



Tecnológico de Estudios Superiores de Ecatepec
División de Ingeniería en Sistemas
Computacionales
Academia en Ciencias de la Ingeniería

Materia

Bases de Datos para Dispositivos Móviles

Grupo 5851

Alumnos

Arreguin Campos Bryan Gustavo

Campero Granados Luis Daniel – Modulo Rutas

Hernandez Cervantes Giovani

Rodriguez Fuentes Axel Roman

Ruiz Teodocio Jose Pablo

Profesora

Cortes Barrera Griselda

Paqueteria

Link:

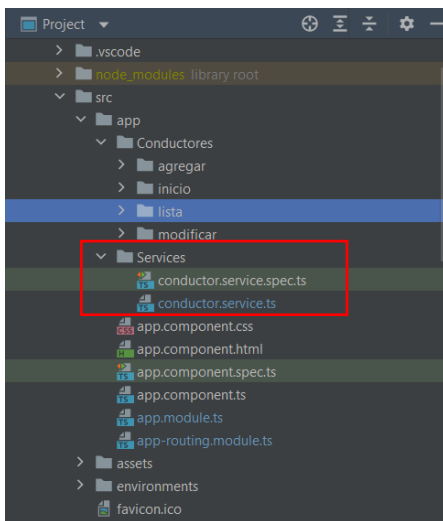
<https://github.com/LuisCampero55/ProyectoPaqueteriaEquipoRoma>

Arreguin Campos Bryan Gustavo

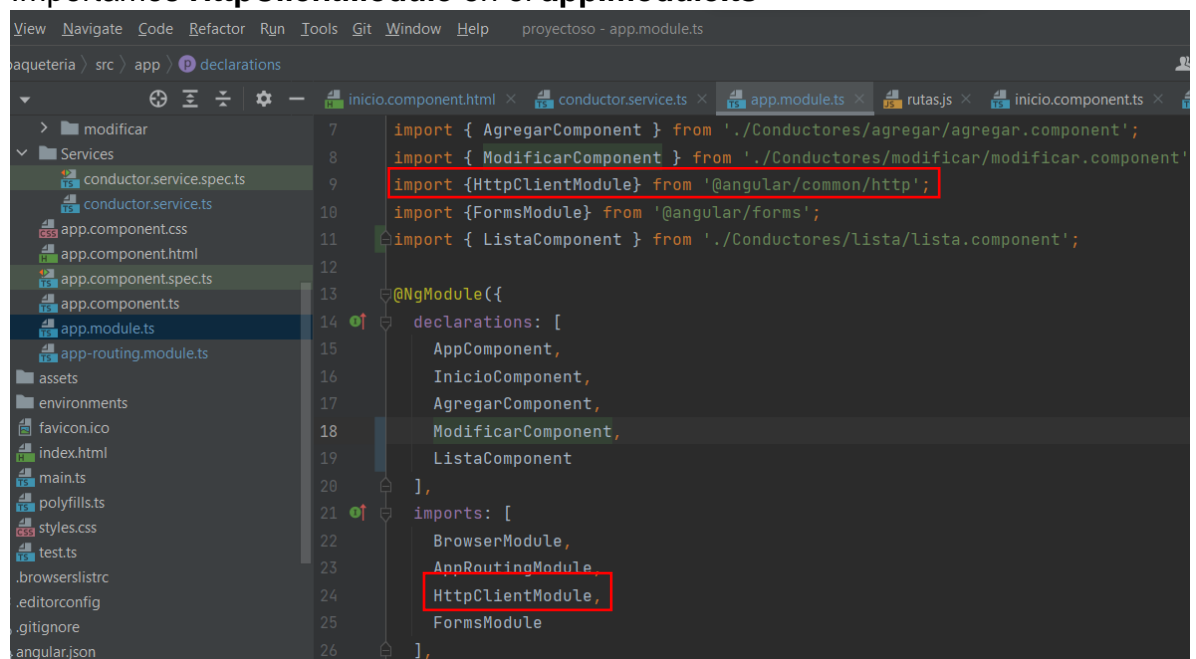
- Lo primero que debemos hacer es crear un servicio en nuestro proyecto y lo pondremos dentro de una carpeta llamada **Services** donde nuestro servicio se llamara **conductor** con el comando siguiente:

```
✓ Compiled successfully.  
PS C:\Users\bryan\Desktop\proyectos\paqueteria> ng g s Services/conductor
```

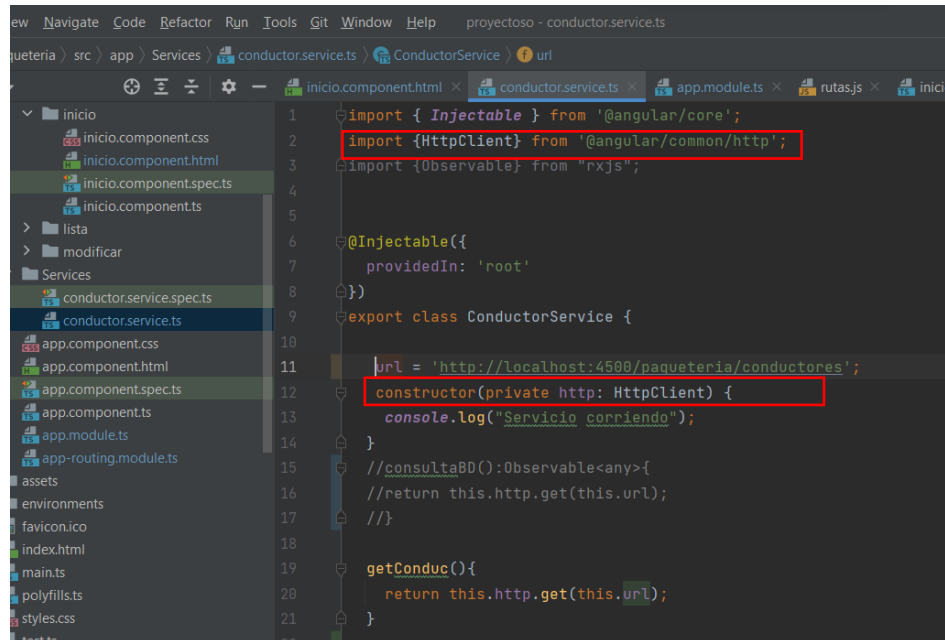
- Al crearlo nos aparecerá en nuestras carpetas



- Importamos **HttpClientModule** en el **app.module.ts**

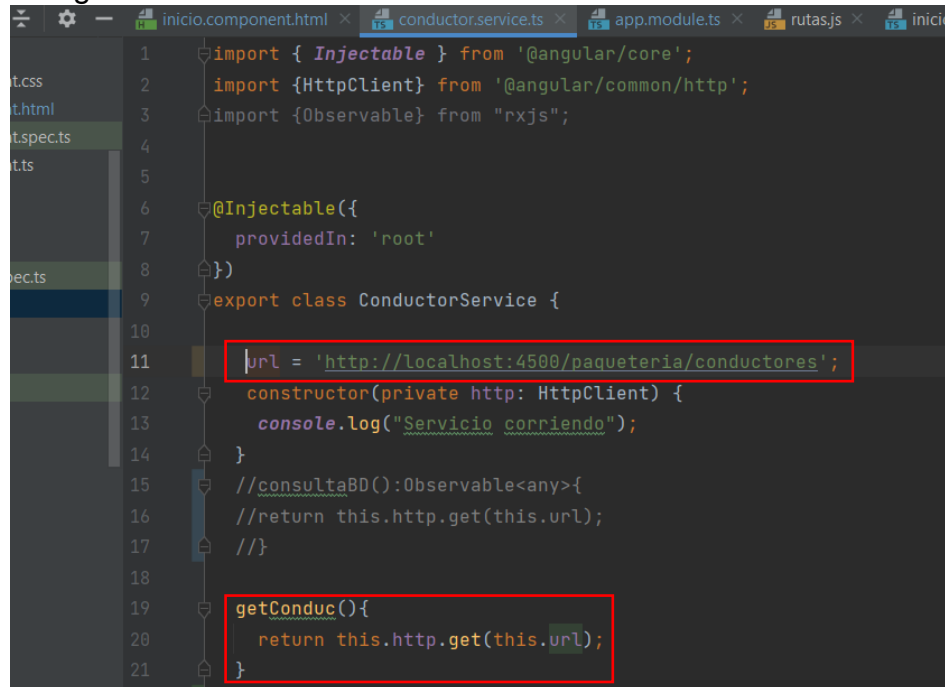


- Después importamos el **HttpClient** e inicializamos en el constructor, esto lo hacemos en nuestro **conductor.service.ts**



```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class ConductorService {
10
11   url = 'http://localhost:4500/paqueteria/conductores';
12   constructor(private http: HttpClient) {
13     console.log("Servicio corriendo");
14   }
15   //consultaBD():Observable<any>{
16     //return this.http.get(this.url);
17   //}
18
19   getConduc(){
20     return this.http.get(this.url);
21   }
22 }
```

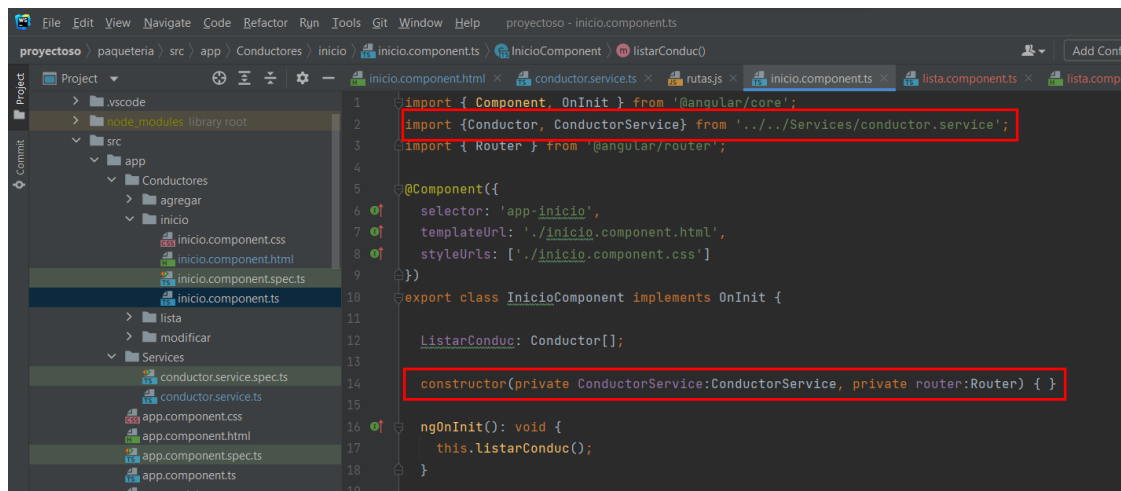
- Primero ingresamos la url de la ruta donde nos muestra los datos de nuestra base de datos y después creamos nuestro método con el cual vamos a recoger los datos de nuestra base de datos.



```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class ConductorService {
10
11   url = 'http://localhost:4500/paqueteria/conductores';
12   constructor(private http: HttpClient) {
13     console.log("Servicio corriendo");
14   }
15   //consultaBD():Observable<any>{
16     //return this.http.get(this.url);
17   //}
18
19   getConduc(){
20     return this.http.get(this.url);
21   }
22 }
```

- Escogemos o creamos en caso de no tener, una carpeta en donde vamos a mostrar el diseño de lo que queremos mostrar, en mi caso en carpeta de **inicio** vamos a mostrar los datos de la tabla **conductores** de nuestra base de datos.

Lo primero que haremos será importar los datos que tengamos en nuestra carpeta **service** en la carpeta que vamos a ocupar para mostrar los datos en este caso usare la carpeta de **inicio**, en el archivo **inicio.component.ts** y la inicializamos en el constructor.

