

**DIPLOMADO DE ACTUALIZACIÓN EN NUEVAS TECNOLOGÍAS PARA EL  
DESARROLLO DE SOFTWARE**

**TALLER UNIDAD 3 FRONTEND “FRUVERMANIA”**

**<https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git>**

**PAOLA ANDREA MONTANCHEZ MONTANCHEZ**

**CODIGO:218034232**

**PRESENTADO A: ING. VICENTE AUX REVELO**

**UNIVERSIDAD DE NARIÑO**

**INGENIERIA DE SISTEMAS**

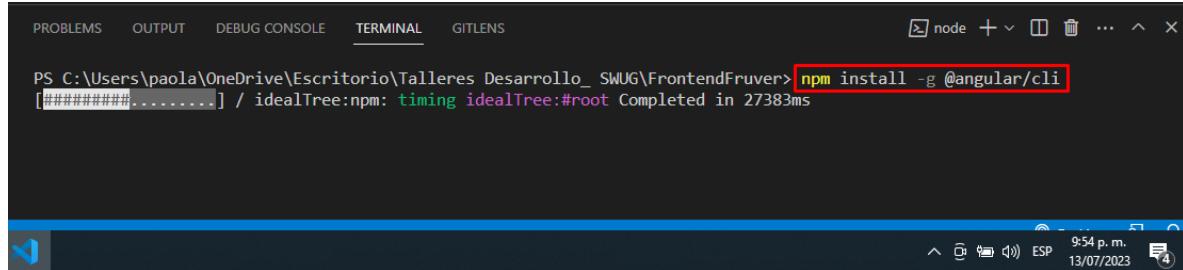
**2023**

**Desarrollar una aplicación Frontend en Angular que haga uso del Backend desarrollado en el taller anterior.**

## 1. Creación de componentes, uso de ngModel, RouterLink, Servicios

### 1.1 Instalación

Instalación de Angular haciendo uso del comando **npm install -g @angular/cli**



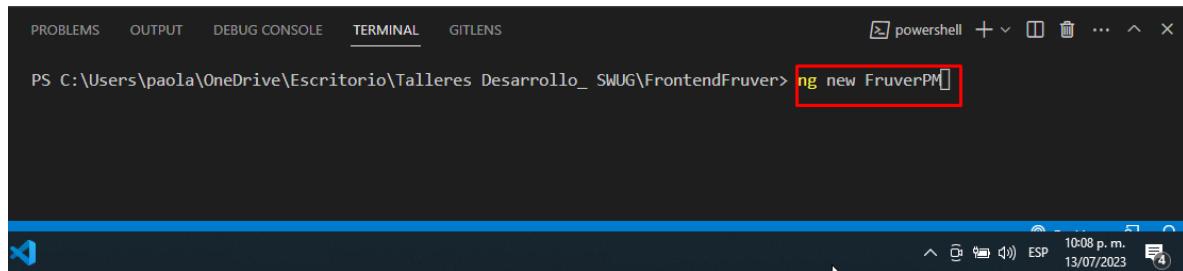
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
```

```
PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo_SWUG\FrontendFruver> npm install -g @angular/cli
[#####.....] / idealTree:pm: timing idealTree:#root Completed in 27383ms
```

```
node + ⌂ ... ^ x
```

```
9:54 p.m. ESP 13/07/2023
```

Creación de un workspace con el nombre **FruverPM**



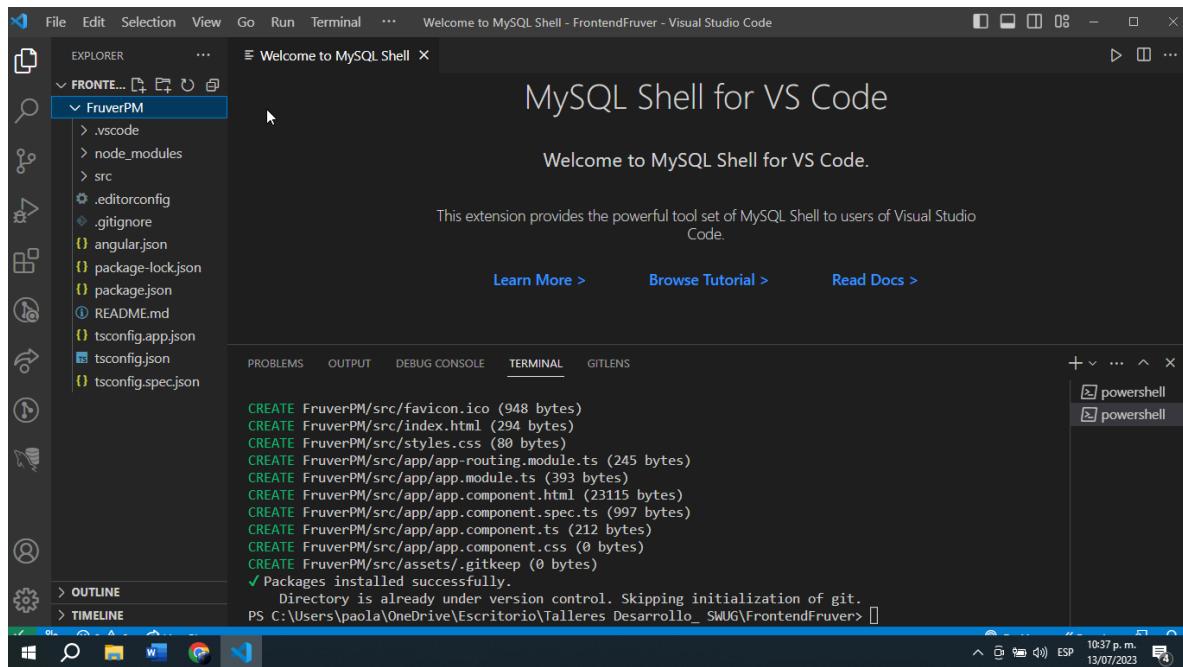
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
```

```
PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo_SWUG\FrontendFruver> ng new FruverPM
```

```
powershell + ⌂ ... ^ x
```

```
10:08 p.m. ESP 13/07/2023
```

**Workspace FruverPM**



The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the directory structure of the 'FruverPM' workspace, including '.vscode', 'node\_modules', 'src', '.editorconfig', '.gitignore', 'angular.json', 'package-lock.json', 'package.json', 'README.md', 'tsconfig.app.json', 'tsconfig.json', and 'tsconfig.spec.json'.
- Terminal:** Displays the output of the 'ng new FruverPM' command, showing the creation of various files and components.
- MySQL Shell for VS Code extension:** A central panel titled "MySQL Shell for VS Code" with the message "Welcome to MySQL Shell for VS Code." It also displays the message "This extension provides the powerful tool set of MySQL Shell to users of Visual Studio Code." and links to "Learn More", "Browse Tutorial", and "Read Docs".
- Status Bar:** Shows the path "C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo\_SWUG\FrontendFruver>" and the date and time "10:37 p.m. 13/07/2023".

Cuando se crea la aplicación se crean varios archivos y carpetas entre los principales tenemos:

**Src:** Esta carpeta engloba todo el código fuente de la aplicación. Es el lugar donde se encuentra la lógica y los archivos necesarios para construir la aplicación.

**App:** Esta carpeta contiene el código principal de la aplicación. Dentro de esta carpeta encontramos componentes, servicios y modelos.

**Componentes:** Son elementos que se pueden reutilizar y componen la interfaz de usuario de la aplicación. Los componentes encapsulan la lógica y la presentación relacionada en una sola unidad. Cada componente puede tener su propio HTML, CSS y lógica TypeScript.

**Servicios:** Son clases que proporcionan las funcionalidades compartidas. Permiten conectarse con el Backend para obtener y enviar datos.

**Modelos:** Representan la estructura de almacenamiento de datos en la aplicación. En TypeScript, los modelos son clases que definen las propiedades y métodos para interactuar con los datos de manera coherente.

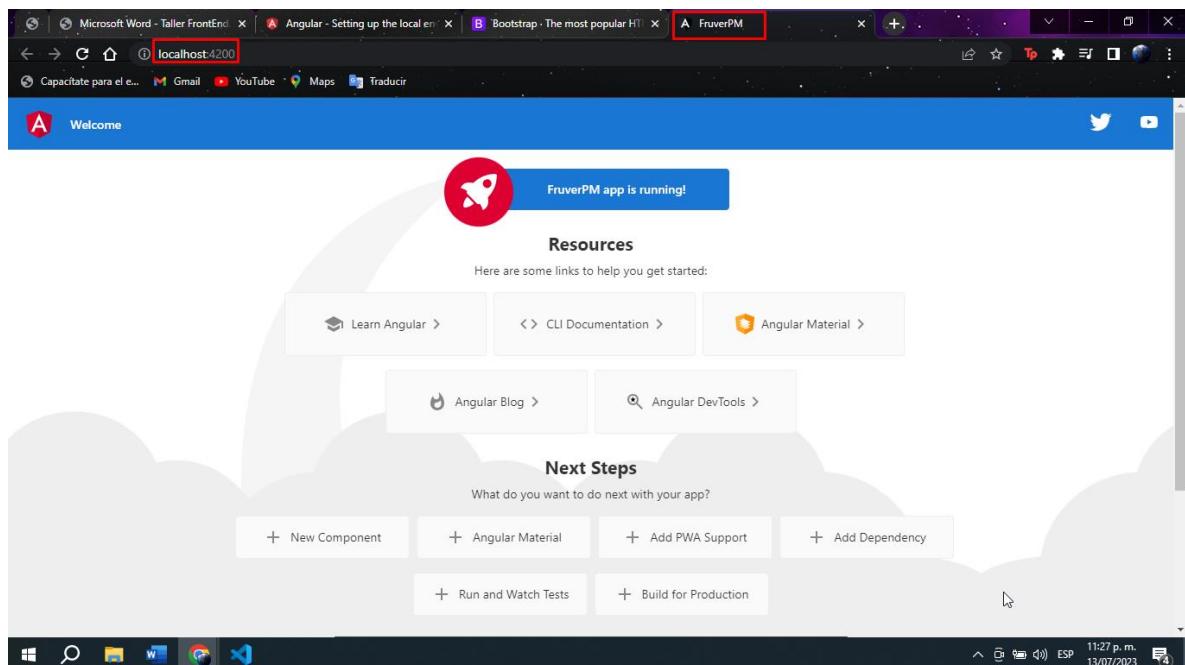
**Assets:** Contiene archivos estáticos de la aplicación, como imágenes, estilos CSS, archivos JSON, fuentes, etc. Estos archivos se pueden usar en la aplicación y se incluyen en el paquete final cuando se realiza la construcción.

**Index:** Es el punto de entrada de la aplicación. Aquí se carga el módulo raíz de Angular y se inicia la aplicación.

**tsconfig.json:** Es un archivo de configuración utilizado por el compilador de TypeScript para especificar cómo se deben compilar los archivos TypeScript en JavaScript.

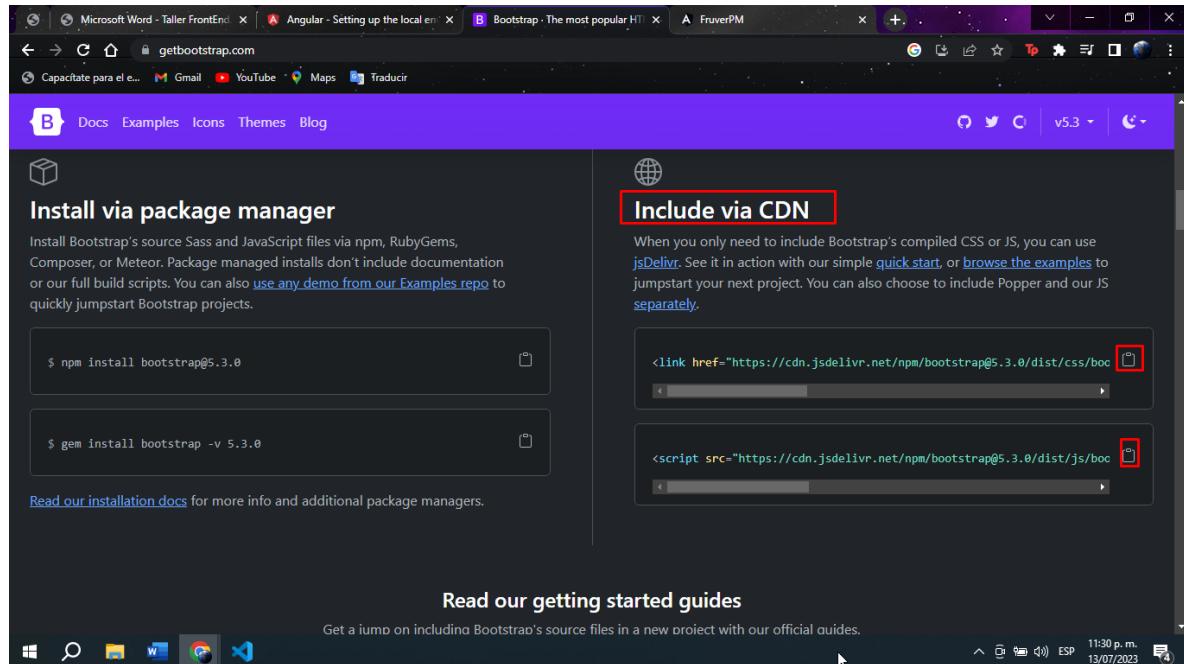
## Ejecutar la aplicación

Para ejecutar la aplicación en el navegador se utiliza el comando **ng serve --open**



## Instalación de CDN de Bootstrap

Ingresamos a la pagina oficial de Bootstrap y seleccionamos los links de CDN



The screenshot shows the official Bootstrap website. On the left, there's a sidebar with icons for Docs, Examples, Icons, Themes, and Blog. The main content area has a purple header with a globe icon and the text 'Include via CDN'. Below it, there's a section titled 'Install via package manager' with two code snippets for npm and gem. To the right, there's another section titled 'Include via CDN' with two code snippets for linking CSS and JS files from cdn.jsdelivr.net.

**Install via package manager**

```
$ npm install bootstrap@5.3.0
```

```
$ gem install bootstrap -v 5.3.0
```

[Read our installation docs](#) for more info and additional package managers.

**Include via CDN**

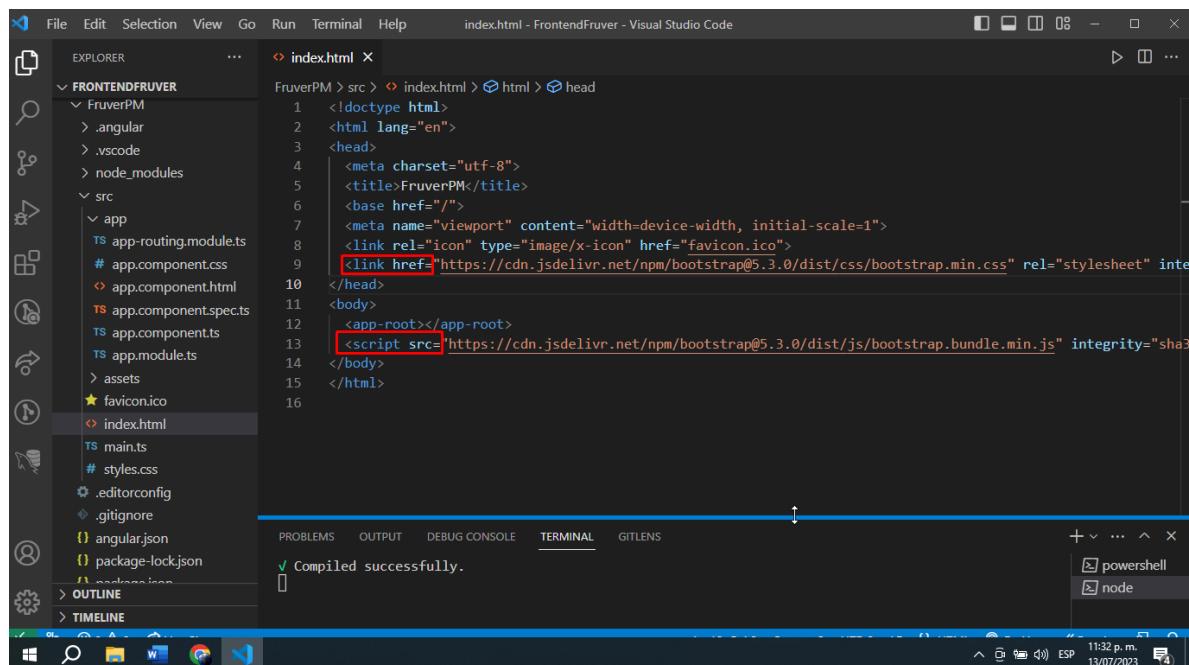
```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4d6戎tBhLH2wJkYxjW1RZDmCJLZoEzG0QKxNpYfZJqFjGZPZMnqJmZqOvXrJ3" crossorigin="anonymous" />
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-KnHt8GjUdM8/2Z4R+NkKw/350y6BwzDZDZ/2UoC9v4LqQj7mIz0/0J&gt;" crossorigin="anonymous" />
```

**Read our getting started guides**

Get a jump on including Bootstrap's source files in a new project with our official guides.

