

Actividad 3 Proyecto Final - Funcionamiento del carrito

Desarrollo de Sistemas Web 2

Ingeniería en Desarrollo de Software

Tutor: Aarón Iván Salazar Macías

Alumno: Fernando Pedraza Garate

Fecha: 18 de Febrero del 2024

Índice

○ Introducción.	Pág. 3 - 4
○ Descripción	Pág. 5 - 7
○ Justificación	Pág. 8 – 11
Etapa 1 – Base de Datos	
○ Desarrollo	Pág. 12
• Diagramas UML	Pág. 13
• Descripción de la BD	Pág. 14 - 18
• Pruebas de la BD	Pág. 19 – 22
Etapa 2 – Diseño Web y Conexión	
○ Desarrollo	Pág. 23 - 42
• Diseño del proyecto	
• Codificación	
Etapa 3 – Funcionamiento del carrito	
○ Desarrollo	Pág. 43-59
• Conexión a la BD	
• Codificación	
• Pruebas del sitio web	
○ Conclusión	Pág. 60-61
○ Referencias	Pág. 62-63

Introducción

Una base de datos es una recopilación organizada de información o datos estructurados que normalmente se almacena de forma electrónica en un sistema informático y está controlada por un sistema de gestión de bases de datos (DBMS), sirve como interfaz entre la base de datos y sus programas o usuarios finales, lo que permite a los usuarios recuperar, actualizar y gestionar cómo se organiza y se optimiza la información, permitiendo una variedad de operaciones administrativas como la supervisión del rendimiento, el ajuste, la copia de seguridad y la recuperación de información, que en conjunto, y junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado solamente a base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos, permitiendo se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos, en la mayoría de las bases de datos se utiliza un lenguaje de consulta estructurada (SQL) para escribir y consultar datos, SQL es un lenguaje de programación que utilizan casi todas las bases de datos relacionales para consultar, manipular y definir los datos, además de proporcionar control de acceso.

El diseño web es una disciplina enfocada en la planificación, ideación, creación e implementación de interfaces y plataformas digitales, es una actividad creativa enfocada en realizar la parte visual de dichos entornos y materiales, encargada de los layouts e interfaces de las plataformas de un sitio web o de una app que realiza una configuración visible y funcional para el usuario al distribuir y conceptualizar diversos elementos, aunque su definición es sencilla, su producción no lo es, el diseño web aúna un gran elenco de factores que interpretan roles diferentes en la creación de un sitio o página web con éxito, tales como, el diseño de la interfaz, el diseño del material y los recursos gráficos, la experiencia de usuario, la adaptabilidad del diseño en todo tipo de dispositivos, etc. estos son solo algunos de los elementos que conforman el diseño web y lo hacen posible, en otras palabras, es el proceso que engloba todos los elementos necesarios para crear una página web.

Una conexión de datos es un vínculo dinámico entre un formulario y un origen de datos que almacena o proporciona datos de ese formulario, el cual puede tener una conexión de datos principal que define el origen de datos principal del formulario, y opcionalmente, puede tener una o varias conexiones de datos secundarios.

Un esquema XML describe cómo se almacenan los datos de origen de datos principal del formulario, solo puede haber una conexión de datos principal de un formulario y se crea automáticamente cuando se crea una plantilla de formulario que se basa en un origen de datos externo, se pueden crear tantas conexiones de datos secundarias como desee al diseñar una plantilla de formulario.

Definición del contexto.

Cerca del “buen fin” la tienda “Sara” requiere se realice una página web que permita a los usuarios manejar un carrito de compras, en específico, la tienda necesita que se realice un diseño de un sitio web e-commerce de micro servicios, en primera instancia, requieren que se cree el diseño de la base de datos, para posteriormente crearla y de acuerdo con la información proporcionada, las palabras en negrita serán las tablas y las que no están en negritas serán las columnas de cada tabla, planificando la creación de la base de datos mediante diagramas: entidad-relación y lógico-relacional para que sea más fácil de analizar y entender su lógica.

Productos:	Categorías:	Marcas:	Carrito:	CarritoDetalle:
-Id	-Id	-Id	-Id	-Id
-Descripción	-Descripción	-Descripción	-FolioVenta	IdCarrito
-Precio			-TotalCompra	IdProducto
-Cantidad			-Estatus	-Cantidad
-IdCategoria			-Fecha	-Subtotal
-IdMarca				

En resumen se deberá analizar y crear la base de datos para el sitio web de e-commerce con micro servicios de acuerdo con la información proporcionada debiendo dar lógica a las relaciones entre tablas por medio de un diagrama entidad-relación y lógico-relacional, para posteriormente, una vez comprendido, generar una nueva base de datos en SQL, utilizando el gestor de BD Microsoft SQL Server Management.

Posteriormente de haber planificado y creado la base de datos, la tienda Sara requiere se realice el diseño del sitio web, donde solicita que este sea con un diseño básico de una tienda en línea, que sea fácil de entender para el usuario y agradable a la vista, contemplando que el sistema web tenga las siguientes interfaces:

- Página de galería de productos
- Página de descripción del producto seleccionado.
- Página de pago del carrito.
- Un menú de categorías que se maneja

Para la galería de los productos, deberá de contar con las siguientes características:

- Los productos deberán de ser acomodados por categoría
- La vista del producto deberá de ser con el nombre, una imagen, y el precio
- Un filtro de búsqueda

El programa deberá de ser codificado con Java, Spring Boot con una API REST la cual debe retornar un JSON con los productos que se insertaron en la base de datos, además de crear un archivo de conexión del sistema con la base de datos.

Por último se deberá dar funcionalidad al apartado del carrito de compras, por lo que se requiere se utilice una API REST para que se puedan crear las altas, modificaciones, consultas y bajas de los productos de la base de datos, en donde los productos seleccionados deberán de sumar la cantidad total de la compra dependiendo de la cantidad de unidades seleccionadas, así como desarrollar el API REST (Alta, eliminación, consulta, modificación) de los productos extraídos de la base de datos, pudiendo visualizarse en el sistema web, y con esto se concluiría el desarrollo de la tienda online de Sara.

Justificación.

Se recomienda cubrir cada uno de los pasos a realizar en este tipo de requerimiento para concentrar toda la información necesaria que permita crear el diseño que tendrá la base de datos, ya que actualmente existe una gran variedad como:

- Las **bases de datos relacionales** donde se organizan como un conjunto de tablas con columnas y filas, proporcionando la forma más eficiente y flexible de acceder a información estructurada.
- La información de una **base de datos orientada a objetos** donde se representa en forma de objetos, tal como es en la programación orientada a objetos.
- Una **base de datos distribuida** es la que consta de dos o más archivos que se encuentran en diferentes sitios, donde esta se puede almacenar en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes.
- Un repositorio central de datos o un **data warehouse** es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos.

- Una **base de datos NoSQL**, o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos).
- Una **base de datos orientada a grafos** almacena datos relacionados con entidades y las relaciones entre entidades.
- **Bases de datos OLTP** es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones.

Otras bases de datos menos comunes se adaptan a funciones científicas, financieras o de otro tipo muy específicas. Además los cambios en los enfoques de desarrollo tecnológico y los avances considerables, como la nube y la automatización, están impulsando a las bases de datos en direcciones completamente nuevas. Algunas de las bases de datos más recientes incluyen

- Un sistema de **base de datos de código abierto** es aquel cuyo código fuente es de código abierto; tales bases de datos pueden ser bases de datos SQL o NoSQL.

- Una **base de datos en la nube** es una recopilación de datos, estructurados o no estructurados, que reside en una plataforma de cloud computing privada, pública o híbrida. Existen dos tipos de modelos de bases de datos en la nube: el modelo tradicional y el de base de datos como servicios (database as a service, DBaaS). Con DBaaS, un proveedor de servicios realiza las tareas administrativas y el mantenimiento.
- Las **bases de datos multimodelo** combinan distintos tipos de modelos de bases de datos en un único servidor integrado. Esto significa que pueden incorporar diferentes tipos de datos.
- Diseñadas para almacenar, recuperar y gestionar información orientada a los documentos, **las bases de datos de documentos** son una forma moderna de almacenar los datos en formato JSON en lugar de hacerlo en filas y columnas.
- El tipo de base de datos más nuevo e innovador, **las bases de datos de autogestión** (también conocidas como bases de datos autónomas) están basadas en la nube y utilizan el machine learning para automatizar el ajuste de la base de datos, la seguridad, las copias de seguridad, las actualizaciones y otras tareas de gestión rutinarias que tradicionalmente realizan los administradores de bases de datos.

Una vez seleccionado el tipo y diseño de BD se podrá cargar toda la información requerida y necesaria para su configuración y uso.

Una de las razones principales de realizar el diseño de la página web es porque una representa el canal de comunicación digital más directo con el usuario o consumidor, es la tarjeta de visita de una empresa en Internet y la que se encarga de transmitir el mensaje al mundo online, por ese motivo, el mensaje debe transmitirse de forma clara, atractiva y adecuada al público al que se pretende llegar, permitiendo satisfacer a los consumidores, a través de una experiencia de usuario perfecta, ajustada a sus necesidades y acompañarlos en su viaje por los contenidos, productos y servicios ofrecidos.

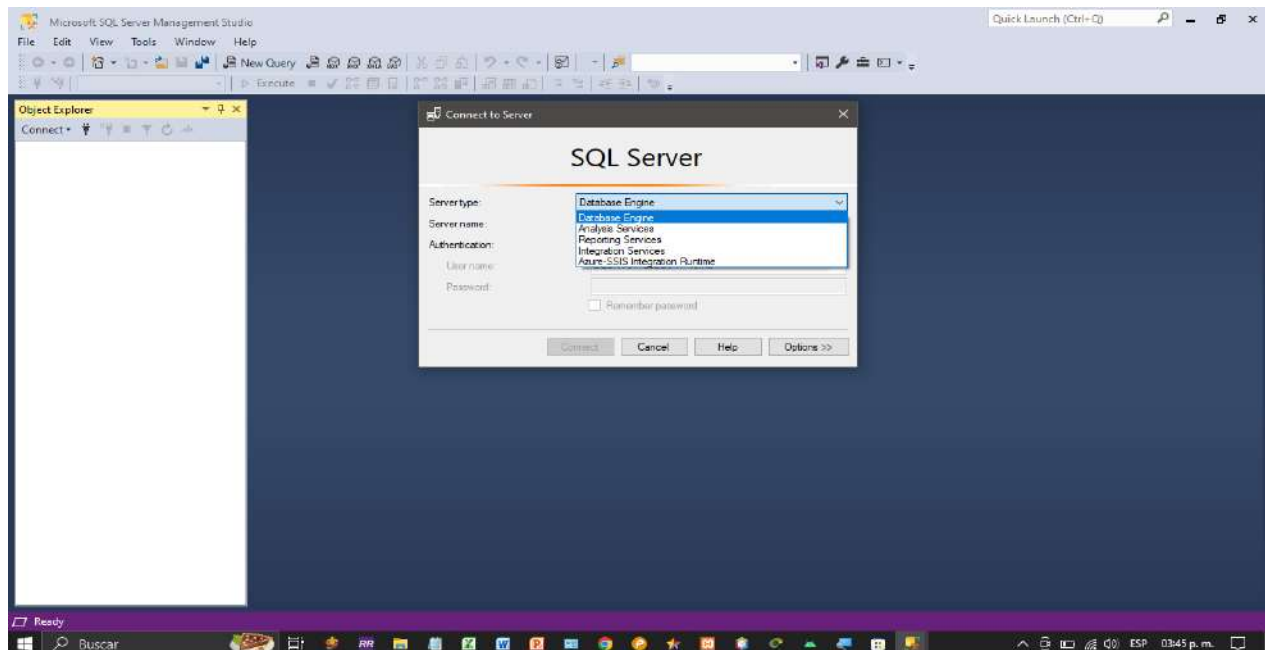
En otras palabras un buen diseño web es la pieza principal que permite abrir paso en la red y conseguir un puesto destacado entre los competidores, además debe estar bien planteado y estructurado desde sus inicios, buscando brindar la oportunidad de posicionarse mejor en los resultados de búsqueda, es decir, tener un sitio web preparado para posicionamiento SEO (Search Engine Optimization) desde el primer minuto.

Desarrollo.

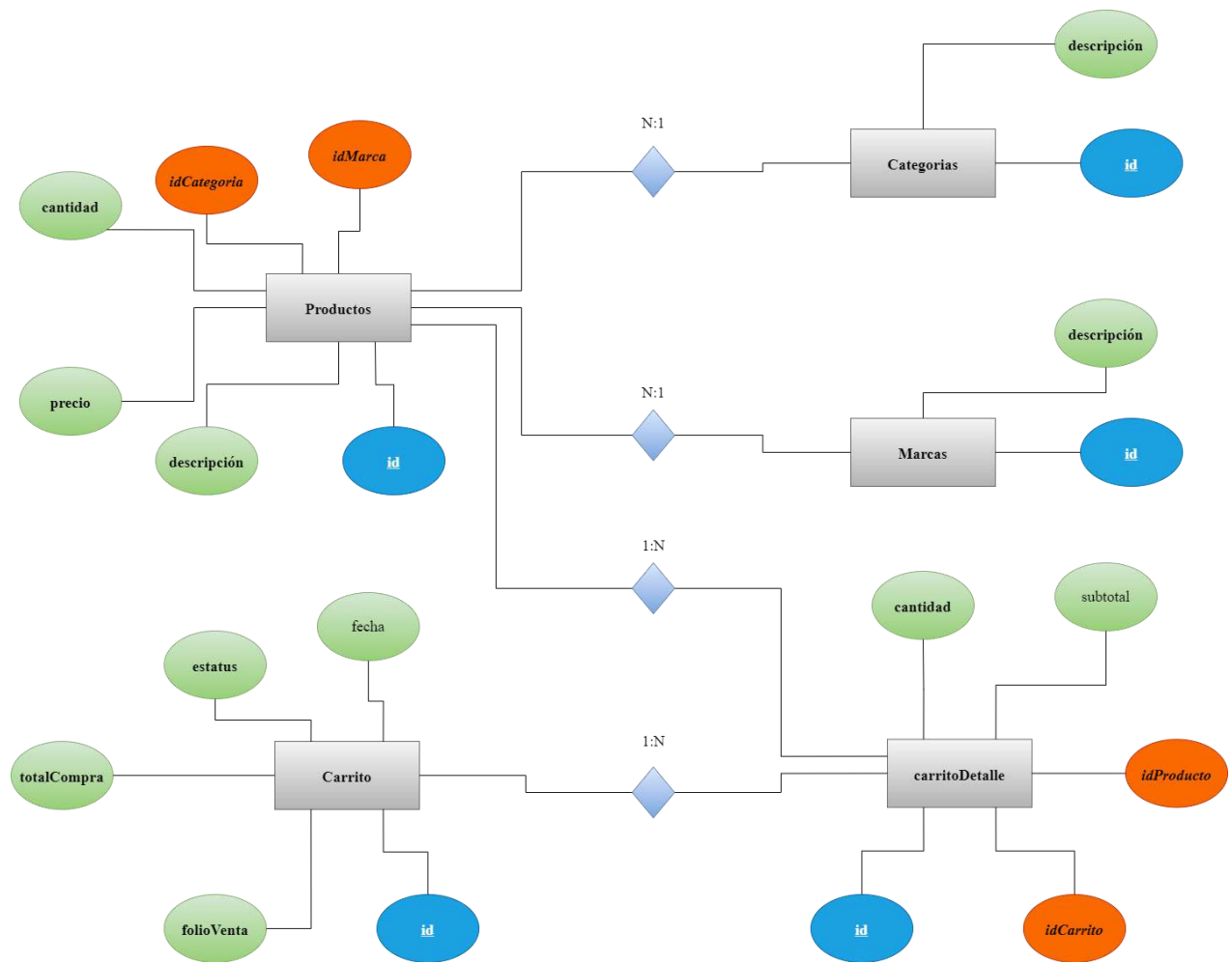
Etapa 1 – Base de Datos



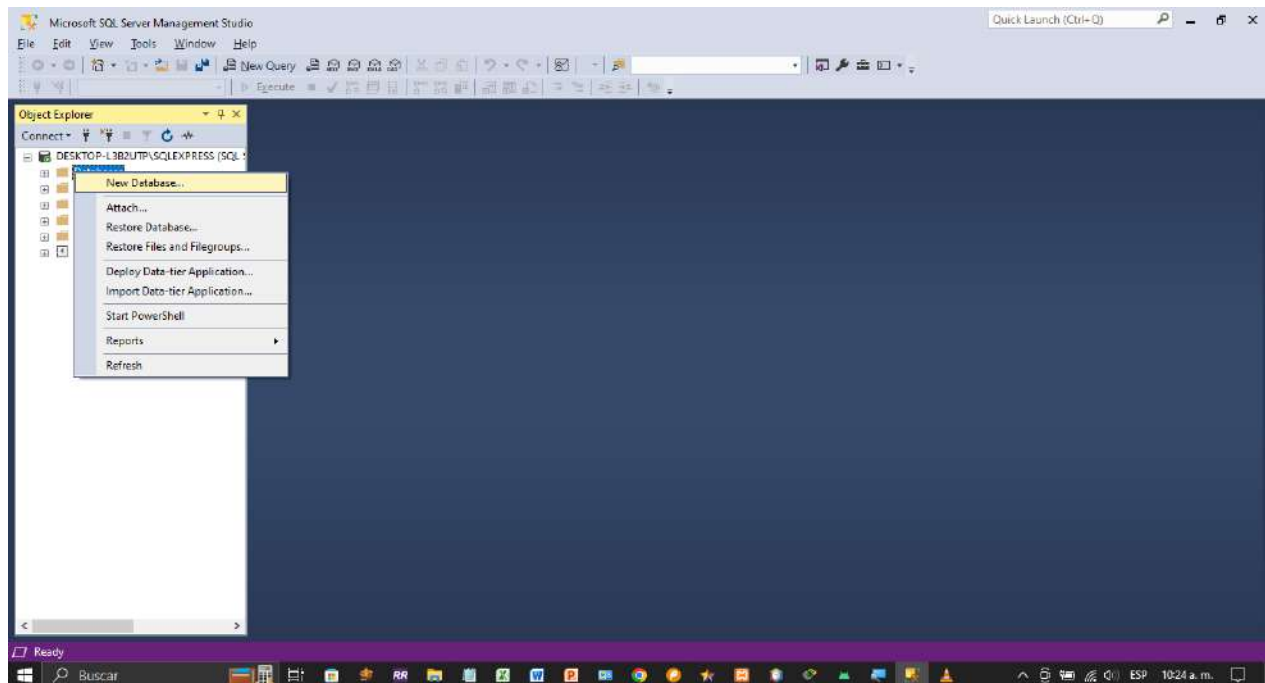
Instalación de SQL Server



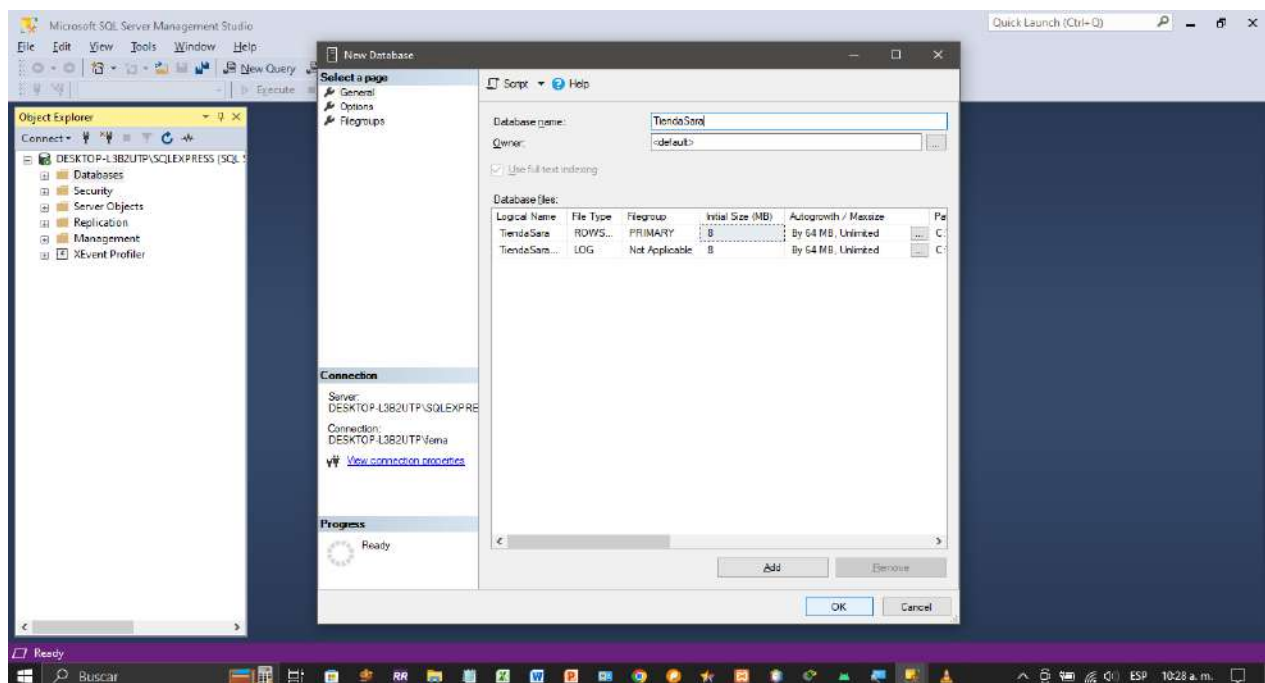
Diagramas UML

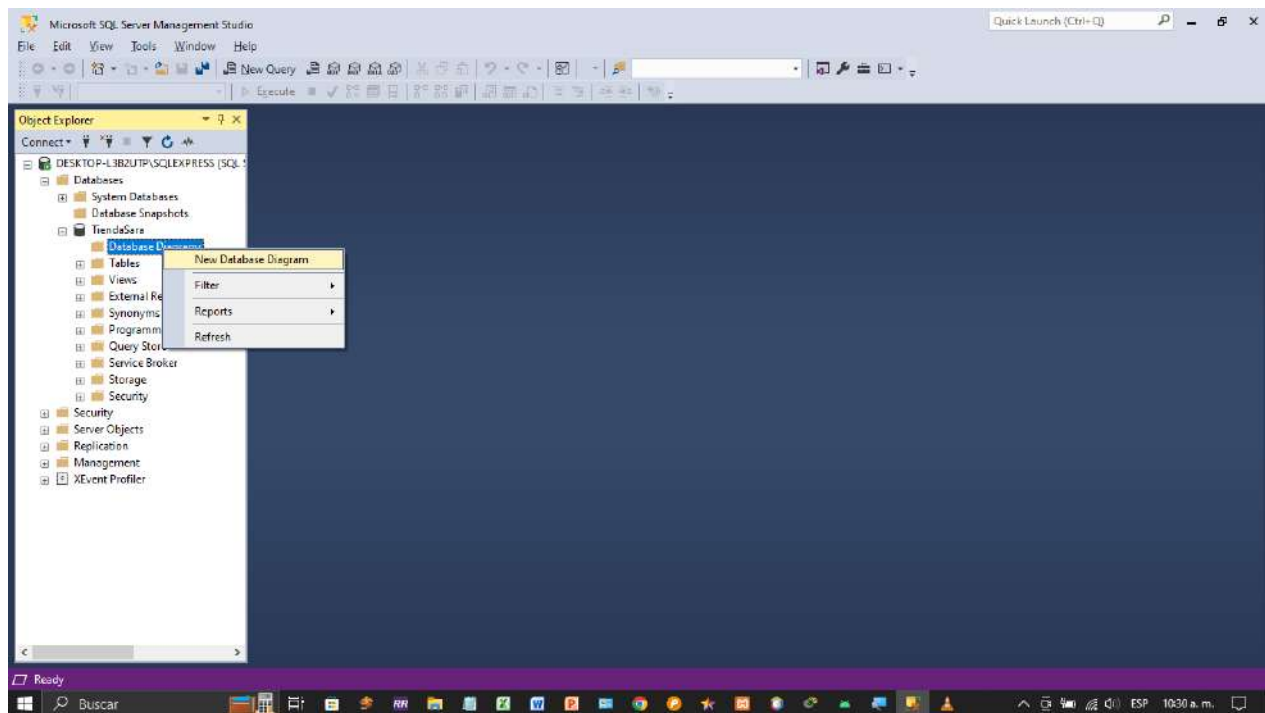


Descripción de la BD

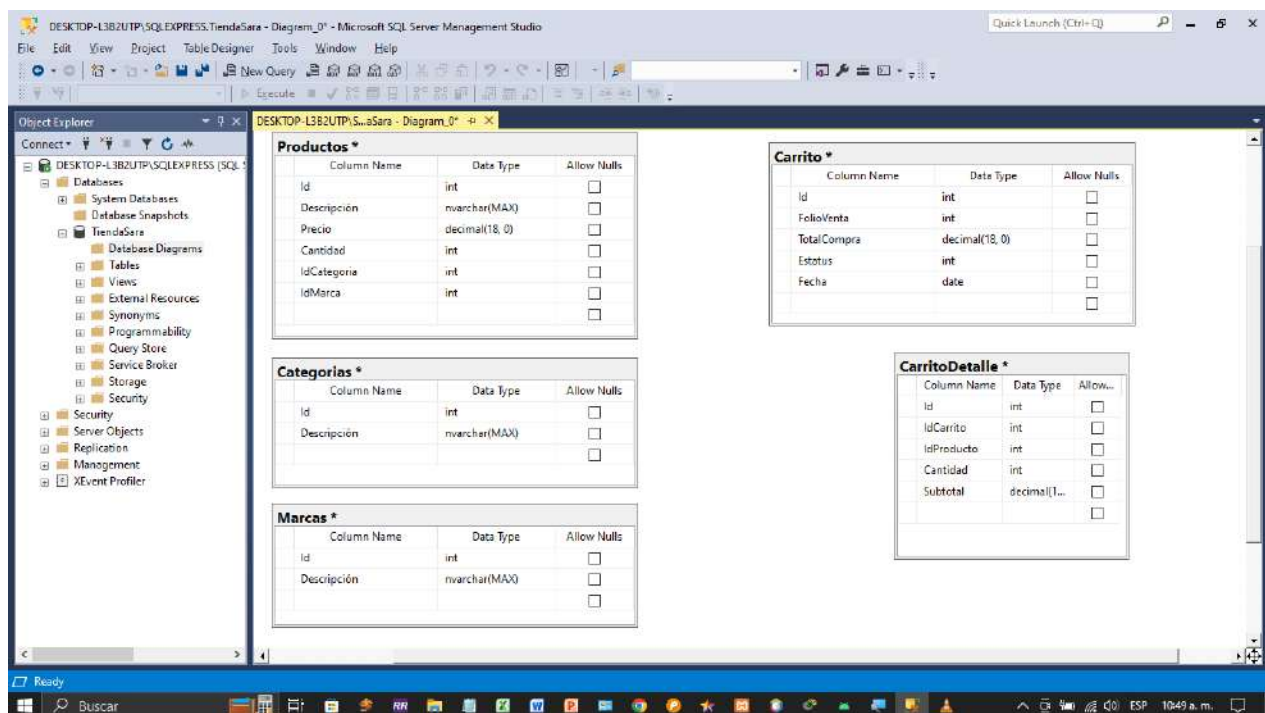


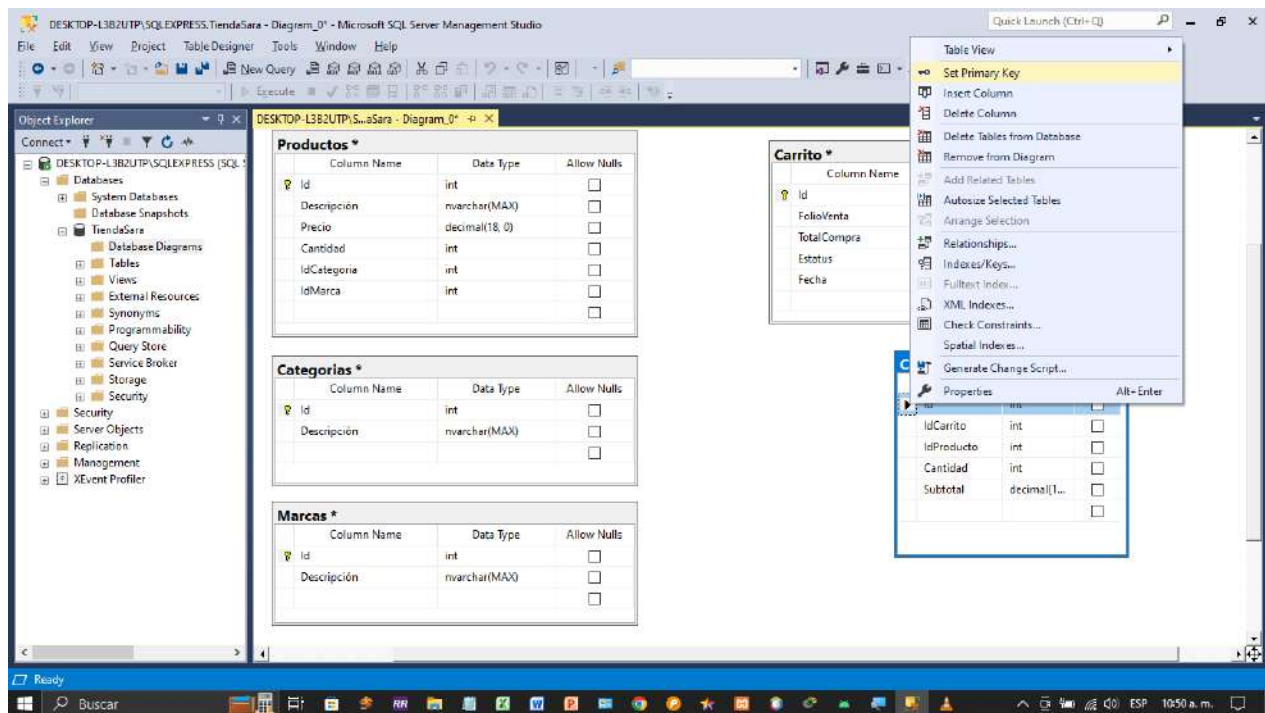
Se crea la nueva base de datos como Tienda Sara



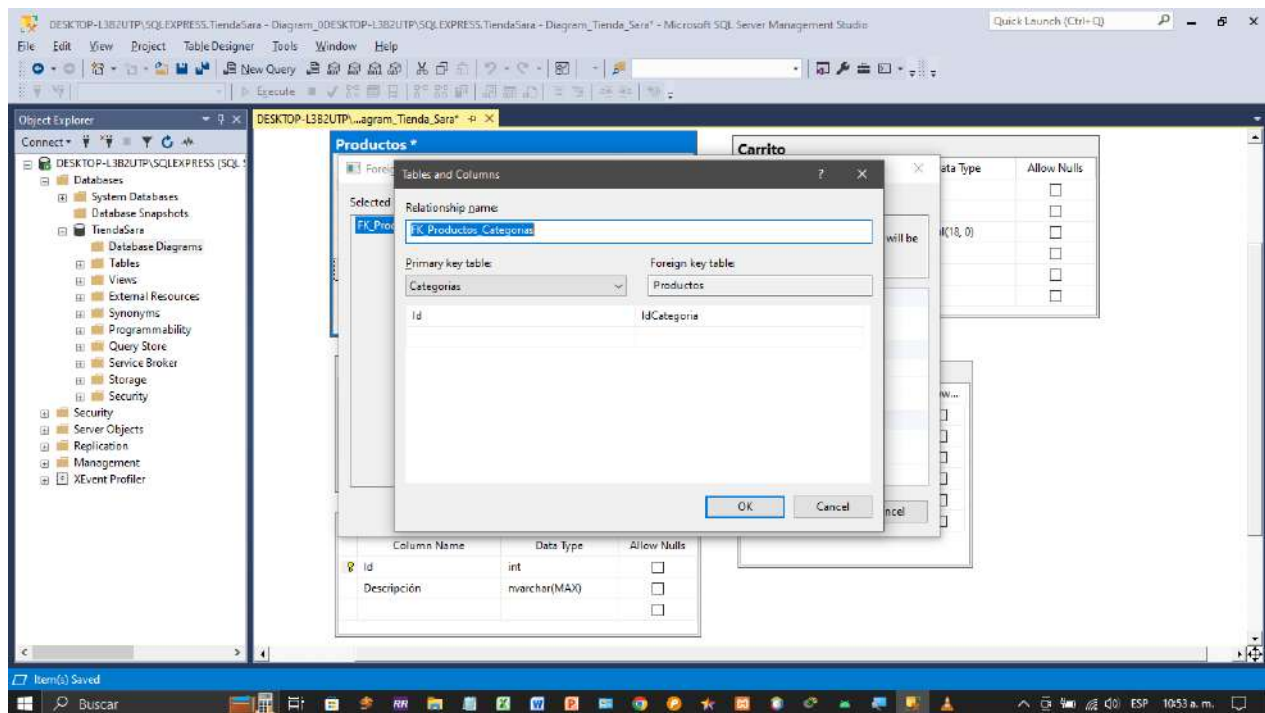


Se crea el diagrama en la BD en base a las tablas especificadas.