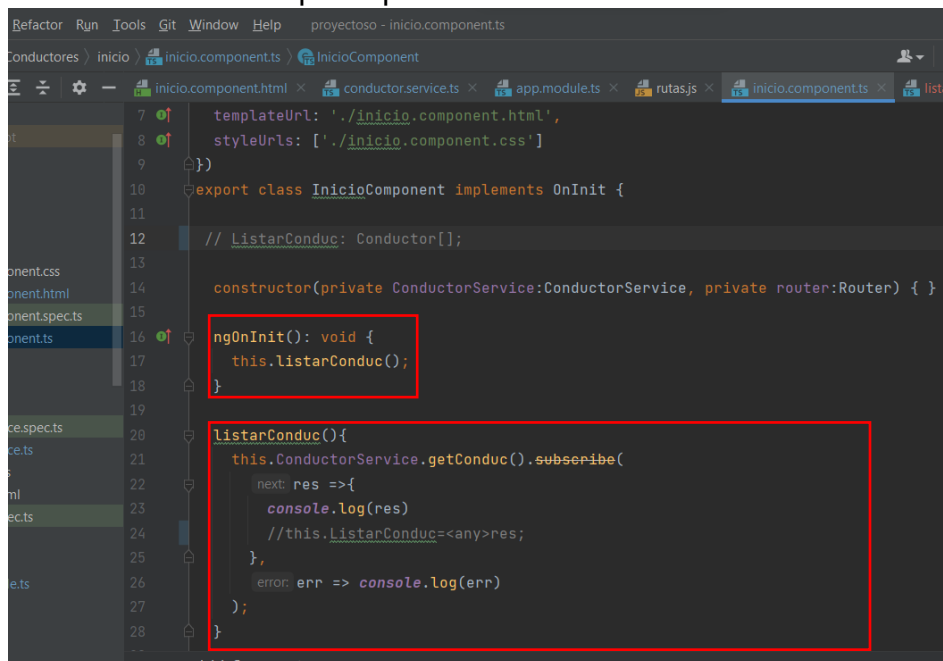


```

1 import { Component, OnInit } from '@angular/core';
2 import { Conductor, ConductorService } from '../Services/conductor.service';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-inicio',
7   templateUrl: './inicio.component.html',
8   styleUrls: ['./inicio.component.css']
9 })
10 export class InicioComponent implements OnInit {
11
12   ListarConduc: Conductor[];
13
14   constructor(private ConductorService: ConductorService, private router: Router) {}
15
16   ngOnInit(): void {
17     this.ListarConduc();
18   }
19 }

```

- Creamos un método para llamar al **getConduc** de nuestro servicio y mandamos a llamar para que lo muestre en consola.

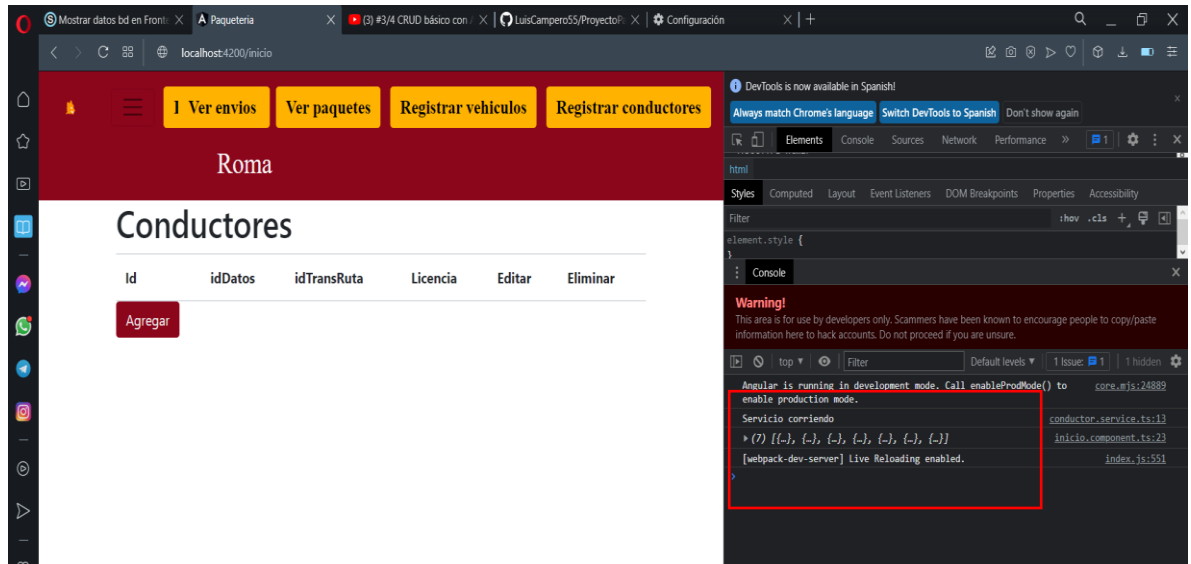


```

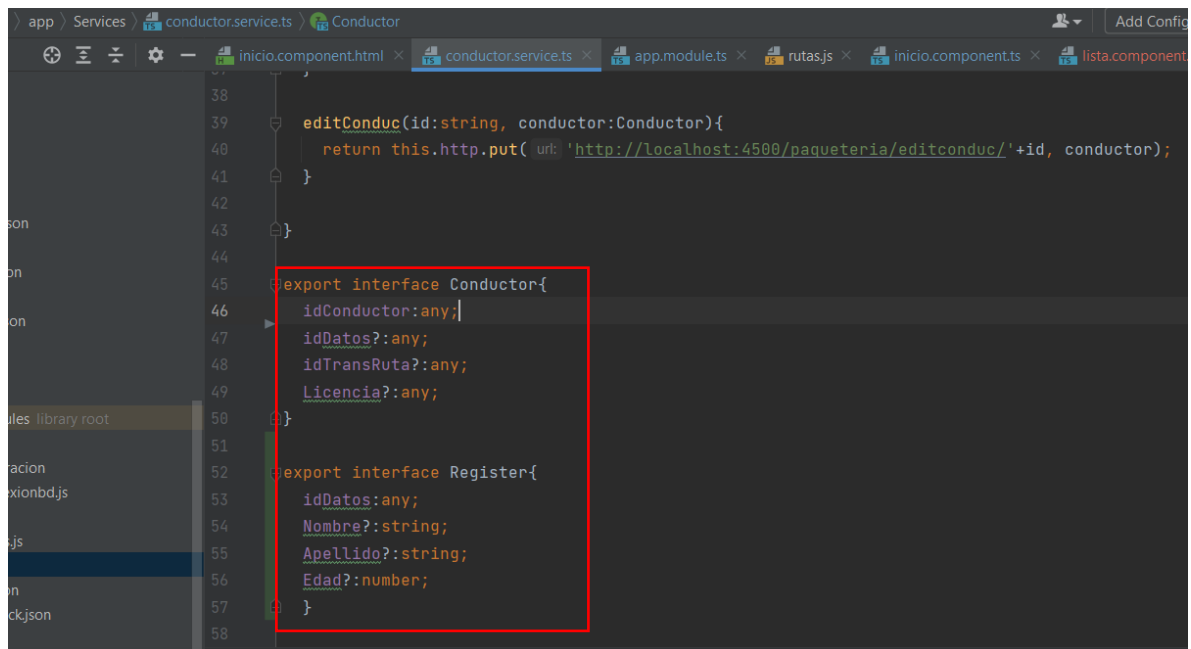
7   templateUrl: './inicio.component.html',
8   styleUrls: ['./inicio.component.css']
9 })
10 export class InicioComponent implements OnInit {
11
12   // ListarConduc: Conductor[];
13
14   constructor(private ConductorService: ConductorService, private router: Router) {}
15
16   ngOnInit(): void {
17     this.ListarConduc();
18   }
19
20   ListarConduc(){
21     this.ConductorService.getConduc().subscribe(
22       next: res =>{
23         console.log(res)
24         //this.ListarConduc=<any>res;
25       },
26       error: err => console.log(err)
27     );
28   }
29 }

```

- Ejecutamos nuestro servidor y en la pantalla donde vamos a mostrar los datos nos vamos a **inspeccionar elemento > console** y nos va a mostrar un array con nuestros datos de la tabla.



- Para mostrar los datos en pantalla lo primero que vamos a hacer es ir al final de nuestro **conductor.service.ts** y exportamos nuestras columnas que tenemos en nuestra tabla.



- En nuestro **inicio.component.ts** vamos a importar nuestra interfaz creada en el servidory crearemos una variable llamada **ListarConduc** en la cual mandaremos a llamar la interface que creamos en nuestro servicio que fue lo anterior que hicimos.

Ponemos el nombre de nuestra interfaz y la mandamos a llamar dentro de nuestro método **listarConduc** para que nos muestre en pantalla los datos y no solo en la consola.

```

1 import { Component, OnInit } from '@angular/core';
2 import { Conductor, ConductorService } from '../Services/conductor.service';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-inicio',
7   templateUrl: './inicio.component.html',
8   styleUrls: ['./inicio.component.css']
9 })
10 export class InicioComponent implements OnInit {
11   ListarConduc: Conductor[];
12
13   constructor(private ConductorService: ConductorService, private router: Router) {}
14
15   ngOnInit(): void {
16     this.listarConduc();
17   }
18
19   listarConduc() {
20     this.ConductorService.getConduc().subscribe(
21       next: res => {
22         console.log(res);
23         this.ListarConduc = <any>res;
24       },
25       error: err => console.log(err)
26     );
27   }
28 }

```

- Finalmente vamos a nuestro **inicio.component.html** para crear la tabla donde se mostraran los datos.

```

1 <div class="container">
2   <p class="h1">Conductores</p>
3
4   <table class="table">
5     <thead>
6       <tr>
7         <th scope="col">Id</th>
8         <!-- <th scope="col">Nombre</th>
9         <th scope="col">Apellido</th>
10        <th scope="col">Edad</th> -->
11        <th scope="col">idDatos</th>
12        <th scope="col">idTransRuta</th>
13        <th scope="col">Licencia</th>
14        <th scope="col">Eliminar</th>
15      </tr>
16    </thead>
17    <tbody>
18      <tr *ngFor="let conductor of ListarConduc">
19        <th>{{conductor.idConductor}}</th>
20        <!-- <td>{{conductor.nombre}}</td>
21        <td>{{conductor.apellido}}</td>
22        <td>{{conductor.edad}}</td> -->
23        <td>{{conductor.idDatos}}</td>
24        <td>{{conductor.idTransRuta}}</td>
25        <td>{{conductor.Licencia}}</td>
26      </tr>
27    </tbody>
28  </table>

```

```

<button (click)="modificar(conductor.idConductor!)" type="button" class="btn btn" style="background-color: #FFB300; color:#000000">Editar</button>
</td>
<td>
<button (click)="eliminar(conductor.idConductor!)" type="button" class="btn btn-danger">Eliminar</button>
</td>
</tr>
<button routerLink="/add" type="button" class="btn" style="background-color: #88851B; color:#f2f2f2">Agregar</button>
</tbody>
</table>
</div>

```

- Creamos ahí mismo un ***ngFor** donde pondremos que de la tabla conductor va a recoger los datos **ListaConduc** junto con los botones de agregar y eliminar que ocuparemos mas adelante.

```

<tr *ngFor="let conductor of ListaConduc">
<th>{{conductor.idConductor}}</th>
<!--<td>{{conductor.nombre}}</td>
<td>{{conductor.apellido}}</td>
<td>{{conductor.edad}}</td>-->
<td>{{conductor.idDatos}}</td>
<td>{{conductor.idTransRuta}}</td>
<td>{{conductor.Licencia}}</td>
<td>
<button (click)="modificar(conductor.idConductor!)" type="button" class="btn btn" style="background-color: #FFB300; color:#000000">Editar</button>
</td>
<td>
<button (click)="eliminar(conductor.idConductor!)" type="button" class="btn btn-danger">Eliminar</button>
</td>
</tr>
</tr>

```

Y así se visualizaría nuestros datos en pantalla

Id	idDatos	idTransRuta	Licencia	Editar	Eliminar
1	1	1	100	Editar	Eliminar
2	2	2	200	Editar	Eliminar
3	3	3	300	Editar	Eliminar
4	3	4	400	Editar	Eliminar
66	5	2	600	Editar	Eliminar
67	2	5	1000	Editar	Eliminar
69	1	2	1101	Editar	Eliminar

Agregar

También con otra tabla llamada **registros** se hizo lo mismo.