Agregamos los enlaces en el archivo index.html



The screenshot shows Visual Studio Code with the 'index.html' file open. The file contains the following HTML code:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>FruverPM</title>
<base href="/">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4d6戎tBhLH2wJkYxjW1RZDmCJLZoEzG0QKxNpYfZJqFjGZPZMnqJmZqOvXrJ3" crossorigin="anonymous" />
</head>
<body>
<app-root></app-root>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-KnHt8GjUdM8/2Z4R+NkKw/350y6BwzDZDZ/2UoC9v4LqQj7mIz0/0J&gt;" crossorigin="anonymous" />
</body>
</html>
```

The 'index.html' file is located in the 'FRONTENDFRUVER/FruverPM/src/app' directory. The status bar at the bottom shows 'Compiled successfully.'

## 1.2 Creación de componentes

Los componentes en Angular son bloques de construcción reutilizables que contienen tanto la lógica como la interfaz de usuario de una parte específica de la aplicación.

Para la creación de los componentes se utiliza el comando → **ng generate component nombre-del-componente** o también su abreviatura **ng g c nombre-del-componente**.

Angular CLI generará automáticamente todos los archivos necesarios para el componente y los ubica en el directorio correspondiente.

**nombre-del-componente.component.ts:** Este archivo contiene la clase del componente, que contiene la lógica del mismo, propiedades y métodos.

**nombre-del-componente.component.html:** Aquí se encuentra la plantilla o vista del componente. Define cómo se visualiza el componente en la interfaz de usuario.

**nombre-del-componente.component.css:** Opcionalmente, puedes agregar estilos CSS específicos para el componente.

**nombre-del-componente.component.spec.ts:** Archivo de prueba unitaria para el componente.

Para nuestra aplicación crearemos los siguientes componentes:

**Lista-productos:** Este componente va a permitir listar los productos

**Editar-productos:** Este componente permite realizar la gestión de productos, como agregar, editar o eliminar productos de la base de datos.

**Detalle Producto:** Este componente permite mostrar información detallada sobre un producto en particular.

**Lista-compras:** Este componente permite realizar para la gestión de compras. como agregar, editar o eliminar productos de la base de datos.

**Login:** Este componente va a permitir gestionar el proceso de autenticación de usuarios en el sistema.

## Creación el componente lista-productos y editar-productos

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "FRUVERPM". The "src/app" folder is expanded, and "lista-productos" is selected.
- Editor:** The "index.html" file is open, displaying the Angular template code.
- Terminal:** The command "ng generate component lista-productos" is being run, and the output shows the creation of the component files: "CREATE src/app/lista-productos/lista-productos.component.html", "CREATE src/app/lista-productos/lista-productos.component.spec.ts", "CREATE src/app/lista-productos/lista-productos.component.ts", and "CREATE src/app/lista-productos/lista-productos.component.css".
- Status Bar:** Shows the date and time as "13/07/2023 11:39 p.m."

## Creación del componente detalle-producto

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "FRUVERPM". The "src/app" folder is expanded, and "detalle-producto" is selected.
- Editor:** The "app-routing.module.ts" file is open, showing the configuration of routes. A specific route for "detalle-producto" is defined.
- Terminal:** The command "ng g c detalle-producto" is being run, and the output shows the creation of the component files: "CREATE src/app/detalle-producto/detalle-producto.component.html", "CREATE src/app/detalle-producto/detalle-producto.component.spec.ts", "CREATE src/app/detalle-producto/detalle-producto.component.ts", and "CREATE src/app/detalle-producto/detalle-producto.component.css".
- Status Bar:** Shows the date and time as "29/07/2023 8:20 a.m."

## Creación del componente lista-compras

```
File Edit Selection View Go Run Terminal Help app-routing.module.ts - FruverPM - Visual Studio Code
EXPLORER ... TS producto.model.ts TS app-routing.module.ts x app.component.html TS producto.service.ts package-lock.json TS app.modules.ts in ...
FRUVERPM
> angular
> .vscode
> node_modules
src
  app
    > detalle-producto
    > editar-productos
      lista-compras
        > lista-compras
        > lista-productos
        > shared
      TS app-routing.module.ts
      # app.component.css
      < app.component.html
      TS app.component.spe...
      TS app.component.ts
      TS app.module.ts
      > assets
      ★ favicon.ico
      < index.html
      TS mains
      # styles.css
      .editorconfig
      .gitignore
      (1) angular.json
      (1) package-lock.json
      (1) package.json
      < OUTLINE
      < TIMELINE
src > app > TS app-routing.module.ts ...
1
2
3 import { NgModule } from '@angular/core';
4 import { RouterModule, Routes } from '@angular/router';
5 import { ListaProductosComponent } from './lista-productos/lista-productos.component';
6 import { EditarProductosComponent } from './editar-productos/editar-productos.component';
7
8 //Definen las rutas y se las anexa a un componente
9 const routes: Routes = [
10   {path: 'productos', component: ListaProductosComponent},
11   {path: 'productos/editar/:idProducto', component: EditarProductosComponent },
12   {path: 'productos/agregar', component: EditarProductosComponent },
13   {path:'**',redirectTo:'/productos',pathMatch:'full'}
14 ];
15
16 @NgModule({
17   imports: [RouterModule.forRoot(routes)],
18   exports: [RouterModule]
19 })
20 export class AppRoutingModule { }
```

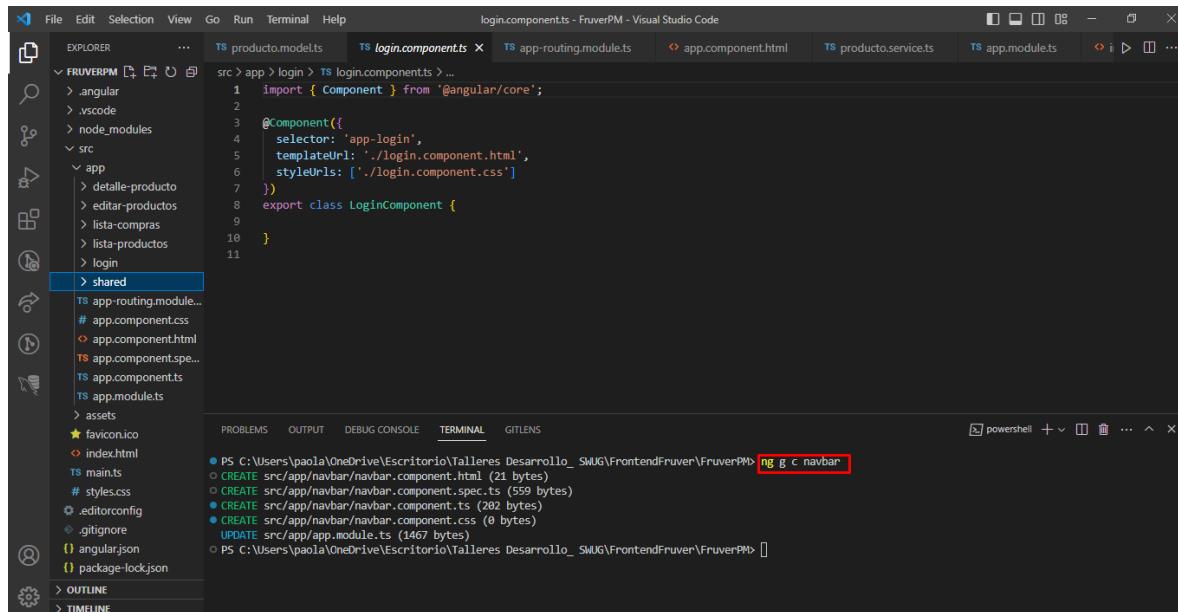
## **Creación del componente login**

```
src/app/login/login.component.html (28 bytes)
CREATE src/app/login/login.component.spec.ts (552 bytes)
CREATE src/app/login/login.component.ts (198 bytes)
CREATE src/app/login/login.component.css (0 bytes)
```

### 1.3 Creación de la carpeta shared

Creamos una carpeta con el nombre shared. Esta carpeta generalmente se utiliza para almacenar archivos y recursos que son compartidos entre múltiples componentes y módulos dentro de la aplicación. Los servicios son un ejemplo típico de recursos compartidos, y por eso a menudo se colocan en la carpeta servicios dentro de shared.

Dentro de esta carpeta procedemos a crear el componente navbar. uno de los elementos más comunes y útiles en una aplicación web. Su función principal es proporcionar una barra de navegación accesible en toda la aplicación, permitiendo a los usuarios moverse fácilmente entre diferentes páginas, secciones o funcionalidades.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "FRUVERPM". A folder named "shared" is selected.
- Code Editor:** Displays the file "login.component.ts" with code related to a login component.
- Terminal:** Shows a PowerShell terminal window with the command "ng g c navbar" highlighted in red, indicating it is being run or has been run.
- Status Bar:** Shows the path "PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo\_SWIG\FrontendFruver\FruverPM" and the command "ng g c navbar".

### 1.4 Servicios

Como se mencionó anteriormente otra parte clave son los servicios porque permiten separar la lógica relacionada con los datos, la comunicación con el servidor y otras operaciones comunes del resto de la aplicación, lo que proporciona varios beneficios importantes como la reutilización de código, evitando así la necesidad de duplicar código.

Para crear un servicio en Angular, se utiliza el comando →**ng generate service nombre-del-servicio**.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under the 'FRUVERPM' folder, including 'src' and 'shared' directories.
- Editor:** The file 'compras.service.ts' is open, containing the following code:
 

```
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class ComprasService {
7
8   constructor() { }
9
10}
```
- Terminal:** Shows the command line output with several 'CREATE' entries and one 'ng generate service' command, all highlighted with red boxes:
  - CREATE src/app/shared/cliente.service.spec.ts (362 bytes)
  - CREATE src/app/shared/cliente.service.ts (136 bytes)
  - CREATE src/app/shared/compras.service.spec.ts (362 bytes)
  - CREATE src/app/shared/compras.service.ts (136 bytes)
  - CREATE src/app/shared/autenticacion.service.spec.ts (392 bytes)
  - CREATE src/app/shared/autenticacion.service.ts (142 bytes)
  - PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo\_SNUG\FrontendFruver\FruverPM\src\app\shared> ng generate service cliente
  - PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo\_SNUG\FrontendFruver\FruverPM\src\app\shared> ng generate service compras
  - PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo\_SNUG\FrontendFruver\FruverPM\src\app\shared> ng generate service autenticacion
- Bottom Status Bar:** Shows the current file as 'compras.service.ts', line 1, column 1, spaces: 2, encoding: UTF-8, and date/time: 2:02 p.m. 29/07/2023.

## 1.5 Modelos

Para definir modelos en Node.js y Angular, podemos utilizar TypeScript para crear interfaces que representen la estructura de los datos de los productos, clientes y compras. TypeScript permite definir tipos estáticos, lo que nos ayuda a tener un código más seguro y claro. A continuación, definimos los modelos para cada tabla.

### Definición de modelo para productos

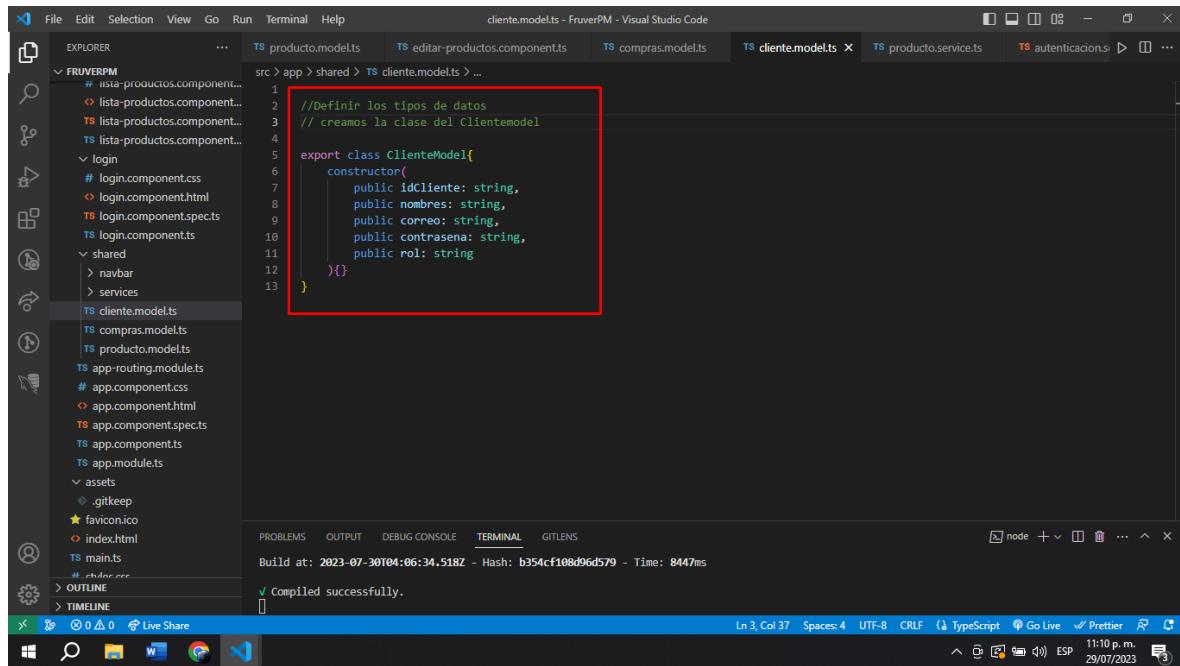
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under the 'FRUVERPM' folder, including 'src' and 'shared' directories.
- Editor:** The file 'producto.model.ts' is open, containing the following code:
 

```
1 //Definir los tipos de datos
2
3 // Creamos la clase del ProductoModel
4 export class ProductoModel {
5   constructor(public idProducto: string, public nombre: string, public detalle:string) {
6
7
8   }
9 }
```
- Terminal:** Shows the command line output with the command 'ng serve --open' and its progress:
 

```
PS C:\Users\paola\OneDrive\Escritorio\Talleres Desarrollo_SNUG\FrontendFruver\FruverPM> ng serve --open
: Generating browser application bundles (phase: building)...[
```
- Bottom Status Bar:** Shows the current file as 'producto.model.ts', line 9, column 2, spaces: 2, encoding: CRLF, and date/time: 10:21 p.m. 29/07/2023.

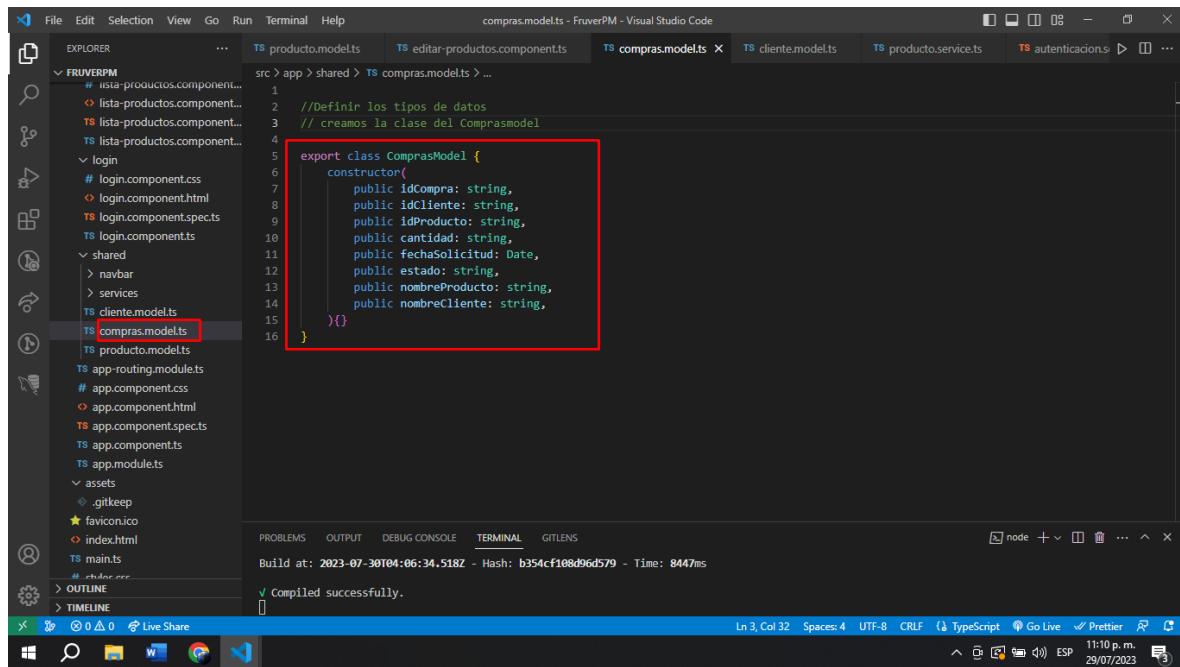
## Definición de modelo para cliente



The screenshot shows the Visual Studio Code interface with the file 'cliente.model.ts' open. The code defines a class 'ClienteModel' with properties: idCliente, nombres, correo, contrasena, and rol. A red box highlights the class definition.

```
//Definir los tipos de datos
// creamos la clase del Clientemodel
export class ClienteModel{
    constructor(
        public idCliente: string,
        public nombres: string,
        public correo: string,
        public contrasena: string,
        public rol: string
   ){}
}
```

## Definición de modelo para compras



The screenshot shows the Visual Studio Code interface with the file 'compras.model.ts' open. The code defines a class 'ComprasModel' with properties: idCompra, idCliente, idProducto, cantidad, fechasolicitud, estado, nombreProducto, and nombreCliente. A red box highlights the class definition.

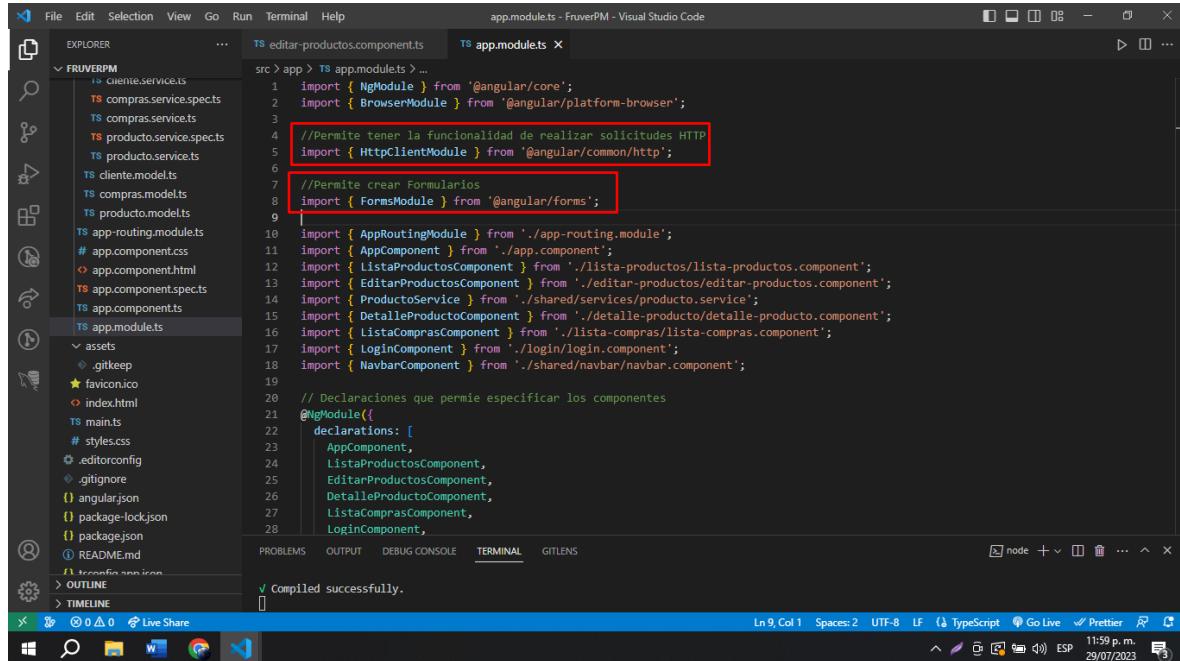
```
//Definir los tipos de datos
// creamos la clase del Comprasmovel
export class ComprasModel {
    constructor(
        public idCompra: string,
        public idCliente: string,
        public idProducto: string,
        public cantidad: string,
        public fechasolicitud: Date,
        public estado: string,
        public nombreProducto: string,
        public nombreCliente: string,
   ){}
}
```

### 1.6 Configuración del archivo app.module.ts

El archivo `app.module.ts` es uno de los archivos más importantes dentro de una aplicación, ya que es el módulo principal que define cómo se organiza y configura la aplicación. Aquí se declaran todos los componentes, servicios, directivas, y otros módulos que se utilizarán en la aplicación.

Para el desarrollo de la aplicación haremos uso de un módulo llamado `FormsModule` que permite trabajar con formularios de manera sencilla y eficiente. Al igual que con `HttpClientModule`, para hacer solicitudes HTTP.

Cabe enfatizar, que para utilizar `ngModel`, es importante que importar el módulo `FormsModule` en el archivo. Una vez que se realice este proceso se podrá aprovechar el poder del enlace bidireccional de datos que proporciona `ngModel` para crear formularios interactivos en tu aplicación Angular.



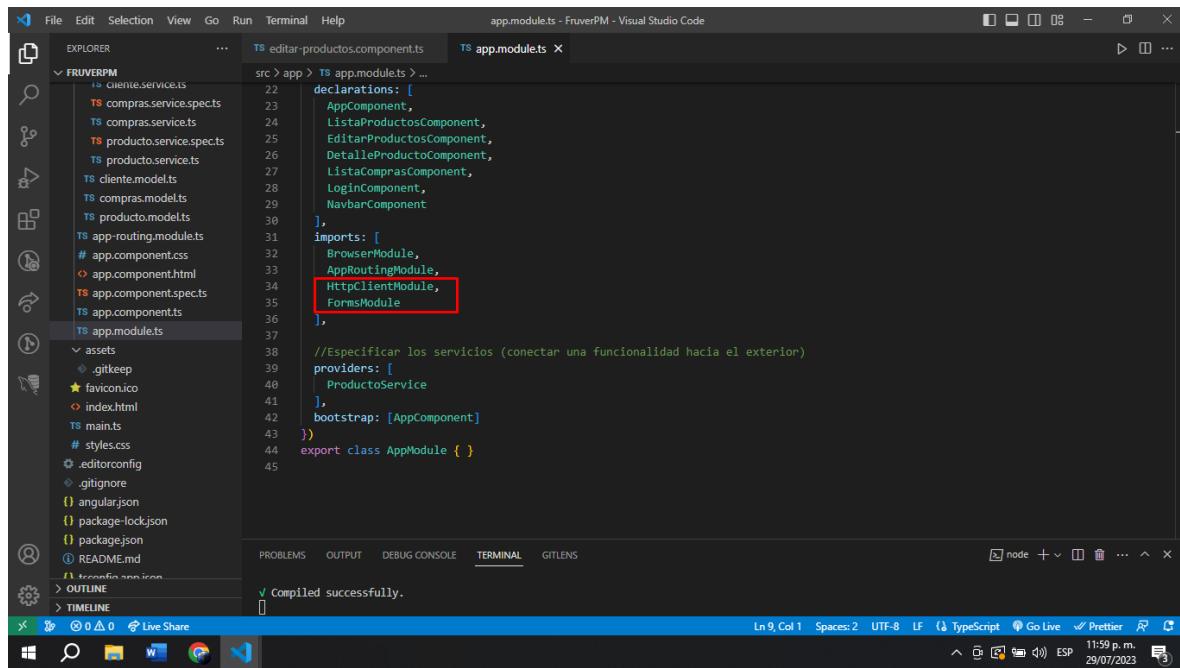
```
File Edit Selection View Go Run Terminal Help app.module.ts - FruberPM - Visual Studio Code

src > app > TS app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 //Permite tener la funcionalidad de realizar solicitudes HTTP
5 import { HttpClientModule } from '@angular/common/http';
6
7 //Permite crear Formularios
8 import { FormsModule } from '@angular/forms';
9
10 import { AppRoutingModule } from './app-routing.module';
11 import { AppComponent } from './app.component';
12 import { ListaProductosComponent } from './lista-productos/lista-productos.component';
13 import { EditarProductosComponent } from './editar-productos/editar-productos.component';
14 import { ProductoService } from './shared/services/producto.service';
15 import { DetalleProductoComponent } from './detalle-producto/detalle-producto.component';
16 import { ListaComprasComponent } from './lista-compras/lista-compras.component';
17 import { LoginComponent } from './login/login.component';
18 import { NavbarComponent } from './shared/navbar/navbar.component';
19
20 // Declaraciones que permite especificar los componentes
21 @NgModule({
22   declarations: [
23     AppComponent,
24     ListaProductosComponent,
25     EditarProductosComponent,
26     DetalleProductoComponent,
27     ListaComprasComponent,
28     LoginComponent,
29     NavbarComponent,
30   ],
31   imports: [
32     BrowserModule,
33     AppRoutingModule,
34     HttpClientModule,
35     FormsModule
36   ],
37
38   //Especificar los servicios (conectar una funcionalidad hacia el exterior)
39   providers: [
40     ProductoService
41   ],
42   bootstrap: [AppComponent]
43 })
44 export class AppModule { }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.

