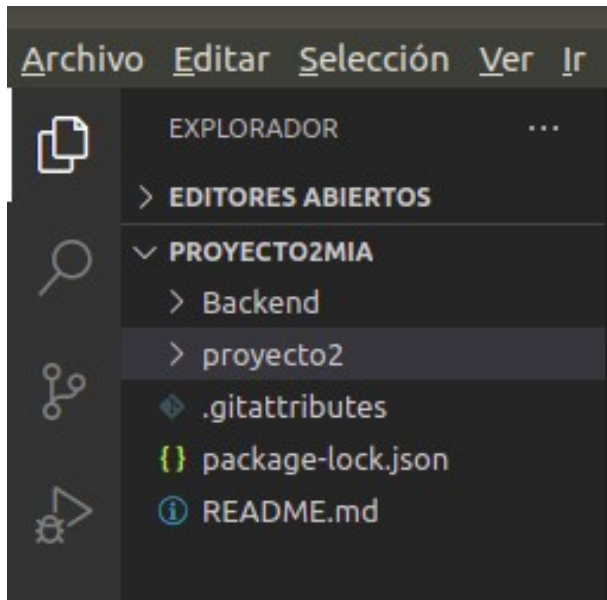


PROYECTO 2
GTSales Marketplace

Manual Técnico

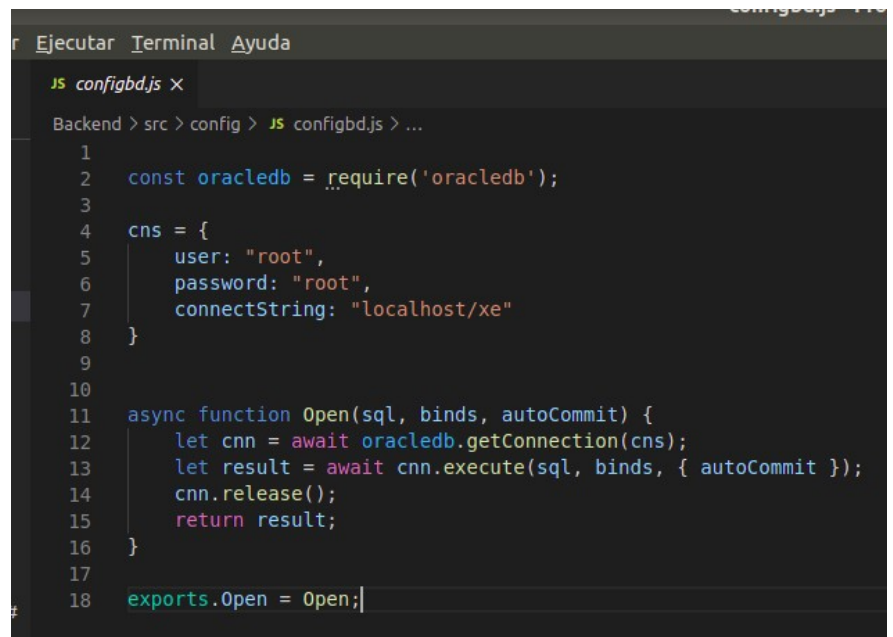
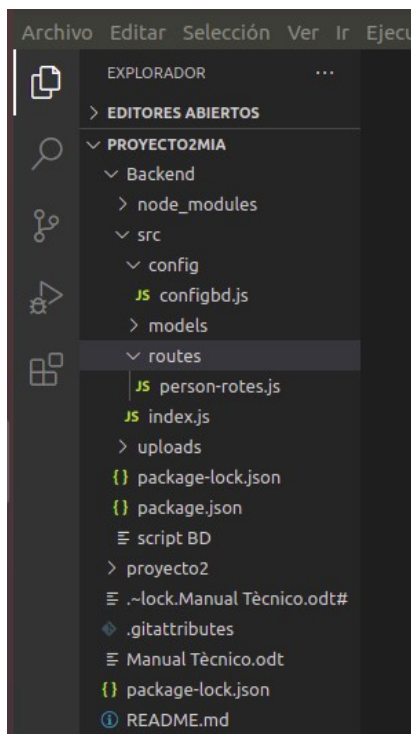


