



# Proyecto de Escatech CA

## Front-End I

**Alumno:** Jose Alejandro Escalona

**Ci:** 28.206.133



## **EstaTech CA**

Escatech es una empresa dedicada a la venta de productos gamer, ofreciendo una amplia variedad de computadoras, periféricos, laptops y teléfonos móviles de alta calidad y rendimiento. La empresa se enfoca en brindar a sus clientes una experiencia de juego inigualable, ofreciendo productos de última generación y de las mejores marcas del mercado.

Además de la venta de productos gamer, Escatech también ofrece servicios de reparación y configuración de equipos. La empresa cuenta con un equipo de técnicos altamente capacitados y experimentados que pueden reparar y configurar todo tipo de equipos. Los servicios de reparación y configuración de Escatech se enfocan en brindar soluciones rápidas y efectivas a los problemas que puedan presentarse en los equipos de sus clientes.





## Explicación del código

El proyecto que he desarrollado es una página web que utiliza el framework Express para el back-end, una base de datos MySQL para almacenar y gestionar datos, y HTML y CSS para el front-end.

Express es un framework de Node.js que se utiliza para crear aplicaciones web y APIs. Con Express, se pueden crear fácilmente rutas, manejar solicitudes HTTP, procesar datos y mucho más. En este proyecto, he utilizado Express para crear una API que se conecta a la base de datos MySQL y maneja las solicitudes de los usuarios.

MySQL es un sistema de gestión de bases de datos que se utiliza para almacenar y gestionar datos. En este proyecto, he utilizado MySQL para crear una base de datos que almacena información sobre los usuarios y los productos de la página web. La base de datos se conecta a la API de Express para permitir que los usuarios realicen acciones como crear cuentas, iniciar sesión, agregar productos al carrito de compras y realizar compras.

HTML y CSS son lenguajes de marcado y estilo que se utilizan para crear la interfaz de usuario de la página web. En este proyecto, he utilizado HTML y CSS para crear la estructura de la página web y darle estilo. También he utilizado JavaScript para agregar interactividad a la página web, como validación de formularios y animaciones.

En este proyecto, se utilizó Webpack para empaquetar y optimizar el código JavaScript y CSS de la página web. Webpack se encargó de combinar todos los archivos JavaScript, HTML y CSS en un solo archivo y comprimirlos para reducir su tamaño. Además, se utilizaron diferentes plugins para optimizar el código y ajustar la configuración de Webpack para que se adapte a las necesidades del proyecto.



## BACK-END

Se presenta el entry point del proyecto, en el cual se ve el proceso principal por donde corre la web:

```
JS app.js > ...
1  const {conectar} = require('./src/ConexionSQL.js')
2  conectar()
3
4  var express = require('express');
5  var createError = require('http-errors');
6  var path = require('path');
7  var cookieParser = require('cookie-parser');
8  var logger = require('morgan');
9
10 var indexRouter = require('./routes/index');
11 var ProductosRouter = require('./routes/productos');
12 var IngresarRouter = require('./routes/ingresar');
13 var RegistrarRouter = require('./routes/registras');
14
15 var app = express();
16
17 app.set('views', path.join(__dirname, 'views'));
18 app.set('view engine', 'ejs');
19
20 app.use(logger('dev'));
21 app.use(express.json());
22 app.use(express.urlencoded({ extended: false }));
23
24 app.use(cookieParser());
25 app.use(express.static(path.join(__dirname, 'public')));
26
27 app.use('/', indexRouter);
28 app.use('/productos', ProductosRouter);
29 app.use('/ingresar', IngresarRouter);
30 app.use('/registras', RegistrarRouter);
31
32
33 app.use(function(req, res, next) {
34   next(createError(404));
35 });
36
37 app.use(function(err, req, res, next) {
38   res.locals.message = err.message;
39   res.locals.error = req.app.get('env') === 'development' ? err : {};
40   res.status(err.status || 500);
41   res.render('error');
42 });
43
44 module.exports = app;
```

Como se mencionó antes, se utilizó express para realizar el Back-End, deparando los routes que se realizaron (4) y haciendo conexión con la base de datos.