Ln 9, Col 1 Spaces: 2 UTF-8 LF ⓘ TypeScript ⓘ Go Live ⓘ Prettier ⓘ 11:59 p.m. ESP 29/07/2023
```

## Especificamos los servicios



```
File Edit Selection View Go Run Terminal Help app.module.ts - FruberPM - Visual Studio Code

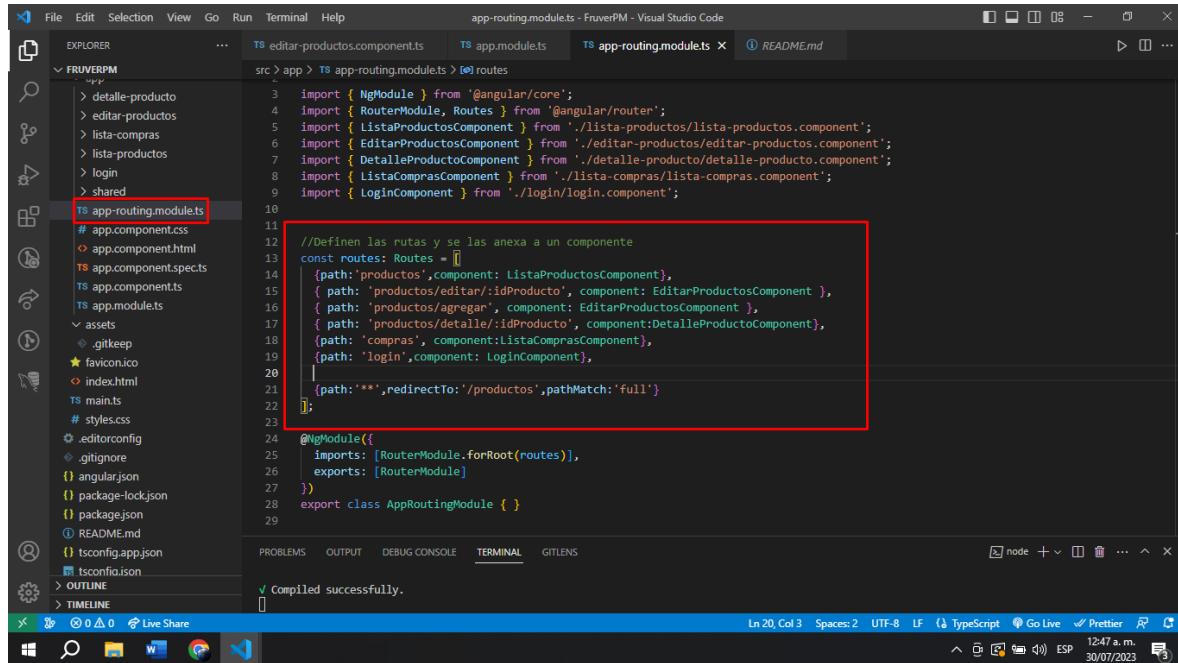
src > app > TS app.module.ts > ...
22 declarations: [
23   AppComponent,
24   ListaProductosComponent,
25   EditarProductosComponent,
26   DetalleProductoComponent,
27   ListaComprasComponent,
28   LoginComponent,
29   NavbarComponent
30 ],
31 imports: [
32   BrowserModule,
33   AppRoutingModule,
34   HttpClientModule,
35   FormsModule
36 ],
37
38 //Especificar los servicios (conectar una funcionalidad hacia el exterior)
39 providers: [
40   ProductoService
41 ],
42 bootstrap: [AppComponent]
43 )
44 export class AppModule { }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.

Ln 9, Col 1 Spaces: 2 UTF-8 LF ⓘ TypeScript ⓘ Go Live ⓘ Prettier ⓘ 11:59 p.m. ESP 29/07/2023
```

## 1.7 Configuración de las rutas en el archivo app-routing.module.ts

El archivo app-routing.module.ts se utiliza para configurar y gestionar las rutas de navegación en la aplicación. Es parte fundamental del enrutador de Angular y permite definir cómo se comportará la navegación entre diferentes componentes y vistas dentro de tu aplicación.



```
File Edit Selection View Go Run Terminal Help app-routing.module.ts - FruverPM - Visual Studio Code
EXPLORER TS editar-productos.component.ts TS app.modules TS app-routing.module.ts ⓘ README.md
src > app > TS app-routing.module.ts > [x] routes
src > app > TS app-routing.module.ts > [x] routes
3 import { NgModule } from '@angular/core';
4 import { RouterModule, Routes } from '@angular/router';
5 import { ListaProductosComponent } from './lista-productos/lista-productos.component';
6 import { EditarProductosComponent } from './editar-productos/editar-productos.component';
7 import { DetalleProductoComponent } from './detalle-producto/detalle-producto.component';
8 import { ListaComprasComponent } from './lista-compras/lista-compras.component';
9 import { LoginComponent } from './login/login.component';
10
11 //Definen las rutas y se las anexa a un componente
12 const routes: Routes = [
13   {path: 'productos', component: ListaProductosComponent},
14   {path: 'productos/editar/:idProducto', component: EditarProductosComponent},
15   {path: 'productos/agregar', component: EditarProductosComponent},
16   {path: 'productos/detalle/:idProducto', component:DetalleProductoComponent},
17   {path: 'compras', component:ListaComprasComponent},
18   {path: 'login',component: LoginComponent},
19   ]
20   [path: '**',redirectTo:'/productos',pathMatch:'full']
21 ];
22
23 @NgModule({
24   imports: [RouterModule.forRoot(routes)],
25   exports: [RouterModule]
26 })
27 }
28 export class AppRoutingModule { }
29 
```

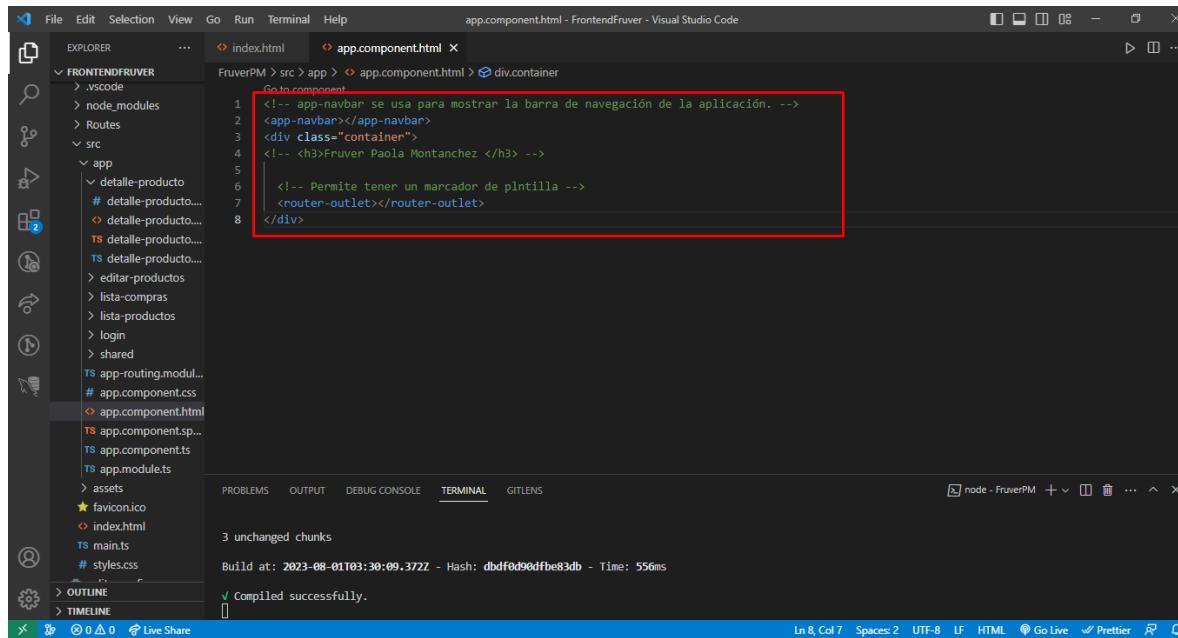
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

✓ Compiled successfully.

Ln 20, Col 3 Spaces: 2 UTF-8 LF TypeScript Go Live Prettier 12:47 a.m. 30/07/2023

## 1.8 Editamos el archivo app.component.html

La etiqueta <router-outlet> se utiliza en el contexto del enrutador de Angular para cargar y mostrar componentes específicos en la vista de una aplicación Angular. En otras palabras, es un marcador de posición en el que se insertarán los componentes asociados a las rutas definidas en la aplicación.



```
File Edit Selection View Go Run Terminal Help app.component.html - FrontendFruver - Visual Studio Code
EXPLORER index.html app.component.html
src > app > app.component.html > div.container
src > app > app.component.html > div.container
1 <!-- app-navbar se usa para mostrar la barra de navegación de la aplicación. -->
2 <app-navbar></app-navbar>
3 <div class="container">
4 <!-- <h3>Fruver Paola Montanchez </h3> -->
5
6 <!-- Permite tener un marcador de plantilla -->
7 <router-outlet></router-outlet>
8 </div>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

3 unchanged chunks

Build at: 2023-08-01T03:30:09.372Z - Hash: dbdf0d90dfbe83db - Time: 556ms

✓ Compiled successfully.

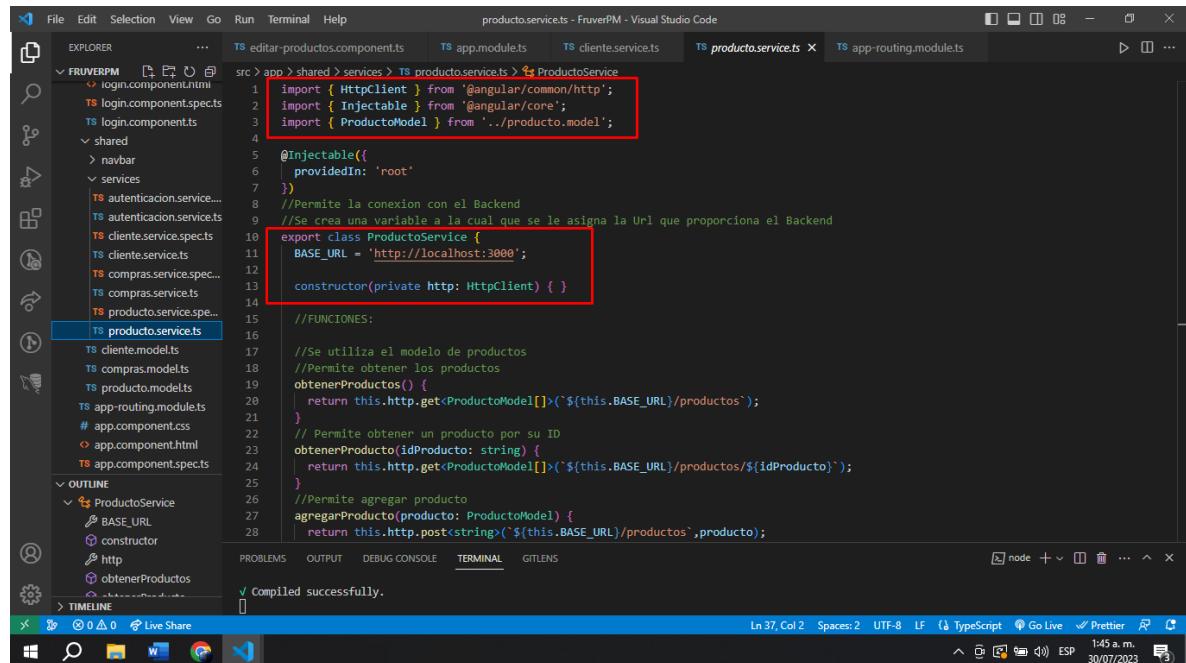
Ln 8, Col 7 Spaces: 2 UTF-8 LF HTML Go Live Prettier 12:47 a.m. 30/07/2023

## 2. Configuración de los servicios

Creamos una variable con el nombre BASE\_URL a la cual se le asigna la Url que proporciona el Backend y el puerto por el cual se presta el servicio, en este caso puerto 3000. Esto se realiza para todos los servicios.

### 2.1 Configuración del archivo producto.service.ts

producto: este servicio se encargará de gestionar los productos dentro de la aplicación. Como son las funciones de obtenerProducto, obtenerProductos, agregarProducto, actualizarProducto, borrarProducto.



The screenshot shows the Visual Studio Code interface with the file `producto.service.ts` open. The code defines a service named `ProductoService` that uses `HttpClient` to interact with a backend at `http://localhost:3000`. It includes methods for getting all products, getting a product by ID, adding a product, updating a product, and deleting a product.

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { ProductoModel } from '../producto.model';

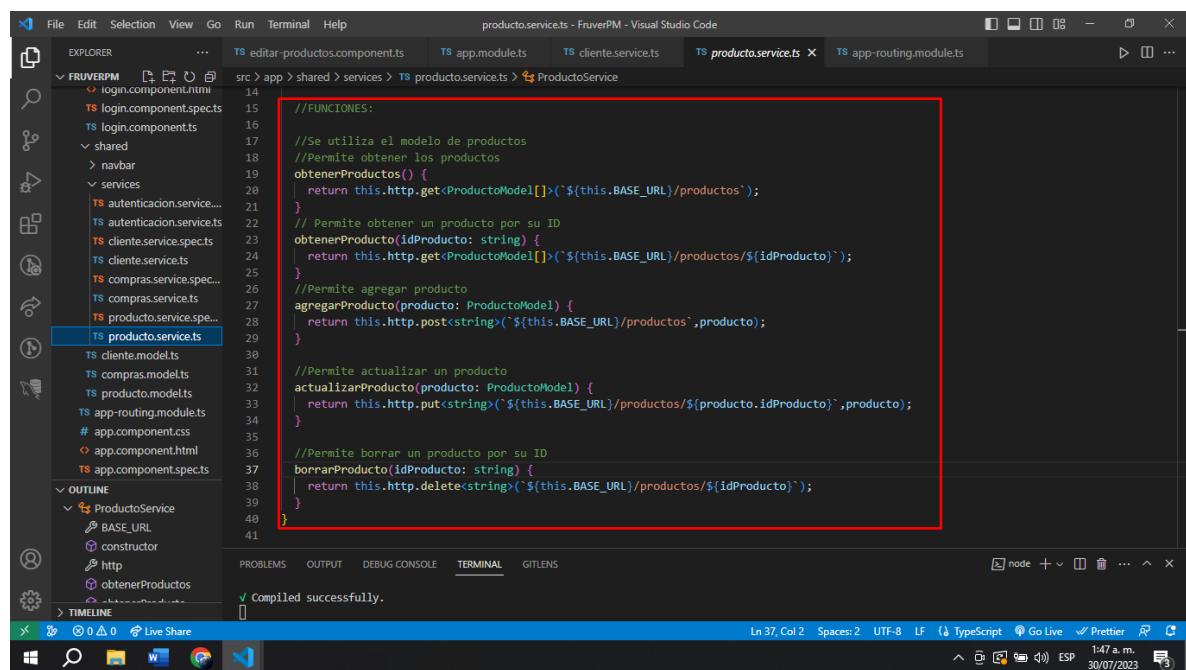
@Injectable({
  providedIn: 'root'
})
//Permite la conexión con el Backend
export class ProductoService {
  BASE_URL = 'http://localhost:3000';

  constructor(private http: HttpClient) { }

  //FUNCIONES:
  //Se utiliza el modelo de productos
  //Permite obtener los productos
  obtenerProductos() {
    return this.http.get<ProductoModel[]>(`${this.BASE_URL}/productos`);
  }
  // Permite obtener un producto por su ID
  obtenerProducto(idProducto: string) {
    return this.http.get<ProductoModel[]>(`${this.BASE_URL}/productos/${idProducto}`);
  }
  //Permite agregar producto
  agregarProducto(producto: ProductoModel) {
    return this.http.post<string>(`${this.BASE_URL}/productos`, producto);
  }
}

//Permite actualizar un producto
actualizarProducto(producto: ProductoModel) {
  return this.http.put<string>(`${this.BASE_URL}/productos/${producto.idProducto}`, producto);
}

//Permite borrar un producto por su ID
borrarProducto(idProducto: string) {
  return this.http.delete<string>(`${this.BASE_URL}/productos/${idProducto}`);
}
```



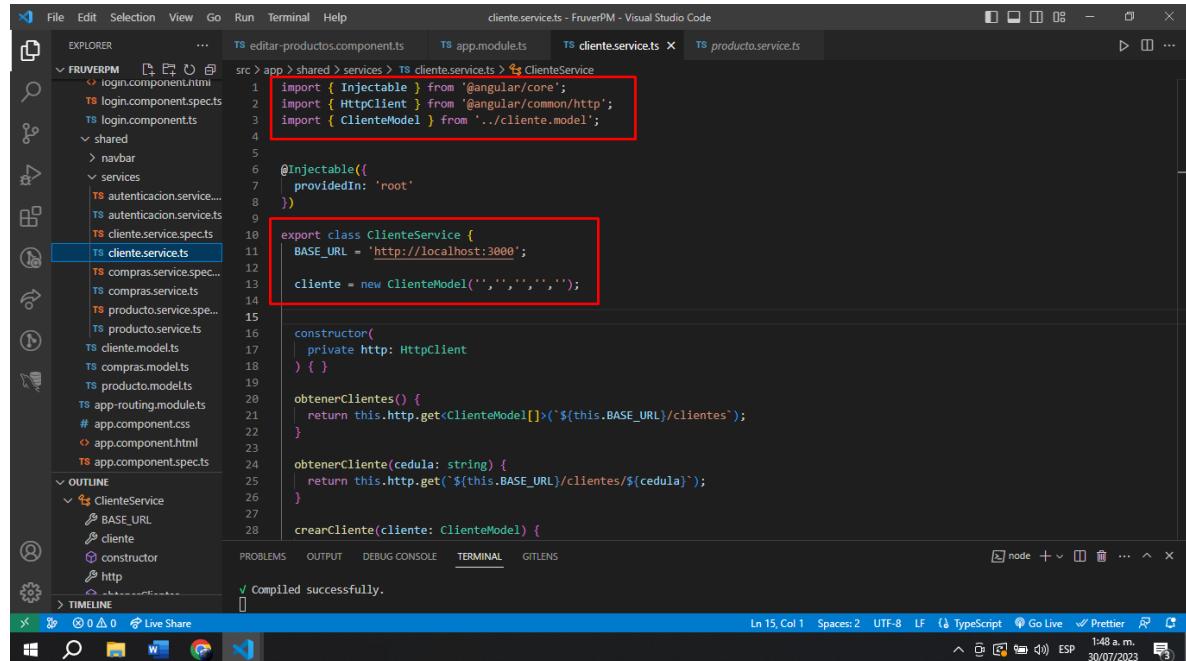
This screenshot shows the same `producto.service.ts` file with additional methods: `obtenerProductos()`, `obtenerProducto(idProducto: string)`, `agregarProducto(producto: ProductoModel)`, `actualizarProducto(producto: ProductoModel)`, and `borrarProducto(idProducto: string)`.