A continuacion se mostraran dos carpetas las cuales esta dividido el proyecto , la parte del Backend y la del Fronted



Creación de Backend en la Aplicación

En esta carpeta hay dos carpetas importantes las cuales son config y routes , que se mostraran a continuacion, la parte de config contiene la configuración para conectarse a la base de Datos creada en Oracle



por otra parte en el archivo de routes contiene todas las peticiones que iran hacia la base de datos para pedir registros. De los cuales contiene Insert, Delete, Update y Select. Podemos ver que cada seccion esta comentada según el tipo de peticion que se requiera... por ejemplo seccion de fotos, seccion de usuario... etc.

```
cutar Terminal Ayuda

person-roles.js x

backend > src > routes > JS person-roles.js > router.get('/api/reportes/componente/reporte/reporte/1/obtener') callback
1  const { Router } = require('express');
2  const router = Router();
3  const BD = require('../config/configbd');
4
5  const nodemailer = require('nodemailer');
6  ////////////////////////////////////////////////// Para las Fotos
7
8
9  const path = require('path');
10 const multer = require('multer');
11 const fs = require('fs');
12
13
14 const storage = multer.diskStorage({
15   //destination: path.join(__dirname, '../public/uploads'),
16   destination: 'uploads/',
17   filename: (req, file, cb) => {
18     cb(null, file.originalname);
19   }
20 })
21 const uploadImage = multer({
22   storage,
23   limits: {fileSize: 10000000} //10 megas
24 }).single('photo');
25
26
27 // RENDER FORM UPLOAD
28 router.get('/images/upload', (req, res) => {
29   res.render('index');
30 });
31
32 router.get('/images', (req, res) => {});
33 ////////////////////////////////////////////////// USUARIOS
34
35 //login solo traigo un registro
36 router.get("/api/usuario/:correo:pass", async (req, res) => {
37   const {pass}= req.params;
38   const {correo}= req.params;
39   console.log(pass)
40
41   sql = "select * from usuario where contrasenia =" + pass + " and correo =" + correo + " and confirmacion ='Co
42   // console.log(sql)
43
44   let result = await BD.Open(sql, [], false);
45   Users = [];
46
47   result.rows.map(user => {
48     let userSchema = {
```

tambien tenemos a continuacion la parte del archivo index.js es el que se va a ejecutar para correr nuestro servidor. Este archivo contiene la configuracion de nuestro servidor por ejemplo el puerto en que se va a correr, las peticiones que se le van a pedir etc. Esta comentado cada seccion de esta parte las cuales son imports, settings, middlewares, routes, static files y run.



```
Ejecutar Terminal Ayuda
JS index.js X
Backend > src > JS index.js > ...
1
2 const express = require('express');
3 const morgan = require('morgan');
4 const cors = require('cors');
5 const app = express();
6 const multer = require('multer');
7 const path = require('path');
8
9
10
11 //imports
12 const personRoutes = require('./routes/person-rotas');
13
14 //settings
15 app.set('port', 3000);
16
17
18 //middlewares
19 app.use(cors());
20 app.use(morgan('dev'));
21 app.use(express.json());
22 app.use(express.urlencoded({ extended: false }));
23
24
25
26 //routes
27 app.use(personRoutes);
28
29 // Static files
30 //app.use(express.static(path.join(__dirname, 'public')));
31 app.use('/uploads', express.static(path.resolve('uploads')));
32
33
34
35
36 //run
37 app.listen(app.get('port'), () => {
38   //console.log('Server on Port 3000')
39   console.log(`Server on port ${app.get('port')}`);
40 })
41
42
```

Para la otra Carpeta llamada proyecto2 este es mi Fronted. El cual contiene un archivo de instrucciones el cual contiene las instrucciones de como correr dicho proyecto..... algunas de ellas son:

```
//ejecutar este frontEnd con el siguiente comando, no se ejecuta con "ng serve" porque uso un  
proxy.conf.json para configurar mis peticiones hacia el backend, esto es para evitar un error de Cors  
npm start
```

```
//el Backend si lo ejecuto con  
npm run dev
```

