

**Diplomado de actualización en nuevas tecnologías para el desarrollo de
software**

Taller Unidad 3 Frontend

LEIDY TATIANA URBANO YELA

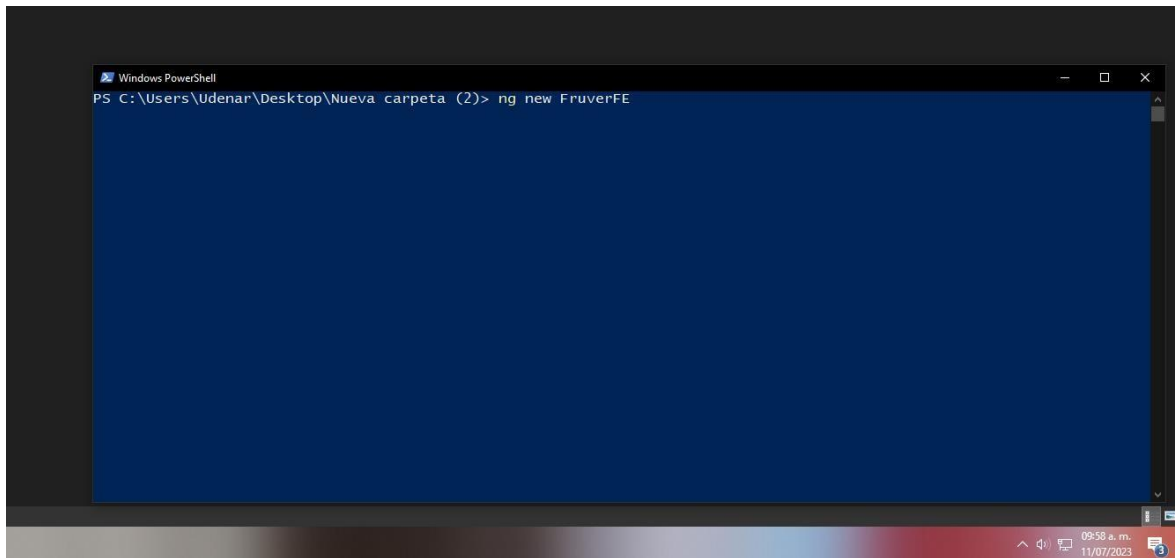
Correo: leidyurbano@udenar.edu.co

UNIVERSIDAD DE NARIÑO

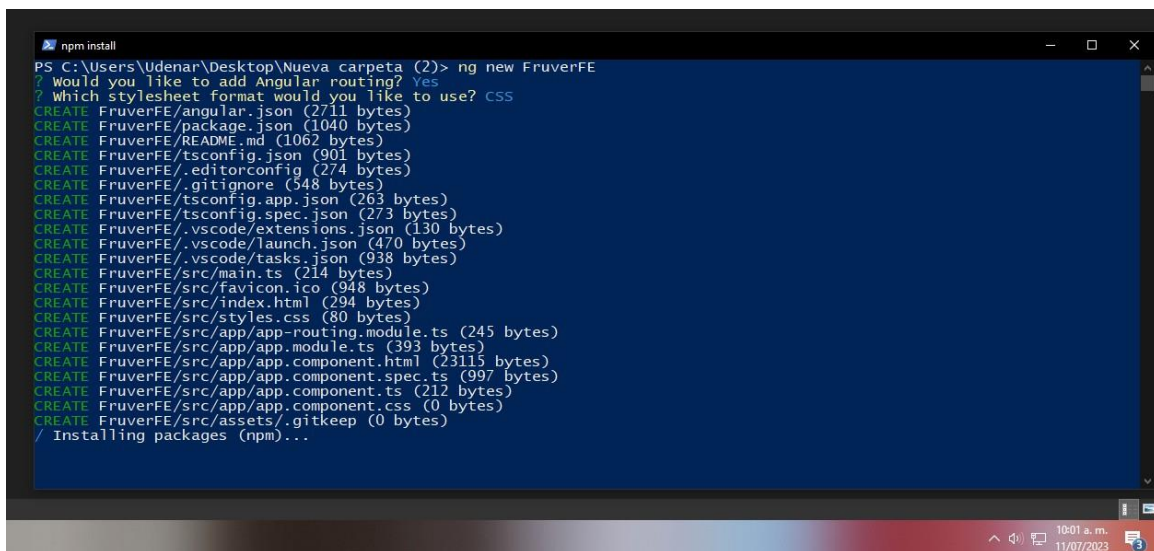
INGENIERIA DE SISTEMAS

2023

En la terminal o línea de comandos, ejecuta el comando “ng new < nombre-del-proyecto”> para crear un nuevo proyecto Angular. Se reemplaza "nombre-del-proyecto" con el nombre que se desea darle al proyecto. Angular CLI creará una nueva carpeta con ese nombre y generará la estructura inicial del proyecto.

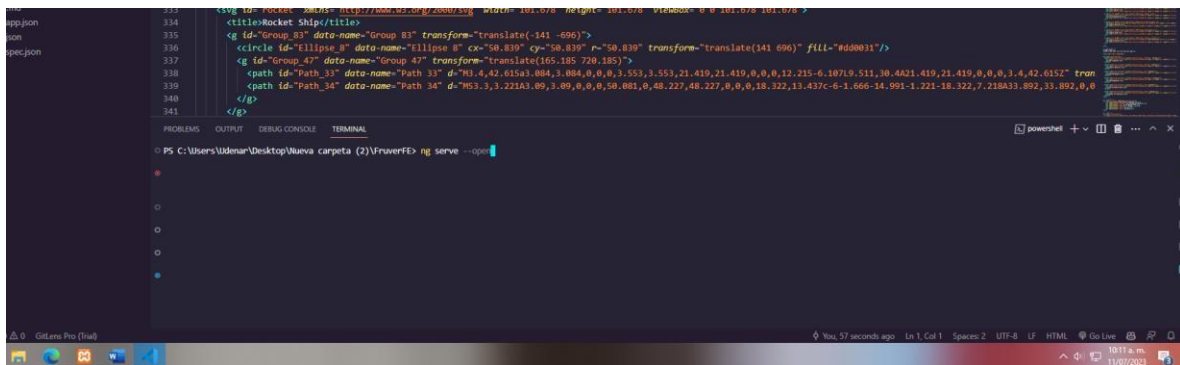


```
Windows PowerShell
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)> ng new FruverFE
```

