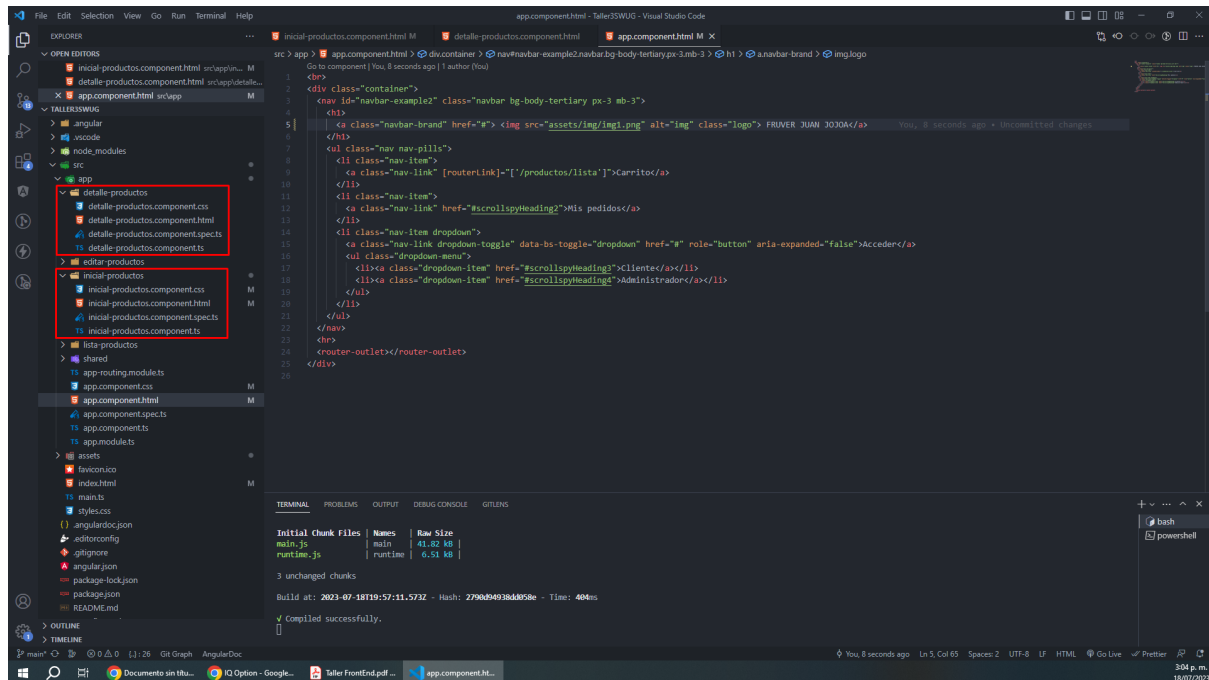
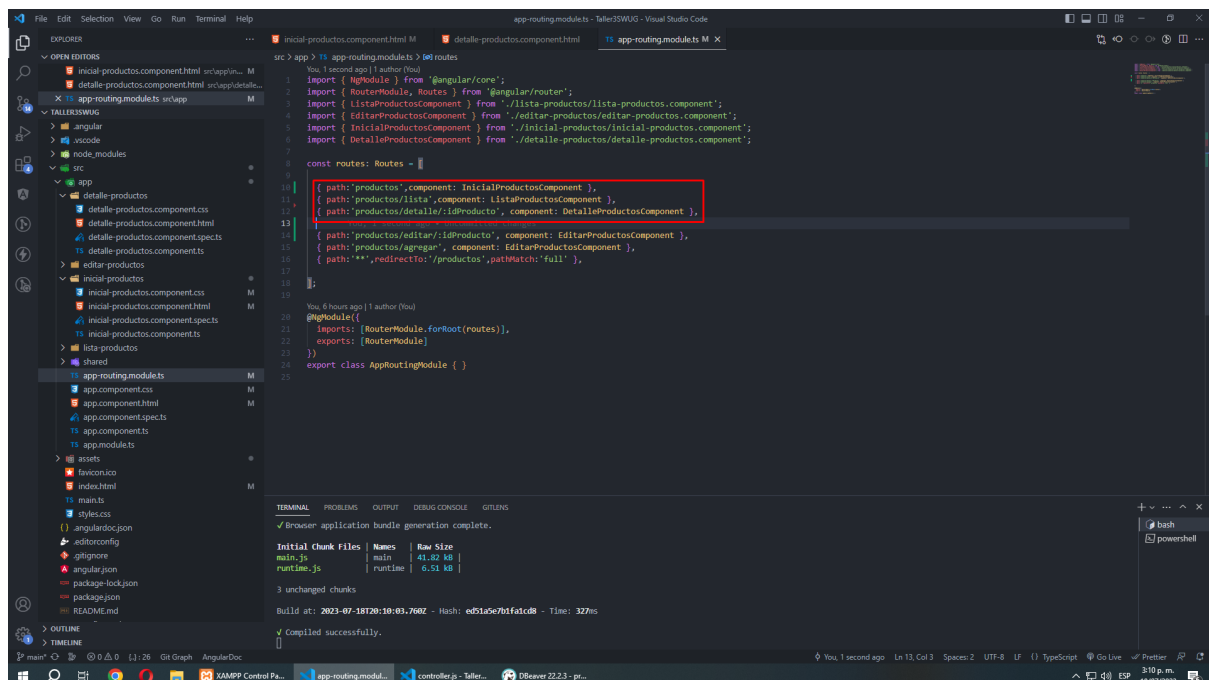