Configuracion para generar una aplicación móvil con Capacitor

Implementación de capacitor y generación de la aplicación móvil

* Instalación de capacitor

```
```bash  
ng add @capacitor/angular
```
```

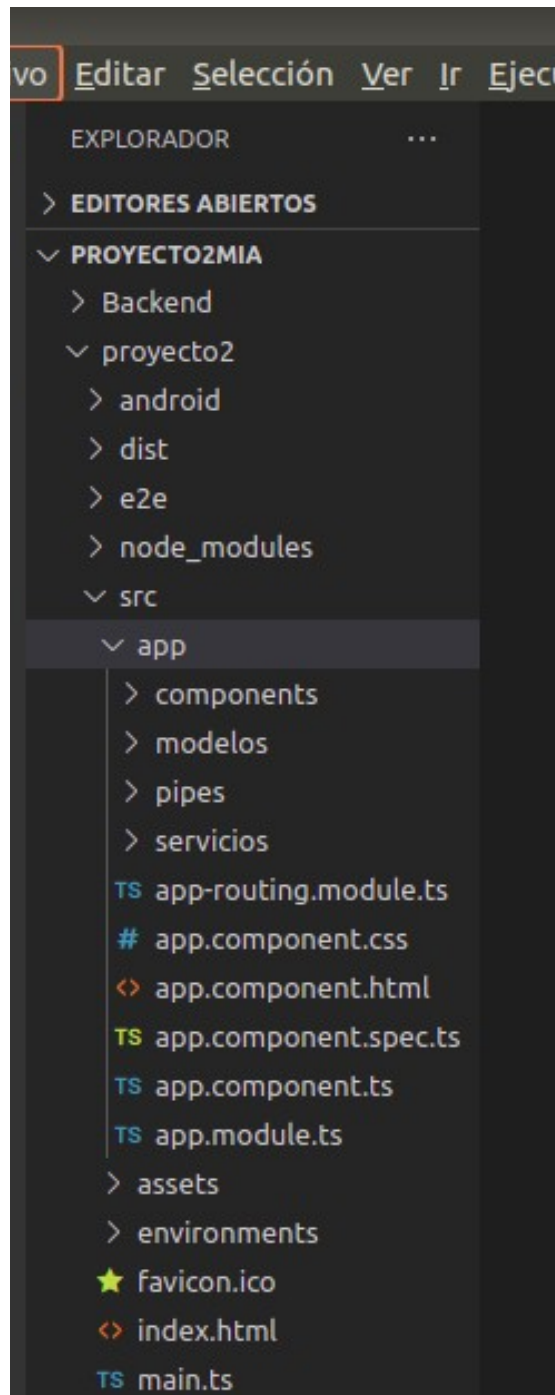
* Solucionar error 'spawn npx ENOENT' en Windows

```
```bash  
npx cap init
```
```