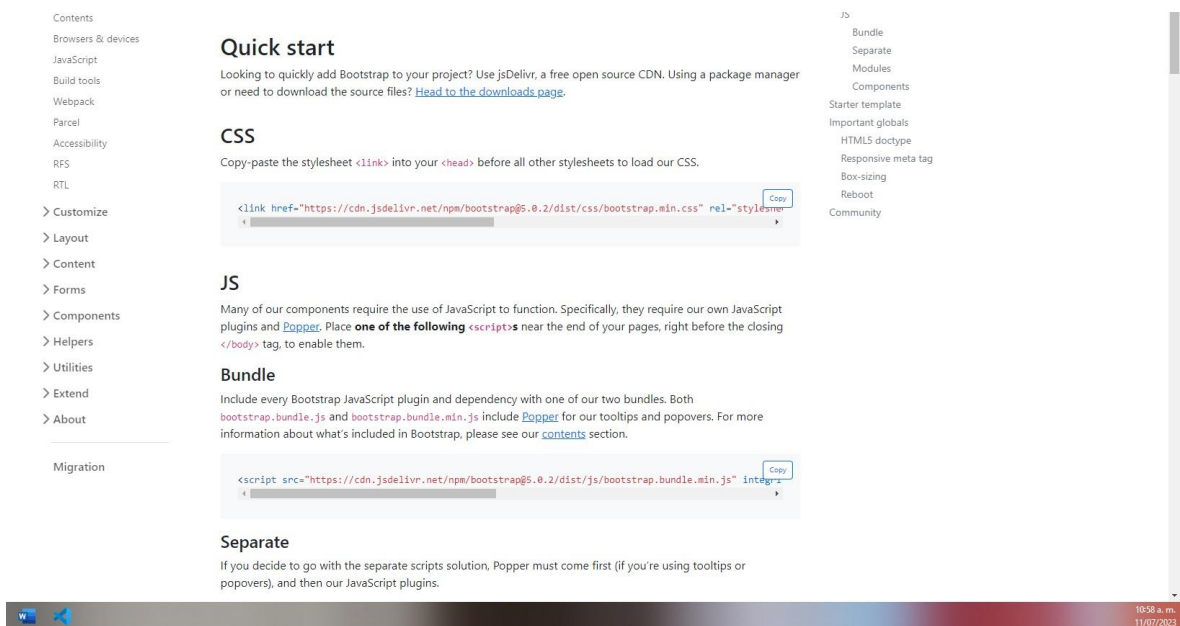


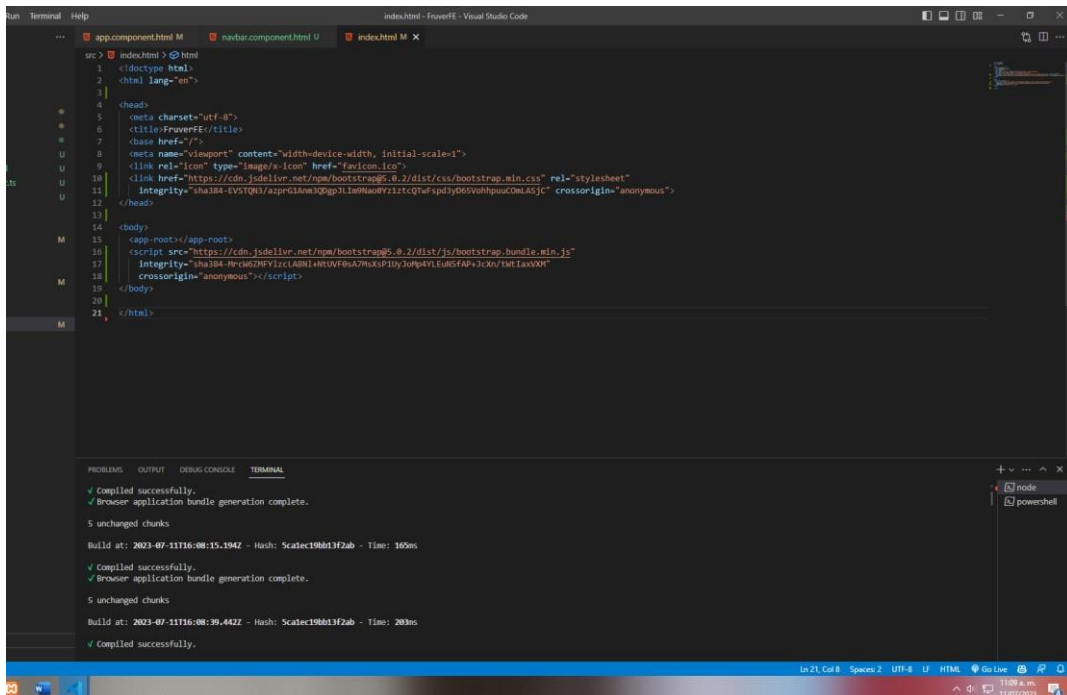
```
npm install
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)> ng new FruverFE
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE FruverFE/angular.json (2711 bytes)
CREATE FruverFE/package.json (1040 bytes)
CREATE FruverFE/README.md (1062 bytes)
CREATE FruverFE/tsconfig.json (901 bytes)
CREATE FruverFE/.editorconfig (274 bytes)
CREATE FruverFE/.gitignore (548 bytes)
CREATE FruverFE/tsconfig.app.json (263 bytes)
CREATE FruverFE/tsconfig.spec.json (273 bytes)
CREATE FruverFE/.vscode/extensions.json (130 bytes)
CREATE FruverFE/.vscode/launch.json (470 bytes)
CREATE FruverFE/.vscode/tasks.json (938 bytes)
CREATE FruverFE/src/main.ts (214 bytes)
CREATE FruverFE/src/favicon.ico (948 bytes)
CREATE FruverFE/src/index.html (294 bytes)
CREATE FruverFE/src/styles.css (80 bytes)
CREATE FruverFE/src/app/app-routing.module.ts (245 bytes)
CREATE FruverFE/src/app/app.module.ts (393 bytes)
CREATE FruverFE/src/app/app.component.html (23115 bytes)
CREATE FruverFE/src/app/app.component.spec.ts (997 bytes)
CREATE FruverFE/src/app/app.component.ts (212 bytes)
CREATE FruverFE/src/app/app.component.css (0 bytes)
CREATE FruverFE/src/assets/.gitkeep (0 bytes)
/ Installing packages (npm)...
```

Después de que se complete la creación del proyecto, ingresa al directorio del proyecto recién creado “cd nombre-del-proyecto” y se ejecuta el comando ng serve –open para iniciar el servidor de desarrollo de Angular.



Copiamos los enlaces CDN de Bootstrap y los agregamos en el archivo index.html de nuestro proyecto, esto con el fin de usar los diferentes estilos que ofrece este framework





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

```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>FruverFE</title>
7   <base href="/">
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="icon" type="image/x-icon" href="favicon.ico">
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
11    integrity="sha384-EVSTQN3/azprG1Anm3QDgp 3L7HRNdO8Yz1tcQ7fspd3yD65uHhhuusCmlASjC" crossorigin="anonymous">
12 </head>
13
14 <body>
15   <app-root></app-root>
16   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
17     integrity="sha384-M6wP8oJFv5Se1Fyde50gM4vkrhphQdiyBNiH4Y7oPzmoDfEaKb1C/dW8pARw"
18     crossorigin="anonymous"></script>
19 </body>
20
21 </html>
```

The terminal at the bottom shows the following output:

```
✓ Compiled successfully.
✓ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2023-07-11T16:08:15.194Z - Hash: Scatec1986t3f2ab - Time: 165ms

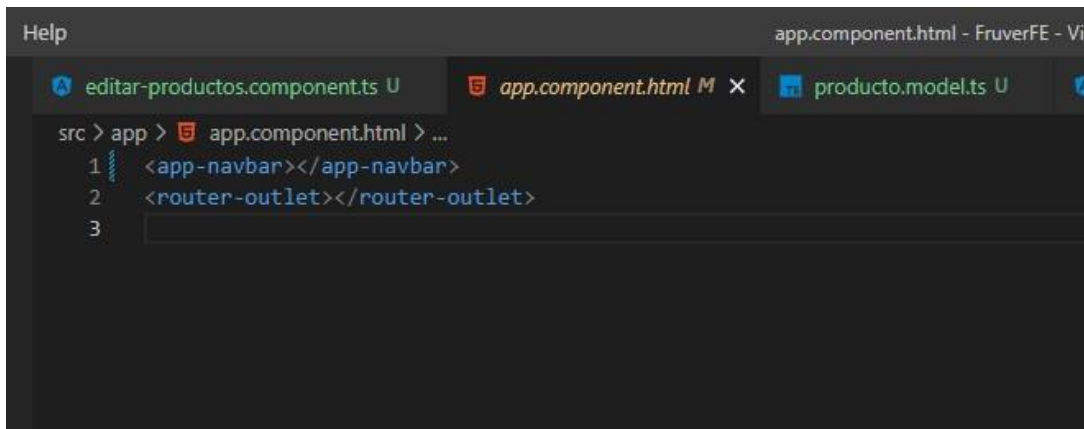
✓ Compiled successfully.
✓ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2023-07-11T16:08:39.442Z - Hash: Scatec1986t3f2ab - Time: 200ms

✓ Compiled successfully.
```

Agregamos la etiqueta `<router-outlet>` en el archivo `app.component.html` lo que nos permitirá mostrar el componente asociado a una ruta específica en una aplicación con enrutamiento.