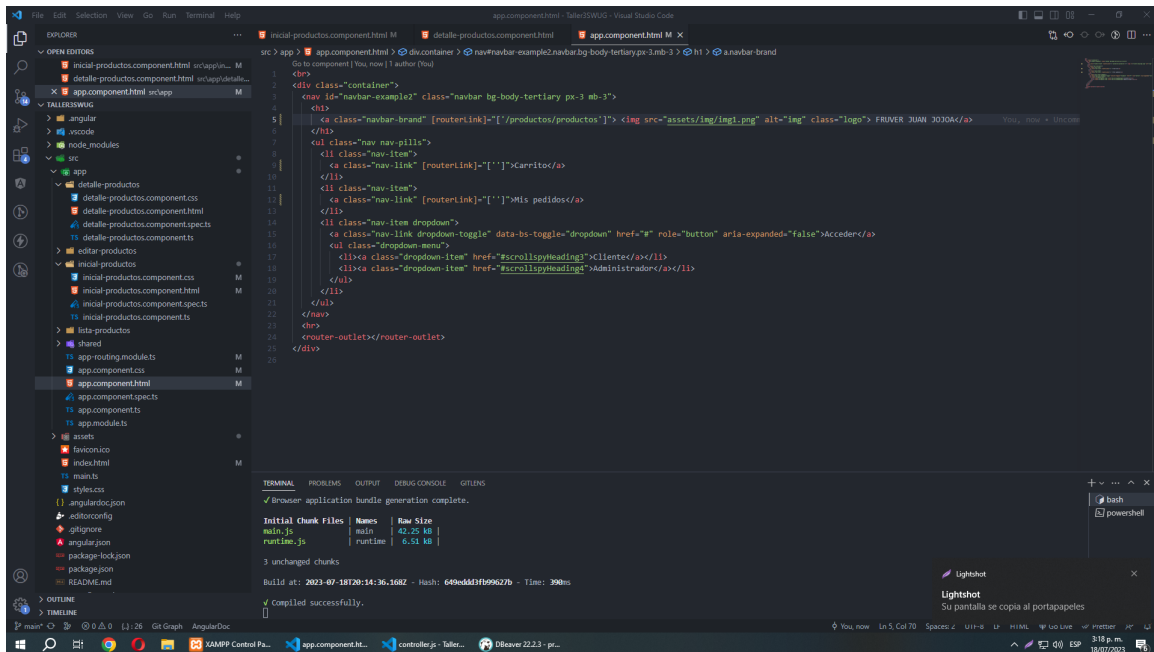
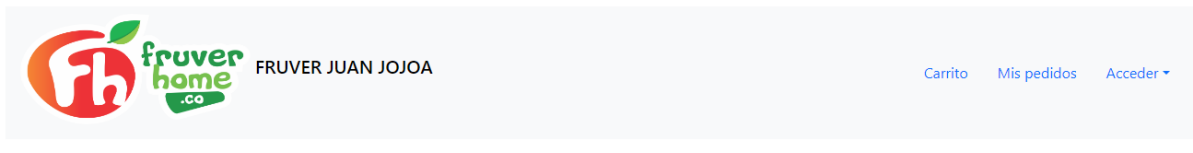
Utilizando el framework angular creamos los componentes respectivos para las 2 vistas que vamos a desarrollar usando el comando *ng generate component*



