

Interfaz Grafica

Requerimientos:

- Editor de código (Visual studio code) <https://code.visualstudio.com/>

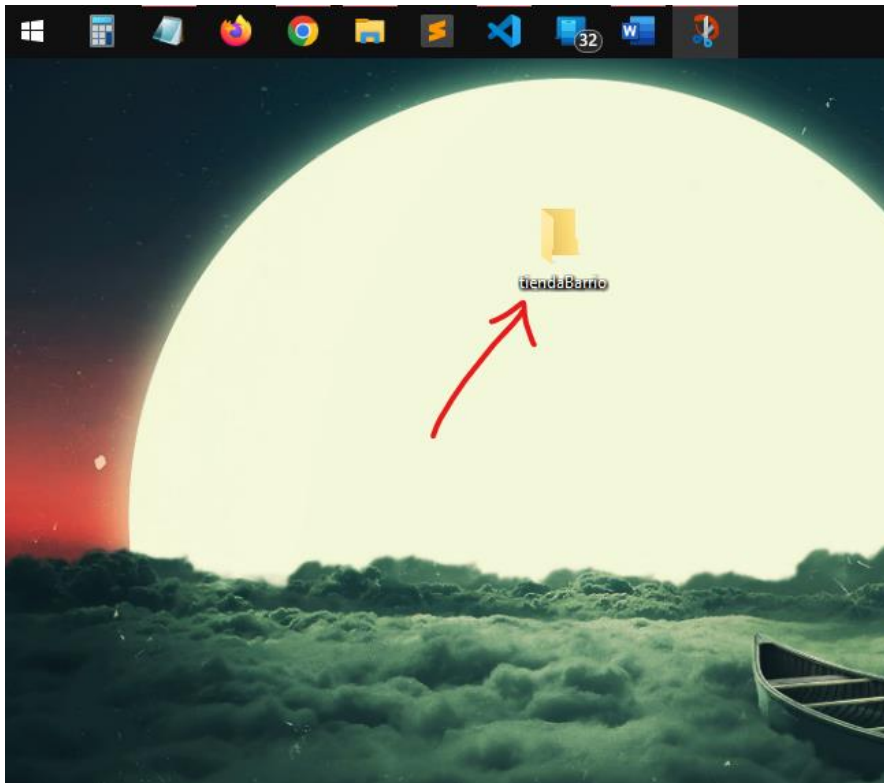
Tecnologías:

- JavaScript
- CSS
- HTML5
- Bootstrap 5
- JQuery 3
- DataTable

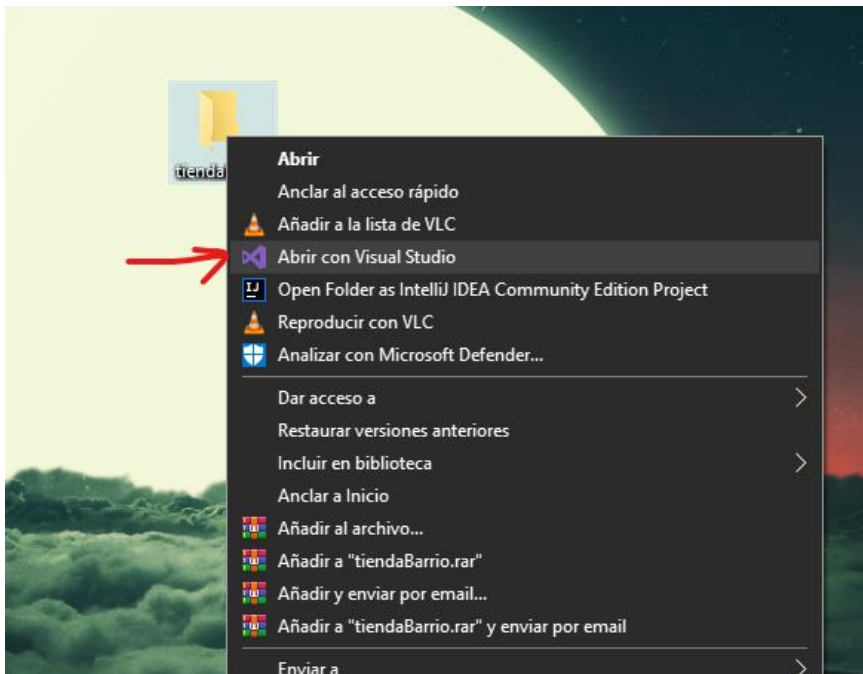
Estructura de carpetas

Primero que todo necesitamos un patrón de diseño el cual nos sirve para estructurar nuestro código, utilizaremos MVC (Modelo Vista Controlador).

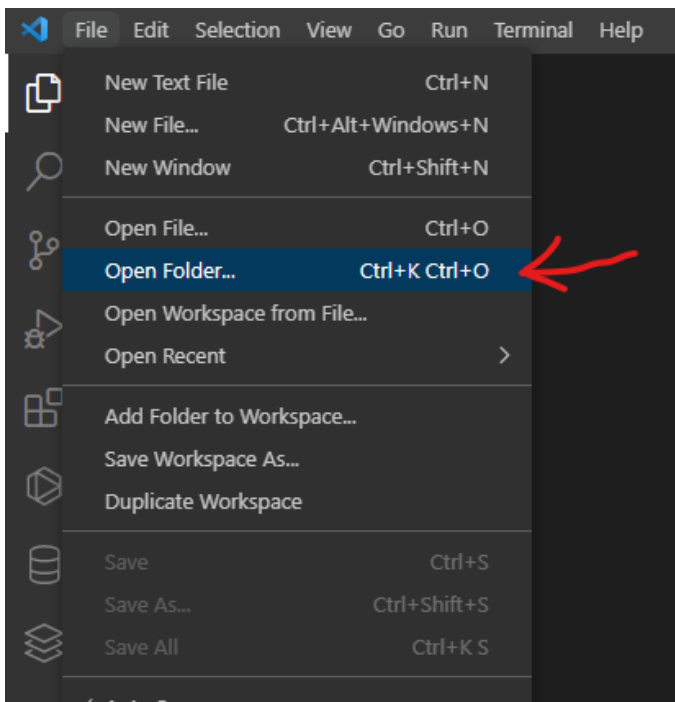
Crearemos nuestras carpetas y archivos, ubicamos la raíz de nuestro proyecto donde vamos almacenar el código. yo voy a realizar en el escritorio tu lo puedes guardar donde quieras ej:



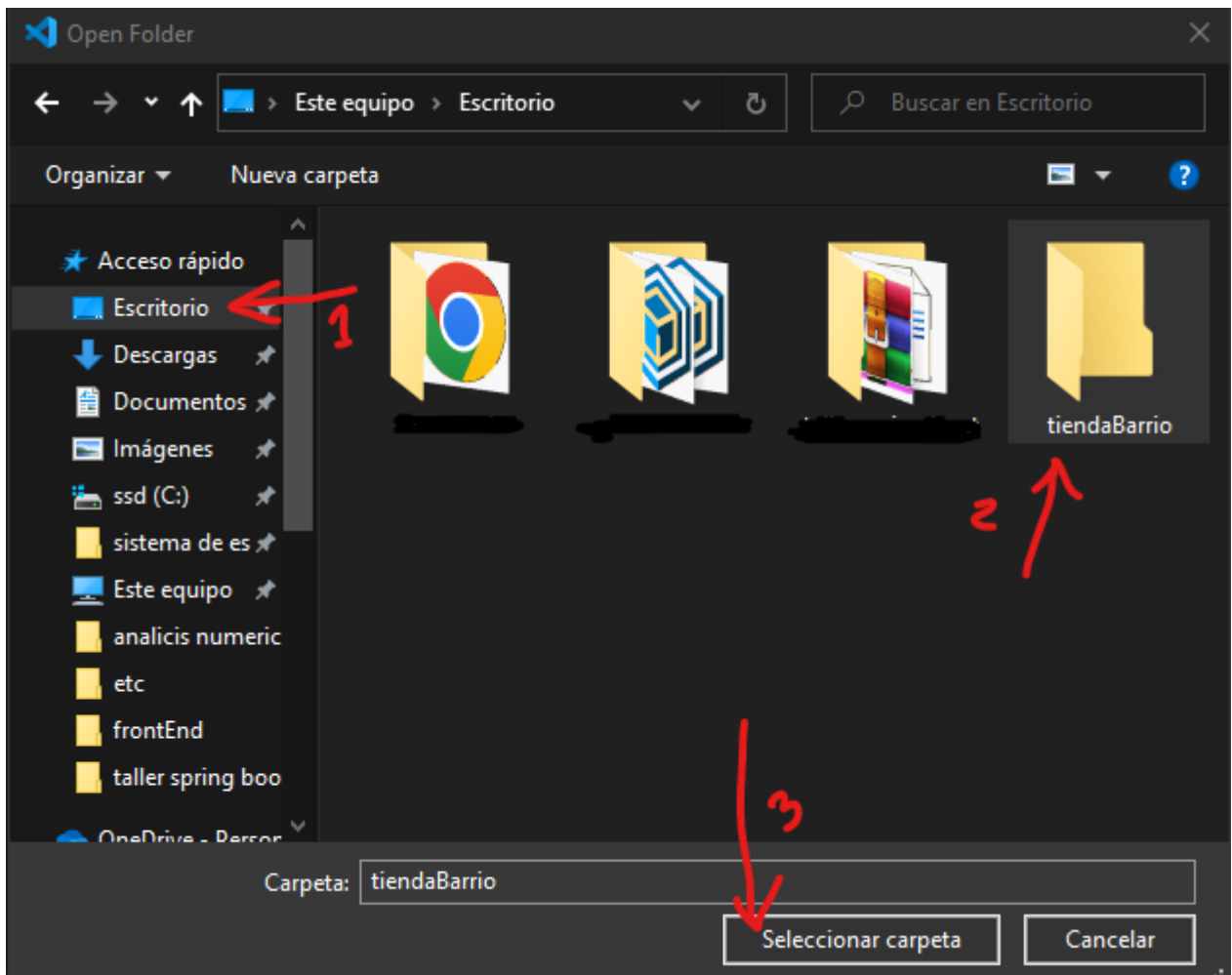
Listo ahora abrimos, le damos clic derecho y la abrimos con visual studio code ej:



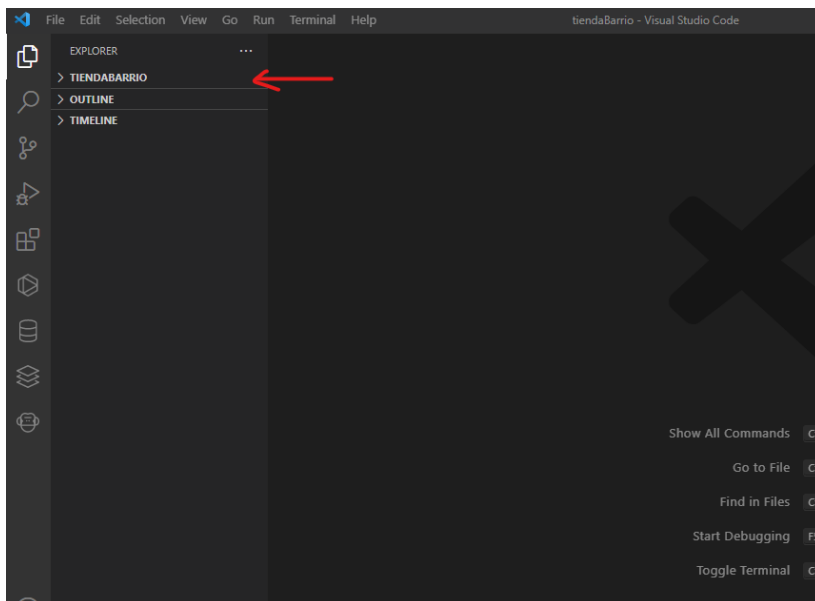
En caso tal no tengas esta opción, solo abres visual studio code normal, y le das en
File→ **abrir carpeta u Open folder** e ingresas a tu carpeta ej:



Luego seleccionas la ruta de tu proyecto, yo lo tengo en escritorio ej:



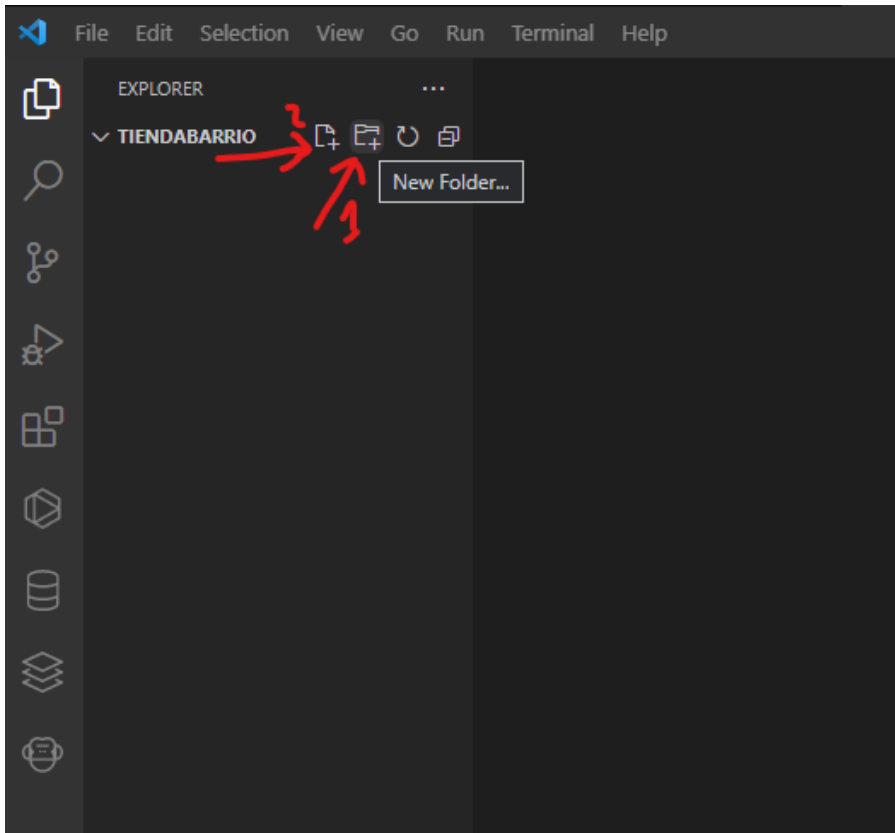
listo ahora solo queda crear nuestra estructura:



Para crear carpetas es super sencillo, le damos clic en el nombre del proyecto, Y nos aparece las opciones de crear carpeta y archivos así:

1 crear carpetas

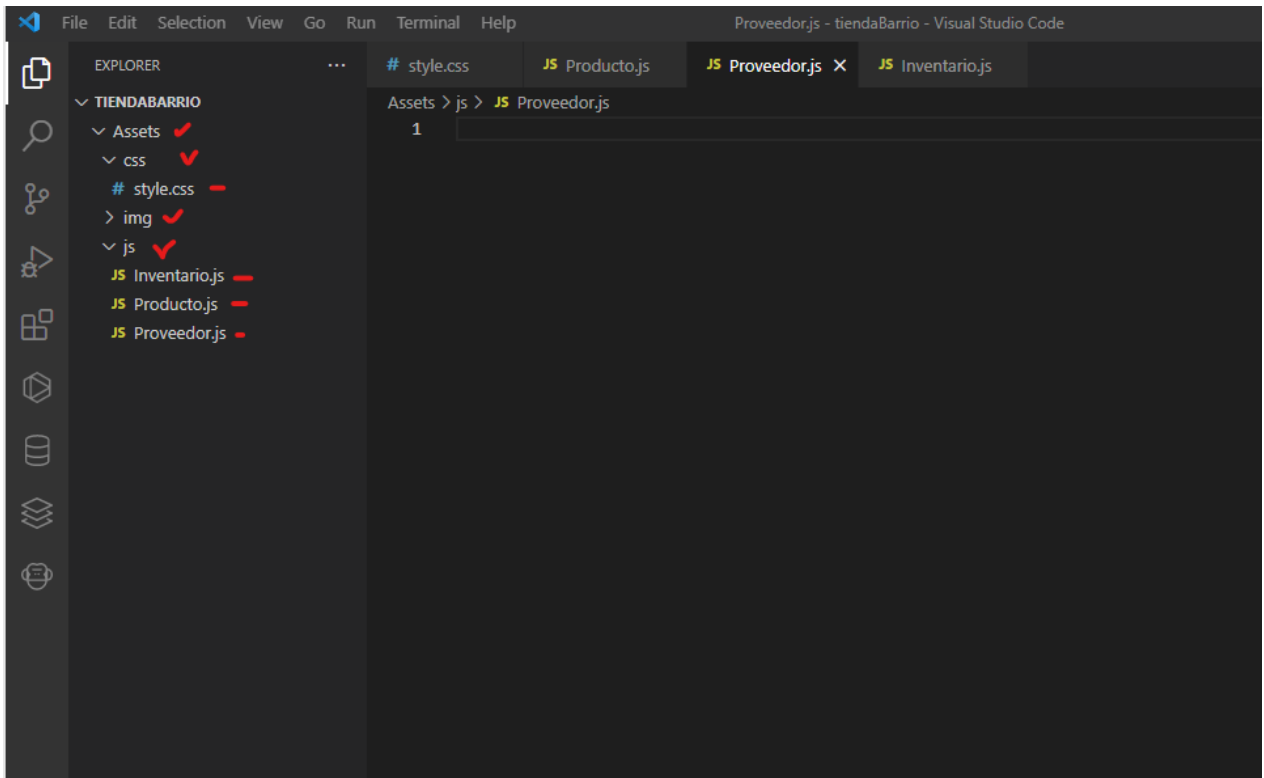
2 crear archivos



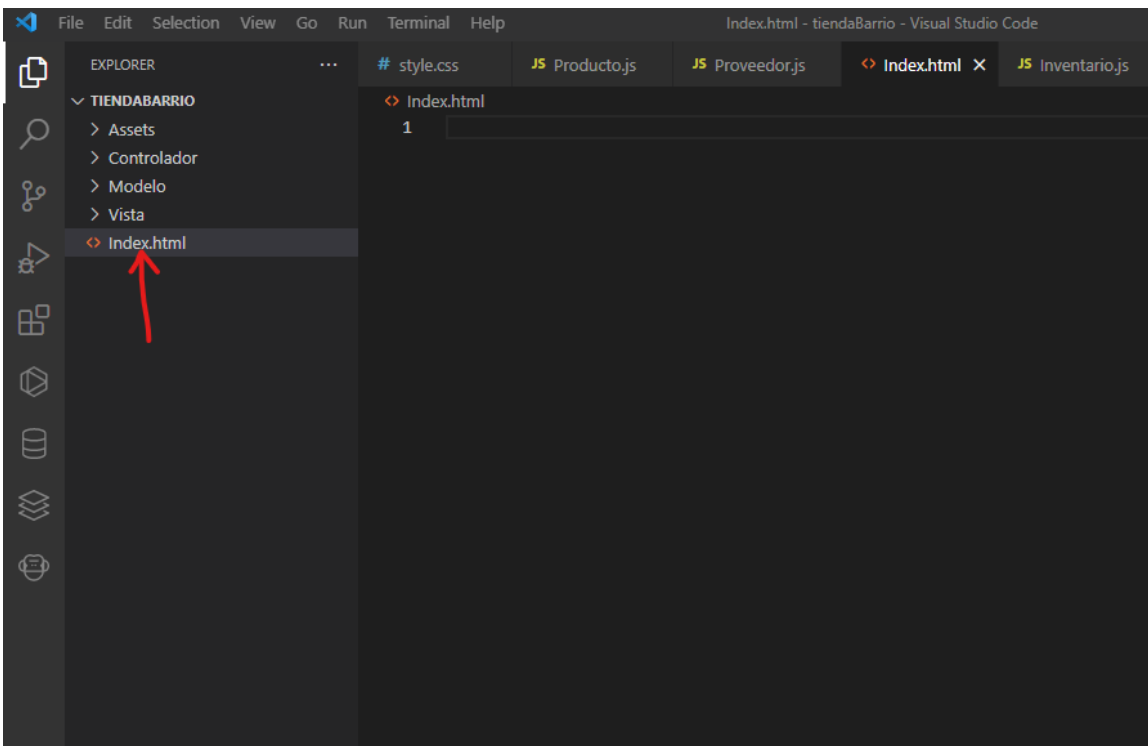
Ahora si armemos nuestro esquema, **primero necesitamos una carpeta de utilidades que nos sirva para guardar los archivos activos css, js, imágenes librería entre otros.**

Nos quedaría así la primera carpeta:

- 1) Assets/
 - a) Css/
 - style.css
 - b) Js/
 - Productos.js
 - Proveedor.js
 - Inventario.js
 - c) Img/



Listo, ahora seguimos con la estructura de modelo vista controlador, y de igual forma necesitas un archivo HTML que nos muestre el inicio de nuestra aplicación. Que este ubicado en la raíz del proyecto, necesaria e importante que se llame index.html para que se reconozca como nuestro archivo principal al iniciar la aplicación así:



Listo ya tenemos nuestro esquema completo ahora sí, iniciemos con el diseño.

Primero iniciemos con nuestro archivo principal index.html.

Utilizaremos la sintaxis de html5 que la podemos identificar al ver la antinomia del proyecto web ej:

`<!DOCTYPE html>` :

La etiqueta doctype declara el tipo de documento, por lo que esta está señalando que este es un documento HTML.

Etiquetas principales de html5

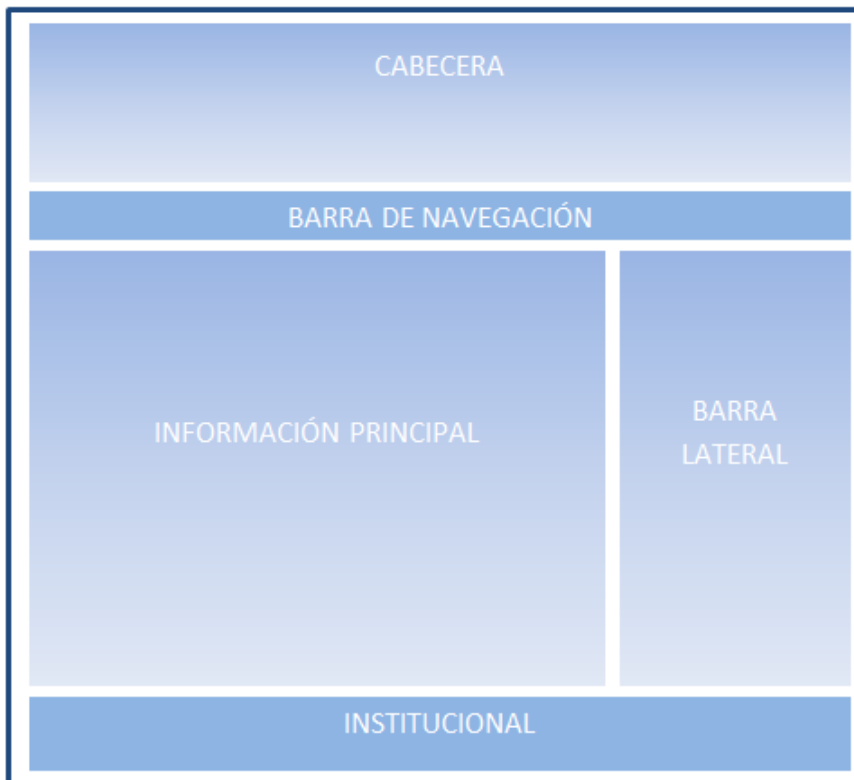
`<header> ... </header>`

`<section> ... </section>`

`<nav> ... </nav>`

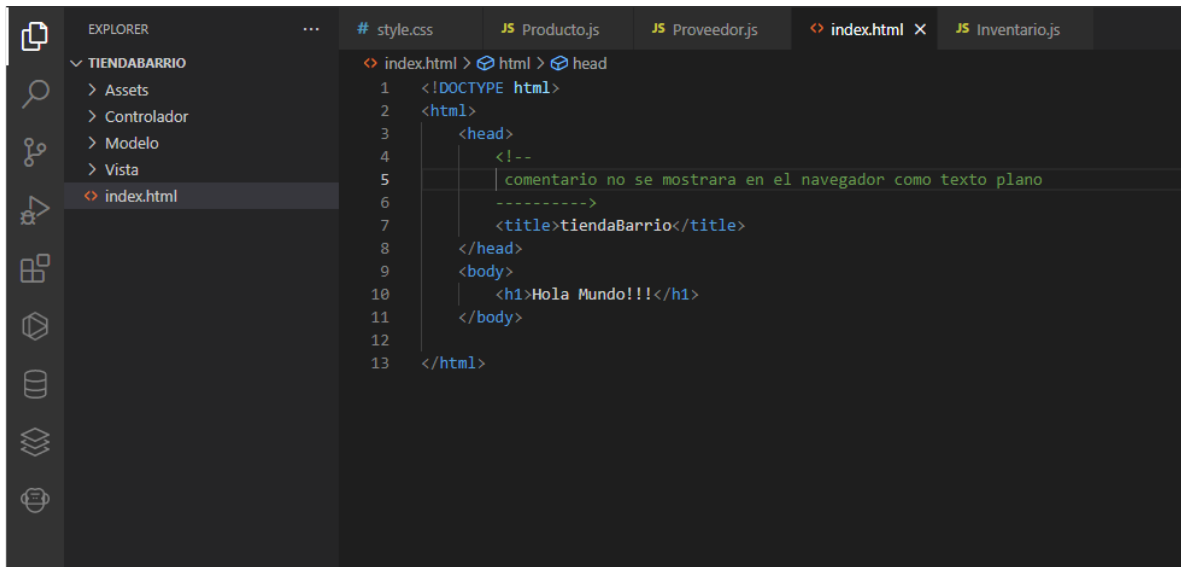
`<footer> ... </footer>`

Para conocer mas sobre las etiquetas y sus funciones aquí <https://lenguajehtml.com/html/>



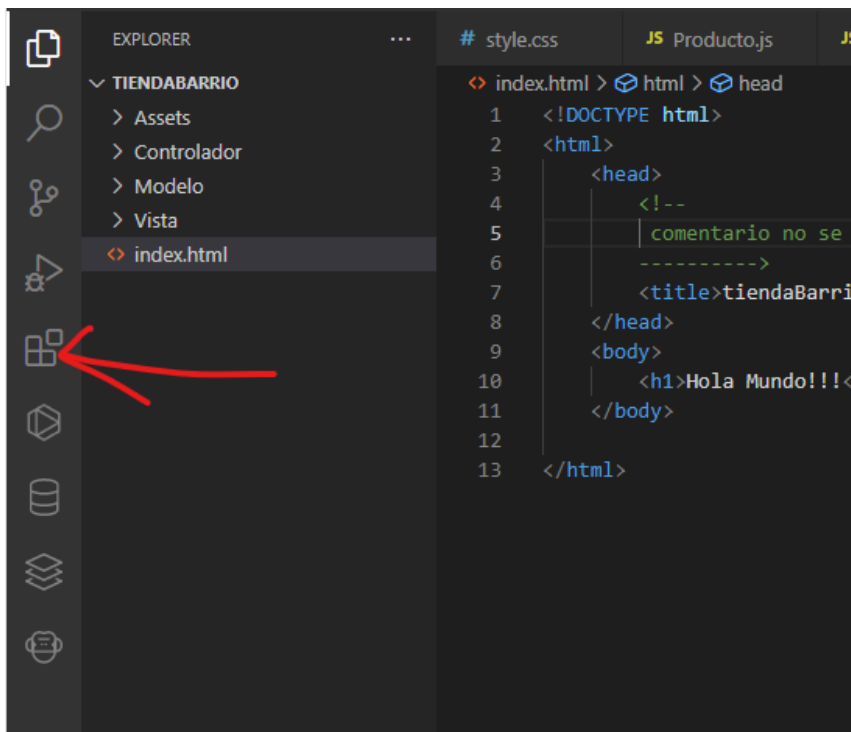
Sabiendo esto podemos iniciar:

Mostremos nuestro **hola mundo** en el navegador, para probar sus funciones

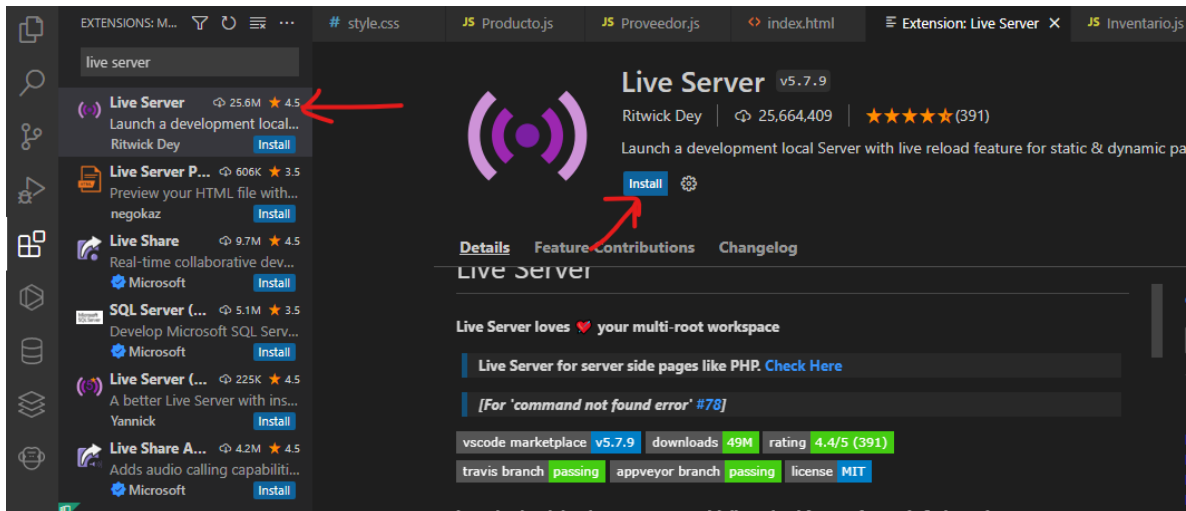


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!--
5      comentario no se mostrara en el navegador como texto plano
6     -->
7     <title>tiendaBarrio</title>
8   </head>
9   <body>
10    <h1>Hola Mundo!!!</h1>
11  </body>
12
13 </html>
```

Ahora podemos integrar un plugin que nos ayuda a visualizar en tiempo real lo que vamos programando en HTML, nos vamos a extensiones, en algunos casos puede estar del otro lado del editor.

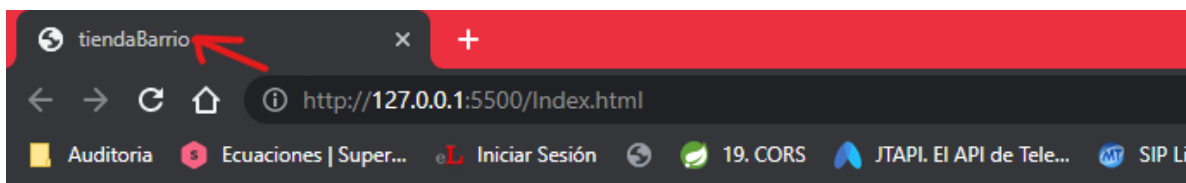


Buscamos el plugin **Live Server** y le damos instalar, esperamos que se instale y cerramos esa pestaña, Vamos a nuestro archivo HTML.



En la parte de abajo en la línea azul veremos un ítem que dice **Go Live** le damos clic y esperamos que nos muestre el navegador.

Así se vería en el navegador:



Hola Mundo!!!

Nos muestra el título de la página y el h1 que le pasamos con el texto, muy bien ahora si integremos todas las funciones. CSS, JS a ver cómo funcionan.

A este elemento le llamamos metadatos el cual nos sirve para hacer enlaces, funciones específicas que debería aceptar nuestro sistema, los principales.

- `<head> </head>`, que indican los metadatos respecto a un documento como los enlaces. Esta clase de información brindada sólo es visible para los navegadores.
- `<title> </title>`, la etiqueta que define el título de una web.
- `<link>`, utilizada para afiliar recursos que son externos a dicho documento.
- `<meta>`, utilizada para poder colocar metadatos como el autor de la web o hasta su descripción. Siempre es usada para datos que no poseen una etiqueta específica.
- `<style></style>`, funciona como medio para ingresar código CSS en el documento.

Configuraciones de la página `<head>`

- **Especifica la codificación de caracteres para el documento HTML**

- `<meta charset="UTF-8">`
- **Declarar el idioma de la página web**
`<html lang="en">`
- **Definir palabras clave para motores de búsqueda:**
`<meta name="keywords" content="tienda virtual,tiendabarrio,ventaproductos">`
- **Define una descripción de tu página web:**
`<meta name="description" content="tienda virtual de barrio">`
- **Definir el autor de una página:**
`<meta name="author" content="John Doe">`
- **Actualizar documento cada 30 segundos:**
`<meta http-equiv="refresh" content="30">`
- **Configuración de la ventana gráfica para que su sitio web se vea bien en todos los dispositivos:**
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

La `<link>` etiqueta define la relación entre el documento actual y un recurso externo.

La `<script>` etiqueta se utiliza para incrustar un script del lado del cliente (JavaScript).

Elementos de secciones

Dividir una página por secciones es más que necesario para que sean visualmente atractivas y posean una estructura que los navegadores comprendan. Es por ello que, mencionamos las versiones más comunes de etiquetas para poder crear secciones dentro de una web:

- `<body> </body>`. Incluye todo el contenido que se desee mostrar dentro de la página web.
- `<nav> </nav>`. Define el contenido que estará ubicado dentro de la sección de navegación de la página.
- `<main> </main>`. Permite elegir el contenido principal que posee un documento, siendo único.
- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`. Permiten crear una estructura para el contenido de una página web. Son esenciales para cumplir con la correcta organización de dicho contenido.
- `<section> </section>`. Indica la sección de un documento HTML.
- `<article> </article>`. Incluye el contenido de una página que es independiente.
- `<aside> </aside>`. Todo el contenido extra de una página web se incluye en estas etiquetas, pues posee poca relevancia, pero coincide con la web.
- `<footer> </footer>`. Incluye el contenido que irá en el pie de página.
- `<header> </header>`. Define la parte superior o cabecera de una web. En ella mayormente se encuentra el logotipo y nombre de la web, así como el menú.

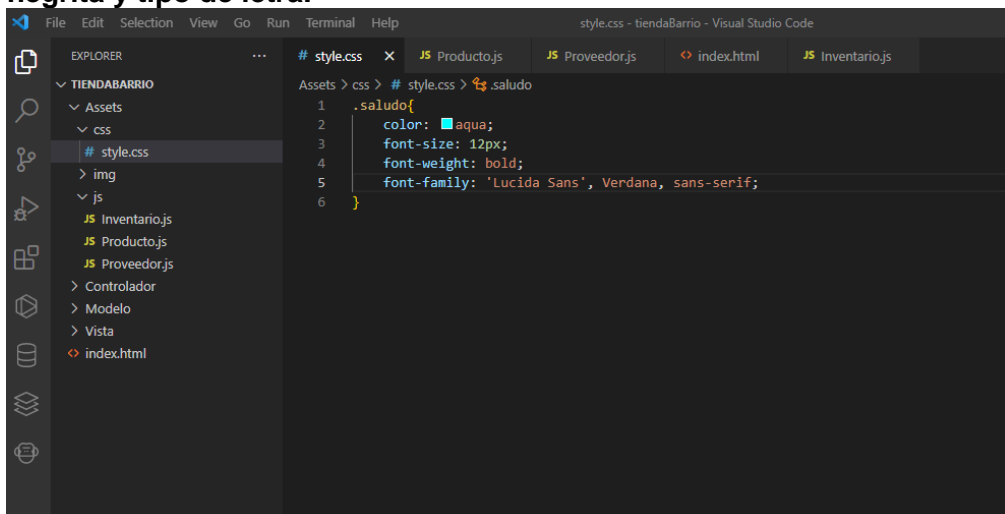
Bueno, después de repasar todo para entender un poco mejor lo que vamos a realizar

Le vamos a dar estilo a nuestro título hola mundo

Lo primero es importar o conectar nuestro archivo externo style.css y después asignar una clase a nuestro `<h1 class="saludo">` quedaría así.

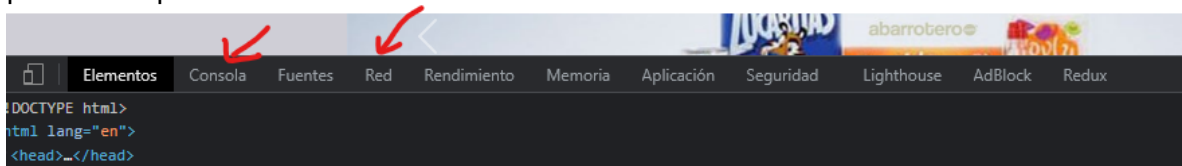
```
<> index.html > html > body > h1.saludo
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <meta name="description" content="tienda virtual de
6      <meta name="keywords" content="tienda virtual, tienda
7      <meta name="author" content="John Doe">
8      <meta name="viewport" content="width=device-width, i
9      <title>TiendaBarrio</title>
10     <link rel="stylesheet" href="Assets/css/style.css">
11   </head>
12   <body>
13     <h1 class="saludo">Hola Mundo!!!</h1>
14   </body>
15
16 </html>
```

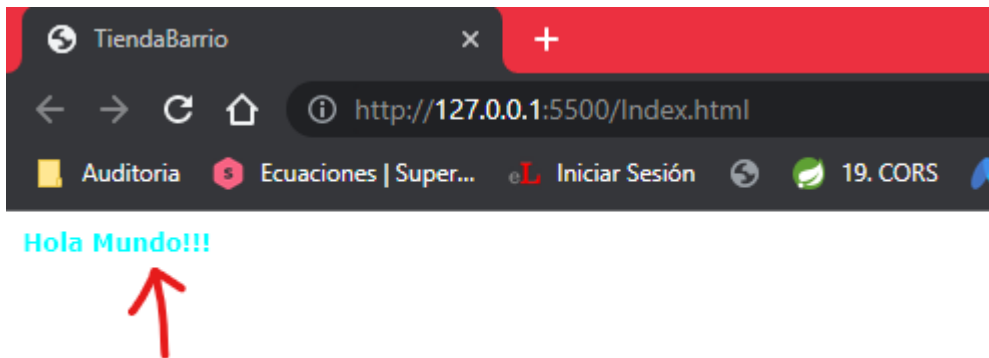
Luego abrimos nuestro archivo css para dar el estilo que necesitamos.
Aquí en este link encontraras todas las sintaxis de html y css que te ayudaran mucho:
<https://www.w3schools.com/css/default.asp>
así quedara nuestro CSS con, los estilos asignados que son **color del texto**, **tamaño**, **negrita** y **tipo de letra**.



```
Assets > css > # style.css > .saludo
1  .saludo{
2    color: aqua;
3    font-size: 12px;
4    font-weight: bold;
5    font-family: 'Lucida Sans', Verdana, sans-serif;
6  }
```

Ahora guardemos y miremos nuestro navegador a ver si se aplicaron los cambios.
De Igual forma nos podemos apoyar de la consola del navegador nos muestra error y las
peticiones que realizaremos mas adelante.





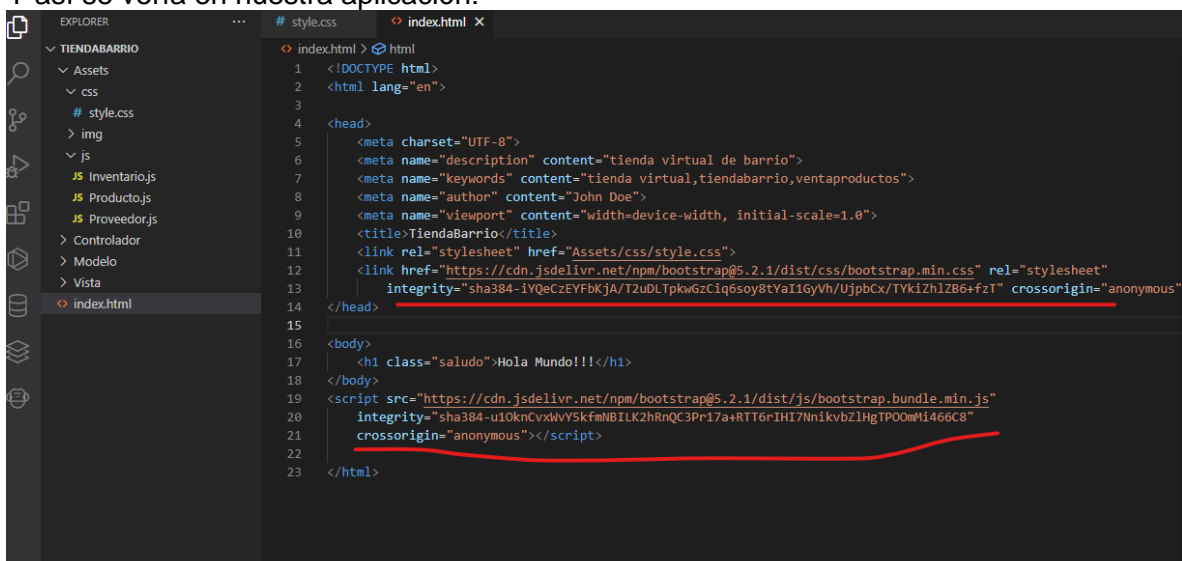
Muy bien, ya hicimos nuestro primer enlace css y sabemos como funciona ahora para evitarnos esto **utilizaremos un framework css** que nos facilitara el uso de etiquetas y diseños mas complejos y para una mejor experiencia de usuario. Primero es integrar sus cdn o Link de css , vamos a su pagina oficial.

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

aquí encontraremos ejemplo muy prácticos de la integración del framework.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  </body>
</html>
```

Y así se vería en nuestra aplicación:



Listo probemos si es real o no. Volvamos a su documentación y busquemos Nav, para que nos muestres las barras de navegación disponibles, con estilos y todo.

Grupo de botones

Tarjeta

Carrusel

botón cerrar

Colapsar

Listas desplegables

Grupo de lista

Modal

barra de navegación

Navegaciones y pestañas

Offcanvas

Paginación

Marcadores de posición

Popovers

Progreso

Desplazamiento

Hilanderos

toctareas

</nav>

También puede usar menús desplegables en su barra de navegación. Los menús desplegables requieren un elemento envolvente para el posicionamiento, así que asegúrese de usar elementos separados y anidados para `.nav-item` `.nav-link` como se muestra a continuación.

barra de navegación

Hogar

Características

Precios

enlace desplegable ▾

HTML

```
<nav class="navbar navbar-expand-lg bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
```

Elijamos esta, copiemos el código debajo y pequemos en nuestro código. Quedaría de esta forma.

EXPLORER

TIENDABARRIO

Assets

css

style.css

img

js

Inventario.js

Producto.js

Proveedor.js

Controlador

Modelo

Vista

index.html

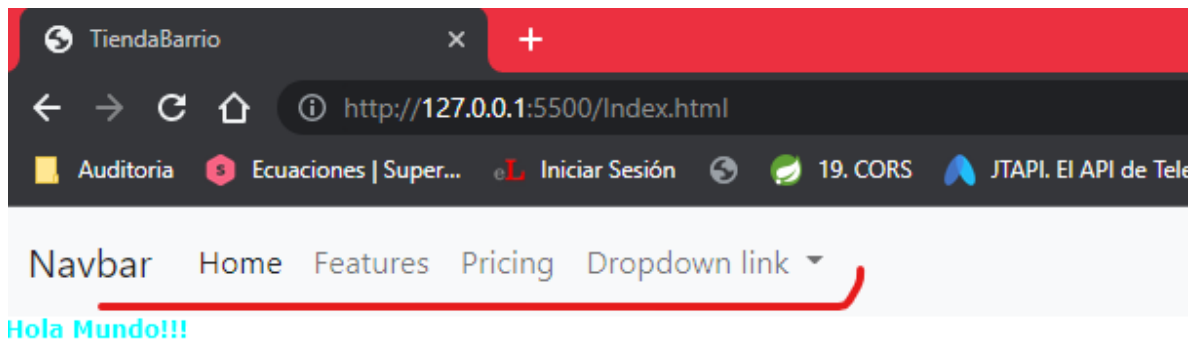
style.css

index.html X

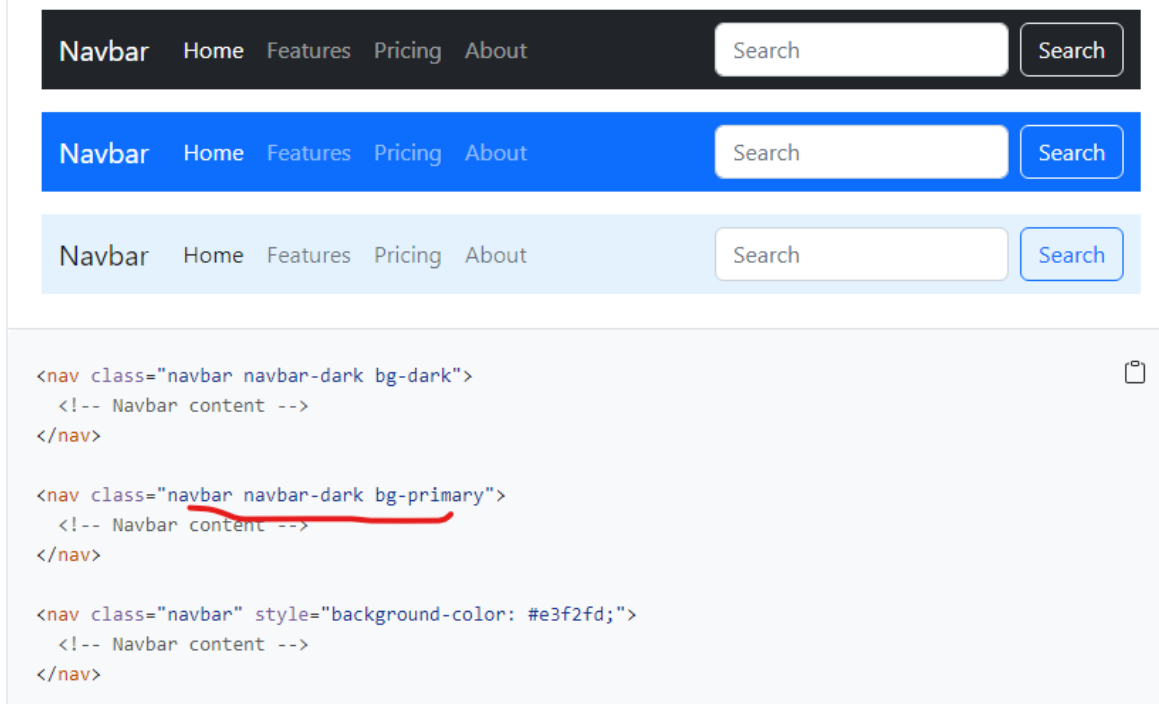
index.html > html > body > nav.navbar.navbar-expand-lg.bg-light

```
2 <html lang="en">
4 <head>
11 <link rel="stylesheet" href="Assets/css/style.css">
12 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
13 integrity="sha384-1YQecZEVFFbKjA/T2uDLTpkwGzCiq6soy8TVaI16yVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="
14 </head>
16 <body>
17 <nav class="navbar navbar-expand-lg bg-light">
18 <div class="container-fluid">
19 <a class="navbar-brand" href="#">Navbar</a>
20 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbar">
21 <span class="navbar-toggler-icon"></span>
22 </button>
23 <div class="collapse navbar-collapse" id="navbarNavDropdown">
24 <ul class="navbar-nav">
25 <li class="nav-item">
26 <a class="nav-link active" aria-current="page" href="#">Home</a>
27 </li>
28 <li class="nav-item">
29 <a class="nav-link" href="#">Features</a>
30 </li>
31 <li class="nav-item">
32 <a class="nav-link" href="#">Pricing</a>
33 </li>
34 <li class="nav-item dropdown">
35 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria
36 | Dropdown link
37 </a>
38 <ul class="dropdown-menu">
39 <li><a class="dropdown-item" href="#">Action</a></li>
40 <li><a class="dropdown-item" href="#">Another action</a></li>
41 <li><a class="dropdown-item" href="#">Something else here</a></li>
42 </ul>
43 </li>
44 </ul>
45 </div>
46 </div>
47 </nav>
48 <h1 class="saludo">Hola Mundo!!!</h1>
49 </body>
50 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
51 integrity="sha384-u10knCvxlvV5KfmbNBLK2hRnQC3Pr17a+RTT6rIHI7NnikvBz1HgTP0OmMi466C8"
52 crossorigin="anonymous"></script>
53
```

Guardamos y vamos a nuestro navegador a ver como quedo.