```
//FUNCIONES:
//Se utiliza el modelo de productos
//Permite obtener los productos
obtenerProductos() {
  return this.http.get<ProductoModel[]>(`${this.BASE_URL}/productos`);
}
// Permite obtener un producto por su ID
obtenerProducto(idProducto: string) {
  return this.http.get<ProductoModel[]>(`${this.BASE_URL}/productos/${idProducto}`);
}
//Permite agregar producto
agregarProducto(producto: ProductoModel) {
  return this.http.post<string>(`${this.BASE_URL}/productos`, producto);
}
//Permite actualizar un producto
actualizarProducto(producto: ProductoModel) {
  return this.http.put<string>(`${this.BASE_URL}/productos/${producto.idProducto}`, producto);
}

//Permite borrar un producto por su ID
borrarProducto(idProducto: string) {
  return this.http.delete<string>(`${this.BASE_URL}/productos/${idProducto}`);
}
```

## 2.2 Configuración del archivo cliente.service.ts

cliente: este servicio se encargará de gestionar los productos dentro de la aplicación. Como son las funciones de obtenerProducto, obtenerProductos, agregarProducto, actualizarProducto, borrarProducto.



The screenshot shows the Visual Studio Code interface with the file 'cliente.service.ts' open. The code defines a service named 'ClienteService' with a constructor that injects an 'HttpClient'. It includes methods for obtaining clients, obtaining a client by ID, creating a client, updating a client, and deleting a client. A red box highlights the constructor and the 'obtenerClientes' method.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { ClienteModel } from '../cliente.model';

@Injectable({
  providedIn: 'root'
})
export class ClienteService {
  BASE_URL = 'http://localhost:3000';

  cliente = new ClienteModel(' ', ' ', ' ', ' ', ' ');

  constructor(
    private http: HttpClient
  ) {}

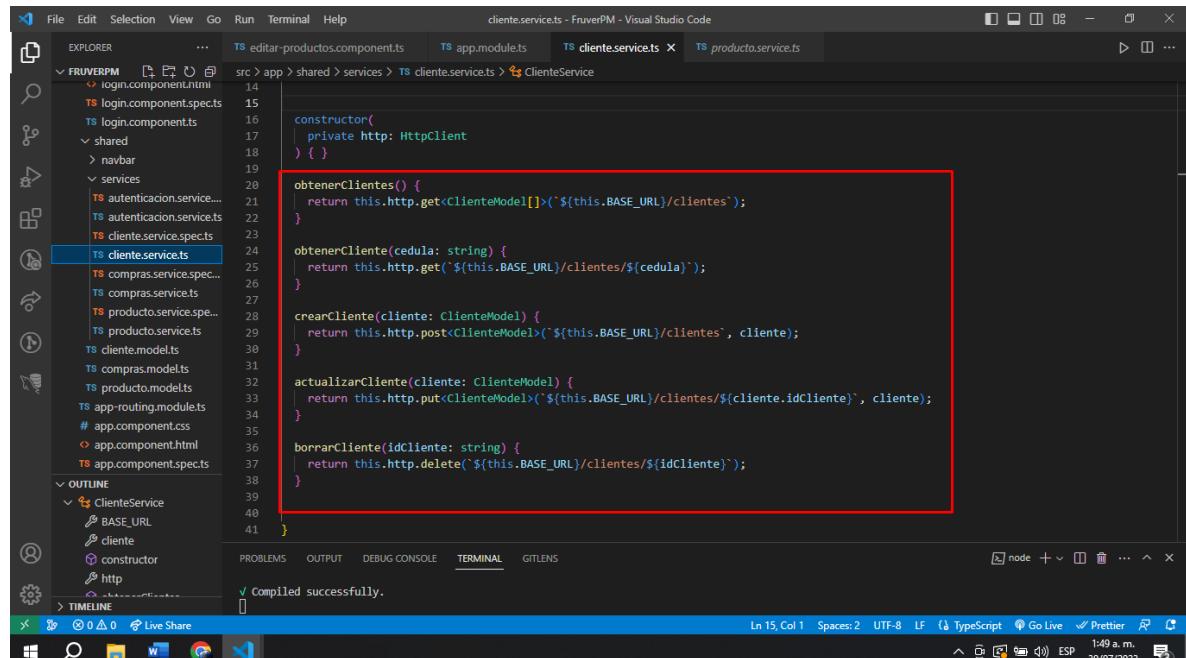
  obtenerClientes() {
    return this.http.get<ClienteModel[]>(`${this.BASE_URL}/clientes`);
  }

  obtenerCliente(cedula: string) {
    return this.http.get(`${this.BASE_URL}/clientes/${cedula}`);
  }

  crearCliente(cliente: ClienteModel) {
    return this.http.post<ClienteModel>(`${this.BASE_URL}/clientes`, cliente);
  }

  actualizarCliente(cliente: ClienteModel) {
    return this.http.put<ClienteModel>(`${this.BASE_URL}/clientes/${cliente.idCliente}`, cliente);
  }

  borrarCliente(idCliente: string) {
    return this.http.delete(`${this.BASE_URL}/clientes/${idCliente}`);
  }
}
```



The screenshot shows the same Visual Studio Code interface with the 'cliente.service.ts' file open. The code now includes additional methods: 'obtenerCientes()' and 'obtenerCliente(cedula: string)'. A red box highlights these two new methods.

```
constructor(
  private http: HttpClient
) {}

obtenerCientes() {
  return this.http.get<ClienteModel[]>(`${this.BASE_URL}/clientes`);
}

obtenerCliente(cedula: string) {
  return this.http.get(`${this.BASE_URL}/clientes/${cedula}`);
}
```

## 2.3 Configuracion del archivo compras.service.ts

compra: este servicio se encargará de gestionar las compras dentro de la aplicación. Como son las funciones de obtenerCompra, obtenerCompras, agregarProducto, actualizarProducto, borrarProducto.

The screenshot shows the Visual Studio Code interface with the file `compras.service.ts` open. The code defines a service named `ComprasService` with methods for getting purchases, creating purchases, updating purchases, and deleting purchases. A red box highlights the constructor and the `actualizarCompra` method.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { ComprasModel } from '../compras.model';

@Injectable({
  providedIn: 'root'
})
export class ComprasService {

  BASE_URL = "http://localhost:3000";

  constructor(private http: HttpClient) {}

  obtenerCompras() {
    return this.http.get<ComprasModel[]>(`${this.BASE_URL}/compras`);
  }

  obtenerCompra(idCompra: string) {
    return this.http.get(` ${this.BASE_URL}/compras/${idCompra}`);
  }

  crearCompra(compra: ComprasModel) {
    return this.http.post<ComprasModel>(`${this.BASE_URL}/compras`, compra);
  }

  actualizarCompra(compra: ComprasModel) {
    return this.http.put<ComprasModel>(` ${this.BASE_URL}/compras/${compra.idCompra}`,
      compra);
  }

  borrarCompra(idCompra: string) {
    return this.http.delete(` ${this.BASE_URL}/compras/${idCompra}`);
  }
}
```

The screenshot shows the Visual Studio Code interface with the file `compras.service.ts` open. The code defines a service named `ComprasService` with methods for getting purchases, creating purchases, updating purchases, and deleting purchases. A red box highlights the `actualizarCompra` method.

```
constructor(private http: HttpClient) {}

obtenerCompras() {
  return this.http.get<ComprasModel[]>(`${this.BASE_URL}/compras`);
}

obtenerCompra(idCompra: string) {
  return this.http.get(` ${this.BASE_URL}/compras/${idCompra}`);
}

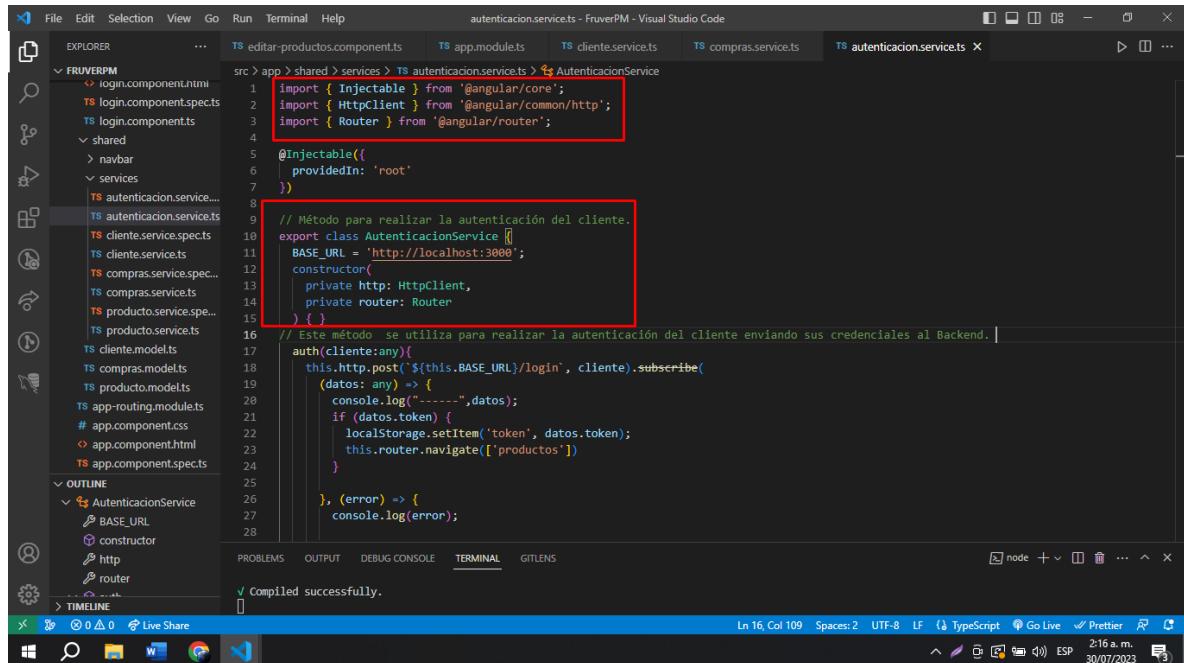
crearCompra(compra: ComprasModel) {
  return this.http.post<ComprasModel>(`${this.BASE_URL}/compras`, compra);
}

actualizarCompra(compra: ComprasModel) {
  return this.http.put<ComprasModel>(` ${this.BASE_URL}/compras/${compra.idCompra}`,
    compra);
}

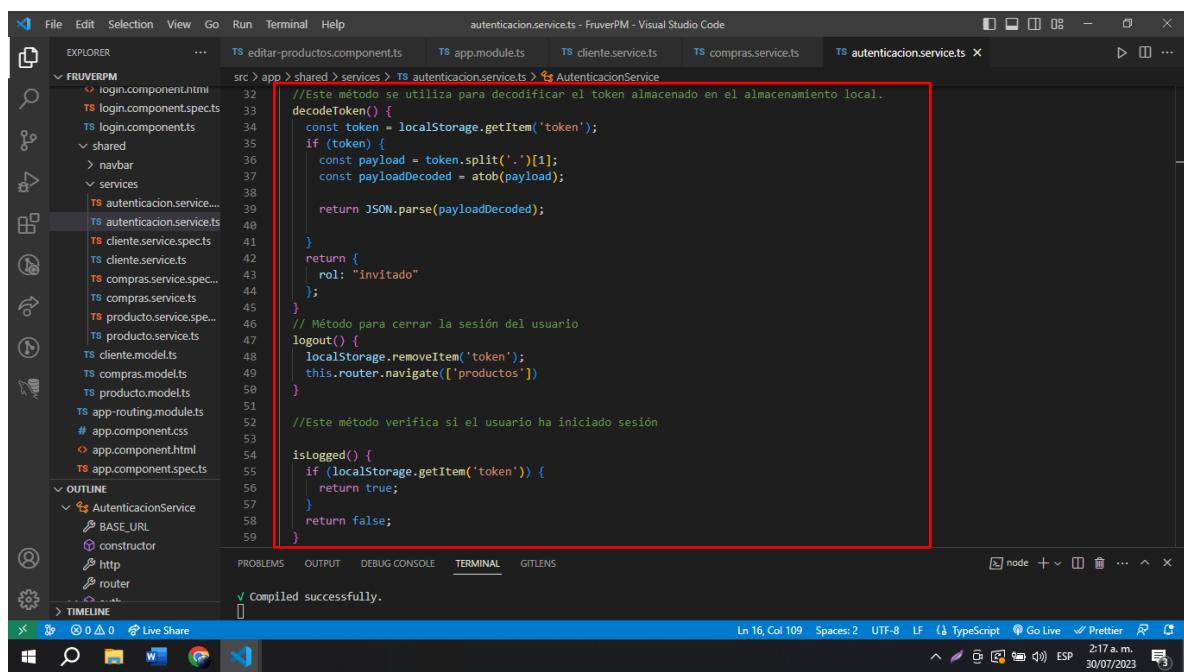
borrarCompra(idCompra: string) {
  return this.http.delete(` ${this.BASE_URL}/compras/${idCompra}`);
}
```

## 2.4 Configuración del archivo autenticacion.service.ts

Autenticación: este servicio se encargará de gestionar el proceso de autenticación y autorización de los usuarios en la aplicación.



```
src > app > shared > services > TS autenticacion.service.ts > AutenticacionService
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Router } from '@angular/router';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8
9 // Método para realizar la autenticación del cliente.
10 export class AutenticacionService {
11   BASE_URL = 'http://localhost:3000';
12   constructor(
13     private http: HttpClient,
14     private router: Router
15   ) {}
16
17   // Este método se utiliza para realizar la autenticación del cliente enviando sus credenciales al Backend.
18   auth(cliente:any){
19     this.http.post(`${this.BASE_URL}/login`, cliente).subscribe(
20       (datos: any) => {
21         console.log("-----",datos);
22         if (datos.token) {
23           localStorage.setItem('token', datos.token);
24           this.router.navigate(['productos']);
25         }
26       },
27       (error) => {
28         console.log(error);
29     }
30   )
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 
```



```
src > app > shared > services > TS autenticacion.service.ts > AutenticacionService
1 //Este método se utiliza para decodificar el token almacenado en el almacenamiento local.
2 decodeToken() {
3   const token = localStorage.getItem('token');
4   if (token) {
5     const payload = token.split('.')[1];
6     const payloadDecoded = atob(payload);
7
8     return JSON.parse(payloadDecoded);
9   }
10   return {
11     rol: "invitado"
12   };
13
14   // Método para cerrar la sesión del usuario
15   logout() {
16     localStorage.removeItem('token');
17     this.router.navigate(['productos']);
18   }
19
20   //Este método verifica si el usuario ha iniciado sesión
21   isLoggedIn() {
22     if (localStorage.getItem('token')) {
23       return true;
24     }
25     return false;
26   }
27
28
29
30
31
32
33
34
35
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
49
50
51
52
53
54
55
56
57
58
59 
```

### **3. Configuración de los componentes**

### 3.1 configuración del componente lista-producto

## lista-productos.component.ts

```
# detalle-producto.component.css
TS lista-compras.component.ts x TS detalle-producto.component.ts
TS detalle-producto.component.hi

src > app > lista-compras > TS lista-compras.component.ts ...
1 import { Component, OnInit } from '@angular/core';
2 import { ComprasModel } from './shared/compras.model';
3 import { ClienteModel } from './shared/cliente.model';
4 import { Observable, of } from 'rxjs';
5 import { ProductoModel } from './shared/producto.model';
6 import { ProductoService } from './shared/services/producto.service';
7 import { ComprasService } from './shared/services/compras.service';
8 import { ClienteService } from './shared/services/cliente.service';
9 import { Router } from '@angular/router';
10 import { AutenticacionService } from './shared/services/autenticacion.service';

@Directive({
  selector: 'app-lista-compras',
  templateUrl: './lista-compras.component.html',
  styleUrls: ['./lista-compras.component.css']
})
export class ListaComprasComponent implements OnInit {
  compras: Observable<ComprasModel[]> | undefined;
  productos: Observable<ProductoModel[]> | undefined;
  clientes: Observable<ClienteModel[]> | undefined;
  cliente = new ClienteModel("", "", ">", "", "");
  tmpCompras: Observable<ComprasModel[]> | undefined;
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.

Ln 1, Col 28 Spaces:2 UTF-8 LF ⓘ TypeScrip Go Live ⌂ Prettier
```

```
File Edit Selection View Go Run Terminal Help lista-productos.component.ts - FrverPM - Visual Studio Code

EXPLORER           ... ente.service.ts  TS compras.service.ts  TS autenticacion.service.ts  o lista-productos.component.html  TS lista-productos.components x > ...
src > app > lista-productos > lista-productos.component.ts  o ListaProductosComponent > ngOnInit
  12  @Component({
  13    selector: 'app-lista-productos',
  14    templateUrl: './lista-productos.component.html',
  15    styleUrls: ['./lista-productos.component.css']
  16  })
  17  export class ListaProductosComponent implements OnInit{
  18    productos: Observable<ProductoModel[]> | undefined;
  19
  20    cliente: ClienteModel = new ClienteModel("", "", "", "", "");
  21    compra: ComprasModel = new ComprasModel("", "", "", new Date(), "", "", "");
  22
  23
  24    constructor(
  25      private productoService: ProductoService,
  26      private comprasService: ComprasService,
  27      public auth: AutenticacionService
  28    ) { }
  29
  30    // Método que se ejecuta cuando el componente se inicializa
  31    ngOnInit(): void {
  32      |
  33      // obtener los productos
  34      if (this.auth.isLoggedIn()){
  35        let user = this.auth.decodeToken();
  36
  37        this.cliente = new ClienteModel(user.idCliente, user.nombres, user.correo, '', user.rol);
  38      } else {
  39        this.cliente.rol = 'invitado';
  40      }
  41    }
  42
  43    // ...
  44
  45    // ...
  46
  47    // ...
  48
  49    // ...
  50
  51    // ...
  52
  53    // ...
  54
  55    // ...
  56
  57    // ...
  58
  59    // ...
  60
  61    // ...
  62
  63    // ...
  64
  65    // ...
  66
  67    // ...
  68
  69    // ...
  70
  71    // ...
  72
  73    // ...
  74
  75    // ...
  76
  77    // ...
  78
  79    // ...
  80
  81    // ...
  82
  83    // ...
  84
  85    // ...
  86
  87    // ...
  88
  89    // ...
  90
  91    // ...
  92
  93    // ...
  94
  95    // ...
  96
  97    // ...
  98
  99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977
 1978
 1978
 1979
 1980
 1981
 1982
 1983
 1984
 1985
 1986
 1987
 1988
 1989
 1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 1999
 2000
 2001
 2002
 2003
 2004
 2005
 2006
 2007
 2008
 2009
 2009
 20
```

## lista-productos.component.html

The screenshot shows the Visual Studio Code interface with the file 'lista-productos.component.html' open. The code is an Angular component template for displaying a list of products. It includes an HTML structure with a table, CSS classes for styling columns, and Angular directives like \*ngFor and routerLink. The code is well-formatted with syntax highlighting.