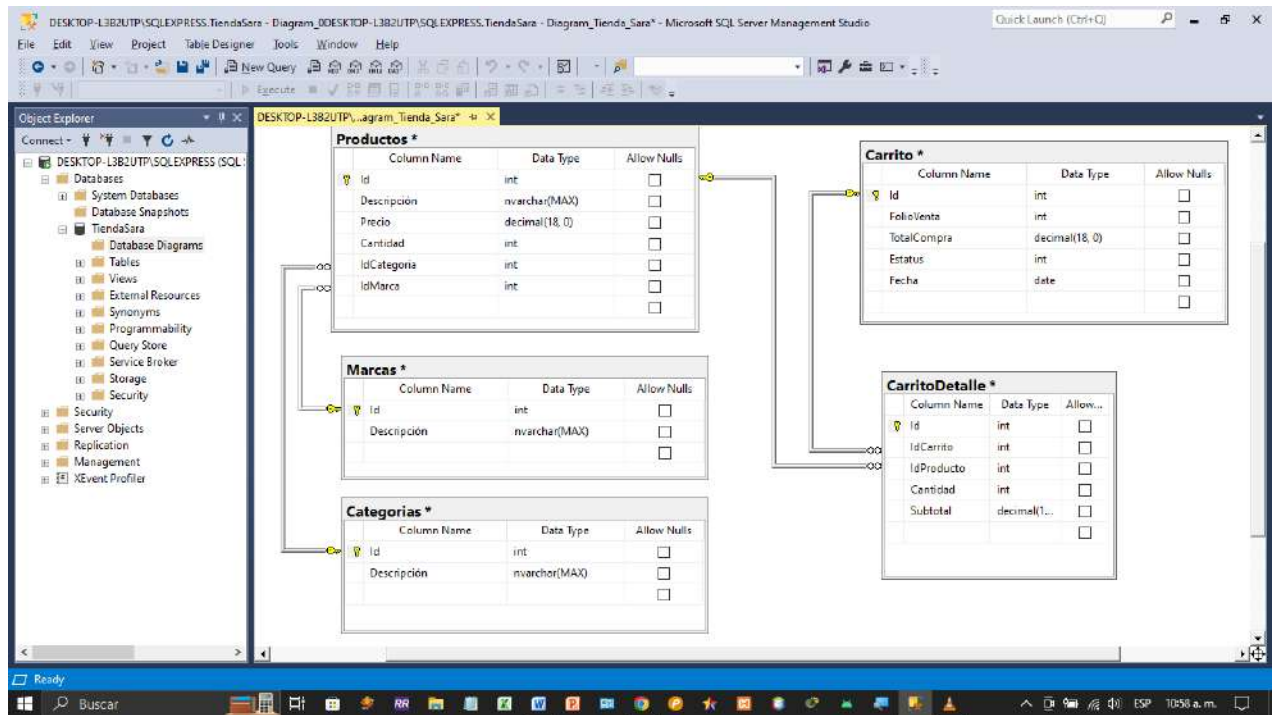




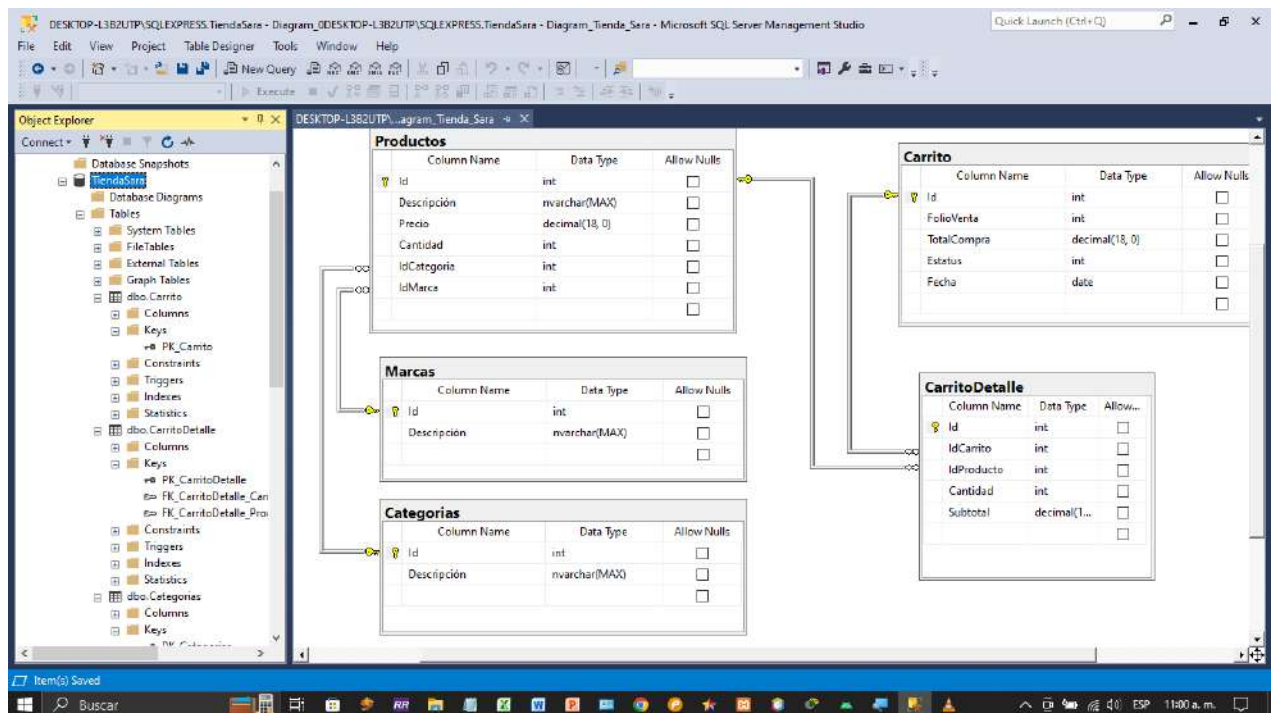
Se establecen las llaves primarias



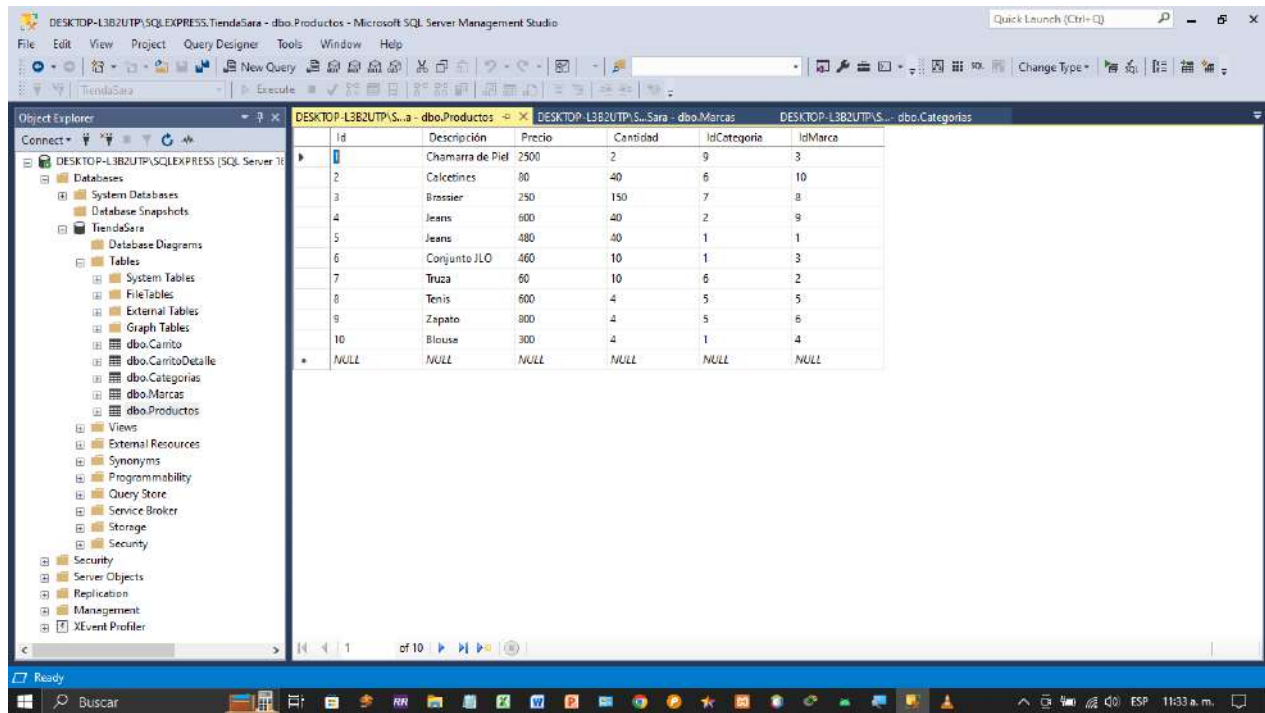
Se establecen las llaves foráneas



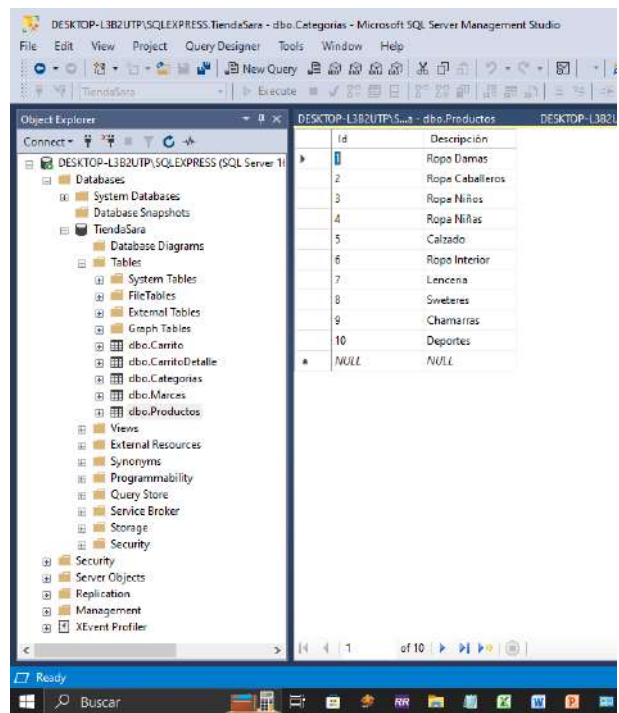
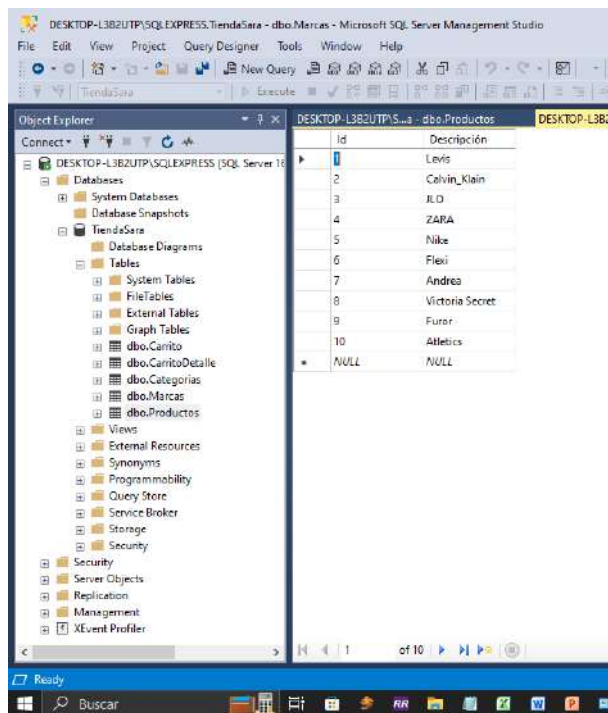
Se guarda



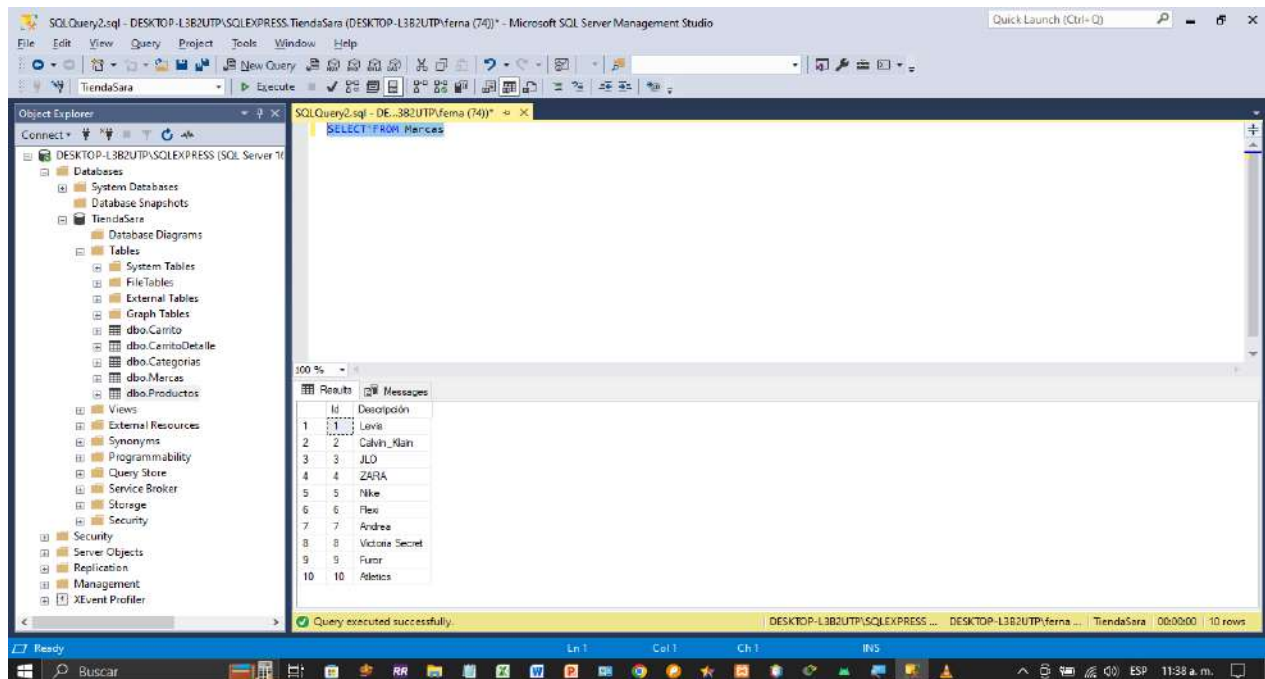
Y aparecen creadas y relacionadas en la BD en el apartado de Tablas



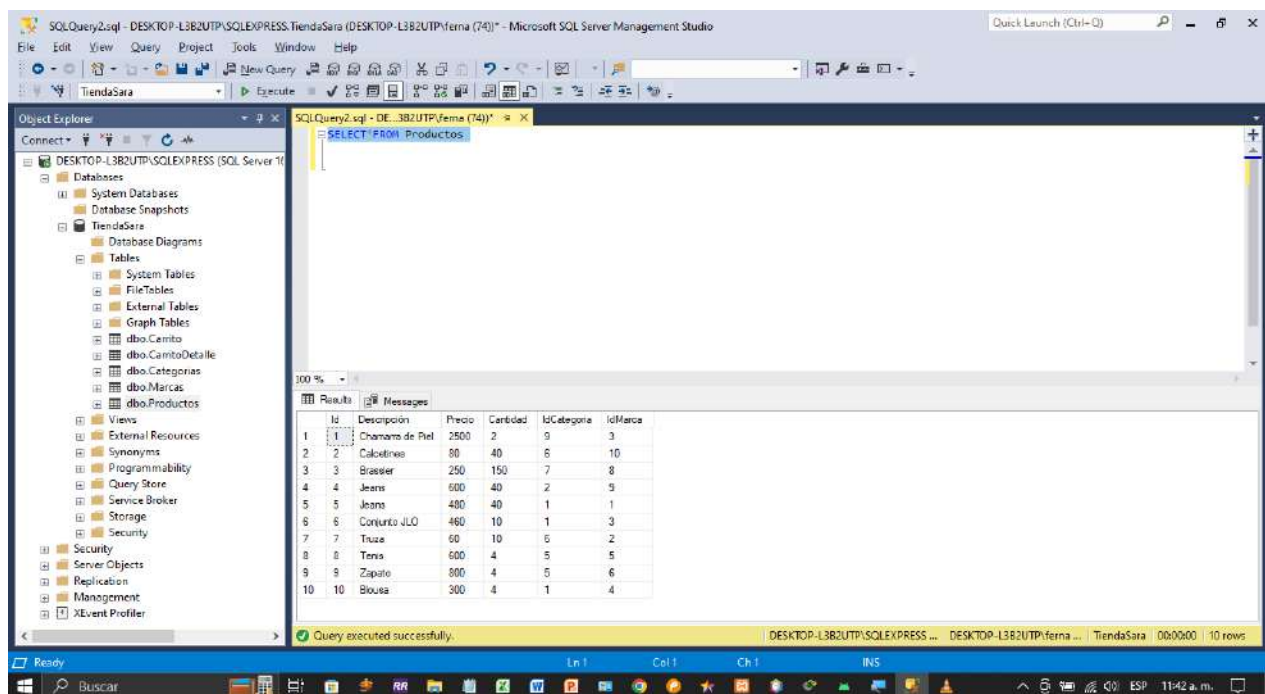
Se crean los registros en las tablas asignadas

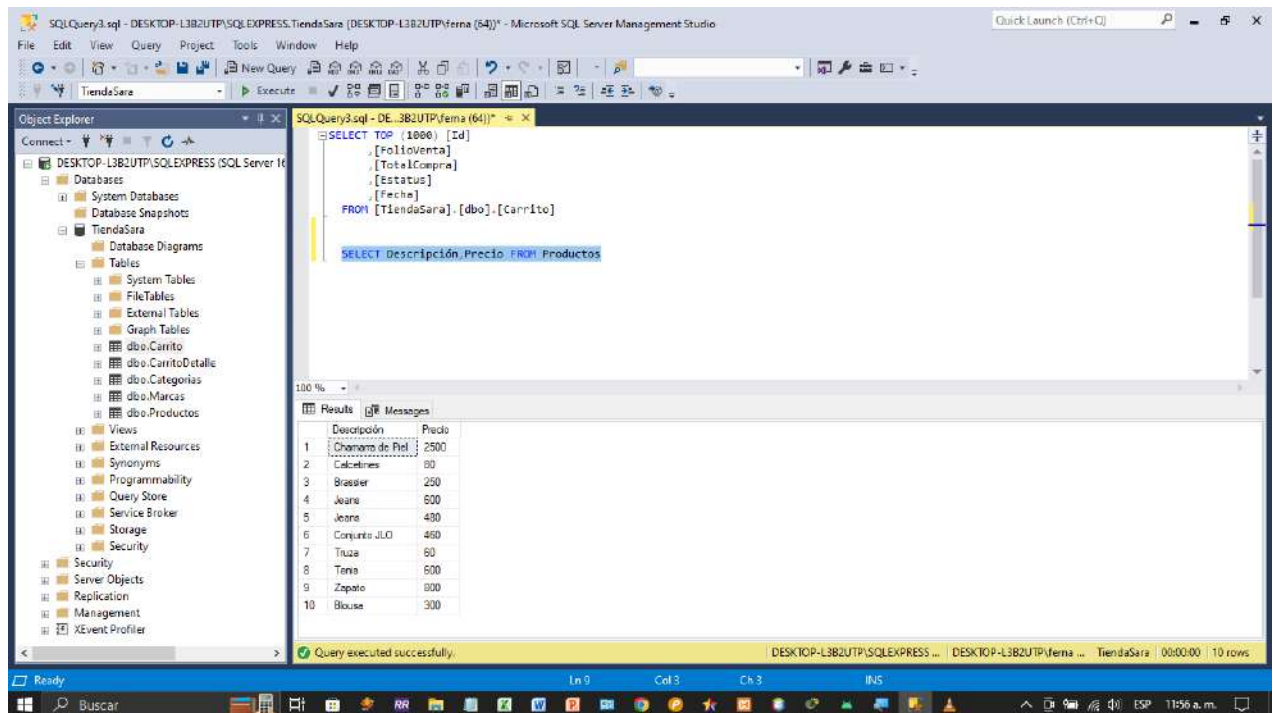
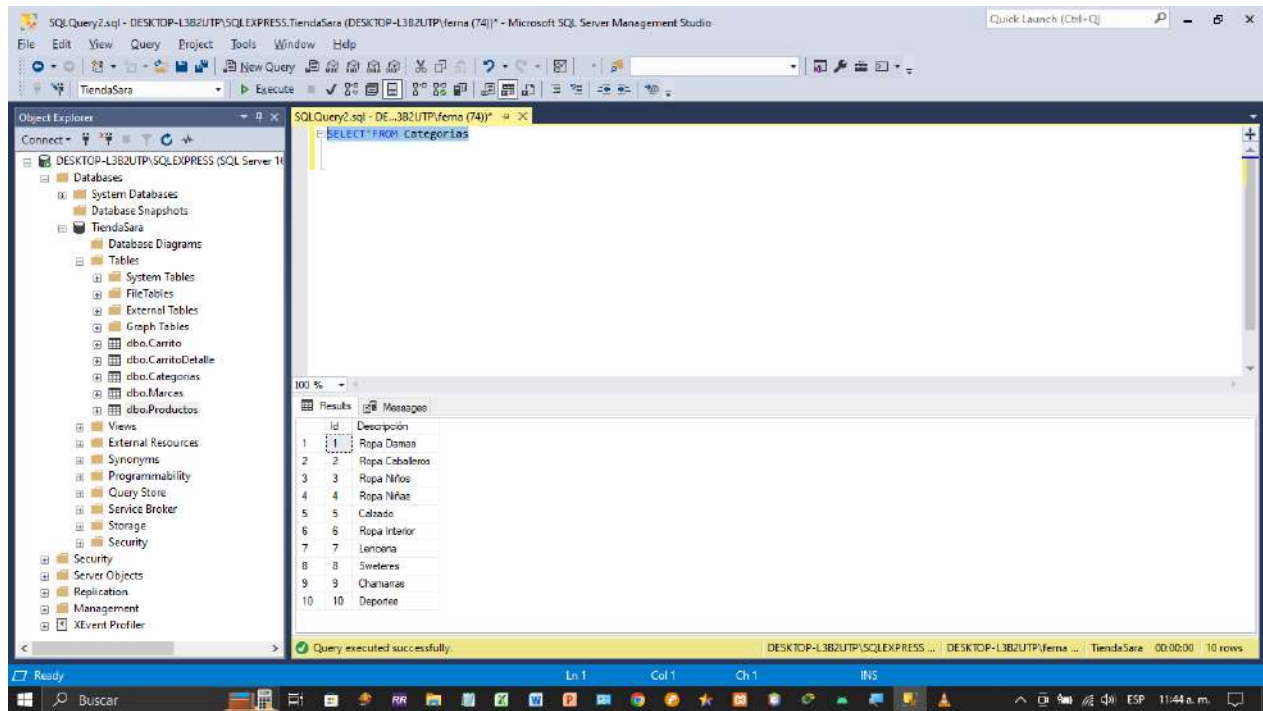


Pruebas de la BD

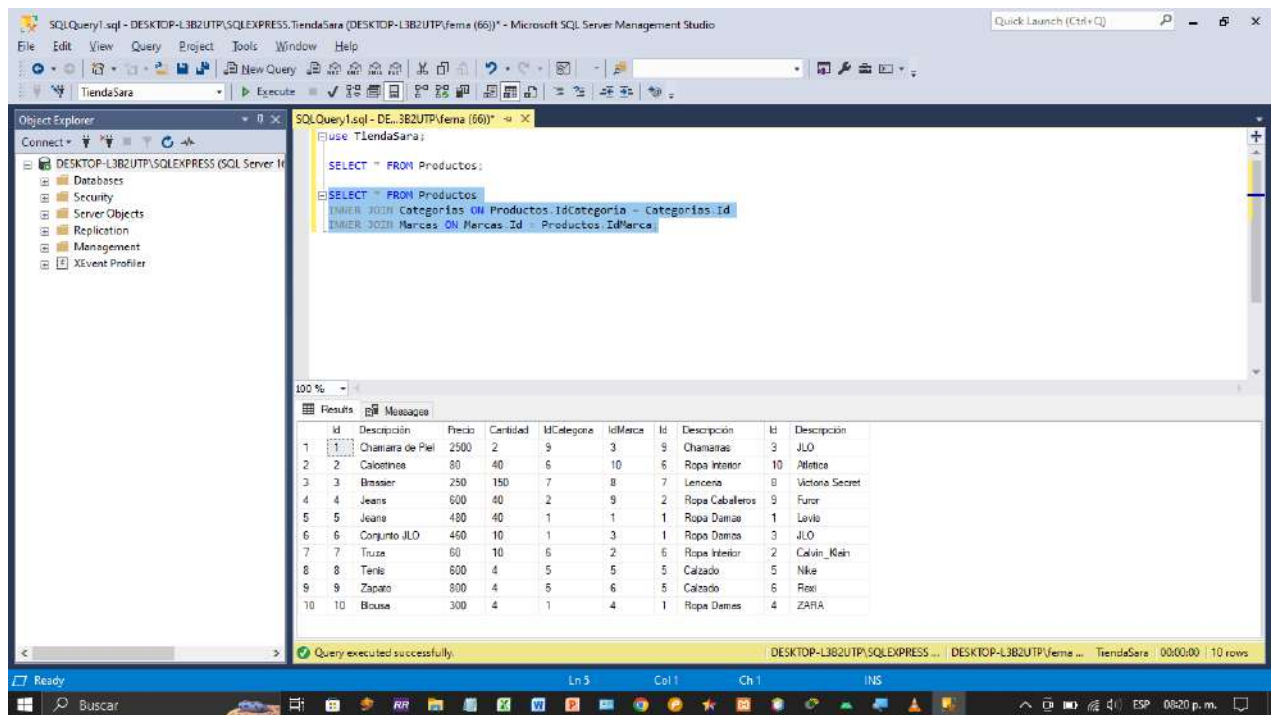


Se hacen pruebas con el comando `SELECT * FROM`

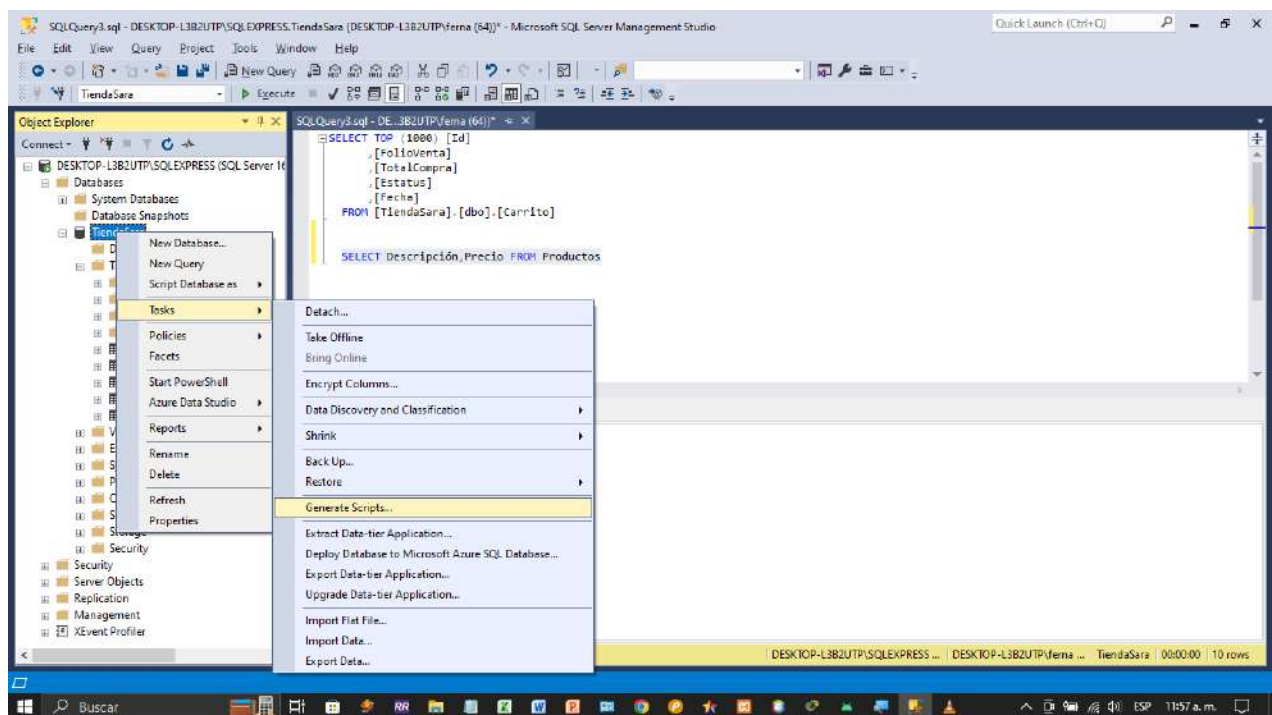




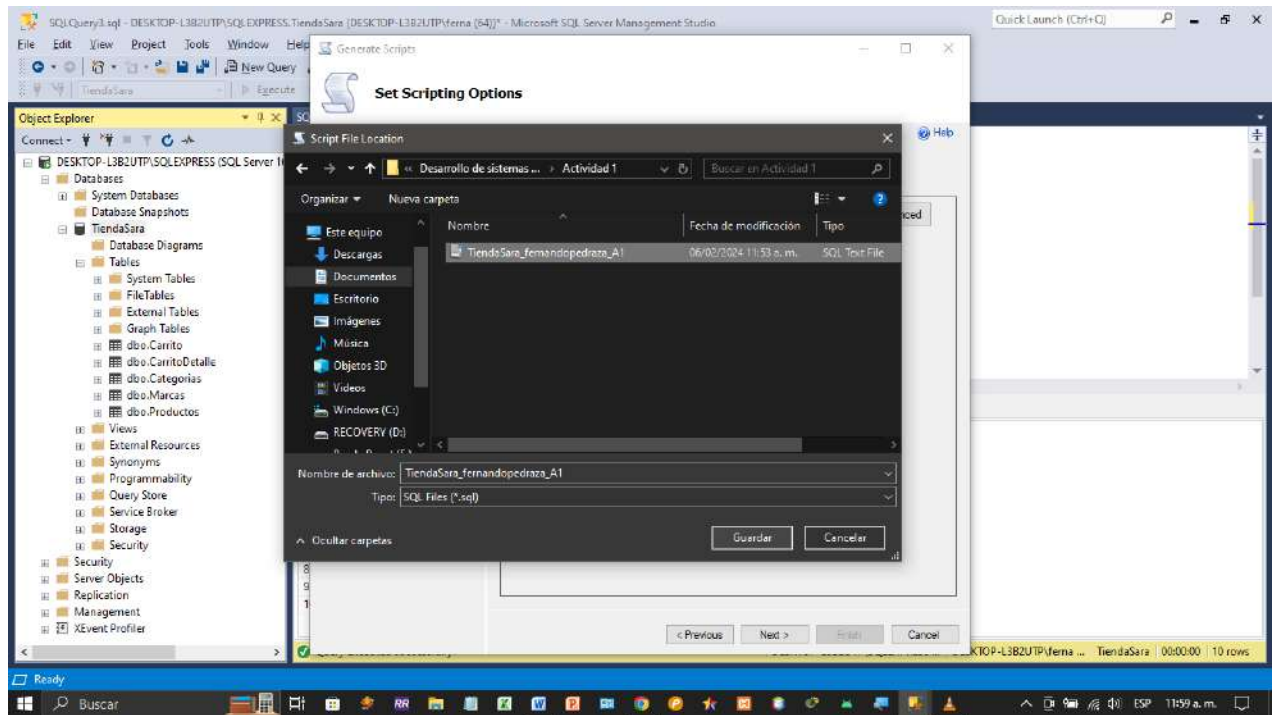
Y de forma específica según el dato que sea de interés a buscar o consultar.



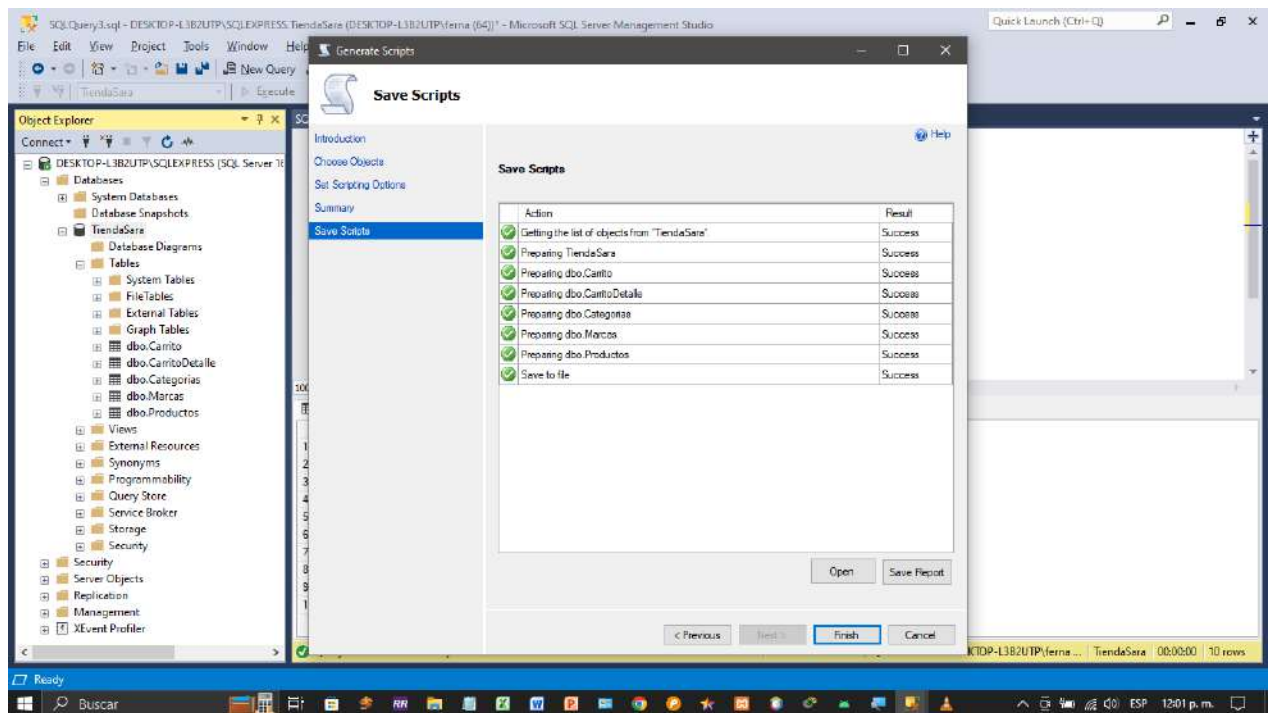
Utilizando INNER JOIN para búsquedas específicas, relacionadas entre diferentes tablas.



Se genera el Script



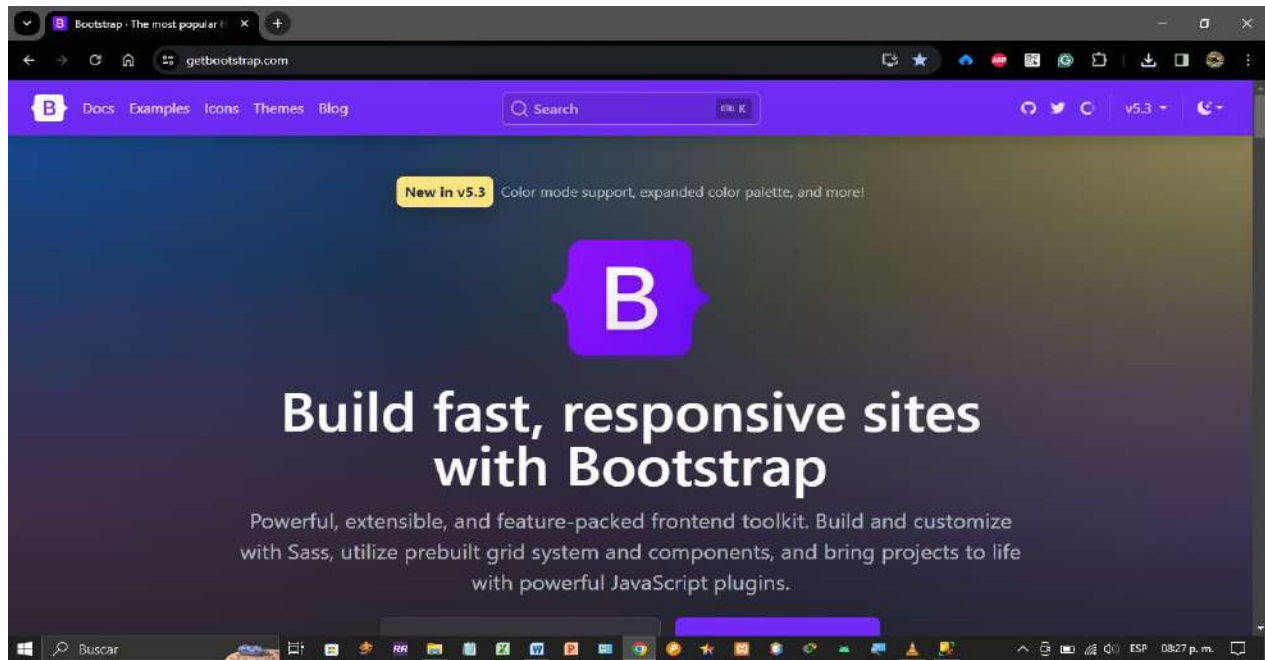
Se selecciona la ruta de destino



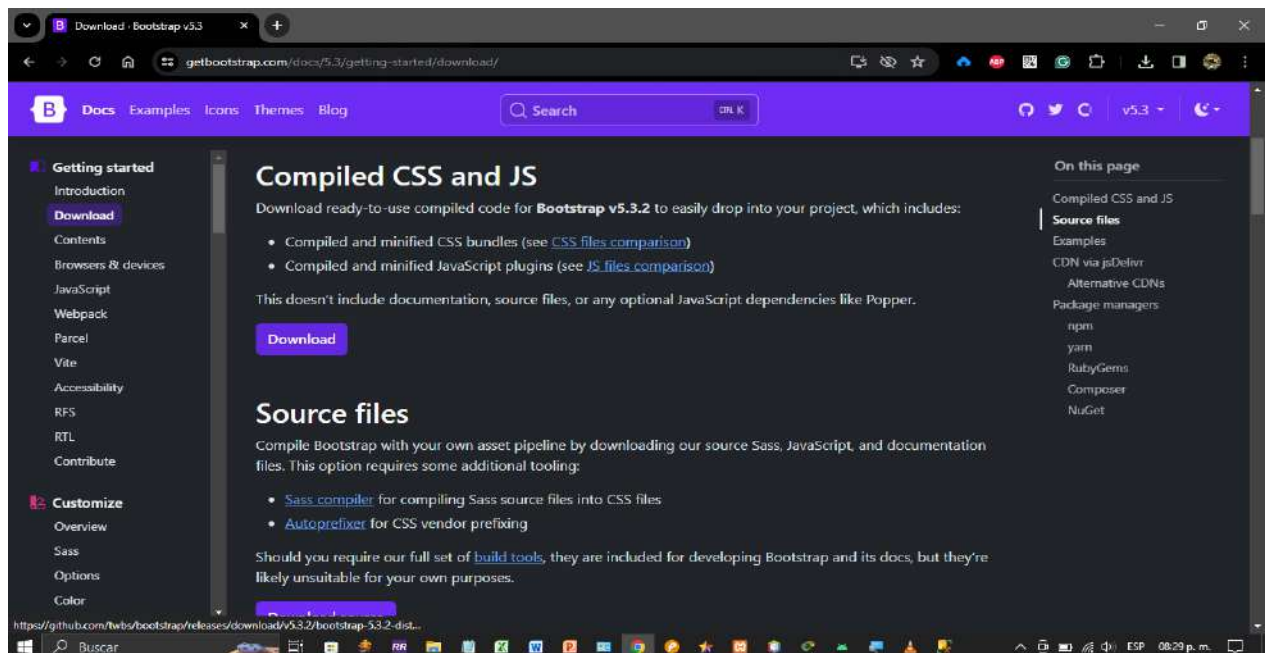
Y se finaliza para crear el Script.