Perfecto, vamos bien ahora cambiemos ese color de navegador vamos de nuevo a Bootstrap y busquemos Color **schemes o esquema de colores**. Copiemos el estilo que tiene la clase de color azul o el negro si prefieres. Solo copiemos nav-dark bg-primary es la clase del color y el texto.



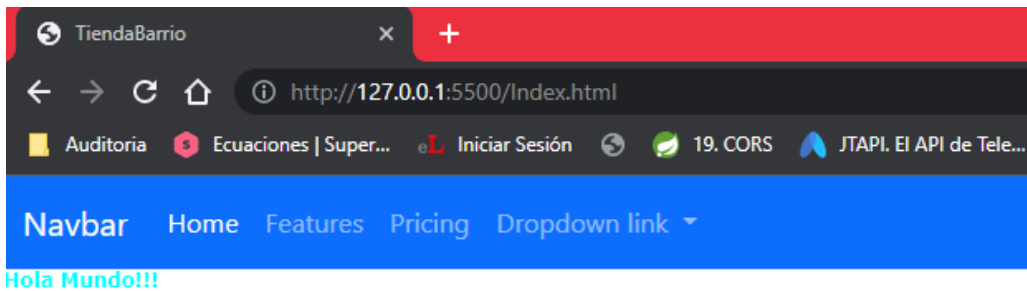
Asi se veria en nuestro codigo

```

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavDropdown">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

```

Vamos al navegador y verificamos. Que todo este bien.



Listo todo está bien, personalizemos los botones, cambiemos los nombres y eliminemos lo que no necesitamos.

Necesitamos el siguiente botones.

Menú principal, producto, proveedor, inventario

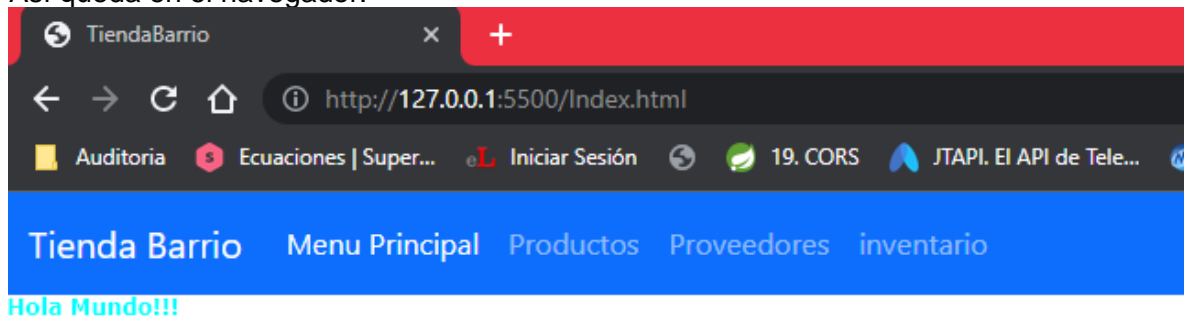
Así quedaría en nuestro código:

```

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Tienda Barrio</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavDropdown">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Menu Principal</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Productos</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Proveedores</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">inventario</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <h1 class="saludo">Hola Mundo!!!</h1>
</body>

```

Así queda en el navegador:



Vamos bien, pongamos unas imágenes de nuestros productos en el menú principal para que sea vea mejor. Vamos a Bootstrap y busquemos, **container-fluid** para crear un contenedor central en la página.

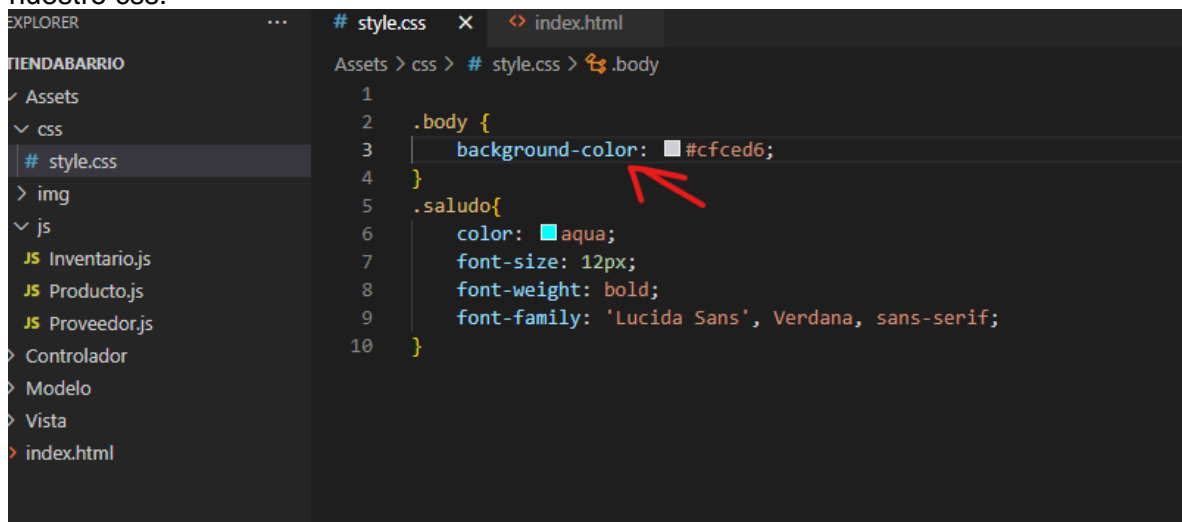
Este será el contenedor de todo el proyecto lo copiamos y agregamos todo el código html dentro.

Fluid containers

Use `.container-fluid` for a full width container, spanning the entire width of the viewport.

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

Luego agregamos una clase al body de nombre body para dar color al fondo desde nuestro css.



Podemos eliminar el css `.saludo`

Ahora creamos la etiqueta `<header>` y dentro metemos todo el `<Nav>` así:

```

<section>
  <div class="contenedor-img">
    <div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
      <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" 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="#carouselExampleControls" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
      </button>
    </div>
  </div>
</section>

```

Luego, creamos una etiqueta de `<section>` y un `<div>` con la **clase contenedor-img** y metemos todo el código **Carousel**. En **Bootstrap buscamos Carousel y Copiemos su código y pegamos en nuestro proyecto**. Cuando ya tengamos el código necesitaremos tres imágenes para la muestra

Puedes poner las que quieras, pero esta vez solo usaremos tres.

En el archivo css , ponemos las clase . contenedor-img

```

.body {
  background-color: #cfcfd6;
}
.contenedor-img{
  margin: auto;
  width: 60%;
  height: 60%;
}

```

Luego pasamos a nuestro **índex** y ponemos el código de **Bootstrap**.

- Accordion
- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel**
- Close button
- Collapse
- Dropdowns
- List group
- Modal
- Navbar
- Navs & tabs
- Offcanvas
- Pagination
- Placeholders
- Popovers
- Progress
- Scrollspy

With controls

Adding in the previous and next controls. We recommend using `<button>` elements, but you can also use `<a>` elements with `role="button"`.

How it works

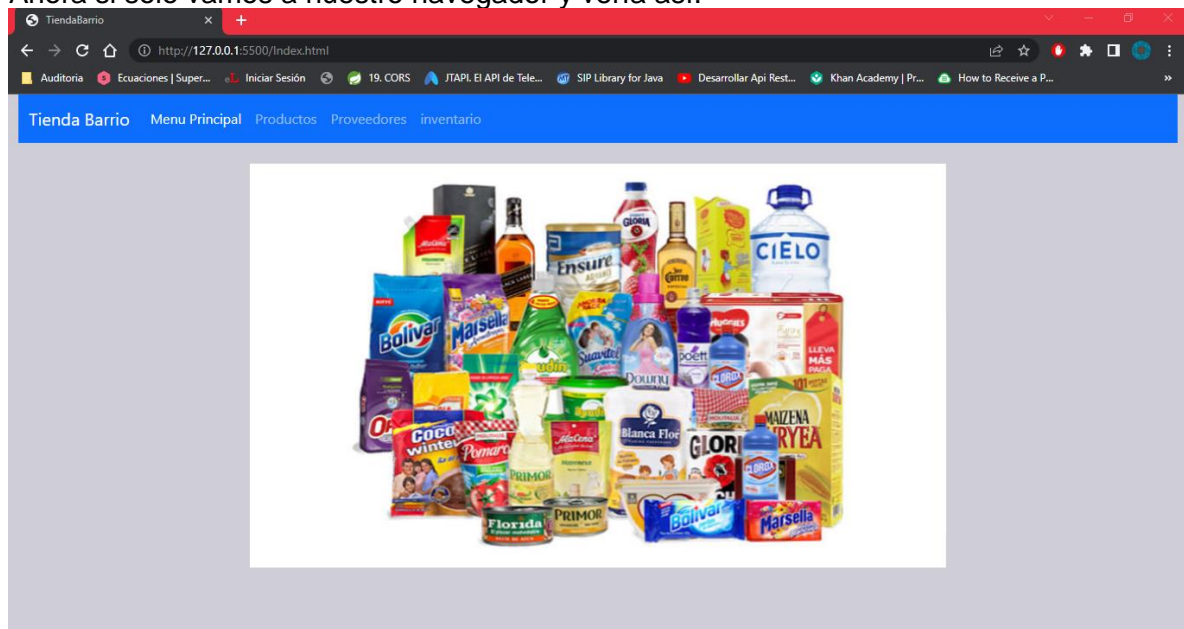
Example

- Slides only
- With controls
- With indicators
- With captions
- Crossfade
- Individual .carousel-item
- Interval
- Disable touch swipe
- Dark variant
- Custom transition
- Sass
- Variables
- Usage
- Via data attributes
- Via JavaScript
- Options
- Methods
- Events

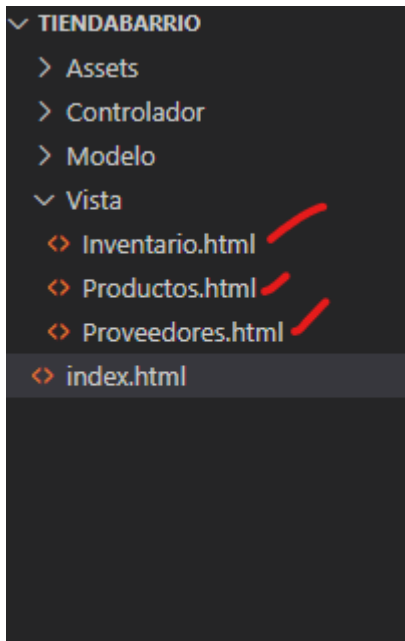
Así quería el código, podemos eliminar el h1 que creamos al inicio en index.html:

```
<section>
  <div class="container-sm">
    <div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
      <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" 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="#carouselExampleControls" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
      </button>
    </div>
  </div>
</section>
<h1 class="saludo">Hola Mundo!!!</h1>
```

Ahora si solo vamos a nuestro navegador y vería así:



Listo tenemos nuestra vista ahora crearemos las redirecciones para cada módulo, Vamos a la carpeta vistas y crearemos las vistas para cada módulo así:



Ahora miramos la redirección en nuestro código:

```
<header>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
      <a class="navbar-brand" href="/">Tienda Barrio</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expand
        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavDropdown">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="/">Menu Principal</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/Vista/Productos.html">Productos</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/Vista/Proveedores.html">Proveedores</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/Vista/Inventario.html">inventario</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>
```

En cada uno de los modulo ponemos código HTML con sus respetivos cuerpo ejemplo:
Pasamos la etiqueta <header> a cada uno de los módulos.y cambiamos sus title y cuerpo.

Vista - Producto

```
<html lang="en">
<head>
  <meta name="description" content="tienda virtual,tiendabarrio,ventaproductos">
  <meta name="keywords" content="tienda virtual,tiendabarrio,ventaproductos">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tienda - Productos</title>
  <link rel="stylesheet" href="Assets/css/style.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1YQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tVaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="
  </head>
<body class="body">
  <div class="container-fluid">
    <header>
      <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <div class="container-fluid">
          <a class="navbar-brand" href="/">Tienda Barrio</a>
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
            data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expand
            aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNavDropdown">
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/">Menu Principal</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Productos.html">Productos</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Proveedores.html">Proveedores</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Inventario.html">Inventario</a>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </header>
    <h1>Productos</h1>
  </div>
</body>
```

Vista - Proveedores

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="tienda virtual de barrio">
  <meta name="keywords" content="tienda virtual,tiendabarrio,ventaproductos">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tienda - Proveedores</title>
  <link rel="stylesheet" href="Assets/css/style.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-iYQeCzEYFbkjA/T2uDLTpkwGzCiq6soy8tVaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="
</head>
<body class="body">
  <div class="container-fluid">
    <header>
      <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <div class="container-fluid">
          <a class="navbar-brand" href="/">Tienda Barrio</a>
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
            data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false"
            aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNavDropdown">
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/">Menu Principal</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Productos.html">Productos</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Proveedores.html">Proveedores</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/Vista/Inventario.html">Inventario</a>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </header>
    <h1>Proveedores</h1>
  </div>

```

Ahora podemos ir al navegador y presionar cualquier botón y nos llevara a cada módulo. Ahora bien, podemos iniciar con el módulo productos el cual recibe una lista de productos, donde podamos crear, eliminar y actualizar un producto.

Lo principal es ir a Bootstrap y buscar **Table** la tabla que va mostrar los productos y encima un botón para ingresar los productos.

Esta seria nuestra tabla, copiamos el código lo ponemos de las etiquetas <sección>

Table group dividers

Add a thicker border, darker between table groups—`<thead>`, `<tbody>`, and `<tfoot>`—with `.table-group-divider`. Customize the color by changing the `border-top-color` (which we don't currently provide a utility class for at this time).

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

HTML

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
```

En la etiqueta `<section>` agregamos una clase **content**, para dar posición a nuestra tabla, luego vamos al archivo css y ponemos lo siguiente, en caso no actualice verificar las rutas o enlaces css.

```
.body {
  background-color: #cfcfd6;
}
.contenedor-img{
  margin: auto;
  width: 60%;
  height: 60%;
}
.site-footer {
  text-align: center;
}
.content{
  margin: auto;
  width: 80%;
  height: 80%;
}
```