```
<!-- Encabezados de la tabla -->
<thead>
  <tr class="row">
    <th class="col-sm-2"> ID </th>
    <th class="col-sm-4"> NOMBRE </th>
    <th class="col-sm-4"> DETALLE </th>
    <th class="col-sm-2"> ACCIONES </th>
  </tr>
</thead>
<tbody>
  <!-- Iteración sobre los productos utilizando *ngFor -->
  <tr *ngFor="let producto of productos | async" class="row">
    <td class="col-sm-2">{{ producto.idProducto }}</td>
    <td class="col-sm-4">{{ producto.nombre }}</td>
    <td class="col-sm-4">{{ producto.detalle }}</td>
    <td class="col-sm-1">
      <fieldset class="form-group col-sm-1">
        <a class="btn btn-info" [routerLink]="/productos/editar/{{ producto.idProducto }}">Editar</a>
      </fieldset>
    </td>
    <!-- Columna para "Borrar" el producto, con evento click que invoca la función borrarProducto(idProducto) -->
    <td class="col-sm-1">
      <fieldset class="form-group col-sm-1">
        <a class="btn btn-danger" (click)="borrarProducto(producto.idProducto)">Borrar</a>
      </fieldset>
    </td>
  </tr>
</tbody>
```

## lista-productos.components.css

The screenshot shows the Visual Studio Code interface with the file 'lista-productos.components.css' open. It contains CSS styles for various elements used in the component, such as .producto, .preciotxt, .txtFruta, .formulario input, and .formulario button. The styles include font sizes, weights, colors, and border properties. The code is annotated with comments explaining the purpose of each style block.

```
/* Estilos para la clase .producto */
.producto{
  width: 100%;
  height: 100%;
  object-fit: cover;
}

/* Estilos para la clase .preciotxt */
.preciotxt{
  font-size: 20px;
  font-weight: bold;
  color: #rgb(68, 68, 68);
}

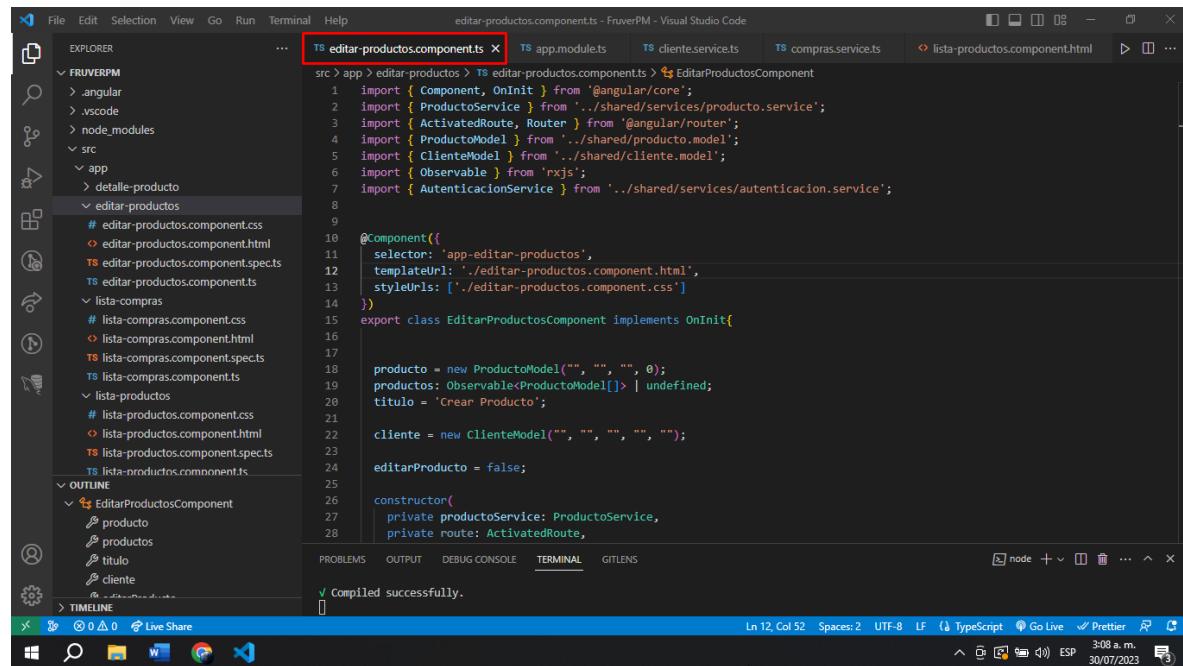
/* Estilos para la clase .txtFruta */
.txtFruta{
  font-size: 20px;
  font-weight: bold;
  color: #rgb(38, 87, 5);
}

/* Estilos para la clase .formulario input */
.formulario input{
  border: 1px solid #rgb(216, 113, 113);
}

/* Estilos para la clase .formulario button */
.formulario button{
```

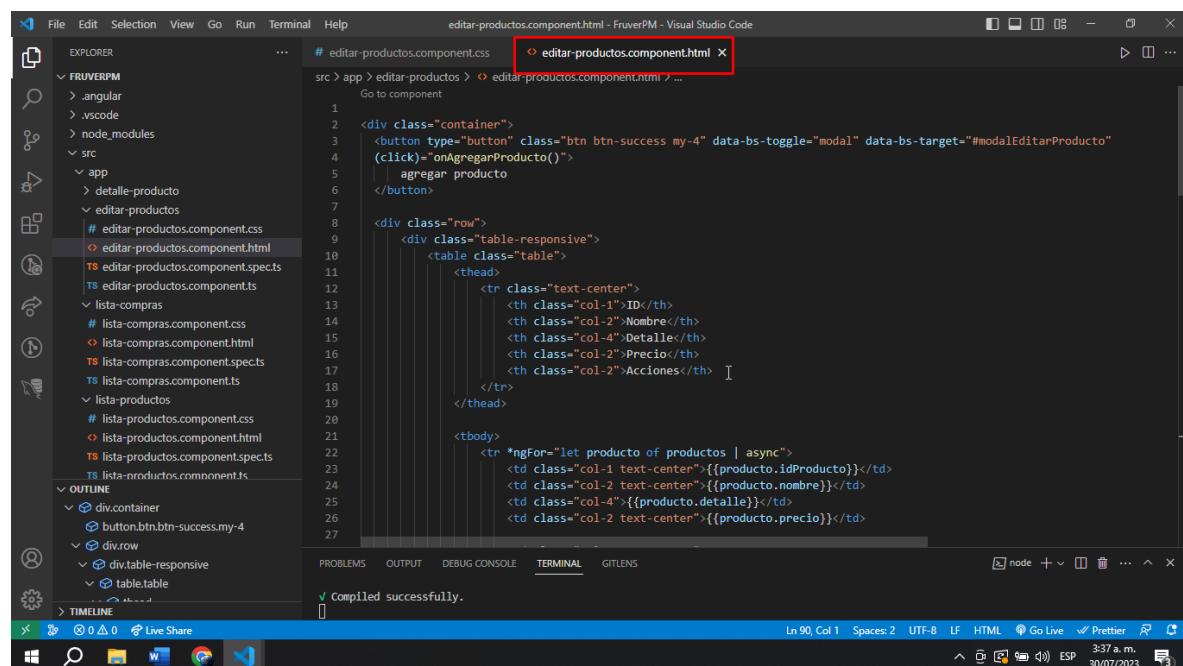
### 3.2 Configuramos el archivo editar-productos.components

#### editar-productos.component.ts



```
src > app > editar-productos > TS editar-productos.component.ts - FruverPM - Visual Studio Code
... TS editar-productos.component.ts TS app.module.ts TS cliente.service.ts TS compras.service.ts ... lista-productos.component.html D ... E ...
EXPLORER
FRUVERPM
> angular
> .vscode
> node_modules
src
> app
  > detalle-producto
  > editar-productos
    # editar-productos.component.css
    < editar-productos.component.html
    TS editar-productos.component.specs
    TS editar-productos.component.ts
    < lista-compras
      # lista-compras.component.css
      < lista-compras.component.html
      TS lista-compras.component.specs
      TS lista-compras.component.ts
      < lista-productos
        # lista-productos.component.css
        < lista-productos.component.html
        TS lista-productos.component.specs
        TS lista-productos.component.ts
    TS lista-productos.component.ts
  < OUTLINE
    < EditarProductosComponent
      producto
      productos
      titulo
      cliente
    < Timeline
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.
Ln 12, Col 52 Spaces: 2 UTF-8 LF TypeScript Go Live Prettier
node + ... 3:09 a.m. ESP 30/07/2023
```

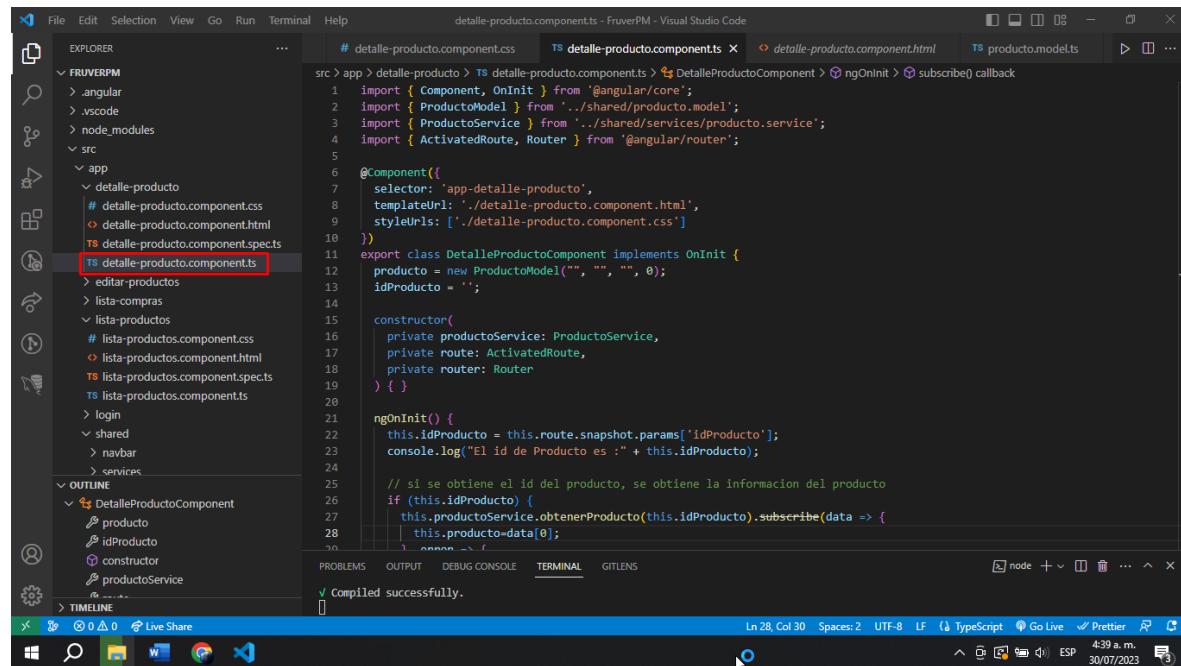
#### editar-productos.components.html



```
src > app > editar-productos > editar-productos.component.html - FruverPM - Visual Studio Code
... editar-productos.component.css < editar-productos.component.html ...
EXPLORER
FRUVERPM
> angular
> .vscode
> node_modules
src
> app
  > detalle-producto
  > editar-productos
    # editar-productos.component.css
    < editar-productos.component.html
    TS editar-productos.component.specs
    TS editar-productos.component.ts
    < lista-compras
      # lista-compras.component.css
      < lista-compras.component.html
      TS lista-compras.component.specs
      TS lista-compras.component.ts
    < lista-productos
      # lista-productos.component.css
      < lista-productos.component.html
      TS lista-productos.component.specs
      TS lista-productos.component.ts
    TS lista-productos.component.ts
  < OUTLINE
    < div.container
      < button.btn.btn-success.my-4 (click)="onAgregarProducto()" agregar producto
    < div.row
      < div.table-responsive
        < table.table
          < thead
            < tr class="text-center">
              < th>ID
              < th>Nombre
              < th>Detalle
              < th>Precio
              < th>Acciones
            </tr>
          </thead>
          <tbody>
            <tr *ngFor="let producto of productos | async">
              <td class="col-1 text-center">{{producto.idProducto}}
              <td class="col-2 text-center">{{producto.nombre}}
              <td class="col-4">{{producto.detalle}}
              <td class="col-2">{{producto.precio}}
              <td class="col-2 text-center">
                <button type="button" class="btn btn-success my-4" data-bs-toggle="modal" data-bs-target="#modalEditarProducto" (click)="onEditarProducto(producto.idProducto)" agregar producto>
                  Editar
                
                <button type="button" class="btn btn-danger my-4" (click)="onEliminarProducto(producto.idProducto)" eliminar producto>
                  Eliminar
                
              
            </tr>
          </tbody>
        </table>
      </div.table-responsive>
    </div.row>
  < Timeline
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.
Ln 90, Col 1 Spaces: 2 UTF-8 LF HTML TypeScript Go Live Prettier
node + ... 3:37 a.m. ESP 30/07/2023
```

### 3.3 Configuramos el archivo detalle-producto

#### detalle-producto.component.ts



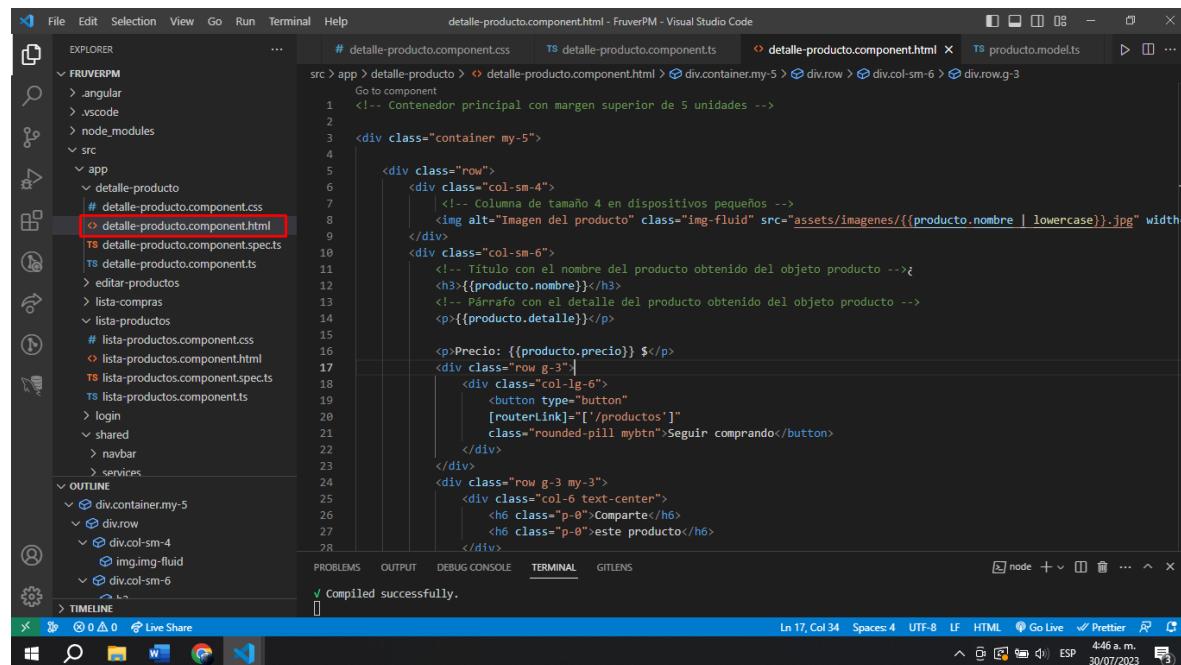
```
# detalle-producto.component.ts - FruverPM - Visual Studio Code
# detalle-producto.component.ts
detalle-producto.component.html
detalle-producto.component.ts
detalle-producto.component.specs
detalle-producto.component.ts

src > app > detalle-producto > TS detalle-producto.component.ts -> DetalleProductoComponent > @ngOnInit() callback
1 import { Component, OnInit } from '@angular/core';
2 import { ProdutoModel } from './shared/produto.model';
3 import { ProdutoService } from '../shared/services/produto.service';
4 import { ActivatedRoute, Router } from '@angular/router';
5
6 @Component({
7   selector: 'app-detalle-producto',
8   templateUrl: './detalle-producto.component.html',
9   styleUrls: ['./detalle-producto.component.css']
10 })
11 export class DetalleProductoComponent implements OnInit {
12   producto = new ProdutoModel("", "", "", 0);
13   idProduto = '';
14
15   constructor(
16     private produtoService: ProdutoService,
17     private route: ActivatedRoute,
18     private router: Router
19   ) {}
20
21   ngOnInit() {
22     this.idProduto = this.route.snapshot.params['idProduto'];
23     console.log("El id de Producto es :" + this.idProduto);
24
25     // si se obtiene el id del producto, se obtiene la informacion del producto
26     if (this.idProduto) {
27       this.produtoService.obtenerProducto(this.idProduto).subscribe(data => {
28         this.producto=data[0];
29       });
30     }
31   }
32 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.

Ln 28, Col 30  Spaces: 2  UTF-8  LF  TypeScript  Go Live  Prettier
node + 4:39 a.m. 30/07/2023
```

#### detalle-producto.component.html



```
# detalle-producto.component.html - FruverPM - Visual Studio Code
# detalle-producto.component.html
detalle-producto.component.ts
detalle-producto.component.html
detalle-producto.component.specs
detalle-producto.component.ts

src > app > detalle-producto > detalle-producto.component.html -> div.container.my-5 > div.row > div.col-sm-6 > div.row.g-3
Go to component
1 <!-- Contenedor principal con margen superior de 5 unidades -->
2
3 <div class="container my-5">
4
5   <div class="row">
6     <div class="col-sm-4">
7       <!-- Columna de tamaño 4 en dispositivos pequeños -->
8       
9     </div>
10    <div class="col-sm-6">
11      <!-- Título con el nombre del producto obtenido del objeto producto -->
12      <h3>{{producto.nombre}}</h3>
13      <!-- Parágrafo con el detalle del producto obtenido del objeto producto -->
14      <p>{{producto.detalle}}</p>
15
16      <p>Precio: {{producto.precio}} $</p>
17      <div class="row g-3">
18        <div class="col-lg-6">
19          <button type="button" [routerLink]="/productos" class="rounded-pill mybtn">Seguir comprando</button>
20        </div>
21      </div>
22      <div class="row g-3 my-3">
23        <div class="col-6 text-center">
24          <h6 class="p-0">Comparte</h6>
25          <h6 class="p-0">Este producto:</h6>
26        </div>
27      </div>
28    </div>
29  </div>
30 </div>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
✓ Compiled successfully.

Ln 17, Col 34  Spaces: 4  UTF-8  LF  HTML  Go Live  Prettier
node + 4:46 a.m. 30/07/2023
```

## detalle-producto.component.css

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "FRUVERPM". The file "detalle-producto.component.css" is selected and highlighted with a red box.
- Editor:** Displays the CSS code for "detalle-producto.component.css".

```
# detalle-producto.component.css
.mybtn{
  background-color: #02b11f;
  border: 0;
  padding: 10px;
  width: 100%;
  color: #fff;
```
- Terminal:** Shows the message "Compiled successfully." at the bottom.
- Bottom Status Bar:** Displays the date and time (30/07/2023, 4:47 a.m.) and other system information.

## 3.4 Configuramos el archivo detalle-producto

### lista-compras.components.ts

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "FRUVERPM". The file "lista-compras.components.ts" is selected and highlighted with a red box.
- Editor:** Displays the TypeScript code for "lista-compras.components.ts".