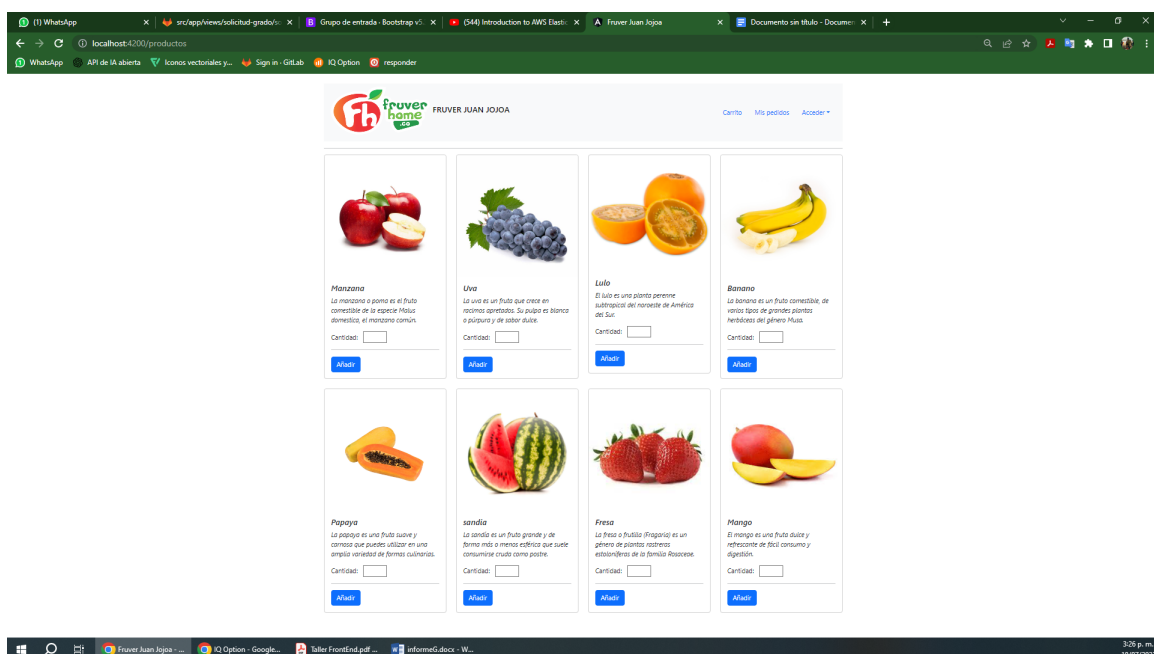
En el archivo *app-routing.module* agregamos las rutas de los 2 nuevos componentes que creamos y definimos el *component: InicialProductosComponent* como predeterminado