Registros

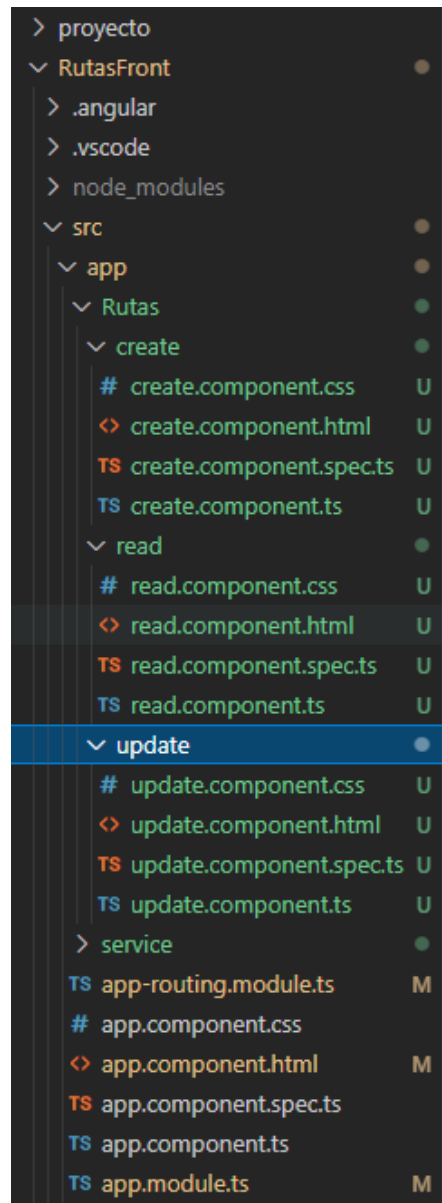
Id	Nombre	Apellido	Edad
1	Luis	Campero	22
2	Bryan	Campos	21
3	Pablo	Ruiz	33
4	Axel	Roman	12
5	Felipe	Rapero	50

Regresar

Campero Granados Luis Daniel

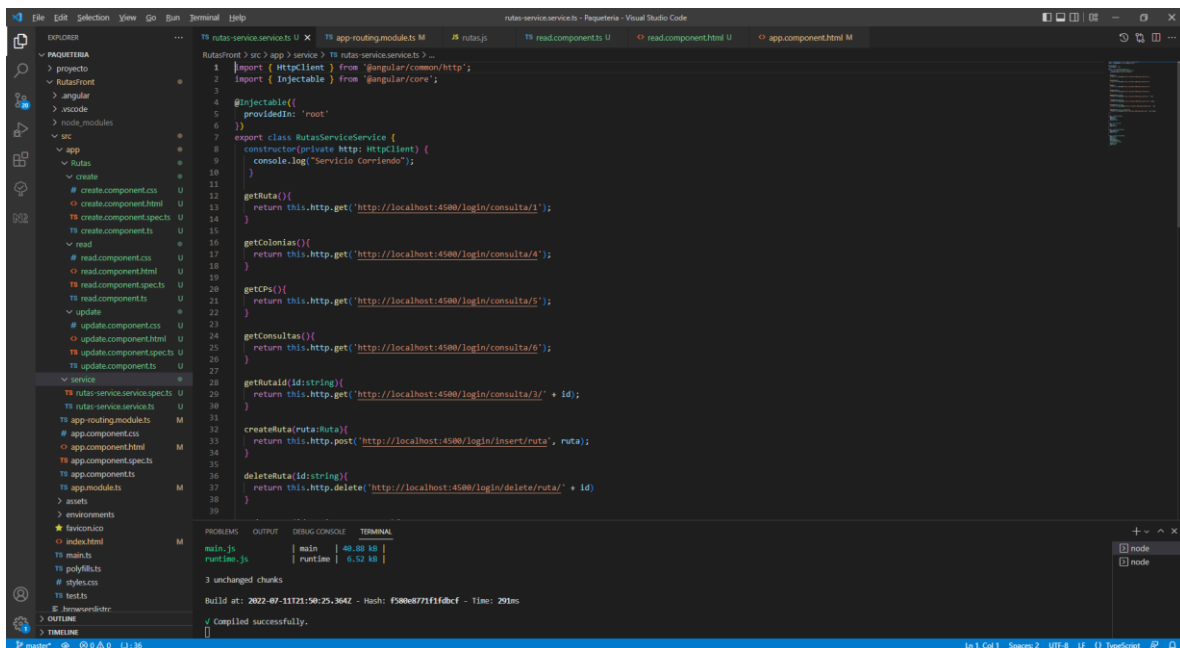
Para poder mostrar nuestras tablas en nuestra pagina tendremos que hacer los siguientes pasos;

1.- Debemos de tener nuestros componentes que se encargaran para altas, modificaciones y consultas



Podemos ver que tenemos de componentes *create*, *read* y *update*.

2.- Después crearemos nuestro *service* que tendremos nuestros métodos en el, para crear el service ingresamos el siguiente comando; `ng g s service/rutas-service`:

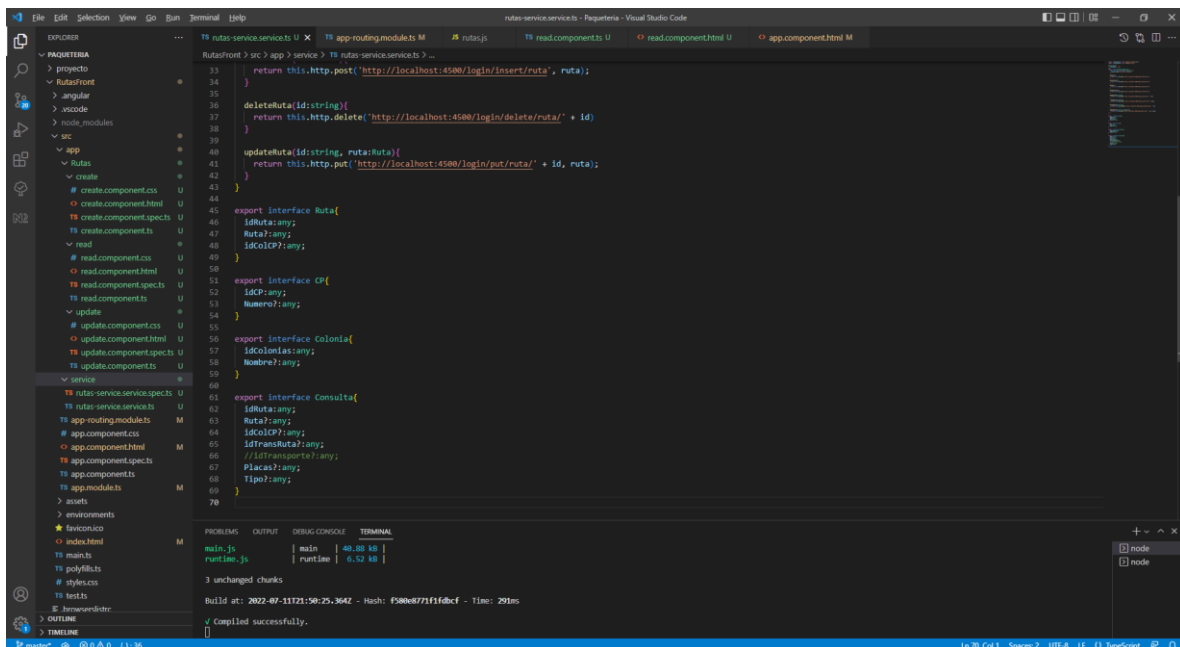


The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with folders like 'projecto', 'RutasFront', 'angular', 'src', 'app', 'rutas', 'create', 'read', 'update', 'service', 'app-routing.module.ts', 'app.component.ts', 'app.module.ts', 'assets', 'environments', 'index.html', 'main.ts', 'polyfills.ts', 'styles.css', 'tests', and 'browser.e2e.ts'. The 'service' folder is expanded, showing files like 'rutas-service.service.ts', 'rutas-service.spec.ts', 'app-routing.module.ts', 'app.component.ts', 'app.module.ts', 'assets', and 'environments'. The 'rutas-service.service.ts' file is open in the editor, showing the following code:

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class RutasServiceService {
8   constructor(private http: HttpClient) {
9     console.log("Servicio Corriendo");
10  }
11
12  getRuta() {
13    return this.http.get('http://localhost:4500/login/consulta/1');
14  }
15
16  getColonias() {
17    return this.http.get('http://localhost:4500/login/consulta/4');
18  }
19
20  getCPis() {
21    return this.http.get('http://localhost:4500/login/consulta/5');
22  }
23
24  getConsultas() {
25    return this.http.get('http://localhost:4500/login/consulta/6');
26  }
27
28  getRutaId(id:string){
29    return this.http.get('http://localhost:4500/login/consulta/3/' + id);
30  }
31
32  createRuta(ruta:Ruta){
33    return this.http.post('http://localhost:4500/login/insert/ruta', ruta);
34  }
35
36  deleteRuta(id:string){
37    return this.http.delete('http://localhost:4500/login/delete/ruta/' + id);
38  }
39
40  updateRuta(id:string, ruta:Ruta){
41    return this.http.put('http://localhost:4500/login/put/ruta/' + id, ruta);
42  }
43
44  export interface Ruta{
45    idRuta:any;
46    Ruta?:any;
47    idColP?:any;
48  }
49
50  export interface CP{
51    idCP:any;
52    Numero?:any;
53  }
54
55  export interface Colonia{
56    idColonia:any;
57    idColP?:any;
58    Nombre?:any;
59  }
60
61  export interface Consulta{
62    idRuta:any;
63    idColP?:any;
64    idTransporte?:any;
65    Placas?:any;
66    Tipo?:any;
67  }
68
69  }
70
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
main.js | main | 40.88 KB |
runtime.js | runtime | 6.52 KB |
3 unchanged chunks
Build at: 2022-07-11T21:50:25.364Z - Hash: F5068771f1f6bdf - Time: 291ms
Compiled successfully.
```



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with folders like 'projecto', 'RutasFront', 'angular', 'src', 'app', 'rutas', 'create', 'read', 'update', 'service', 'app-routing.module.ts', 'app.component.ts', 'app.module.ts', 'assets', 'environments', 'index.html', 'main.ts', 'polyfills.ts', 'styles.css', 'tests', and 'browser.e2e.ts'. The 'service' folder is expanded, showing files like 'rutas-service.service.ts', 'rutas-service.spec.ts', 'app-routing.module.ts', 'app.component.ts', 'app.module.ts', 'assets', and 'environments'. The 'rutas-service.service.ts' file is open in the editor, showing the following code:

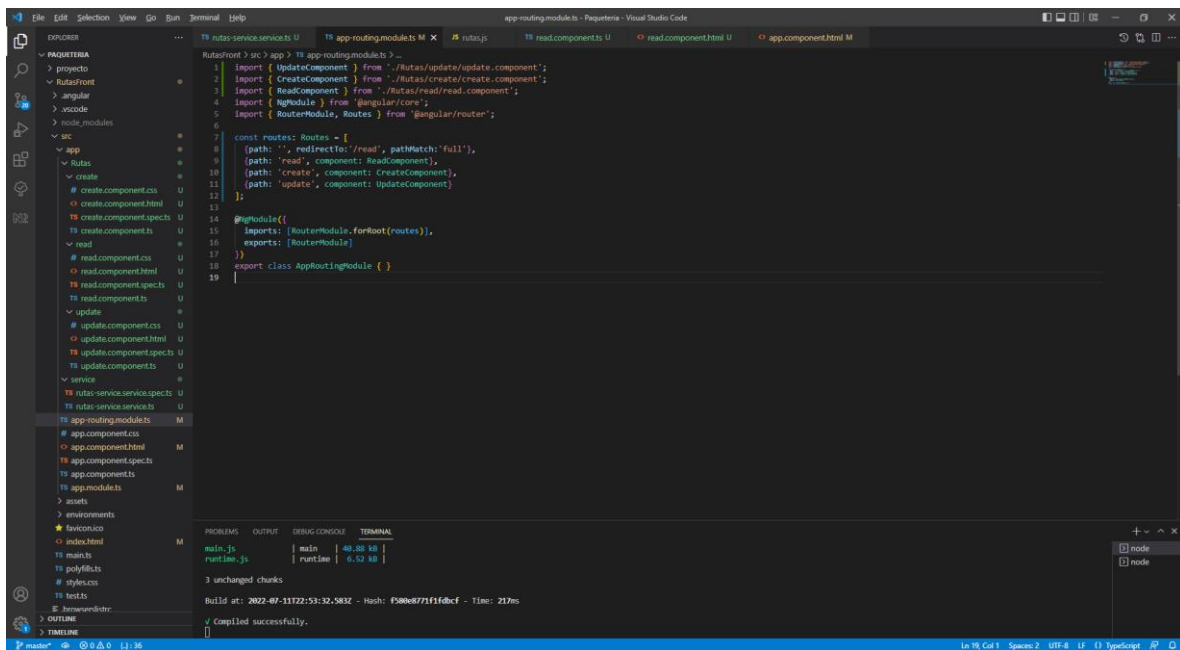
```
33 return this.http.post('http://localhost:4500/login/insert/ruta', ruta);
34 }
35
36 deleteRuta(id:string){
37   return this.http.delete('http://localhost:4500/login/delete/ruta/' + id);
38 }
39
40 updateRuta(id:string, ruta:Ruta){
41   return this.http.put('http://localhost:4500/login/put/ruta/' + id, ruta);
42 }
43
44 export interface Ruta{
45   idRuta:any;
46   Ruta?:any;
47   idColP?:any;
48 }
49
50 export interface CP{
51   idCP:any;
52   Numero?:any;
53 }
54
55 export interface Colonia{
56   idColonia:any;
57   idColP?:any;
58   Nombre?:any;
59 }
60
61 export interface Consulta{
62   idRuta:any;
63   idColP?:any;
64   idTransporte?:any;
65   Placas?:any;
66   Tipo?:any;
67 }
68
69 }
70
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
main.js | main | 40.88 KB |
runtime.js | runtime | 6.52 KB |
3 unchanged chunks
Build at: 2022-07-11T21:50:25.364Z - Hash: F5068771f1f6bdf - Time: 291ms
Compiled successfully.
```

Podemos ver que tenemos los métodos para las consultas además de que tendremos las interfaces de las mismas consultas.

3.- Después tendremos que ver si nuestras rutas están correctas;



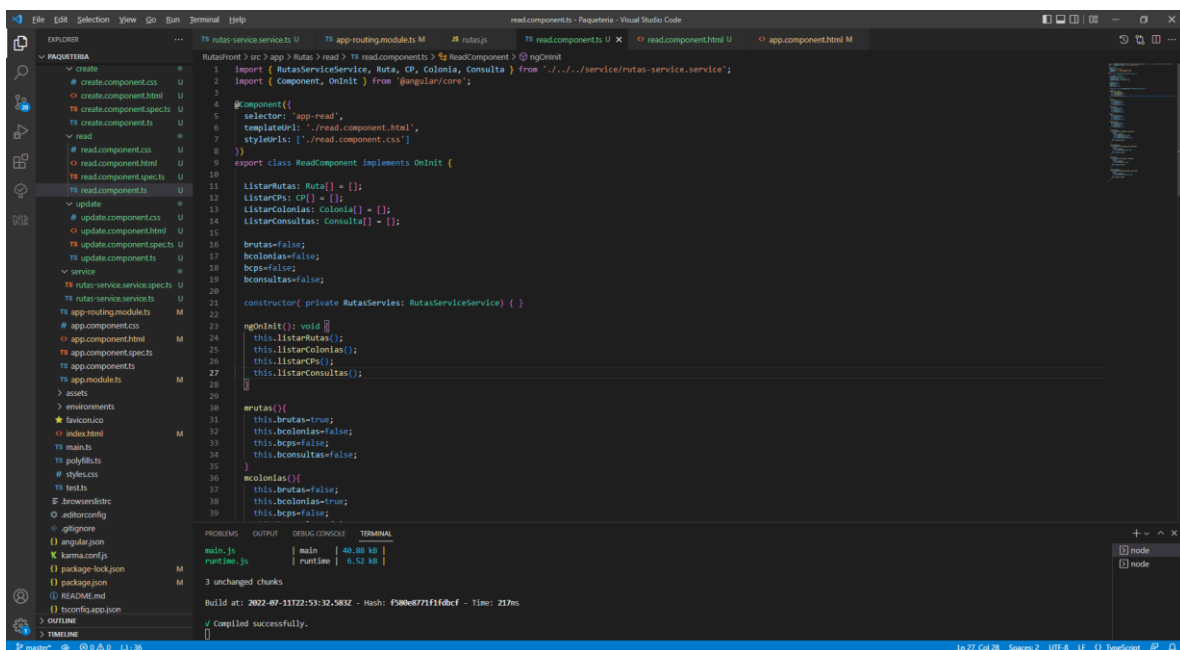
The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'src' folder is expanded, showing the 'app' folder. The 'app' folder contains 'create', 'read', 'update', and 'service' subfolders. The 'read' folder contains 'read.component.css', 'read.component.html', 'read.component.spec.ts', and 'read.component.ts'. The 'read.component.ts' file is selected, showing the following code:

```
1 import { UpdateComponent } from './routes/update/update.component';
2 import { CreateComponent } from './routes/create/create.component';
3 import { ReadComponent } from './routes/read/read.component';
4 import { NgModule } from '@angular/core';
5 import { RouterModule, Routes } from '@angular/router';
6
7 const routes: Routes = [
8   { path: '', redirectTo: '/read', pathMatch: 'full' },
9   { path: 'read', component: ReadComponent },
10  { path: 'create', component: CreateComponent },
11  { path: 'update', component: UpdateComponent }
12 ];
13
14 @NgModule({
15   imports: [RouterModule.forRoot(routes)],
16   exports: [RouterModule]
17 })
18 export class AppRoutingModule {}
19
```

The bottom panel shows the 'PROBLEMS' and 'OUTPUT' tabs. The 'OUTPUT' tab shows the following message:

```
main.js | main | 40.88 KB |
runtime.js | runtime | 6.52 KB |
3 unchanged chunks
Build at: 2022-07-11T22:53:32.583Z - Hash: F580e877f1f16bdf - Time: 217ms
✓ Compiled successfully.
```

4.- Para finalizar en nuestro componente de consulta tendremos que codificar nuestro archivo .ts



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'src' folder is expanded, showing the 'app' folder. The 'app' folder contains 'create', 'read', 'update', and 'service' subfolders. The 'read' folder contains 'read.component.css', 'read.component.html', 'read.component.spec.ts', and 'read.component.ts'. The 'read.component.ts' file is selected, showing the following code:

```
1 import { RutasServiceService, Ruta, CP, Colonia, Consulta } from './../service/rutas-service.service';
2 import { Component, OnInit } from '@angular/core';
3
4 @Component({
5   selector: 'app-read',
6   templateUrl: './read.component.html',
7   styleUrls: ['./read.component.css']
8 })
9 export class ReadComponent implements OnInit {
10
11   listarRutas: Ruta[] = [];
12   listarCPs: CP[] = [];
13   listarColonias: Colonia[] = [];
14   listarConsultas: Consulta[] = [];
15
16   brutas=false;
17   bcolonias=false;
18   bcps=false;
19   bconsultas=false;
20
21   constructor( private RutasService: RutasServiceService ) {}
22
23   ngOnInit(): void {
24     this.listarRutas();
25     this.listarColonias();
26     this.listarCPs();
27     this.listarConsultas();
28   }
29
30   listarRutas() {
31     this.butas=true;
32     this.bcolonias=false;
33     this.bcps=false;
34     this.bconsultas=false;
35   }
36
37   listarColonias() {
38     this.butas=false;
39     this.bcolonias=true;
40     this.bcps=false;
41     this.bconsultas=false;
42   }
43
44   listarCPs() {
45     this.butas=true;
46     this.bcolonias=false;
47     this.bcps=true;
48     this.bconsultas=false;
49   }
50
51   listarConsultas() {
52     this.butas=false;
53     this.bcolonias=false;
54     this.bcps=false;
55     this.bconsultas=true;
56   }
57
58   ngOnInit(): void {
59     this.listarRutas();
60     this.listarColonias();
61     this.listarCPs();
62     this.listarConsultas();
63   }
64
65   ngOnInit(): void {
66     this.listarRutas();
67     this.listarColonias();
68     this.listarCPs();
69     this.listarConsultas();
70   }
71
72   ngOnInit(): void {
73     this.listarRutas();
74     this.listarColonias();
75     this.listarCPs();
76     this.listarConsultas();
77   }
78
79   ngOnInit(): void {
80     this.listarRutas();
81     this.listarColonias();
82     this.listarCPs();
83     this.listarConsultas();
84   }
85
86   ngOnInit(): void {
87     this.listarRutas();
88     this.listarColonias();
89     this.listarCPs();
90     this.listarConsultas();
91   }
92
93   ngOnInit(): void {
94     this.listarRutas();
95     this.listarColonias();
96     this.listarCPs();
97     this.listarConsultas();
98   }
99
100  ngOnInit(): void {
101    this.listarRutas();
102    this.listarColonias();
103    this.listarCPs();
104    this.listarConsultas();
105  }
106
107  ngOnInit(): void {
108    this.listarRutas();
109    this.listarColonias();
110    this.listarCPs();
111    this.listarConsultas();
112  }
113
114  ngOnInit(): void {
115    this.listarRutas();
116    this.listarColonias();
117    this.listarCPs();
118    this.listarConsultas();
119  }
120
121  ngOnInit(): void {
122    this.listarRutas();
123    this.listarColonias();
124    this.listarCPs();
125    this.listarConsultas();
126  }
127
128  ngOnInit(): void {
129    this.listarRutas();
130    this.listarColonias();
131    this.listarCPs();
132    this.listarConsultas();
133  }
134
135  ngOnInit(): void {
136    this.listarRutas();
137    this.listarColonias();
138    this.listarCPs();
139    this.listarConsultas();
140  }
141
142  ngOnInit(): void {
143    this.listarRutas();
144    this.listarColonias();
145    this.listarCPs();
146    this.listarConsultas();
147  }
148
149  ngOnInit(): void {
150    this.listarRutas();
151    this.listarColonias();
152    this.listarCPs();
153    this.listarConsultas();
154  }
155
156  ngOnInit(): void {
157    this.listarRutas();
158    this.listarColonias();
159    this.listarCPs();
160    this.listarConsultas();
161  }
162
163  ngOnInit(): void {
164    this.listarRutas();
165    this.listarColonias();
166    this.listarCPs();
167    this.listarConsultas();
168  }
169
170  ngOnInit(): void {
171    this.listarRutas();
172    this.listarColonias();
173    this.listarCPs();
174    this.listarConsultas();
175  }
176
177  ngOnInit(): void {
178    this.listarRutas();
179    this.listarColonias();
180    this.listarCPs();
181    this.listarConsultas();
182  }
183
184  ngOnInit(): void {
185    this.listarRutas();
186    this.listarColonias();
187    this.listarCPs();
188    this.listarConsultas();
189  }
190
191  ngOnInit(): void {
192    this.listarRutas();
193    this.listarColonias();
194    this.listarCPs();
195    this.listarConsultas();
196  }
197
198  ngOnInit(): void {
199    this.listarRutas();
200    this.listarColonias();
201    this.listarCPs();
202    this.listarConsultas();
203  }
204
205  ngOnInit(): void {
206    this.listarRutas();
207    this.listarColonias();
208    this.listarCPs();
209    this.listarConsultas();
210  }
211
212  ngOnInit(): void {
213    this.listarRutas();
214    this.listarColonias();
215    this.listarCPs();
216    this.listarConsultas();
217  }
218
219  ngOnInit(): void {
220    this.listarRutas();
221    this.listarColonias();
222    this.listarCPs();
223    this.listarConsultas();
224  }
225
226  ngOnInit(): void {
227    this.listarRutas();
228    this.listarColonias();
229    this.listarCPs();
230    this.listarConsultas();
231  }
232
233  ngOnInit(): void {
234    this.listarRutas();
235    this.listarColonias();
236    this.listarCPs();
237    this.listarConsultas();
238  }
239
240  ngOnInit(): void {
241    this.listarRutas();
242    this.listarColonias();
243    this.listarCPs();
244    this.listarConsultas();
245  }
246
247  ngOnInit(): void {
248    this.listarRutas();
249    this.listarColonias();
250    this.listarCPs();
251    this.listarConsultas();
252  }
253
254  ngOnInit(): void {
255    this.listarRutas();
256    this.listarColonias();
257    this.listarCPs();
258    this.listarConsultas();
259  }
260
261  ngOnInit(): void {
262    this.listarRutas();
263    this.listarColonias();
264    this.listarCPs();
265    this.listarConsultas();
266  }
267
268  ngOnInit(): void {
269    this.listarRutas();
270    this.listarColonias();
271    this.listarCPs();
272    this.listarConsultas();
273  }
274
275  ngOnInit(): void {
276    this.listarRutas();
277    this.listarColonias();
278    this.listarCPs();
279    this.listarConsultas();
280  }
281
282  ngOnInit(): void {
283    this.listarRutas();
284    this.listarColonias();
285    this.listarCPs();
286    this.listarConsultas();
287  }
288
289  ngOnInit(): void {
290    this.listarRutas();
291    this.listarColonias();
292    this.listarCPs();
293    this.listarConsultas();
294  }
295
296  ngOnInit(): void {
297    this.listarRutas();
298    this.listarColonias();
299    this.listarCPs();
300    this.listarConsultas();
301  }
302
303  ngOnInit(): void {
304    this.listarRutas();
305    this.listarColonias();
306    this.listarCPs();
307    this.listarConsultas();
308  }
309
310  ngOnInit(): void {
311    this.listarRutas();
312    this.listarColonias();
313    this.listarCPs();
314    this.listarConsultas();
315  }
316
317  ngOnInit(): void {
318    this.listarRutas();
319    this.listarColonias();
320    this.listarCPs();
321    this.listarConsultas();
322  }
323
324  ngOnInit(): void {
325    this.listarRutas();
326    this.listarColonias();
327    this.listarCPs();
328    this.listarConsultas();
329  }
330
331  ngOnInit(): void {
332    this.listarRutas();
333    this.listarColonias();
334    this.listarCPs();
335    this.listarConsultas();
336  }
337
338  ngOnInit(): void {
339    this.listarRutas();
340    this.listarColonias();
341    this.listarCPs();
342    this.listarConsultas();
343  }
344
345  ngOnInit(): void {
346    this.listarRutas();
347    this.listarColonias();
348    this.listarCPs();
349    this.listarConsultas();
350  }
351
352  ngOnInit(): void {
353    this.listarRutas();
354    this.listarColonias();
355    this.listarCPs();
356    this.listarConsultas();
357  }
358
359  ngOnInit(): void {
360    this.listarRutas();
361    this.listarColonias();
362    this.listarCPs();
363    this.listarConsultas();
364  }
365
366  ngOnInit(): void {
367    this.listarRutas();
368    this.listarColonias();
369    this.listarCPs();
370    this.listarConsultas();
371  }
372
373  ngOnInit(): void {
374    this.listarRutas();
375    this.listarColonias();
376    this.listarCPs();
377    this.listarConsultas();
378  }
379
380  ngOnInit(): void {
381    this.listarRutas();
382    this.listarColonias();
383    this.listarCPs();
384    this.listarConsultas();
385  }
386
387  ngOnInit(): void {
388    this.listarRutas();
389    this.listarColonias();
390    this.listarCPs();
391    this.listarConsultas();
392  }
393
394  ngOnInit(): void {
395    this.listarRutas();
396    this.listarColonias();
397    this.listarCPs();
398    this.listarConsultas();
399  }
400
401  ngOnInit(): void {
402    this.listarRutas();
403    this.listarColonias();
404    this.listarCPs();
405    this.listarConsultas();
406  }
407
408  ngOnInit(): void {
409    this.listarRutas();
410    this.listarColonias();
411    this.listarCPs();
412    this.listarConsultas();
413  }
414
415  ngOnInit(): void {
416    this.listarRutas();
417    this.listarColonias();
418    this.listarCPs();
419    this.listarConsultas();
420  }
421
422  ngOnInit(): void {
423    this.listarRutas();
424    this.listarColonias();
425    this.listarCPs();
426    this.listarConsultas();
427  }
428
429  ngOnInit(): void {
430    this.listarRutas();
431    this.listarColonias();
432    this.listarCPs();
433    this.listarConsultas();
434  }
435
436  ngOnInit(): void {
437    this.listarRutas();
438    this.listarColonias();
439    this.listarCPs();
440    this.listarConsultas();
441  }
442
443  ngOnInit(): void {
444    this.listarRutas();
445    this.listarColonias();
446    this.listarCPs();
447    this.listarConsultas();
448  }
449
450  ngOnInit(): void {
451    this.listarRutas();
452    this.listarColonias();
453    this.listarCPs();
454    this.listarConsultas();
455  }
456
457  ngOnInit(): void {
458    this.listarRutas();
459    this.listarColonias();
460    this.listarCPs();
461    this.listarConsultas();
462  }
463
464  ngOnInit(): void {
465    this.listarRutas();
466    this.listarColonias();
467    this.listarCPs();
468    this.listarConsultas();
469  }
470
471  ngOnInit(): void {
472    this.listarRutas();
473    this.listarColonias();
474    this.listarCPs();
475    this.listarConsultas();
476  }
477
478  ngOnInit(): void {
479    this.listarRutas();
480    this.listarColonias();
481    this.listarCPs();
482    this.listarConsultas();
483  }
484
485  ngOnInit(): void {
486    this.listarRutas();
487    this.listarColonias();
488    this.listarCPs();
489    this.listarConsultas();
490  }
491
492  ngOnInit(): void {
493    this.listarRutas();
494    this.listarColonias();
495    this.listarCPs();
496    this.listarConsultas();
497  }
498
499  ngOnInit(): void {
500    this.listarRutas();
501    this.listarColonias();
502    this.listarCPs();
503    this.listarConsultas();
504  }
505
506  ngOnInit(): void {
507    this.listarRutas();
508    this.listarColonias();
509    this.listarCPs();
510    this.listarConsultas();
511  }
512
513  ngOnInit(): void {
514    this.listarRutas();
515    this.listarColonias();
516    this.listarCPs();
517    this.listarConsultas();
518  }
519
520  ngOnInit(): void {
521    this.listarRutas();
522    this.listarColonias();
523    this.listarCPs();
524    this.listarConsultas();
525  }
526
527  ngOnInit(): void {
528    this.listarRutas();
529    this.listarColonias();
530    this.listarCPs();
531    this.listarConsultas();
532  }
533
534  ngOnInit(): void {
535    this.listarRutas();
536    this.listarColonias();
537    this.listarCPs();
538    this.listarConsultas();
539  }
540
541  ngOnInit(): void {
542    this.listarRutas();
543    this.listarColonias();
544    this.listarCPs();
545    this.listarConsultas();
546  }
547
548  ngOnInit(): void {
549    this.listarRutas();
550    this.listarColonias();
551    this.listarCPs();
552    this.listarConsultas();
553  }
554
555  ngOnInit(): void {
556    this.listarRutas();
557    this.listarColonias();
558    this.listarCPs();
559    this.listarConsultas();
560  }
561
562  ngOnInit(): void {
563    this.listarRutas();
564    this.listarColonias();
565    this.listarCPs();
566    this.listarConsultas();
567  }
568
569  ngOnInit(): void {
570    this.listarRutas();
571    this.listarColonias();
572    this.listarCPs();
573    this.listarConsultas();
574  }
575
576  ngOnInit(): void {
577    this.listarRutas();
578    this.listarColonias();
579    this.listarCPs();
580    this.listarConsultas();
581  }
582
583  ngOnInit(): void {
584    this.listarRutas();
585    this.listarColonias();
586    this.listarCPs();
587    this.listarConsultas();
588  }
589
590  ngOnInit(): void {
591    this.listarRutas();
592    this.listarColonias();
593    this.listarCPs();
594    this.listarConsultas();
595  }
596
597  ngOnInit(): void {
598    this.listarRutas();
599    this.listarColonias();
600    this.listarCPs();
601    this.listarConsultas();
602  }
603
604  ngOnInit(): void {
605    this.listarRutas();
606    this.listarColonias();
607    this.listarCPs();
608    this.listarConsultas();
609  }
610
611  ngOnInit(): void {
612    this.listarRutas();
613    this.listarColonias();
614    this.listarCPs();
615    this.listarConsultas();
616  }
617
618  ngOnInit(): void {
619    this.listarRutas();
620    this.listarColonias();
621    this.listarCPs();
622    this.listarConsultas();
623  }
624
625  ngOnInit(): void {
626    this.listarRutas();
627    this.listarColonias();
628    this.listarCPs();
629    this.listarConsultas();
630  }
631
632  ngOnInit(): void {
633    this.listarRutas();
634    this.listarColonias();
635    this.listarCPs();
636    this.listarConsultas();
637  }
638
639  ngOnInit(): void {
640    this.listarRutas();
