

# DESAFÍO BSALE 2022

## EMPEZAR

Tienda online que despliegue productos agrupados por la categoría a la que pertenecen  
<https://github.com/ESCOBAR741/DesafioBSALE.git>

- Comenzamos añadiendo el framework de bootstrap 5 dentro del Head de nuestra aplicación web

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Tienda Online</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BeE4kWBq781YhF1dvKuhF7A76auUbtT94NrHftjDbrCExSU1oBogyl2QvZ6jIM3" crossorigin="anonymous">
<link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EV51QNi3/azprG1Ane3Q0gpJLIn9Nao@vz1rtCqTfWfspd3y065VohhpuuCOeLASjC" crossorigin="anonymous">
```

- Luego añadiendo el script en linea

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
```

# HTML5

Añadimos la base de la aplicación web con el lenguaje enmarcado de hiper texto definiendo las etiquetas con clases de bootstrap para la mejora del frontend obteniendo así una considerable mejora para la experiencia del usuario.

- Definimos la barra de navegación de tamaño mediano usando la clase de bootstrap "navbar navbar-expand-md" asignándole un color oscuro "navbar-dark", Luego definimos el contenedor con la clase "container fluid" momentaneamente sin ningún hipervínculo por lo que en su lugar dejamos un signo gato.

```
<header>
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Tineda Online</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"
      aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
```

- Definimos la clase para la barra de búsqueda declaramos que recibirá datos con la etiqueta "input" asignándole el un "id" para llamarlo luego entro del script, Luego agregamos un botón para buscar con la etiqueta "btn btn-outline-success".

```
<form class="d-flex">
  <input class="form-control me-2" id="formulario" type="search" placeholder="Search" aria-label="Search">
  <button class="btn btn-outline-success" id="boton" type="submit" >Buscar</button>
</form>
```

# VANILLA JAVASCRIPT

Para agregar la barra de búsqueda con javascript creamos un nuevo documento con extensión .js el que usaremos para el backend.

- Creamos una objeto para los productos con los nombres de los productos y el precio para buscarlos de ambas maneras separándolos por comillas simples.

```
const productos = [  
  {nombre: 'Zephyrus Duo 15', Sale:2700000},  
  {nombre: 'Asus ROG Strix G15', Sale:1399999},  
  {nombre: 'Zenbook Pro Duo', Sale:900000},  
]
```

- Creamos contantes para: formulario, botón y resultado con la etiqueta "document.querySelector" que devuelve el primer elemento del documento que permitirá la navegación entre "id".

```
const formulario =document.querySelector('#formulario');  
const boton = document.querySelector('#boton');  
const resultado = document.querySelector('#resultado');
```

- Comenzamos la función con el string básico "innerHTML" para que cada búsqueda comience desde cero, hacemos un recorrido de todos los productos con el ciclo "for" llamando a "producto" que viene del objeto "productos" usando la función que se encarga de pasar a minúsculas el texto introducido en el formulario para una mejor comparación con el objeto de productos con ".toLowerCase()".

Luego hacemos la pregunta "if" el texto con "indexOf" introducido en la barra de búsqueda coincide al objeto "producto" lo retorna y de no ser así muestra "-1" por pantalla.

Luego hacemos la pregunta "if" el texto con "indexOf" introducido en la barra de búsqueda coincide al objeto "producto" lo retorna y de no ser así muestra "-1" por pantalla, Luego a la constante "resultado" le aplicamos la propiedad "innerHTML" que establece el html de un elemento para así dejar el resultado de la búsqueda por listas y el precio del producto.

```
const filtrar = ()=>{
  resultado.innerHTML = '';

  const texto = formulario.value.toLowerCase();
  for(let producto of productos) {
    let nombre = producto.nombre.toLowerCase();
    if(nombre.indexOf(texto) !== -1) {
      resultado.innerHTML += `
      <li>${producto.nombre} - Sale: ${producto.Sale}</li> `
    }
  }
}
```

- Es mucho más práctico mostrar por pantalla un mensaje de error que un "-1" del indexOf por lo que anidamos otro "if" y hacemos la pregunta "if" contante resultado sigue siendo un string que sigue vacio o que nunca entró al ciclo "if" dela línea 18 mostramos por pantalla "producto no disponible".

```
if(resultado.innerHTML === '') {
  resultado.innerHTML += `
  <li>Producto No Disponible</li> `
}
```

- El método "addEventListener()" escucha el "click" que le hacemos al botón y te lleva a la función de "filtrar" y ejecuta todo el javascript escrito anteriormente, para una mayor practicidad añadimos el evento "keyup" el que ejecuta la búsqueda en productos cada vez que presionemos una letra dejando la función "filtrar" activa, Dejando las etiquetas script de tipo source dentro del "index.html".

```
boton.addEventListener('click', filtrar);
formulario.addEventListener('keyup', filtrar)

filtrar();
```

```
<div class="container mt-5">
  <ul id="resultado">
  </ul>
</div>
<script src="config/buscador.js"></script>
```

# HTML5

Continuamos con la aplicación web realizando un efecto carrusel con imágenes de los notebook.