The screenshot shows the Visual Studio Code editor with the `app.component.html` file open. The file contains the following HTML code:

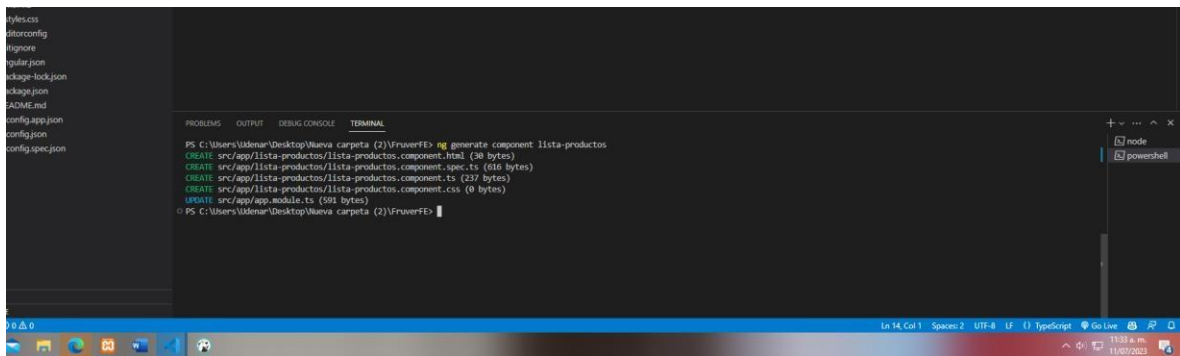
```
1 <app-navbar></app-navbar>
2 <router-outlet></router-outlet>
3
```

Con el comando “ng generate component <nombre-del-componente>” creamos todos los componentes que vamos a utilizar:

- Lista productos.
- Lista pedidos.
- Carrito.
- Detalle producto.
- Editar productos.
- Iniciar sesión.
- Navbar.

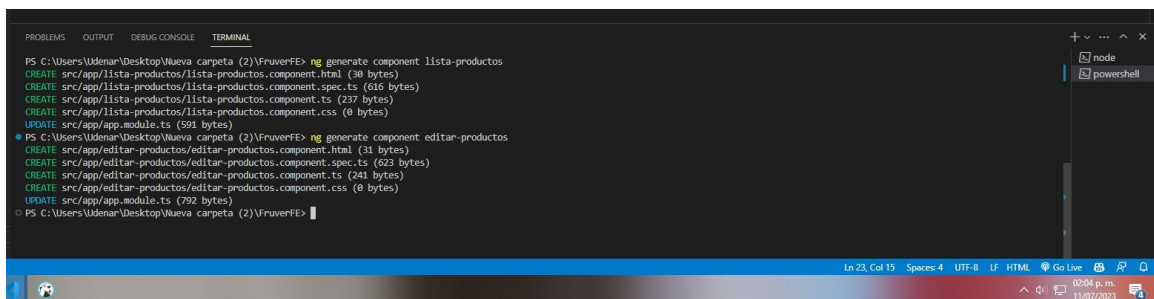
Con el comando “ng generate service <nombre-del-servicio>” creamos todos los servicios que vamos a utilizar, estos tendrán la lógica para realizar peticiones http al servidor.

- Cliente.
- Pedido.
- Producto.
- Pedido producto.
- Login.



```
styles.css
editorconfig
tsconfig
tsconfig.json
tsconfig.spec.json
tsconfig.app.json
tsconfig.spec.json
tsconfig.spec.json

PS C:\Users\Udenar\Desktop\Nueva carpeta (2)\FruterFE> ng generate component lista-productos
CREATE src/app/lista-productos/lista-productos.component.html (30 bytes)
CREATE src/app/lista-productos/lista-productos.component.spec.ts (616 bytes)
CREATE src/app/lista-productos/lista-productos.component.ts (237 bytes)
CREATE src/app/lista-productos/lista-productos.component.css (0 bytes)
UPDATE src/app/app.module.ts (591 bytes)
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)\FruterFE>
```



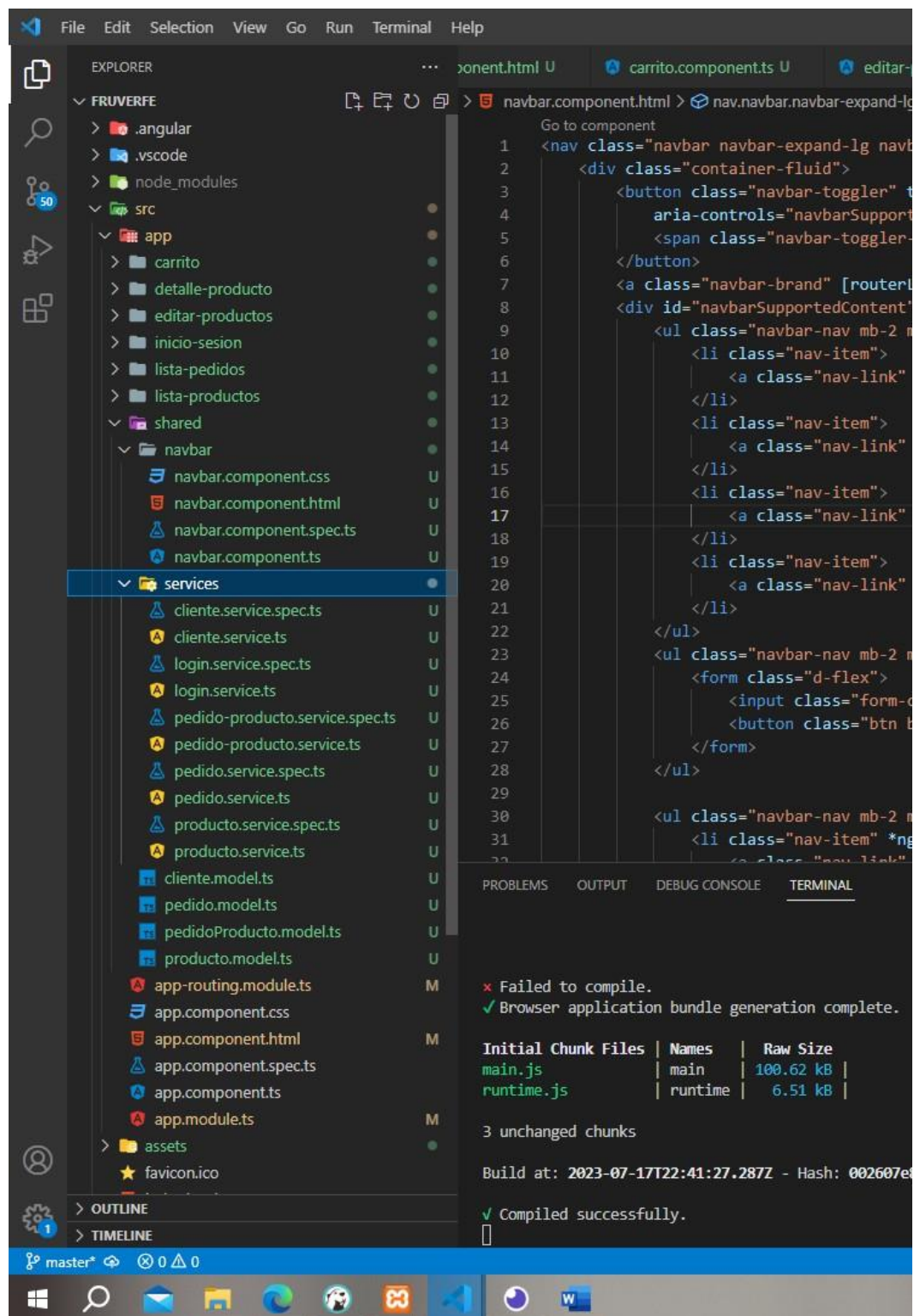
```
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)\FruterFE> ng generate component editar-productos
CREATE src/app/editar-productos/editar-productos.component.html (31 bytes)
CREATE src/app/editar-productos/editar-productos.component.spec.ts (623 bytes)
CREATE src/app/editar-productos/editar-productos.component.ts (241 bytes)
CREATE src/app/editar-productos/editar-productos.component.css (0 bytes)
UPDATE src/app/app.module.ts (792 bytes)
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)\FruterFE>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE> ng generate component carrito
CREATE src/app/carrito/carrito.component.html (22 bytes)
CREATE src/app/carrito/carrito.component.spec.ts (566 bytes)
CREATE src/app/carrito/carrito.component.ts (206 bytes)
CREATE src/app/carrito/carrito.component.css (0 bytes)
UPDATE src/app/app.module.ts (1176 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE>
```

