

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

Taller Unidad 3 Frontend

LEIDY TATIANA URBANO YELA

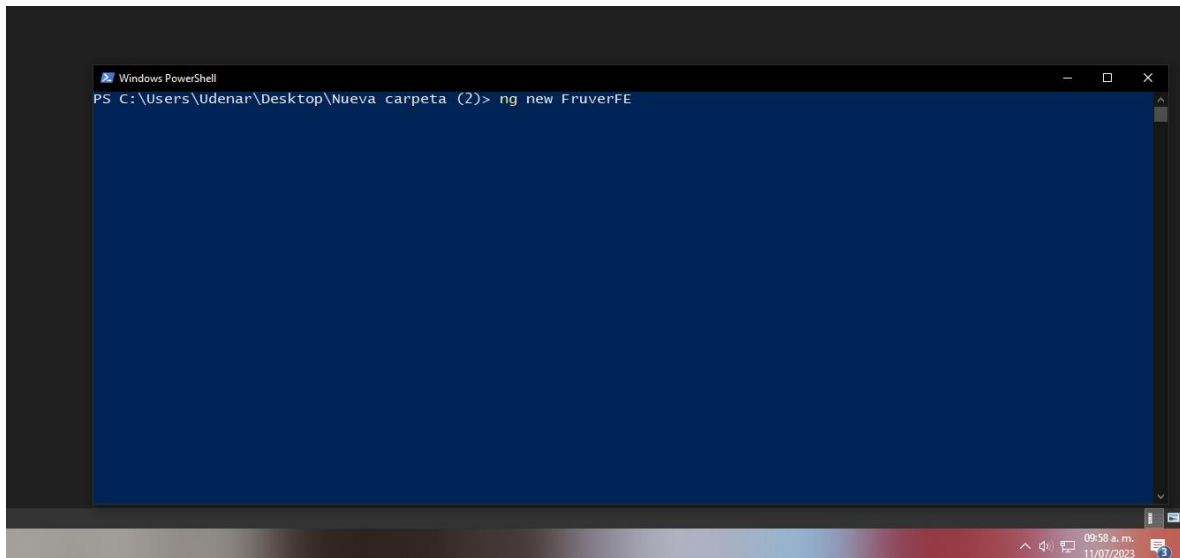
Correo: leidyurbano@udenar.edu.co

UNIVERSIDAD DE NARIÑO

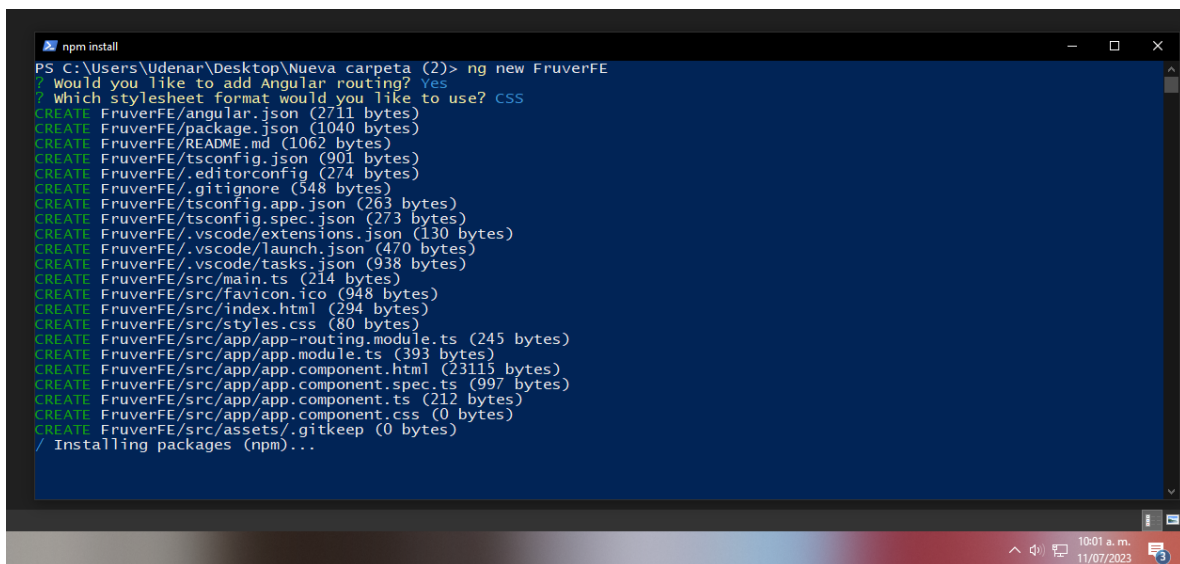
INGENIERIA DE SISTEMAS

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.

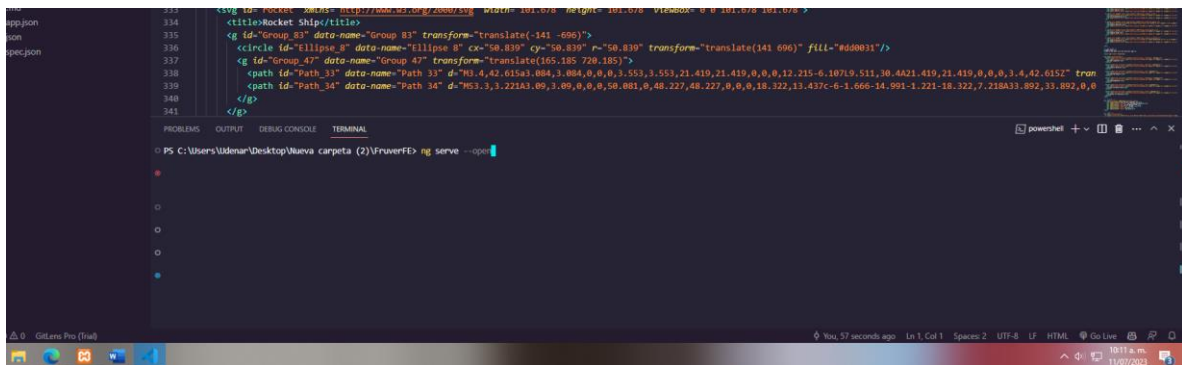


```
Windows PowerShell
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)> ng new FruverFE
```