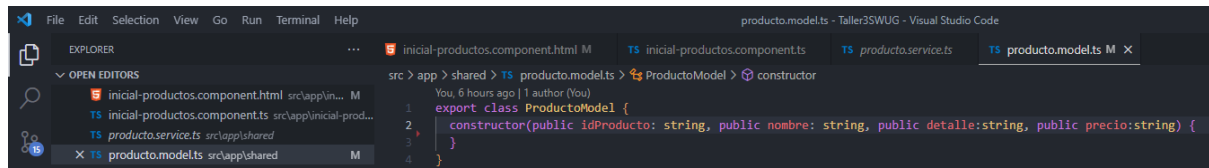
Creamos la navbar en el archivo `app.component` para que esta aparezca en todas las páginas a las que nos dirijamos y agregamos un `[routerLink]` al logo para poder regresar a la página inicial cuando lo necesitemos



Desarrollamos la vista inicial que contendrá los productos almacenados en nuestra base de datos anteriormente creada

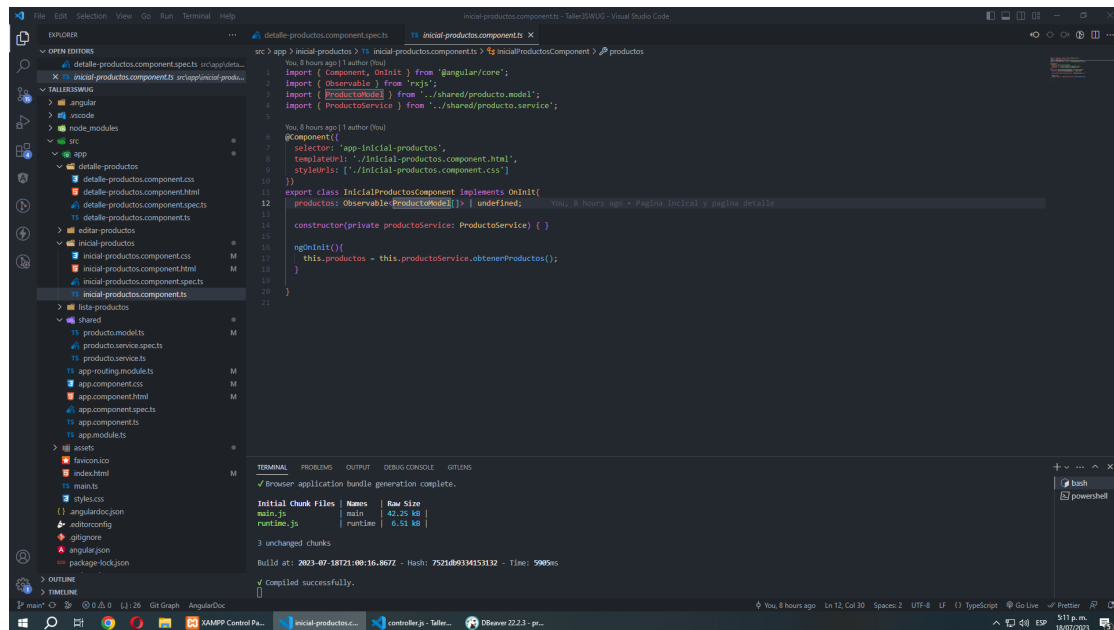


Para ello usaremos un modelo con el cual traeremos todos los datos que necesitamos de los diferentes productos



```
1 You, 6 hours ago | 1 author (You)
2 export class ProductModel {
3   constructor(public idProducto: string, public nombre: string, public detalle:string, public precio:string) {
4   }
5 }
```

Llamamos el modelo desde *inicial-productos.components.ts* y hacemos uso del servicio