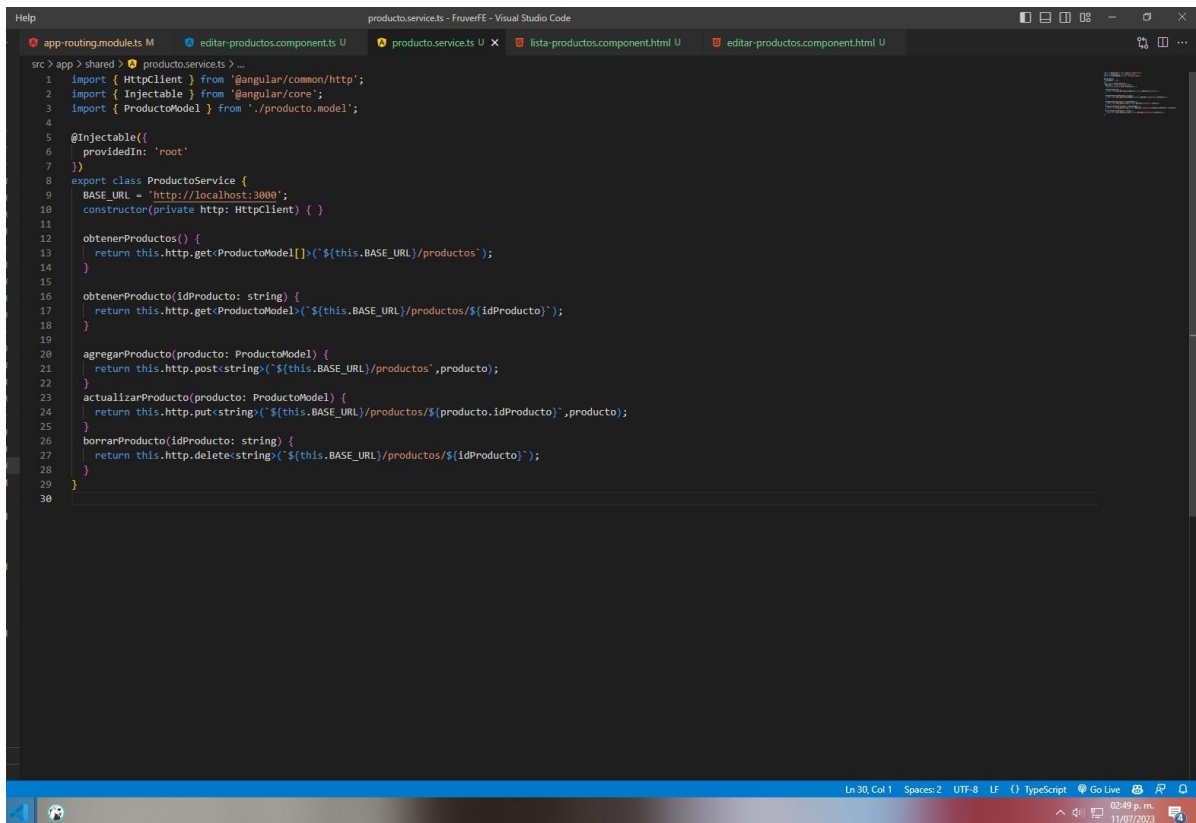
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE> ng generate service shared/services/pedido
CREATE src/app/shared/services/pedido.service.spec.ts (357 bytes)
CREATE src/app/shared/services/pedido.service.ts (135 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE> ng generate service shared/services/pedidoProducto
CREATE src/app/shared/services/pedido-producto.service.spec.ts (398 bytes)
CREATE src/app/shared/services/pedido-producto.service.ts (143 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE> ng generate service shared/services/login
CREATE src/app/shared/services/login.service.spec.ts (352 bytes)
CREATE src/app/shared/services/login.service.ts (134 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE> ng generate component shared/navbar
CREATE src/app/shared/navbar/navbar.component.html (21 bytes)
CREATE src/app/shared/navbar/navbar.component.spec.ts (559 bytes)
CREATE src/app/shared/navbar/navbar.component.ts (282 bytes)
CREATE src/app/shared/navbar/navbar.component.css (0 bytes)
UPDATE src/app/app.module.ts (982 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FrueverFE>
```

La estructura del proyecto quedará de la siguiente forma:

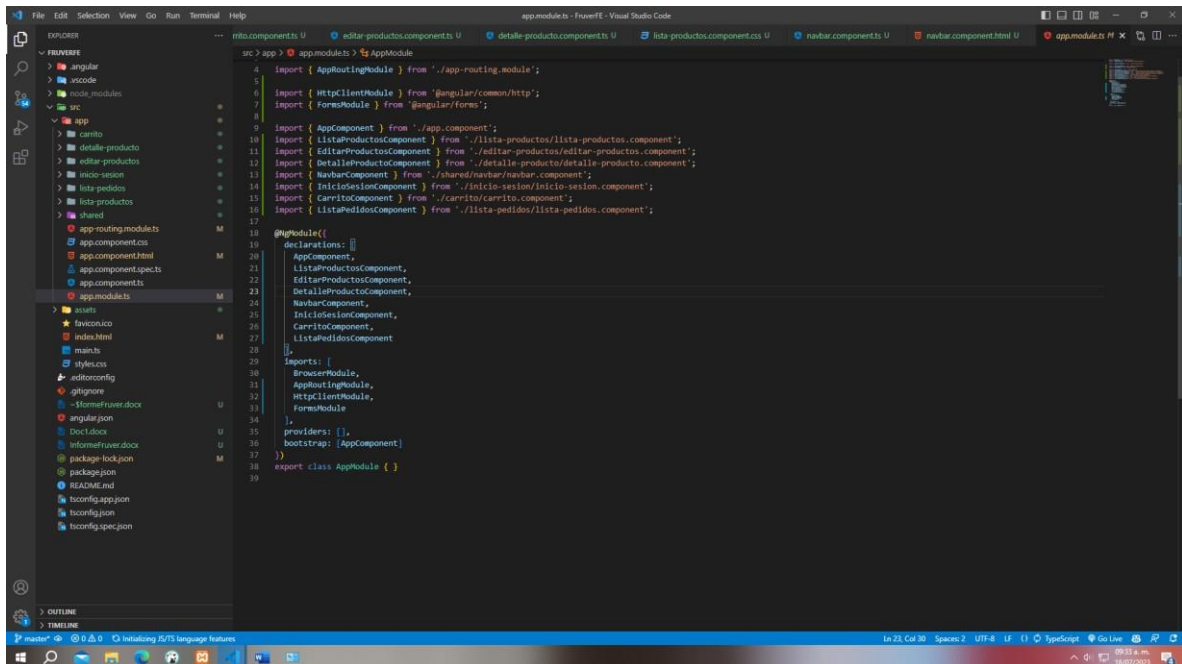


Producto service tiene los métodos para la gestión de los datos de los productos. Se encarga de realizar las peticiones para crear, actualizar, obtener y eliminar productos.

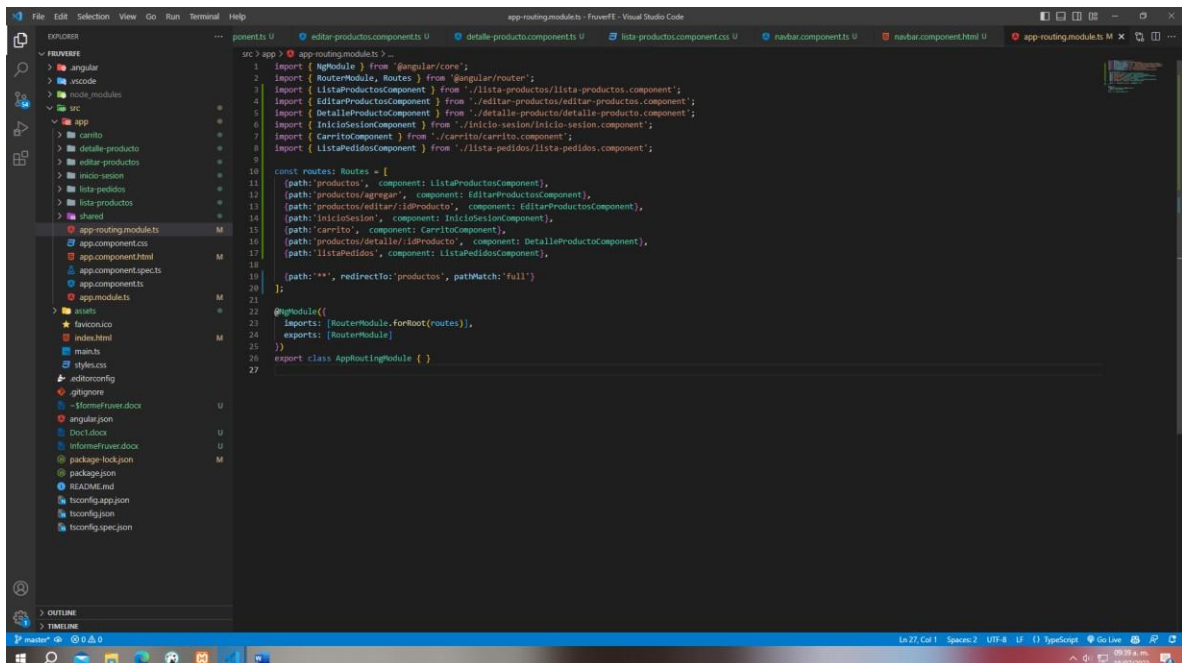