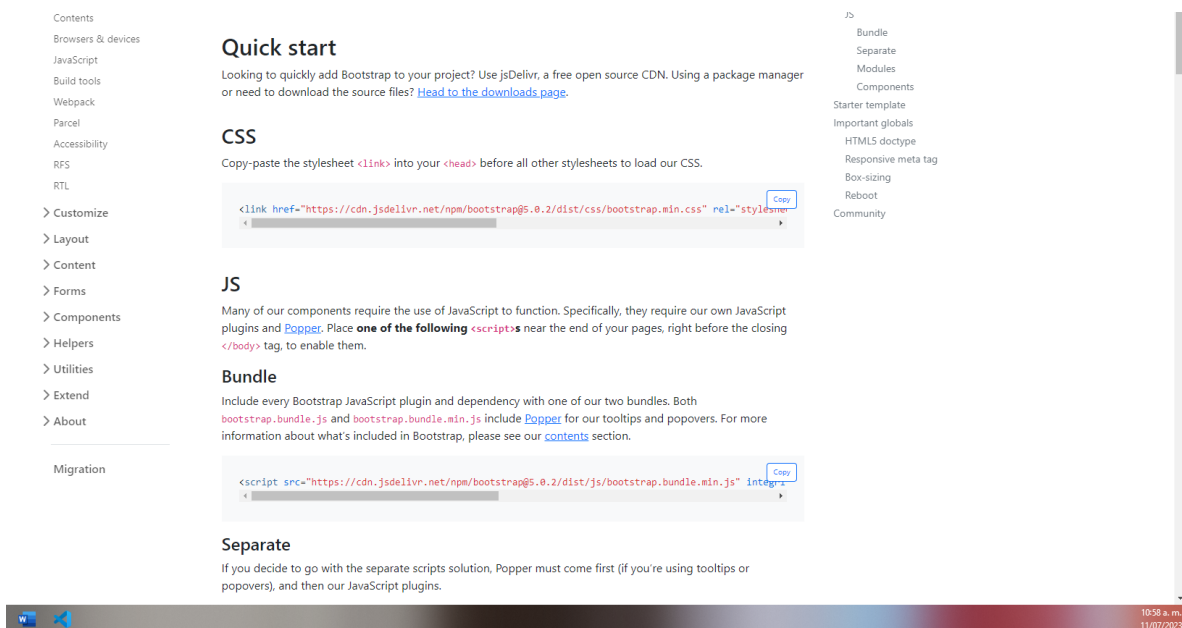
```
npm install
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)> ng new FruverFE
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE FruverFE/angular.json (2711 bytes)
CREATE FruverFE/package.json (1040 bytes)
CREATE FruverFE/README.md (1062 bytes)
CREATE FruverFE/tsconfig.json (901 bytes)
CREATE FruverFE/.editorconfig (274 bytes)
CREATE FruverFE/.gitignore (348 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.

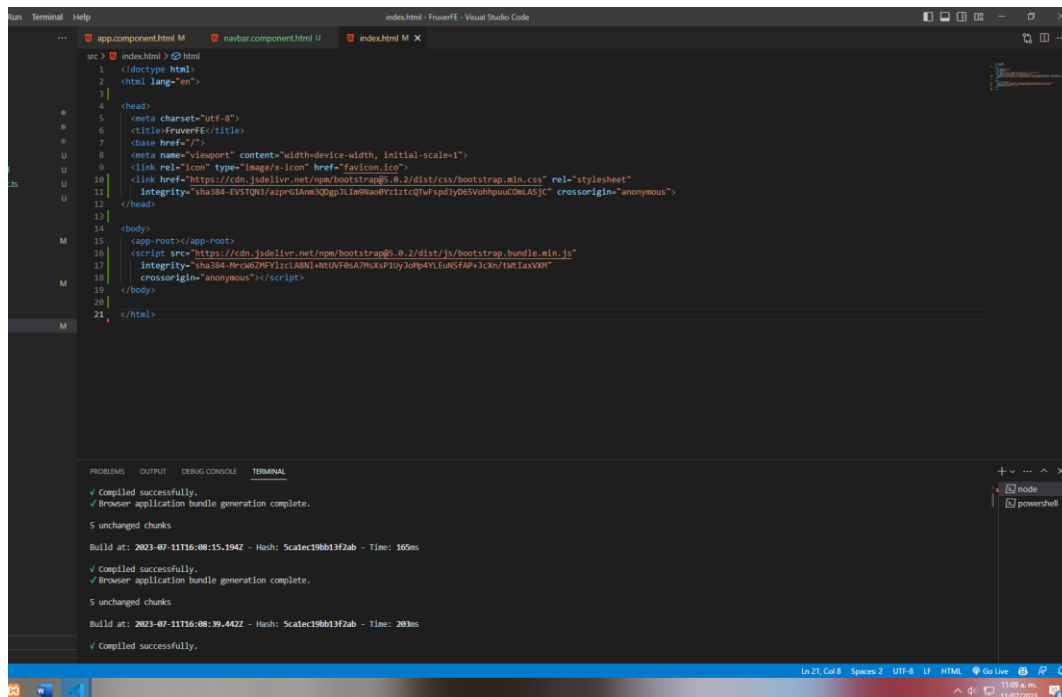


The screenshot shows a Visual Studio Code editor with a file explorer on the left displaying a project structure with files like 'app.json', 'son', and 'spec.json'. The main editor area shows an SVG file with various graphical elements defined using XML-like tags. Below the editor, a terminal window is open, showing the command 'PS C:\Users\Usuario\Desktop\Nueva carpeta (2)\fruerFE> ng serve --open' being executed. The status bar at the bottom indicates the file is 'HTML' and the editor is in 'UTF-8' encoding.

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 displays the 'Quick start' page of the Bootstrap documentation. On the left, there is a navigation menu with links to 'Contents', 'Browsers & devices', 'JavaScript', 'Build tools', 'Webpack', 'Parcel', 'Accessibility', 'RFS', 'RTL', 'Customize', 'Layout', 'Content', 'Forms', 'Components', 'Helpers', 'Utilities', 'Extend', 'About', and 'Migration'. The main content area is titled 'Quick start' and includes a sub-header 'CSS'. Below this, it instructs users to copy-paste the stylesheet link into their <head> before all other stylesheets. A code block shows the link: <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">. To the right of the code block is a 'Copy' button. Below the CSS section, there is a 'JS' section with a sub-header 'Bundle'. It explains that every Bootstrap JavaScript plugin and dependency is included in one of two bundles: 'bootstrap.bundle.js' and 'bootstrap.bundle.min.js'. A code block shows the script tag: <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="...". To the right of the code block is another 'Copy' button. Below the JS section, there is a 'Separate' section with a sub-header 'Separate'. It explains that if you decide to go with the separate scripts solution, Popper must come first (if you're using tooltips or popovers), and then our JavaScript plugins. On the right side of the page, there is a vertical list of links: 'Bundle', 'Separate', 'Modules', 'Components', 'Starter template', 'Important globals', 'HTML5 doctype', 'Responsive meta tag', 'Box-sizing', 'Reboot', and 'Community'. The bottom of the page shows a Windows taskbar with the time '10:58 a.m.' and the date '11/07/2023'.



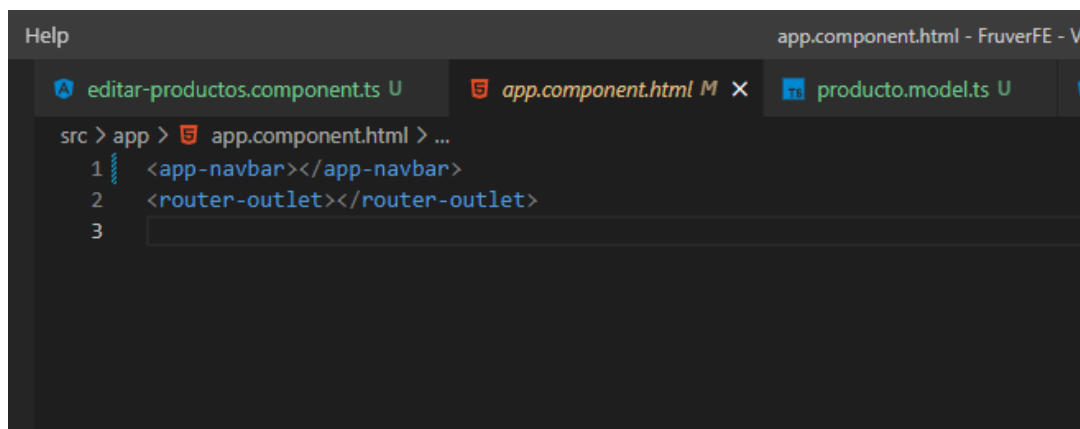
The screenshot shows the Visual Studio Code editor with the `index.html` file open. The file contains the following HTML code:

```
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 3Lw9J9U9q95u9r9k89lT26Lh53506u60eL6f794466af47498" 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-M6Wt0F0Wdb3pIzZyXpe48Zf22h8QV8QCuq06gmT41uX19Lz+tdA5X3N80e9864" crossorigin="anonymous"></script>
18 </body>
19 </html>
```

The terminal at the bottom shows the output of the build process:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
✓ Compiled successfully.
✓ Browser application bundle generation complete.
5 unchanged chunks
Build at: 2023-07-11T16:08:15.194Z - Hash: ScateC198613f2ab - Time: 165ms
✓ Compiled successfully.
✓ Browser application bundle generation complete.
5 unchanged chunks
Build at: 2023-07-11T16:08:39.442Z - Hash: ScateC198613f2ab - Time: 203ms
✓ 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:

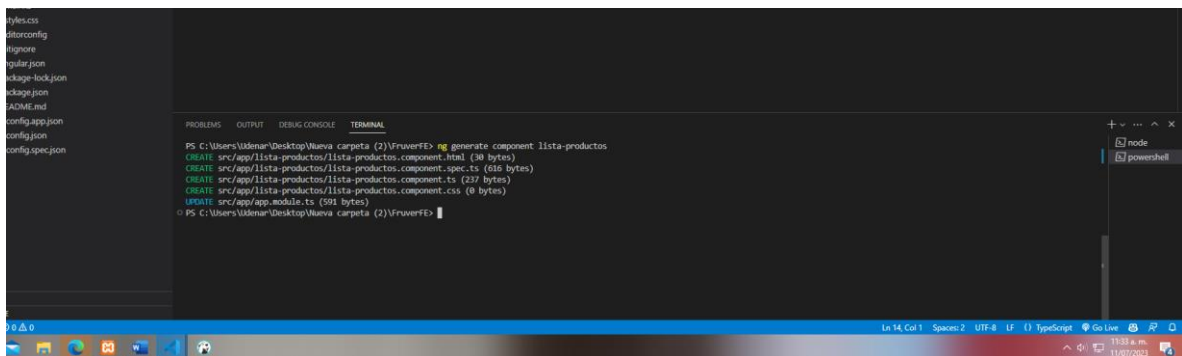
```
src > app > app.component.html > ...
1 <app-navbar></app-navbar>
2 <router-outlet></router-outlet>
3
```

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

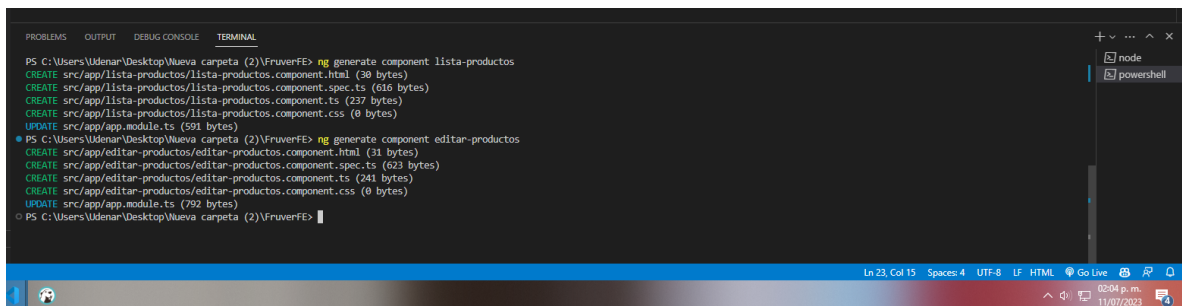
- Lista productos.
- Lista pedidos.
- Carrito.
- Detalle producto.
- Editar productos.
- Iniciar sesión.
- Navbar.

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.