- Abrimos la etiqueta "div" la que establece el inicio y fin de la clase "carousel slide" declarando el carrusel con 3 imágenes de clase activa con el "id=myCarousel".

```
<main>
<div id="myCarousel" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="2" aria-label="Slide 3"></button>
  </div>
```

- Abrimos una nueva etiqueta "div" llamando al carrusel con la clase "carousel-inner" después abriendo una nueva etiqueta "div" que activa el efecto carrusel con la clase "carousel-item active" y abrimos las etiquetas de imagen "img src" en la que establecemos la ruta de la imagen de clase "bd-placeholder-img" para que bootstrap reconozca la imagen luego abrimos un nuevo div de clase "carousel-caption text-end" para ver el texto al lado de la imagen y finalizando con un botón de color azul que usaremos para el precio de clase "btn btn-lg btn-primary" sin ningún hipervínculo cerrando las etiquetas "div".

```
<div class="carousel-inner">
  <div class="carousel-item active">

    <div class="container">
      <div class="carousel-caption text-end">
        <h1>Zephyrus Duo 15</h1>
        <p><a class="btn btn-lg btn-primary" href="#">$ 900.000</a></p>
      </div>
    </div>
  </div>
```

# HTML5

- Creamos un botón se anterior con la clase "carousel-control-prev" de tipo "botón" con el id "#myCarousel" declarando así que es parte de este abriendo la etiqueta span que permite asignar el estilo a un elemento pequeño con la clase "carousel-control prev-icon".

```
<button class="carousel-control-prev" type="button" data-bs-target="#myCarousel" data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previous</span>
</button>
```

- Una vez terminado el controlador derecho realizamos otro para el izquierdo de una manera muy similar cambiando las clase del botón a "carousel-control-next" y de la etiqueta "span" a "carousel-control-next-icon"

```
<button class="carousel-control-prev" type="button" data-bs-target="#myCarousel" data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#myCarousel" data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Next</span>
</button>
```

# HTML5

Establecemos columnas para las imágenes de clase "container marketing"

- Abrimos la etiqueta div con la clase "row" que establece que el contenido dentro de la etiqueta será por filas para enseguida abrir una nueva etiqueta "div" de clase "col-lg-4" para dispositivos con pantallas menores a 992 píxeles de una manera más óptima abriendo la etiqueta "img" asignándole la ruta de la imagen de clase "bd-placeholder-img rounded-circle" el cual permite que las imágenes se ven en círculos, la etiqueta "alt" permite introducir texto que se verá en vez de la imagen cuando esta no pueda cargarse

```
<div class="row">
  <div class="col-lg-4">
    
    <h2>Zephyrus Duo 15</h2>
    <p><b>CPU:</b> Intel Core i9 <b> GRÁFICOS:</b> RTX 2080 SUPER <b>RAM:</b> 32 GB mhz <b>PANTALLA:</b> ROG ScreenPad Plus que tiene
    <p><a class="btn btn-secondary" href="#">Más Información &raquo;</a></p>
```

- Repetimos el proceso para el resto de las imágenes cambiando estas y el texto según los requerimientos quedando de la siguiente manera

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

  <div class="row">
    <div class="col-lg-4">
      
      <h2>Zephyrus Duo 15</h2>
      <p><b>CPU:</b> Intel Core i9 <b> GRÁFICOS:</b> RTX 2080 SUPER <b>RAM:</b> 32 GB mhz <b>PANTALLA:</b> ROG ScreenPad Plus que tiene
      <p><a class="btn btn-secondary" href="#">Más Información &raquo;</a></p>
    </div>
    <div class="col-lg-4">
      
      <h2>Asus ROG Strix G15</h2>
      <p><b>CPU:</b> AMD Ryzen 9 Serie 5000<b> GRÁFICOS:</b> GeForce RTX 3060 <b>RAM:</b> 16 GB 3200 mhz <b>PANTALLA:</b> 15,6 OLED 4K
      <p><a class="btn btn-secondary" href="#">Más Información &raquo;</a></p>
    </div>
    <div class="col-lg-4">
      
      <h2>ZENBOOK PRO DUO </h2>
      <p><b>CPU:</b> Intel Core i9 <b> GRÁFICOS:</b> GeForce RTX 2060 <b>RAM:</b> 32 GB <b>PANTALLA:</b> 15,6 OLED 4K táctil ScreenPad
      <p><a class="btn btn-secondary" href="#">Más Información &raquo;</a></p>
    </div>
  </div>
```