El diseño de la base de datos está guardado en la carpeta src, para probar el funcionamiento del proyecto.

```
1 create database DBEscatech;
2
3 use DBEscatech;
4
5 create table Clientes(
6     NombreCliente varchar(100) not null,
7     CedulaCliente mediumint not null primary key,
8     Correo varchar(100) not null,
9     Usuario varchar(50) not null,
10    Contraseña varchar (50) not null,
11    Telefono mediumint,
12    Direccion varchar (150) not null
13 );
14
15 create table Operadores(
16     NombreOperador varchar(100) not null,
17     CedulaOperador mediumint not null primary key,
18     Correo varchar(100) not null,
19     Usuario varchar(50) not null,
20     Contraseña varchar (50) not null,
21     Telefono mediumint,
22     Direccion varchar(150) not null,
23     Rol varchar(50) not null
24 );
25
26 create table Productos(
27     IdProducto mediumint not null primary key auto_increment,
28     NombreProducto varchar(100) not null,
```

Los routes hacen el render del HTML de la web utilizando el método GET para consultas, y con el método POST, guarda la información brindada en los formularios a la base de datos.

## Routes index

```
routes > JS index.js > ...
1  var express = require('express');
2  var router = express.Router();
3
4  var controller = require('../controllers/indexcontroller.js')
5
6  router.post(
7    '/',
8
9    function(req, res){
10      let DatosFormulario = req.body;
11      controller.Agregar(DatosFormulario)
12        .then((resultado)=>{
13          res.render('index');
14        })
15        .catch((err)=>{
16          res.send(err)
17        })
18    });
19
20
21 router.get(
22   '/',
23   function(req, res) {
24     controller.ListarDestacados()
25       .then((resultado)=>{
26         res.render('index', { "JSON": JSON.stringify(resultado)});
27       })
28       .catch((err)=>{
29         res.send(err)
30       })
31     });
32
33
34
35 module.exports = router;
36
```



## Routes ingresar

```
routes > JS ingresar.js > ...
1  var express = require('express');
2  var router = express.Router();
3
4  router.get('/', function(req, res, next) {
5    |   res.render('ingresar');
6    | });
7
8
9  module.exports = router;
10 |
```

## Routes productos

```
routes > JS productos.js > ...
1  var express = require('express');
2  var router = express.Router();
3
4  router.get('/', function(req, res, next) {
5    |   res.render('productos');
6    | });
7
8  module.exports = router;
9
```

## Routes registrar

```
routes > JS registrar.js > ...
1  var express = require('express');
2  var router = express.Router();
3
4  var controller = require('../controllers/registrarcontroller.js')
5
6  router.get('/', function(req, res, next) {
7    |   res.render('registrar');
8    | });
9
10 router.post(
11   '/',
12   function(req, res){
13     let DatosRegistrar = req.body;
14     controller.Agregar(DatosRegistrar)
15       .then((resultado)=>{
16         res.render('registrar');
17       })
18       .catch((err)=>{
19         res.send(err)
20       })
21   });
22
23
24
25 module.exports = router;
26
```



Por el momento todos los routes hacen consulta GET, pero solamente el index y el register hacen POST, por desarrollo de esta aplicación (Se quiere agregar otros elementos que no se especifican en la actividad, pero es más desarrollo de Back-End que de Front-End).

Se realizaron controladores diferentes para cada routes que hacen POST, impidiendo que a la larga hayan problemas con visualizar el código fuente.

```
1  var FormularioSQL = require('../src/FormularioIndexSQL.js')
2
3  class FormularioIndexController {
4
5      Agregar(DatosFormulario){
6          return new Promise ((resolve, reject)=>{
7              if (!DatosFormulario.Nombre || !DatosFormulario.Correo || !DatosFormulario.Equipo || !DatosFormulario.Mensaje) {
8                  return resolve("Compruebe uno de los datos a ingresar");
9              }
10
11              FormularioSQL.Agregar(DatosFormulario)
12                  .then((resultado)=>{
13                      resolve (resultado)
14                  })
15                  .catch((err)=>{
16                      reject(err)
17                  });
18          })
19      }
20
21      ListarDestacados(){
22          return new Promise ((resolve, reject)=>{
23              FormularioSQL.Listar()
24                  .then((resultado)=>{
25                      resolve (resultado)
26                  })
27                  .catch((err)=>{
28                      reject(err)
29                  });
30          })
31      }
32  }
33
34  }
35
36  const ControladorIndex = new FormularioIndexController()
37
38  module.exports = ControladorIndex
39
```