```
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)\FruverFE> 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)\FruverFE>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FruverFE> 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)\FruverFE> 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)\FruverFE>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FruverFE> 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 (124 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FruverFE> 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 (262 bytes)
CREATE src/app/shared/navbar/navbar.component.css (0 bytes)
UPDATE src/app/app.module.ts (902 bytes)
PS C:\Users\Udenar\Desktop\Vueva carpeta (2)\FruverFE>
```

La estructura del proyecto quedará de la siguiente forma:

The screenshot shows the Visual Studio Code interface with the Explorer view on the left, the Editor view in the center, and the Output view at the bottom.

Project Structure (Explorer):

- FRUVERFE
 - .angular
 - .vscode
 - node_modules
 - src
 - app
 - carrito
 - detalle-producto
 - editar-productos
 - inicio-sesion
 - lista-pedidos
 - lista-productos
 - shared
 - navbar
 - navbar.component.css
 - navbar.component.html
 - navbar.component.spec.ts
 - navbar.component.ts
 - services
 - cliente.service.spec.ts
 - cliente.service.ts
 - login.service.spec.ts
 - login.service.ts
 - pedido-producto.service.spec.ts
 - pedido-producto.service.ts
 - pedido.service.spec.ts
 - pedido.service.ts
 - producto.service.spec.ts
 - producto.service.ts
 - cliente.model.ts
 - pedido.model.ts
 - pedidoProducto.model.ts
 - producto.model.ts
 - app-routing.module.ts
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - favicon.ico

Editor View: The file `navbar.component.html` is open, showing the following HTML structure:

```
1 <nav class="navbar navbar-expand-lg navbar-light">
2   <div class="container-fluid">
3     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
4       aria-controls="navbarSupportedContent" data-bs-key="navbarSupportContent" data-bs-clos="navbarSupportContent"
5     <span class="navbar-toggler-icon"></span>
6   </button>
7   <a class="navbar-brand" [routerLink]="['/']"></a>
8   <div id="navbarSupportContent" class="collapse navbar-collapse">
9     <ul class="navbar-nav mb-2 mb-lg-0">
10      <li class="nav-item">
11        <a class="nav-link" href="#">Inicio</a>
12      </li>
13      <li class="nav-item">
14        <a class="nav-link" href="#">Productos</a>
15      </li>
16      <li class="nav-item">
17        <a class="nav-link" href="#">Pedidos</a>
18      </li>
19      <li class="nav-item">
20        <a class="nav-link" href="#">Clientes</a>
21      </li>
22    </ul>
23    <ul class="navbar-nav mb-2 mb-lg-0">
24      <form class="d-flex">
25        <input type="text" class="form-control">
26        <button type="button" class="btn btn-light">Buscar</button>
27      </form>
28    </ul>
29  </div>
30 </div>
31 </nav>
```

Output View: The terminal shows the following output:

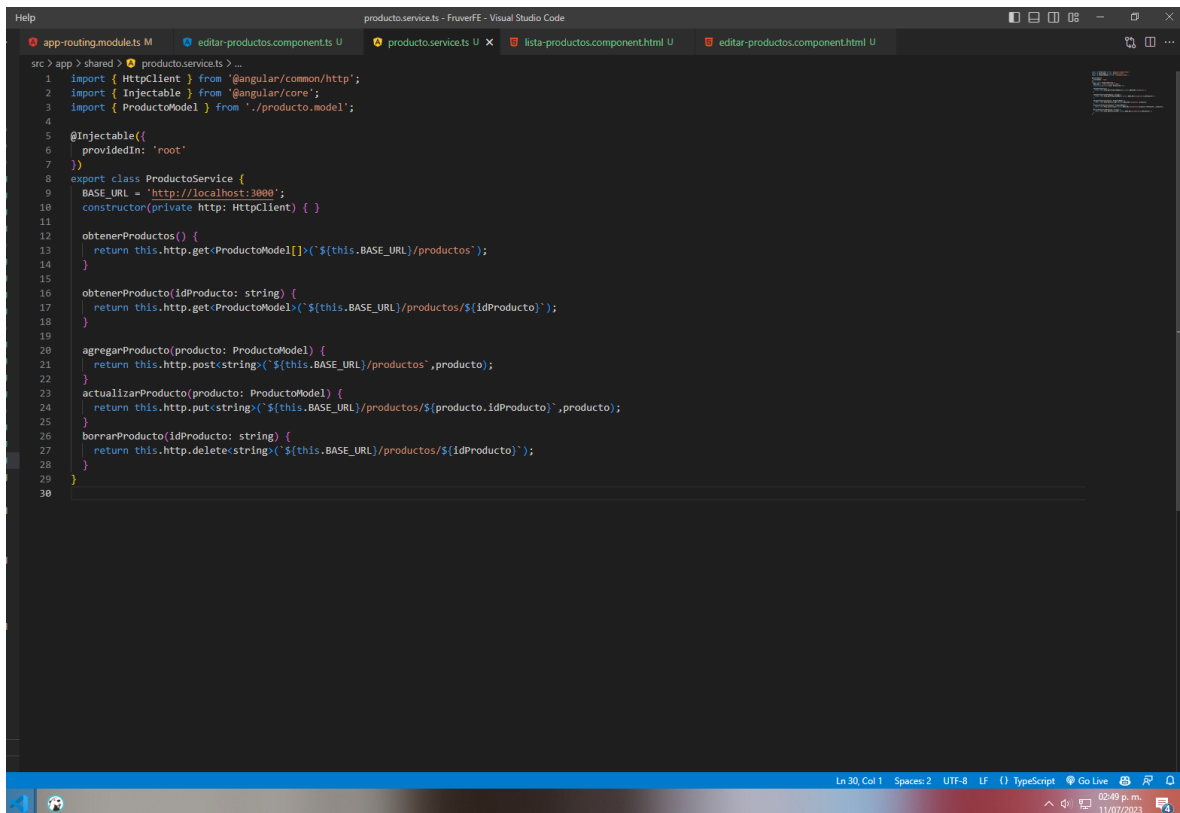
```
Failed to compile.
Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
main.js             | main  | 100.62 kB |
runtime.js          | runtime | 6.51 kB |