```
src > app > shared > producto.service.ts > ...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { ProductoModel } from './producto.model';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class ProductoService {
9   BASE_URL = 'http://localhost:3000';
10   constructor(private http: HttpClient) { }
11
12   obtenerProductos() {
13     return this.http.get<ProductoModel[]>(`${this.BASE_URL}/productos`);
14   }
15
16   obtenerProducto(idProducto: string) {
17     return this.http.get<ProductoModel>(`${this.BASE_URL}/productos/${idProducto}`);
18   }
19
20   agregarProducto(producto: ProductoModel) {
21     return this.http.post<string>(`${this.BASE_URL}/productos`, producto);
22   }
23   actualizarProducto(producto: ProductoModel) {
24     return this.http.put<string>(`${this.BASE_URL}/productos/${producto.idProducto}`, producto);
25   }
26   borrarProducto(idProducto: string) {
27     return this.http.delete<string>(`${this.BASE_URL}/productos/${idProducto}`);
28   }
29 }
30
```

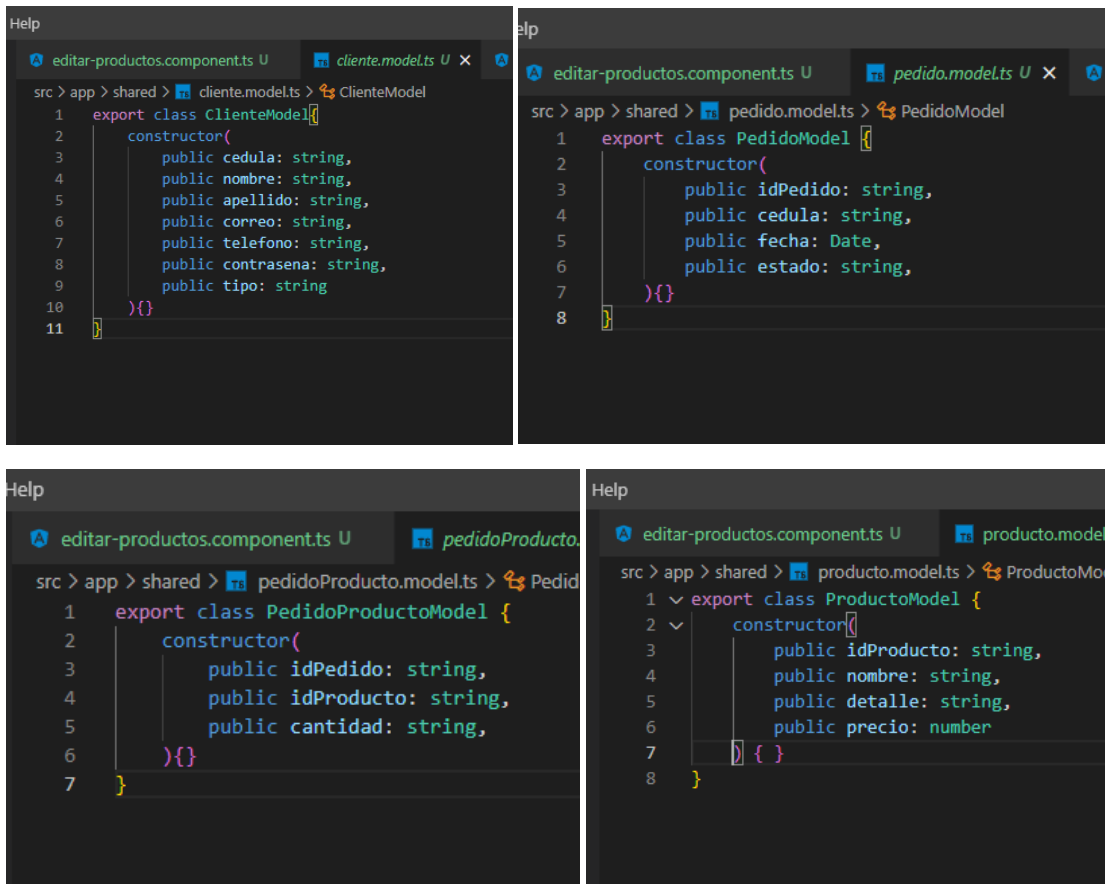
Todos los componentes que se hayan creado o se crearan deben importarse en el archivo app.module.ts



Cuando se crea un modulo se debe agregar una ruta en el archivo app-routing.module.ts que redirija a ese componente.



Se crean modelos de productos, pedido, pedidoProducto y cliente que representan las tablas de la base de datos.



The image displays four screenshots of a code editor, likely Visual Studio Code, showing the creation of TypeScript models for a database. Each screenshot shows a file explorer on the left and a code editor on the right. The code is written in TypeScript and uses the 'export class' syntax to define the models.

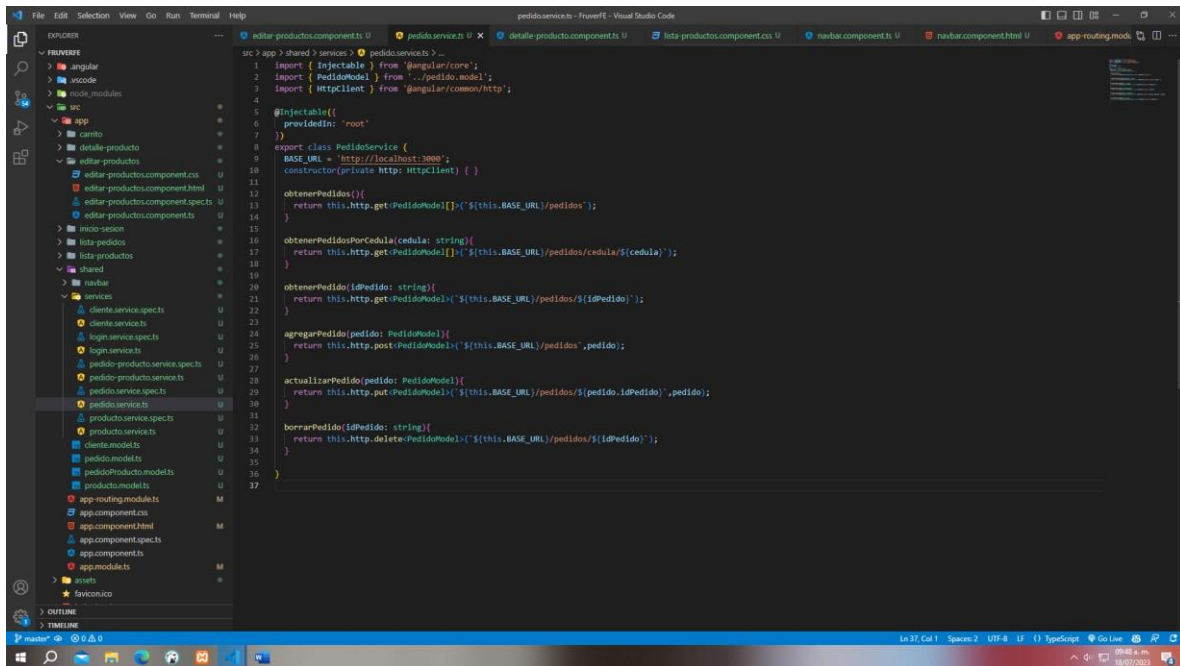
```
src > app > shared > cliente.model.ts > ClienteModel
1 export class ClienteModel {
2   constructor(
3     public cedula: string,
4     public nombre: string,
5     public apellido: string,
6     public correo: string,
7     public telefono: string,
8     public contrasena: string,
9     public tipo: string
10  ) {}
11 }
```