En los cuales, requieren de la información que viene de la base de datos SQL



```
src > $5 FormularioIndexSQL.js > ...
1  const mysql = require('mysql');
2
3  const connection = mysql.createConnection({
4    host: process.env.HOST,
5    user: process.env.USER,
6    password: process.env.PASSWORD,
7    database: process.env.DATABASE
8  })
9
10 class FormularioSQL {
11
12   Agregar(DatosFormulario){
13     let nombre = ''+DatosFormulario.Nombre+''
14     let correo = ''+DatosFormulario.Correo+''
15     let Equipo = ''+DatosFormulario.Equipo+''
16     let Mensaje = ''+DatosFormulario.Mensaje+''
17
18
19     return new Promise((resolve, reject) => {
20
21       connection.query('INSERT INTO servicioreparacion (Nombre, Correo, Equipo, Mensaje) VALUES (' +nombre+ ', ' +correo+ ', ' +Equipo+ ', ' +Mensaje+ ')
22       function (error, results, fields){
23         if(error) throw error;
24         console.log('Se ha guardado el registro');
25         resolve(DatosFormulario)
26       }
27     });
28   })
29 }
30
31 Listar(){
32   return new Promise((resolve, reject)=>{
33
34     connection.query('SELECT * from Productos', function (error, results, fields) {
35       if (error) throw error;
36       resolve(results);
37     });
38   })
39 }
40 }
41
42
43 const ClaseFormulario = new FormularioSQL();
44 module.exports = ClaseFormulario;
45
```

Aquí hace el guardado de los datos que ingresa el cliente en el formulario.

El router del index hace un GET a los productos que hay guardados en el inventario, y los muestra en el apartado de productos destacados.







Opciones extra

		IdProducto	NombreProducto	Modelo	Marca	Detalles	Precio	Estatus	Foto		
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Mouse Redragon Cobra M711	Cobra M711	Redragon	8 Botones, Conexión USB, DPI mínimo 100, Frecuenci...	25,00	Activo	images/Mouse2.png
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	HP H220S Audífonos Gamer	H220S	HP	Conexión de audio de 3.5 mm, Compatibilidad: PS4, ...	50,00	Activo	images/Audífonos4.jpg
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Procesador AMD Ryzen 5 2gen	Ryzen 5	AMD	6 Nucleos, 12 Hilos, Hasta 4.4GHz, Caché L2 total ...	150,00	Activo	images/Ryzen5-2gen
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	Memoria RAM DDR4	F4-3200C16D-16GTZR	G.Skill	8GB, PC4 25600, DDR43200MHz	40,00	Activo	images/MemoriaRam1.jpg
<div> <input type="checkbox"/> Seleccionar todo</div> <div>Para los elementos que están marcados:</div> <div> Editar</div> <div> Copiar</div> <div> Borrar</div> <div> Exportar</div>											

Se muestran en ese apartado ya que se habría que hacer un algoritmo en el que busque que productos han sido los más vendidos, los de menor precio, ect. Pero ese es más trabajo de Back-End.

## FRONT-END

Se quiso llevar a cabo un diseño acorde a los productos vendidos; se añadieron colores llamativos con efectos de luces RGB, mostrando también los productos más destacados en la tienda, las categorías de ventas, ubicación y contacto, además de un menú de ingreso y registro de perfil, para ver tu carrito de compras.

### Imagen del index



## Index

Para ello se desarrolló todo en el main, utilizando sections para las diferentes secciones de la web. Incluido a esto su header y li para las listas de productos y navegación.

### header

```
<header>

  <div class="EscaTech-Logo">
    <a href="/">  </a>
    <a href="/">  </a>
  </div>

  <nav class="NavHeader">
    <ul>
      <li>
        <ion-icon name="search-outline"></ion-icon>
        <input type="text" placeholder="Buscar" style="text-indent: 10px;">
      </li>

      <select class="selectheader" onchange="window.location.href = this.value;" id="selectheader">
        <option value="/"> Inicio </option>
        <option value="/productos"> Productos</option>
        <option value="#"> Contacto </option>
        <option value="/ingresar"> Acceder </option>
        <option value="/registrar" class="RegistrarLista"> Registrar </option>
      </select>

      <div class="NavPC">
        <li> <a href="/">Inicio</a> </li>

        <li> <a href="/productos">Productos</a> </li>

        <li> <a href="#">Contacto</a> </li>

        <li> <button><a href="/ingresar">Acceder</a></button> </li>

        <li> <button class="BotonRegistrar"><a href="/registrar">Registrar</a></button></li>
      </div>
    </ul>
  </nav>
</header>
```