```
# detalle-producto.component.css
TS lista-compras.components.ts
TS detalle-producto.component.ts
TS detalle-producto.html
src > app > lista-compras > TS lista-compras.components.ts ...
```

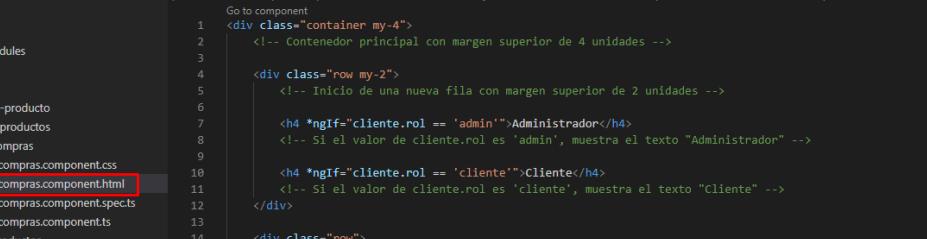
```
import { Component, OnInit } from '@angular/core';
import { ComprasModel } from './shared/compras.model';
import { ClienteModel } from './shared/cliente.model';
import { Observable, of } from 'rxjs';
import { ProductoModel } from './shared/producto.model';
import { ProductoService } from '../shared/services/producto.service';
import { ComprasService } from '../shared/services/compras.service';
import { ClienteService } from '../shared/services/cliente.service';
import { Router } from '@angular/router';
import { AutenticacionService } from '../shared/services/autenticacion.service';

@Component({
  selector: 'app-lista-compras',
  templateUrl: './lista-compras.component.html',
  styleUrls: ['./lista-compras.component.css']
})
export class ListaComprasComponent implements OnInit {

  compras: Observable<ComprasModel[]> | undefined;
  productos: Observable<ProductoModel[]> | undefined;
  clientes: Observable<ClienteModel[]> | undefined;
  cliente = new ClienteModel("", "", "", "", "", "");

  tmpCompras: Observable<ComprasModel[]> | undefined;
```
- Terminal:** Shows the message "Compiled successfully." at the bottom.
- Bottom Status Bar:** Displays the date and time (30/07/2023, 4:56 a.m.) and other system information.

## lista-compras.components.html



```
# detalle-producto.component.css    TS lista-compras.component.ts    lista-compras.component.html    TS detalle-producto.component.ts
File Edit Selection View Go Run Terminal Help lista-compras.component.html - FruruPM - Visual Studio Code
F RUVERPM ... # detalle-producto.component.css    TS lista-compras.component.ts    lista-compras.component.html    TS detalle-producto.component.ts
explorer > lista-compras.component.html > div.container.my-4 > div.row > div.table-responsive > table.table.table-striped.table-hover > thead > tr.text-center
    # Contenedor principal con margen superior de 4 unidades -->
    <div class="row my-2">
        <!-- Inicio de una nueva fila con margen superior de 2 unidades -->
        <!-- Si el valor de cliente.rol es 'admin', muestra el texto "Administrador" -->
        <h4 *ngIf="cliente.rol == 'admin'">Administrador</h4>
        <!-- Si el valor de cliente.rol es 'cliente', muestra el texto "Cliente" -->
        <h4 *ngIf="cliente.rol == 'cliente'">Cliente</h4>
    </div>

    <div class="row">
        <!-- Inicio de una nueva fila -->
        <div class="table-responsive">
            <!-- Div con clase "table-responsive" para hacer que la tabla sea receptiva en dispositivos pequeños -->
            <table class="table table-striped table-hover">
                <!-- Tabla con clases de Bootstrap para estilo -->
                <thead>
                    <!-- Encabezado de la tabla -->
                    <tr class="text-center">
                        <!-- Fila de encabezado con contenido centrado horizontalmente -->
                </thead>
                <tbody>
                    <tr>
                        <td>1</td>
                        <td>Nombre del Cliente</td>
                        <td>Apellido del Cliente</td>
                        <td>Número de Documento</td>
                        <td>Número de Telefono</td>
                        <td>Email</td>
                        <td>Opciones</td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>
```

## lista-compras.component.css

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** Shows the project structure under "FRUVERPM". The "lista-compras" folder is expanded, showing "lista-compras.component.css", "lista-compras.component.html", "lista-compras.component.spec.ts", "lista-compras.component.ts", and "lista-compras.module.ts".
- Editor:** Displays the contents of "lista-compras.component.css". The code defines a CSS rule for ".opcion button":

```
# detalle Producto.component.css  TS lista-compras.component.ts  # lista-compras.component.html
src > app > lista-compras > # lista-compras.component.css > .opcion button
1 .opcion button {
2   margin: 0 5px;
3   font-size: 10px;
4   font-weight: bold;
5 }
```
- Terminal:** Shows the output of a build process:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS
node +  □  ☰  ...  ^
✓ Compiled successfully.
⠄ Generating browser application bundles (phase: building)...
```
- Bottom Bar:** Includes icons for file operations, a "Live Share" button, and system status indicators.

### 3.5 login.components.

## login.component.ts

```
src > app > login > login.component.ts ...
1 import { Component } from '@angular/core';
2 import { AutenticacionService } from '../shared/services/autenticacion.service';
3
4 @Component({
5   selector: 'app-login',
6   templateUrl: './login.component.html',
7   styleUrls: ['./login.component.css']
8 })
9 export class LoginComponent {
10   credenciales = {
11     correo: '',
12     contraseña: ''
13   }
14
15   constructor(
16     private auth: AutenticacionService
17  ){}
18
19
20
21   login(){
22     console.log(this.credenciales);
23     this.auth.auth(this.credenciales)
24   }
25
26 }
27
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

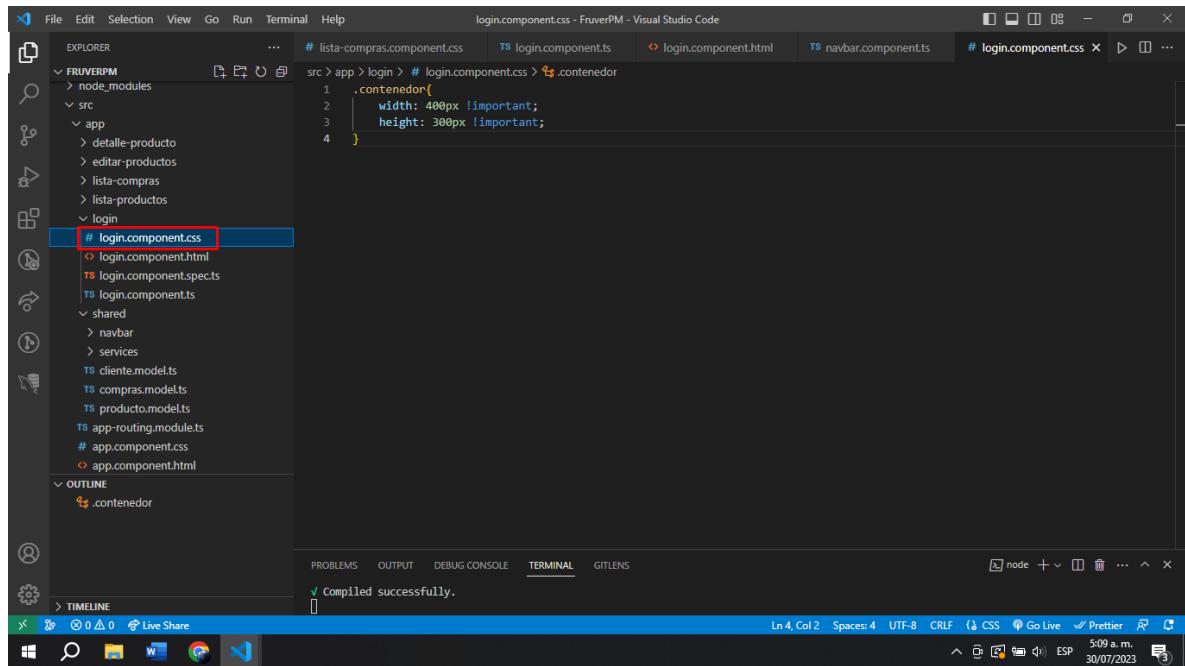
Compiled successfully.

## login.components.html

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure under "FRUVERPM". The file "login.component.html" is selected and highlighted with a red box.
- Code Editor:** The main area displays the content of "login.component.html". The code includes HTML and CSS, such as a container with a margin of 5 units and a central text area with an icon.
- Terminal:** At the bottom, the terminal shows the message "Compiled successfully."
- Status Bar:** The status bar at the bottom right shows the current file is "login.component.html", the line number is "Ln 56 Col 1", and the date is "30/07/2023".

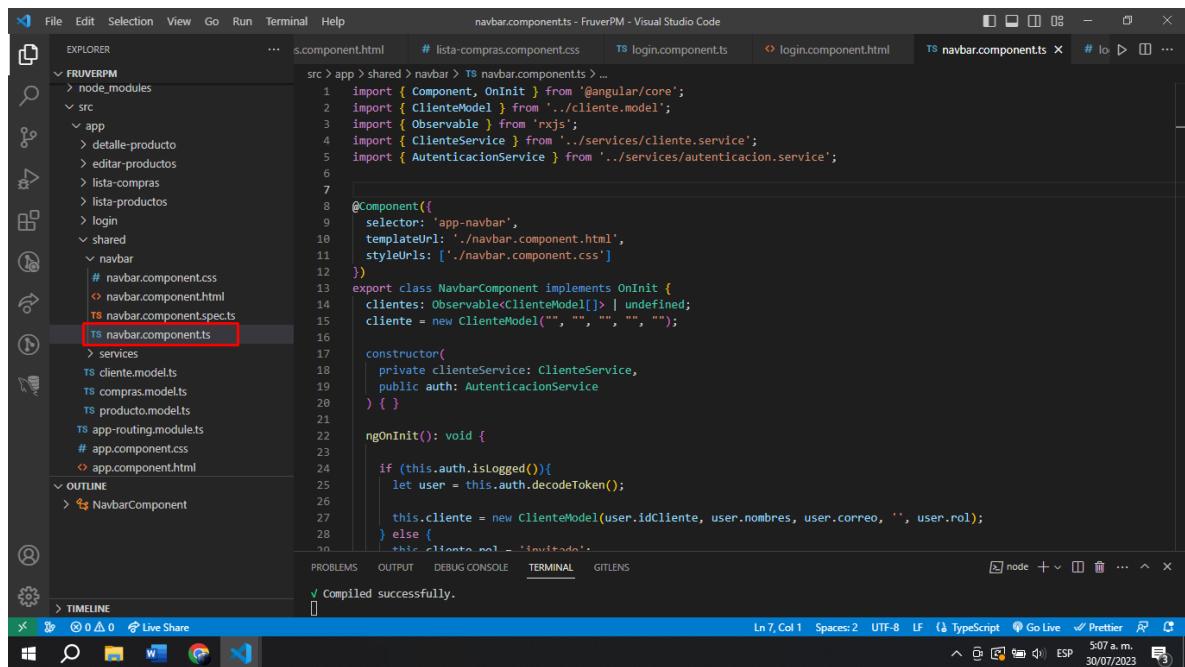
## login.components.css



```
# lista-compras.component.css    TS login.component.ts    login.component.html    TS navbar.component.ts    # login.component.css    D...    ...
src > app > login > # login.component.css > .contenedor
1 .contenedor{
2     width: 400px !important;
3     height: 300px !important;
4 }
```

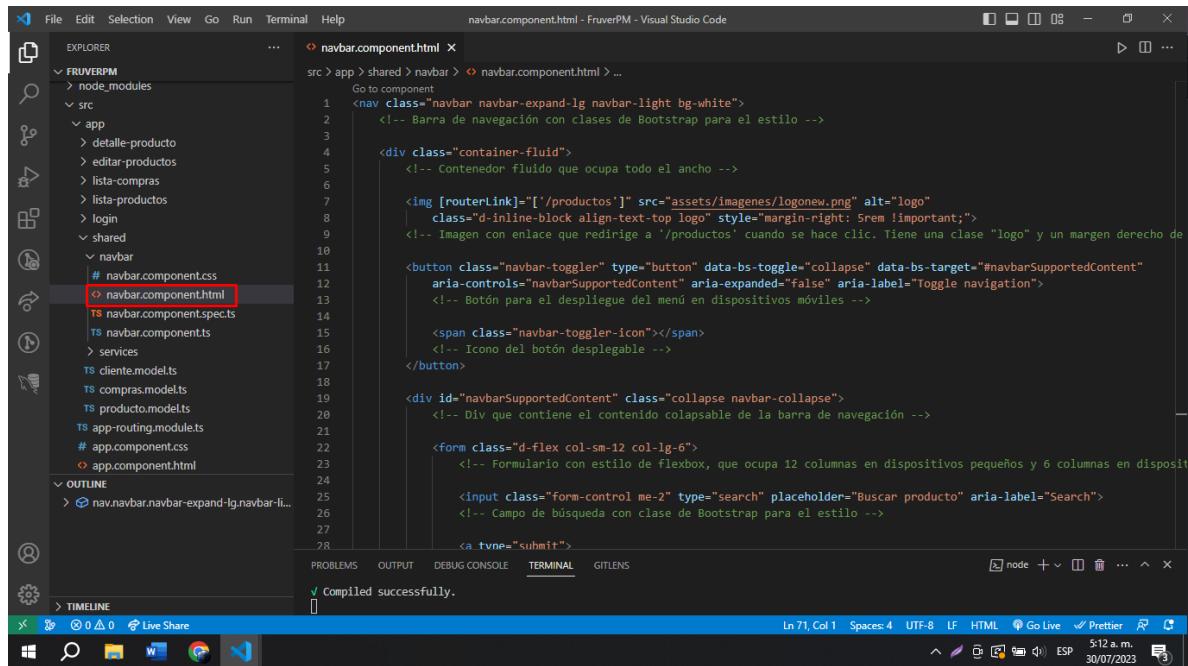
## 3.6 Componente navbar

### navbar.component.ts



```
s.component.html    # lista-compras.component.css    TS login.component.ts    login.component.html    TS navbar.component.ts    # navbar.component.ts    ...
src > app > shared > navbar > TS navbar.component.ts > ...
1 import { Component, OnInit } from '@angular/core';
2 import { ClienteModel } from '../cliente.model';
3 import { Observable } from 'rxjs';
4 import { ClienteService } from '../services/cliente.service';
5 import { AutenticacionService } from '../services/autenticacion.service';
6
7
8 @Component({
9   selector: 'app-navbar',
10  templateUrl: './navbar.component.html',
11  styleUrls: ['./navbar.component.css']
12 })
13 export class NavbarComponent implements OnInit {
14   clientes: Observable<ClienteModel[]> | undefined;
15   cliente = new ClienteModel("", "", "", "", "");
16
17   constructor(
18     private clienteService: ClienteService,
19     public auth: AutenticacionService
20   ) {}
21
22   ngOnInit(): void {
23
24     if (this.auth.isLoggedIn()){
25       let user = this.auth.decodeToken();
26
27       this.cliente = new ClienteModel(user.idCliente, user.nombres, user.correo, "", user.rol);
28     } else {
29       this.cliente = new ClienteModel("", "", "", "", "");
30     }
31   }
32 }
```

## navbar.component.html



The screenshot shows the Visual Studio Code interface with the file `navbar.component.html` open in the center editor pane. The code is an Angular component template. It includes Bootstrap classes like `navbar navbar-expand-lg navbar-light bg-white` for the navigation bar, and `container-fluid` for the main content area. It features a logo image, a toggle button for the menu, and a search form with a submit button. The sidebar on the left shows the project structure under the `FRUVERPM` folder, with `navbar.component.html` highlighted. The bottom status bar indicates the file is compiled successfully.

```
<nav class="navbar navbar-expand-lg navbar-light bg-white">
  <!-- Barra de navegación con clases de Bootstrap para el estilo -->
  <div class="container-fluid">
    <!-- Contenedor fluido que ocupa todo el ancho -->

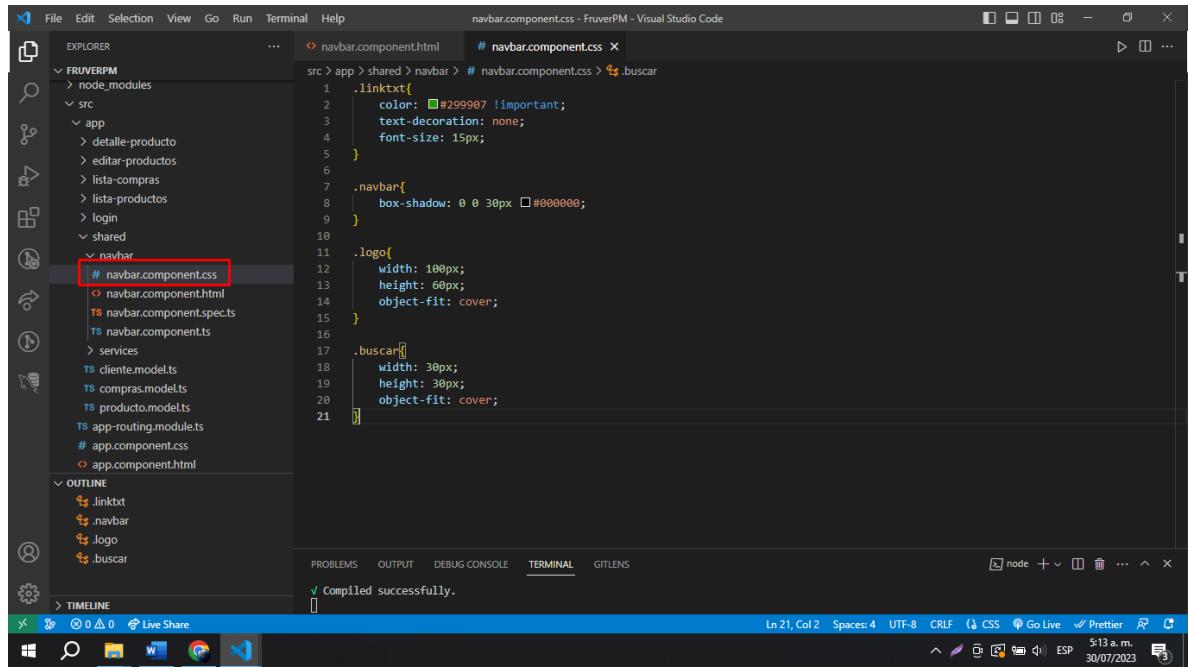
    
    <!-- Imagen con enlace que redirige a '/productos' cuando se hace clic. Tiene una clase "logo" y un margen derecho de 5rem -->

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
      <!-- Ícono del botón desplegable -->
    </button>

    <div id="navbarSupportedContent" class="collapse navbar-collapse">
      <!-- Div que contiene el contenido collapsible de la barra de navegación -->

      <form class="d-flex col-sm-12 col-lg-6">
        <!-- Formulario con estilo de flexbox, que ocupa 12 columnas en dispositivos pequeños y 6 columnas en dispositivos grandes -->
        <input class="form-control me-2" type="search" placeholder="Buscar producto" aria-label="Search">
        <!-- Campo de búsqueda con clase de Bootstrap para el estilo -->
        <button type="submit" class="btn btn-primary">Buscar</button>
      </form>
    </div>
  </div>
</nav>
```

## navbar component.css



The screenshot shows the Visual Studio Code interface with the file `navbar.component.css` open in the center editor pane. It contains CSS rules for the `.linktxt`, `.navbar`, `.logo`, and `.buscar` classes. The `.linktxt` class sets a color of #299907 and a font size of 15px. The `.navbar` class adds a box shadow. The `.logo` class sets dimensions and object-fit for the logo image. The `.buscar` class sets dimensions and object-fit for the search input. The sidebar on the left shows the project structure under the `FRUVERPM` folder, with `navbar.component.css` highlighted. The bottom status bar indicates the file is compiled successfully.