3 unchanged chunks

Build at: 2023-07-17T22:41:27.287Z - Hash: 002607e8
Compiled successfully.
```

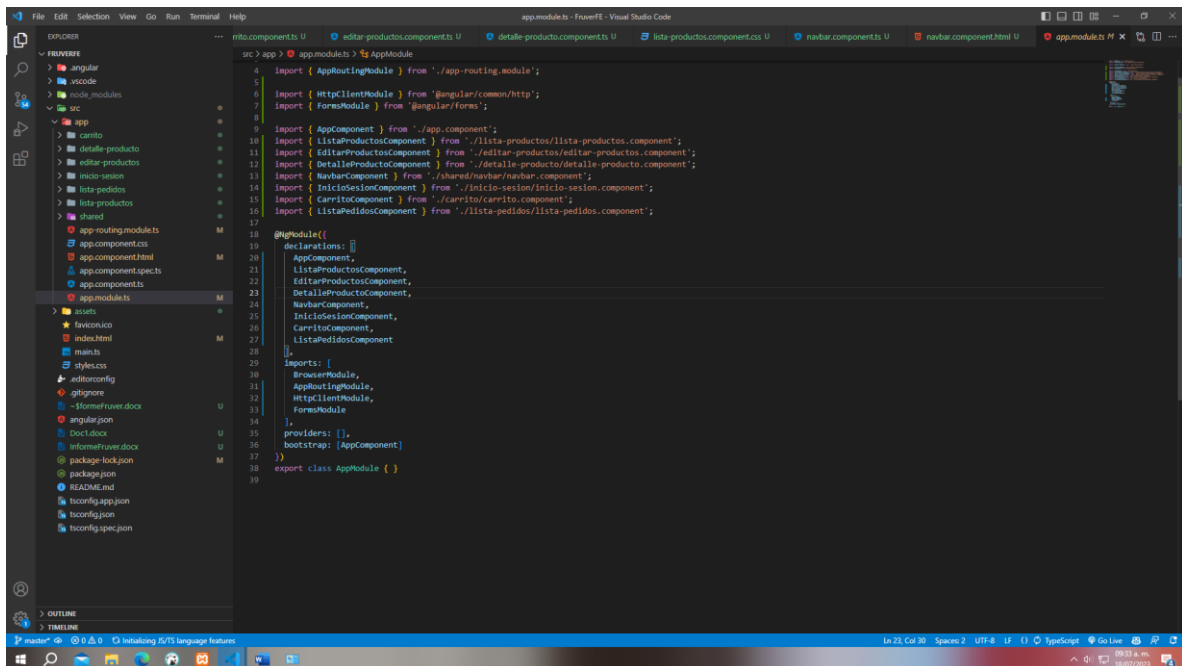
Producto service tiene los métodos para la gestión de los datos de los productos. Se encarga de realizar las peticiones para crear, actualizar, obtener y eliminar productos.

A screenshot of the Visual Studio Code editor interface. The title bar at the top reads "producto.service.ts - FroverFE - Visual Studio Code". The editor has several tabs open: "app-routing.module.ts M", "editar-productos.component.ts U", "producto.service.ts U" (which is the active tab), "lista-productos.component.html U", and "editar-productos.component.html U". The active file, "producto.service.ts", contains the following TypeScript code:

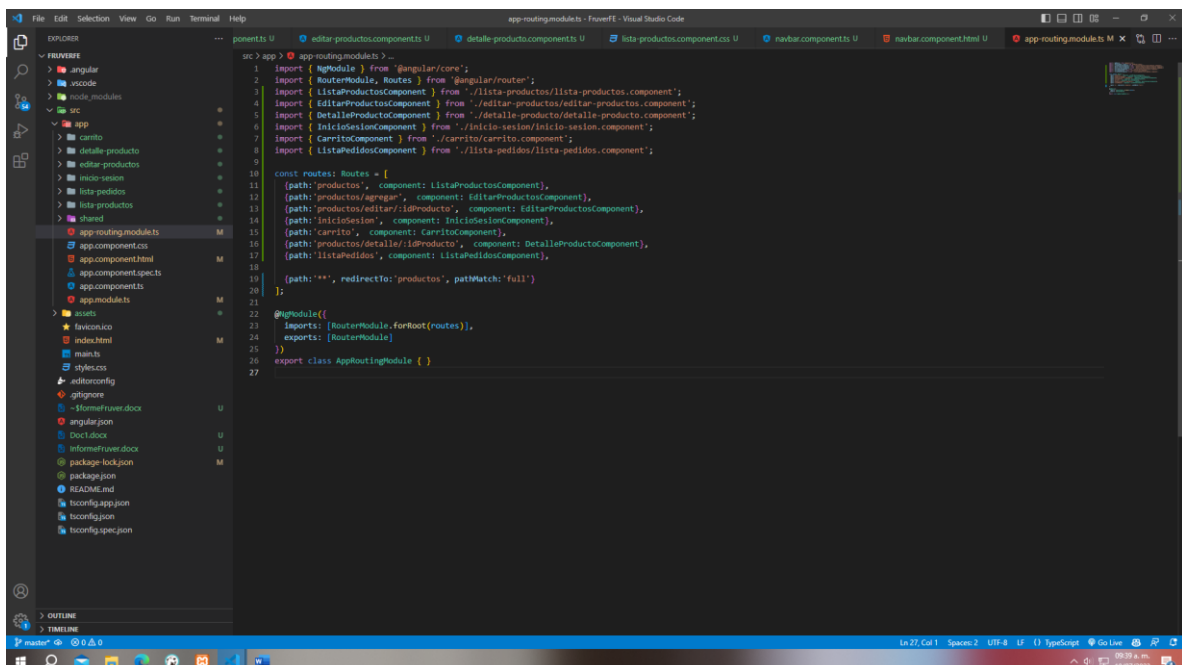
```
src > app > shared > producto.service.ts > ...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { ProductoModel } from '../producto.model';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class ProductoService {
9   BASE_URL = 'http://localhost:3880';
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
24   actualizarProducto(producto: ProductoModel) {