### Section (Información de quienes somos)

```
<section style="background-image: linear-gradient(to bottom, #a6e2fd, #e0f2f8, #a6e2fd);">

  <div class="QuienesSomos">
    
    <h2>Quiénes somos</h2>
    <p> Somos una empresa especializada en la venta de equipos y componentes de informática para gamers. Nuestro objetivo es proporcionar a nuestros clientes la mejor experiencia de juego posible. En nuestra tienda, encontrarás una amplia variedad de productos para gamers, incluyendo equipos de escritorio, portátiles, monitores, teclados, ratones, auriculares y más. Además, ofrecemos servicios de asesoramiento y soporte técnico para ayudarte a elegir el mejor equipo para tus necesidades.</p>
  </div>
</section>
```



## Section (Productos destacados)

```
<section style="background-image: linear-gradient(to bottom, rgb(1, 26, 110), rgb(7, 50, 189), rgb(1, 26, 110));">
  <div class="Productosdestacados">
    <h2>Productos destacados</h2>
    <div style="display: flex; align-items: center; justify-content:center;">
      <button id="BotonIzquierda"></button>
      <div class="carousel">
        <ul id="DatosDestacado">
          </ul>
        </div>
      <button id="BotonDerecha"></button>
    </div>
  </div>
  <a class="MostrarMas" href="/productos">Mostrar mas productos</a>
</section>
```

## Section (Categorías)

```
</section>
<section>
  <ul class="Categorias">
    <li onclick="window.location.href='#' class="Computadoras"><a href="#">Computadoras</a></li>
    <li onclick="window.location.href='#' class="Telefonos"><a href="#">Teléfonos</a></li>
    <li onclick="window.location.href='#' class="Perifericos"><a href="#">Periféricos</a></li>
    <li onclick="window.location.href='#' class="Laptops"><a href="#">Laptops</a></li>
  </ul>
</section>
```



## Section (Formulario y ubicación)

```
<section class="Contacto">
  <div class="InformacionContacto">
    <table>
      <tr>
        <td class="TextoContacto">
          <h2 style="margin-top: 0;">Contacto para servicios de reparación y configuración de equipos</h2>
          <p> En nuestra empresa, ofrecemos servicios de reparación y configuración de equipos informáticos para particulares y empresas. Si tu equipo presenta algún problema o no funciona, nuestro equipo de expertos en configuración de equipos podrá ayudarte a resolverlo.
          <p> Además, si necesitas configurar tu equipo para un uso específico, como por ejemplo para trabajar desde casa, nuestro equipo de expertos en configuración de equipos podrá ayudarte a hacerlo.
          <p> No importa si se trata de un problema de hardware o software, o si necesitas ayuda para configurar tu equipo para un uso específico, nuestro equipo de técnicos altamente capacitados te ayudará a resolverlo.
        </td>
        <td class="ImagenContacto">  </td>
      </tr>
      <tr>
        <td class="ImagenContacto2">  </td>
      </tr>
    </table>
  </div>
  <div class="Formulario">
    <form action="" method="POST">
      <h2> Agrega tus datos al formulario</h2>
      <label for="nombre">Nombre:</label>
      <ion-icon name="person-outline"></ion-icon>
      <input type="text" id="nombre" name="Nombre" class="Texto"><br>
      <label for="email">Email:</label>
      <ion-icon name="mail-outline"></ion-icon>
      <input type="email" id="correo" name="Correo" class="Texto"><br>
      <label for="equipo">Equipo:</label>
      <ion-icon name="desktop-outline"></ion-icon>
      <input type="text" id="equipo" name="Equipo" class="Texto"><br>
      <label for="mensaje">Mensaje:</label>
      <ion-icon name="text-outline"></ion-icon>
      <input type="text" id="mensaje" name="Mensaje" class="TextoMensaje"><br>
      <input type="submit" value="Enviar formulario">
    </form>
    <div class="mapa">
      <h2 style="margin-top: 0;">Nuestra ubicación</h2>
      <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d62994.161199668335!2d-70.6560383693167!3d9.32129763203319!2m3!1f0!2f0!3f0!3m2!1!1024!2!768!4f13!1!3m3!1m2!1s0x8e632869:~</div>
    </div>
    <div class="mapa2">
      <h2 style="margin-top: 0;">Nuestra ubicación</h2>
      <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d62994.161199668335!2d-70.6560383693167!3d9.32129763203319!2m3!1f0!2f0!3f0!3m2!1!1024!2!768!4f13!1!3m3!1m2!1s0x8e632869:~</div>
    </div>
  </section>
```