```
.linktxt{
  color: #299907 !important;
  text-decoration: none;
  font-size: 15px;
}

.navbar{
  box-shadow: 0 0 30px #000000;
}

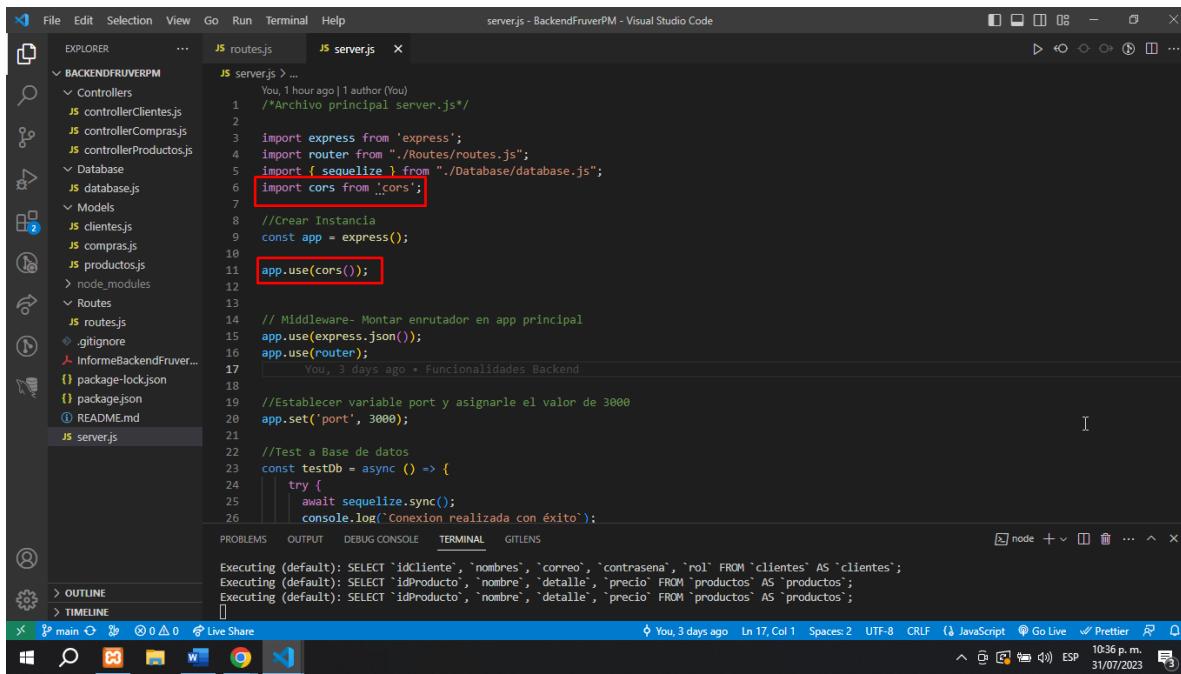
.logo{
  width: 100px;
  height: 60px;
  object-fit: cover;
}

.buscar{
  width: 30px;
  height: 30px;
  object-fit: cover;
}
```

## 4. Conexión con el Backend

Es importante resaltar que se debe importar la biblioteca **CORS** en el Backend. Cors (Cross-Origin Resource Sharing) es una herramienta que permite a los servidores web configurar y gestionar el acceso a recursos desde diferentes orígenes (dominios, protocolos o puertos) en el navegador.

Con el middleware cors configurado, el backend podrá recibir solicitudes del Frontend sin restricciones de seguridad CORS y se podrá establecer una conexión adecuada entre ambas partes de la aplicación.



```

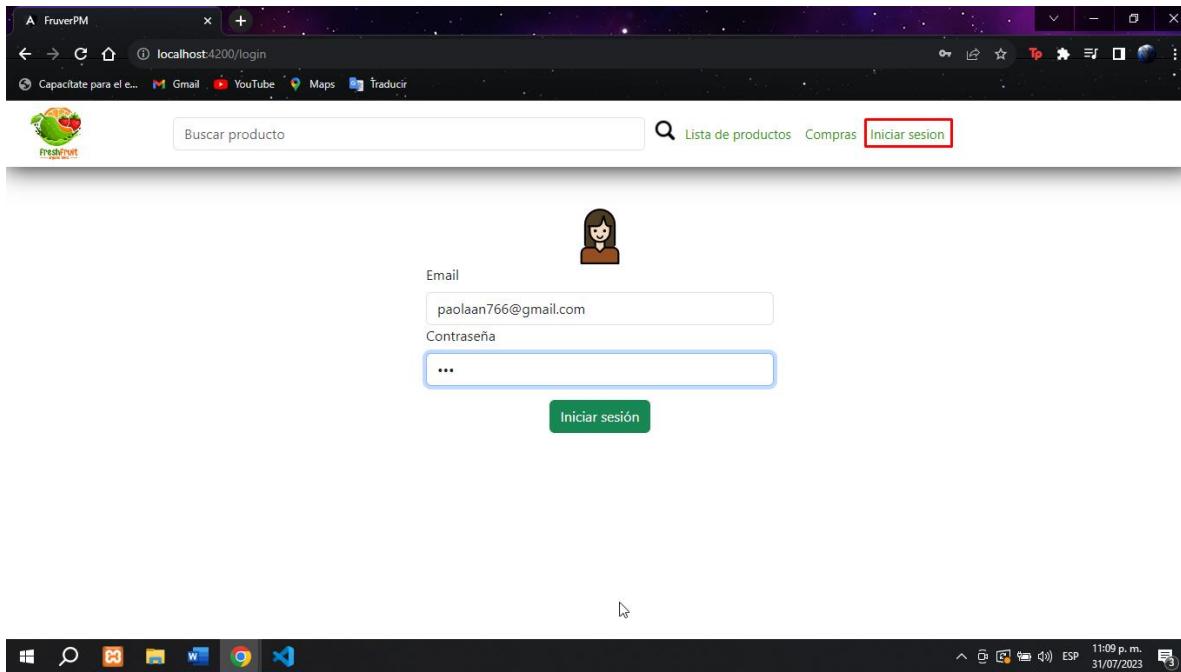
File Edit Selection View Go Run Terminal Help
server.js - BackendFruverPM - Visual Studio Code
EXPLORER JS routes.js JS server.js
BACKENDFRUVERPM
  Controllers
    controllerCuentas.js
    controllerCompras.js
    controllerProductos.js
  Database
    database.js
  Models
    clientes.js
    compras.js
    productos.js
  node_modules
  Routes
    routes.js
    .gitignore
  InformeBackendFruver...
  package-lock.json
  package.json
  README.md
  server.js