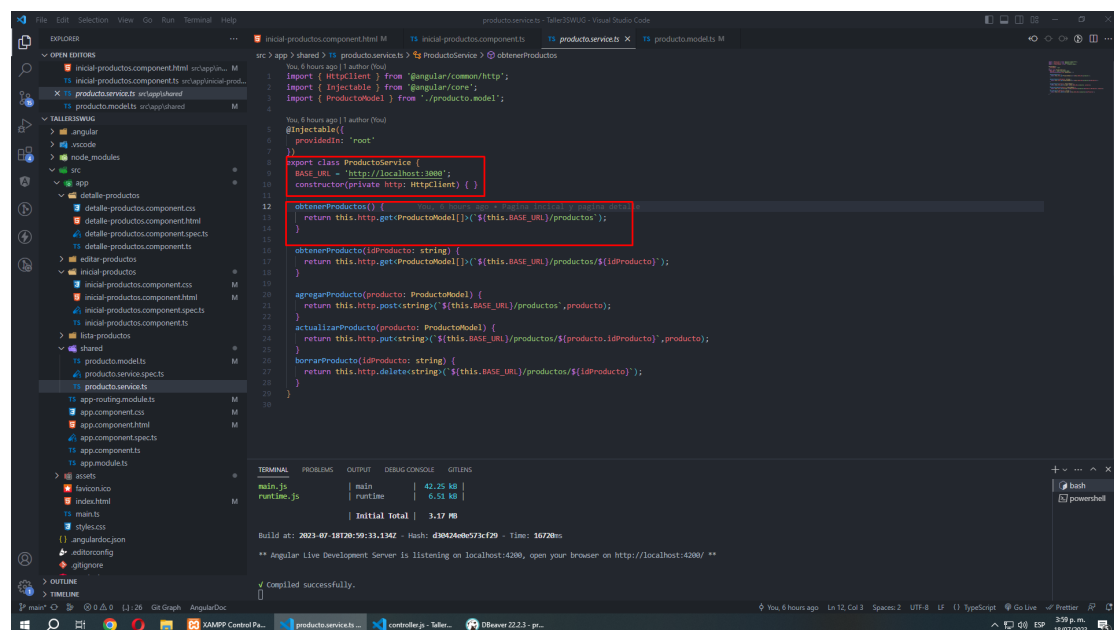


```
1 You, 8 hours ago | 1 author (You)
2 import { Component, OnInit } from '@angular/core';
3 import { Observable } from 'rxjs';
4 import { ProductModel } from '../shared/producto.model';
5 import { ProductService } from '../shared/producto.service';
6
7 @Component({
8   selector: 'app-inicial-productos',
9   templateUrl: './inicial-productos.component.html',
10   styleUrls: ['./inicial-productos.component.css']
11 })
12 export class InicialProductosComponent implements OnInit {
13   productos: Observable<ProductModel> | undefined;
14   constructor(private productService: ProductService) {}
15
16   ngOnInit() {
17     this.productos = this.productService.obtenerProductos();
18   }
19 }
20
21
```

Initial Chunk Files	Names	Raw Size
main.js	main	42.25 kb
runtime.js	runtime	6.51 kb

Build at: 2023-07-18T21:08:16.867Z - Hash: 7521d8934151312 - Time: 590ms

Haremos uso del *ProductoService* que se encuentra en el archivo *producto.service.ts*



```
1 You, 6 hours ago | 1 author (You)
2 import { HttpClient } from '@angular/common/http';
3 import { Injectable } from '@angular/core';
4 import { ProductModel } from '../producto.model';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class ProductService {
10   private http = http://localhost:4200;
11   constructor(private http: HttpClient) {}
12
13   obtenerProductos() {
14     return this.http.get<ProductModel[]>(`${this.BASE_URL}/productos`);
15   }
16
17   obtenerProducto(idProducto: string) {
18     return this.http.get<ProductModel[]>(`${this.BASE_URL}/productos/${idProducto}`);
19   }
20
21   agregarProducto(producto: ProductModel) {
22     return this.http.post<string>(`${this.BASE_URL}/productos', producto);
23   }
24
25   actualizarProducto(producto: ProductModel) {
26     return this.http.put<string>(`${this.BASE_URL}/productos/${producto.idProducto}', producto);
27   }
28
29   borrarProducto(idProducto: string) {
30     return this.http.delete<string>(`${this.BASE_URL}/productos/${idProducto}`);
31   }
32 }
33
```

Initial Total	Initial Total
3.17 MB	

Build at: 2023-07-18T20:39:33.134Z - Hash: d30424e0b573cf29 - Time: 1672ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Posteriormente crearemos la vista con las diferentes *card* de los productos que a su vez se crean usando un `*ngFor` para cada producto recuperado de la base de datos y le agregamos un `[routerLink]` a cada imagen de los productos que serán el medio por el cual nos dirigiremos a los detalles de cada uno.