```
src > app > shared > pedido.model.ts > PedidoModel
1 export class PedidoModel {
2   constructor(
3     public idPedido: string,
4     public cedula: string,
5     public fecha: Date,
6     public estado: string,
7   ) {}
8 }
```

```
src > app > shared > pedidoProducto.model.ts > PedidoProductoModel
1 export class PedidoProductoModel {
2   constructor(
3     public idPedido: string,
4     public idProducto: string,
5     public cantidad: string,
6   ) {}
7 }
```

```
src > app > shared > producto.model.ts > ProductoModel
1 export class ProductoModel {
2   constructor(
3     public idProducto: string,
4     public nombre: string,
5     public detalle: string,
6     public precio: number
7   ) {}
8 }
```

En los servicios creados se implementarán los métodos de buscar, crear, editar y eliminar realizando solicitudes http al servidor, estos servicios retornan los datos Json en forma de objetos que instancian a los modelos ya creados.



Para hacer uso del ngModel, es necesario importar la librería FormsModule en el archivo app.module.ts.

```
4 import { AppRoutingModule } from './app-routing.module';
5
6 import { HttpClientModule } from '@angular/common/http';
7 import { FormsModule } from '@angular/forms';
8
9 import { AppComponent } from './app.component';
10 import { ListaProductosComponent } from './lista-productos/lista-productos.component';
11 import { EditarProductosComponent } from './editar-productos/editar-productos.component';
12 import { DetalleProductoComponent } from './detalle-producto/detalle-producto.component';
13 import { NavbarComponent } from './shared/navbar/navbar.component';
14 import { InicioSesionComponent } from './inicio-sesion/inicio-sesion.component';
15 import { CarritoComponent } from './carrito/carrito.component';
16 import { ListaPedidosComponent } from './lista-pedidos/lista-pedidos.component';
17
18 @NgModule({
19   declarations: [
20     AppComponent,
21     ListaProductosComponent,
22     EditarProductosComponent,
23     DetalleProductoComponent,
24     NavbarComponent,
25     InicioSesionComponent,
26     CarritoComponent,
27     ListaPedidosComponent
28   ],
29   imports: [
30     BrowserModule,
31     AppRoutingModule,
32     HttpClientModule,
33     FormsModule
34   ],
35   providers: []
36 })
37 export class AppModule {}
```

Se creará un formulario en el archivo html del componente editar-producto, usando la directiva Ngform para validar el formulario.

En el botón de tipo submit tiene una condición que deshabilitará el botón cuando el formulario no esté completo.

```
1
2 <div class="container">
3   <h3 class="my-4">{{titulo}}</h3>
4   <div class="row">
5     <div class="col-sm-12">
6       <form (ngSubmit)="onSubmit()" #productoForm="ngForm">
7         <fieldset class="form-group">
8           <div class="col-sm-12">
9             <label class="control-label" for="nombre">Nombre</label>
10            <input type="text" class="form-control" required [(ngModel)]="producto.nombre" name="nombre">
11          </div>
12        </fieldset>
13        <fieldset class="form-group">
14          <div class="col-sm-12">
15            <label class="control-label" for="email">Detalle</label>
16            <textarea class="form-control" required [(ngModel)]="producto.detalle" name="detalle" rows="3"></textarea>
17          </div>
18        </fieldset>
19        <fieldset class="form-group">
20          <div class="col-sm-12">
21            <label class="control-label" for="telefono">Precio</label>
22            <input type="text" class="form-control" required [(ngModel)]="producto.precio" name="precio">
23          </div>
24        </fieldset>
25        <fieldset class="form-group my-3">
26          <div class="col-sm-offset-2 col-sm-10">
27            <button type="submit" class="btn btn-primary" [disabled]="!productoForm.form.valid">Enviar</button>
28          </div>
29        </fieldset>
30      </form>
31    </div>
32  </div>
33</div>
34
35
```

Y en el archivo typescript del componente se crea un objeto que instancia al modelo producto y se implementa la lógica para crear o editar el producto, al finalizar la operación se redirigirá a la página principal (/productos) usando el objeto router y la ruta definida en el archivo app-routing.module.ts.

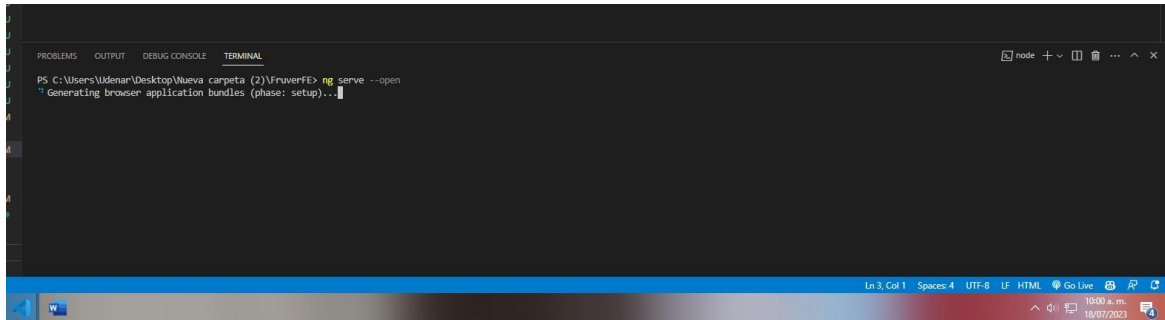
```
help | editar-productos.component.ts - FruverFE - Visual Studio Code

editar-productos.component.ts U x producto.model.ts U detalle-producto.component.ts U lista-productos.component.css U n