You, 1 hour ago | 1 author (You)
1  /*Archivo principal server.js*/
2
3  import express from 'express';
4  import router from './Routes/routes.js';
5  import { sequelize } from './Database/database.js';
6  import cors from 'cors';
7
8  //Crear Instancia
9  const app = express();
10
11  app.use(cors());
12
13
14  // Middleware- Montar enrutador en app principal
15  app.use(express.json());
16  app.use(router);
17
18  You, 3 days ago • Funcionalidades Backend
19
20  //Establecer variable port y asignarle el valor de 3000
21  app.set('port', 3000);
22
23  //Test a Base de datos
24  const testDb = async () => {
25    try {
26      await sequelize.sync();
27      console.log('Conexion realizada con éxito');
28
29      Executing (default): SELECT `idCliente`, `nombres`, `correo`, `contraseña`, `rol` FROM `clientes` AS `clientes`;
30      Executing (default): SELECT `idProducto`, `nombre`, `detalle`, `precio` FROM `productos` AS `productos`;
31      Executing (default): SELECT `idProducto`, `nombre`, `detalle`, `precio` FROM `productos` AS `productos`;
32
33    }
34  }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
99

```

## 5. Producto Final FRUVERMANIA

### Menú de inicio de sesión



## Menú de inicio

A FruverPM x + localhost:4200/productos Capacitate para el e... Gmail YouTube Maps Traducir Buscar producto Lista de productos Compras Iniciar sesión

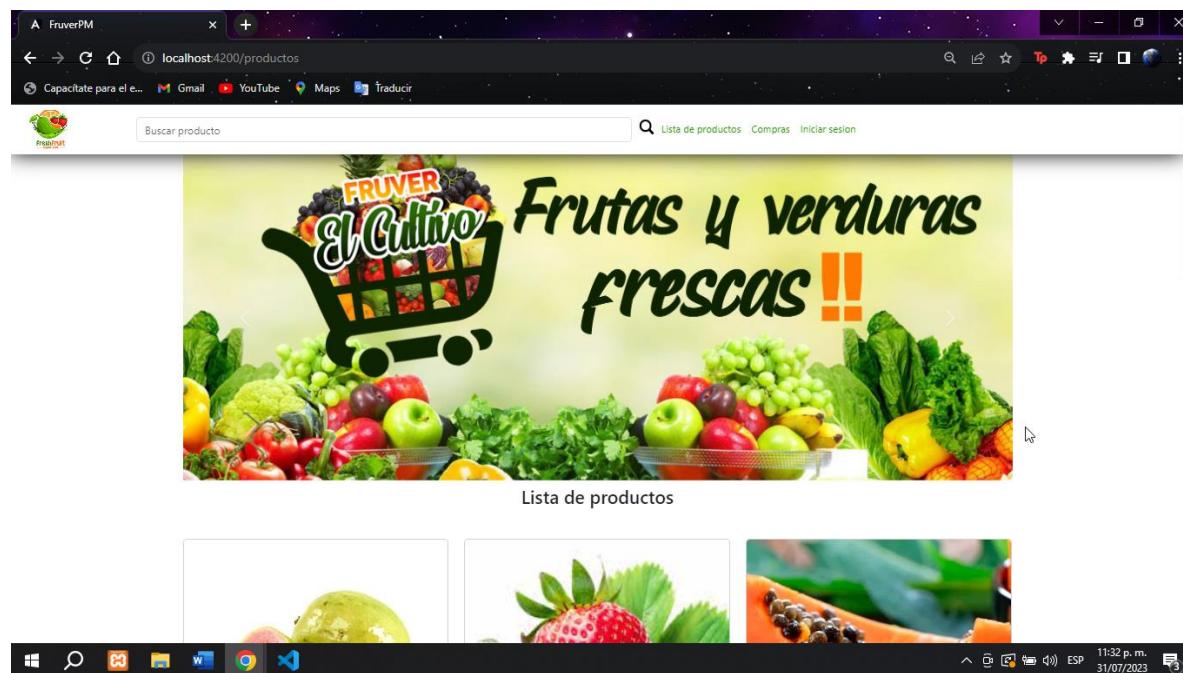
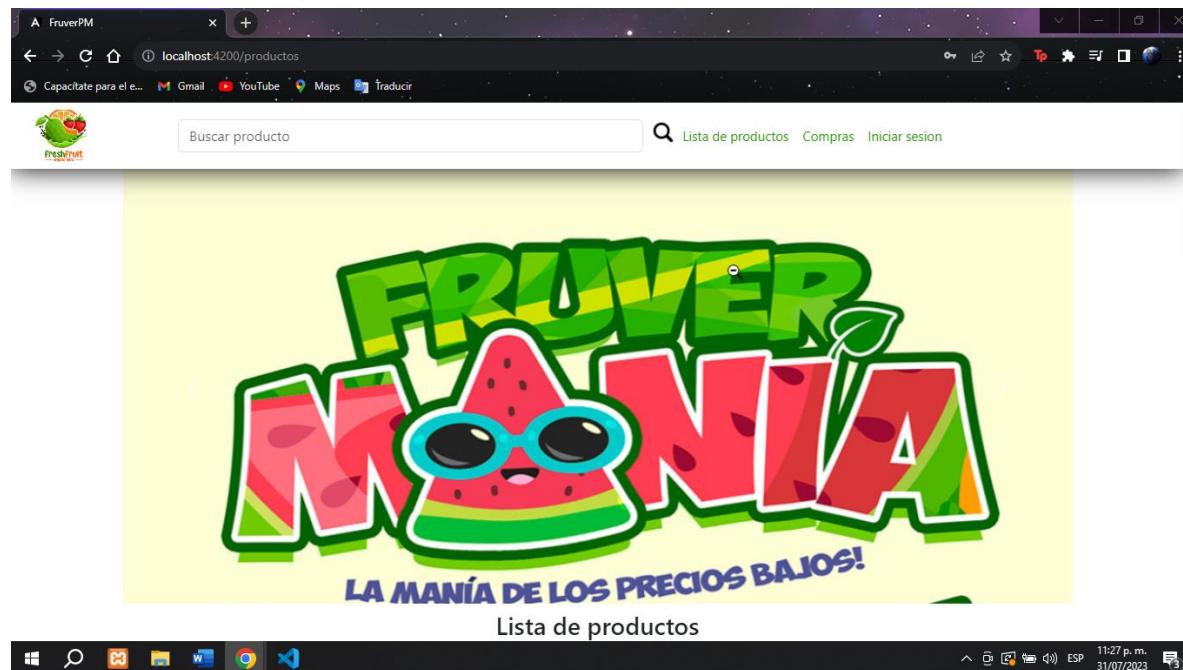
# FRUVER MANIA

LA MANÍA DE LOS PRECIOS BAJOS!

**Lista de productos**

<b>Guayaba</b> Por sólo: \$1300 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Fresas</b> Por sólo: \$3500 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Papaya</b> Por sólo: \$4000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>
<b>Manzanas</b> Por sólo: \$5000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Mangos</b> Por sólo: \$3000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Uchuva</b> Por sólo: \$3000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>
<b>Cebolla</b> Por sólo: \$1000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Papa</b> Por sólo: \$7000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>	<b>Lechuga</b> Por sólo: \$2000 <input type="button" value="1"/> <input type="button" value="Agregar al carrito"/> <a href="#">Ver más</a>

## Carousel



## Menú de lista de productos

A FruverPM x +

localhost:4200/productos

Capacitate para el e... Gmail YouTube Maps Traducir

### Lista de productos



**Guayaba**  
Por sólo: \$1300

[Ver más](#)



**Fresas**  
Por sólo: \$3500

[Ver más](#)



**Papaya**  
Por sólo: \$4000

[Ver más](#)

12:38 a. m. ESP 1/08/2023

A FruverPM x +

localhost:4200/productos

Capacitate para el e... Gmail YouTube Maps Traducir



**Manzanas**  
Por sólo: \$5000

[Ver más](#)



**Mangos**  
Por sólo: \$3000

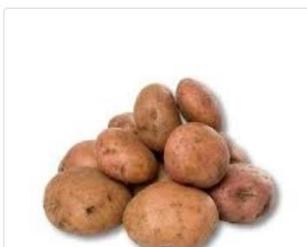
[Ver más](#)



**Uchuva**  
Por sólo: \$3000

[Ver más](#)

12:39 a. m. ESP 1/08/2023

		
<p><b>Cebolla</b> Por sólo: \$1000</p> <p><input type="button" value="1"/> <input type="button" value="Añadir al carrito"/></p> <p><a href="#">Ver más</a></p>	<p><b>Papa</b> Por sólo: \$7000</p> <p><input type="button" value="1"/> <input type="button" value="Añadir al carrito"/></p> <p><a href="#">Ver más</a></p>	<p><b>Lechuga</b> Por sólo: \$2000</p> <p><input type="button" value="1"/> <input type="button" value="Añadir al carrito"/></p> <p><a href="#">Ver más</a></p>

## Menú detalle de producto

A screenshot of a web browser showing a product detail page for mangoes. The page has a header with a logo for 'FruterPM', a search bar, and navigation links for 'Lista de productos', 'Agregar producto', 'Compras', and 'Iniciar sesion'. The main content features a large image of several ripe mangoes in shades of red, orange, and yellow. Below the image is the title 'Mangos' and a detailed description: 'Los mangos son frutas tropicales que pertenecen a la familia Anacardiaceae y al género Mangifera. Son ampliamente conocidos por su sabor dulce, jugoso y delicioso, lo que los convierte en una de las frutas más populares en todo el mundo. Además de su delicioso sabor, los mangos también ofrecen numerosos beneficios para la salud. Son una excelente fuente de vitaminas y minerales, como vitamina C, vitamina A, vitamina E, vitamina K y potasio.' A price of 'Precio: 3000 \$' is listed, followed by a green button labeled 'Seguir comprando'. Below the button is a 'Comparte este producto' section with social media sharing icons for Facebook, Email, and Twitter. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

A FruverPM x +

localhost:4200/productos/detalle/15

Capacitate para el e... Gmail YouTube Maps Traducir

Buscar producto

Lista de productos Agregar producto Compras Iniciar sesión



## Uchuva

La uchuva es una fuente rica de vitamina C, vitamina A, vitamina B y antioxidantes. También contiene minerales como el fósforo y el hierro. Gracias a su perfil nutricional, la uchuva es considerada una fruta saludable y nutritiva, beneficiosa para el sistema inmunológico y la salud ocular, entre otros aspectos.

Precio: 3000 \$

Seguir comprando

Comparte este producto



Windows 12:47 a.m. 1/08/2023

A FruverPM x +

localhost:4200/productos/detalle/16

Capacitate para el e... Gmail YouTube Maps Traducir

Buscar producto

Lista de productos Compras Iniciar sesión



## Cebolla

La cebolla es uno de los ingredientes más versátiles en la cocina y se utiliza ampliamente en la preparación de numerosos platos. Se puede consumir cruda, cortada en rodajas finas para ensaladas o salsas, o cocida en diferentes formas, como salteada, caramelizada, asada o hervida. Además de ser un ingrediente esencial en muchas recetas, la cebolla se utiliza para dar sabor y realzar el sabor de otros ingredientes en una amplia variedad de platos.

Precio: 1000 \$

Seguir comprando

Comparte este producto



Windows 12:51 a.m. 1/08/2023

A FruverPM x +

localhost:4200/

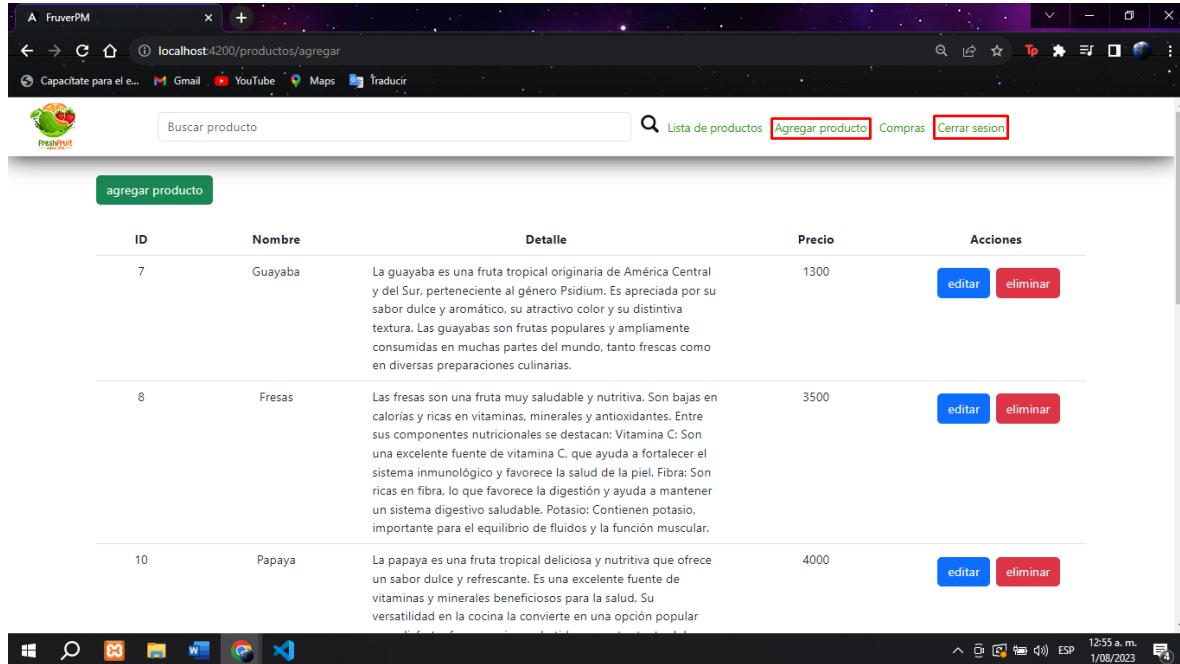
Capacitate para el e... Gmail YouTube Maps Traducir

Buscar producto

Lista de productos Compras Iniciar sesión

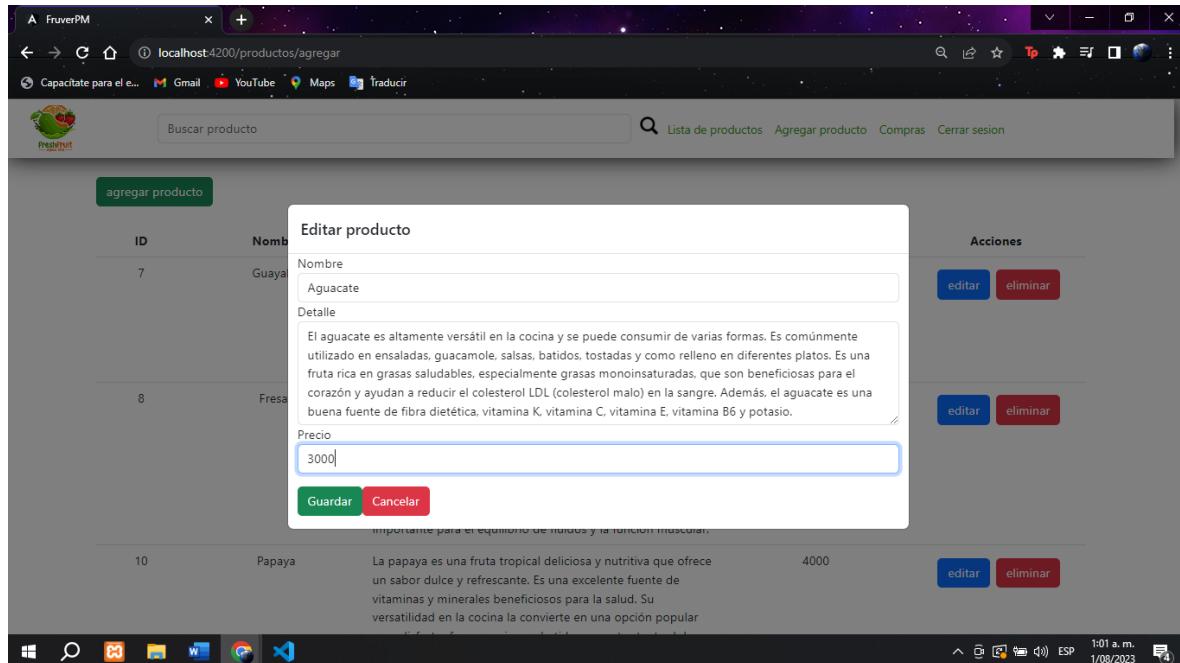
## Menú de agregar producto

Para poder agregar, modificar o eliminar un producto debe ser un usuario administrador, de lo contrario no está habilitada esta opción.



The screenshot shows a web application interface for managing products. At the top, there's a navigation bar with links for 'Lista de productos', 'Agregar producto' (which is highlighted with a red box), 'Compras', and 'Cerrar sesión'. Below the navigation is a search bar labeled 'Buscar producto' and a logo for 'freshnut'. The main content area displays a table of products with columns for 'ID', 'Nombre', 'Detalle', 'Precio', and 'Acciones'. Each row contains a product entry: 'Guayaba' (ID 7), 'Fresas' (ID 8), and 'Papaya' (ID 10). Each row has 'editar' and 'eliminar' buttons under 'Acciones'. A green 'agregar producto' button is located at the top left of the table area. The status bar at the bottom right shows the date and time: '12:55 a.m. 1/08/2023'.

Como ejemplo, procedemos a agregar el producto Aguacate



This screenshot shows the same web application after an item has been selected for editing. A modal dialog box titled 'Editar producto' is open over the main content. It contains fields for 'Nombre' (Aguacate) and 'Detalle' (with a detailed description of the avocado's benefits). Below these is a 'Precio' field containing '3000'. At the bottom of the dialog are 'Guardar' and 'Cancelar' buttons. The background table remains visible, showing the other products. The status bar at the bottom right shows the date and time: '1:01 a.m. 1/08/2023'.

<p>Todas las formas para ensaladas o salsas, o cocida en diferentes formas, como salteada, caramelizada, asada o hervida. Además de ser un ingrediente esencial en muchas recetas, la cebolla se utiliza para dar sabor y realzar el sabor de otros ingredientes en una amplia variedad de platos.</p>			
17	Papa	Las papas son extremadamente versátiles en la cocina y se pueden preparar de diversas formas. Se pueden cocinar hervidas, asadas, fritas, al vapor o incluso en puré. También se utilizan para preparar platos populares como papas fritas, papas al horno, ensalada de papa, puré de papas y muchas otras recetas.	7000
18	Lechuga	Esta hortaliza es conocida por su contenido en agua, lo que la hace hidratante y de bajo contenido calórico, lo que la convierte en una excelente opción para dietas saludables. Además, la lechuga es una buena fuente de vitaminas y minerales, incluyendo vitamina A, vitamina K, ácido fólico y potasio.	2000
19	Aguacate	El aguacate es altamente versátil en la cocina y se puede consumir de varias formas. Es comúnmente utilizado en ensaladas, guacamole, salsas, batidos, tostadas y como relleno en diferentes platos. Es una fruta rica en grasas saludables, especialmente grasas monoinsaturadas, que son beneficiosas para el corazón y ayudan a reducir el colesterol LDL (colesterol malo) en la sangre. Además, el aguacate es una buena fuente de fibra dietética, vitamina K, vitamina C, vitamina E, vitamina B6 y potasio.	3000

A FruverPM

localhost:4200/productos/detalle/19

Capacitate para el e... Gmail YouTube Maps Traducir

Buscar producto



**Aguacate**

El aguacate es altamente versátil en la cocina y se puede consumir de varias formas. Es comúnmente utilizado en ensaladas, guacamole, salsas, batidos, tostadas y como relleno en diferentes platos. Es una fruta rica en grasas saludables, especialmente grasas monoinsaturadas, que son beneficiosas para el corazón y ayudan a reducir el colesterol LDL (colesterol malo) en la sangre. Además, el aguacate es una buena fuente de fibra dietética, vitamina K, vitamina C, vitamina E, vitamina B6 y potasio.

Precio: 3000 \$

[Seguir comprando](#)

Comparte este producto



## Menú de compras Administrador

A screenshot of a web browser window titled "FruverPM" showing the "Administrador" section. The URL is "localhost:4200/compras". The page displays a table of purchases with columns: ID, Nombre, Producto, Cantidad, Fecha, Estado, and Acciones. The purchases listed are:

ID	Nombre	Producto	Cantidad	Fecha	Estado	Acciones
15	paola	Fresas	2	25/07/2023	Rechazado	
16	paola	Manzanas	2	25/07/2023	Aceptado	
18	paola	Papaya	7	26/07/2023	Aceptado	
25	Rosita	Manzanas	10	29/07/2023	Aceptado	
26	ingrid	Uchuva	8	29/07/2023	Aceptado	
30	Rosita	Cebolla	5	29/07/2023	Rechazado	
38	ingrid	Cebolla	2	01/08/2023	Pendiente	<button>Aceptar</button> <button>X</button>
40	Rosita	Guayaba	4	01/08/2023	Pendiente	<button>Aceptar</button> <button>X</button>

## Menú compras cliente

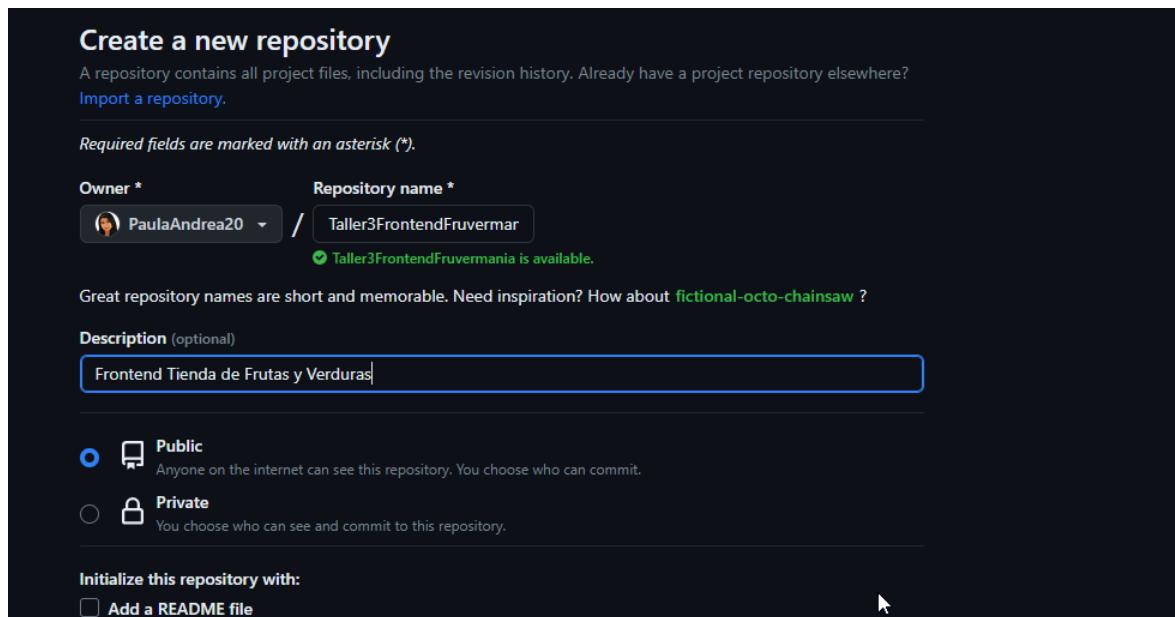
A screenshot of a web browser window titled "FruverPM" showing the "Cliente" section. The URL is "localhost:4200/compras". The page displays a table of purchases with columns: ID, Nombre, Producto, Cantidad, Fecha, and Estado. The purchases listed are:

ID	Nombre	Producto	Cantidad	Fecha	Estado
25	Rosita	Manzanas	10	29/07/2023	Aceptado
30	Rosita	Cebolla	5	29/07/2023	Rechazado
40	Rosita	Guayaba	4	01/08/2023	Pendiente



## 6. Subir el código y el informe a un repositorio remoto (GitHub)

- 6.1 Ingresamos a GitHub y Creamos un nuevo repositorio con el nombre Taller2FrontendFruvermania.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* Repository name \*

PaulaAndrea20 / Taller3FrontendFruvermar

Taller3FrontendFruvermania is available.

Great repository names are short and memorable. Need inspiration? How about [fictional-octo-chainsaw](#) ?

Description (optional)

Frontend Tienda de Frutas y Verduras

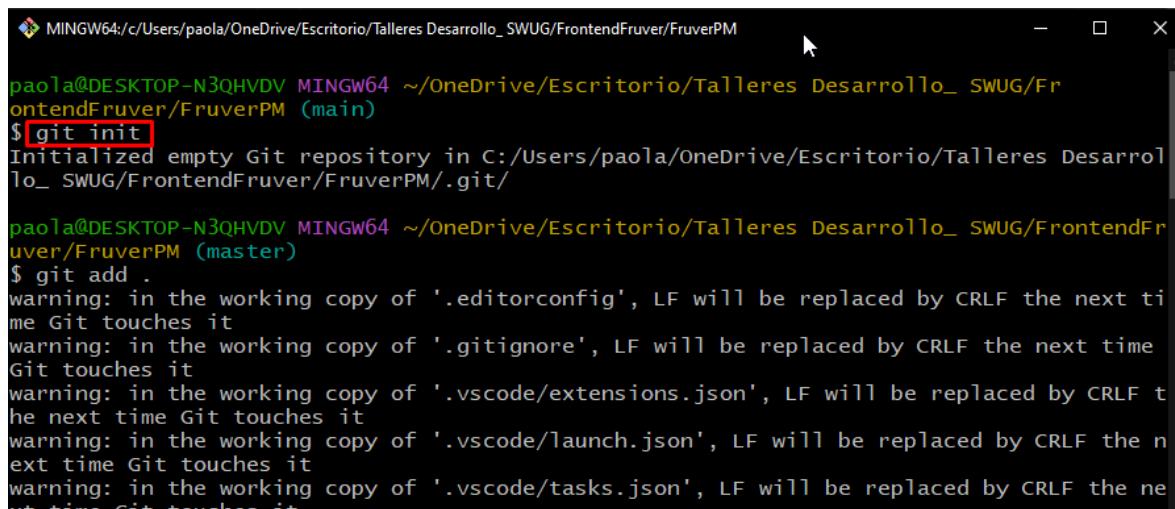
Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

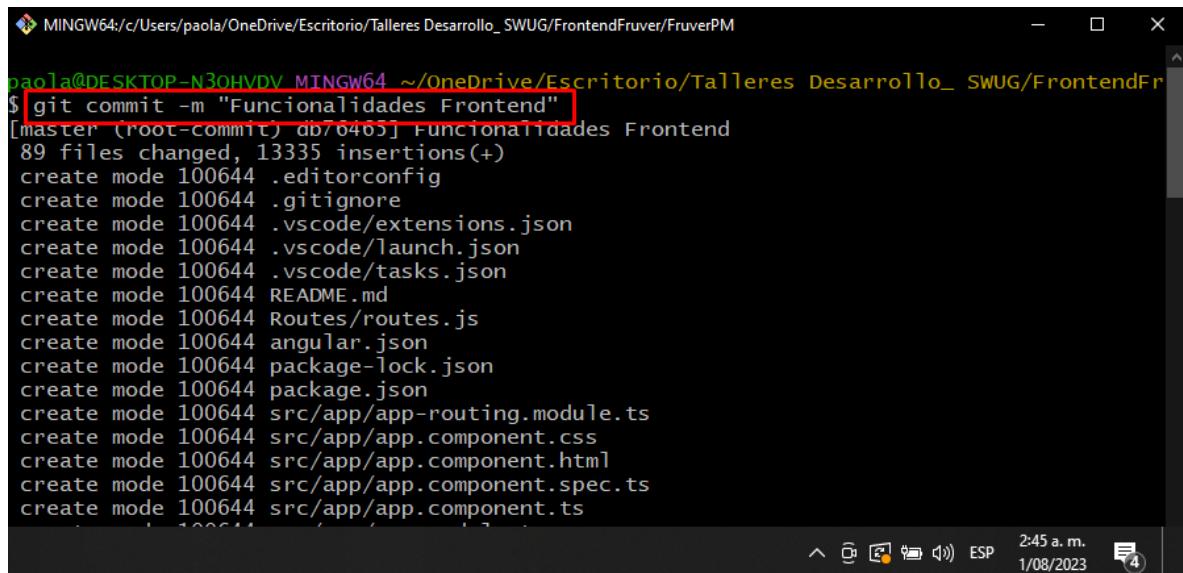
Add a README file

- 6.2 Inicializamos un repositorio local de nuestro directorio, agregamos todos los archivos y verificamos que efectivamente se hayan agregado.



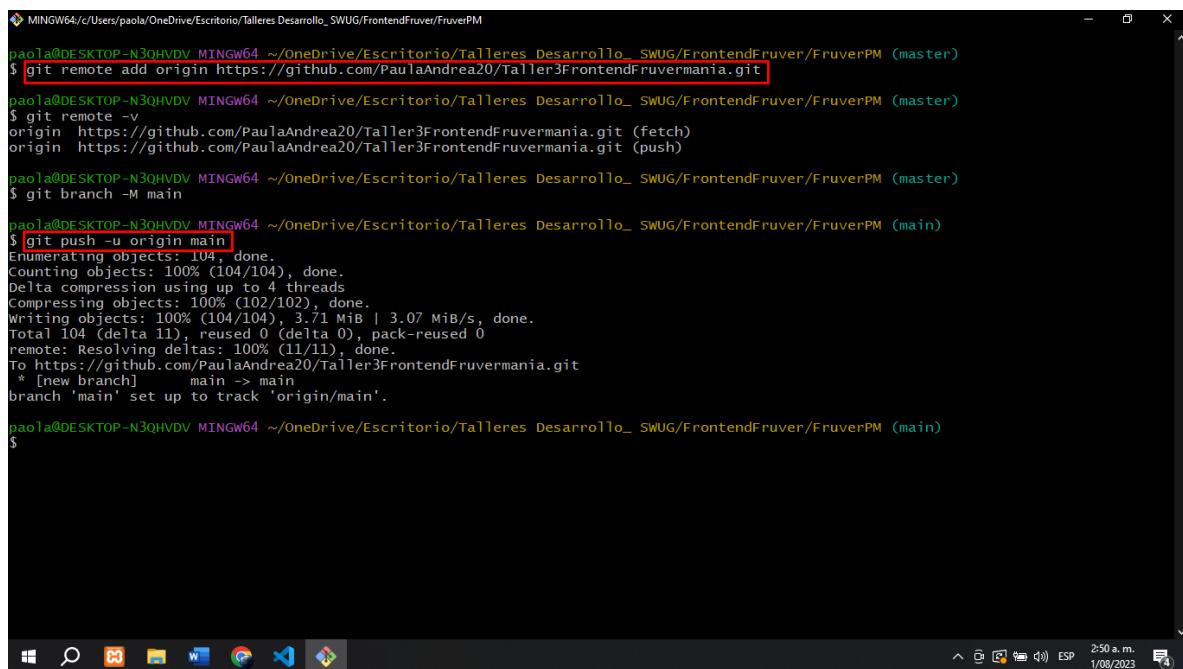
```
pao1a@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM
$ git init
Initialized empty Git repository in c:/Users/pao1a/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM/.git/
pao1a@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM
$ git add .
warning: in the working copy of '.editorconfig', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.vscode/extensions.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.vscode/launch.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.vscode/tasks.json', LF will be replaced by CRLF the next time Git touches it
```

**6.3** Realizamos un commit con el nombre **Funcionalidades Frontend** para guardar los cambios en el repositorio.



```
paola@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM
$ git commit -m "Funcionalidades Frontend"
[master (root-commit) db7d046] Funcionalidades Frontend
 89 files changed, 13335 insertions(+)
  create mode 100644 .editorconfig
  create mode 100644 .gitignore
  create mode 100644 .vscode/extensions.json
  create mode 100644 .vscode/launch.json
  create mode 100644 .vscode/tasks.json
  create mode 100644 README.md
  create mode 100644 Routes/routes.js
  create mode 100644 angular.json
  create mode 100644 package-lock.json
  create mode 100644 package.json
  create mode 100644 src/app/app-routing.module.ts
  create mode 100644 src/app/app.component.css
  create mode 100644 src/app/app.component.html
  create mode 100644 src/app/app.component.spec.ts
  create mode 100644 src/app/app.component.ts
```

**6.4** Por último, vinculamos con el repositorio remoto.



```
paola@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM (master)
$ git remote add origin https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git
paola@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM (master)
$ git remote -v
origin https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git (fetch)
origin https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git (push)

paola@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM (master)
$ git branch -M main

paola@DESKTOP-N3QHVDV MINGW64 ~/OneDrive/Escritorio/Talleres Desarrollo_SWUG/FrontendFruver/FruverPM (main)
$ git push -u origin main
Enumerating objects: 104, done.
Counting objects: 100% (104/104), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (102/102), done.
Writing objects: 100% (104/104), 3.71 MiB | 3.07 MiB/s, done.
Total 104 (delta 11), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (11/11), done.
To https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Podemos verificar que efectivamente se configuro el repositorio remoto.

Enlace de GIT: <https://github.com/PaulaAndrea20/Taller3FrontendFruvermania.git>