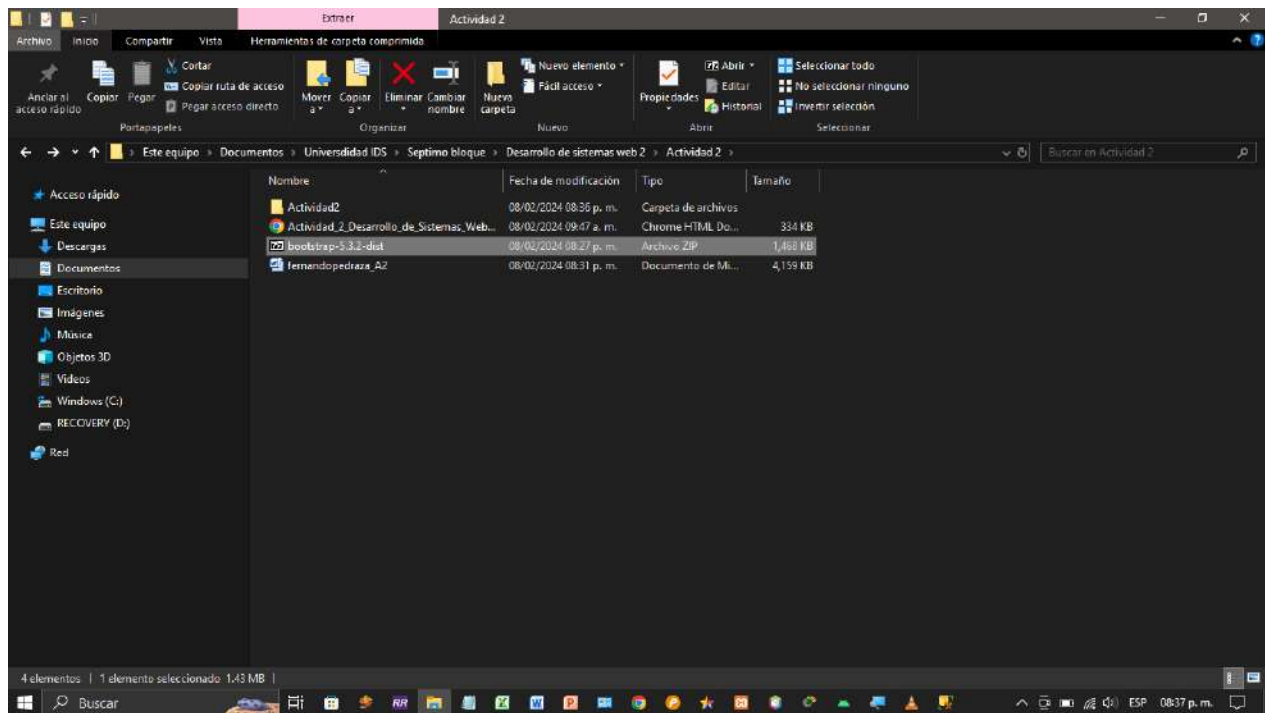
Desarrollo.

Etapa 2 – Diseño Web y Conexión

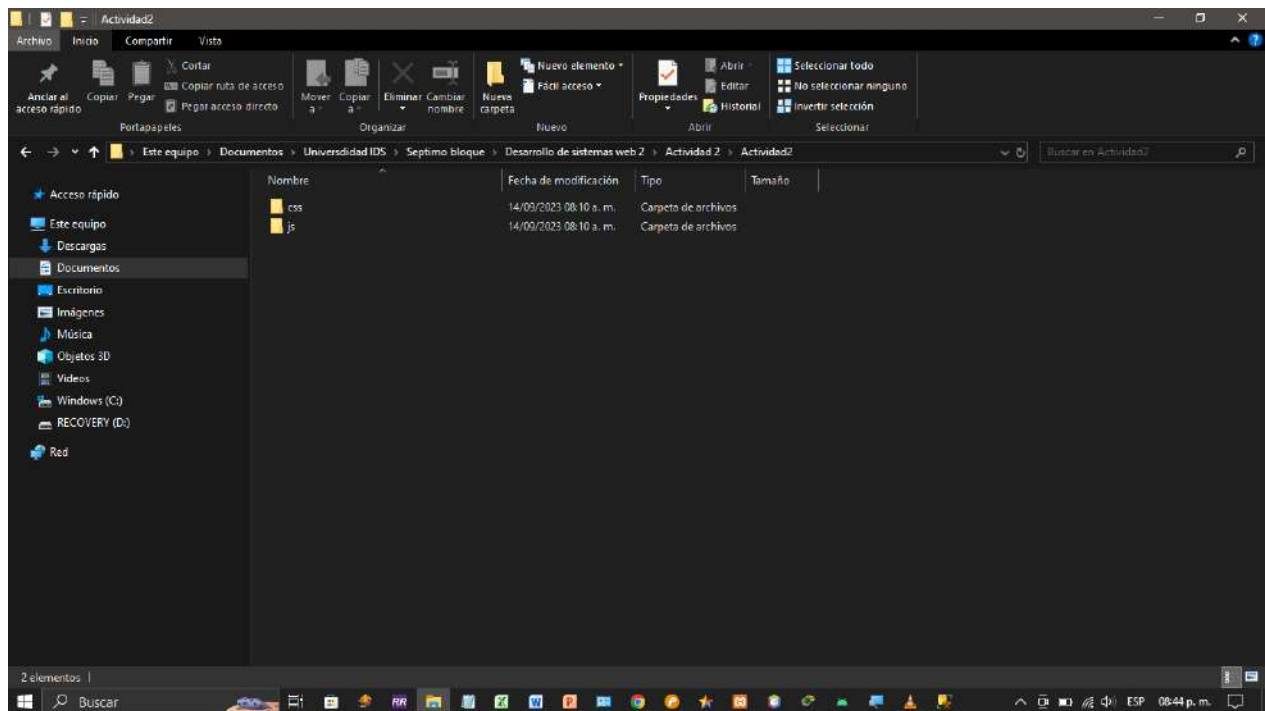


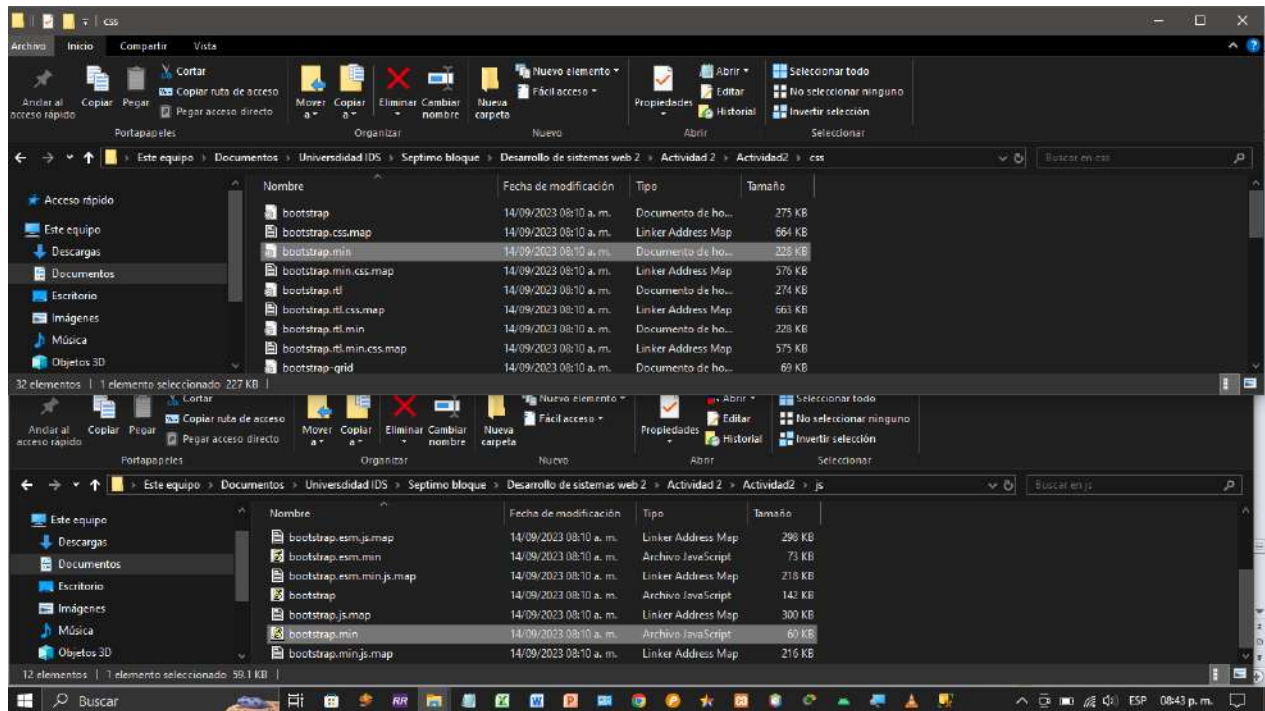
Ingresamos a Bootstrap para descargar el compilado CSS y JS para el diseño a elaborar.



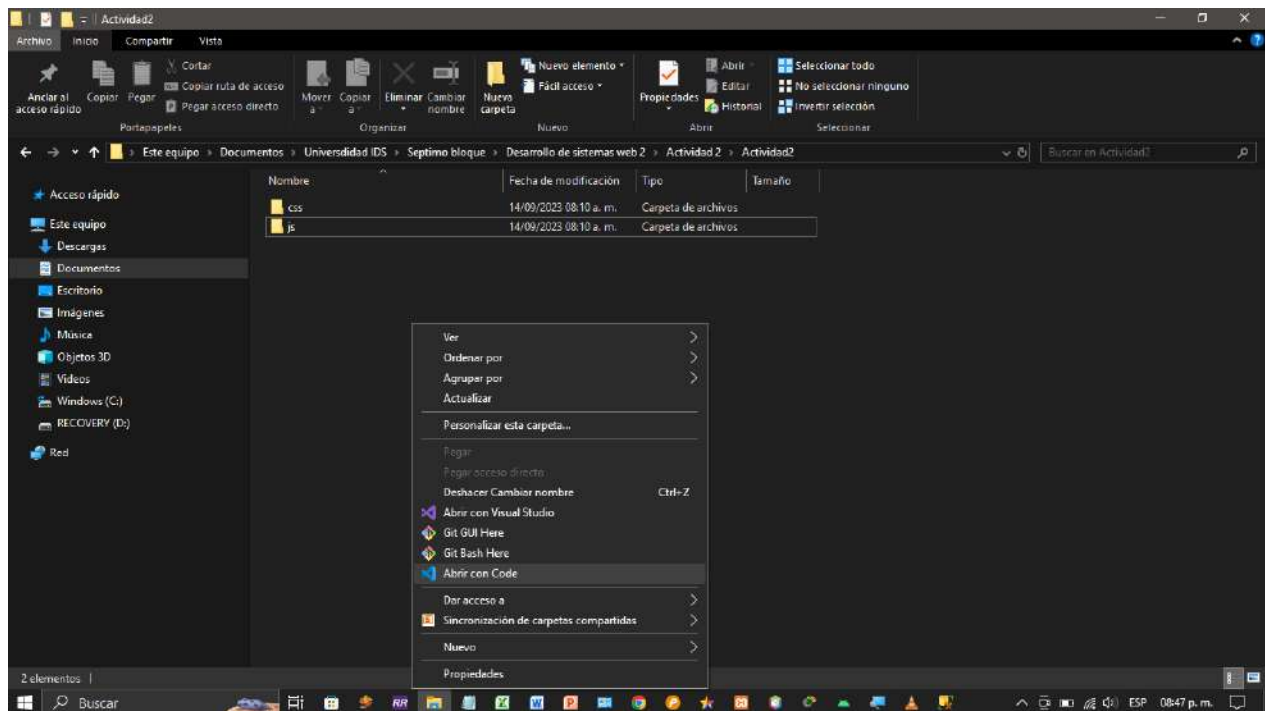


Una vez descargado se extraen los archivos y se guardan en una carpeta nombrada Actividad2.

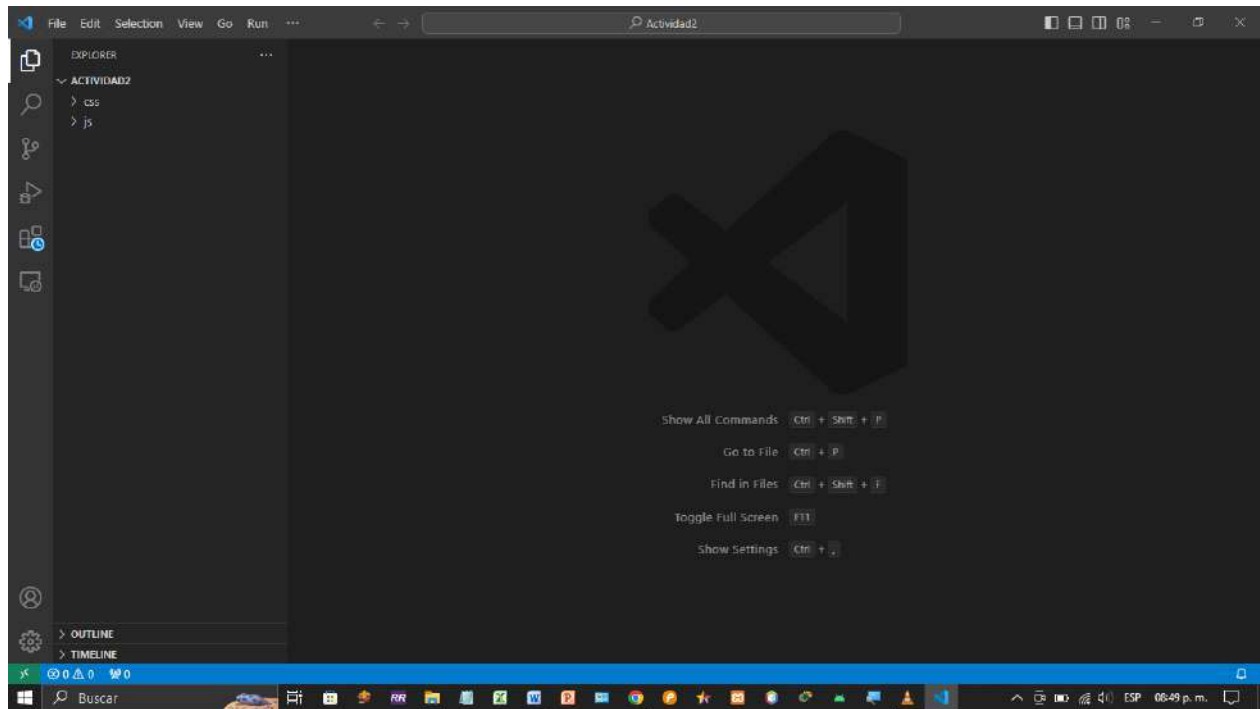




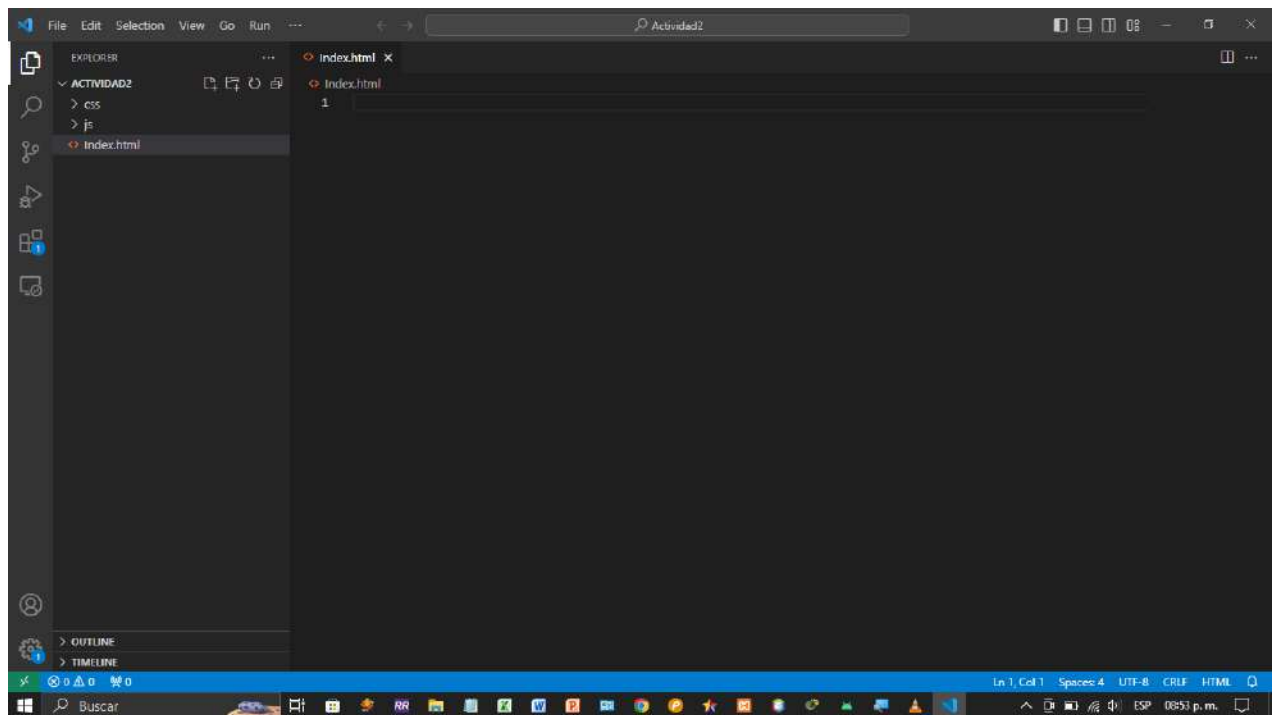
Solo se utilizara el archivo bootstrap.min de la carpeta css y js



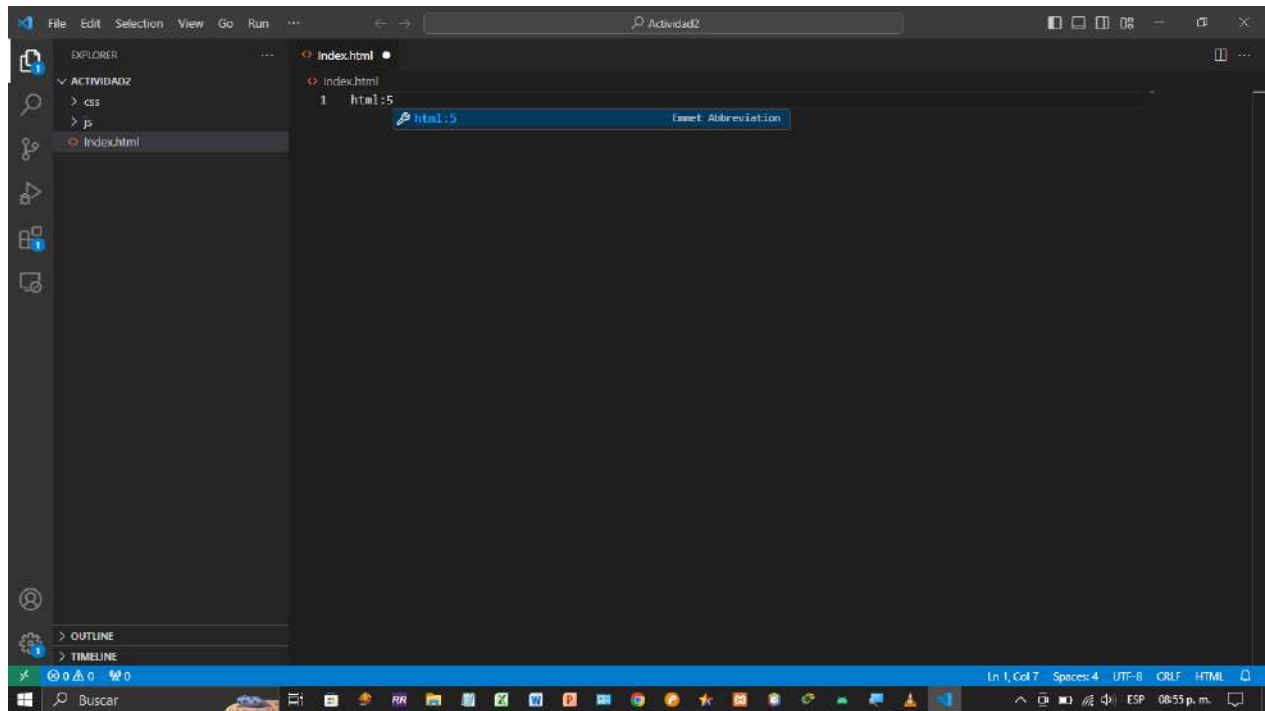
Abrimos el contenido de las carpetas con Visual Studio Code



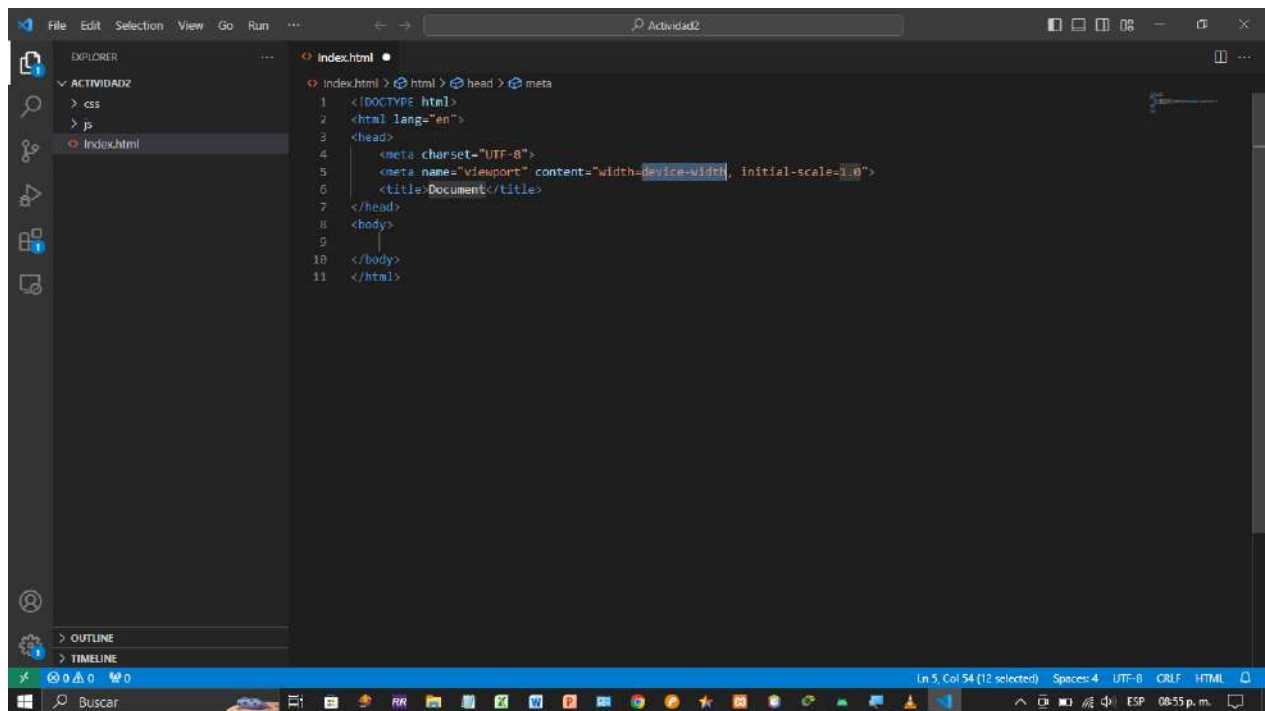
Se crea un archivo nuevo

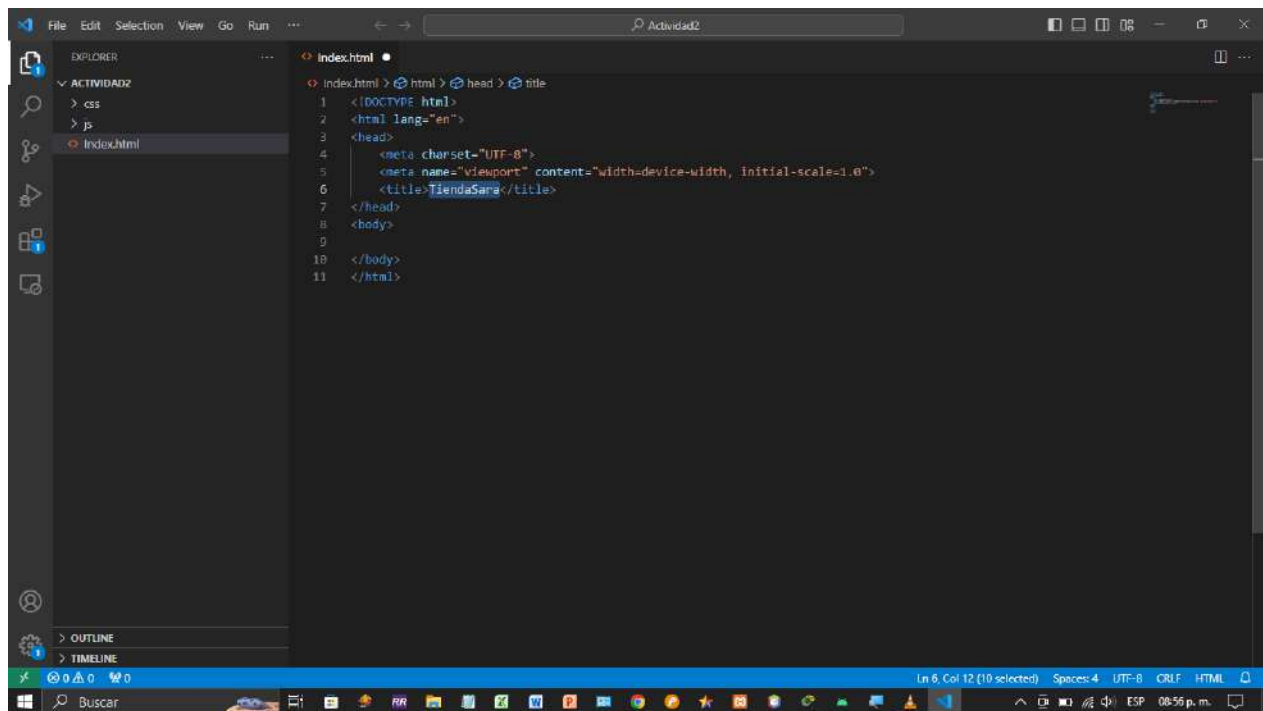


Con el nombre Index.html



Se crea la estructura básica con el comando `html:5`



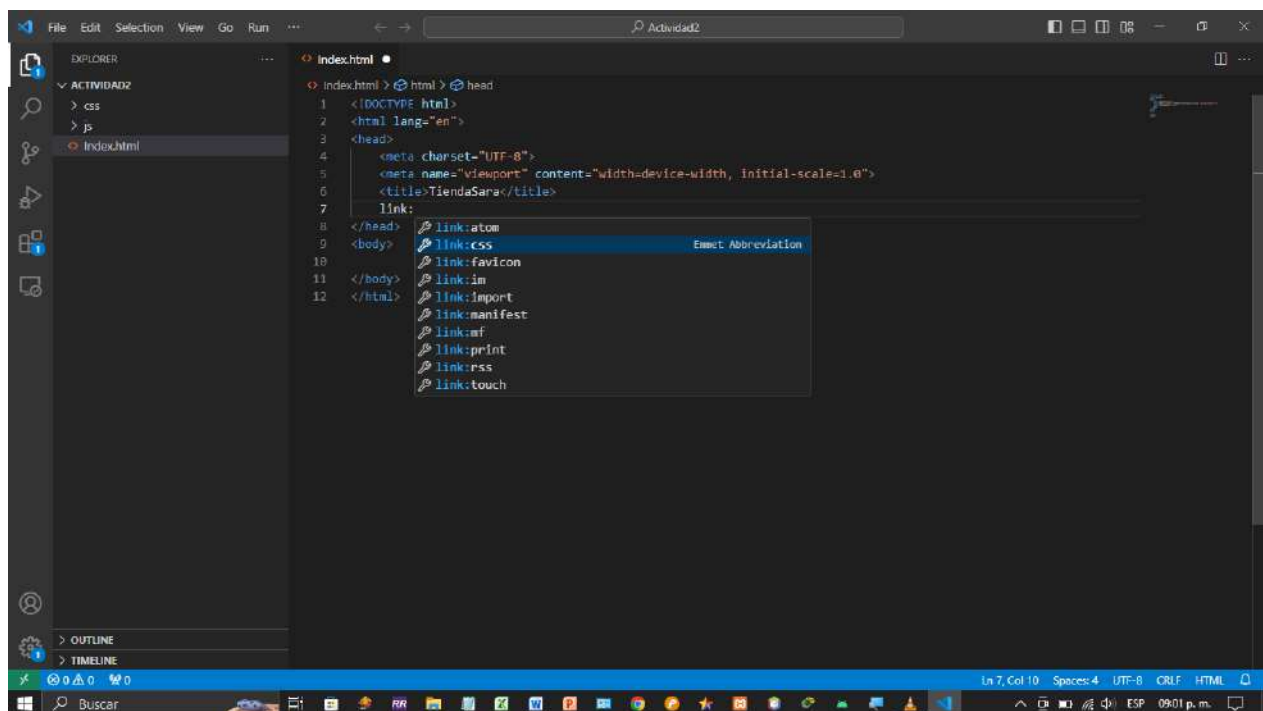


The screenshot shows the Visual Studio Code editor with a file named 'index.html' open. The Explorer sidebar on the left shows a project structure with 'ACTIVIDAD2' containing 'css', 'js', and 'index.html'. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>TiendaSara</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

The status bar at the bottom indicates 'Ln 6, Col 12 (10 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

En el apartado del encabezado se asigna el título que tendrá la página como: TiendaSara

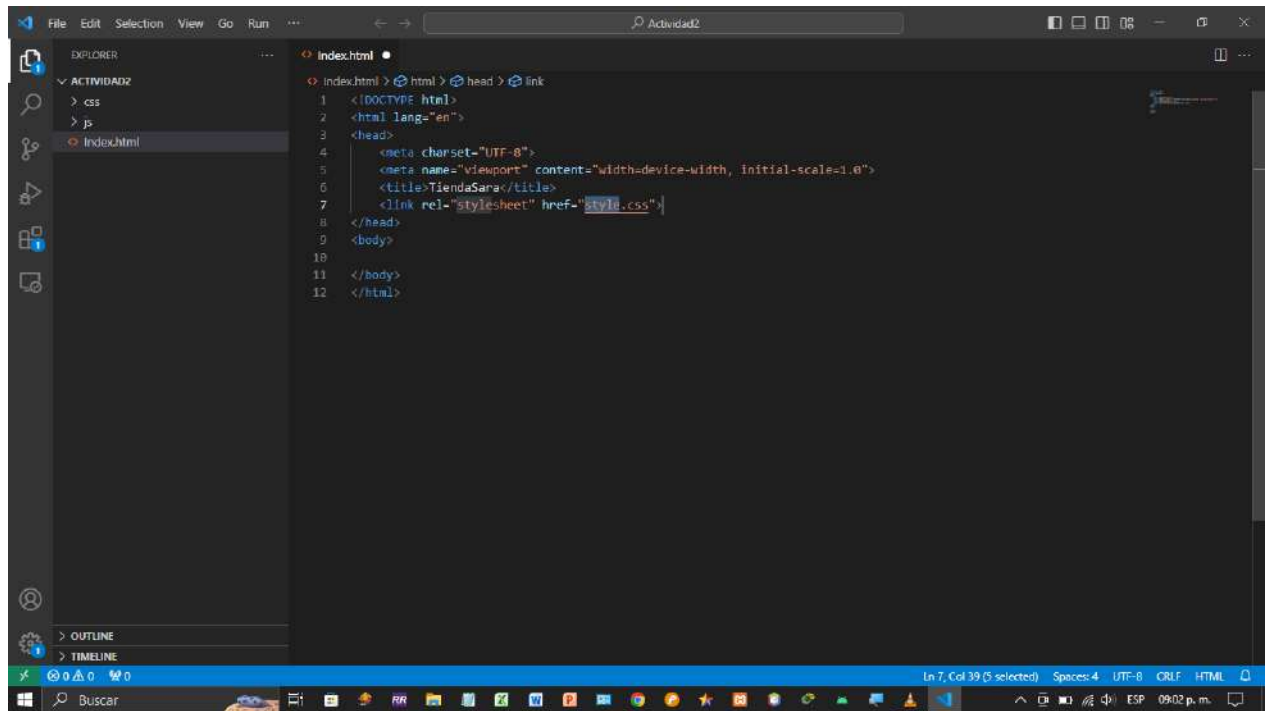


This screenshot shows the same VS Code editor with the 'index.html' file. The code is now being edited to add a link tag. The following code is visible:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>TiendaSara</title>
7   <link:
8 </head>
9 <body>
10
11 </body>
12 </html>
```

An 'Emmet Abbreviation' dropdown menu is open, showing a list of link types: 'link:atom', 'link:css' (which is highlighted), 'link:favicon', 'link:im', 'link:import', 'link:manifest', 'link:mf', 'link:print', 'link:rss', and 'link:touch'. The status bar at the bottom shows 'Ln 7, Col 10', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

Posteriormente se agrega una etiqueta link como: link:css

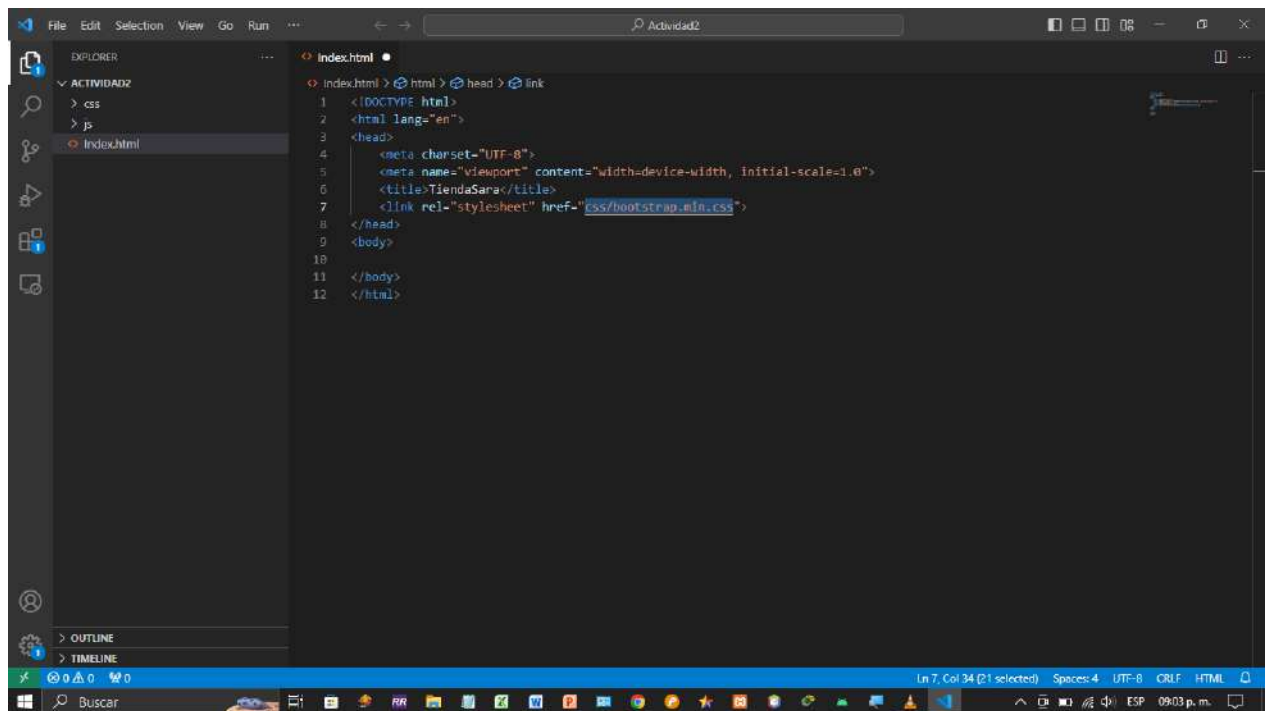


A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'ACTIVIDAD2' with subfolders 'css' and 'js', and a file 'Index.html'. The main editor area displays the content of 'Index.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>TiendaSana</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10
11 </body>
12 </html>
```