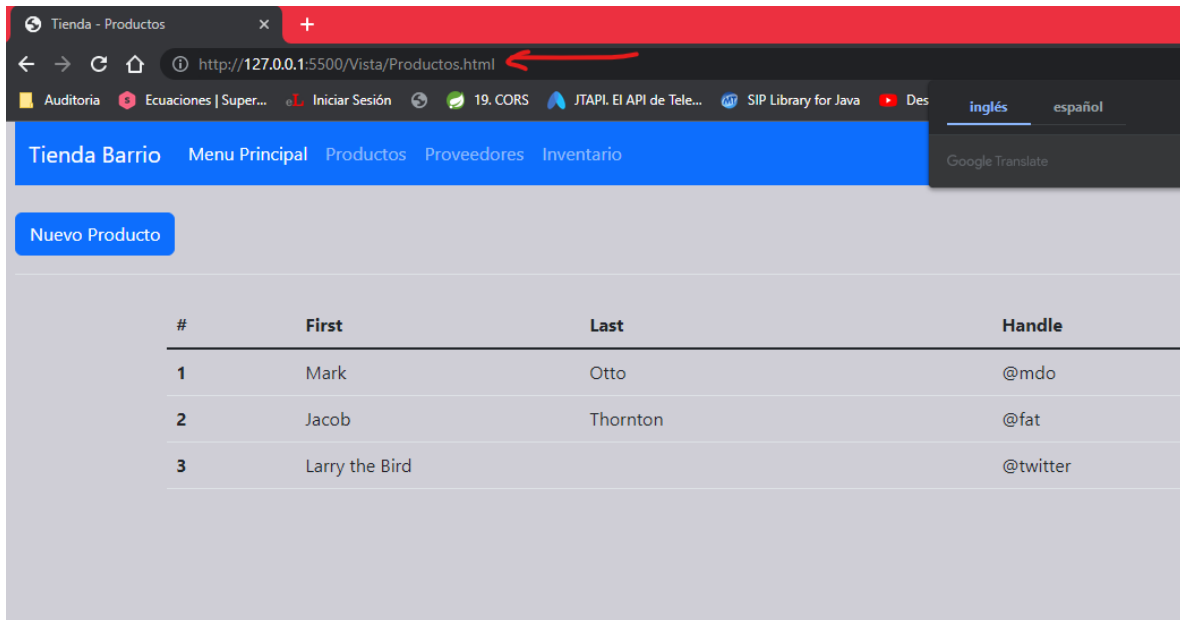
Así sería nuestro código de la tabla en productos:

```

<header class="pb-3 mb-4 border-bottom">
  <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
    <a href="/" class="btn btn-primary">Nuevo Producto</a>
  </a>
</header>
<section class="content">
  <table class="table">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">First</th>
        <th scope="col">Last</th>
        <th scope="col">Handle</th>
      </tr>
    </thead>
    <tbody class="table-group-divider">
      <tr>
        <th scope="row">1</th>
        <td>Mark</td>
        <td>Otto</td>
        <td>@mdo</td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td>Jacob</td>
        <td>Thornton</td>
        <td>@fat</td>
      </tr>
      <tr>
        <th scope="row">3</th>
        <td colspan="2">Larry the Bird</td>
        <td>@twitter</td>
      </tr>
    </tbody>
  </table>
</section>

```

En el navegador:



Entonces tenemos listos nuestros módulos, ahora sigue traer los datos de nuestro api y pintarlos en nuestra tabla productos.

Para esto utilizaremos **javascript** y dos librerías llamadas **DataTable** y **JQuery**

Aquí te dejo la documentación si quieres conocer más sobre esta librería.

<https://datatables.net/>

Lo primero es importar las librerías a nuestro proyecto lo primero es ir a su pagina oficial y ubicar sus librerías. Seguimos los pasos primero guardamos el **css** luego el **js** listo podemos usar sus funciones.

1 - Include these two files ↴

CSS `//cdn.datatables.net/1.12.1/css/jquery.dataTables`

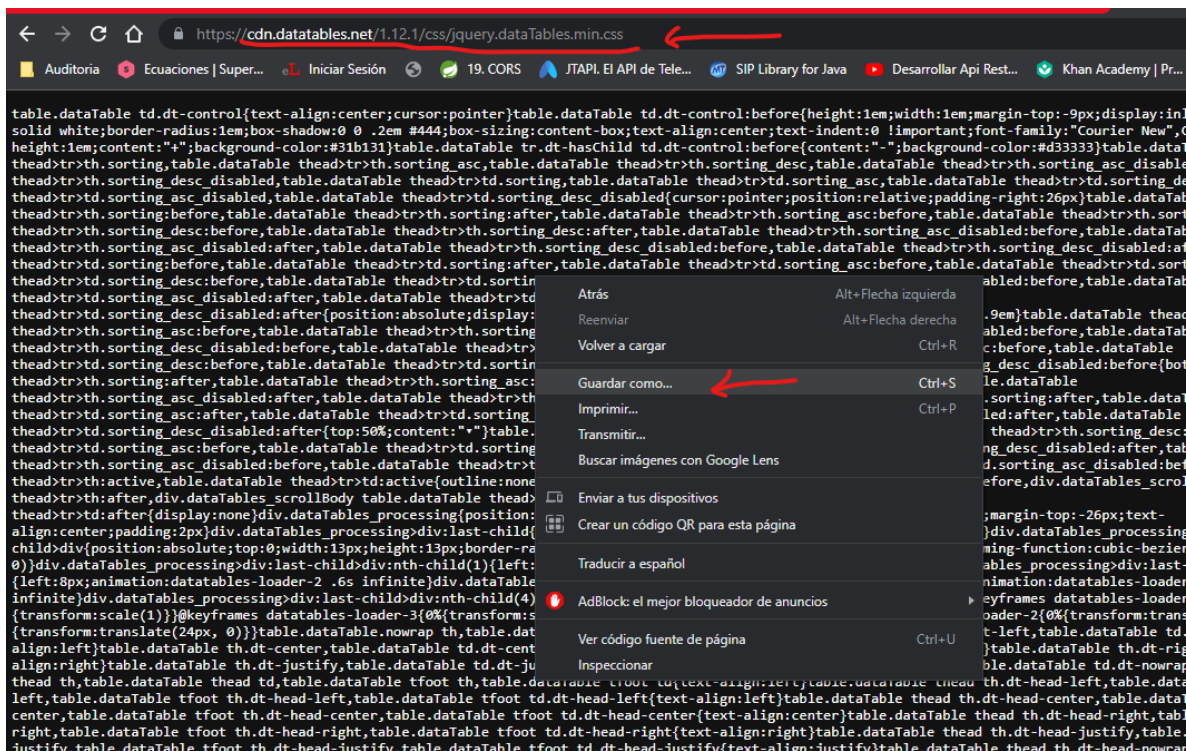
JS `//cdn.datatables.net/1.12.1/js/jquery.dataTables`

2 - Call this single function ↴

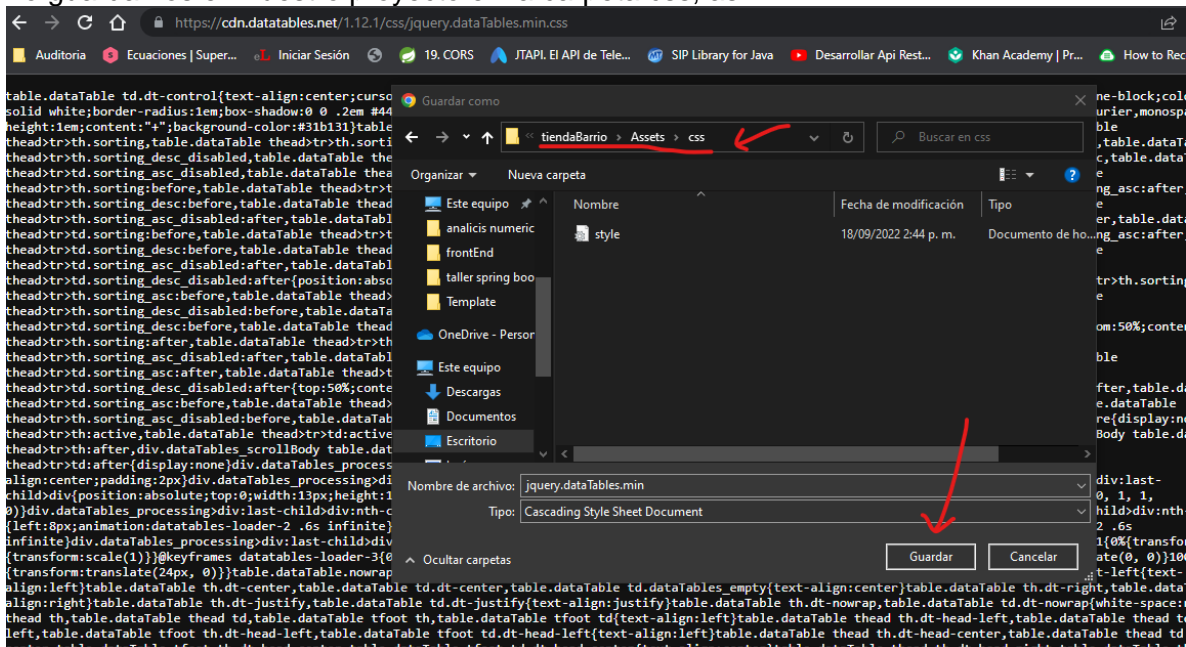
```
1 $(document).ready( function () {  
2     $('#myTable').DataTable();  
3 } );
```

3 - You get a fully interactive table →

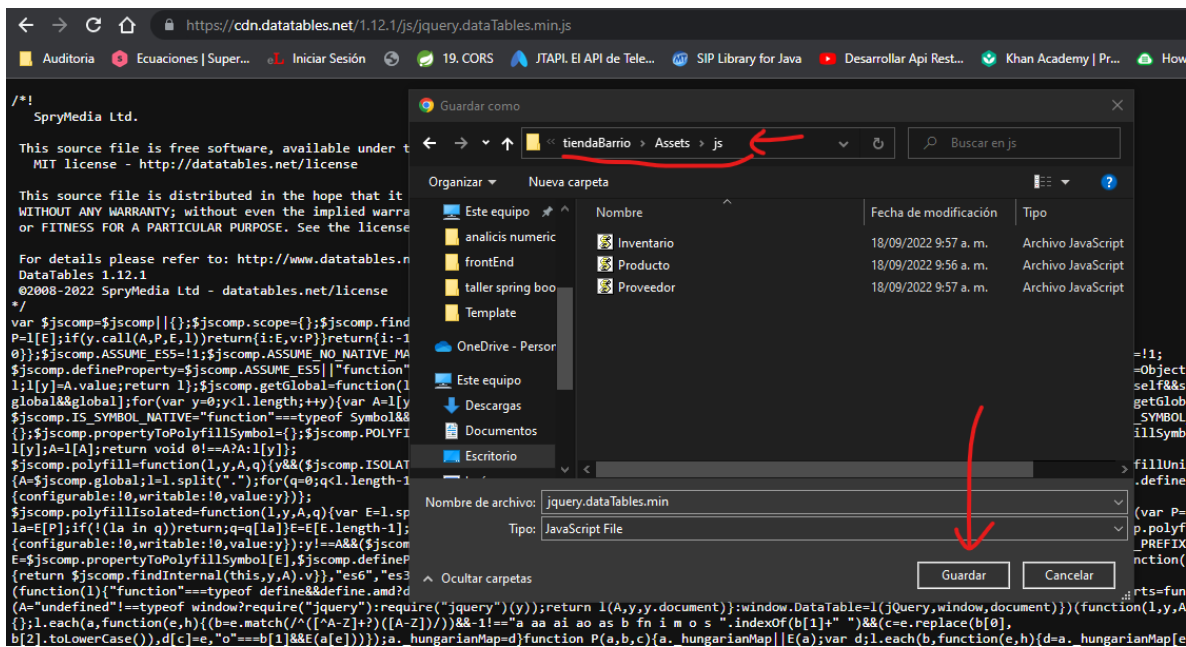
Como lo hacemos copia el link que esta en el css y lo pegamos en el navegador nos aparece la librería le damos guardar como y lo ubicamos en nuestro proyecto de esta forma:



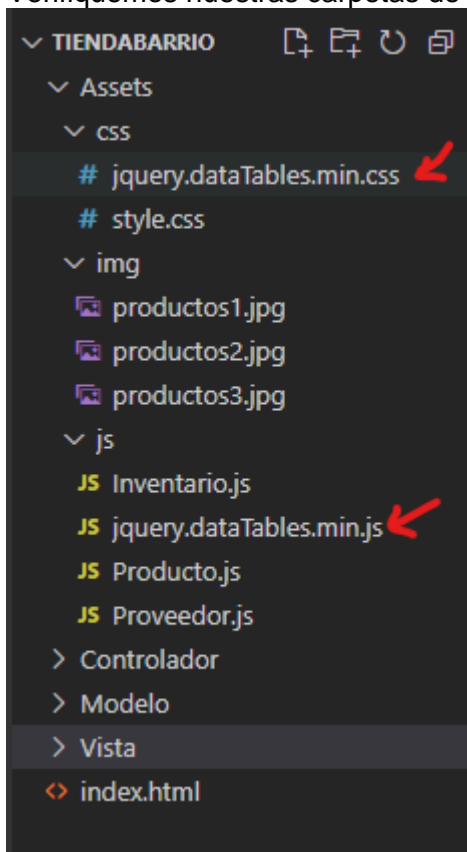
Lo guardamos en nuestro proyecto en la carpeta css, así:



Listo hacemos el mismo proceso para el siguiente archivo js



Verifiquemos nuestras carpetas de archivos, tendríamos algo parecido.



Bien ahora ya podemos trabajar con **dataTable**

Importemos su librería en las tres vistas donde usaremos las tablas, **producto.html**, **proveedor.html**, **inventario.html** así.

Datatable.css

```

<head>
  <meta charset="UTF-8">
  <meta name="description" content="tienda virtual de barrio">
  <meta name="keywords" content="tienda virtual,tiendabarrio,ventaproductos">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tienda - Productos</title>
  <link rel="stylesheet" href="/Assets/css/style.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="sty
    integrity="sha384-iYQeCzEYFbkjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" cross
  <link rel="stylesheet" href="/Assets/css/jquery.dataTables.min.css">
</head>
<body class="body">

```

Datatable.js se pone en la parte de abajo del archivo

```

</nav>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxxwvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
</body>

```

Listo solo falta importar la librería de JQuery la podemos encontrar en <https://releases.jquery.com/> le das clic en **uncompressed** y copias el cdn



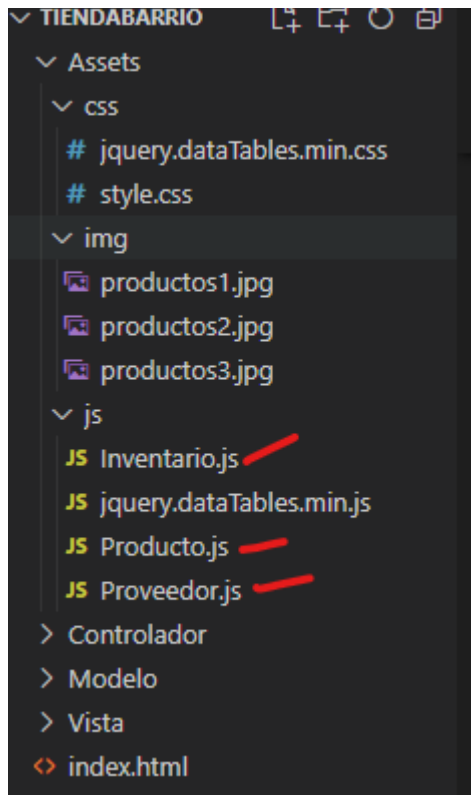
así se vería nuestro código

```

</nav>
<script src="https://code.jquery.com/jquery-3.6.1.js"
  integrity="sha256-3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFxKc2vC/7pNI=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxxwvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
<script src="/Assets/js/Producto.js"></script>
</body>

```

Tenemos listo, probemos que esté funcionando correctamente, Enlacemos el archivo js que creamos para producto proveedor, inventario. **cada un en su archivo html**. Iniciemos con producto



Así quedaría la integración de nuestro archivo js a producto.html

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxxW5kfMNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
<script src="/Assets/js/Producto.js"></script>
</body>
```

Lo mismo para proveedor e inventario

```
</body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxxW5kfMNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
<script src="/Assets/js/Proveedor.js"></script>
</html>
```

Inventario