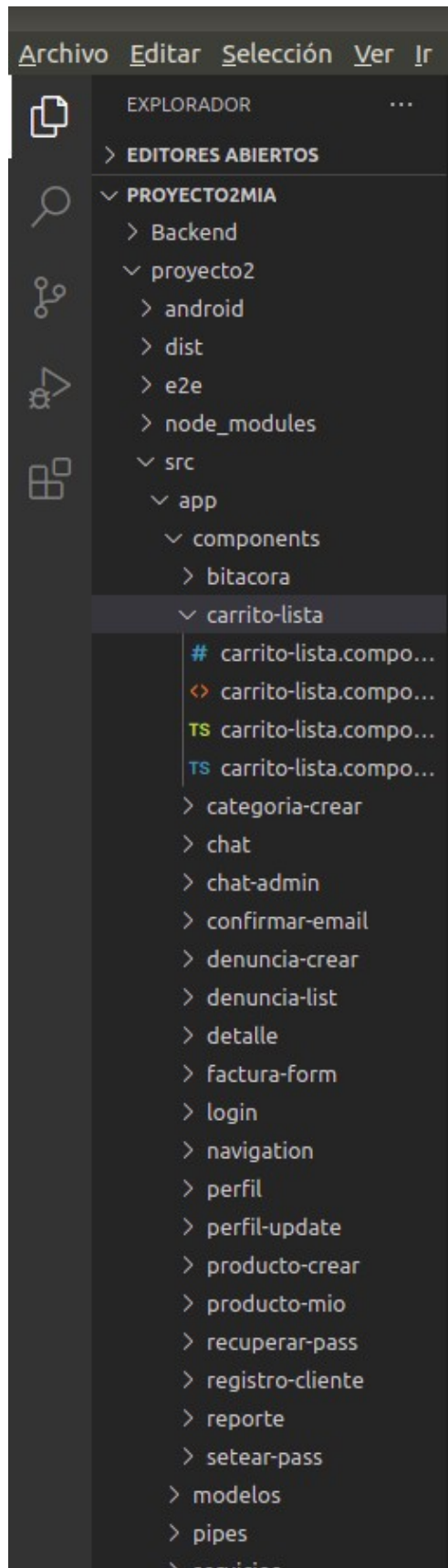
* Modificar el webDir con la carpeta de salida del compilado de angular en el archivo capacitor.config.json

```
```json  
{
 "webDir" : "dist/proyecto2"
}
```

También tenemos una carpeta muy importante que está en la siguiente ruta `src/app` en donde encontramos una carpeta de componentes, modelos, pipes (angular), servicios y assets

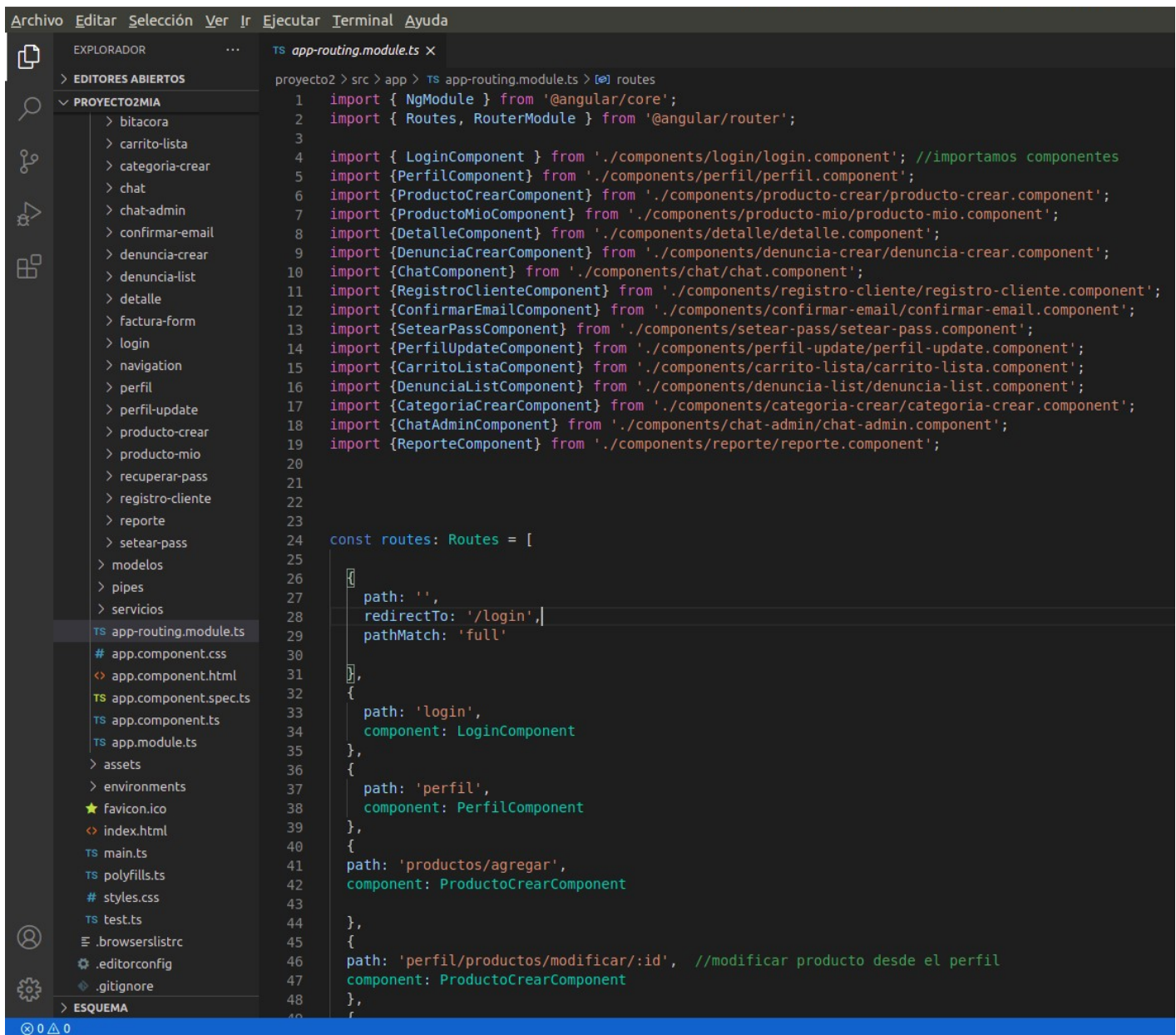


Dentro de la capeta components podemos encontrar todos las paginas que van a interactuar con los usuarios





Otro archivo importante es el `app-routing.module.ts` donde configuramos todas las rutas que nos dirigen a los componentes dentro de la aplicación