Footer (Información de copy) y los scripts (Utilizando innerHTML para agregar los li de la consulta GET)

```
<footer>
  <p>Derechos reservados &copy; 2023 Tienda de tecnología</p>
</footer>

<p id="DatosExtraidos" style="display: none;"> <%= JSON %> </p>
<script>
  let res = document.getElementById("DatosDestacado")
  let Datosjson = JSON.parse(document.getElementById("DatosExtraidos").innerText)
  res.innerHTML = ""

  let contador = 0

  for(let item of Datosjson){
    contador = contador + 1
    res.innerHTML += `
      <li onclick="window.location.href='#' " class="Productos">
        
        <a>${item.NombreProducto}<br><b>Precio ${item.Precio}</b></a>
      </li>
    `
  }
</script>
<script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
```



## Productos

Todas las rutas tienen el mismo diseño, lo único que cambian son las vistas que se hacen, en este se muestran los productos por categorías, además de una barra lateral en la que puedes ver las diferentes categorías que hay en la tienda.

Escatech CA

Buscar

Inicio

Productos

Contacto

Ingresar

Registrar

Categorías

Computadoras

Teléfonos


Periféricos

Laptops


Productos disponibles en Escatech

Ordenar por: Más relevante


Computadoras



PC Gamer Intel i3 12100, Rtx 3060, 32gb Ram  
Precio \$1000,00




PC Gamer Intel i5 12400F, 16gb, Rtx 3060, 16gb Ram  
Precio \$1400,00




PC Gamer Alienware i7, 16gb Ram, HDD 1TB, Nvidia Gtx 760 4 gb  
Precio \$1900,00


Teléfonos



Samsung Galaxy S20 FE 5G  
Precio \$350,00




Samsung Galaxy S21 5G  
Precio \$600,00




Redmagic 7 5G  
Precio \$500,00


Periféricos



Mouse Redragon Cobra M711  
Precio \$25,00




HP H220S Audifonos Gamer  
Precio \$50,00




Monitor Gaming Curvo 27 Pulgadas, Led-Full Hd, 185hz  
Precio \$320,00

Laptops



Acer Nitro Gaming Laptop Intel Core i7 11th, Rtx 3050 Ti  
Precio \$1200,00



Laptop Lenovo Gaming Core i5-11300h, 8gb Ram 556gb Video High  
Precio \$1000,00

Derechos reservados © 2023 Tienda de tecnología



## Barra lateral

```
<nav class="BarraLateral">
  <ul>
    <li> <h2>Categorías</h2> </li>

    <li> <a href="#">Computadoras</a> </li>

    <li> <a href="#">Teléfonos</a> </li>

    <li> <a href="#">Periféricos</a> </li>

    <li> <a href="#">Laptops</a> </li>

  </ul>
</nav>
```

Se utilizó un Nav para hacer una navegación dentro de la misma ruta, con una lista de las categorías que hay.

Se encuentran diferentes secciones para todas las categorías, por lo que se muestra una sola section, las demás son similar a esta.

```
<section style="background-image: linear-gradient(to bottom, rgb(1, 26, 110), rgb(7, 50, 189), rgb(1, 26, 110)); color: white;">
  <h2 style="text-align: center; margin-bottom: 0;"> Teléfonos </h2>

  <div class="Productos">
    <ul class="ListaProductos">

      <li onclick="window.location.href='#'">
        
        <a>Samsung Galaxy S20 FE 5G<br><b>Precio $350,00</b></a>
      </li>

      <li onclick="window.location.href='#'">
        
        <a>Samsung Galaxy S21 5G<br><b>Precio $600,00</b></a>
      </li>

      <li onclick="window.location.href='#'">
        
        <a>Redmagic 7 5G<br><b>Precio $500,00</b></a>
      </li>

      <li onclick="window.location.href='#'">
        
        <a>Xiaomi 12 Pro<br><b>Precio $270,00</b></a>
      </li>

    </ul>
  </div>
</section>

<section style="background-image: linear-gradient(to bottom, rgb(226, 233, 255), rgb(255, 255, 255), rgb(226, 233, 255))">
```

A futuro, cuando se implementa la misma estructura que la de productos destacados en el index, cargará todos los productos por categoría en la lista.



## Register y Login

Ambas rutas tienen la misma estructura, cambiando algunas cosas en ellas, como la cantidad de labels e inputs de texto. Ellos tienen un video de fondo con método de parallax, que se queda estático mediante vas bajando en el scroll.