The status bar at the bottom indicates 'Ln 7, Col 39 (5 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

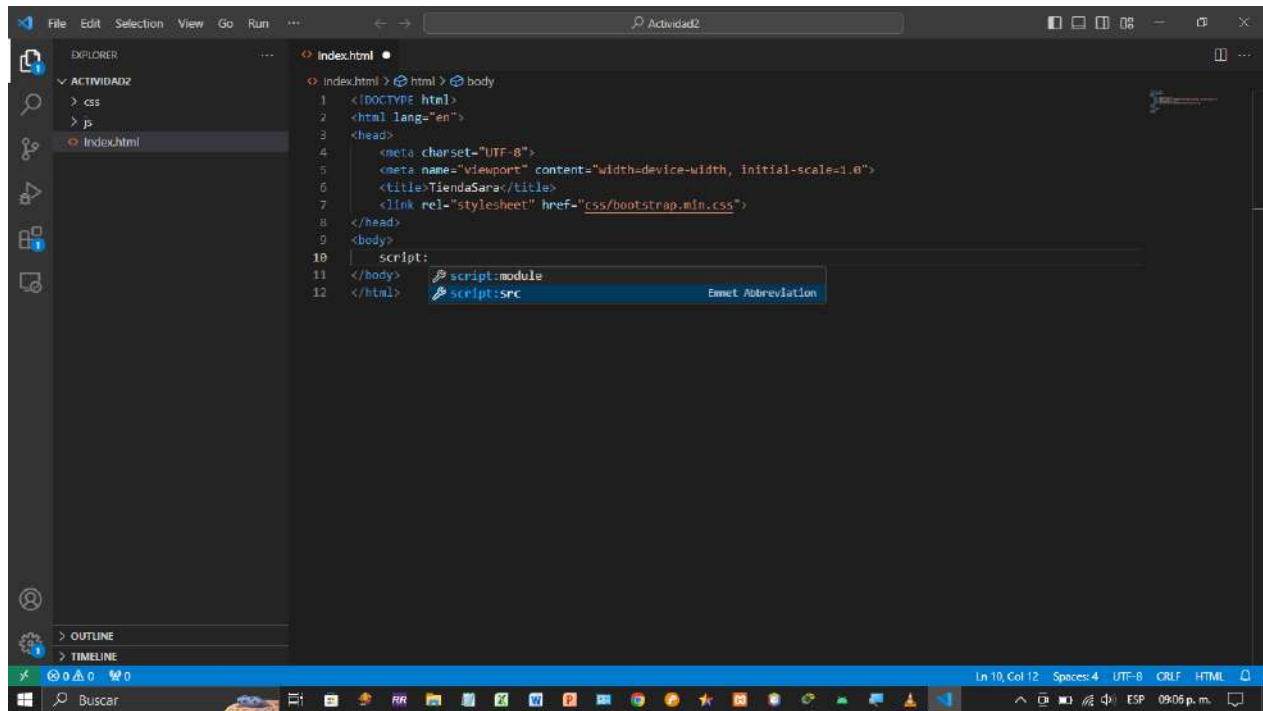
En href se cambia style.css por css/bootstrap.min.css



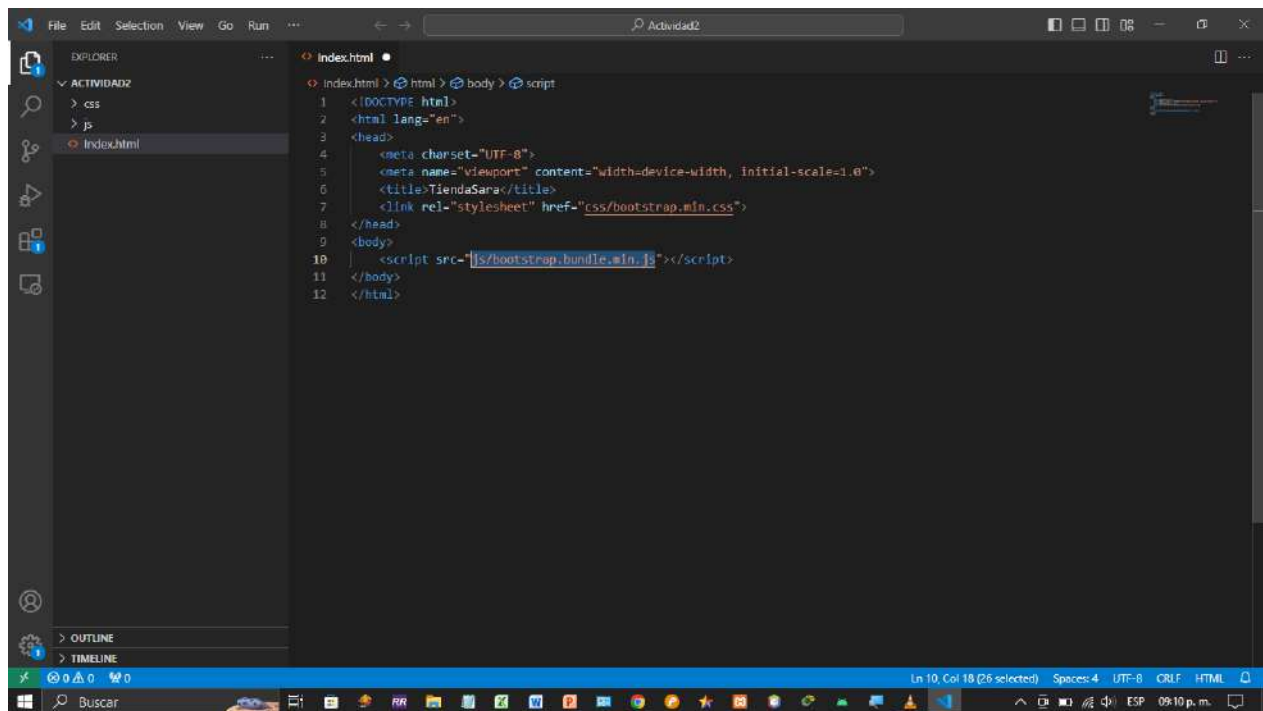
A screenshot of the Visual Studio Code editor interface, similar to the one above. The Explorer sidebar shows the same project structure. The main editor area displays the content of 'Index.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>TiendaSana</title>
7   <link rel="stylesheet" href="css/bootstrap.min.css">
8 </head>
9 <body>
10
11 </body>
12 </html>
```

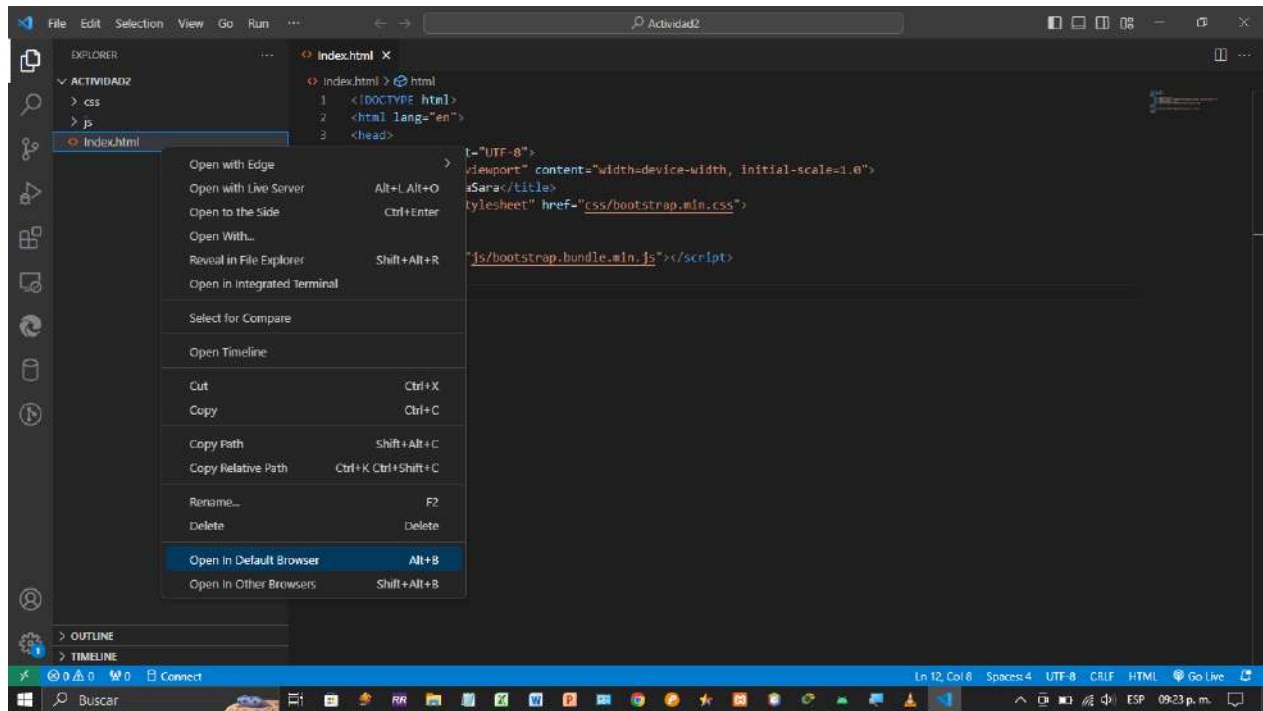
The status bar at the bottom indicates 'Ln 7, Col 34 (21 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.



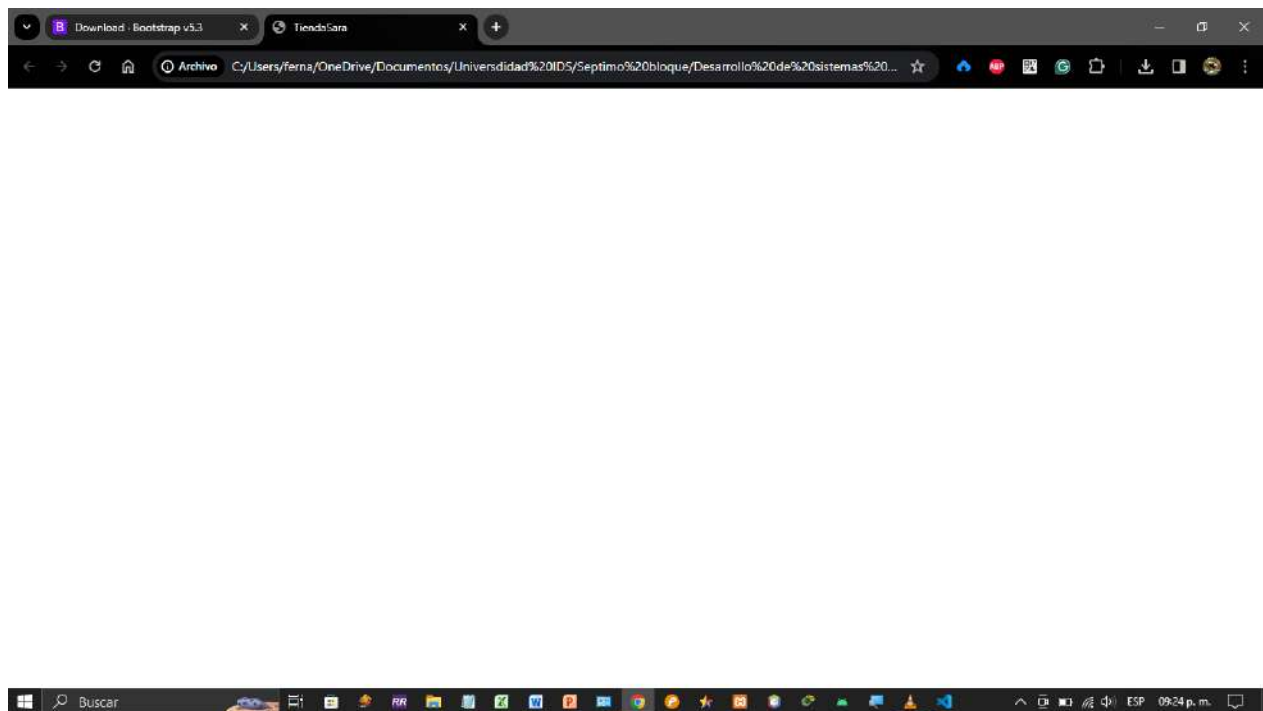
En la parte del cuerpo se agrega un script como: script:src



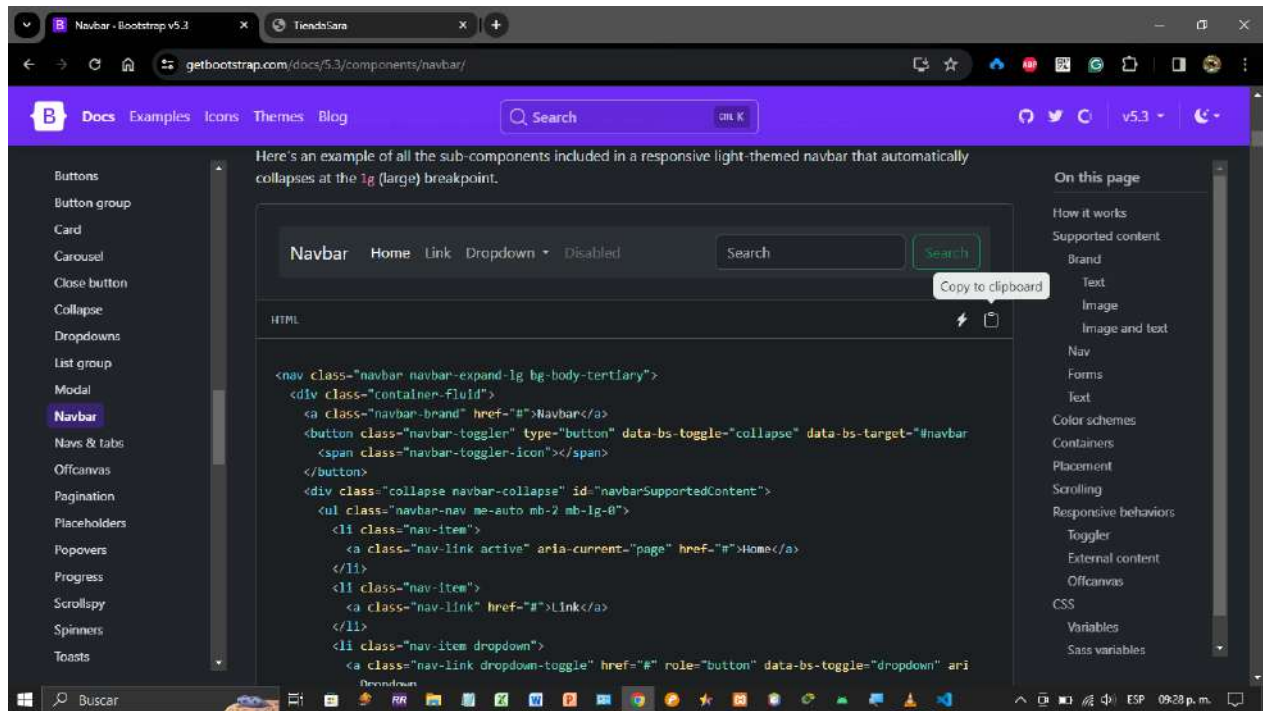
Y se le asigna el nombre de: js/bootstrap.bundle.min.js



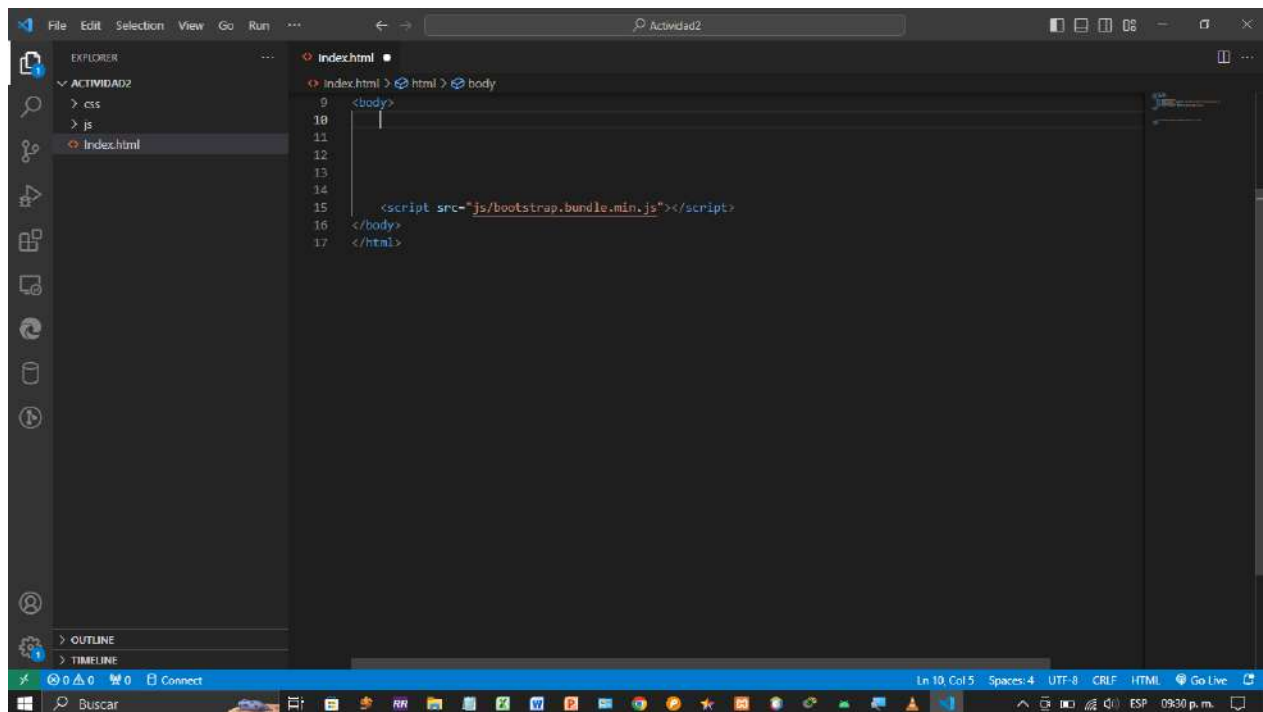
Se verifica que abra en el navegador

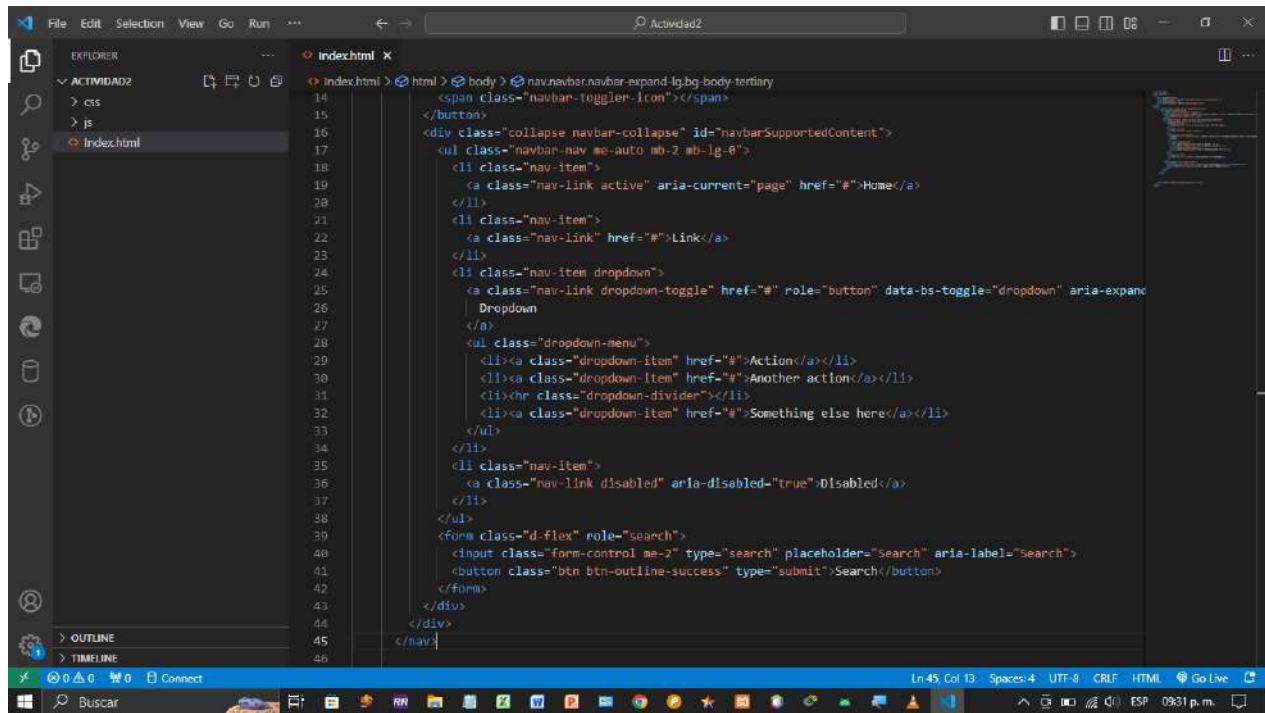


Obteniendo un lienzo en blanco con el título de TiendaSara



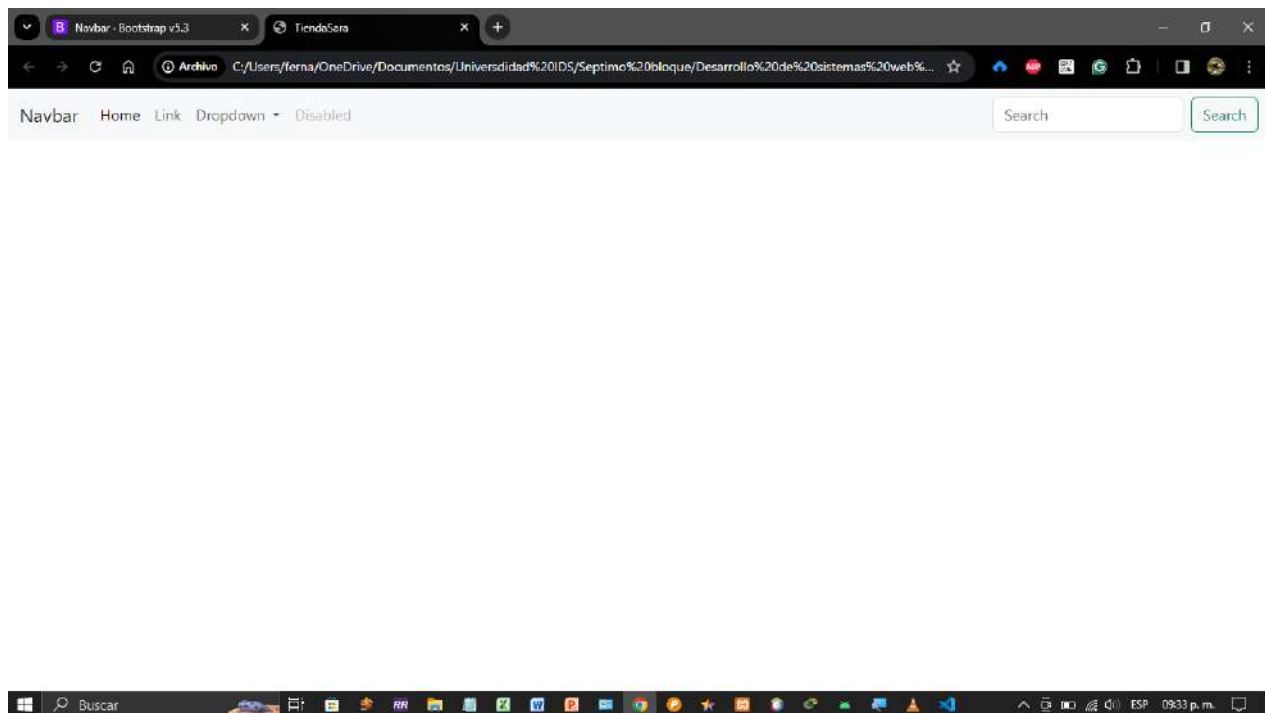
De la página de Bootstrap se copia el código que mostrara la barra de navegación en la página a diseñar y se pega en el apartado del cuerpo del archivo html.

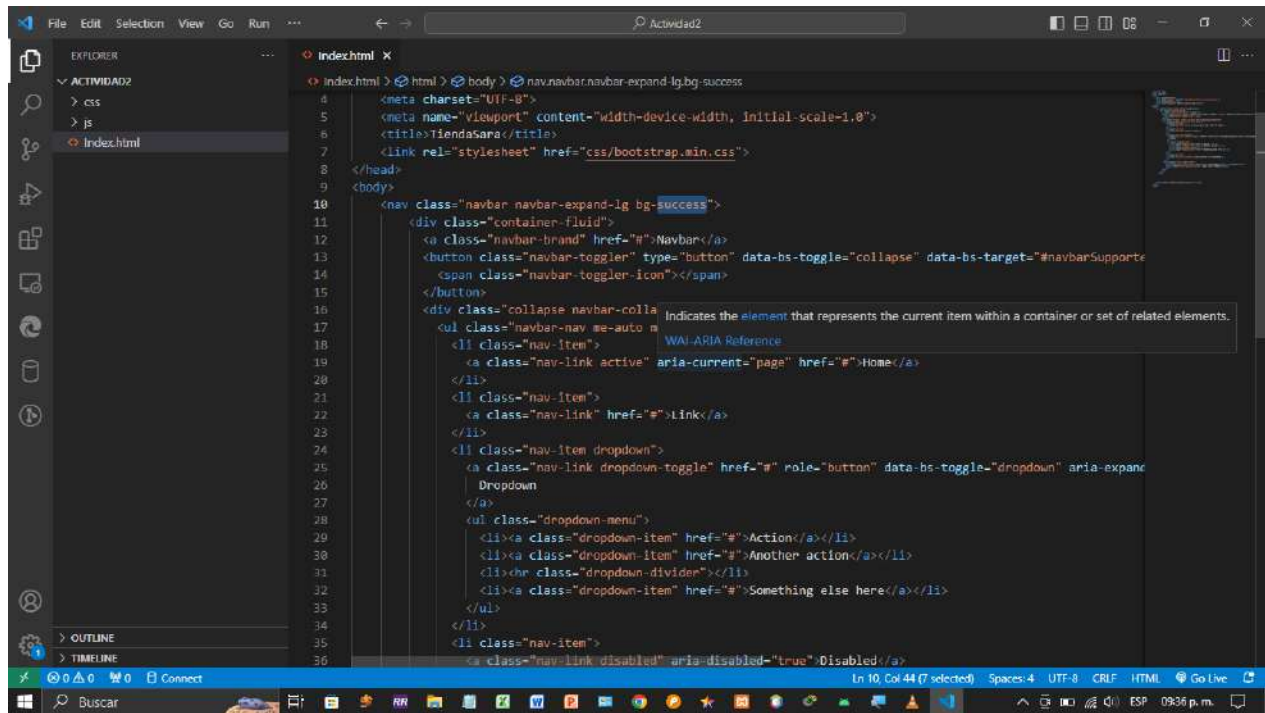




```
14 <span class="navbar-toggler-icon"></span>
15 </button>
16 <div class="collapse navbar-collapse" id="navbarSupportedContent">
17 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
18 <li class="nav-item">
19 <a class="nav-link active" aria-current="page" href="#">Home</a>
20 </li>
21 <li class="nav-item">
22 <a class="nav-link" href="#">Link</a>
23 </li>
24 <li class="nav-item dropdown">
25 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
26   Dropdown
27 </a>
28 <ul class="dropdown-menu">
29 <li><a class="dropdown-item" href="#">Action</a></li>
30 <li><a class="dropdown-item" href="#">Another action</a></li>
31 <li><hr class="dropdown-divider"></li>
32 <li><a class="dropdown-item" href="#">Something else here</a></li>
33 </ul>
34 </li>
35 <li class="nav-item">
36 <a class="nav-link disabled" aria-disabled="true">Disabled</a>
37 </li>
38 </ul>
39 <form class="d-flex" role="search">
40 <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
41 <button class="btn btn-outline-success" type="submit">Search</button>
42 </form>
43 </div>
44 </nav>
45 </div>
46 </div>
```

Para visualizarse en el navegador

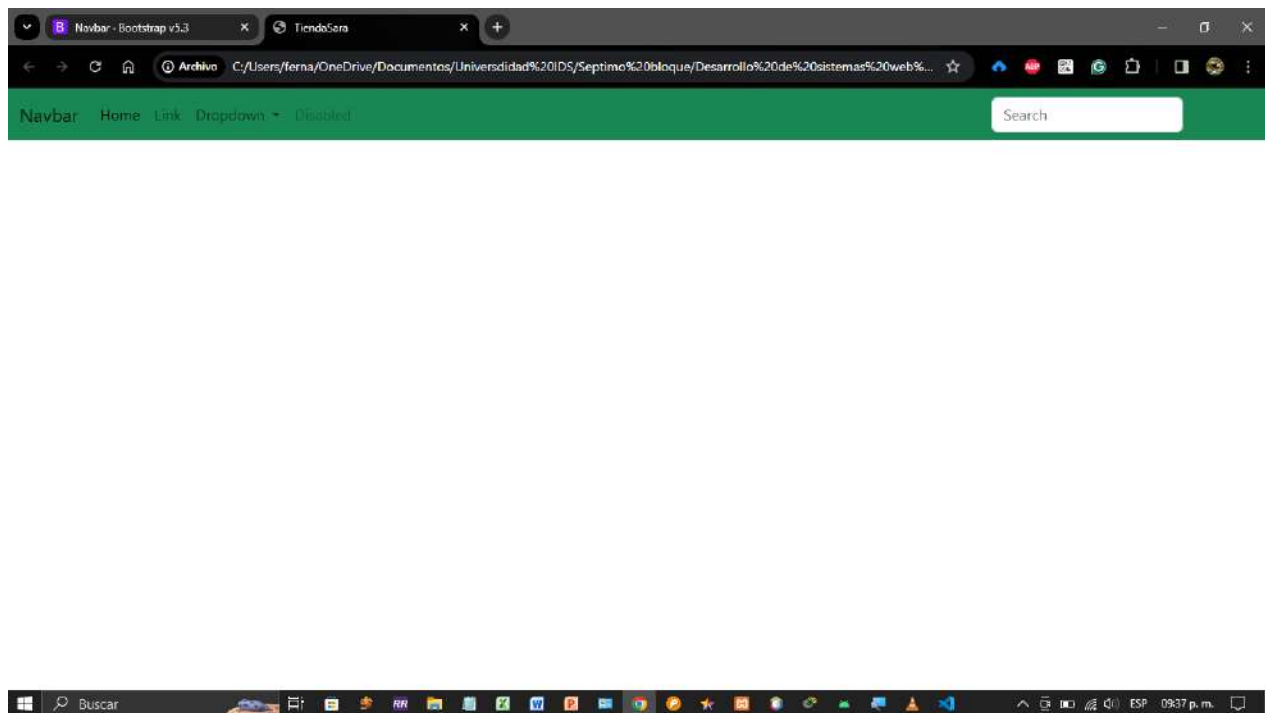


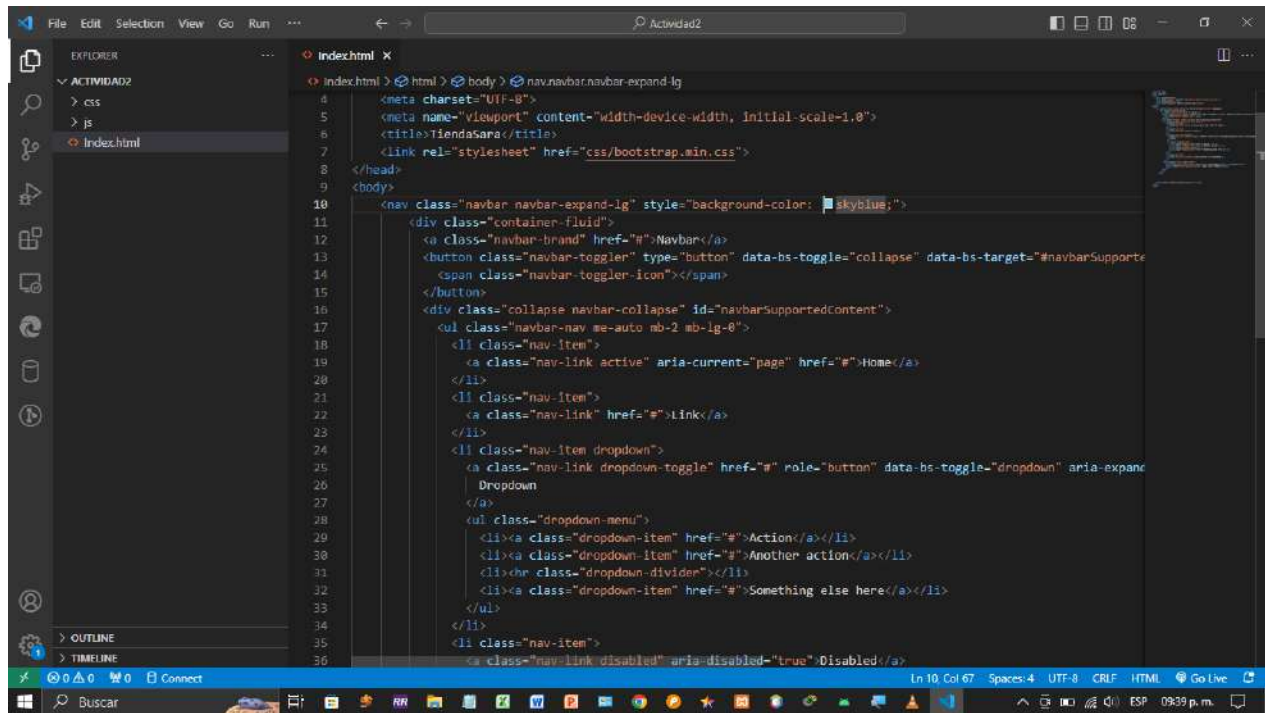