```
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXPLORADOR ... TS app-routing.module.ts x

> EDITORES ABIERTOS
 PROYECTO2MIA
 > bitacora
 > carrito-lista
 > categoria-crear
 > chat
 > chat-admin
 > confirmar-email
 > denuncia-crear
 > denuncia-list
 > detalle
 > factura-form
 > login
 > navigation
 > perfil
 > perfil-update
 > producto-crear
 > producto-mio
 > recuperar-pass
 > registro-cliente
 > reporte
 > setear-pass
 > modelos
 > pipes
 > servicios
 TS app-routing.module.ts
 # app.component.css
 <> app.component.html
 TS app.component.spec.ts
 TS app.component.ts
 TS app.module.ts
 > assets
 > environments
 ★ favicon.ico
 <> index.html
 TS main.ts
 TS polyfills.ts
 # styles.css
 TS test.ts
 ≡ .browserslistrc
 ⚙ .editorconfig
 ⚙ .gitignore
 > ESQUEMA

proyecto2 > src > app > TS app-routing.module.ts > routes
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 import { LoginComponent } from './components/login/login.component'; //importamos componentes
5 import { PerfilComponent } from './components/perfil/perfil.component';
6 import { ProductoCrearComponent } from './components/producto-crear/producto-crear.component';
7 import { ProductoMioComponent } from './components/producto-mio/producto-mio.component';
8 import { DetalleComponent } from './components/detalle/detalle.component';
9 import { DenunciaCrearComponent } from './components/denuncia-crear/denuncia-crear.component';
10 import { ChatComponent } from './components/chat/chat.component';
11 import { RegistroClienteComponent } from './components/registro-cliente/registro-cliente.component';
12 import { ConfirmarEmailComponent } from './components/confirmar-email/confirmar-email.component';
13 import { SetearPassComponent } from './components/setear-pass/setear-pass.component';
14 import { PerfilUpdateComponent } from './components/perfil-update/perfil-update.component';
15 import { CarritoListaComponent } from './components/carrito-lista/carrito-lista.component';
16 import { DenunciaListComponent } from './components/denuncia-list/denuncia-list.component';
17 import { CategoriaCrearComponent } from './components/categoria-crear/categoria-crear.component';
18 import { ChatAdminComponent } from './components/chat-admin/chat-admin.component';
19 import { ReporteComponent } from './components/reporte/reporte.component';
20
21
22
23
24 const routes: Routes = [
25 {
26 path: '',
27 redirectTo: '/login',
28 pathMatch: 'full'
29 },
30 {
31 path: 'login',
32 component: LoginComponent
33 },
34 {
35 path: 'perfil',
36 component: PerfilComponent
37 },
38 {
39 path: 'productos/agregar',
40 component: ProductoCrearComponent
41 },
42 {
43 path: 'perfil/productos/modificar/:id', //modificar producto desde el perfil
44 component: ProductoCrearComponent
45 },
46 {
47 path: 'reporte',
48 component: ReporteComponent
49 },
50]
```