En el `[routerLink]` debemos recuperar el id de cada producto y enviarlo junto a la ruta para que cuando vayamos a mostrar el detalle lo haga simplemente del producto que seleccionamos, el `[routerLink]` tendrá la ruta a *component*: *DetalleProductosComponent* que se explicará más adelante

```

1  <!-- detalle-productos.component.html -->
2  <!-- DOCTYPE html -->
3  <html lang="en">
4
5  <head>
6    <meta charset="UTF-8">
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <title>Document</title>
9  </head>
10
11 <body>
12   <div class="row">
13     <div class="col-sm-3" *ngFor="let producto of productos | async">
14       <div class="card">
15         <a [routerLink]="['/productos/detalle/', producto.idProducto]">
16           <img src='../assets/img/productos/{{producto.idProducto}}.jpg' class="card-img-top" alt="img">
17         </a>
18         <div class="card-body">
19           <h5 class="card-title">{{producto.nombre}}</h5>
20           <p class="card-text">{{producto.detalle}}</p>
21           <label for="Cantidad">Cantidad: </label> <input type="number" style="width: 60px;">
22           <br>
23           <a class="btn btn-primary">Añadir</a>
24         </div>
25       </div>
26     </div>
27   </div>
28 </body>
29 </html>

```

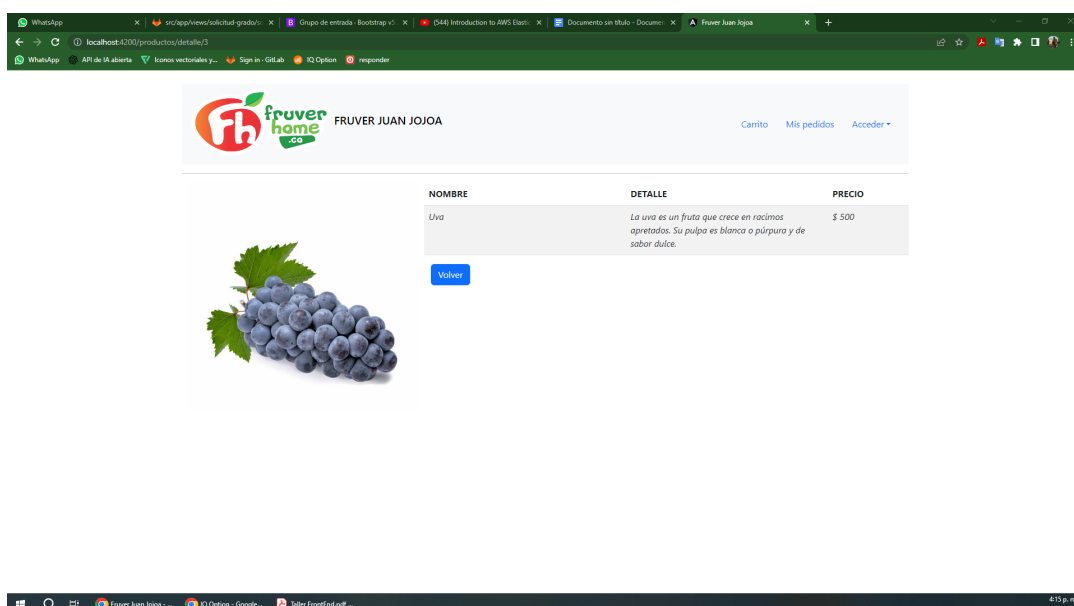
Terminal output:

```

✓ Browser application bundle generation complete.
Initial Chunk Files | Names | Raw Size
main.js             | main | 42.25 KB
runtime.js          | runtime | 6.51 KB
3 unchanged chunks
Build at: 2023-07-18T21:00:16.867Z - Hash: 7521d6934153132 - Time: 590ms
✓ compiled successfully.