The screenshot shows the 'Login' form on the EscaTech CA website. The form is centered on a dark blue background with a parallax effect of a city skyline at night. The form has a light blue header with the title 'Login'. Below the title, there are two input fields: 'Correo o Usuario' with a user icon and 'Contraseña' with a lock icon. A blue 'Ingresar' button is at the bottom of the form. The website header includes the EscaTech CA logo, a search bar, and navigation links: Inicio, Productos, Contacto, Ingresar, and Registrar. The footer contains the text 'Derechos reservados © 2023 Tienda de tecnología'.

The screenshot shows the 'Registrar' form on the EscaTech CA website. The form is centered on a dark blue background with a parallax effect of a city skyline at night. The form has a light blue header with the title 'Registrar'. Below the title, there are five input fields: 'Nombre' with an 'Aa' icon, 'Cédula' with a lock icon, 'Correo' with an envelope icon, 'Usuario' with a user icon, and 'Contraseña' with a lock icon. A blue 'Registrar' button is at the bottom of the form. The website header includes the EscaTech CA logo, a search bar, and navigation links: Inicio, Productos, Contacto, Ingresar, and Registrar. The footer contains the text 'Derechos reservados © 2023 Tienda de tecnología'.



Estas rutas tienen un solo section, el cual es el que tiene los labels e inputs para hacer consultas o agregar datos a la base de datos.

```
<section>
  <div class="centro">
    <div>
      <form action="" method="POST" id="loginForm">
        <h2>Login</h2>

        <div class="cajastexto">
          <ion-icon name="person-outline"></ion-icon>
          <input type="User" name="Usuario" id="Usuario" required>
          <label for="">Correo o Usuario</label>
        </div>

        <div class="cajastexto">
          <ion-icon name="lock-closed-outline"></ion-icon>
          <input type="password" name="Contraseña" id="Contraseña" required>
          <label for="">Contraseña</label>
        </div>

        <button type="submit">Ingresar</button>
      </form>
    </div>
  </div>
</section>
```

Se pueden ver los input text y el botón que hace la consulta o ingresa datos.

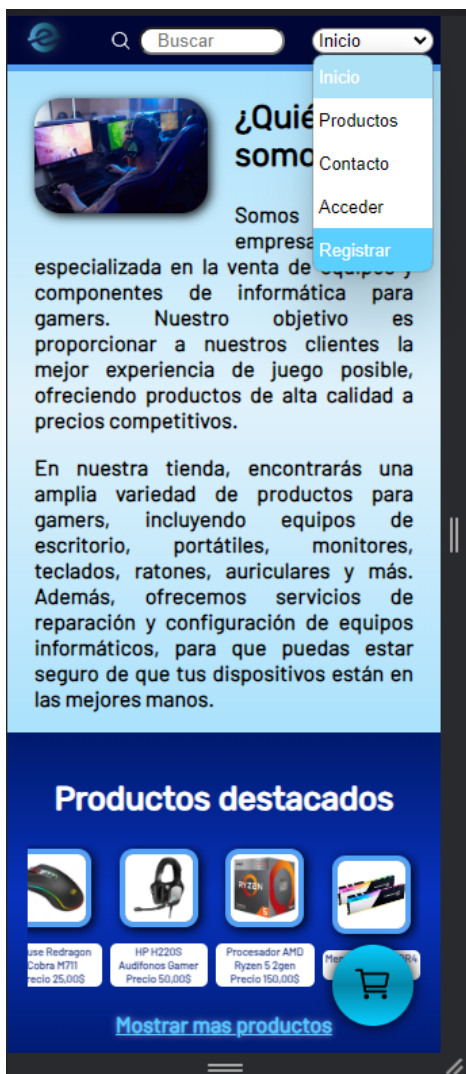




Para el diseño responsive de las páginas, se optó por las dimensiones de los dispositivos mostrados por Google, las cuales son, para móvil máximo 425px - 767px, para Tablet 768px – 767px y para portátil o PC 1440px – 767px.

En móvil, el diseño cambia, el header se vuelve un select, con las diferentes options, esto es ocurrido por la optimización del espacio, ya que los textos tan largos no caben en esas dimensiones.

El imagotipo también es cambiado por el isologotipo, para también optimizar el espacio brindado



En la parte de los productos destacados en el index se agregó una función de carrusel, la cual hace que los productos se vean al desplazarlos



En el apartado de productos, se hizo lo mismo en el header, pero en la lista de productos se muestran de máximo 2 por fila, para que no se vea saturado el espacio





Y por ultimo el diseño en Tablet es parecido al de PC y portátil, pero escalando las secciones un poco para que abarquen el espacio adecuado.