```
</body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxxW5kfMNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
<script src="/Assets/js/Inventario.js"></script>
</html>
```

Una vez tengamos ya enlazado todas librerías y nuestros archivos podemos continuar probando el código.

Ahora en nuestro archivo **producto.html** le asignamos un estilo y nombre de clase y un id a nuestra tabla, de igual forma limpiamos nuestra tabla quedaría así:

```
<header class="pb-3 mb-4 border-bottom">
  <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
    <a href="/" class="btn btn-info">Nuevo Producto</a>
  </a>
</header>
<section class="content">
  <table class="table display" id="tablaProducto" style="width:100%">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Descripcion</th>
        <th scope="col">Costos</th>
        <th scope="col">Acciones</th>
      </tr>
    </thead>
    <tbody class="table-group-divider">
      <tr>
        <td>1</td>
        <td>Producto 1</td>
        <td>100</td>
        <td><a href="#">Ver</a> <a href="#">Editar</a> <a href="#">Eliminar</a>
        </td>
      </tr>
    </tbody>
  </table>
</section>
```

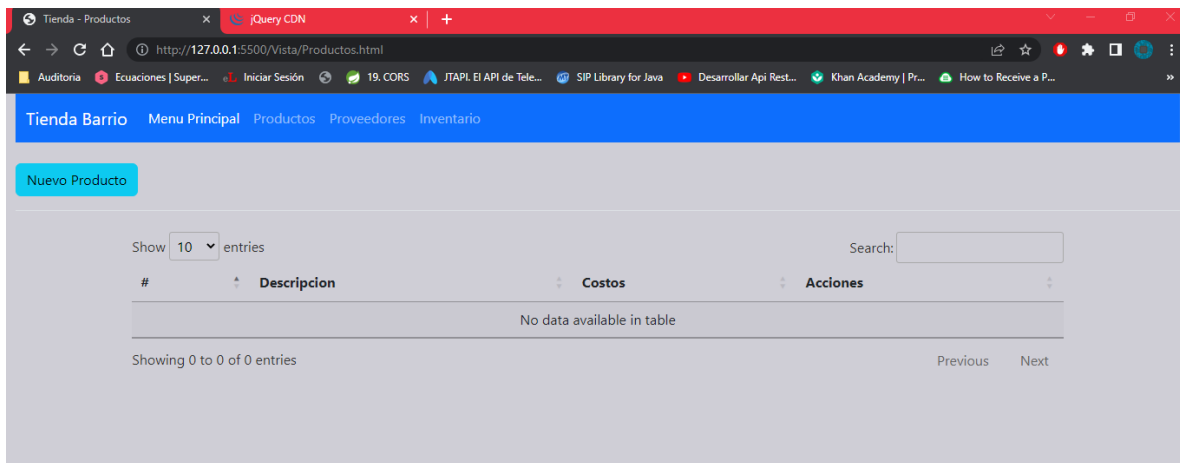
Abrimos nuestro archivo **producto.js** y ponemos el siguiente código. (JavaScript)

2 - Call this single function ↴

```
1 $(document).ready( function () {
2     $('#myTable').DataTable();
3 } );
```

Donde dice **#mytable** ponemos **#tablaProducto** listo probemos a ver qué pasa.

Así se vería nuestra página con los plugin cargados.



En nuestro archivo ponemos el siguiente código.

Agregamos un evento en el documento que cuando inicie el archivo JS llame al **datatable** y consuma nuestro api. Mas abajo se crean las dos funciones de editar y eliminar

```
# style.css  <> index.html  <> Productos.html  JS Producto.js X
Assets > js > JS Producto.js > document.addEventListener("DOMContentLoaded") callback
1 let tableProducto;
2 document.addEventListener("DOMContentLoaded", function () {
3     $.fn.dataTable.ext.errMode = 'throw'; //ocultar error del datatable
4     tableProducto = $('#tablaProducto').dataTable({
5         "Processing": true,
6         "ServerSide": true,
7         "ajax": {
8             "url": "http://localhost:8080/producto", //ruta de la api
9             "dataSrc": ""
10        },
11        "columns": [
12            { "data": "idpk" },
13            { "data": "descripcion" },
14            { "data": "costo" },
15            { "data": "accion" }
16        ],
17        columnDefs: [
18            { targets: 3, visible: true },
19            {
20                targets: -1,
21                orderable: false,
22                data: null,
23                render: function (data, type, row, meta) {
24                    let botones = `
25                    <div class="btn-group">
26                    <button onClick="EditarProdu(`+row.idpk+`);" class="btn btn-primary">Editar</button>
27                    <button onClick="eliminarProd(`+row.idpk+`);" class="btn btn-danger">Eliminar</button>
28                    </div>;
29                    return botones;
30                }
31            }
32        ],
33        'dom': 'lBfrtip',
34        "resonsieve": "true",
35        "bDestroy": true,
36        "iDisplayLength": 10,
37        "order": [[0, "asc"]]
38    });
39
40
41 }, false);
```

Si por casualidad tienes un error de **CORS** como este, no te preocupes es fácil de solucionar, ve a tu api en java en clase controller pones el siguiente código así:

@CrossOrigin(origins = {"aquí va la ruta desde donde se ejecuta de aplicación FrontEnd"})

Aquí te dejo mas información sobre los CORS. <https://docs.spring.io/spring-security/site/docs/4.2.x/reference/html/cors.html>

```

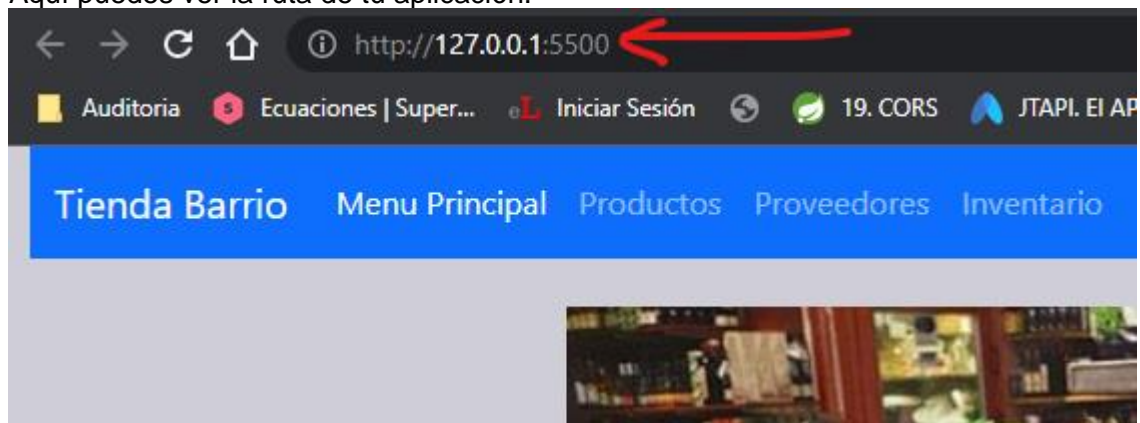
@RestController
@CrossOrigin(origins = {"http://127.0.0.1:5500"})
@RequestMapping(value = "/producto")
public class ProductoController {

    @Autowired
    private ServiceProductoImpl serviceProducto;

    @GetMapping(value = "")
    public ResponseEntity<List<Producto>> ListaProducto() {
        List<Producto> lista = serviceProducto.getListProducto();
        return ResponseEntity.ok(lista);
    }
}

```

Aquí puedes ver la ruta de tu aplicación:



Listo sigamos con el código, ahora en el código aparecen dos funciones muy importantes que son el botón de eliminar y el de editar, la cual tenemos que definir abajo estas son las funciones.

```

columnDefs: [
  { targets: 3, visible: true },
  {
    targets: -1,
    orderable: false,
    data: null,
    render: function (data, type, row, meta) {
      let botones = `
      <div class="btn-group">
        <button onClick="EditarProdu(`+row.idpk+`);" class="btn btn-primary">Editar</button>
        <button onClick="eliminarProd(`+row.idpk+`);" class="btn btn-danger">Eliminar</button>
      </div>`;
      return botones;
    }
  }
]

```

Así quedarían definidas las funciones con una alerta de prueba

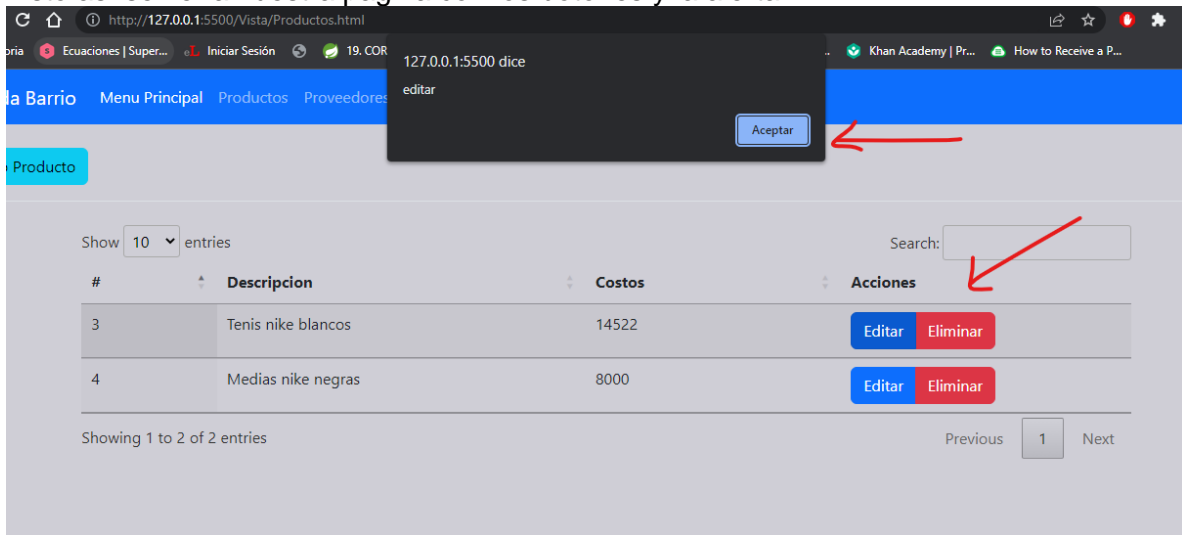
```

}, false);

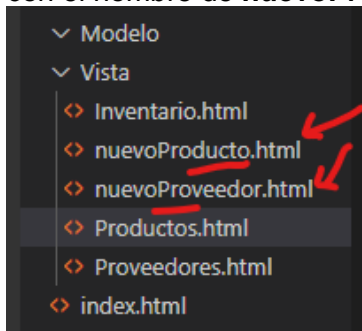
function EditarProdu(){
    alert("editar");
}
function eliminarProd(){
    alert("eliminado")
}
$("#tablaProducto").DataTable();

```

Listo así se vería nuestra pagina con los botones y la alerta.



Ahora si demos las funcionalidades a los botones, iniciemos con crear nuevo producto. Vamos a la carpeta vistas y creamos dos vistas una para proveedor y otra para producto con el nombre de **nuevoProducto** y **nuevoProveedor**.



Abrimos nuevoProducto y creamos un formulario para ingresar los nuevos productos, vamos a **Bootstrap** y podemos **Form** y seleccionamos el que mas les guste en mi caso seleccione este.

Example

Email address

Example textarea

HTML

```
<div class="mb-3">
```

Copiamos el código y lo pegamos en nuestro archivo nuevoProducto nos quedaría así. Tener en cuenta que hay que replicar el código **html de head y nav y section**. Ej:

```
# style.css index.html Productos.html nuevoProducto.html x
Vista > nuevoProducto.html > html > body.body > div.container-fluid
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="description" content="tienda virtual de barrio">
6   <meta name="keywords" content="tienda virtual,tiendabarrio,ventaproductos">
7   <meta name="author" content="John Doe">
8   <meta name="viewport" content="width=device-width, initial-scale=1.0">
9   <title>Tienda - Nuevo Producto</title>
10  <link rel="stylesheet" href="/Assets/css/style.css">
11  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
12    integrity="sha384-1YQeCzEYFbkJA/T2uDLTpkwGzCiq6soy8tYa11GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="and
13 </head>
14 <body class="body">
15   <div class="container-fluid">
16     <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
17       <div class="container-fluid">
18         <a class="navbar-brand" href="/">Tienda Barrio</a>
19         <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
20           data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="fa
21           aria-label="Toggle navigation">
22           <span class="navbar-toggler-icon"></span>
23         </button>
24         <div class="collapse navbar-collapse" id="navbarNavDropdown">
25           <ul class="navbar-nav">
26             <li class="nav-item">
27               <a class="nav-link active" aria-current="page" href="/">Menu Principal</a>
28             </li>
29             <li class="nav-item">
30               <a class="nav-link" href="/Vista/Productos.html">Productos</a>
31             </li>
32             <li class="nav-item">
33               <a class="nav-link" href="/Vista/Proveedores.html">Proveedores</a>
34             </li>
35             <li class="nav-item">
36               <a class="nav-link" href="/Vista/Inventario.html">Inventario</a>
37             </li>
38           </ul>
39         </div>
40       </div>
41     </nav>
```

Ahora si nuestro formulario: También vamos a Bootstrap y buscamos **Alerts** para mostrar unos mensajes en caso tengamos error o éxito así se vería nuestro código.


```

<form id="formProductos" >
  <div class="mb-3">
    <input type="hidden" class="form-control" name="idpk" value="0" id="idproducto" >
    <label for="textDescripcion" class="form-label">Descripcion</label>
    <input type="text" class="form-control" name="descripcion" id="textDescripcion"
      placeholder="Descripcion Producto">
    </div>
    <div class="mb-3">
      <label for="numCosto" class="form-label">Costo</label>
      <input type="number" class="form-control" id="numCosto" name="costo"
        placeholder="Costo Producto"/>
      </div>
      <button class="btn btn-primary" onclick="GuardarProducto();">Guardar</button>
    </form>

    <div class="alert alert-success" role="alert" style="display:none" id="alertSuccess">
    </div>
    <div class="alert alert-danger" role="alert" style="display:none" id="alertError">
    </div>
  </section>
</nav>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/Producto.js"></script>
</body>
</html>

```

Listo ahora que ya tenemos el formulario capturemos los datos para enviarlos a nuestra Api nos vamos a nuestro archivo producto.js ponemos el siguiente código en las funciones así se verían nuestras funciones de **guardar, eliminar y actualizar**.

Función guardar producto y actualizar

```
# style.css  index.html  Productos.html  nuevoProducto.html  JS Producto.js X
Assets > js > JS Producto.js > GuardarProducto
56 function GuardarProducto() {
57   if (document.querySelector("#formProductos")) { //validamos que exista el formulario
58     let formProductos = document.querySelector("#formProductos"); //seleccionamos el formulario
59     formProductos.onsubmit = function (e) {
60       e.preventDefault(); //evitamos que recargue la pagina al precionar el boton
61       let id = document.querySelector("#idproducto").value;
62       let strDescrip= document.querySelector("#textDescripcion").value;
63       let intCost = document.querySelector("#numCosto").value;
64
65       if (strDescrip == '' || intCost == '' ) {
66         let error = document.getElementById('alertError');
67         error.innerHTML='Todos los campos son obligatorios.'
68         error.style.display = 'flex';
69         return false;
70       }
71       let request = (window.XMLHttpRequest) ?
72         new XMLHttpRequest() :
73         new ActiveXObject('Microsoft.XMLHTTP');
74       let ajaxUrl = 'http://localhost:8080/producto'; //rutas api metodo post
75       let json = JSON.stringify({idpk:id,descripcion:strDescrip,costo:intCost}); //creamos un json
76       request.open("POST", ajaxUrl, true);
77       request.setRequestHeader("Content-Type", "application/json");
78       request.send(json);
79       request.onreadystatechange = function () {
80         if (request.status == 200) {
81           let exito = document.getElementById('alertSuccess'); //mensajes de exito
82           exito.innerHTML='Muy Bien, Se Guardo un Producto Correctamente.'
83           exito.style.display = 'flex';
84           window.location.href = "/Vista/Productos.html";
85         }else{
86           let error = document.getElementById('alertError'); //mensajes de error
87           error.innerHTML='Error !! No se puedo Guardar el Producto'
88           error.style.display = 'flex';
89           return false;
90         }
91         return false;
92       }
93     }
94   }
95 }
96 function EditarProdu(id,descripcion,costo){
97   window.location.href = "/Vista/nuevoProducto.html?id="+id+"&descripcion="+descripcion+"&costo="+costo;
98 }
99 function eliminarProd(){
100   alert("eliminado")
101 }
```

Este método funciona de la siguiente manera cuando presionamos nuevo producto no enviamos Id del producto así que ya sabemos que es un producto nuevo, pero si enviamos un id es una actualización. También tenemos que agregar este método que nos recuperar los datos para actualizar. Te señalo la ruta con los parámetros para actualizar un producto.