```

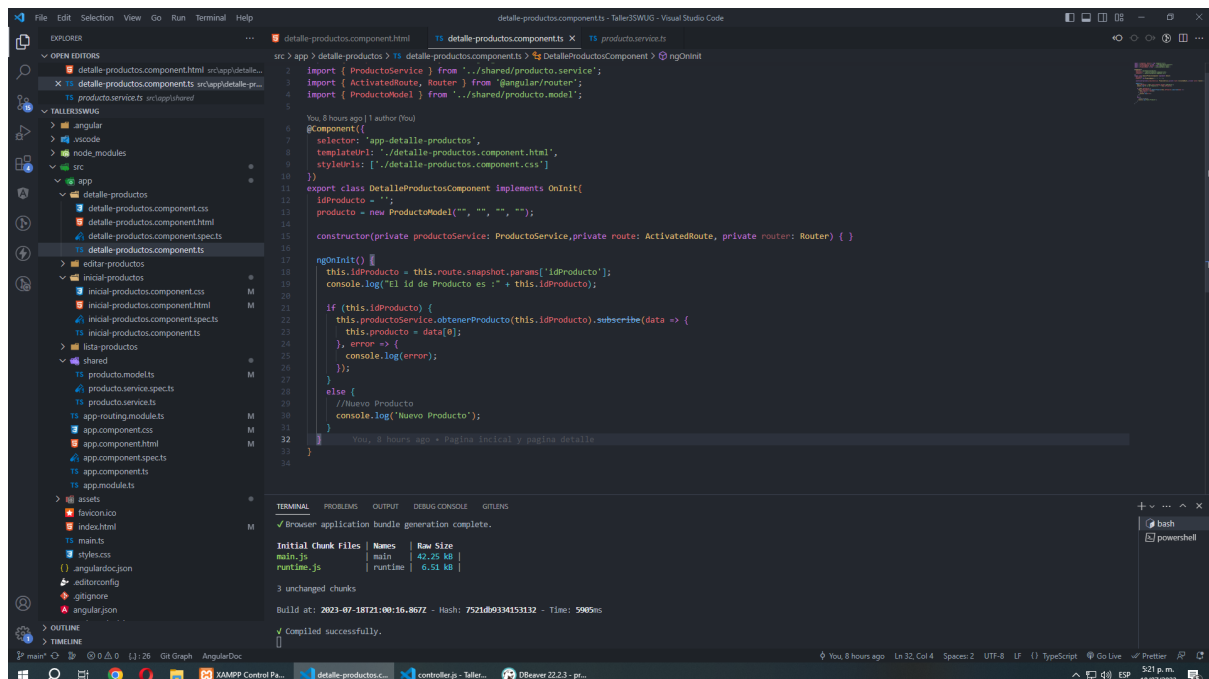
Desarrollamos la vista detalle para los productos de nuestra base de datos



En este módulo usaremos el *ProductoModel* mencionado anteriormente



Incorporaremos en el archivo *detalle-productos.component.ts* el uso del servicio *obtenerProducto* al cual le enviaremos el *idProducto* que recuperamos cuando le damos click a la imagen del producto, esto nos retornara el modelo del producto con los datos de ese *idProducto* para mostrar el detalle del producto especifico.



Para que esto funcione debemos crear la const *getProducto* en el controlador y crear la ruta en el Backend



```
JS controller.js JS routes.js X
Routes > JS routes.js > ...
38 //-----
39 router.get("/productos/:idProducto", getProducto);
40
```

En el archivo *detalle-productos.component.html* hacemos la estructura que tendrá la página del detalle de productos en la cual llamaremos los datos que necesitemos y que estarán almacenados en la variable *producto*, esta tendrá el modelo con los datos del producto seleccionado e identificado con el *idProducto*