```
index.html > html > body > nav.navbar.navbar-expand-lg.bg-success
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>TiendaSara</title>
7 <link rel="stylesheet" href="css/bootstrap.min.css">
8 </head>
9 <body>
10 <nav class="navbar navbar-expand-lg bg-success">
11   <div class="container-fluid">
12     <a class="navbar-brand" href="#">Navbar</a>
13     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupport">
14       <span class="navbar-toggler-icon"></span>
15     </button>
16     <div class="collapse navbar-collapse">
17       <ul class="navbar-nav me-auto">
18         <li class="nav-item">
19           <a class="nav-link active" aria-current="page" href="#">Home</a>
20         </li>
21         <li class="nav-item">
22           <a class="nav-link" href="#">Link</a>
23         </li>
24         <li class="nav-item dropdown">
25           <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
26             Dropdown
27           </a>
28           <ul class="dropdown-menu">
29             <li><a class="dropdown-item" href="#">Action</a></li>
30             <li><a class="dropdown-item" href="#">Another action</a></li>
31             <li><hr class="dropdown-divider"></li>
32             <li><a class="dropdown-item" href="#">Something else here</a></li>
33           </ul>
34         </li>
35         <li class="nav-item">
36           <a class="nav-link disabled" aria-disabled="true">Disabled</a>
```

Indicates the element that represents the current item within a container or set of related elements.

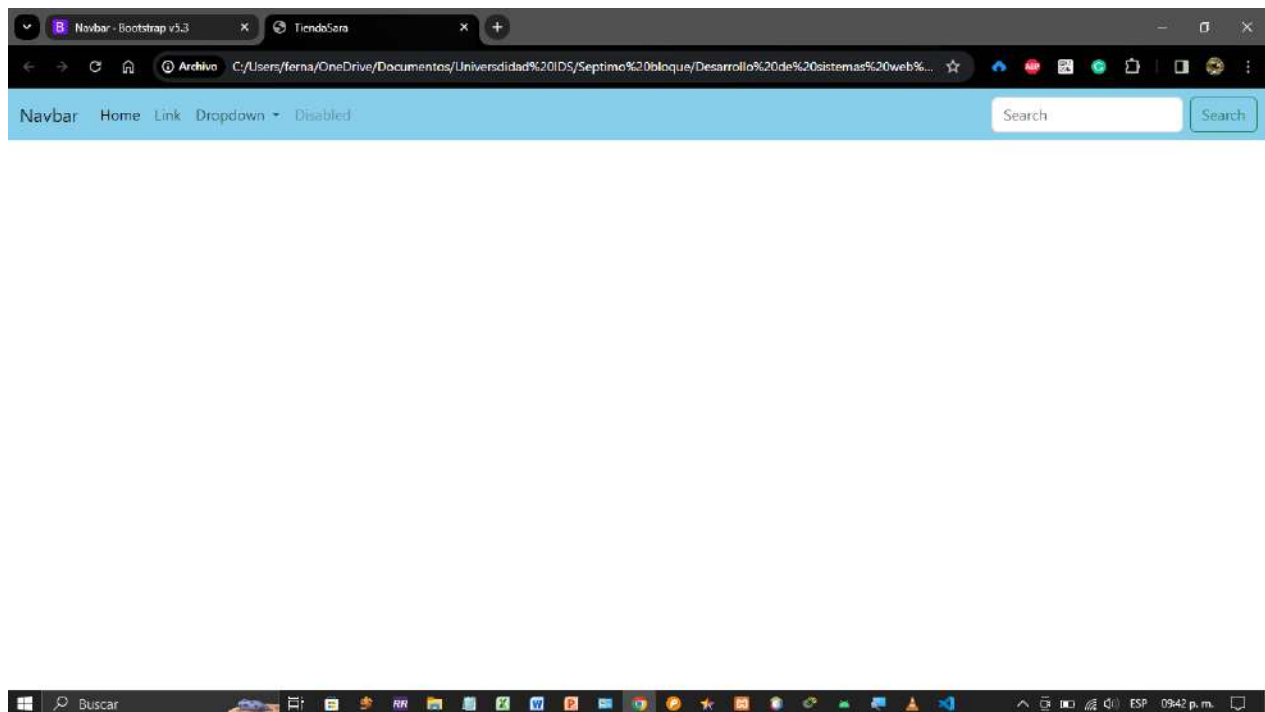
Se cambia el color de la barra de navegación a success

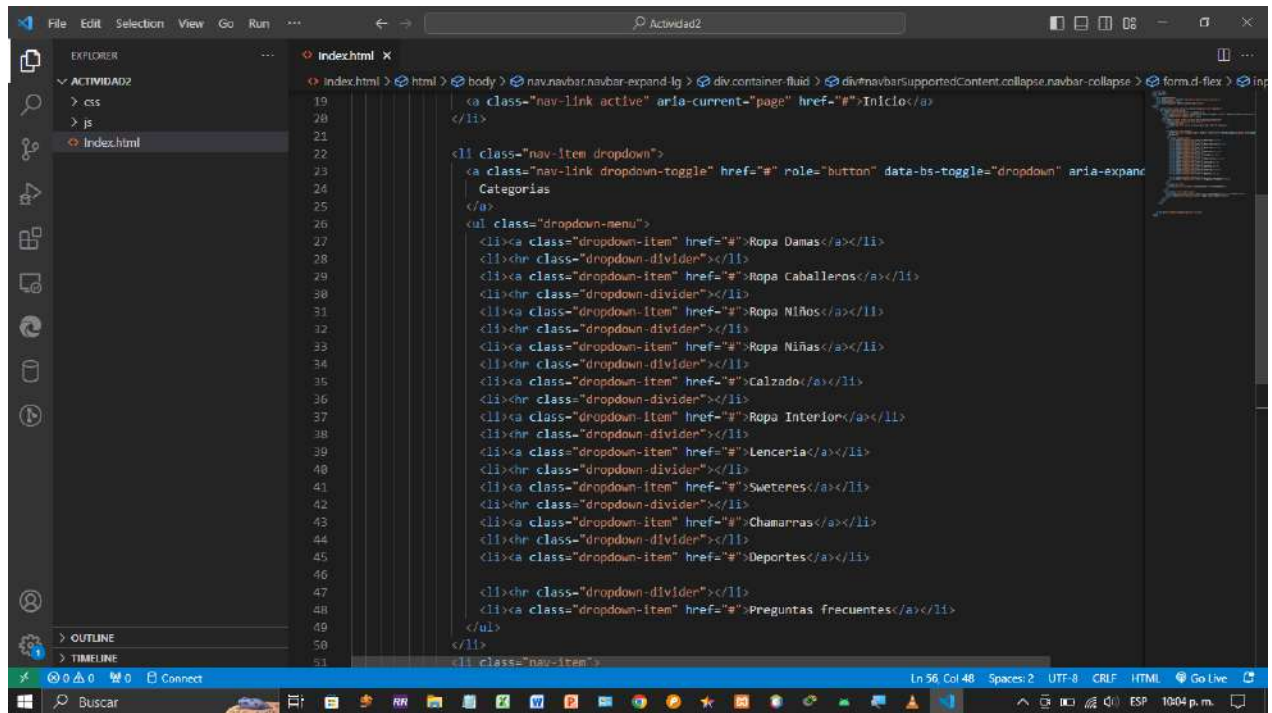