```
function getParameterByName(name) {
  name = name.replace(/[\/]/, "\\").replace(/[\\]/, "\\");
  var regex = new RegExp("[\\?&]" + name + "=(^&#*)"),
      results = regex.exec(location.search);
  return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));
}
```

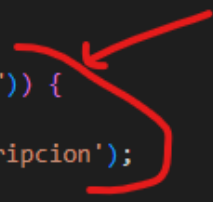
Entonces al agregarlo nos quedaría así en nuestro código.

Creamos un evento a parte en el documento para que inicie retrasado y verifique si hay parámetros, si los hay los pone en el formulario para actualizar. Y si no, no pasa nada mas

```

window.addEventListener('load', function () {
  setTimeout(() => {
    if (document.querySelector("#formProductos")) {
      var prodId = getParameterByName('id');
      var descripcion = getParameterByName('descripcion');
      var costo = getParameterByName('costo');
      if (prodId != null || prodId != '' || descripcion != '') {
        document.querySelector('#idproducto').value = prodId;
        document.querySelector('#textDescripcion').value = descripcion;
        document.querySelector('#numCosto').value = costo;
      }
    }
    GuardarProducto();
  }, 500);
}, false);

```





Listo ahora seguimos con el método de eliminar el producto, es casi el mismo código solo cambia el endpoint de conexión por DELETE así se vería nuestro código. en nuestro archivo producto.html agregamos lo siguiente alertas.

```

<header class="pb-3 mb-4 border-bottom">
  <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
    <a href="/Vista/nuevoProducto.html" class="btn btn-info">Nuevo Producto</a>
  </a>
  <div class="alert alert-success" role="alert" style="display:none;" id="alertSuccessmenu">
  </div>
  <div class="alert alert-danger" role="alert" style="display:none;" id="alertErrormenu">
  </div>
</header>
<section class="content">
  <table class="table display" id="tablaProducto" style="width:100%">
    <thead>
      <tr>

```

Bueno después de esto si procedemos con el método para eliminar el producto

```

function eliminarProd(id) {
    window.confirm("Esta seguro de Eliminar este Producto");
    let request = (window.XMLHttpRequest) ?
        new XMLHttpRequest() :
        new ActiveXObject('Microsoft.XMLHTTP');
    let ajaxUrl = 'http://localhost:8080/producto/' + id; //rutas api metodo post
    request.open("DELETE", ajaxUrl);
    request.setRequestHeader("Accept", "*/*");
    request.setRequestHeader("Content-Type", "application/json");
    request.send();
    request.onreadystatechange = function () {
        if (request.status === 4) {
            let error = document.getElementById('alertSuccessmenu'); //mensajes de error
            error.innerHTML = 'Muy Bien , Producto Eliminado';
            error.style.display = 'flex';
            window.location.reload();
        } else {
            let error = document.getElementById('alertErrormenu'); //mensajes de error
            error.innerHTML = 'Error !! No se puede Eliminar el Producto';
            error.style.display = 'flex';
            window.location.reload();
        }
    };
    return false;
}
}

```

Listo tenemos completo el **CRUD** para producto a partir de aquí te pongo como quedaron cada una de los demás métodos, siguiendo el mismo proceso.

Iniciemos con proveedor este es el método que carga la tabla proveedor:

```

Inventario.html JS Proveedor.js X
Assets > js > JS Proveedor.js > document.addEventListener('DOMContentLoaded') callback
1 let tablaProveedor;
2 document.addEventListener('DOMContentLoaded', function () {
3     $.fn.dataTable.ext.errMode = 'throw'; //ocultar error del datatable
4     tablaProveedor = $('#tablaProveedor').dataTable({
5         "Processing": true,
6         "ServerSide": true,
7         "ajax": {
8             "url": "http://localhost:8080/proveedor", //ruta de la api
9             "dataSrc": ""
10        },
11        "columns": [
12            { "data": "idpk" },
13            { "data": "nombre" },
14            { "data": "nit" },
15            { "data": "telefono" },
16            { "data": "accion" }
17        ], columnDefs: [
18            { targets: 4, visible: true },
19            {
20                targets: -1,
21                orderable: false,
22                data: null,
23                render: function (data, type, row, meta) {
24                    let botones = `
25                    <div class="btn-group">
26                    <button onClick="EditarProv('${ row.idpk }', '${ row.nombre }', '${ row.nit }', '${ row.telefono }');" class="btn btn-primary">Editar</button>
27                    <button onClick="eliminarProv('${ row.idpk }');" class="btn btn-danger">Eliminar</button>
28                    </div>`;
29                    return botones; } }
30        ],
31        'dom': 'lBfrtip',
32        "resonsieve": "true",
33        "bDestroy": true,
34        "iDisplayLength": 10,
35        "order": [[0, "asc"]]
36    });
37 }, false);

```

Esta función inicia después de cierto tiempo en DOM y carga la función guardar.

```
40
41 window.addEventListener('load', function () {
42     setTimeout(() => {
43         if (document.querySelector("#formProveedor")) {
44             var id = getParameterByName('id');
45             var nom = getParameterByName('nom');
46             var nit = getParameterByName('nit');
47             var tel = getParameterByName('tel');
48             if (id != '' || nom != '' || nit != '' || tel != '') {
49                 document.querySelector('#idproveedor').value = id;
50                 document.querySelector('#textnombre').value = nom;
51                 document.querySelector('#textNit').value = nit;
52                 document.querySelector('#telefono').value = tel;
53             }
54         }
55     }, 500);
56     GuardarProveedor();
57 }, false);
```

Método Guardar Proveedor

```
59
60 function GuardarProveedor() {
61     if (document.querySelector("#formProveedor")) { //validamos que exista el formulario
62         let formProductos = document.querySelector("#formProveedor"); //seleccionamos el formulario
63         formProductos.onsubmit = function (e) {
64             e.preventDefault(); //evitamos que recargue la pagina al precionar el boton
65             let id = document.querySelector('#idproveedor').value;
66             let nombre = document.querySelector('#textnombre').value;
67             let nit = document.querySelector('#textNit').value;
68             let tel = document.querySelector('#telefono').value;
69             if (nombre == '' || nit == '') {
70                 let error = document.getElementById('alertError');
71                 error.innerHTML = 'Todos los campos son obligatorios.'
72                 error.style.display = 'flex';
73                 return false; }
74             let request = (window.XMLHttpRequest) ?
75                 new XMLHttpRequest() :
76                 new ActiveXObject('Microsoft.XMLHTTP');
77             let ajaxUrl = 'http://localhost:8080/proveedor'; //rutas api metodo post
78             let json = JSON.stringify({ idpk: id, nombre: nombre, nit: nit, telefono: tel }) //creamos un json
79             request.open("POST", ajaxUrl, true);
80             request.setRequestHeader("Content-Type", "application/json");
81             request.send(json);
82             request.onreadystatechange = function () {
83                 if (request.status == 200) {
84                     let exito = document.getElementById('alertSuccess'); //mensajes de exito
85                     exito.innerHTML = 'Muy Bien, Se Guardo un Proveedor Correctamente.'
86                     exito.style.display = 'flex';
87                     window.location.href = "/Vista/Proveedores.html";
88                 } else {
89                     let error = document.getElementById('alertError'); //mensajes de error
90                     error.innerHTML = 'Error !! No se puedo Guardar el Proveedor'
91                     error.style.display = 'flex';
92                     return false;
93                 }
94                 return false;
95             }
96         }
97     }
98 }
```

Métodos EDITAR (reenvía los datos al formulario)

Método Eliminar

Y el método que recura la información por medio get url

```
Assets > js > JS Proveedor.js > eliminarProv
98 }
99
100 function EditarProv(id,nom,nit,tel){
101     window.location.href = "/Vista/nuevoProveedor.html?id=" + id + "&nom=" + nom + "&nit=" +
102     nit+ "&tel=" + tel;
103 }
104 function eliminarProv(idpk){
105     window.confirm("Esta seguro de Eliminar este Proveedor?");
106     let request = (window.XMLHttpRequest) ?
107         new XMLHttpRequest() :
108         new ActiveXObject('Microsoft.XMLHTTP');
109     let ajaxUrl = 'http://localhost:8080/proveedor/' + idpk; //rutas api metodo post
110     request.open("DELETE", ajaxUrl);
111     request.setRequestHeader("Accept", "*/");
112     request.setRequestHeader("Content-Type", "application/json");
113     request.send();
114     request.onreadystatechange = function () {
115         if (request.status === 200) {
116             let error = document.getElementById('alertSuccessmenu'); //mensajes de error
117             error.innerHTML = 'Muy Bien , Proveedor Eliminado'
118             error.style.display = 'flex';
119             Window.confirm('Proveedor Eliminado');
120             window.location.reload();
121         }else{
122             let error = document.getElementById('alertErrormenu'); //mensajes de error
123             error.innerHTML = 'Error !! No se puedo Eliminar el Proveedor'
124             error.style.display = 'flex';
125         }
126         return false;
127     }
128 }
129
130
131 function getParameterByName(name) {
132     name = name.replace(/\[/, "\\[").replace(/\]/, "\\]");
133     var regex = new RegExp("[\\?&]" + name + "=(^&#)*"),
134         results = regex.exec(location.search);
135     return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));
136 }
```

Listo ahora les mostrare el proveedor.html, solo el header y el cuerpo lo de arriba es lo mismo para todos.

```
46 </div>
47 <header class="pb-3 mb-4 border-bottom">
48   <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
49     <a href="/Vista/nuevoProveedor.html" class="btn btn-info">Nuevo Proveedor</a>
50   </a>
51 </header>
52 <div class="alert alert-success" role="alert" style="display:none;" id="alertSuccessmenu">
53 </div>
54 <div class="alert alert-danger" role="alert" style="display:none;" id="alertErrormenu">
55 </div>
56 <section class="content">
57   <table class="table display" id="tablaProveedor" style="width:100%">
58     <thead>
59       <tr>
60         <th scope="col">#</th>
61         <th scope="col">Nombre</th>
62         <th scope="col">Nit</th>
63         <th scope="col">Telefono</th>
64         <th scope="col">Acciones</th>
65       </tr>
66     </thead>
67     <tbody class="table-group-divider">
68
69     </tbody>
70   </table>
71 </section>
72 </div>
73 <script src="https://code.jquery.com/jquery-3.6.1.js"
74   integrity="sha256-3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFxCc2vC/7pNI=" crossorigin="anonymous"></script>
75 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
76   integrity="sha384-u10knCvxWvY5KfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZlHgTP00mMi466C8"
77   crossorigin="anonymous"></script>
78 <script src="/Assets/js/jquery.dataTables.min.js"></script>
79 <script src="/Assets/js/Proveedor.js"></script>
80 </body>
81
82 </html>
```

Vista de nuevo proveedor:

```
<div>
  <header class="pb-3 mb-4 border-bottom">
    <h3 class="title">Crear Proveedor</h3>
  </header>
  <section class="content">
    <form id="formProveedor" accept-charset="utf-8">
      <div class="mb-3">
        <input type="hidden" class="form-control" name="idpk" value="0" id="idproveedor">
        <label for="textnombre" class="form-label">Nombre</label>
        <input type="text" class="form-control" name="nombre" id="textnombre"
          placeholder="Nombre Proveedor">
      </div>
      <div class="mb-3">
        <label for="textNit" class="form-label">NIT</label>
        <input type="number" class="form-control" id="textNit" name="nit"
          placeholder="Identificacion Proveedor" />
      </div>
      <div class="mb-3">
        <label for="telefono" class="form-label">Telefono</label>
        <input type="number" class="form-control" id="telefono" name="telefono"
          placeholder="Numero de Contacto" />
      </div>
      <button class="btn btn-primary" onclick="GuardarProveedor();" id="GuardarProveedor">Guardar</button>
    </form>

    <div class="alert alert-success" role="alert" style="display:none" id="alertSuccess">
    </div>
    <div class="alert alert-danger" role="alert" style="display:none" id="alertError">
    </div>
  </section>
</div>
</nav>
<script src="https://code.jquery.com/jquery-3.6.1.js"
  integrity="sha256-3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFx_Cc2vC/7pNI=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvXWVY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvzZlHgTP00mMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/Proveedor.js"></script>
</body>
</html>
```

Listo esto es todo por parte de Proveedor ahora seguimos con inventario la cual se encarga de recuperar los productos y proveedores para asignarse a un inventario.

Método que carga la tabla:

```
JS Inventario.js X
Assets > js > JS Inventario.js > document.addEventListener("DOMContentLoaded") callback > columnDefs > render > botones
1 let tableInventario;
2 document.addEventListener('DOMContentLoaded', function () {
3   $.fn.dataTable.ext.errMode = 'throw'; //ocultar error del datatable
4   tableInventario = $('#tablaInventario').dataTable({
5     "Processing": true,
6     "ServerSide": true,
7     "ajax": {
8       "url": "http://localhost:8080/inventario", //ruta de la api
9       "dataSrc": ""
10    },
11    "columns": [
12      { "data": "idpk" },
13      { "data": "producto.descripcion" },
14      { "data": "proveedor.nombre" },
15      { "data": "stock" },
16      { "data": "secursal" },
17      { "data": "accion" }
18    ], columnDefs: [
19      { targets: 5, visible: true },
20      { targets: -1,
21        orderable: false,
22        data: null,
23        render: function (data, type, row, meta) {
24          let botones = `
25            <div class="btn-group">
26              <button onClick="ActualizarStock(`+ row.idpk + `,`+
27                row.stock + `);" class="btn btn-danger">Stock</button>
28            </div>`;
29          return botones;}}
30    ],
31    'dom': 'lBfrtip',
32    "resonsieve": "true",
33    "bDestroy": true,
34    "iDisplayLength": 10,
35    "order": [[0, "asc"]]
36  });
37 }, false);
38
```

Función que nicia después de cargar el DOM y llama a dos métodos que es actualizar y guardar.

```

38
39 window.addEventListener('load', function () {
40     setTimeout(() => {
41         if (document.querySelector("#formstock")) {
42             var invenId = getParameterByName('id');
43             var stock = getParameterByName('stock');
44             if (invenId != null ) {
45                 document.querySelector('#idinvent').value = invenId;
46                 document.querySelector('#numStock').value = stock;
47             }
48             Actualizar();
49         }
50     }, 500);
51     GuardarInventario();
52 }, false);
53

```

Método guardar

```

function GuardarInventario() {
    if (document.querySelector("#formInventario")) { //validamos que exista el formulario
        let formProductos = document.querySelector("#formInventario"); //seleccionamos el formulario
        formProductos.onsubmit = function (e) {
            e.preventDefault(); //evitamos que recargue la pagina al precionar el boton
            let idproducto = document.querySelector('#intproducto').value;
            let idproveedor = document.querySelector('#intproveedor').value;
            let idsucursal = document.querySelector('#numSucursal').value;
            let stock = document.querySelector('#numStock').value;
            if (idproducto == '' || idproveedor == '' || idsucursal == '' || stock == '') {
                let error = document.getElementById('alertError');
                error.innerHTML = 'Todos los campos son obligatorios.'
                error.style.display = 'flex';
                return false;
            }
            let request = (window.XMLHttpRequest) ?
                new XMLHttpRequest() :
                new ActiveXObject('Microsoft.XMLHTTP');
            let ajaxUrl = 'http://localhost:8080/producto'; //rutas api metodo post
            let json = JSON.stringify({producto:{idpk:idproducto}, proveedor:{idpk:idproveedor},
                secursal:idsucursal, stock: stock }) //creamos un json
            request.open("POST", ajaxUrl, true);
            request.setRequestHeader("Content-Type", "application/json");
            request.send(json);
            request.onreadystatechange = function () {
                if (request.status == 200) {
                    let exito = document.getElementById('alertSuccess'); //mensajes de exito
                    exito.innerHTML = 'Muy Bien, Se Guardo el Inventario Correctamente.'
                    exito.style.display = 'flex';
                    window.location.href = "/Vista/Inventario.html";
                } else {
                    let error = document.getElementById('alertError'); //mensajes de error
                    error.innerHTML = 'Error !! No se pudo Inventario el Producto'
                    error.style.display = 'flex';
                    return false;
                }
            }
            return false;
        }
    }
}

```

Método actualizar y el la función que redirige los datos para actualizar.