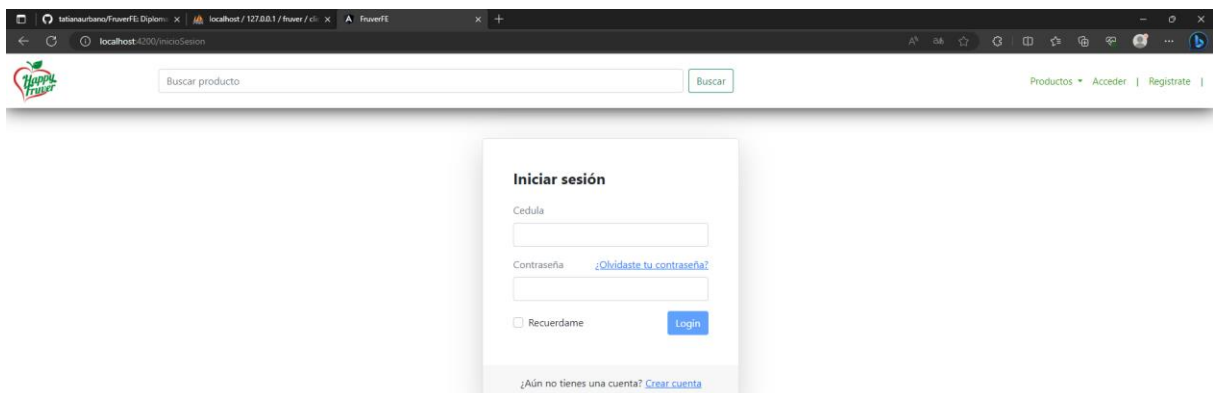
src > app > editar-productos > editar-productos.component.ts > EditorProductosComponent > ngOnInit
1 import { Component, OnInit } from '@angular/core';
2 import { ProductoModel } from '../shared/producto.model';
3 import { ProductoService } from '../shared/services/producto.service';
4 import { ActivatedRoute, Router } from '@angular/router';
5 import { LoginService } from '../shared/services/login.service';
6
7 @Component({
8   selector: 'app-editar-productos',
9   templateUrl: './editar-productos.component.html',
10  styleUrls: ['./editar-productos.component.css']
11 })
12 export class EditorProductosComponent implements OnInit {
13
14   idProducto = '';
15   producto = new ProductoModel("", "", "", 0);
16   mensaje = '';
17   titulo = 'Crear Producto';
18
19   constructor(private productoService: ProductoService, private route: ActivatedRoute, private router: Router,
20     public loginService: LoginService ) { }
21
22 }
```

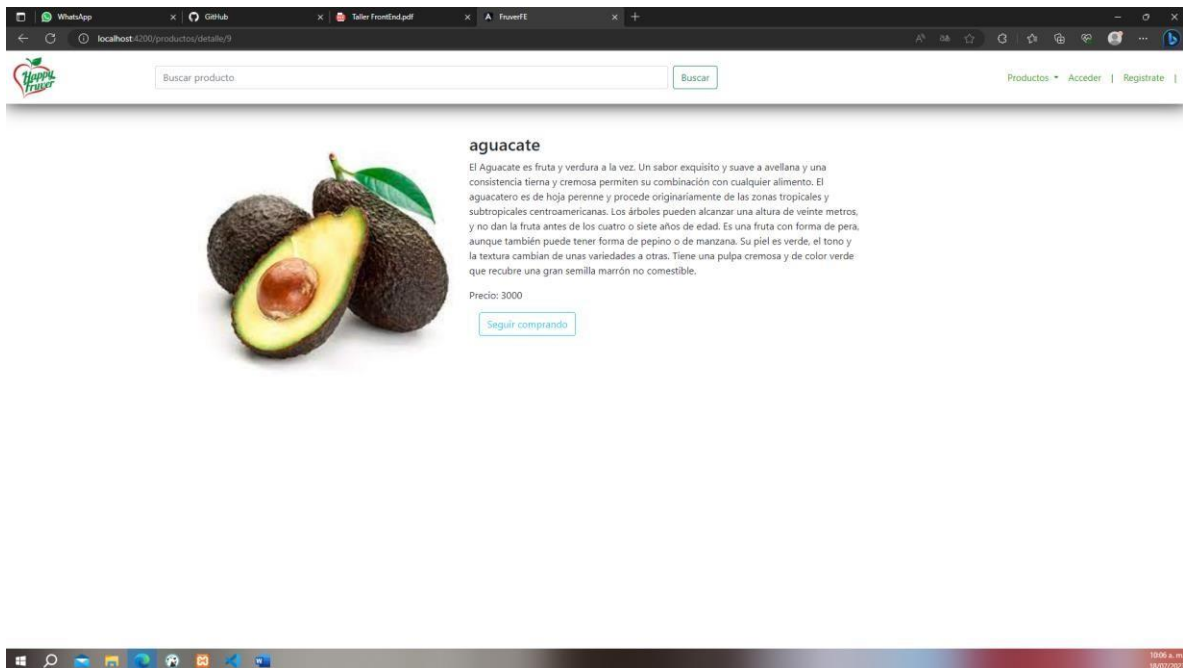
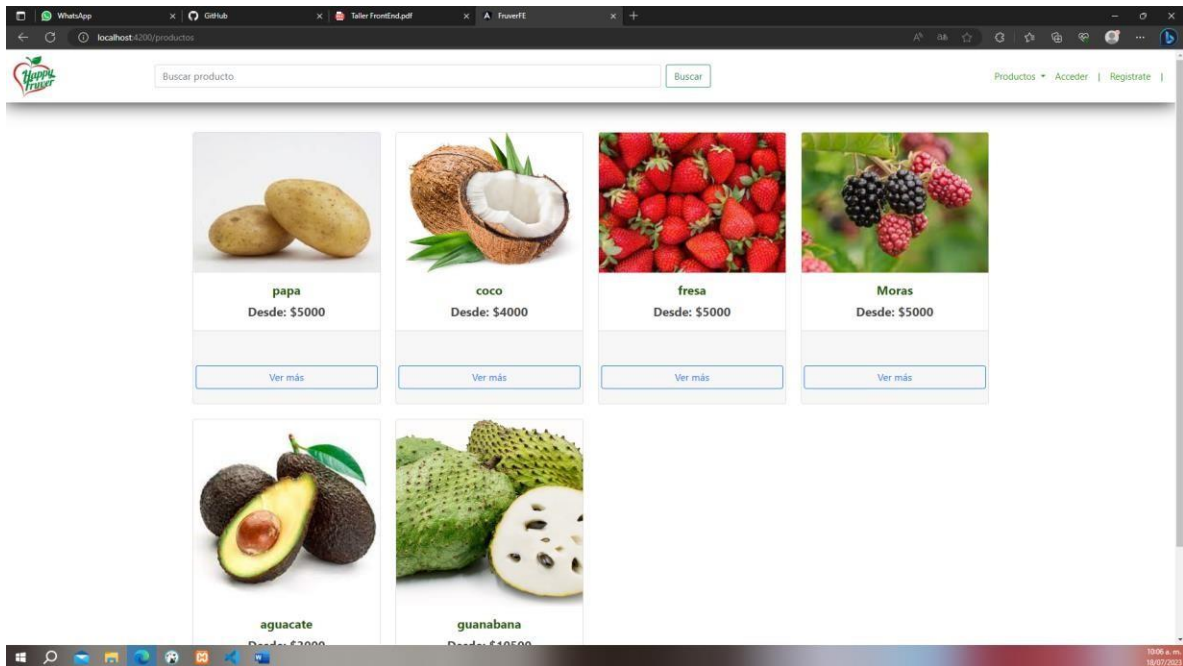
```
46
47 // funcion para crear o editar un producto
48 onSubmit() {
49   console.log("Submit realizado");
50   if (this.producto.idProducto) {
51     //Viene de Editar
52     this.productoService.actualizarProducto(this.producto).subscribe(
53       data => {
54         console.log(data);
55         this.router.navigate(['/productos']);
56       }
57     );
58   }
59   else {
60     //Viene de crear Nuevo Producto
61     console.log('Nuevo Producto');
62     this.productoService.agregarProducto(this.producto).subscribe(data => {
63       this.router.navigate(['/productos']);
64     });
65   }
66
67 }
68
69 }
```

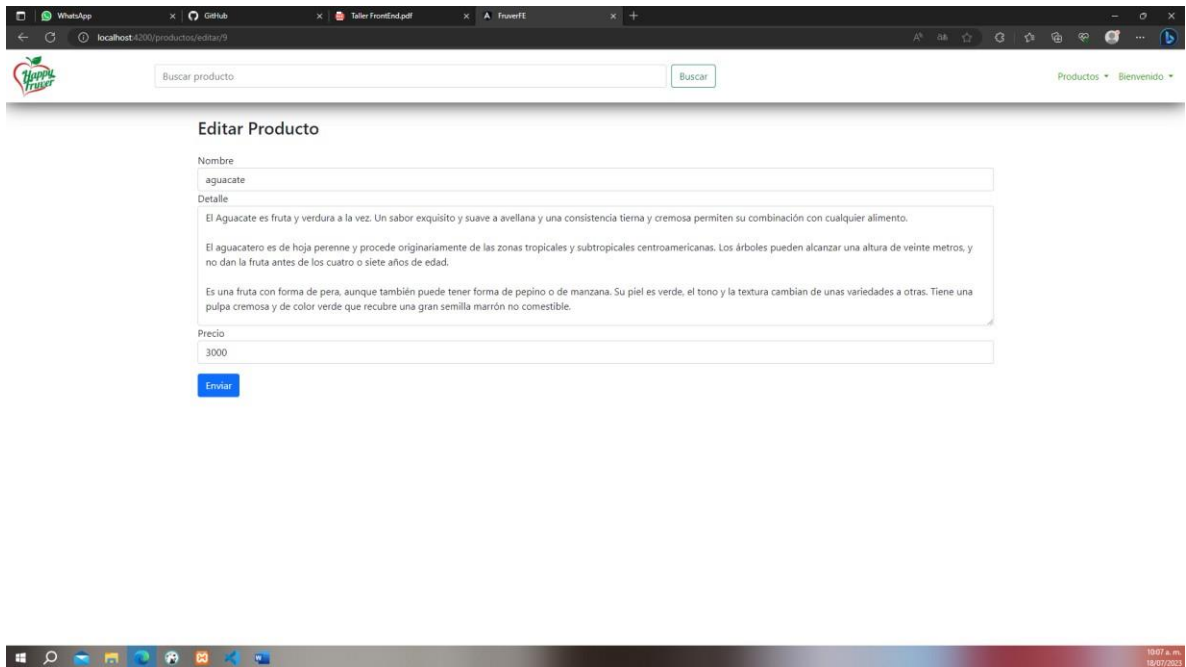
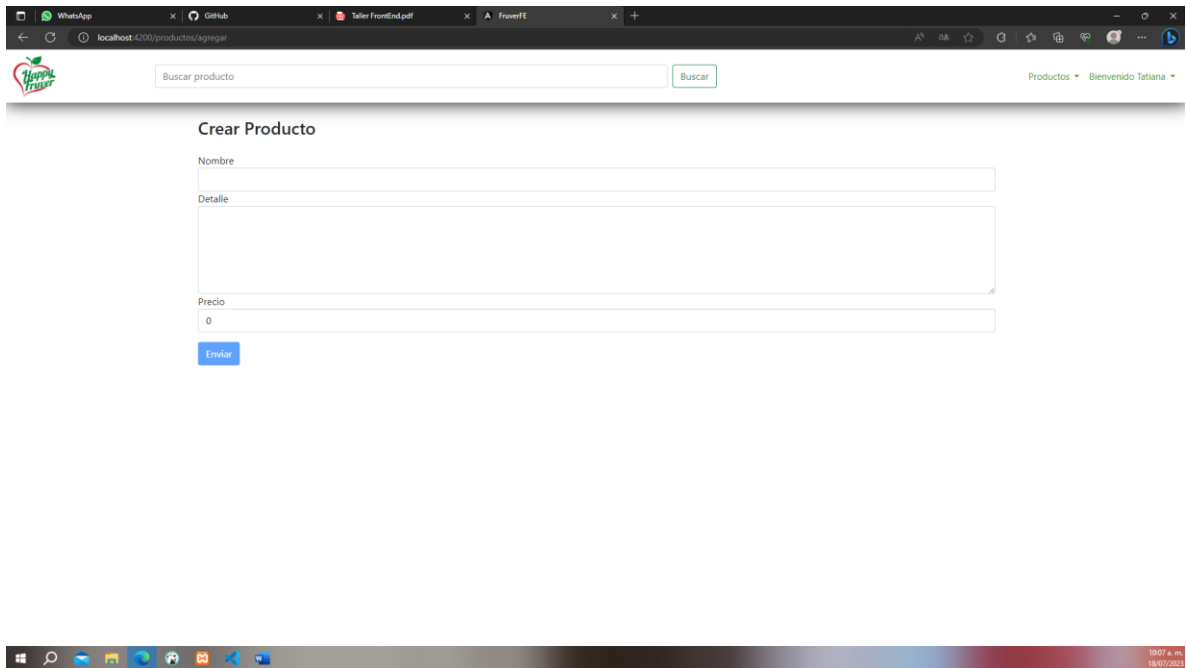
Con el comando `ng serve --open` iniciamos nuestro proyecto



Resultado







tatianaurbano/FruveFE: Diplom... localhost / 127.0.0.1 / fru... FruveFE

localhost:4200/carrito