```
index.html <html> <body> <nav.navbar.navbar-expand-lg>
4
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>TiendaSara</title>
8 <link rel="stylesheet" href="css/bootstrap.min.css">
9 </head>
10 <body>
11 <nav class="navbar navbar-expand-lg" style="background-color: skyblue;">
12 <div class="container-fluid">
13 <a class="navbar-brand" href="#">Navbar</a>
14 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportContent"
15 <span class="navbar-toggler-icon"></span>
16 </button>
17 <div class="collapse navbar-collapse" id="navbarSupportContent">
18 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
19 <li class="nav-item">
20 <a class="nav-link active" aria-current="page" href="#">Home</a>
21 </li>
22 <li class="nav-item">
23 <a class="nav-link" href="#">Link</a>
24 </li>
25 <li class="nav-item dropdown">
26 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
27 Dropdown
28 </a>
29 <ul class="dropdown-menu">
30 <li><a class="dropdown-item" href="#">Action</a></li>
31 <li><a class="dropdown-item" href="#">Another action</a></li>
32 <li><hr class="dropdown-divider"></li>
33 <li><a class="dropdown-item" href="#">Something else here</a></li>
34 </ul>
35 </li>
36 <li class="nav-item">
37 <a class="nav-link disabled" aria-disabled="true">Disabled</a>
38 </li>
39 </ul>
40 </div>
41 </div>
42 </nav>
43 </body>
44 </html>
```

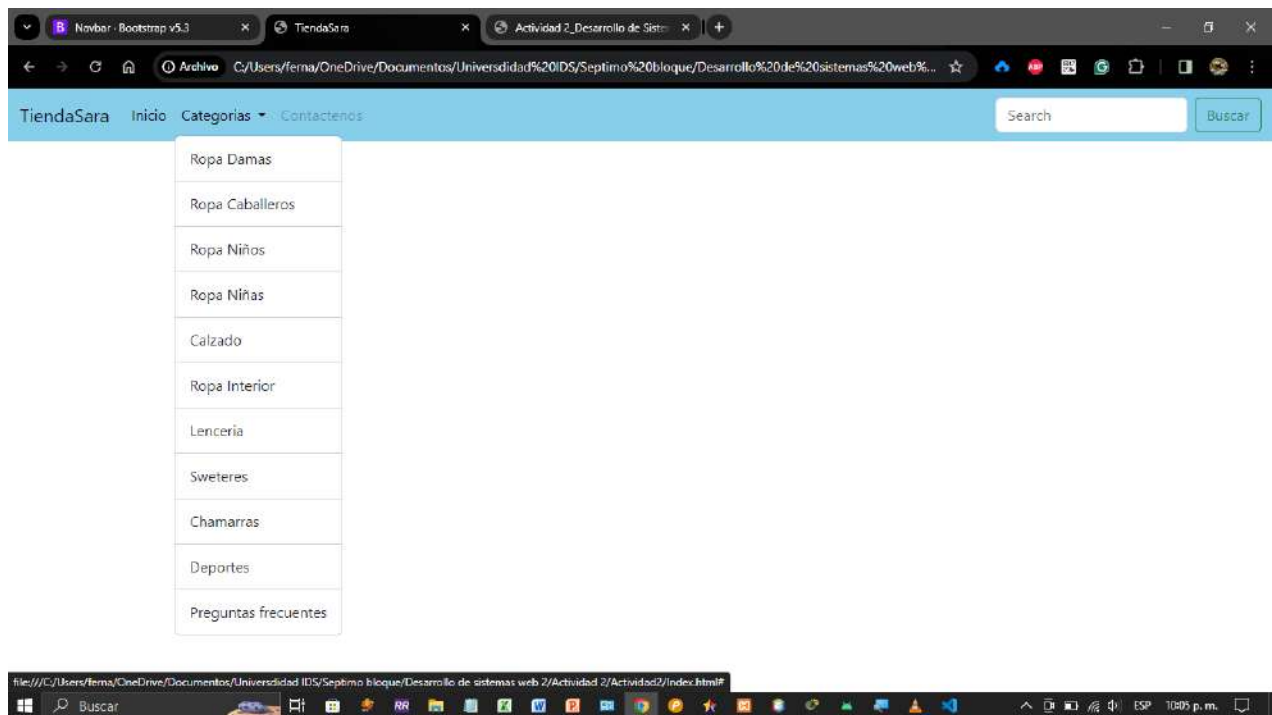
O se selecciona un color específico con style="background-color: skyblue;

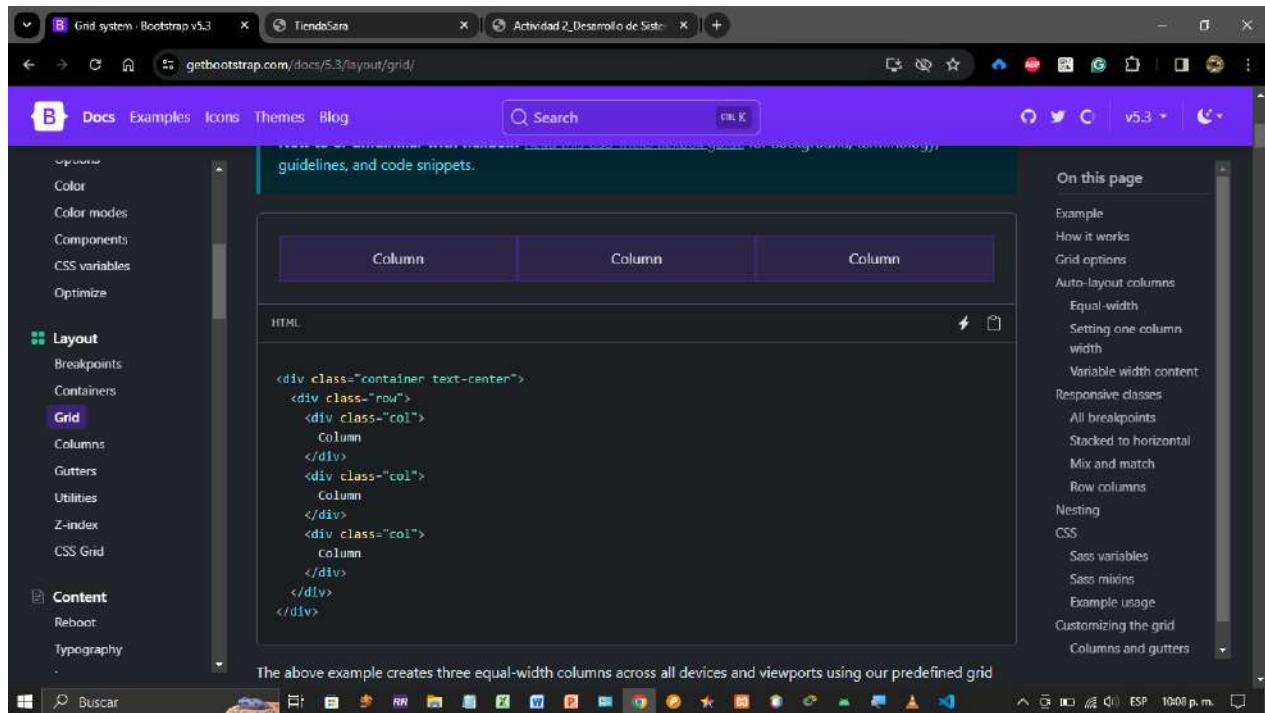




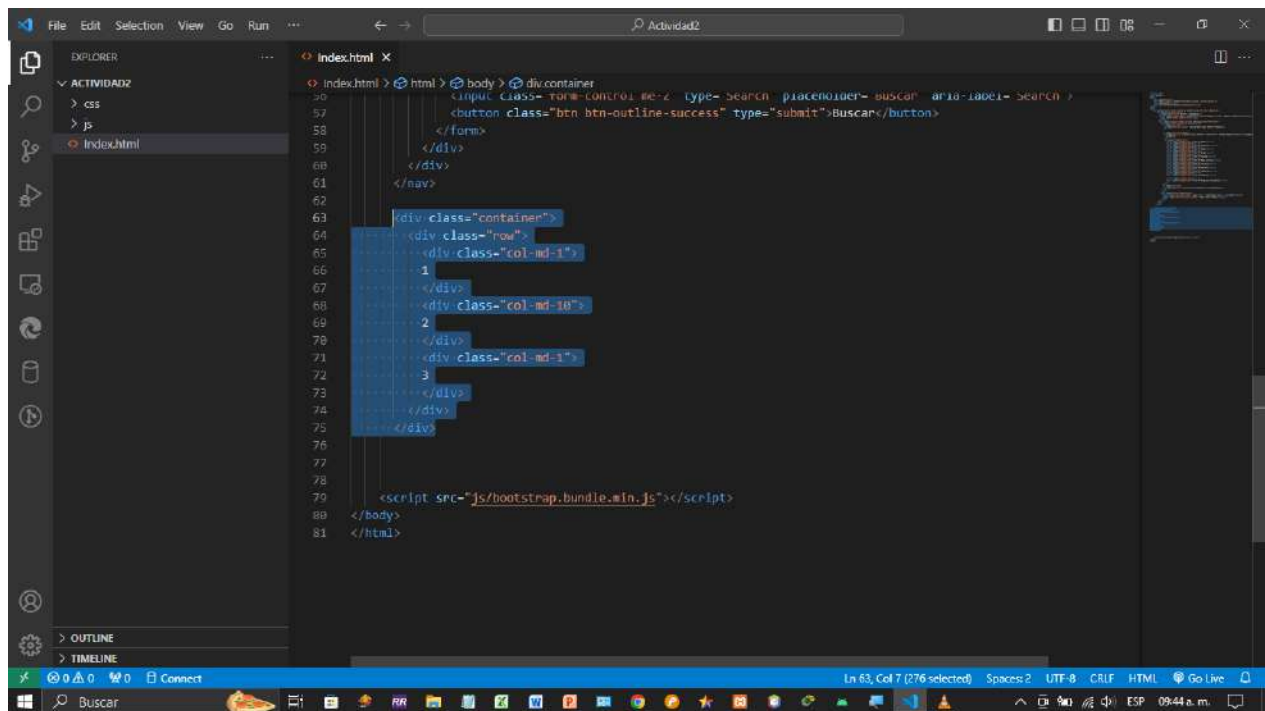
```
19 <a class="nav-link active" aria-current="page" href="#">Inicio</a>
20 </li>
21
22 <li class="nav-item dropdown">
23   <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
24     Categorías
25   </a>
26   <ul class="dropdown-menu">
27     <li><a class="dropdown-item" href="#">Ropa Damas</a></li>
28     <li><hr class="dropdown-divider" /></li>
29     <li><a class="dropdown-item" href="#">Ropa Caballeros</a></li>
30     <li><hr class="dropdown-divider" /></li>
31     <li><a class="dropdown-item" href="#">Ropa Niños</a></li>
32     <li><hr class="dropdown-divider" /></li>
33     <li><a class="dropdown-item" href="#">Ropa Niñas</a></li>
34     <li><hr class="dropdown-divider" /></li>
35     <li><a class="dropdown-item" href="#">Calzado</a></li>
36     <li><hr class="dropdown-divider" /></li>
37     <li><a class="dropdown-item" href="#">Ropa Interior</a></li>
38     <li><hr class="dropdown-divider" /></li>
39     <li><a class="dropdown-item" href="#">Lencería</a></li>
40     <li><hr class="dropdown-divider" /></li>
41     <li><a class="dropdown-item" href="#">Sweteres</a></li>
42     <li><hr class="dropdown-divider" /></li>
43     <li><a class="dropdown-item" href="#">Chamarras</a></li>
44     <li><hr class="dropdown-divider" /></li>
45     <li><a class="dropdown-item" href="#">Deportes</a></li>
46
47     <li><hr class="dropdown-divider" /></li>
48     <li><a class="dropdown-item" href="#">Preguntas frecuentes</a></li>
49   </ul>
50 </li>
51 <li class="nav-item">
```

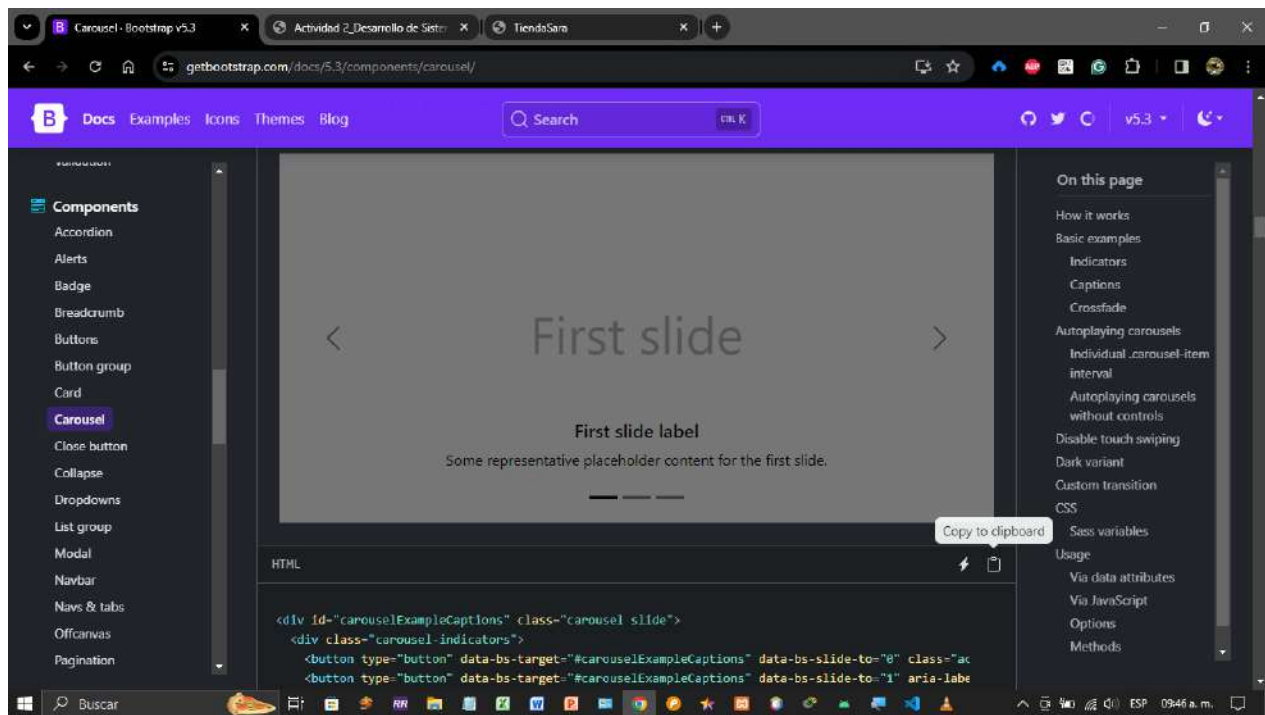
Se crean los atributos que tendrá la barra de navegación



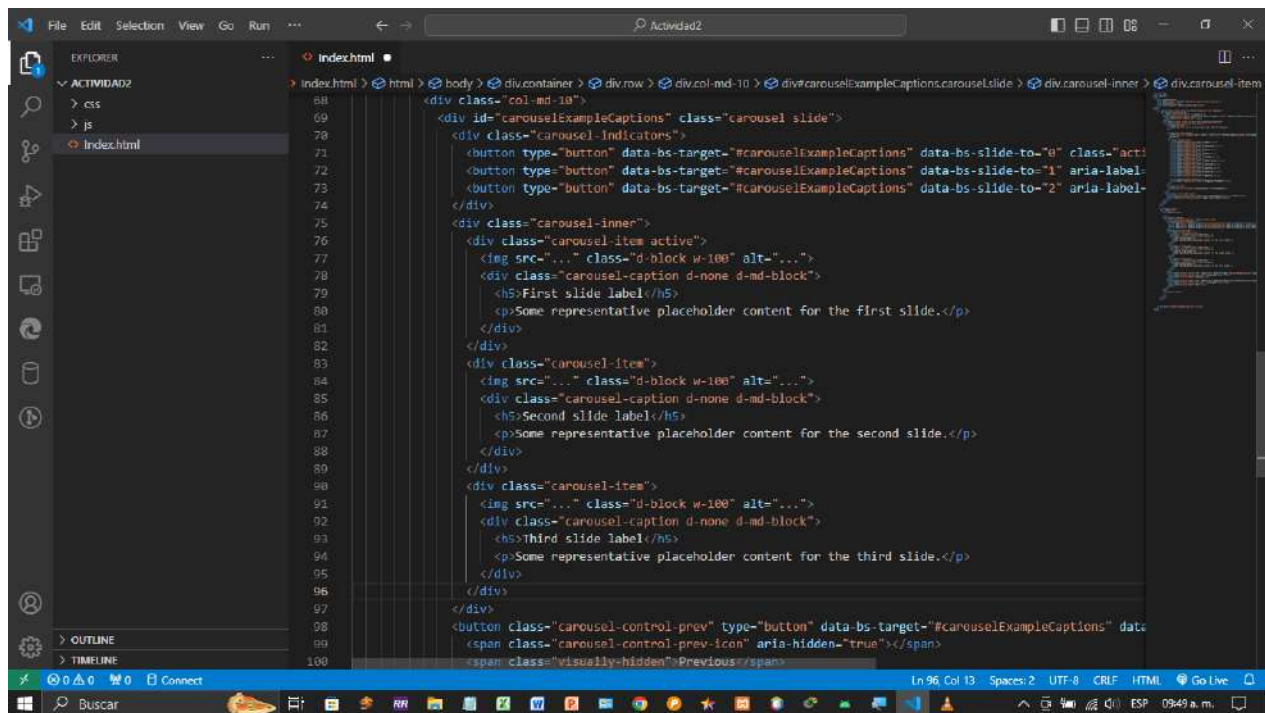


Se selecciona el código de la cuadrícula a utilizar y se distribuye de acuerdo al diseño a realizar.

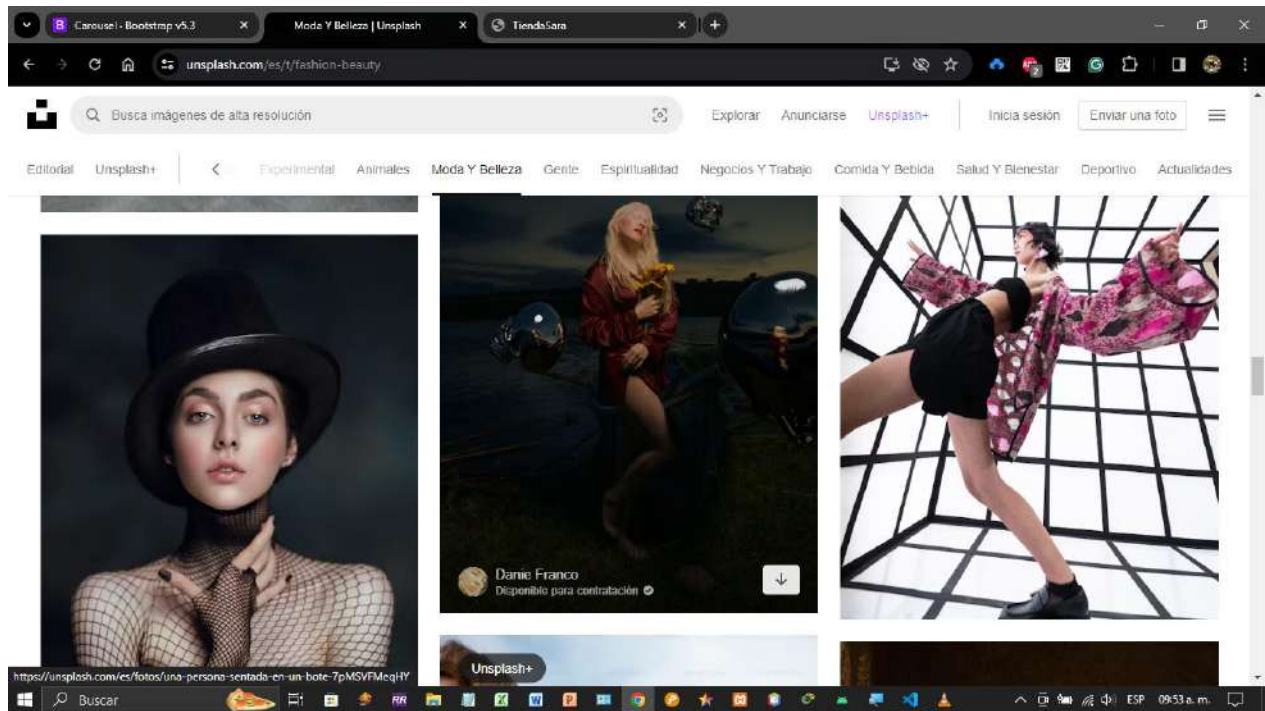




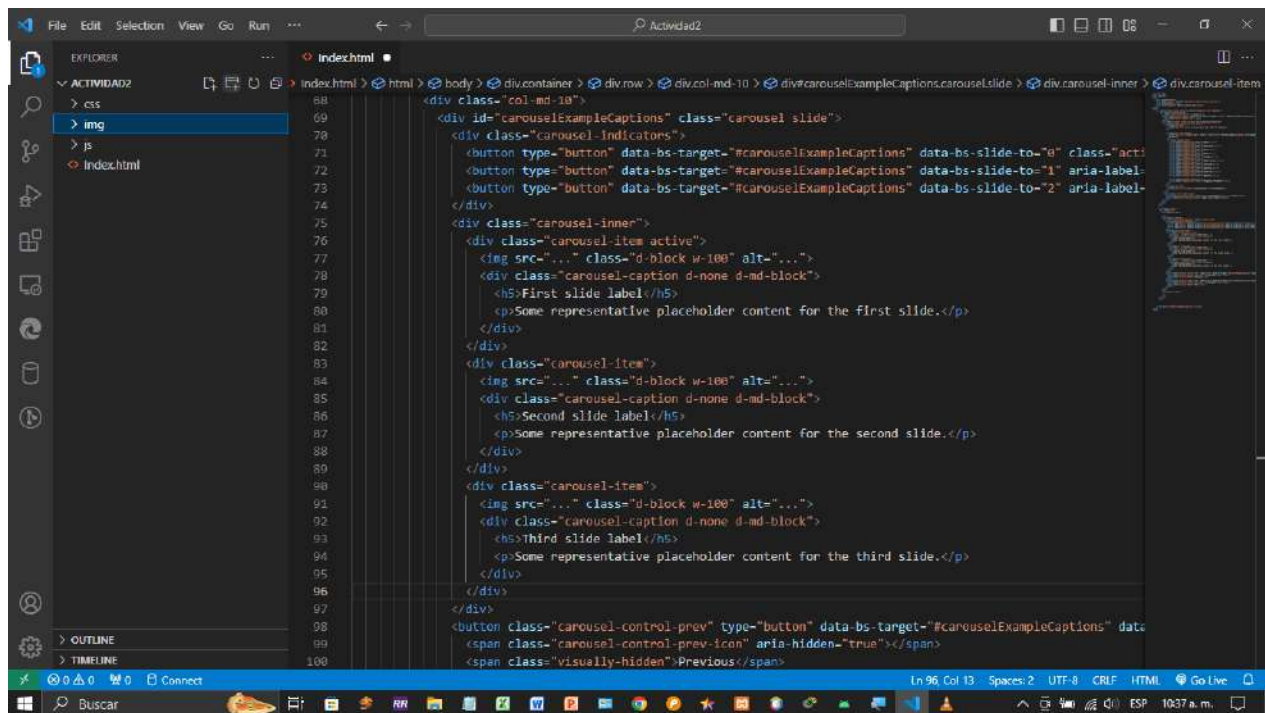
Se selecciona el código de carrusel para insertarlo en la cuadrícula



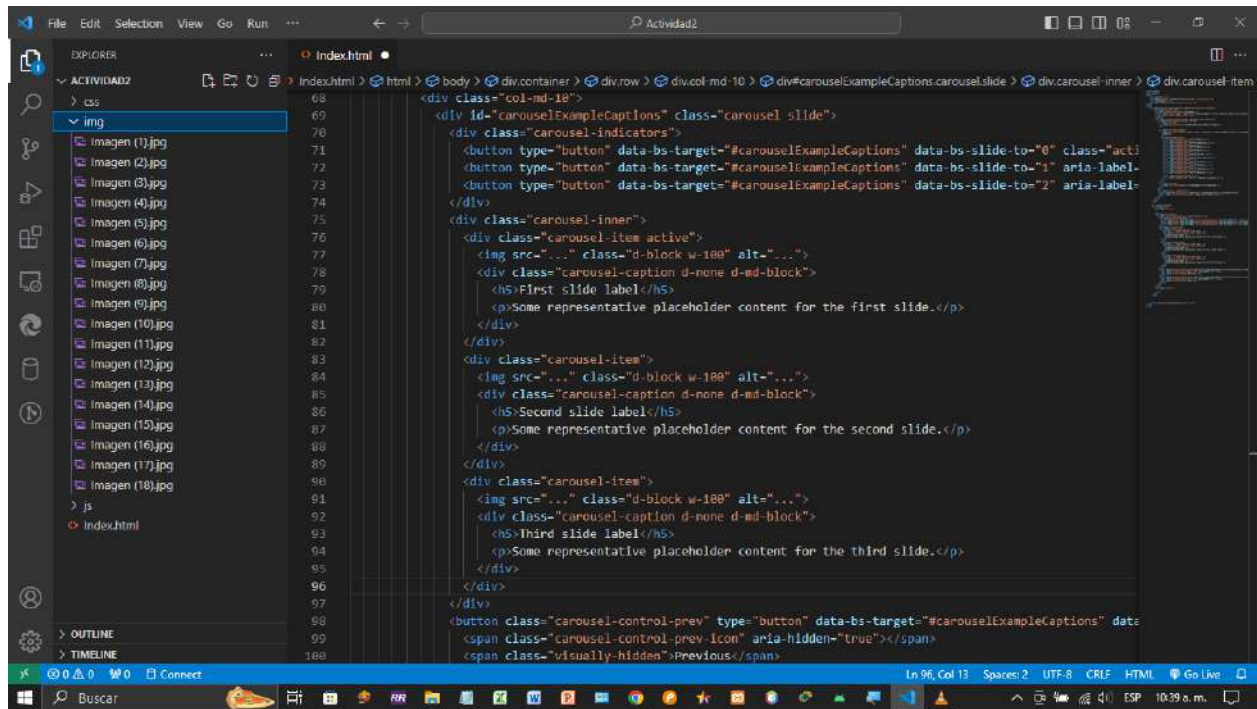
Se pega en el segundo apartado de la cuadrícula



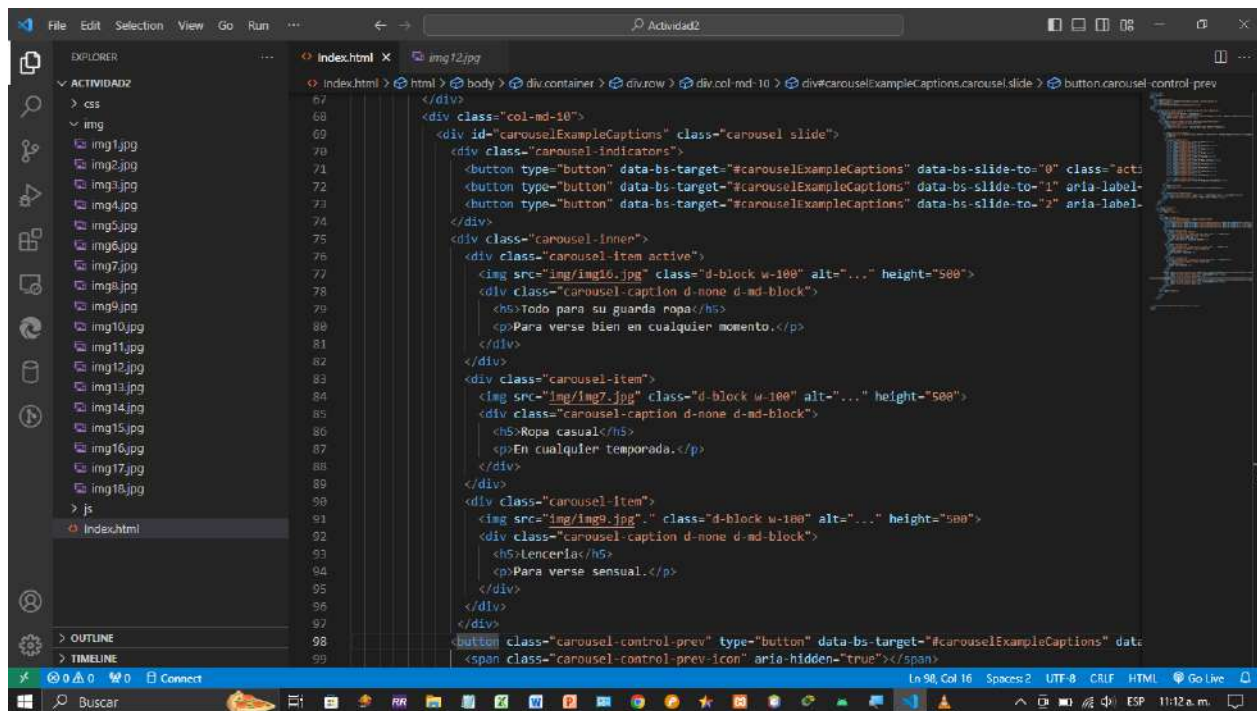
Se seleccionan las imágenes a utilizar



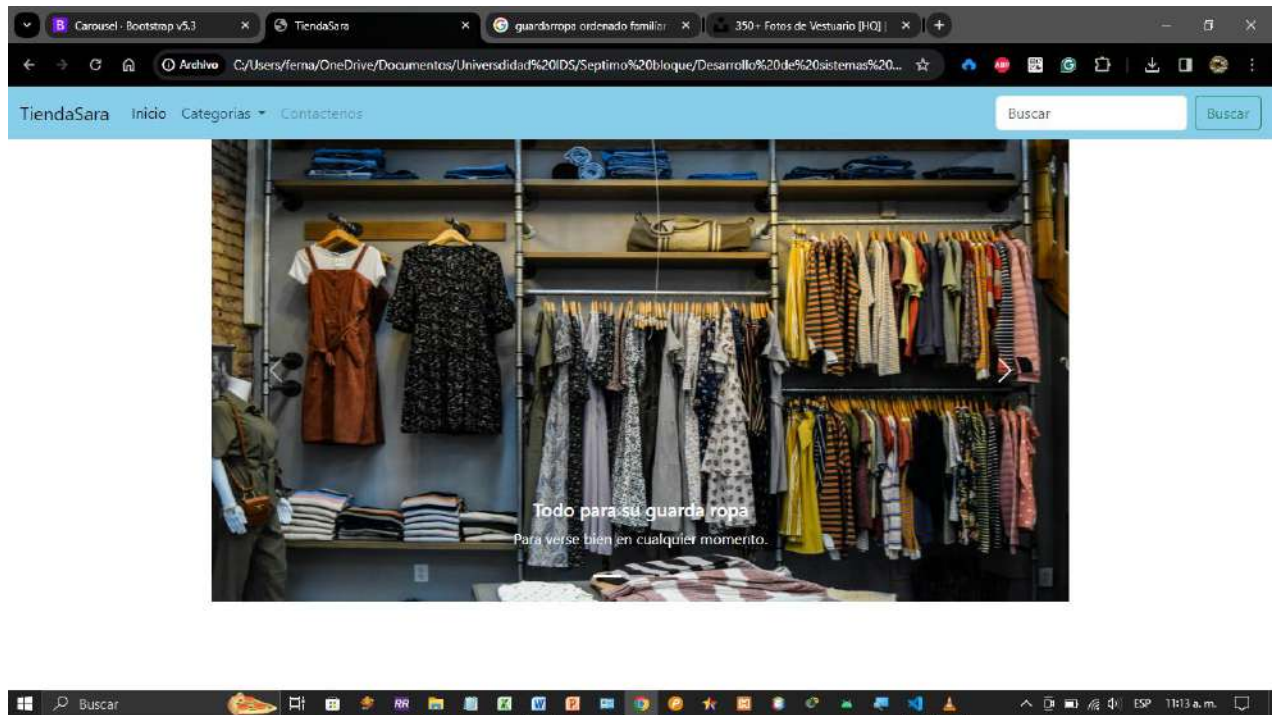
Se crea la carpeta img



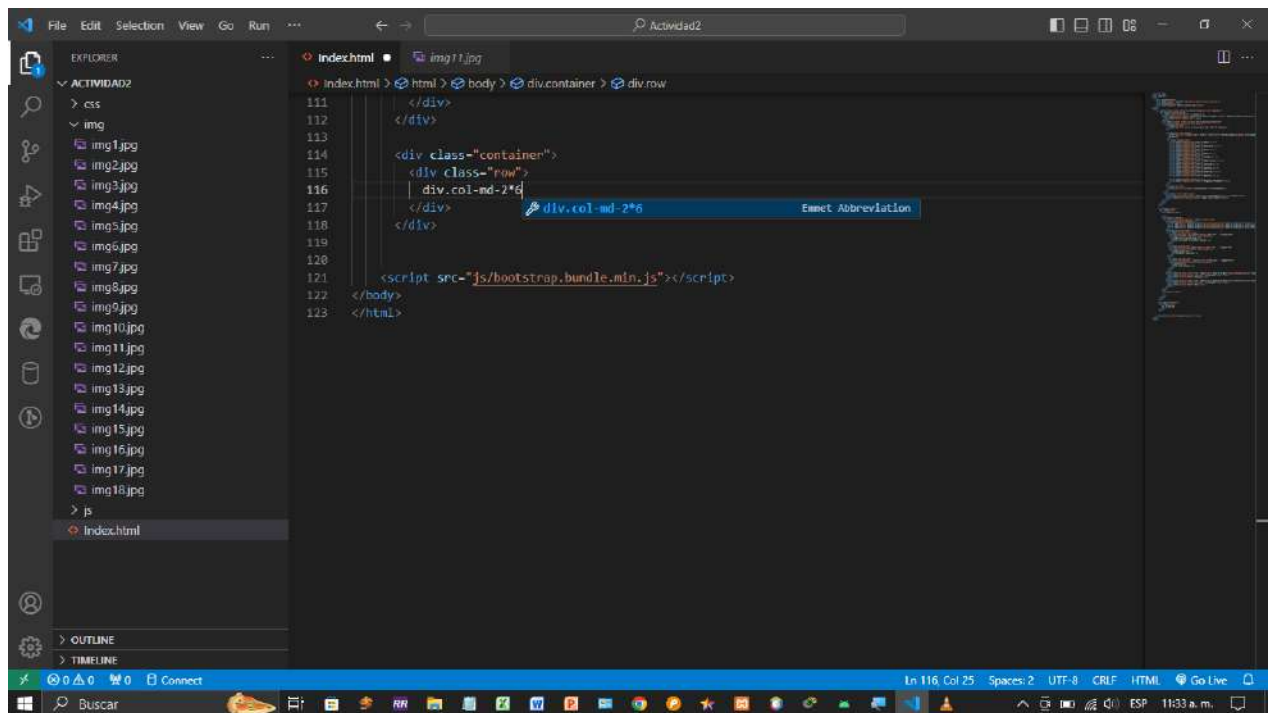
Y se guardan las imágenes a utilizar



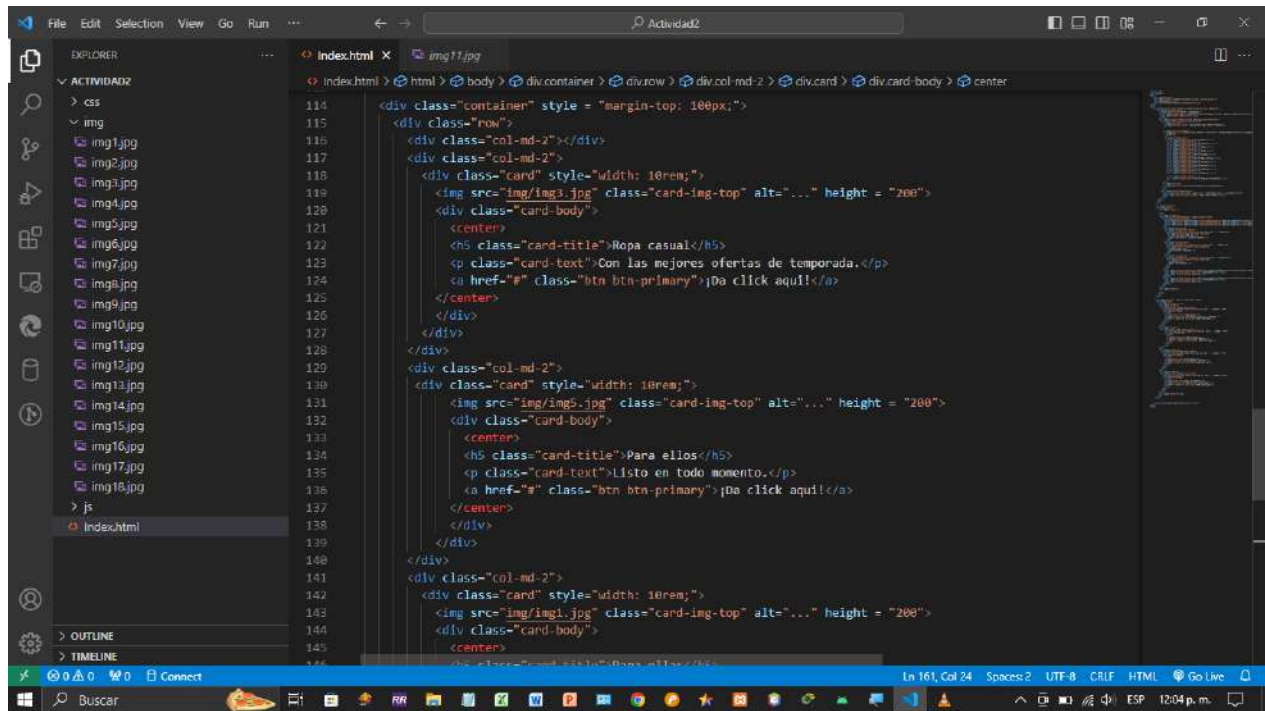
Se edita el contexto e imágenes a utilizar en el diseño



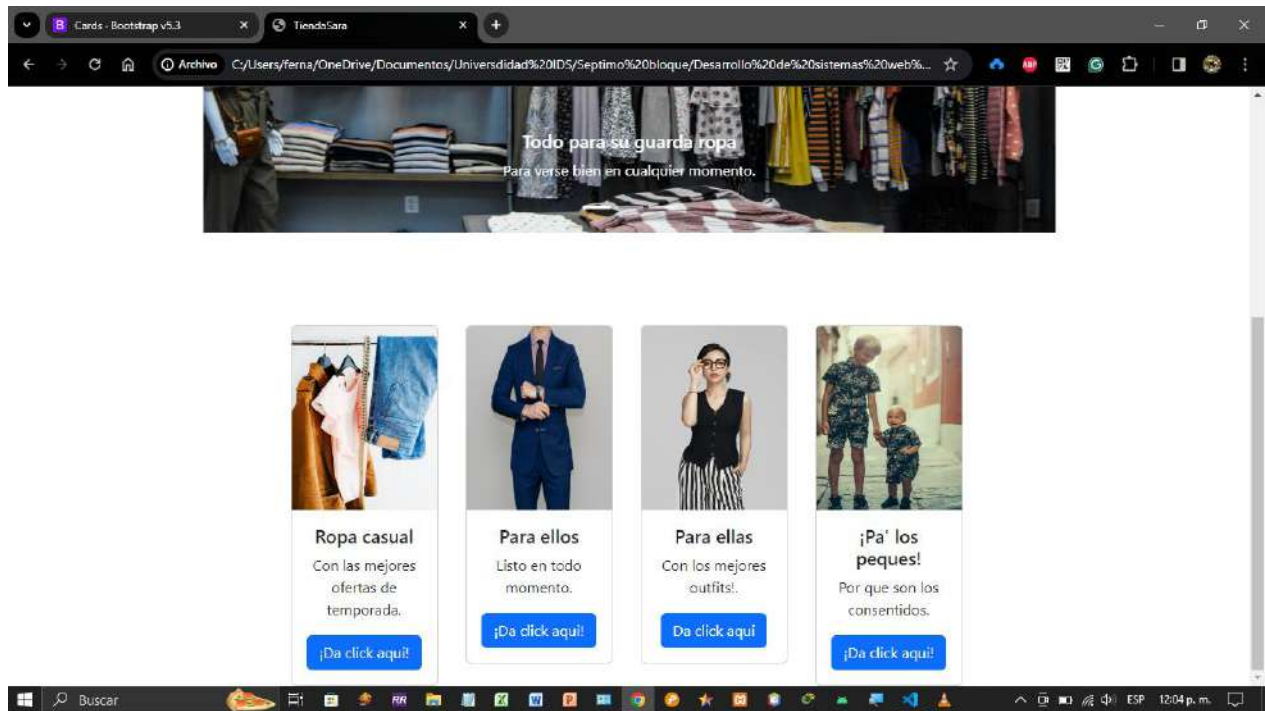
Para mostrarse en la página principal



Se crean los apartados que se tendrá para las cartas

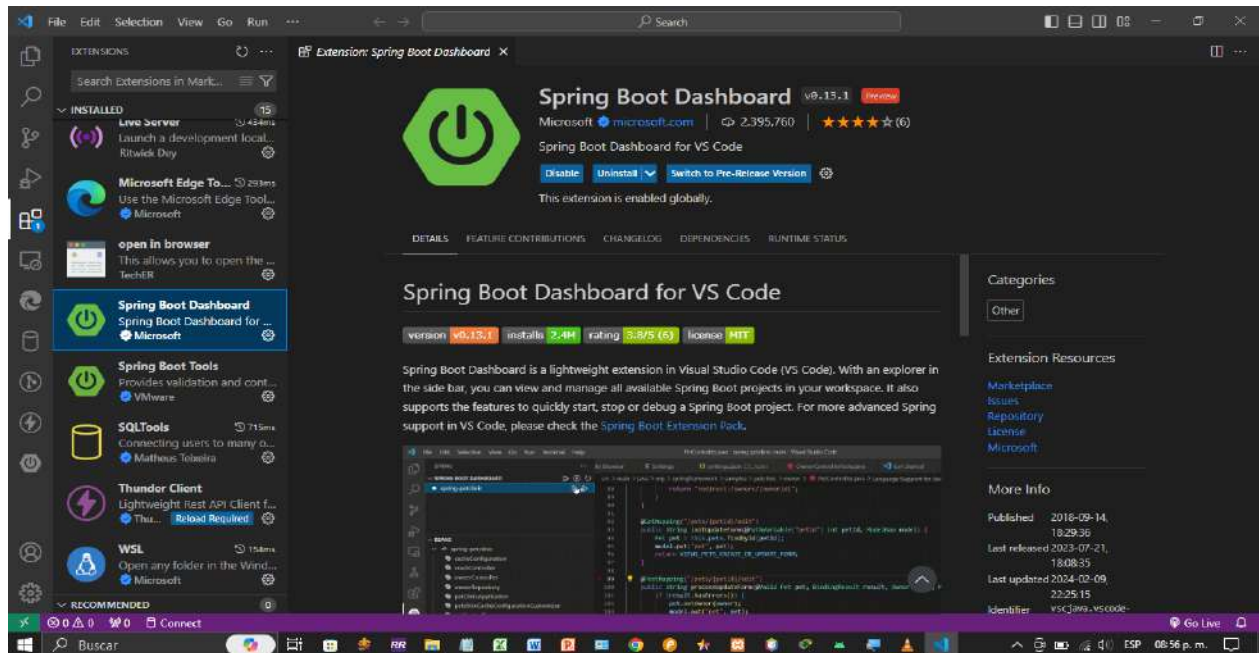


Se edita de acuerdo al diseño

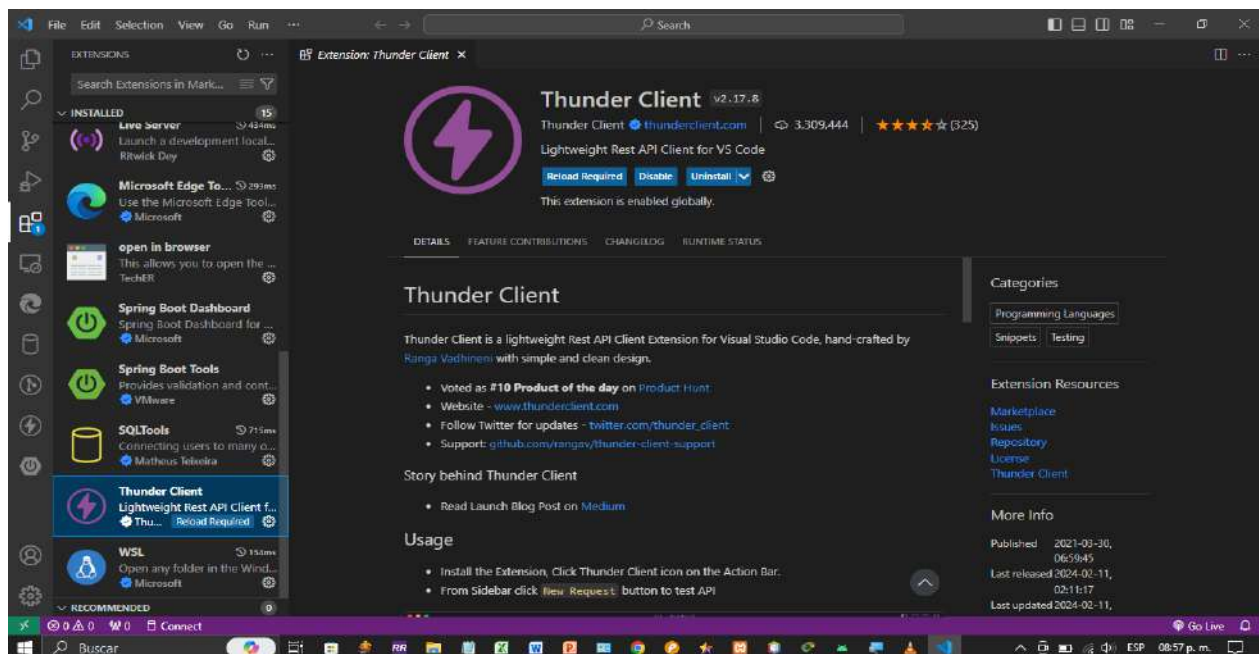


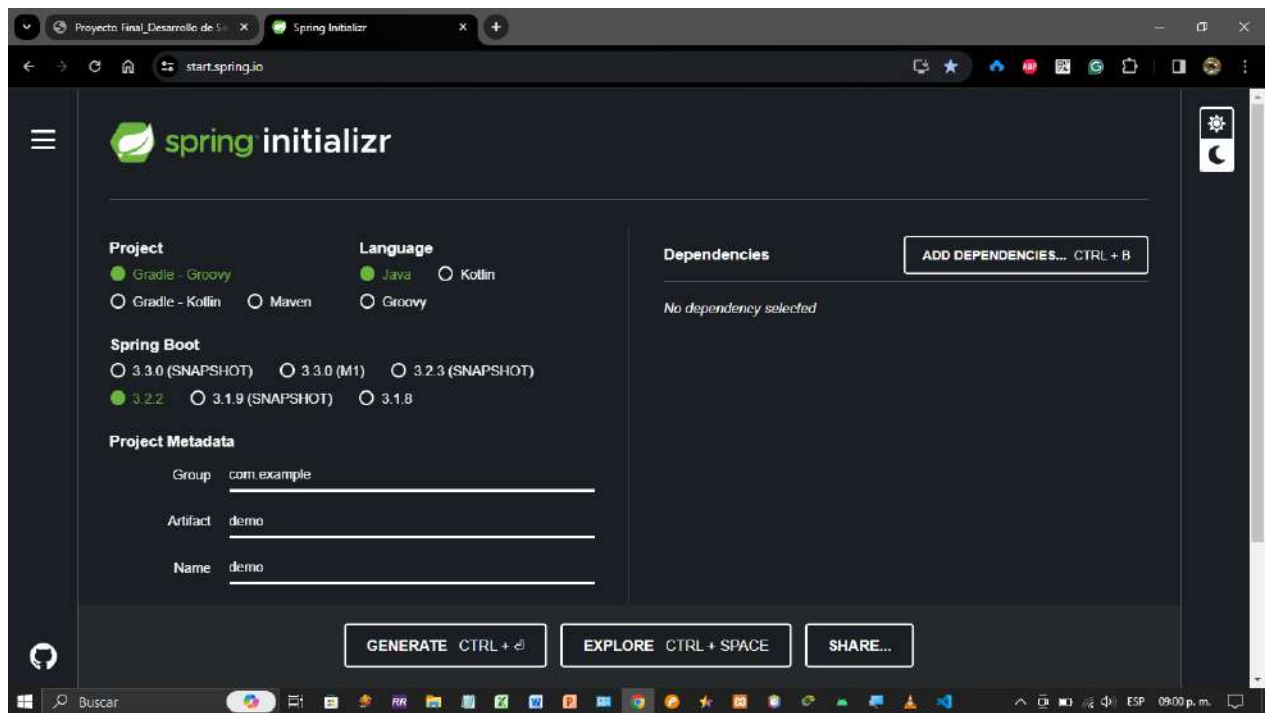
Desarrollo.

Eta3a 3 – Funcionamiento del carrito

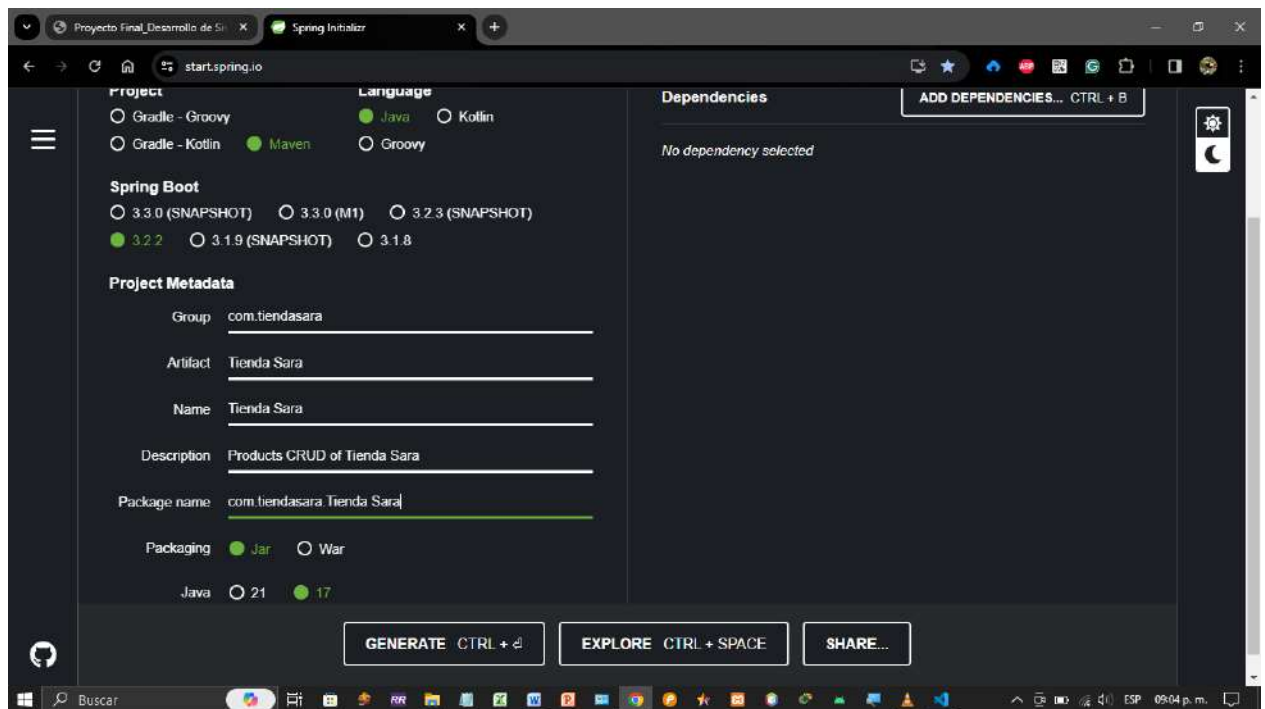


Se agregan las extensiones Spring Boot Dashboard y Thunder Client en Visual Studio Code.

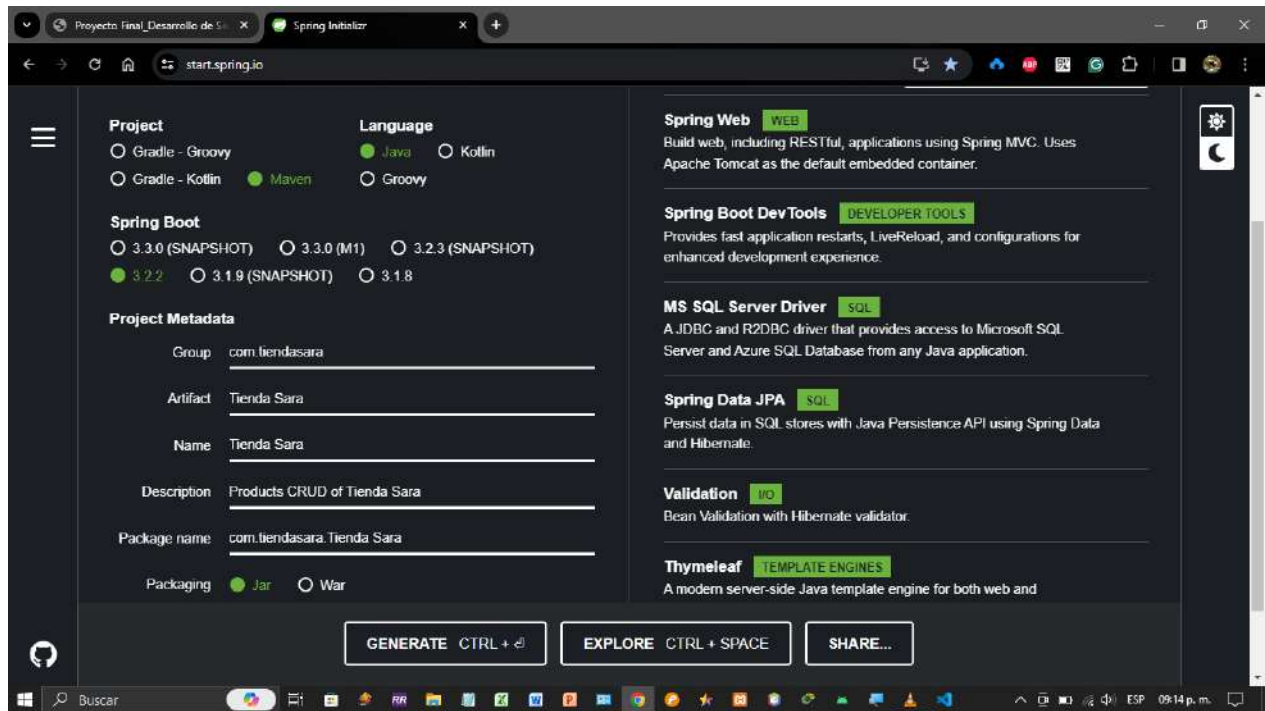




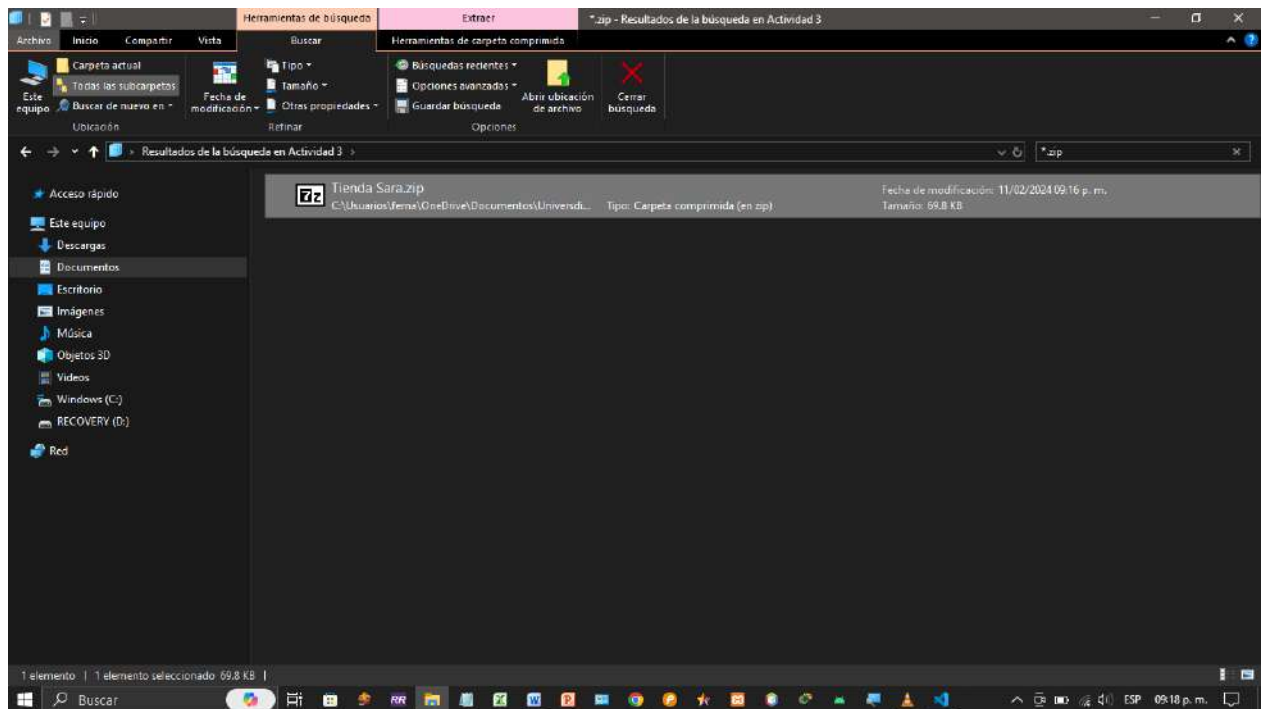
Se ingresa a la página start.spring.io



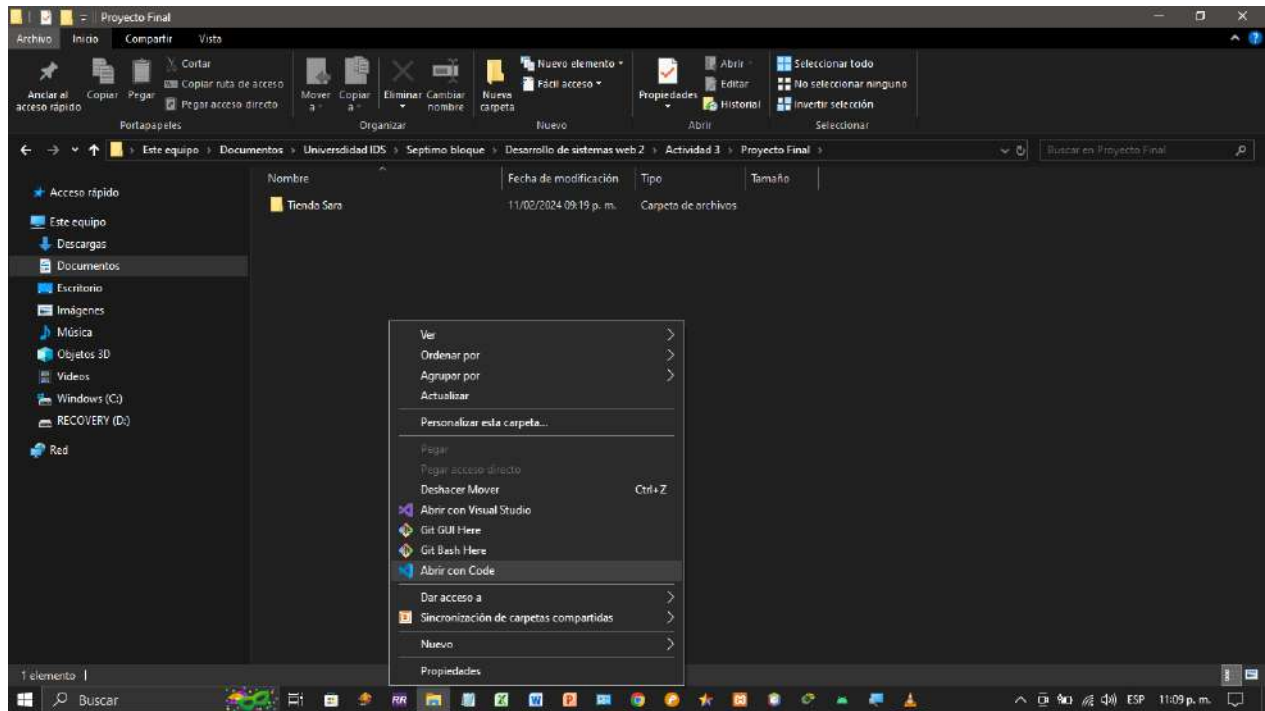
Se selecciona las opciones y el perfil de proyecto



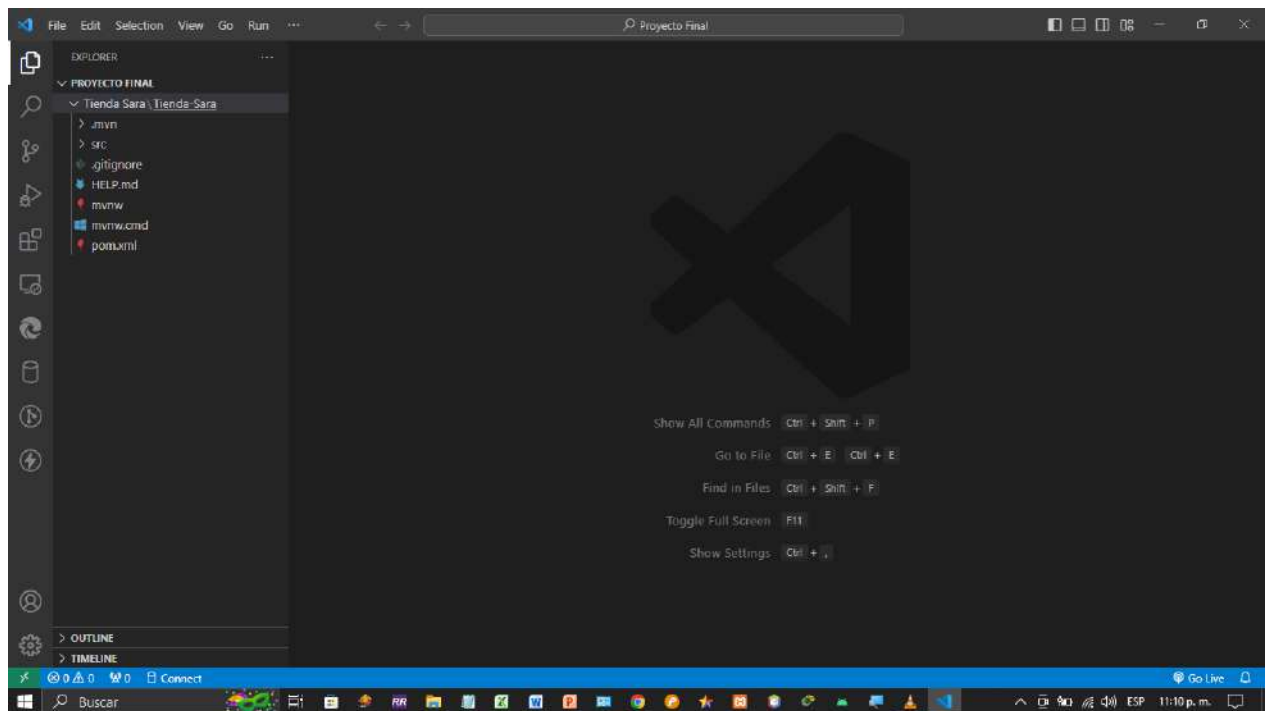
Se agregan las dependencias a utilizar y se genera el proyecto.

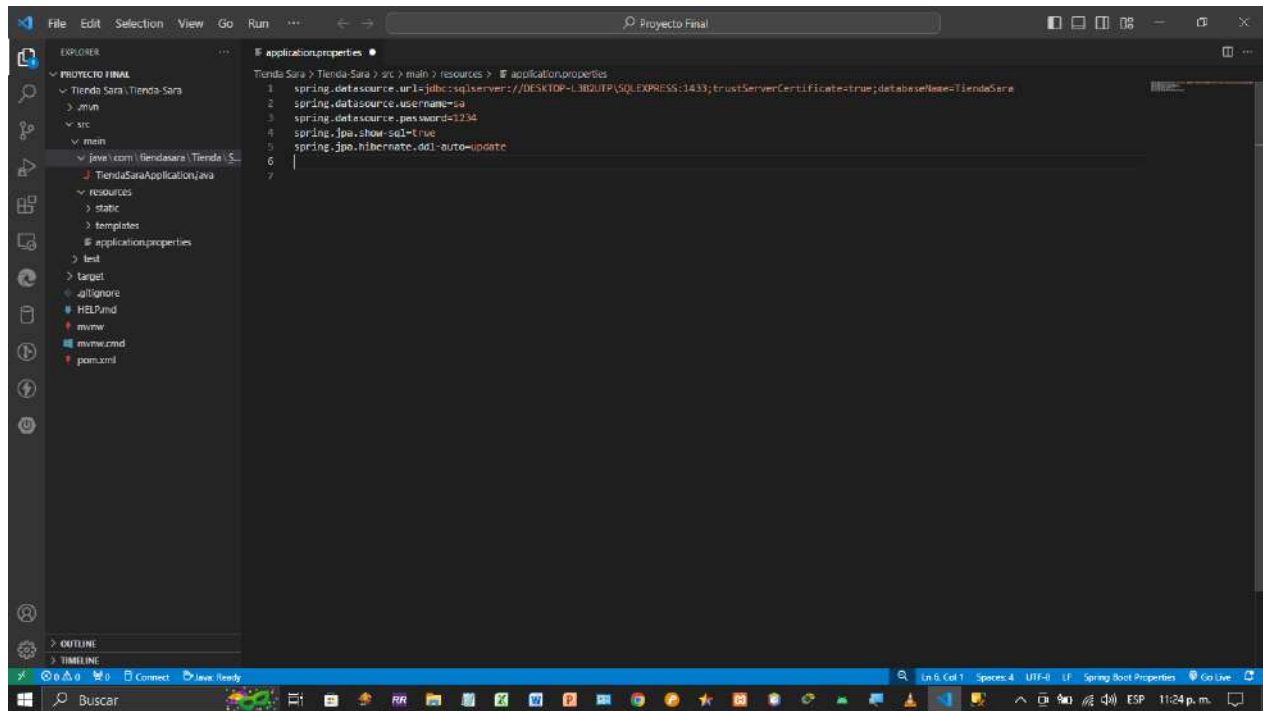


Se descarga y se descomprime para abrirlo con Visual Studio Code

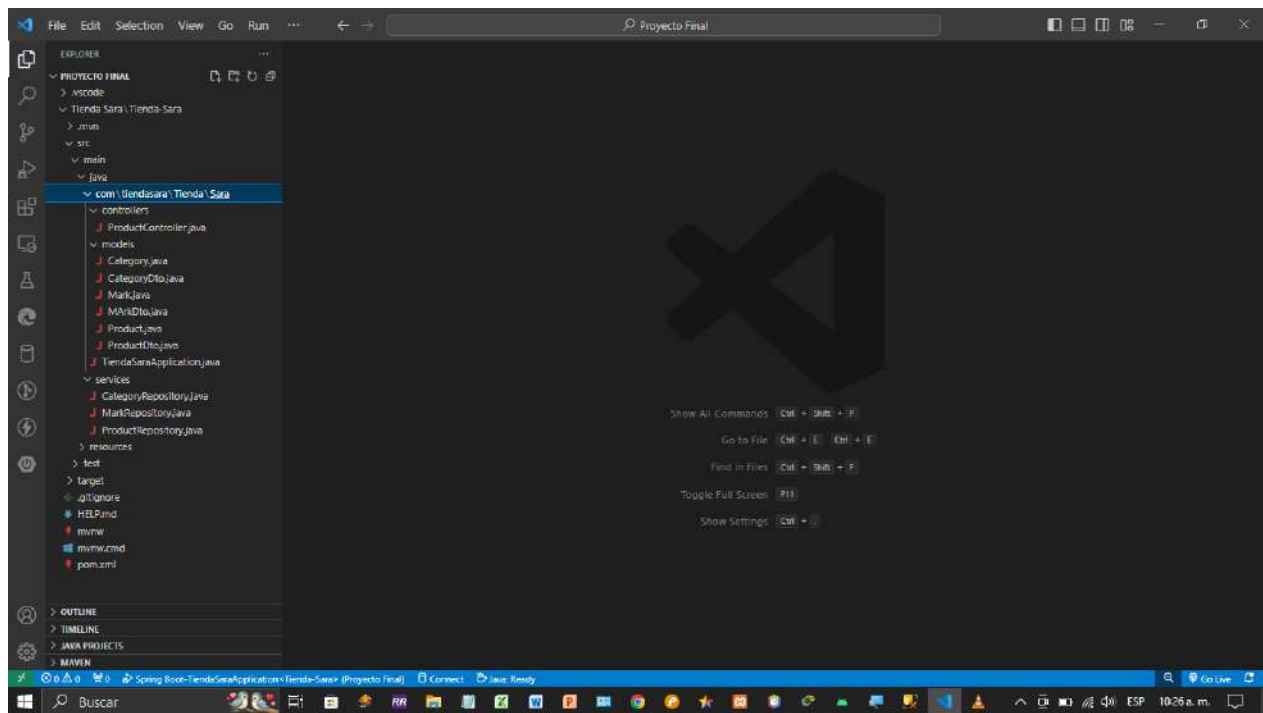


Se abre con Visual Studio Code

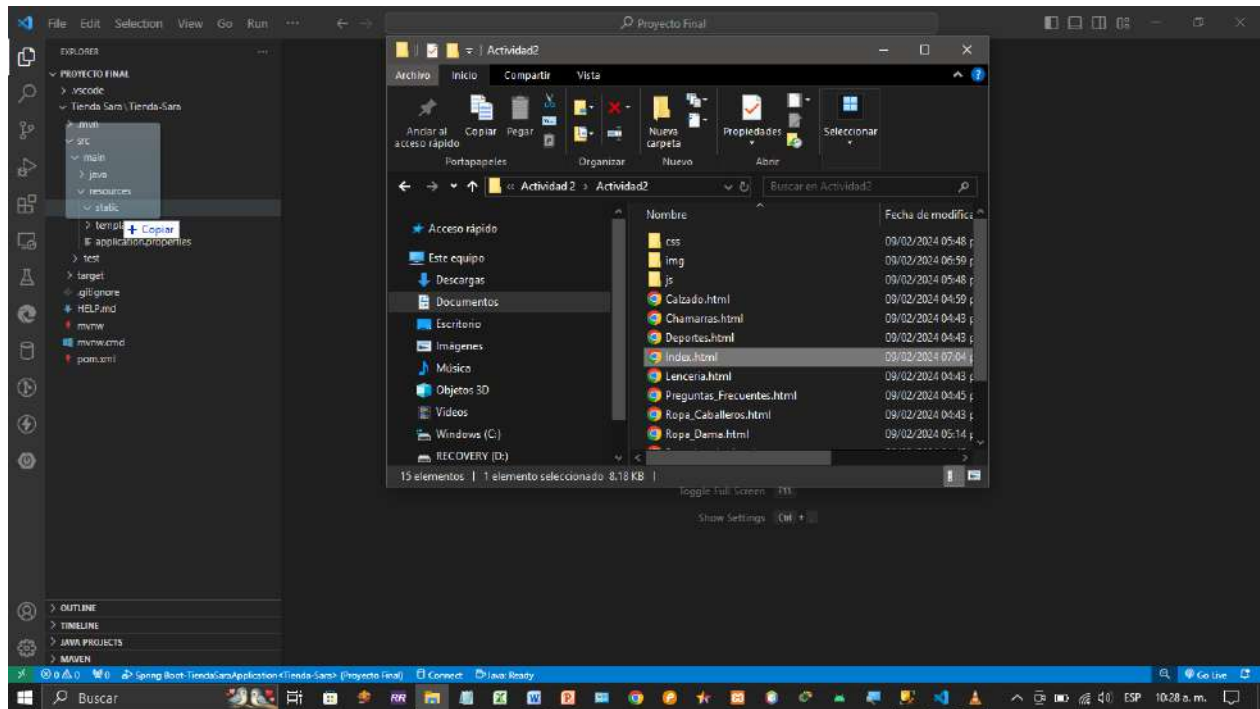




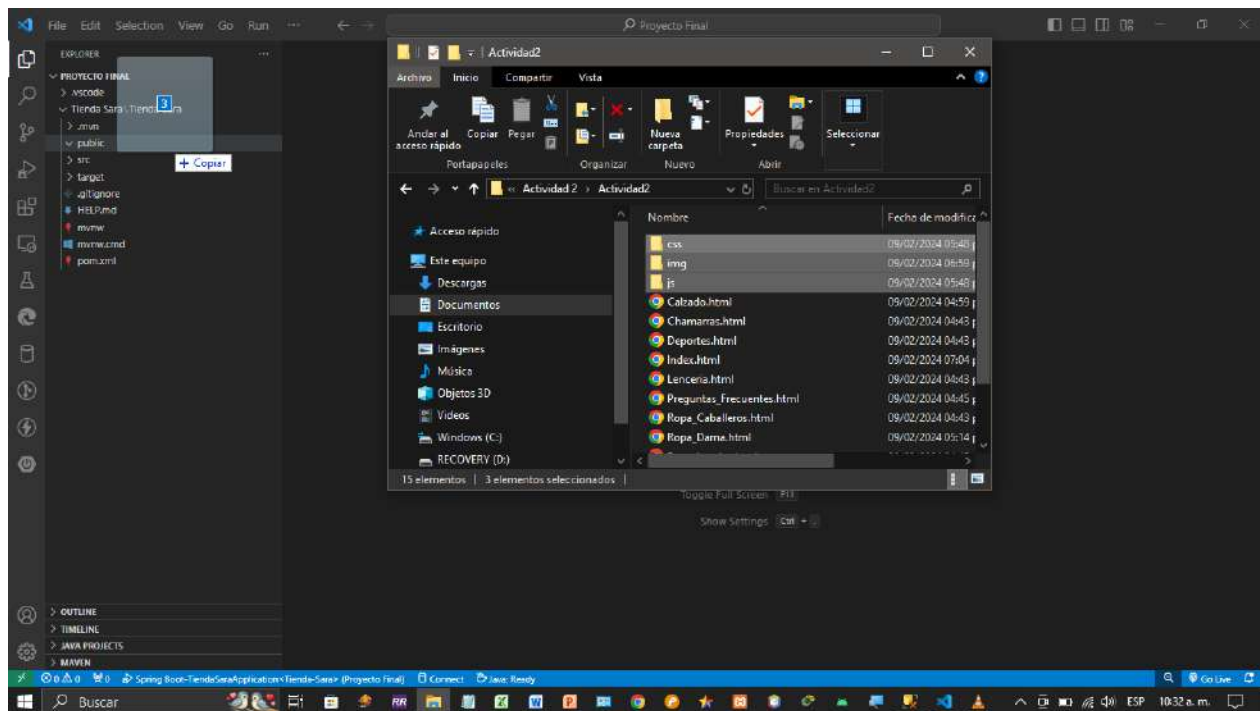
En application.properties se asignan las directivas a utilizar



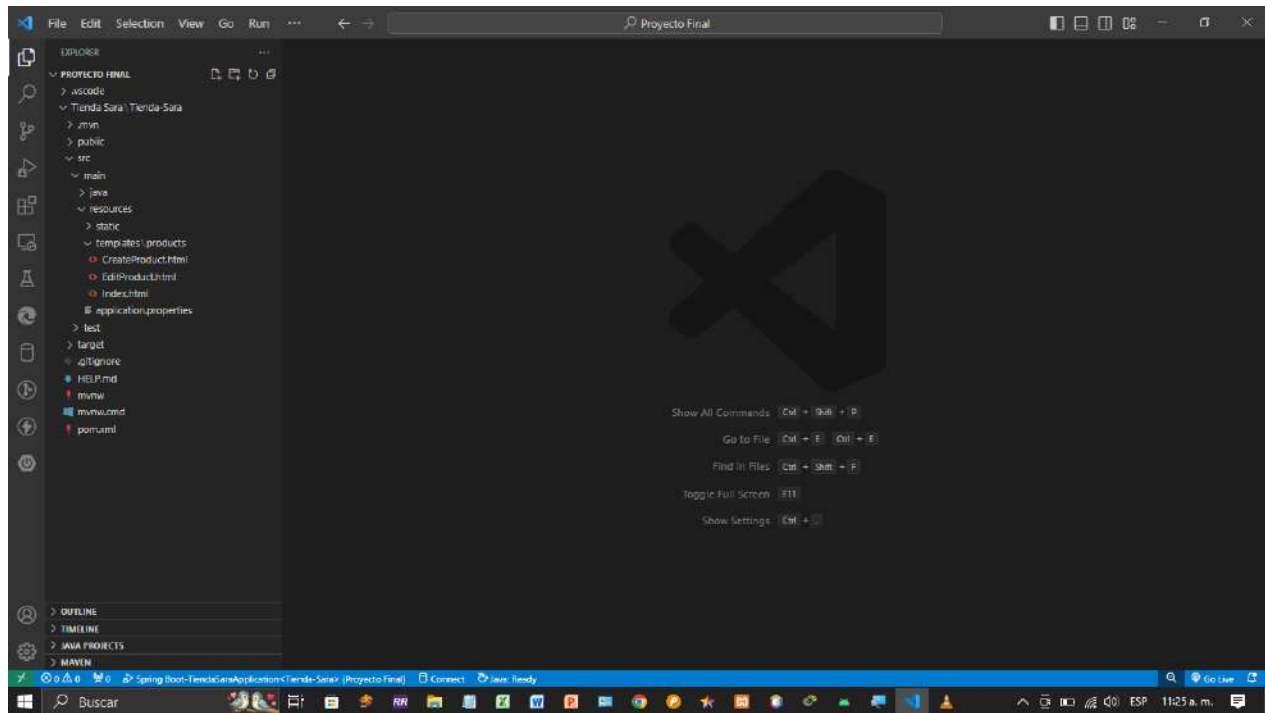
Se crean las carpetas, archivos e interfaces en la ruta src\main\java\com\Tienda\Sara\



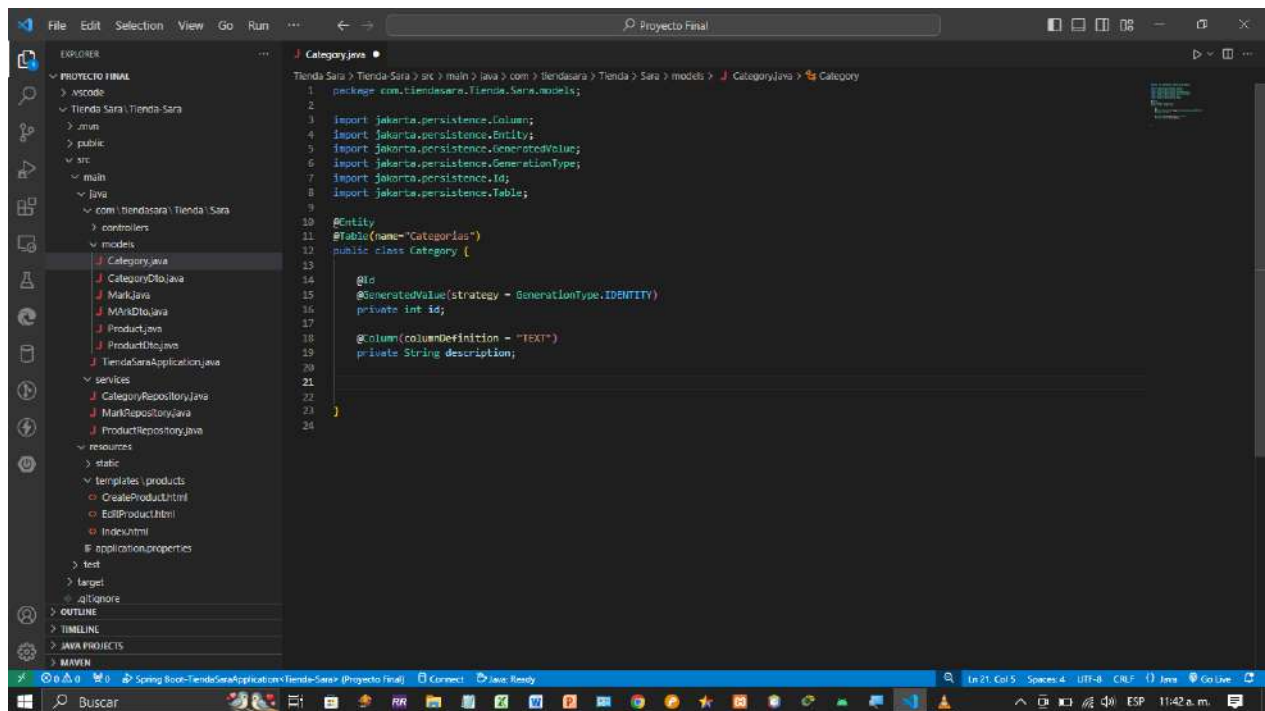
Se mueve el archivo Index.html creado en SQL en el apartado resources\static



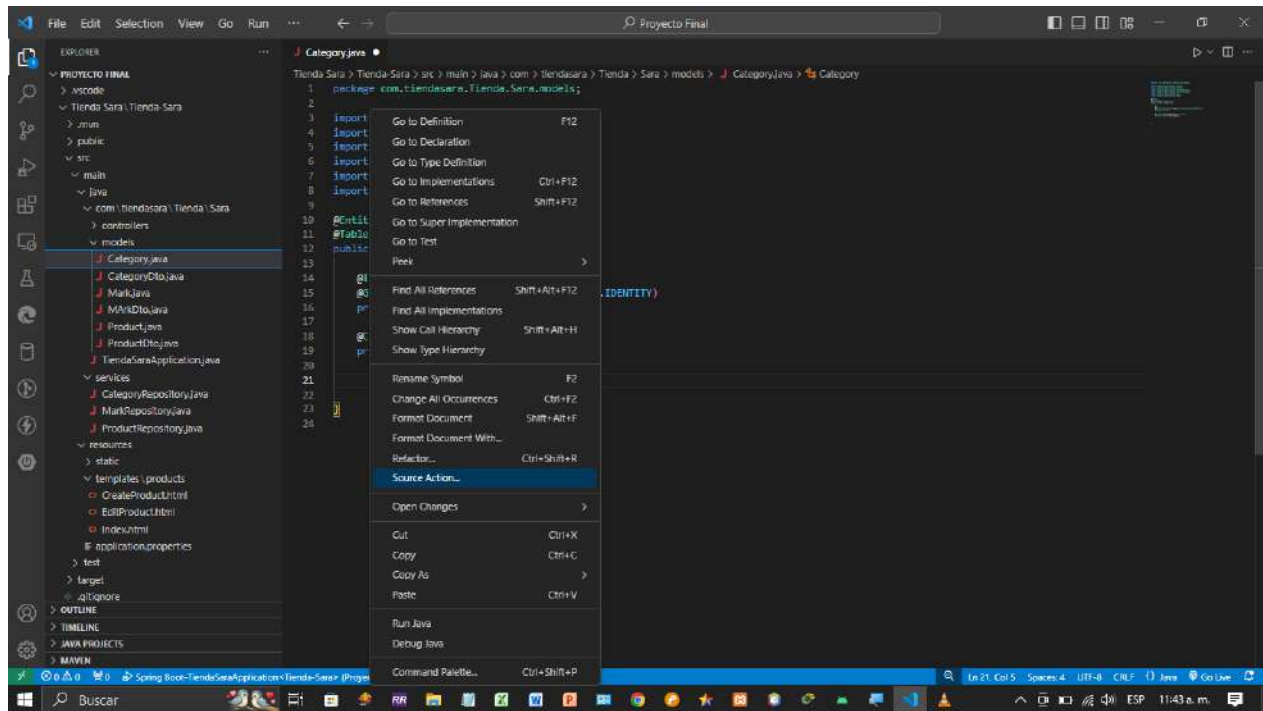
Se crea la carpeta public y se mueven ahí los archivos css, img y js



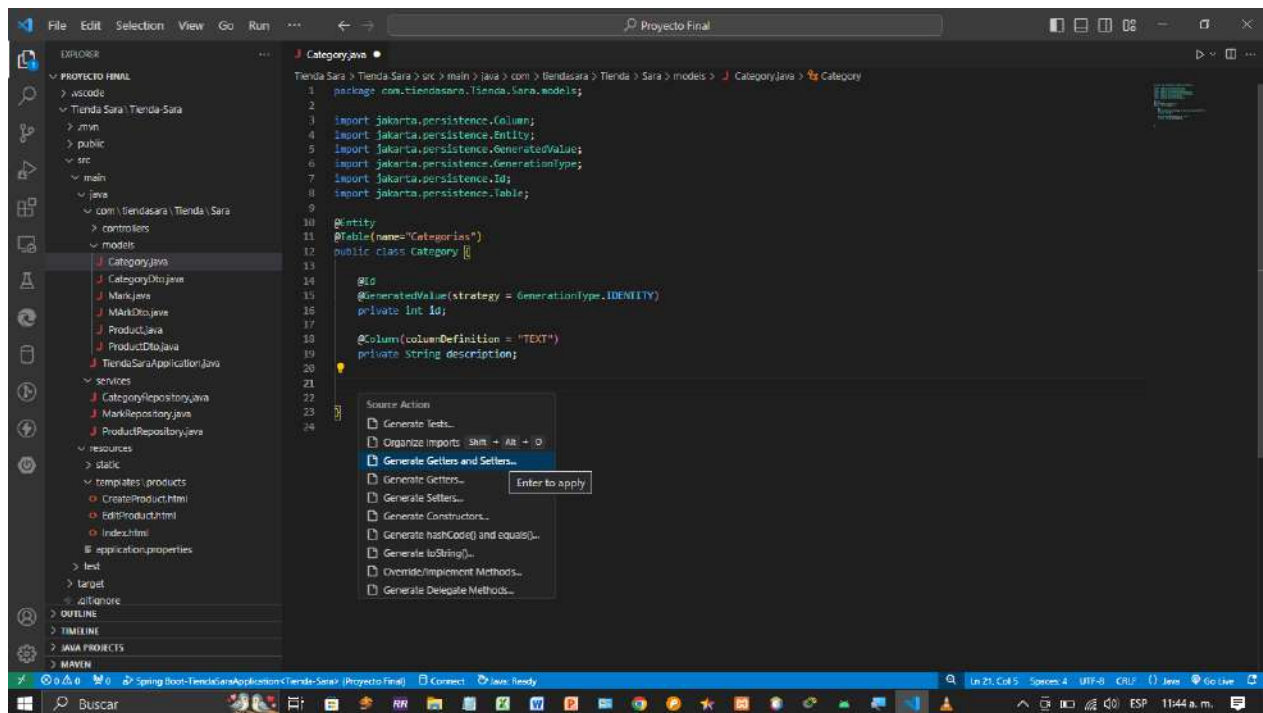
Se crean los templates para products y si se tiene que hacer para otras categorías se hace lo mismo



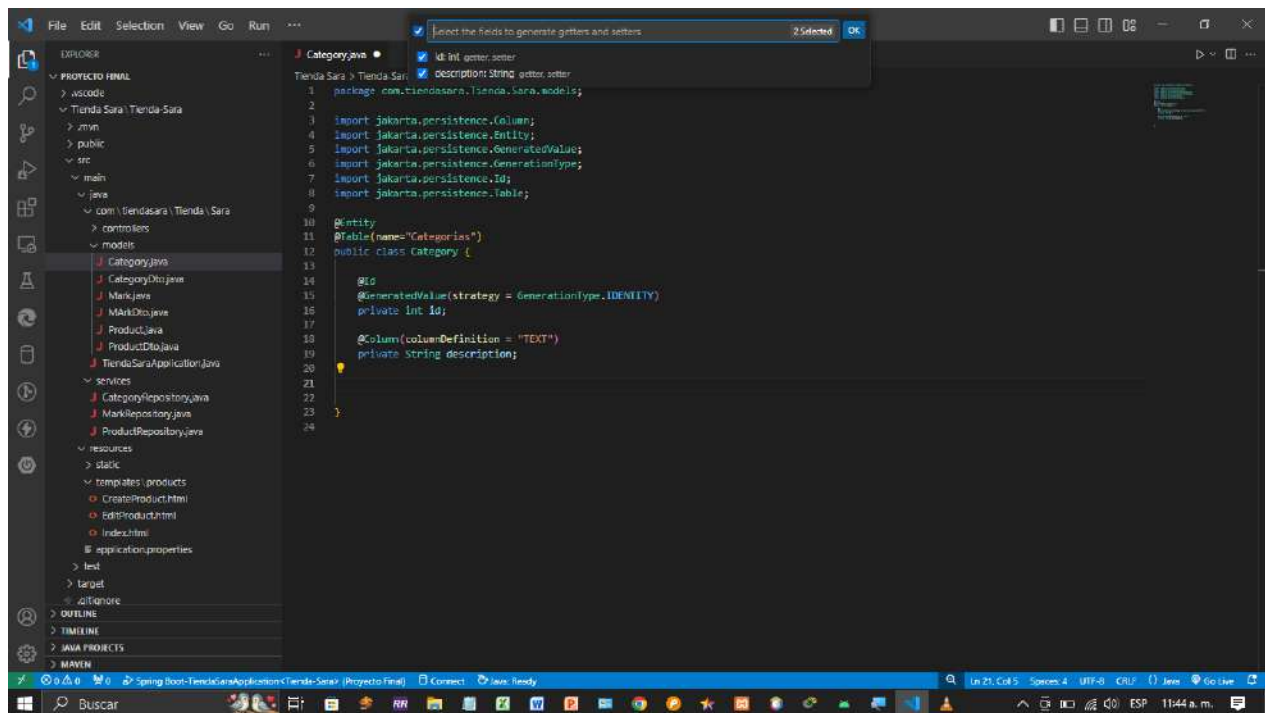
Se indica de donde se va a importar la información y se crea la variable @id



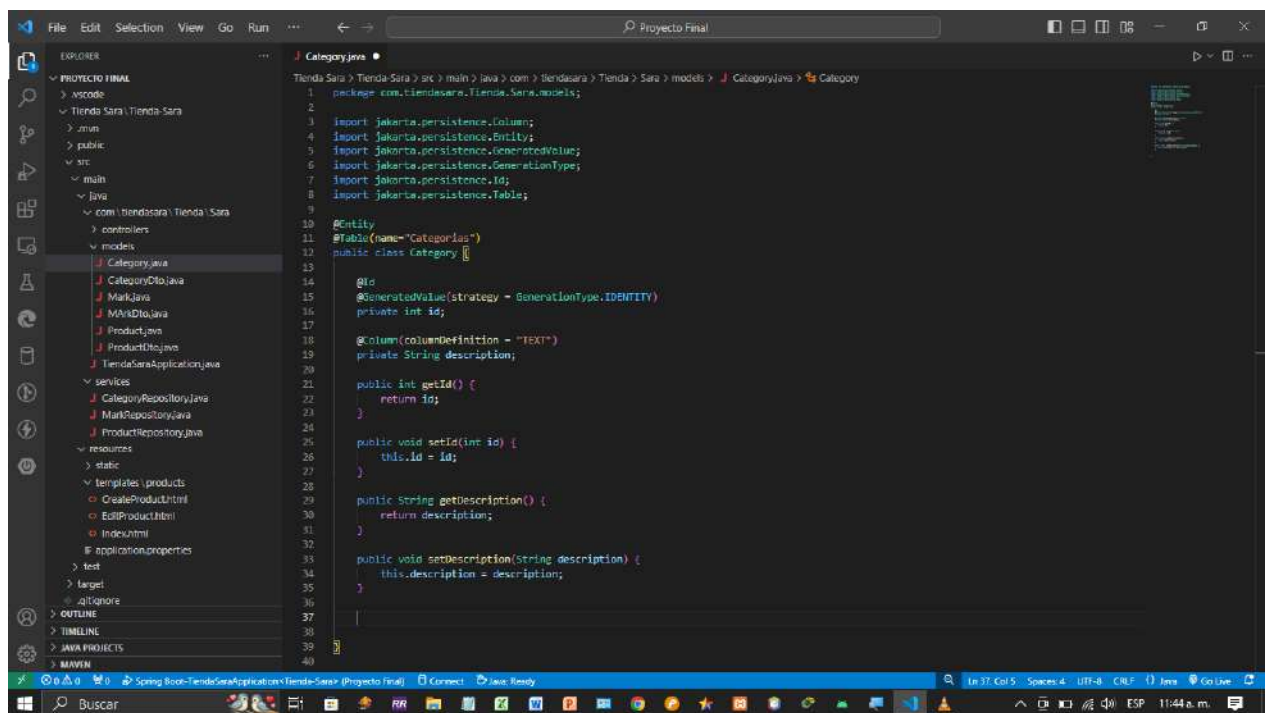
Se selecciona Source Action



Para crear los getters y setters



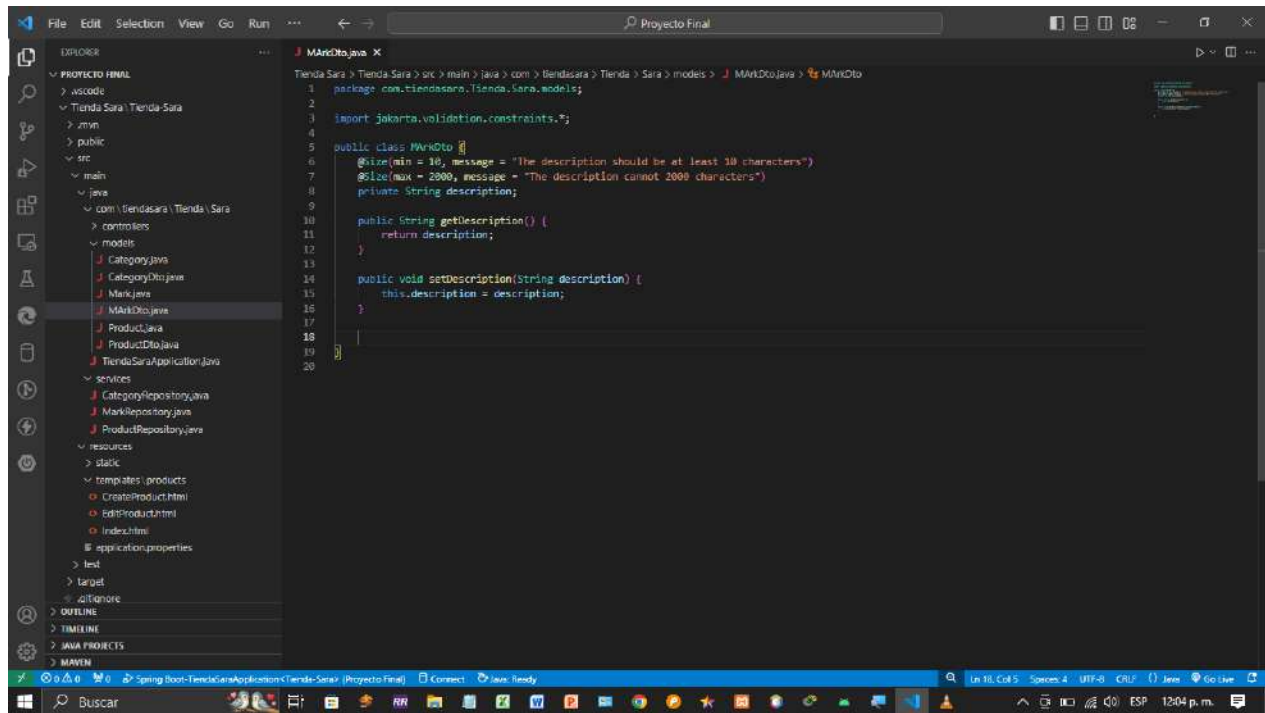
Se seleccionan y se da ok



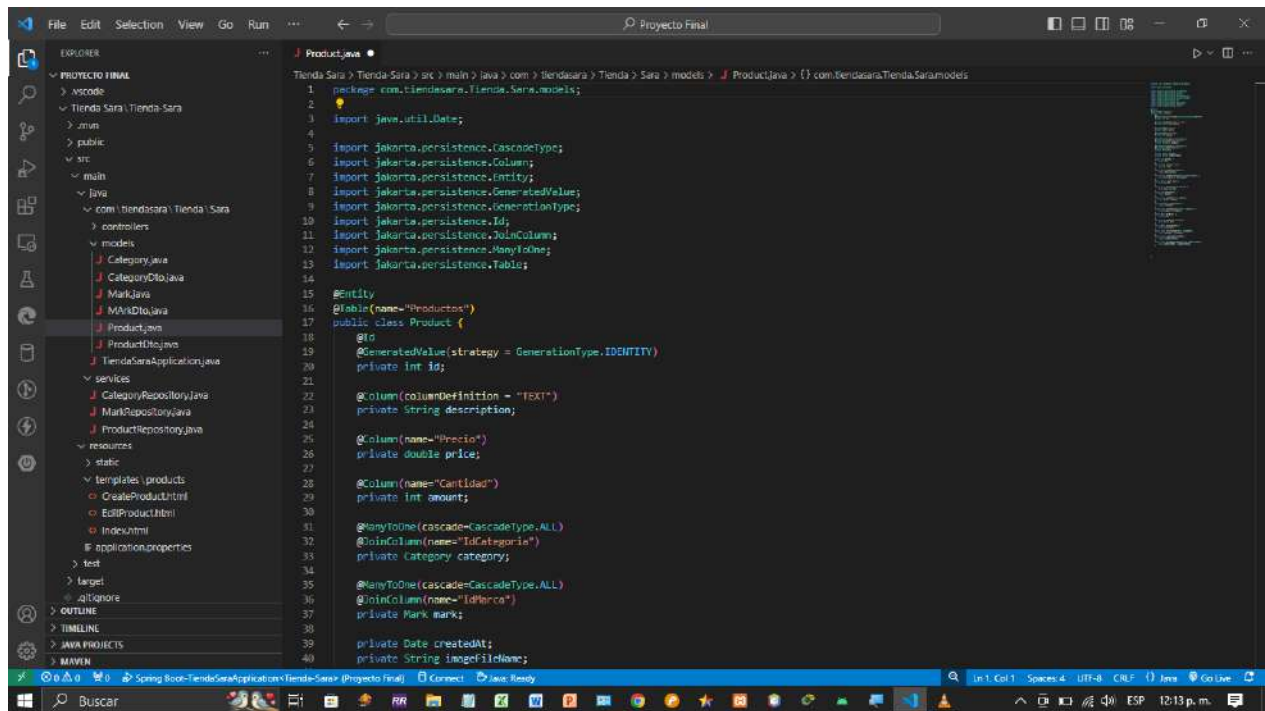
Para que aparezcan en automático