```
function ActualizarStock(idpk,stock) {  
    window.location.href = "/Vista/actualizarStock.html?id="+idpk+"&stock="+stock;  
}  
function Actualizar() {  
    if (document.querySelector("#formstock")) { //validamos que exista el formulario  
        let formProductos = document.querySelector("#formstock"); //seleccionamos el formulario  
        formProductos.onsubmit = function (e) {  
            e.preventDefault(); //evitamos que recargue la pagina al precionar el boton  
            let idinvent= document.querySelector('#idinvent').value;  
            let stock = document.querySelector('#numStock').value;  
            if (idinvent == '' || stock == '') {  
                let error = document.getElementById('alertError');  
                error.innerHTML = 'Todos los campos son obligatorios.'  
                error.style.display = 'flex';  
                return false;  
            }  
            let request = (window.XMLHttpRequest) ?  
                new XMLHttpRequest() :  
                new ActiveXObject('Microsoft.XMLHTTP');  
            let ajaxUrl = 'http://localhost:8080/inventario'; //rutas api metodo post  
            let json = JSON.stringify({idpk:idinvent, stock:stock }) //creamos un json  
            request.open("PUT", ajaxUrl, true);  
            request.setRequestHeader("Content-Type", "application/json");  
            request.send(json);  
            request.onreadystatechange = function () {  
                if (request.status == 200) {  
                    let exito = document.getElementById('alertSuccess'); //mensajes de exito  
                    exito.innerHTML = 'Muy Bien, Se Actualizo el Inventario Correctamente.'  
                    exito.style.display = 'flex';  
                    window.location.href = "/Vista/Inventario.html";  
                } else {  
                    let error = document.getElementById('alertError'); //mensajes de error  
                    error.innerHTML = 'Error !! No se puedo Actualizar el Inventario'  
                    error.style.display = 'flex';  
                    return false;  
                }  
                return false;  
            }  
        }  
    }  
}
```

```
138  
139 function getParameterByName(name) {  
140     name = name.replace(/[\/]/, "\\").replace(/[\\]/, "\\");  
141     var regex = new RegExp("[\\?&]" + name + "=(^&#)*"),  
142         results = regex.exec(location.search);  
143     return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));  
144 }
```

Vistas nuevo inventario

```
<header class="pb-3 mb-4 border-bottom">
  <h3 class="title">Crear Inventario</h3>
</header>
<section class="content">
  <form id="formInventario" accept-charset="utf-8" >
    <div class="mb-3">
      <label for="intproducto" class="form-label">Producto</label>
      <input type="number" class="form-control" name="idproducto" id="intproducto"
        placeholder="id del Producto">
    </div>
    <div class="mb-3">
      <label for="intproveedor" class="form-label">Proveedor</label>
      <input type="number" class="form-control" name="idproveedor" id="intproveedor"
        placeholder="id del Proveedor">
    </div>
    <div class="mb-3">
      <label for="numSucursal" class="form-label">Sucursal</label>
      <input type="number" class="form-control" id="numSucursal" name="idsucursal"
        placeholder="Id de la Sucursal"/>
    </div>
    <div class="mb-3">
      <label for="numStock" class="form-label">Stock</label>
      <input type="number" class="form-control" id="numStock" name="stock"
        placeholder="cantidad de Productos"/>
    </div>
    <button class="btn btn-primary" onclick="GuardarInventario();" id="GuardarInventario">Guardar</button>
  </form>

  <div class="alert alert-success" role="alert" style="display:none" id="alertSuccess">
  </div>
  <div class="alert alert-danger" role="alert" style="display:none" id="alertError">
  </div>
</section>
</nav>
```

Vista actualizar stock

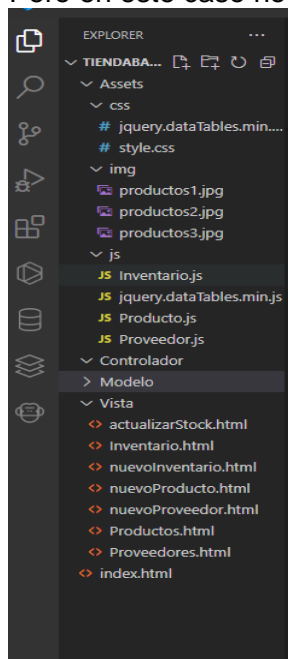
```
<header class="pb-3 mb-4 border-bottom">
  <h3 class="title">Crear Inventario</h3>
</header>
<section class="content">
  <form id="formstock" accept-charset="utf-8" >
    <div class="mb-3">
      <input type="hidden" class="form-control" id="idinvent" name="idinvent" />
      <label for="numStock" class="form-label">Stock</label>
      <input type="number" class="form-control" id="numStock" name="stock"
        placeholder="cantidad de Productos"/>
    </div>
    <button class="btn btn-primary" onclick="actualizar();" id="GuardarInventario">Actualizar</button>
  </form>

  <div class="alert alert-success" role="alert" style="display:none" id="alertSuccess">
  </div>
  <div class="alert alert-danger" role="alert" style="display:none" id="alertError">
  </div>
</section>
</nav>
<script src="https://code.jquery.com/jquery-3.6.1.js"
  integrity="sha256-3zlB5s2woUzrXK3BT7AX3FyvojsraNFx_Cc2vC/7pNI=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-u10knCvwwKfMNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvZ1HgTPOOmMi466C8"
  crossorigin="anonymous"></script>
<script src="/Assets/js/Inventario.js"></script>
</body>
</html>
```

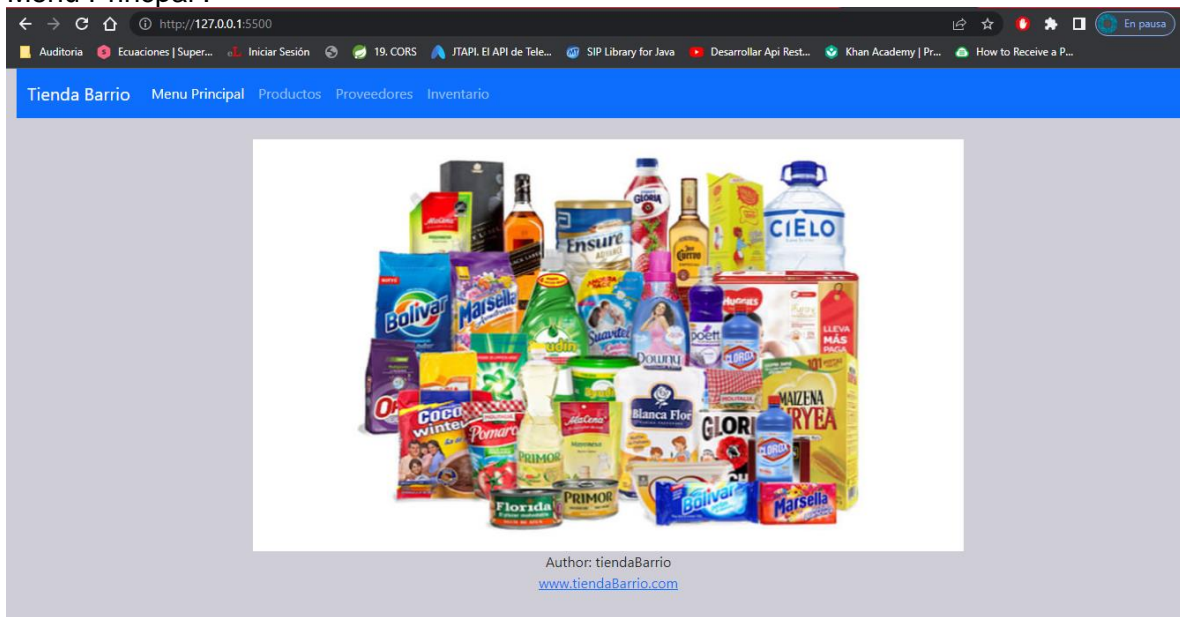
Vista inventario.html

```
</div>
<div class="pb-3 mb-4 border-bottom">
  <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
    <a href="/Vista/nuevoInventario.html" class="btn btn-info">Nuevo Inventario</a>
  </a>
</div>
</header>
<div class="alert alert-success" role="alert" style="display:none;" id="alertSuccessmenu">
</div>
<div class="alert alert-danger" role="alert" style="display:none;" id="alertErrormenu">
</div>
</div>
</header>
<section class="content">
  <table class="table display" id="tablaInventario" style="width:100%">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Producto</th>
        <th scope="col">Proveedor</th>
        <th scope="col">stock</th>
        <th scope="col">Sucursal</th>
        <th scope="col">Acciones</th>
      </tr>
    </thead>
    <tbody class="table-group-divider">
      <tr>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </tbody>
  </table>
</section>
</div>
</body>
<script src="https://code.jquery.com/jquery-3.6.1.js"
integrity="sha256-3zl85s2uwoUzrXK3BT7AX3FyvojsraNFxCc2vC/7pNI=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u10knCvxywY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
<script src="/Assets/js/jquery.dataTables.min.js"></script>
<script src="/Assets/js/Inventario.js"></script>
</html>
```

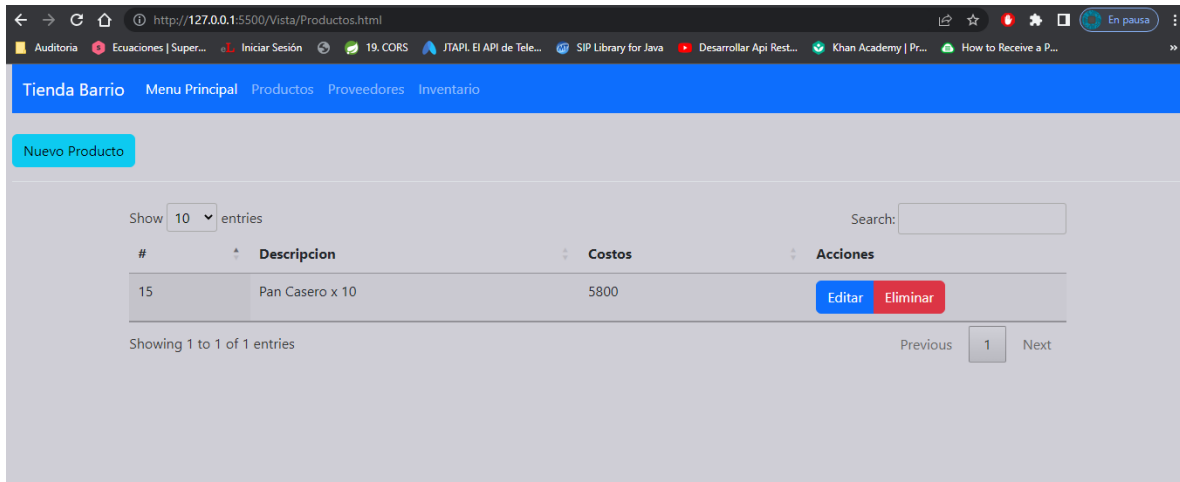
Así quedaría nuestro proyecto final, aunque no utilizamos lógica para controlador y tampoco lógica para modelo ya que toda venia desde el API. Aunque se pueden realizar. Pero en este caso no se utilizó para ser más práctico.



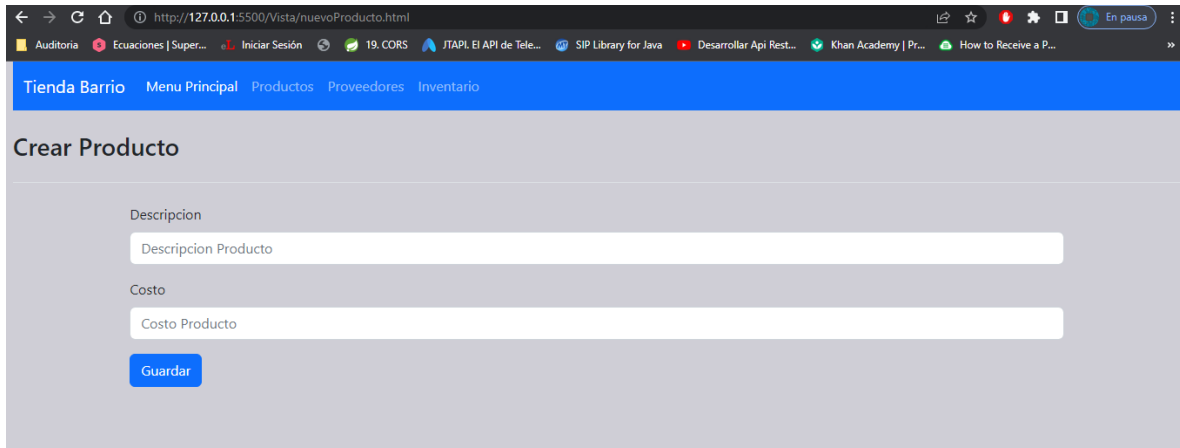
Menu Principal :



Productos :



Nuevo Producto:



Proveedor:

← → ↻ 🏠 http://127.0.0.1:5500/Vista/Proveedores.html 🔖 ⚙️ 🖨️ En pausa ⋮

Auditoria Ecuaciones | Super... Iniciar Sesión 19. CORS JTAPL El API de Tele... SIP Library for Java Desarrollar Api Rest... Khan Academy | Pr... How to Receive a P...

Tienda Barrio Menu Principal Productos Proveedores Inventario

Nuevo Proveedor

Show 10 entries Search:

#	Nombre	Nit	Telefono	Acciones
16	Surtidora Medrano	545565669	2147483647	Editar Eliminar

Showing 1 to 1 of 1 entries Previous 1 Next

Nuevo proveedor:

← → ↻ 🏠 http://127.0.0.1:5500/Vista/nuevoProveedor.html 🔖 ⚙️ 🖨️ En pausa ⋮

Auditoria Ecuaciones | Super... Iniciar Sesión 19. CORS JTAPL El API de Tele... SIP Library for Java Desarrollar Api Rest... Khan Academy | Pr... How to Receive a P...

Tienda Barrio Menu Principal Productos Proveedores Inventario

Crear Proveedor

Nombre
Nombre Proveedor

NIT
Identificacion Proveedor

Telefono
Numero de Contacto

Guardar

Inventario :

← → ↻ 🏠 http://127.0.0.1:5500/Vista/Inventario.html 🔖 ⚙️ 🖨️ En pausa ⋮

Auditoria Ecuaciones | Super... Iniciar Sesión 19. CORS JTAPL El API de Tele... SIP Library for Java Desarrollar Api Rest... Khan Academy | Pr... How to Receive a P...

Tienda Barrio Menu Principal Productos Proveedores Inventario

Nuevo Inventario

Show 10 entries Search:

#	Producto	Proveedor	stock	Sucursal	Acciones
24	Pan Casero x 10	Surtidora Medrano	7	5	Stock

Showing 1 to 1 of 1 entries Previous 1 Next

Nuevo inventario:

Producto

id del Producto

Proveedor

id del Proveedor

Sucursal

Id de la Sucursal

Stock

cantidad de Productos

Guardar

Actualizar Stock:

Stock

7

Actualizar

Listo esto sería todo, ya tendríamos funcionando nuestro proyecto.