25     return this.http.put<string>(`${this.BASE_URL}/productos/${producto.idProducto}`, producto);
26   }
27
28   borrarProducto(idProducto: string) {
29     return this.http.delete<string>(`${this.BASE_URL}/productos/${idProducto}`);
30   }
31 }
```

The status bar at the bottom indicates "Ln 30, Col 1", "Spaces: 2", "UTF-8", "LF", "TypeScript", and "Go Live". The system tray at the bottom right shows the time as "02:49 p. m." and the date as "11/07/2023".

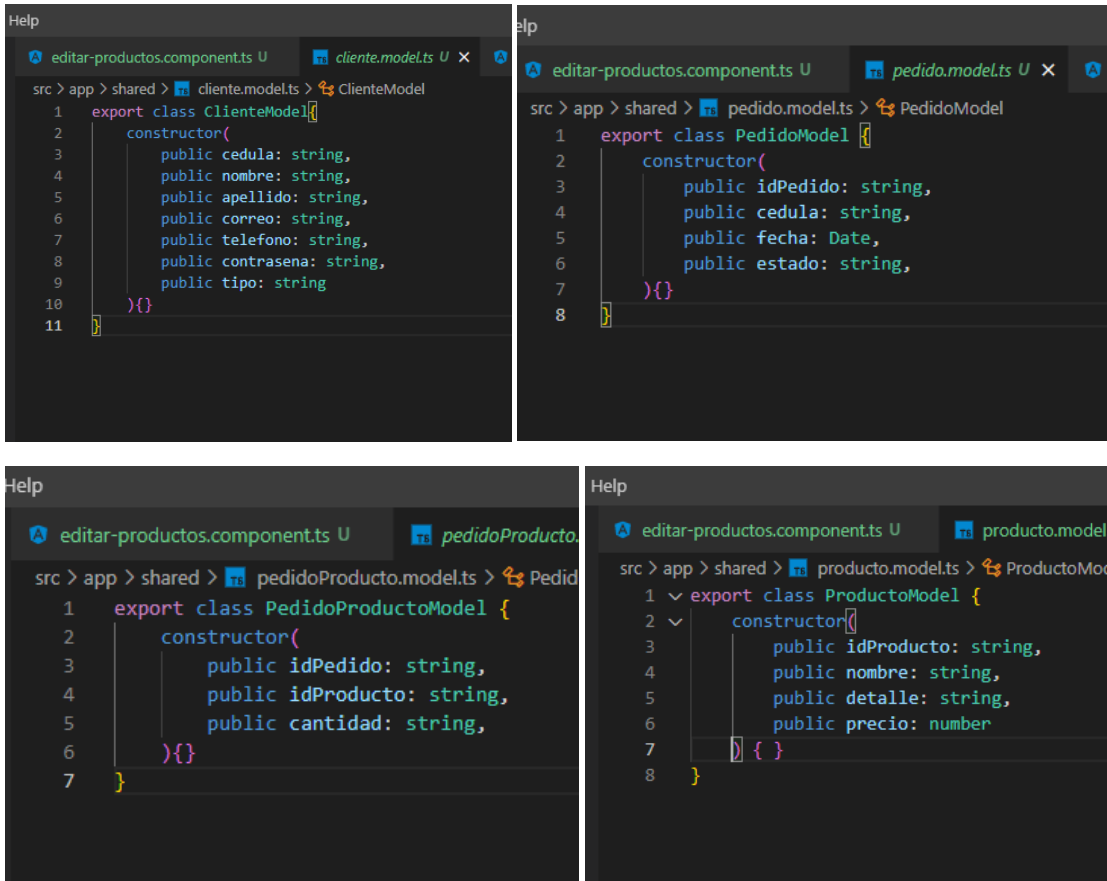
Todos los componentes que se hayan creado o se crearan deben importarse en el archivo app.module.ts



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



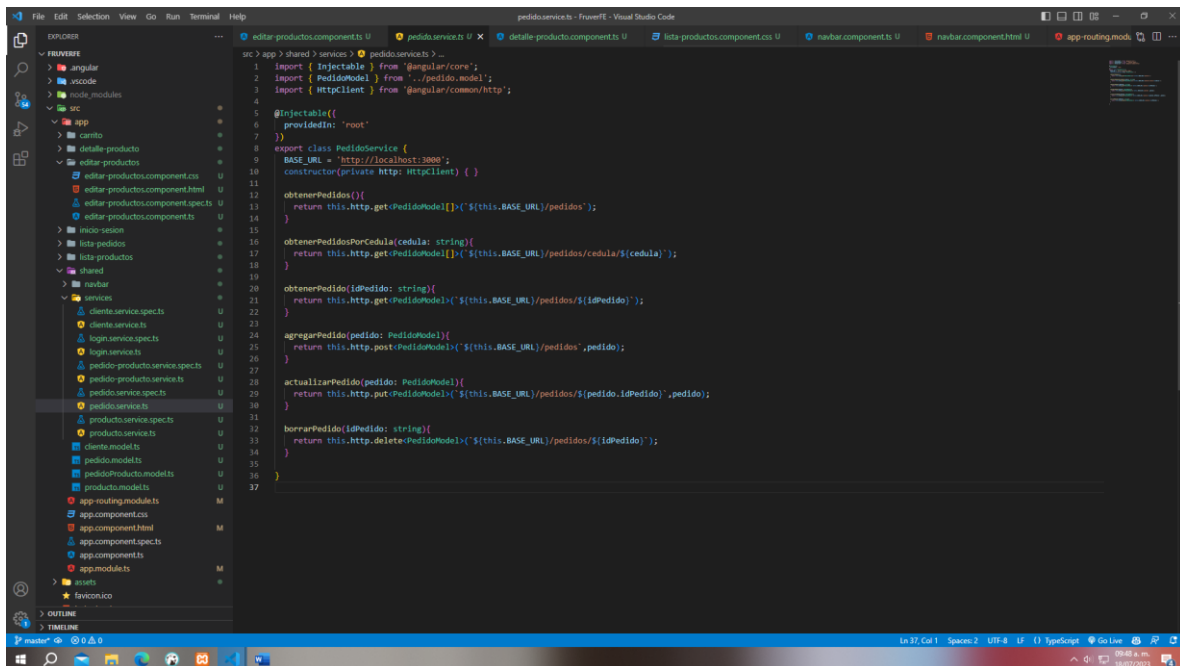
Se crean modelos de productos, pedido, pedidoProducto y cliente que representan las tablas de la base de datos.



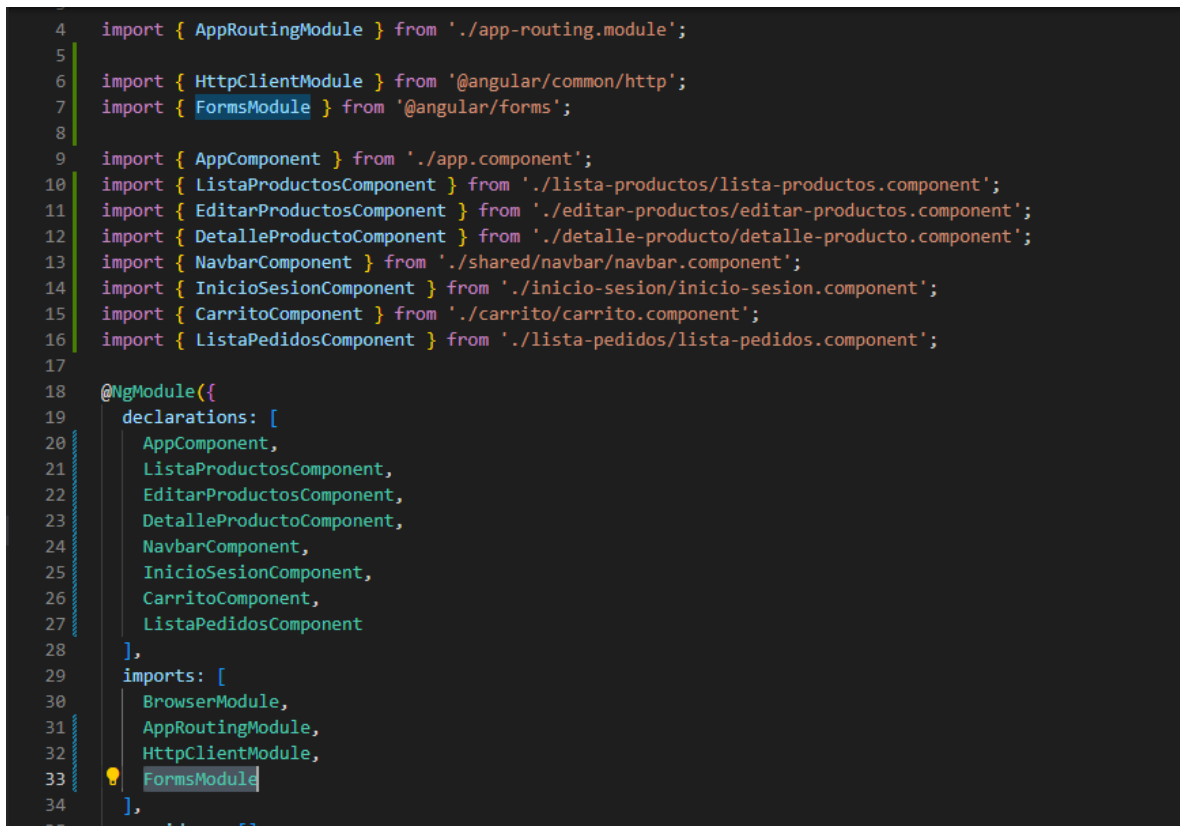
The image displays four separate screenshots of a code editor (VS Code) showing the creation of TypeScript model classes. Each screenshot includes a 'Help' sidebar on the left and a breadcrumb navigation path at the top of the editor area.