641    this.listarColonias();
642    this.listarCPs();
643    this.listarConsultas();
644  }
645
646  ngOnInit(): void {
647    this.listarRutas();
648    this.listarColonias();
649    this.listarCPs();
650    this.listarConsultas();
651  }
652
653  ngOnInit(): void {
654    this.listarRutas();
655    this.listarColonias();
656    this.listarCPs();
657    this.listarConsultas();
658  }
659
660  ngOnInit(): void {
661    this.listarRutas();
662    this.listarColonias();
663    this.listarCPs();
664    this.listarConsultas();
665  }
666
667  ngOnInit(): void {
668    this.listarRutas();
669    this.listarColonias();
670    this.listarCPs();
671    this.listarConsultas();
672  }
673
674  ngOnInit(): void {
675    this.listarRutas();
676    this.listarColonias();
677    this.listarCPs();
678    this.listarConsultas();
679  }
680
681  ngOnInit(): void {
682    this.listarRutas();
683    this.listarColonias();
684    this.listarCPs();
685    this.listarConsultas();
686  }
687
688  ngOnInit(): void {
689    this.listarRutas();
690    this.listarColonias();
691    this.listarCPs();
692    this.listarConsultas();
693  }
694
695  ngOnInit(): void {
696    this.listarRutas();
697    this.listarColonias();
698    this.listarCPs();
699    this.listarConsultas();
700  }
701
702  ngOnInit(): void {
703    this.listarRutas();
704    this.listarColonias();
705    this.listarCPs();
706    this.listarConsultas();
707  }
708
709  ngOnInit(): void {
710    this.listarRutas();
711    this.listarColonias();
712    this.listarCPs();
713    this.listarConsultas();
714  }
715
716  ngOnInit(): void {
717    this.listarRutas();
718    this.listarColonias();
719    this.listarCPs();
720    this.listarConsultas();
721  }
722
723  ngOnInit(): void {
724    this.listarRutas();
725    this.listarColonias();
726    this.listarCPs();
727    this.listarConsultas();
728  }
729
730  ngOnInit(): void {
731    this.listarRutas();
732    this.listarColonias();
733    this.listarCPs();
734    this.listarConsultas();
735  }
736
737  ngOnInit(): void {
738    this.listarRutas();
739    this.listarColonias();
740    this.listarCPs();
741    this.listarConsultas();
742  }
743
744  ngOnInit(): void {
745    this.listarRutas();
746    this.listarColonias();
747    this.listarCPs();
748    this.listarConsultas();
749  }
750
751  ngOnInit(): void {
752    this.listarRutas();
753    this.listarColonias();
754    this.listarCPs();
755    this.listarConsultas();
756  }
757
758  ngOnInit(): void {
759    this.listarRutas();
760    this.listarColonias();
761    this.listarCPs();
762    this.listarConsultas();
763  }
764
765  ngOnInit(): void {
766    this.listarRutas();
767    this.listarColonias();
768    this.listarCPs();
769    this.listarConsultas();
770  }
771
772  ngOnInit(): void {
773    this.listarRutas();
774    this.listarColonias();
775    this.listarCPs();
776    this.listarConsultas();
777  }
778
779  ngOnInit(): void {
780    this.listarRutas();
781    this.listarColonias();
782    this.listarCPs();
783    this.listarConsultas();
784  }
785
786  ngOnInit(): void {
787    this.listarRutas();
788    this.listarColonias();
789    this.listarCPs();
790    this.listarConsultas();
791  }
792
793  ngOnInit(): void {
794    this.listarRutas();
795    this.listarColonias();
796    this.listarCPs();
797    this.listarConsultas();
798  }
799
800  ngOnInit(): void {
801    this.listarRutas();
802    this.listarColonias();
803    this.listarCPs();
804    this.listarConsultas();
805  }
806
807  ngOnInit(): void {
808    this.listarRutas();
809    this.listarColonias();
810    this.listarCPs();
811    this.listarConsultas();
812  }
813
814  ngOnInit(): void {
815    this.listarRutas();
816    this.listarColonias();
817    this.listarCPs();
818    this.listarConsultas();
819  }
820
821  ngOnInit(): void {
822    this.listarRutas();
823    this.listarColonias();
824    this.listarCPs();
825    this.listarConsultas();
826  }
827
828  ngOnInit(): void {
829    this.listarRutas();
830    this.listarColonias();
831    this.listarCPs();
832    this.listarConsultas();
833  }
834
835  ngOnInit(): void {
836    this.listarRutas();
837    this.listarColonias();
838    this.listarCPs();
839    this.listarConsultas();
840  }
841
842  ngOnInit(): void {
843    this.listarRutas();
844    this.listarColonias();
845    this.listarCPs();
846    this.listarConsultas();
847  }
848
849  ngOnInit(): void {
850    this.listarRutas();
851    this.listarColonias();
852    this.listarCPs();
853    this.listarConsultas();
854  }
855
856  ngOnInit(): void {
857    this.listarRutas();
858    this.listarColonias();
859    this.listarCPs();
860    this.listarConsultas();
861  }
862
863  ngOnInit(): void {
864    this.listarRutas();
865    this.listarColonias();
866    this.listarCPs();
867    this.listarConsultas();
868  }
869
870  ngOnInit(): void {
871    this.listarRutas();
872    this.listarColonias();
873    this.listarCPs();
874    this.listarConsultas();
875  }
876
877  ngOnInit(): void {
878    this.listarRutas();
879    this.listarColonias();
880    this.listarCPs();
881    this.listarConsultas();
882  }
883
884  ngOnInit(): void {
885    this.listarRutas();
886    this.listarColonias();
887    this.listarCPs();
888    this.listarConsultas();
889  }
890
891  ngOnInit(): void {
892    this.listarRutas();
893    this.listarColonias();
894    this.listarCPs();
895    this.listarConsultas();
896  }
897
898  ngOnInit(): void {
899    this.listarRutas();
900    this.listarColonias();
901    this.listarCPs();
902    this.listarConsultas();
903  }
904
905  ngOnInit(): void {
906    this.listarRutas();
907    this.listarColonias();
908    this.listarCPs();
909    this.listarConsultas();
910  }
911
912  ngOnInit(): void {
913    this.listarRutas();
914    this.listarColonias();
915    this.listarCPs();
916    this.listarConsultas();
917  }
918
919  ngOnInit(): void {
920    this.listarRutas();
921    this.listarColonias();
922    this.listarCPs();
923    this.listarConsultas();
924  }
925
926  ngOnInit(): void {
927    this.listarRutas();
928    this.listarColonias();
929    this.listarCPs();
930    this.listarConsultas();
931  }
932
933  ngOnInit(): void {
934    this.listarRutas();
935    this.listarColonias();
936    this.listarCPs();
937    this.listarConsultas();
938  }
939
940  ngOnInit(): void {
941    this.listarRutas();
942    this.listarColonias();
943    this.listarCPs();
944    this.listarConsultas();
945  }
946
947  ngOnInit(): void {
948    this.listarRutas();
949    this.listarColonias();
950    this.listarCPs();
951    this.listarConsultas();
952  }
953
954  ngOnInit(): void {
955    this.listarRutas();
956    this.listarColonias();
957    this.listarCPs();
958    this.listarConsultas();
959  }
960
961  ngOnInit(): void {
962    this.listarRutas();
963    this.listarColonias();
964    this.listarCPs();
965    this.listarConsultas();
966  }
967
968  ngOnInit(): void {
969    this.listarRutas();
970    this.listarColonias();
971    this.listarCPs();
972    this.listarConsultas();
973  }
974
975  ngOnInit(): void {
976    this.listarRutas();
977    this.listarColonias();
978    this.listarCPs();
979    this.listarConsultas();
980  }
981
982  ngOnInit(): void {
983    this.listarRutas();
984    this.listarColonias();
985    this.listarCPs();
986    this.listarConsultas();
987  }
988
989  ngOnInit(): void {
990    this.listarRutas();
991    this.listarColonias();
992    this.listarCPs();
993    this.listarConsultas();
994  }
995
996  ngOnInit(): void {
997    this.listarRutas();
998    this.listarColonias();
999    this.listarCPs();
1000   this.listarConsultas();
1001  }
1002
1003  ngOnInit(): void {
1004    this.listarRutas();
1005    this.listarColonias();
1006    this.listarCPs();
1007    this.listarConsultas();
1008  }
1009
1010  ngOnInit(): void {
1011    this.listarRutas();
1012    this.listarColonias();
1013    this.listarCPs();
1014    this.listarConsultas();
1015  }
1016
1017  ngOnInit(): void {
1018    this.listarRutas();
1019    this.listarColonias();
1020    this.listarCPs();
1021    this.listarConsultas();
1022  }
1023
1024  ngOnInit(): void {
1025    this.listarRutas();
1026    this.listarColonias();
1027    this.listarCPs();
1028    this.listarConsultas();
1029  }
1030
1031  ngOnInit(): void {
1032    this.listarRutas();
1033    this.listarColonias();
1034    this.listarCPs();
1035    this.listarConsultas();
1036  }
1037
1038  ngOnInit(): void {
1039    this.listarRutas();
1040    this.listarColonias();
1041    this.listarCPs();
1042    this.listarConsultas();
1043  }
1044
1045  ngOnInit(): void {
1046    this.listarRutas();
1047    this.listarColonias();
1048    this.listarCPs();
1049    this.listarConsultas();
1050  }
1051
1052  ngOnInit(): void {
1053    this.listarRutas();
1054    this.listarColonias();
1055    this.listarCPs();
1056    this.listarConsultas();
1057  }
1058
1059  ngOnInit(): void {
1060    this.listarRutas();
1061    this.listarColonias();
1062    this.listarCPs();
1063    this.listarConsultas();
1064  }
1065
1066  ngOnInit(): void {
1067    this.listarRutas();
1068    this.listarColonias();
1069    this.listarCPs();
1070    this.listarConsultas();
1071  }
1072
1073  ngOnInit(): void {
1074    this.listarRutas();
1075    this.listarColonias();
1076    this.listarCPs();
1077    this.listarConsultas();
1078  }
1079
1080  ngOnInit(): void {
1081    this.listarRutas();
1082    this.listarColonias();
1083    this.listarCPs();
1084    this.listarConsultas();
1085  }
1086
1087  ngOnInit(): void {
1088    this.listarRutas();
1089    this.listarColonias();
1090    this.listarCPs();
1091    this.listarConsultas();
1092  }
1093
1094  ngOnInit(): void {
1095    this.listarRutas();
1096    this.listarColonias();
1097    this.listarCPs();
1098    this.listarConsultas();
1099  }
1100
1101  ngOnInit(): void {
1102    this.listarRutas();
1103    this.listarColonias();
1104    this.listarCPs();
1105    this.listarConsultas();
1106  }
1107
1108  ngOnInit(): void {
1109    this.listarRutas();
1110    this.listarColonias();
1111    this.listarCPs();
1112    this.listarConsultas();
1113  }
1114
1115  ngOnInit(): void {
1116    this.listarRutas();
1117    this.listarColonias();
1118    this.listarCPs();
1119    this.listarConsultas();
1120  }
1121
1122  ngOnInit(): void {
1123    this.listarRutas();
1124    this.listarColonias();
1125    this.listarCPs();
1126    this.listarConsultas();
1127  }
1128
1129  ngOnInit(): void {
1130    this.listarRutas();
1131    this.listarColonias();
1132    this.listarCPs();
1133    this.listarConsultas();
1134  }
1135
1136  ngOnInit(): void {
1137    this.listarRutas();
1138    this.listarColonias();
1139    this.listarCPs();
1140    this.listarConsultas();
1141  }
1142
1143  ngOnInit(): void {
1144    this.listarRutas();
1145    this.listarColonias();
1146    this.listarCPs();
1147    this.listarConsultas();
1148  }
1149
1150  ngOnInit(): void {
1151    this.listarRutas();
1152    this.listarColonias();
1153    this.listarCPs();
1154    this.listarConsultas();
1155  }
1156
1157  ngOnInit(): void {
1158    this.listarRutas();
1159    this.listarColonias();
1160    this.listarCPs();
1161    this.listarConsultas();
1162  }
1163
1164  ngOnInit(): void {
1165    this.listarRutas();
1166    this.listarColonias();
1167    this.listarCPs();
1168    this.listarConsultas();
1169  }
1170
1171  ngOnInit(): void {
1172    this.listarRutas();
1173    this.listarColonias();
1174    this.listarCPs();
1175    this.listarConsultas();
1176  }
1177
1178  ngOnInit(): void {
1179    this.listarRutas();
1180    this.listarColonias();
1181    this.listarCPs();
1182    this.listarConsultas();
1183  }
1184
1185  ngOnInit(): void {
1186    this.listarRutas();
1187    this.listarColonias();
1188    this.listarCPs();
1189    this.listarConsultas();
1190  }
1191
1192  ngOnInit(): void {
1193    this.listarRutas();
1194    this.listarColonias();
1195    this.listarCPs();
1196    this.listarConsultas();
1197  }
1198
1199  ngOnInit(): void {
1200    this.listarRutas();
1201    this.listarColonias();
1202    this.listarCPs();
1203    this.listarConsultas();
1204  }
1205
1206  ngOnInit(): void {
1207    this.listarRutas();
1208    this.listarColonias();
1209    this.listarCPs();
1210    this.listarConsultas();
1211  }
1212
1213  ngOnInit(): void {
1214    this.listarRutas();
1215    this.listarColonias();
1216    this.listarCPs();
1217    this.listarConsultas();
1218  }
1219
1220  ngOnInit(): void {
1221    this.listarRutas();
1222    this.listarColonias();
1223    this.listarCPs();
1224    this.listarConsultas();
1225  }
1226
1227  ngOnInit(): void {
1228    this.listarRutas();
1229    this.listarColonias();
1230    this.listarCPs();
1231    this.listarConsultas();
1232  }
1233
1234  ngOnInit(): void {
1235    this.listarRutas();
1236    this.listarColonias();
1237    this.listarCPs();
1238    this.listarConsultas();
1239  }
1240
1241  ngOnInit(): void {
1242    this.listarRutas();
1243    this.listarColonias();
1244    this.listarCPs();
1245    this.listarConsultas();
1246  }
1247
1248  ngOnInit(): void {
1249    this.listarRutas();
1250    this.listarColonias();
1251    this.listarCPs();
1252    this.listarConsultas();
1253  }
1254
1255  ngOnInit(): void {
1256    this.listarRutas();
1257    this.listarColonias();
1258    this.listarCPs();
1259    this.listarConsultas();
1260  }
1261
1262  ngOnInit(): void {
1263    this.listarRutas();
1264    this.listarColonias();
1265    this.listarCPs();
1266    this.listarConsultas();
1267  }
1268
1269  ngOnInit(): void {
1270    this.listarRutas();
1271    this.listarColonias();
1272    this.listarCPs
```

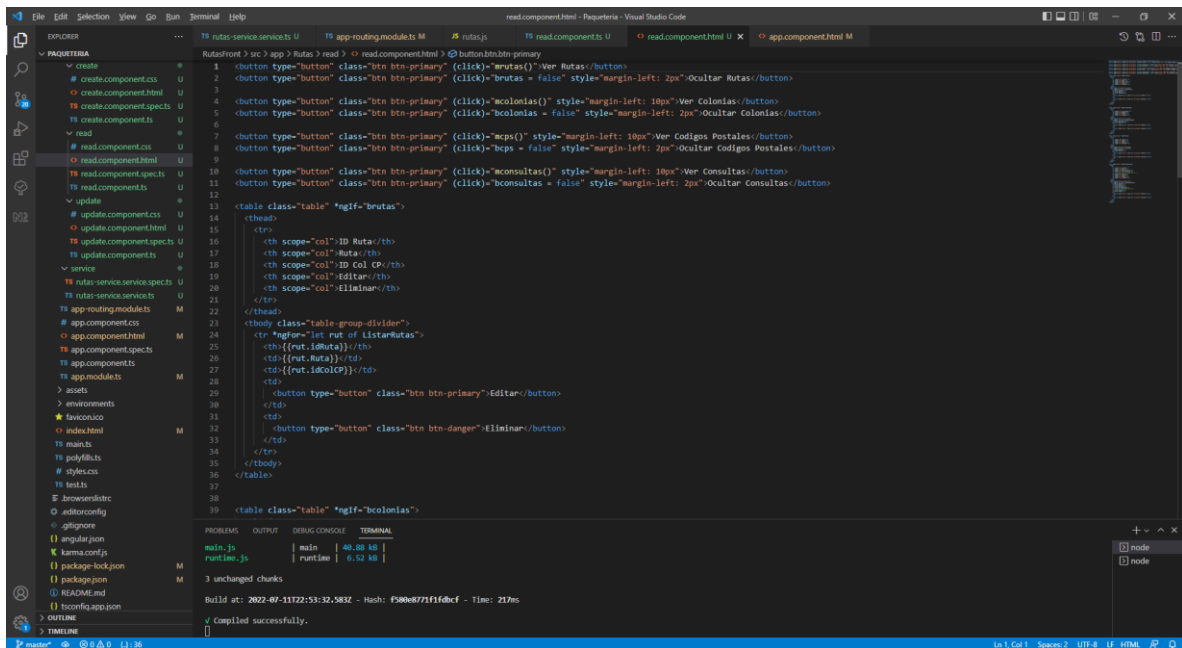
```