Buscar producto

Productos • Bienvenido Leidy •

Producto	Cantidad	Precio	Fecha solicitud	Estado	Total	Acciones
papa	17	5000\$	2023-07-13	Aceptado	85000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
coco	30	4000\$	2023-07-13	Rechazado	120000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
coco	<input type="text" value="8"/>	4000\$	2023-07-27	Pendiente	32000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
Moras	<input type="text" value="8"/>	5000\$	2023-07-27	Pendiente	40000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
guanabana	<input type="text" value="8"/>	10500\$	2023-07-27	Pendiente	84000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
papa	<input type="text" value="1"/>	5000\$	2023-07-27	Pendiente	5000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>
aguacate	<input type="text" value="1"/>	3000\$	2023-07-27	Pendiente	3000\$	<input type="button" value="Actualizar"/> <input type="button" value="Borrar"/>

tatianaurbano/FruveFE: Diplom... localhost / 127.0.0.1 / fru... FruveFE

localhost:4200/ListaPedidos

Buscar producto

Productos • Bienvenido Tatiana •

Cliente	Producto	Cantidad	Precio	Fecha solicitud	Estado	Total	Acciones
Tatiana Urbano	papa	17	5000\$	2023-07-13	Aceptado	85000\$	<input type="button" value="Aprobar"/> <input type="button" value="Rechazar"/>
Tatiana Urbano	coco	30	4000\$	2023-07-13	Rechazado	120000\$	<input type="button" value="Aprobar"/> <input type="button" value="Rechazar"/>
Leidy Urbano	coco	<input type="text" value="27"/>	4000\$	2023-07-27	Pendiente	108000\$	<input type="button" value="Aprobar"/> <input type="button" value="Rechazar"/>
Leidy Urbano	Moras	<input type="text" value="8"/>	5000\$	2023-07-27	Pendiente	40000\$	<input type="button" value="Aprobar"/> <input type="button" value="Rechazar"/>
Leidy Urbano	guanabana	<input type="text" value="8"/>	10500\$	2023-07-27	Pendiente	84000\$	<input type="button" value="Aprobar"/> <input type="button" value="Rechazar"/>