- Top Left:** The file is `cliente.model.ts` located at `src > app > shared > cliente.model.ts`. It defines the `ClienteModel` class with a constructor that takes `cedula`, `nombre`, `apellido`, `correo`, `telefono`, `contrasena`, and `tipo` as strings.
- Top Right:** The file is `pedido.model.ts` located at `src > app > shared > pedido.model.ts`. It defines the `PedidoModel` class with a constructor that takes `idPedido` (string), `cedula` (string), `fecha` (Date), and `estado` (string).
- Bottom Left:** The file is `pedidoProducto.model.ts` located at `src > app > shared > pedidoProducto.model.ts`. It defines the `PedidoProductoModel` class with a constructor that takes `idPedido` (string), `idProducto` (string), and `cantidad` (string).
- Bottom Right:** The file is `producto.model.ts` located at `src > app > shared > producto.model.ts`. It defines the `ProductoModel` class with a constructor that takes `idProducto` (string), `nombre` (string), `detalle` (string), and `precio` (number).

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.



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.

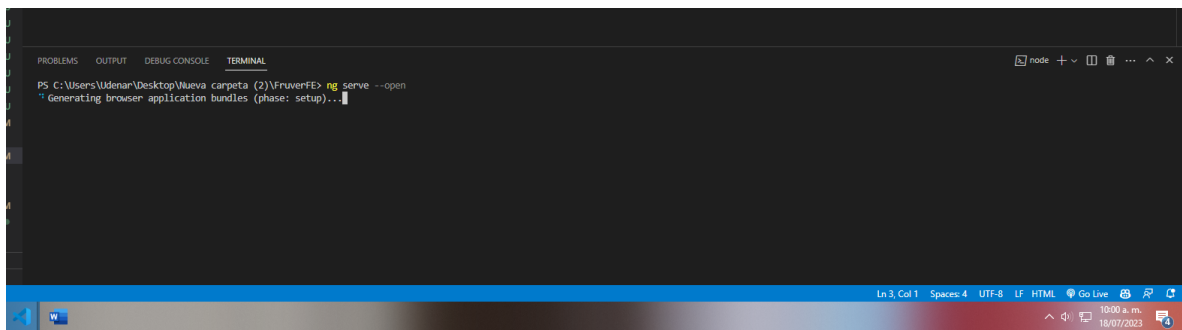
```
editar-productos.component.ts - FruverFE - Visual Studio Code