36 mcolonias() {
37   this.bautas=false;
38   this.bcolonias=true;
39   this.bcpis=false;
40   this.bconsultas=false;
41 }
42 mcpis() {
43   this.bautas=false;
44   this.bcolonias=false;
45   this.bcpis=true;
46   this.bconsultas=false;
47 }
48 mconsultas() {
49   this.bautas=false;
50   this.bcolonias=false;
51   this.bcpis=false;
52   this.bconsultas=true;
53 }
54 listarRutas() {
55   this.RutasService.getRutas().subscribe(
56     res=>{
57       console.log(res);
58       this.listarRutas=<any> res;
59       this.listarCPis=<any> res;
60       this.listarColonias=<any> res;
61     },
62     err=>console.log(err)
63   );
64 }
65 listarColonias() {
66   this.RutasService.getColonias().subscribe(
67     res=>{
68       console.log(res);
69       this.listarColonias=<any> res;
70     },
71     err=>console.log(err)
72   );
73 }
74
75 listarCPis() {
76   this.RutasService.getCPis().subscribe(
77     res=>{
78       console.log(res);
79       this.listarCPis=<any> res;
80     },
81     err=>console.log(err)
82   );
83 }
84 listarConsultas() {
85   this.RutasService.getConsultas().subscribe(
86     res=>{
87       console.log(res);
88       this.listarConsultas=<any> res;
89     },
90     err=>console.log(err)
91   );
92 }
93
94
95
96
97

