo de si está en un viewport largo, mediano, pequeño o muy pequeño.

```
<div class="col-xs-12 col-sm-6 col-md-4"> <!-- 1 columna de 12 en pantallas extra small (teléfonos), 2 columnas de 6 en pantallas small (tablets) y 3 columnas de 4 en pantallas medium (escritorio y notebooks) -->  
  
</div>
```

¿Cómo escoger el apellido para col?

Puedes utilizar la siguiente lista para verificar los tamaños.

- col-xs-N es para pantallas extra small (< 576px).
- col-sm-N es para dispositivos small ($\geq 576\text{px}$).
- col-md-N es para dispositivos medium ($\geq 768\text{px}$).
- col-lg-N es para dispositivos large ($\geq 992\text{px}$).
- col-xl-N es para dispositivos extra large ($\geq 1200\text{px}$).
- col-xxl-N es para dispositivos extra extra large ($\geq 1400\text{px}$).



Es importante entender que estos tamaños heredan desde el más chico al más grande, es decir, si ponemos "col-4" en un elemento, este tomará esa cantidad de columnas para small, medium, large y extra large y extra extra large.

Otro detalle importante es que cuando utilizamos la grilla de esta forma no es necesario introducir el quiebre con `<div class="w-100">` debido a que le estamos indicando, a través de la plantilla de 12 columnas, cuando empezar una línea nueva, y esto es justamente cuando nos pasamos de esa cantidad.



Actividad 6

- Duplica el proyecto de la galería y en el proyecto nuevo remueve el quiebre de línea.
- Luego, modifica las clases "col" de forma que se muestre 1 sola columna en teléfonos, 2 en tablets, 3 en tamaño medio (notebook y computadores) y 4 en pantallas grandes y de alta resolución.

Resumen

- Un framework CSS provee una estructura básica que nos ayudará a resolver problemas típicos de estilo dentro del desarrollo web.
- Bootstrap es un framework CSS, desarrollado por Twitter que sirve para maquetar páginas web de manera rápida y sencilla.
- Es de código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.
- Sus principales características son que es fácil de integrar, cuenta con un sistema de grillas, estilos en la mayoría de los elementos HTML, tiene una gran lista de componentes y una gran documentación.
- Bootstrap es un framework responsive, tiene herramientas como containers y grillas que nos permiten adaptar fácilmente nuestro contenido a múltiples dispositivos.

¡Continúa aprendiendo y practicando!