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 > EditarProductosComponent > 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 EditarProductosComponent 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 }
```

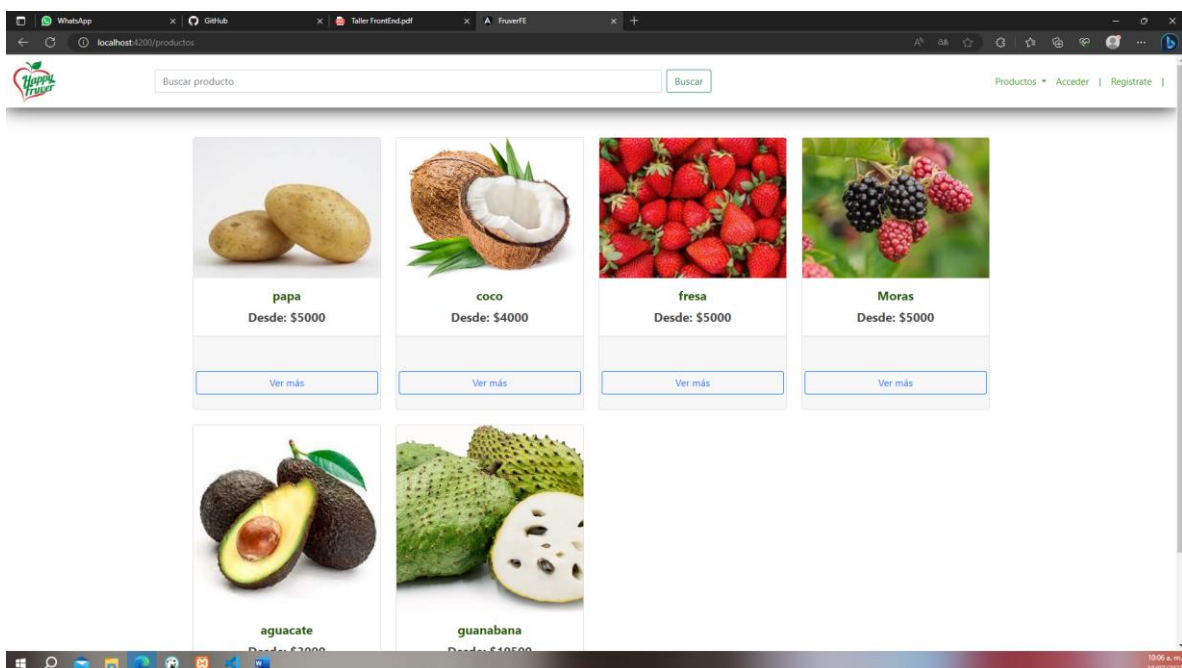
```
46
47 // funcion para crear o editar un producto
48 onSubmit() {
49   console.log("Submit realizado");
50   if (this.producto.idProducto) {
51     //Viene de Editar
52     this.productoService.actualizarProducto(this.producto).subscribe(
53       data => {
54         console.log(data);
55         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 }
```

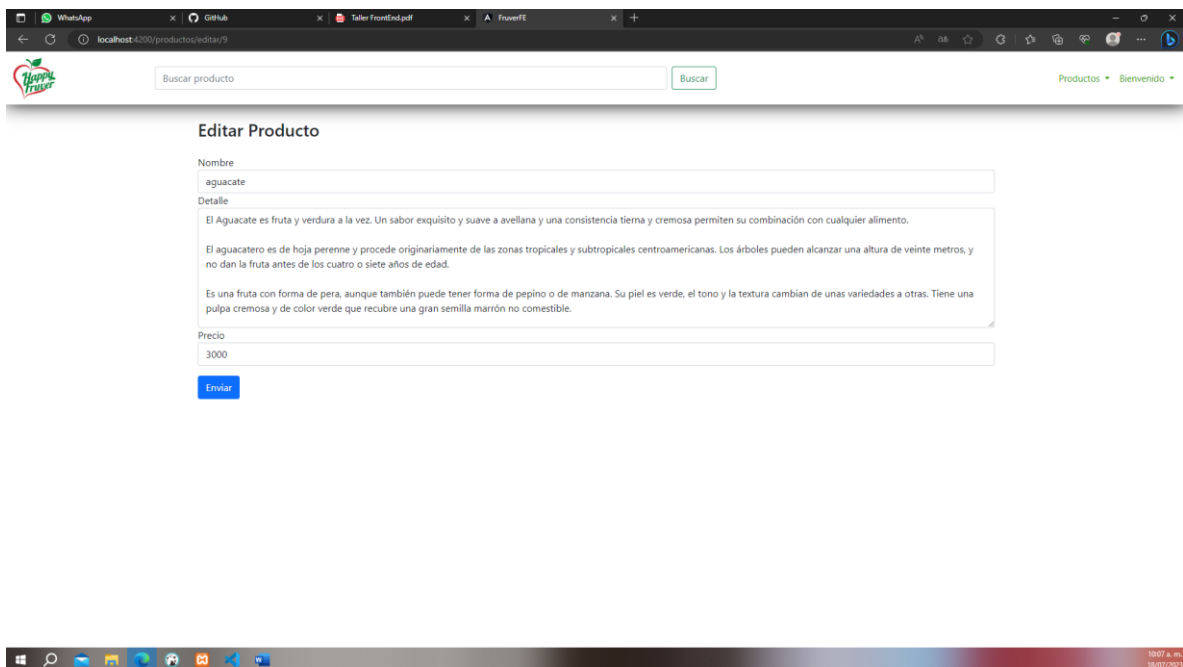
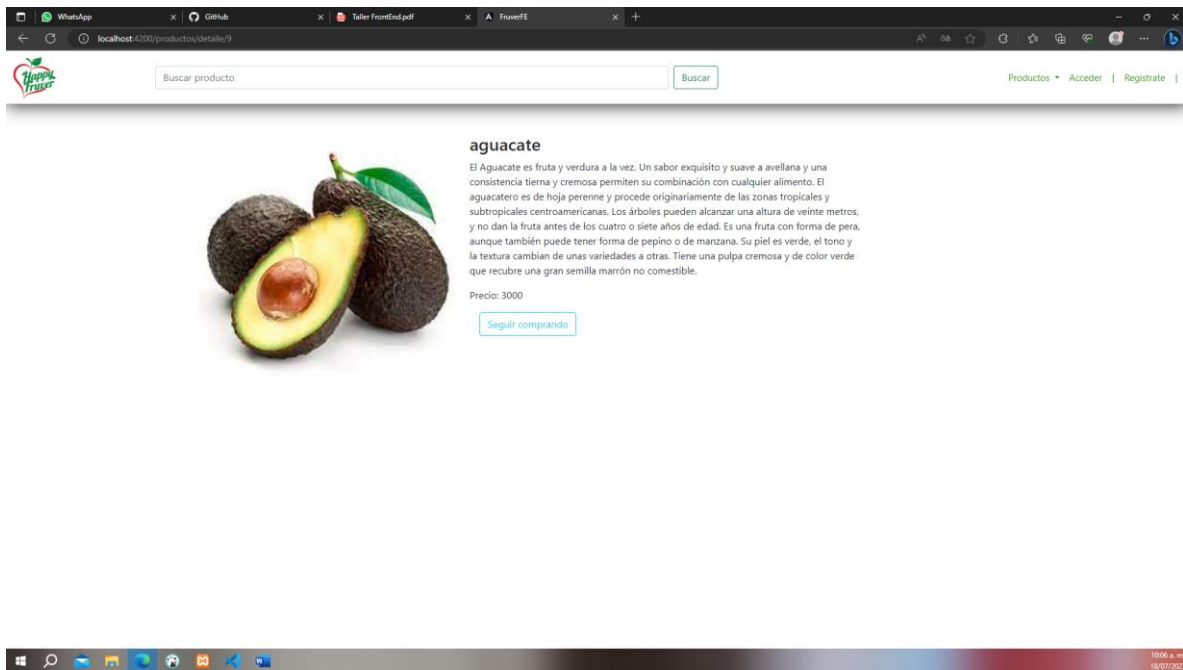
Con el comando `ng serve --open` iniciamos nuestro proyecto



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\Udenar\Desktop\Nueva carpeta (2)\FruterFE> ng serve --open
* generating browser application bundles (phase: setup)...
```

Resultado





Crear Producto

Nombre

Detalle

Precio