```
1 package com.tiendasara.tienda.sara.models;
2
3 import jakarta.validation.constraints.*;
4
5 public class CategoryOto {
6     @Size(min = 10, message = "The description should be at least 10 characters")
7     @Size(max = 2000, message = "The description cannot 2000 characters")
8     private String description;
9
10    public String getDescription() {
11        return description;
12    }
13
14    public void setDescription(String description) {
15        this.description = description;
16    }
17
18 }
19
20
21
```

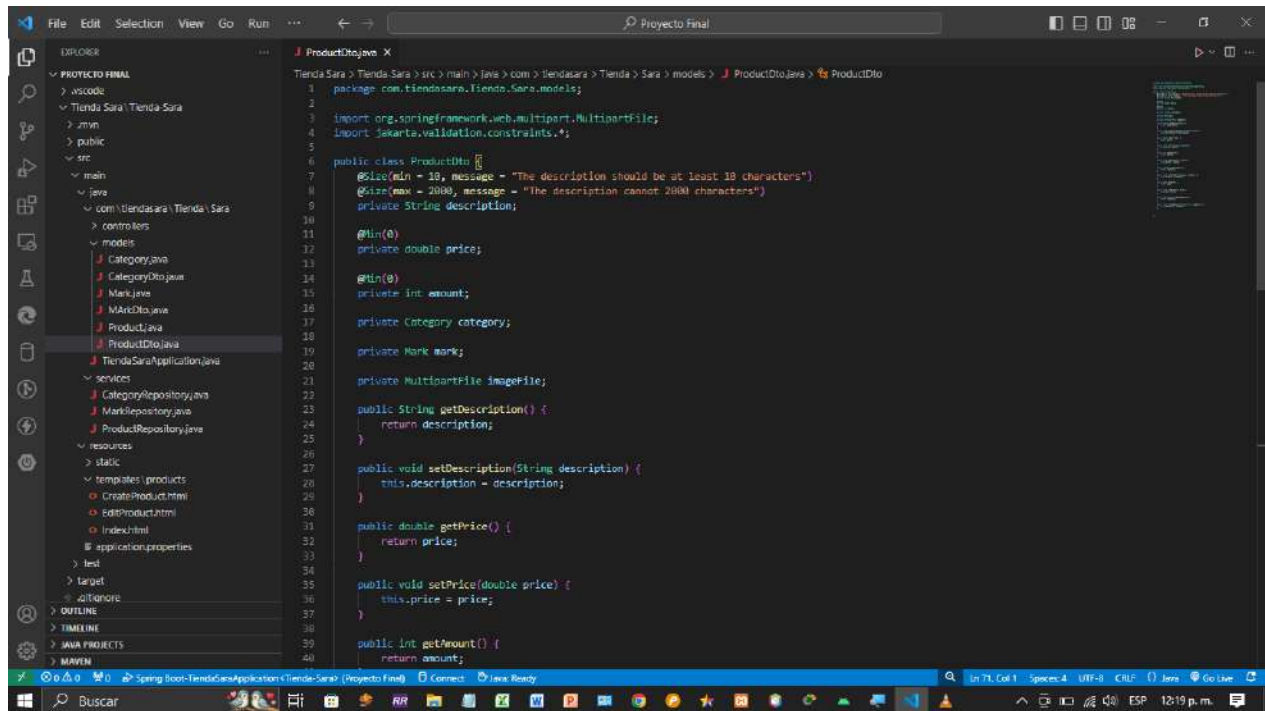
```
1 package com.tiendasara.tienda.sara.models;
2
3 import jakarta.persistence.Column;
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.Table;
9
10 @Entity
11 @Table(name="Marcas")
12 public class Mark {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private int id;
16
17     @Column(columnDefinition = "TEXT")
18     private String description;
19
20     public int getId() {
21         return id;
22     }
23
24     public void setId(int id) {
25         this.id = id;
26     }
27
28     public String getDescription() {
29         return description;
30     }
31
32     public void setDescription(String description) {
33         this.description = description;
34     }
35
36 }
37
38
39
```



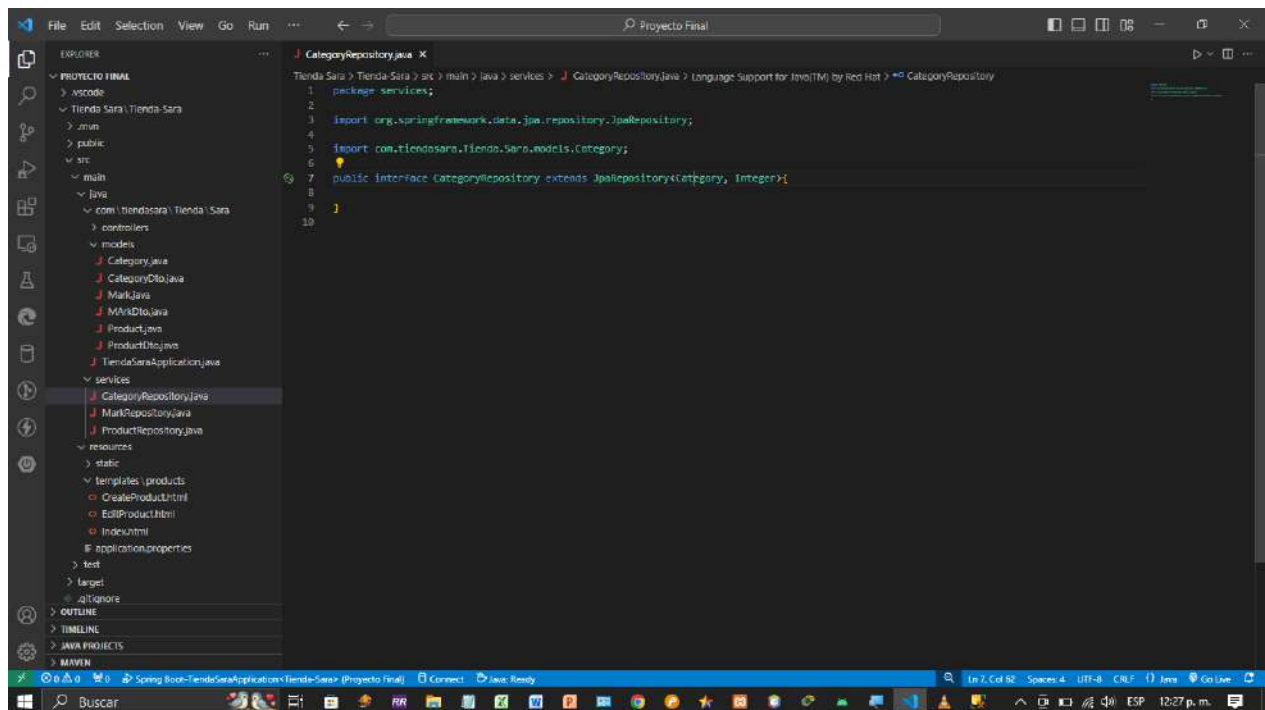
```
1 package com.tiendasarara.tienda.sara.models;
2
3 import jakarta.validation.constraints.*;
4
5 public class MarkDto {
6     @Size(min = 10, message = "The description should be at least 10 characters")
7     @Size(max = 2000, message = "The description cannot 2000 characters")
8     private String description;
9
10
11     public String getDescription() {
12         return description;
13     }
14
15     public void setDescription(String description) {
16         this.description = description;
17     }
18 }
19
20
```

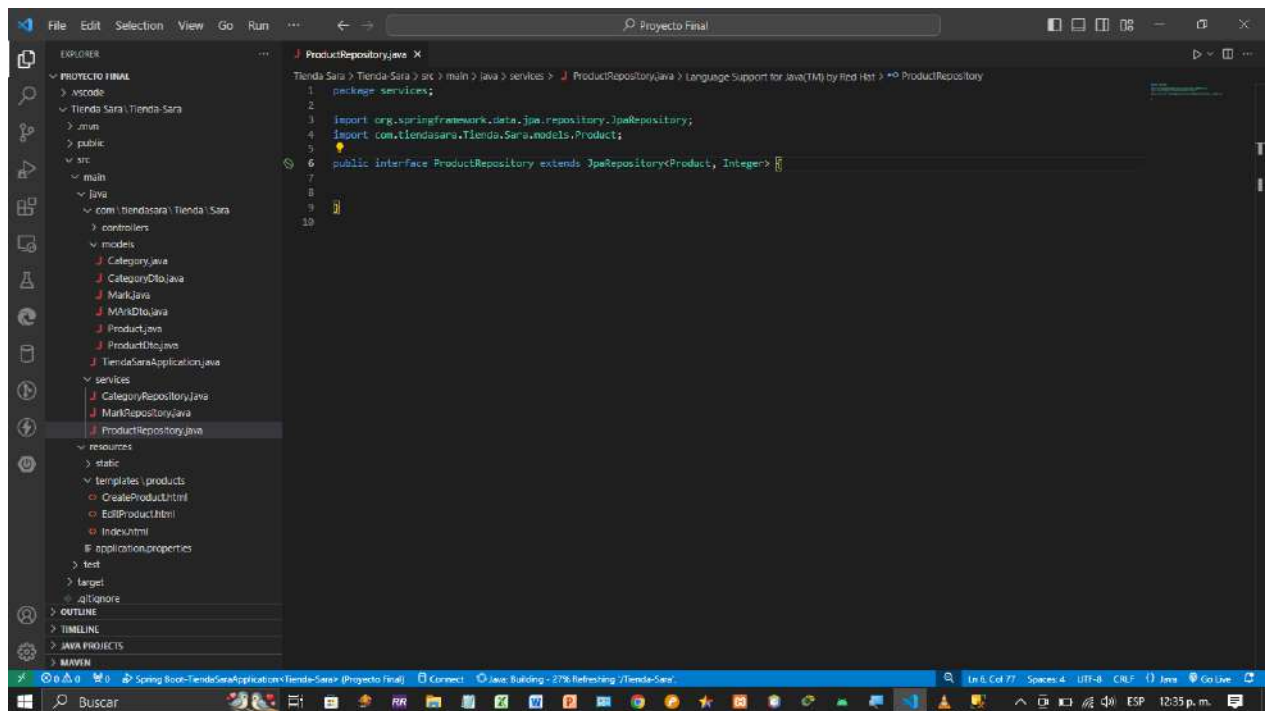
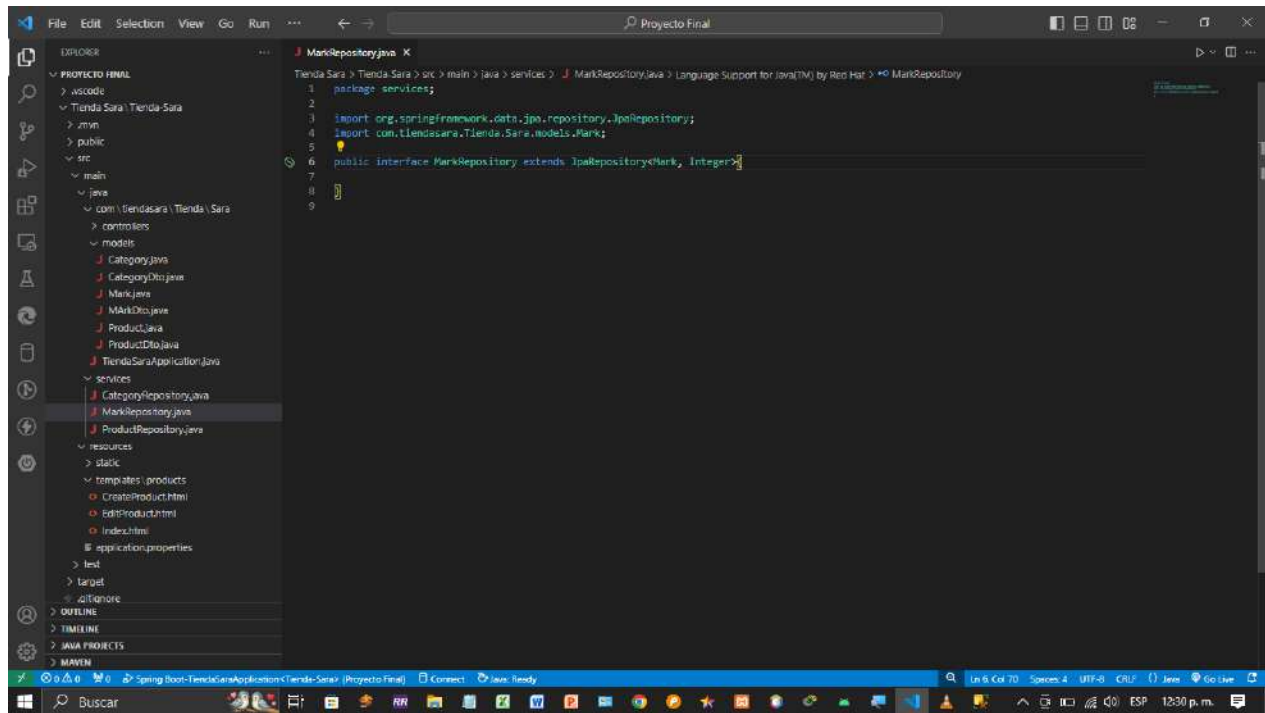


```
1 package com.tiendasarara.tienda.sara.models;
2
3 import java.util.Date;
4
5 import jakarta.persistence.CascadeType;
6 import jakarta.persistence.Column;
7 import jakarta.persistence.Entity;
8 import jakarta.persistence.GeneratedValue;
9 import jakarta.persistence.GenerationType;
10 import jakarta.persistence.Id;
11 import jakarta.persistence.JoinColumn;
12 import jakarta.persistence.ManyToOne;
13 import jakarta.persistence.Table;
14
15 @Entity
16 @Table(name="Productos")
17 public class Product {
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     private int id;
21
22     @Column(columnDefinition = "TEXT")
23     private String description;
24
25     @Column(name="Precio")
26     private double price;
27
28     @Column(name="Cantidad")
29     private int amount;
30
31     @ManyToOne(cascade=CascadeType.ALL)
32     @JoinColumn(name="IdCategoria")
33     private Category category;
34
35     @ManyToOne(cascade=CascadeType.ALL)
36     @JoinColumn(name="IdMarca")
37     private Mark mark;
38
39     private Date createdAt;
40     private String imageUrlName;
41 }
42
```

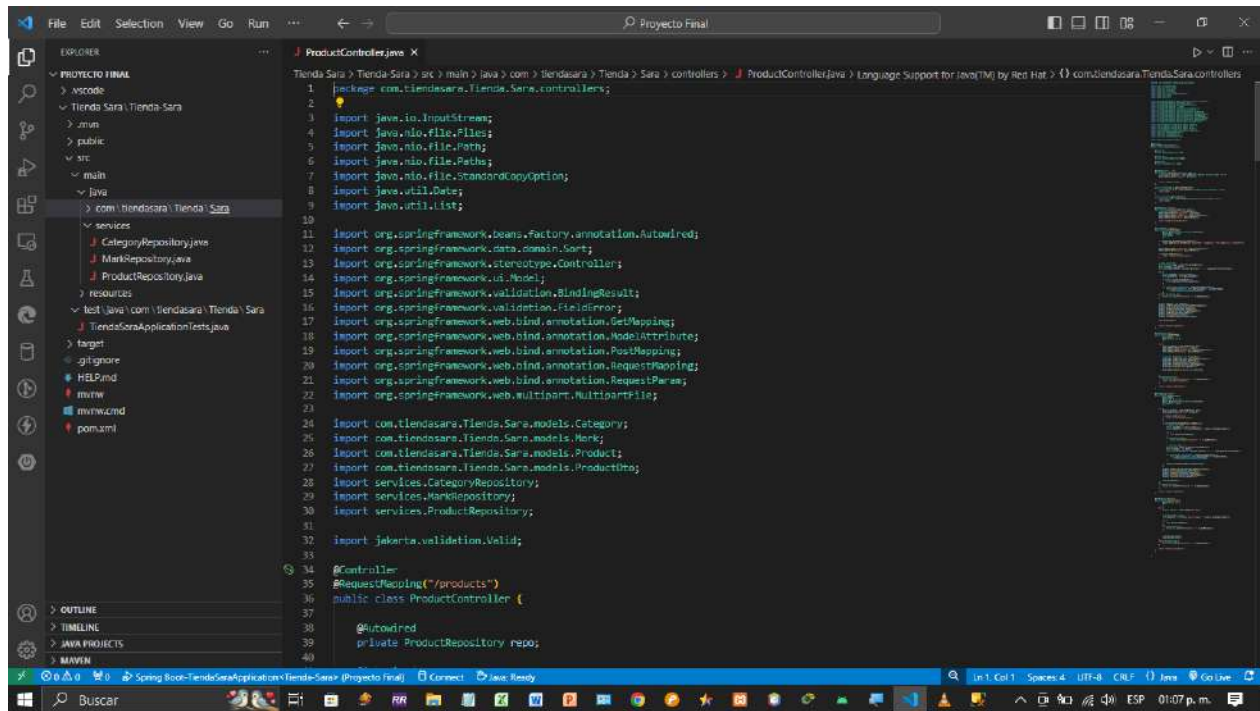



Se obtienen los getters y setters de CategoryDto.java, Mark.java, MarkDto.java, Product.java y ProductDto.java

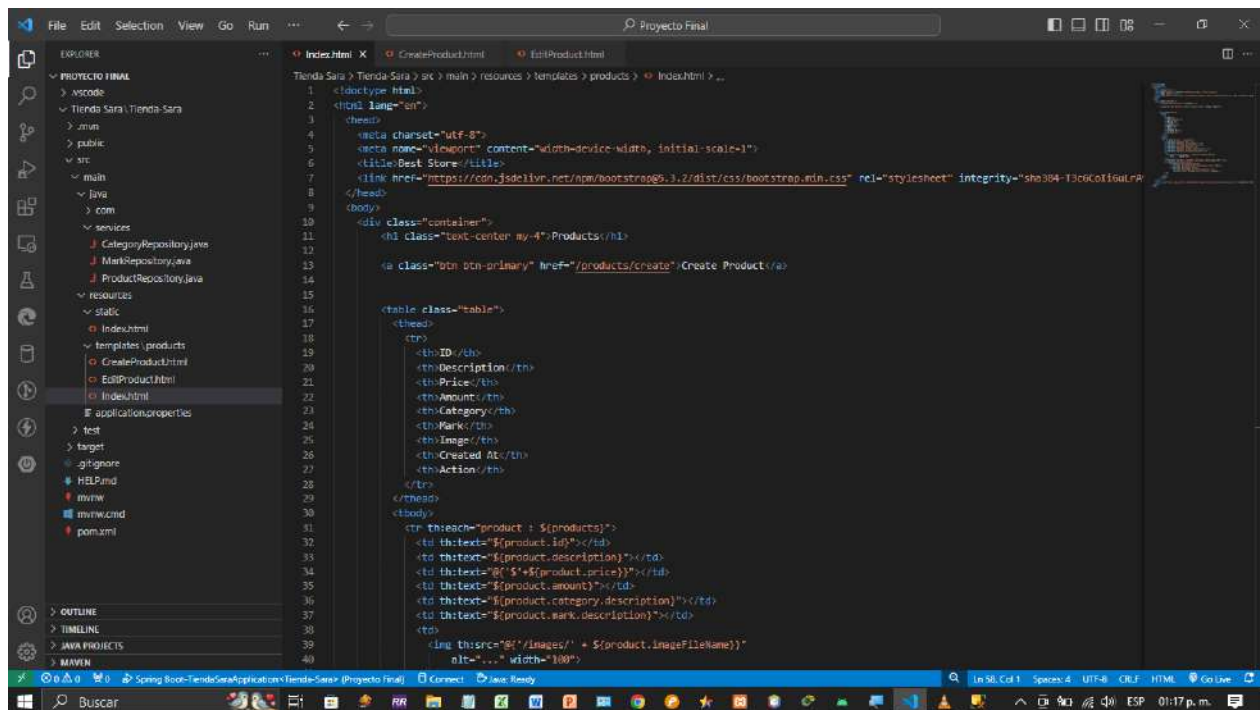


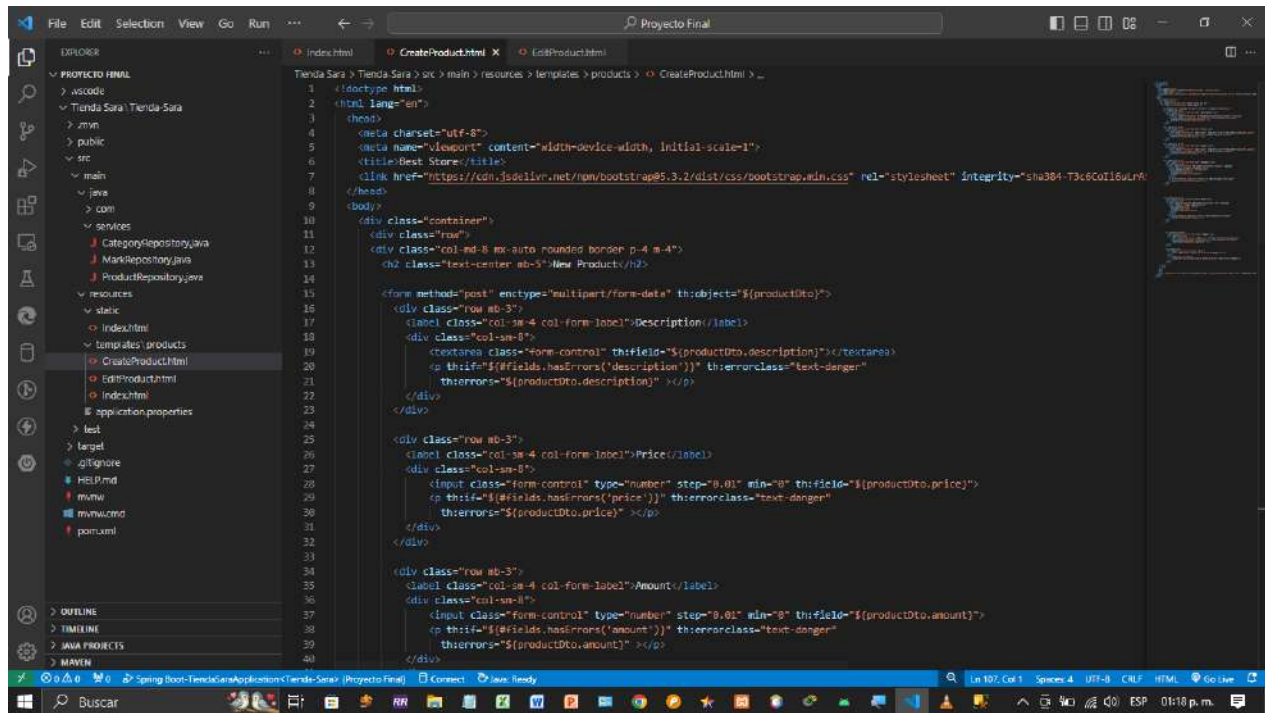


Se definen los extends para Category, Mark, y ProductRepository.

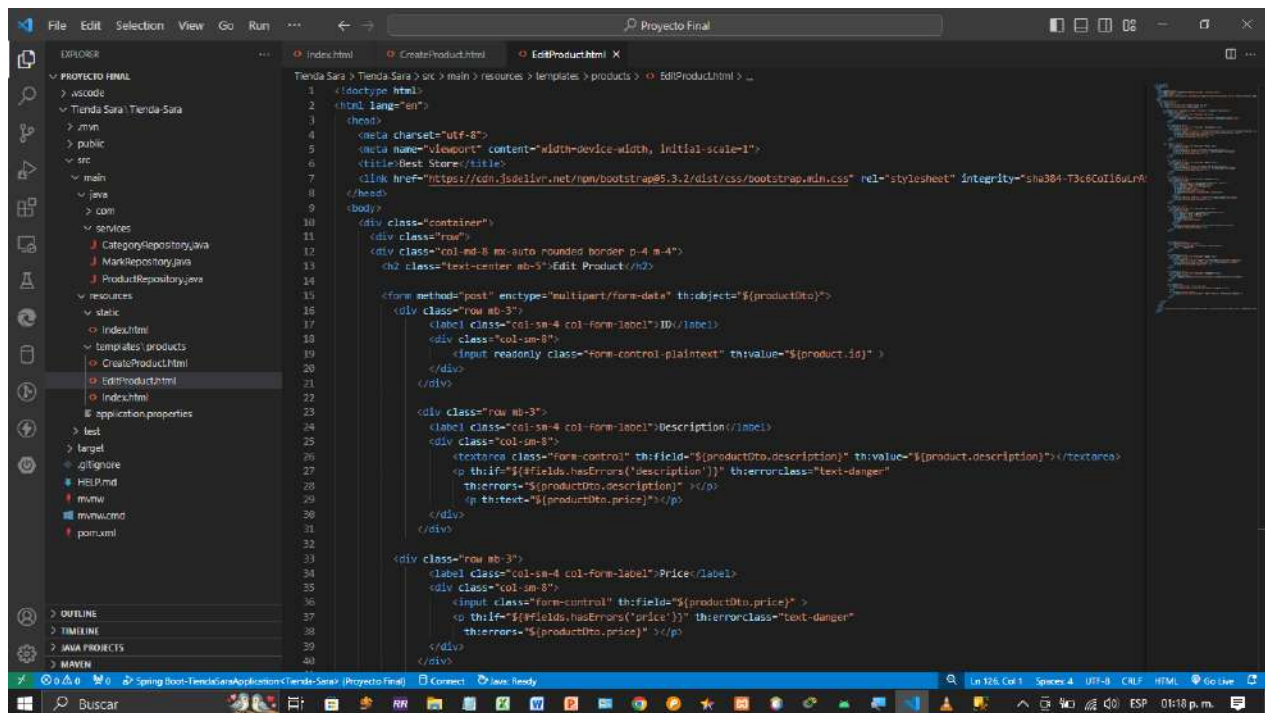


Dentro del archivo productController se definen las importaciones y se verifica que la ruta de los archivos java sea la correcta para hacer las peticiones.



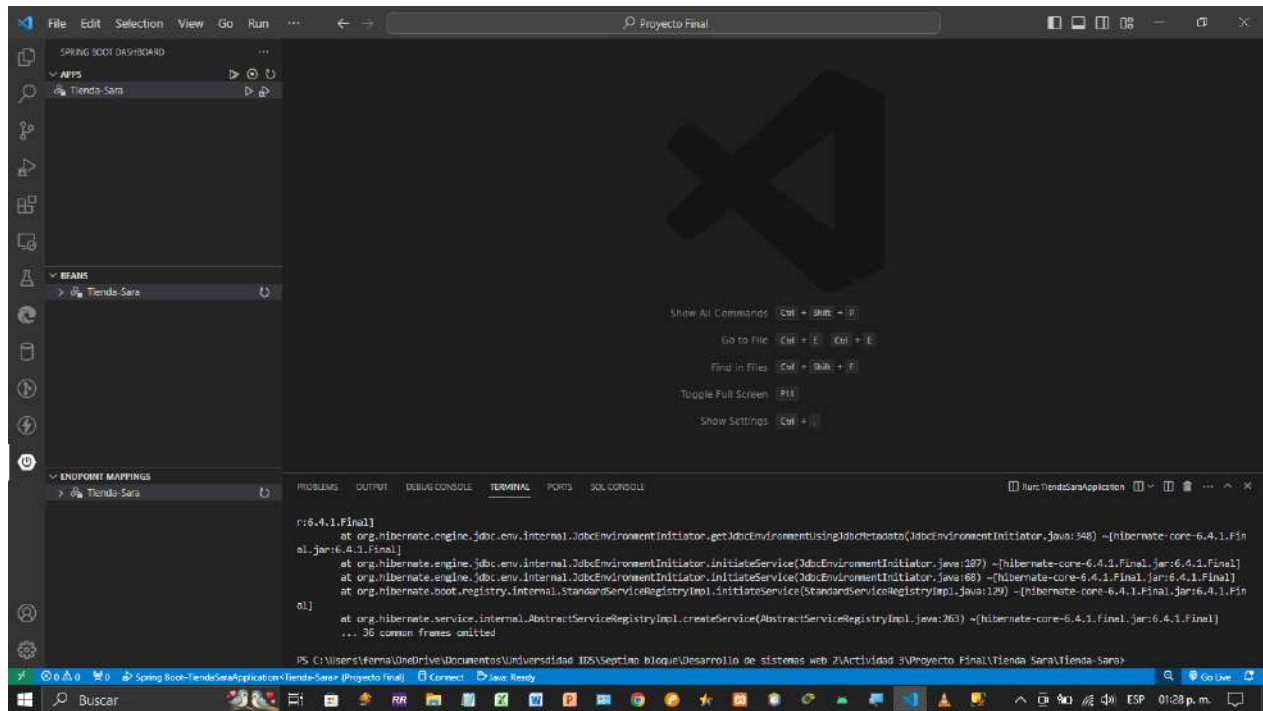


```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Best Store</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3C801611R"
8 </head>
9 <body>
10 <div class="container">
11 <div class="row">
12 <div class="col-md-8 mx-auto rounded border p-4 m-4">
13 <h2 class="text-center mb-5">New Product</h2>
14
15 <form method="post" enctype="multipart/form-data" th:object="${productDto}>
16 <div class="row mb-3">
17 <label class="col-sm-4 col-form-label">Description</label>
18 <div class="col-sm-8">
19 <textarea class="form-control" th:field="${productDto.description}"></textarea>
20 <p th:if="${fields.hasErrors('description')}" th:errorclass="text-danger">
21 <th:errors="${productDto.description}"></p>
22 </div>
23 </div>
24
25 <div class="row mb-3">
26 <label class="col-sm-4 col-form-label">Price</label>
27 <div class="col-sm-8">
28 <input class="form-control" type="number" step="0.01" min="0" th:field="${productDto.price}">
29 <p th:if="${fields.hasErrors('price')}" th:errorclass="text-danger">
30 <th:errors="${productDto.price}"></p>
31 </div>
32 </div>
33
34 <div class="row mb-3">
35 <label class="col-sm-4 col-form-label">Amount</label>
36 <div class="col-sm-8">
37 <input class="form-control" type="number" step="0.01" min="0" th:field="${productDto.amount}">
38 <p th:if="${fields.hasErrors('amount')}" th:errorclass="text-danger">
39 <th:errors="${productDto.amount}"></p>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </body>
46 </html>
```

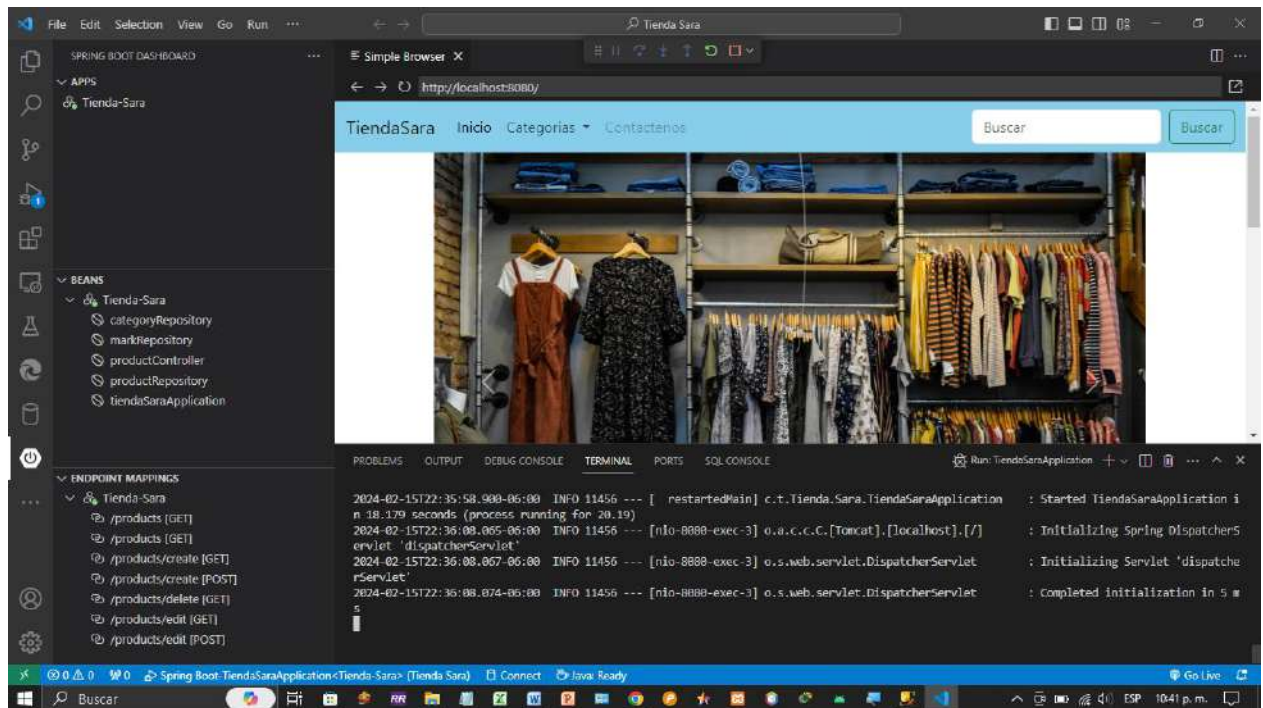


```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Best Store</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3C801611R"
8 </head>
9 <body>
10 <div class="container">
11 <div class="row">
12 <div class="col-md-8 mx-auto rounded border p-4 m-4">
13 <h2 class="text-center mb-5">Edit Product</h2>
14
15 <form method="post" enctype="multipart/form-data" th:object="${productDto}>
16 <div class="row mb-3">
17 <label class="col-sm-4 col-form-label">ID</label>
18 <div class="col-sm-8">
19 <input readonly class="form-control plaintext" th:value="${product.id}">
20 </div>
21 </div>
22
23 <div class="row mb-3">
24 <label class="col-sm-4 col-form-label">Description</label>
25 <div class="col-sm-8">
26 <textarea class="form-control" th:field="${productDto.description}" th:value="${product.description}"></textarea>
27 <p th:if="${fields.hasErrors('description')}" th:errorclass="text-danger">
28 <th:errors="${productDto.description}"></p>
29 <p th:text="${productDto.price}"></p>
30 </div>
31 </div>
32
33 <div class="row mb-3">
34 <label class="col-sm-4 col-form-label">Price</label>
35 <div class="col-sm-8">
36 <input class="form-control" th:field="${productDto.price}">
37 <p th:if="${fields.hasErrors('price')}" th:errorclass="text-danger">
38 <th:errors="${productDto.price}"></p>
39 </div>
40 </div>
41 </div>
42 </div>
43 </body>
44 </html>
```

Se crean los códigos para los templates que permitirán modificar, borrar o editar las tablas.



Se ejecuta la aplicación (Se corrige el nombre de la base de datos a DESKTOP-L3B2UTP:1433 para poder visualizar la página en el dashboard de Visual Studio Code en el puerto 1433



Conclusión.

En conclusión es importante realizar cada parte del proceso antes de crear cualquier base de datos, para así, evitar trabajo innecesario, que la utilidad del diagrama UML representa de forma clara el cómo se relacionara la información antes de diseñar la base de datos, optimizando tiempos a la hora de diseñar y que la información a capturar sea la adecuada en base a las necesidades del negocio o compañía que lo requiera, relacionando cada tabla de acuerdo a la información establecida permitiendo definir desde un inicio el tipo de base de datos que mejor se adecue a las necesidades de los clientes, con respecto al diseño Web es notable el tiempo que se debe invertir para diseñar, ya que aun utilizando plantillas prediseñadas si no se ordena y clasifica de forma adecuada y se revisa la información a mostrar se podría causar un problema en vez de una solución, como al intentar ejecutar el CRUD indicando que hay un error de contexto sin poder localizarlo debiendo invertir más tiempo en encontrar la solución aun cuando el programa corre de forma exitosa.

¿Qué aprendo?

Que aún me falta mucho por aprender en esta materia, que se debe invertir el tiempo necesario para solucionar los detalles si no se está familiarizado con el espacio de trabajo y que estas herramientas son muy útiles a la hora de diseñar, que al utilizarlas de forma adecuada las posibilidades de crear cualquier base de datos es infinita, permitiendo agilizar la información en tiempo y forma, logrando administrar y conseguir de forma sencilla la información que sea necesaria y relevante al momento de requerirla, así como que toda la información a mostrar se debe verificar para no causar errores o daños propios y de terceros.

Enlace Github: <https://github.com/Chifer888/Desarrollo-de-sistemas-web-2.git>

Referencias

¿Qué es una base de datos? (s. f.). <https://www.oracle.com/mx/database/what-is-database/>

The Coder Cave esp. (2021, 31 marzo). *Cómo instalar SQL Server 2019 FÁCIL y GRATIS [2023]* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=SqZQffksa0w>

Sánchez, F. (2022, 14 junio). ¿Qué es el diseño web? y para qué sirve. *La importancia del diseño web*. <https://www.latevaweb.com/que-es-diseno-web>

Coppola, M. (2023, 4 agosto). Qué es el diseño web: definición, características e importancia. *Qué Es el Diseño Web: Definición, Características E Importancia*. <https://blog.hubspot.es/website/disenio-web>

Introducción a las conexiones de datos - Soporte técnico de Microsoft. (s. f.).

<https://support.microsoft.com/es-es/office/introducci%C3%B3n-a-las-conexiones-de-datos-266b64f7-c6cd-4077-9da7-eebc3ae0e4bf#:~:text=conexiones%20de%20datos-,Una%20conexi%C3%B3n%20de%20datos%20es%20un%20v%C3%ADnculo%20din%C3%A1mico%20entre%20un,varias%20conexiones%20de%20datos%20secundarios.>

Slack. (s. f.). *Slack*. <https://app.slack.com/client/T03N5755L74/D06HFN6DTHD>