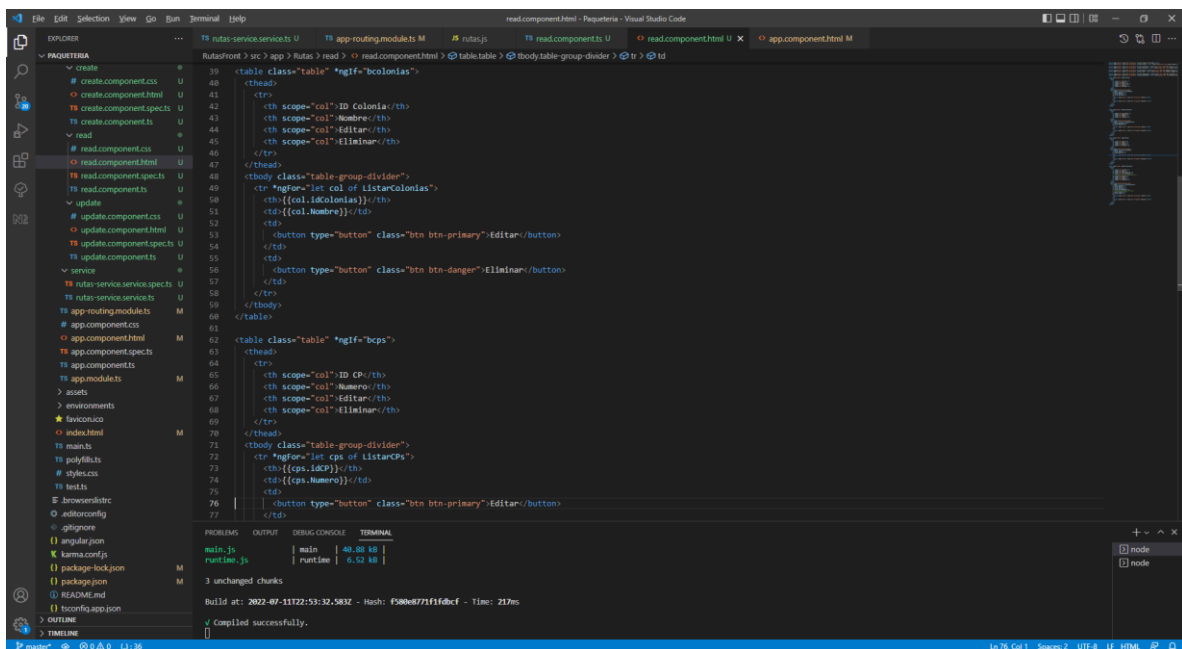
```

Podemos ver que en mi código tengo *banderas* para ocultar las tablas y volver a mostrarlas.

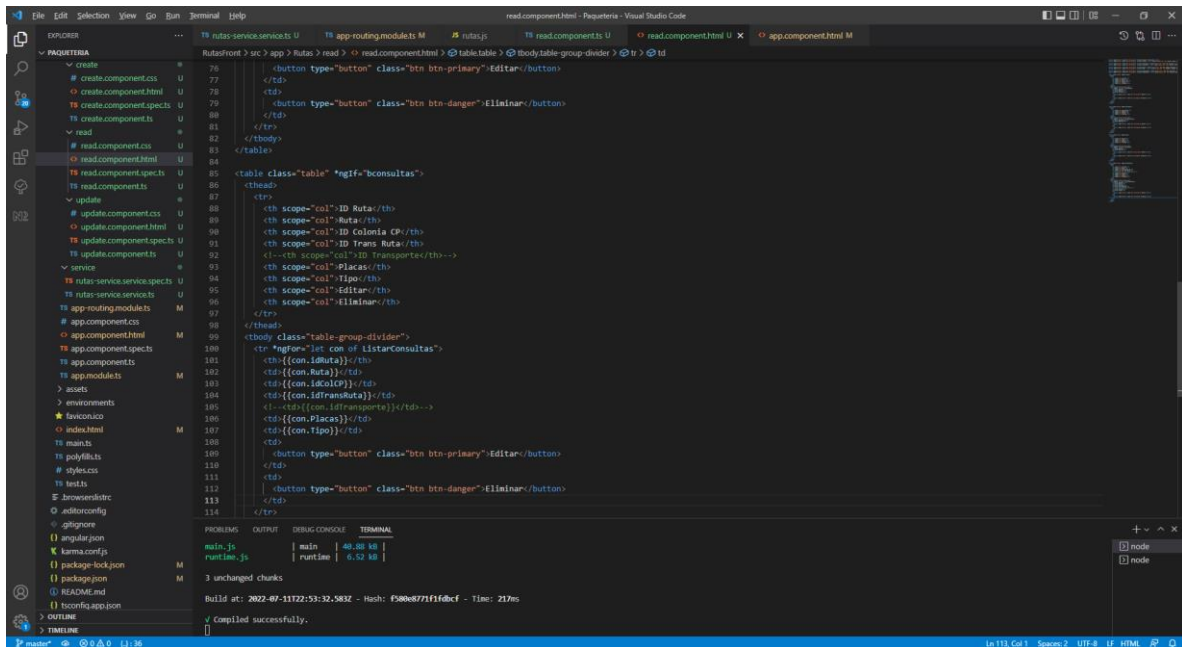
5.- Para finalizar checamos nuestro archivo HTML de nuestro mismo componente, en este debemos de tener las tablas



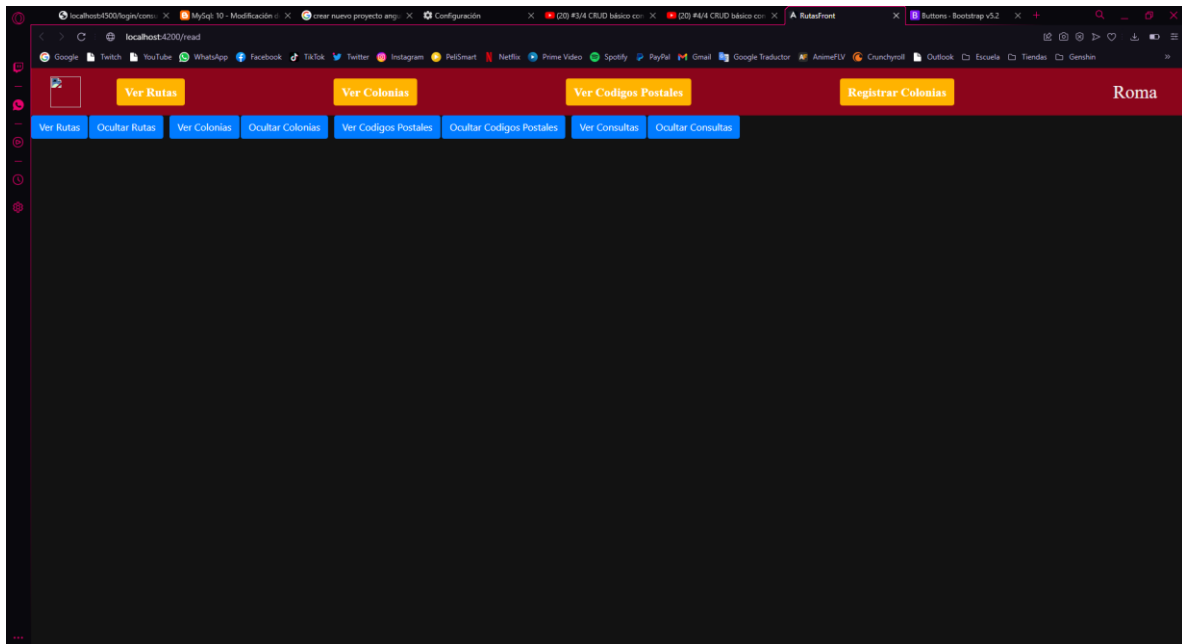
```
1 <button type="button" class="btn btn-primary" (click)="brutas()">Ver Rutas</button>
2 <button type="button" class="btn btn-primary" (click)="brutas = false" style="margin-left: 2px">Ocultar Rutas</button>
3
4 <button type="button" class="btn btn-primary" (click)="bcolonias()" style="margin-left: 10px">Ver Colonias</button>
5 <button type="button" class="btn btn-primary" (click)="bcolonias = false" style="margin-left: 2px">Ocultar Colonias</button>
6
7 <button type="button" class="btn btn-primary" (click)="mcpis()" style="margin-left: 10px">Ver Códigos Postales</button>
8 <button type="button" class="btn btn-primary" (click)="mcpis = false" style="margin-left: 2px">Ocultar Códigos Postales</button>
9
10 <button type="button" class="btn btn-primary" (click)="mconsultas()" style="margin-left: 10px">Ver Consultas</button>
11 <button type="button" class="btn btn-primary" (click)="bconsultas = false" style="margin-left: 2px">Ocultar Consultas</button>
12
13 <table class="table" *ngIf="brutas">
14   <thead>
15     <tr>
16       <th scope="col">ID Ruta</th>
17       <th scope="col">Ruta</th>
18       <th scope="col">ID Col CP</th>
19       <th scope="col">Editar</th>
20       <th scope="col">Eliminar</th>
21     </tr>
22   </thead>
23   <tbody class="table-group-divider">
24     <tr *ngFor="let rut of listarRutas">
25       <td>{{rut.idRuta}}</td>
26       <td>{{rut.Ruta}}</td>
27       <td>{{rut.idColCP}}</td>
28       <td>
29         <button type="button" class="btn btn-primary">Editar</button>
30       </td>
31       <td>
32         <button type="button" class="btn btn-danger">Eliminar</button>
33       </td>
34     </tr>
35   </tbody>
36 </table>
37
38 <table class="table" *ngIf="bcolonias">
39   <thead>
40     <tr>
41       <th scope="col">ID Colonia</th>
42       <th scope="col">Nombre</th>
43       <th scope="col">Editar</th>
44       <th scope="col">Eliminar</th>
45     </tr>
46   </thead>
47   <tbody class="table-group-divider">
48     <tr *ngFor="let col of listarColonias">
49       <td>{{col.idColonias}}</td>
50       <td>{{col.Nombre}}</td>
51       <td>
52         <button type="button" class="btn btn-primary">Editar</button>
53       </td>
54       <td>
55         <button type="button" class="btn btn-danger">Eliminar</button>
56       </td>
57     </tr>
58   </tbody>
59 </table>
60
61 <table class="table" *ngIf="bcps">
62   <thead>
63     <tr>
64       <th scope="col">ID CP</th>
65       <th scope="col">Numero</th>
66       <th scope="col">Editar</th>
67       <th scope="col">Eliminar</th>
68     </tr>
69   </thead>
70   <tbody class="table-group-divider">
71     <tr *ngFor="let cps of listarCps">
72       <td>{{cps.idCP}}</td>
73       <td>{{cps.Numero}}</td>
74       <td>
75         <button type="button" class="btn btn-primary">Editar</button>
76       </td>
77     </tr>
78   </tbody>
79 </table>
```

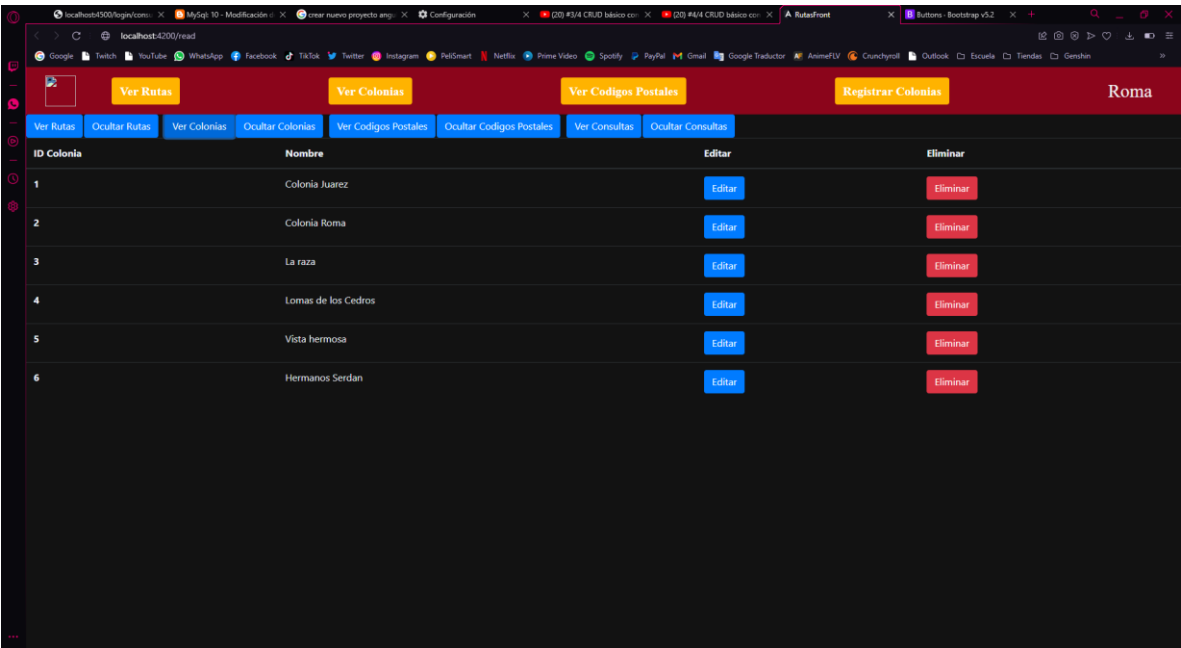
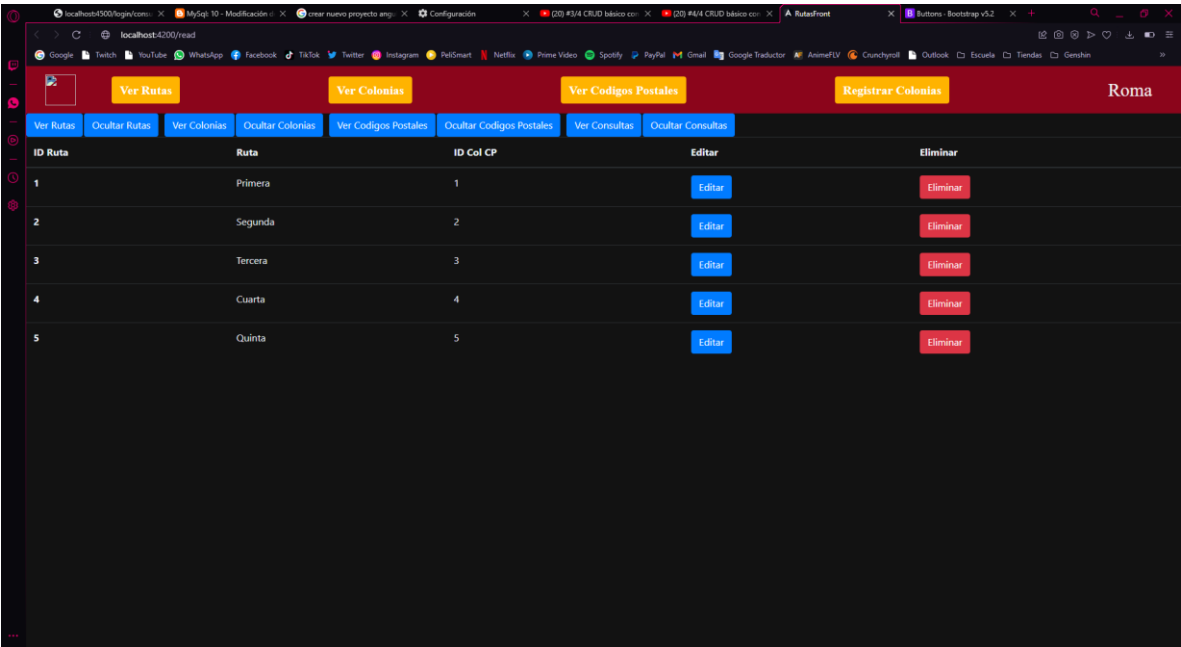


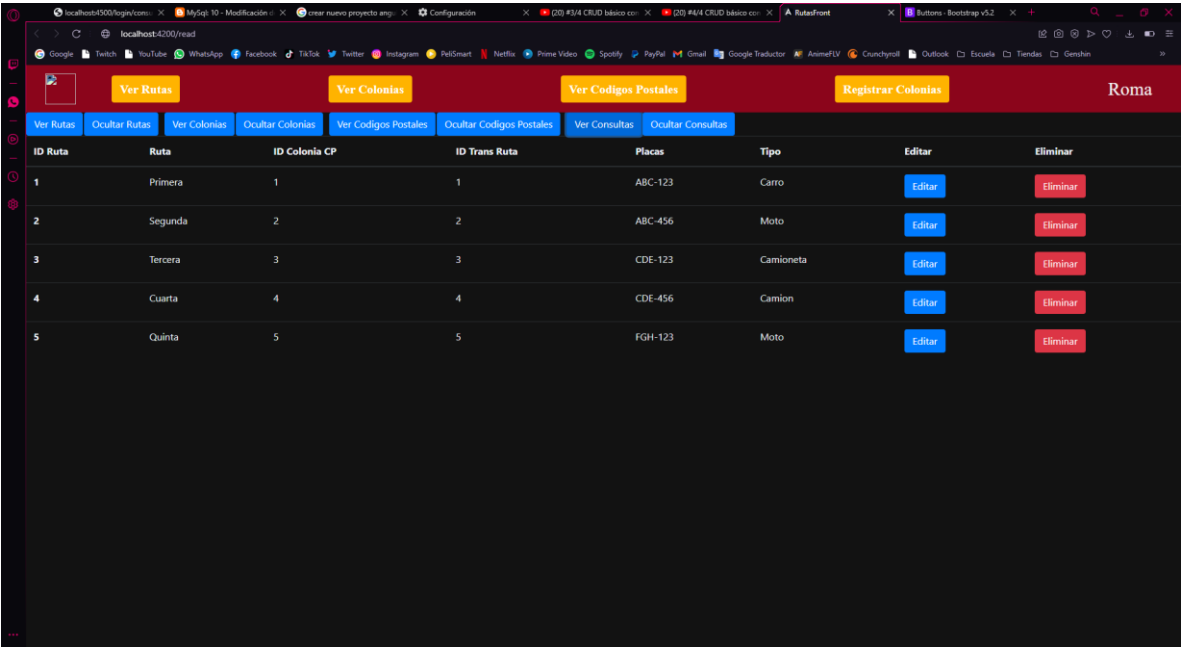
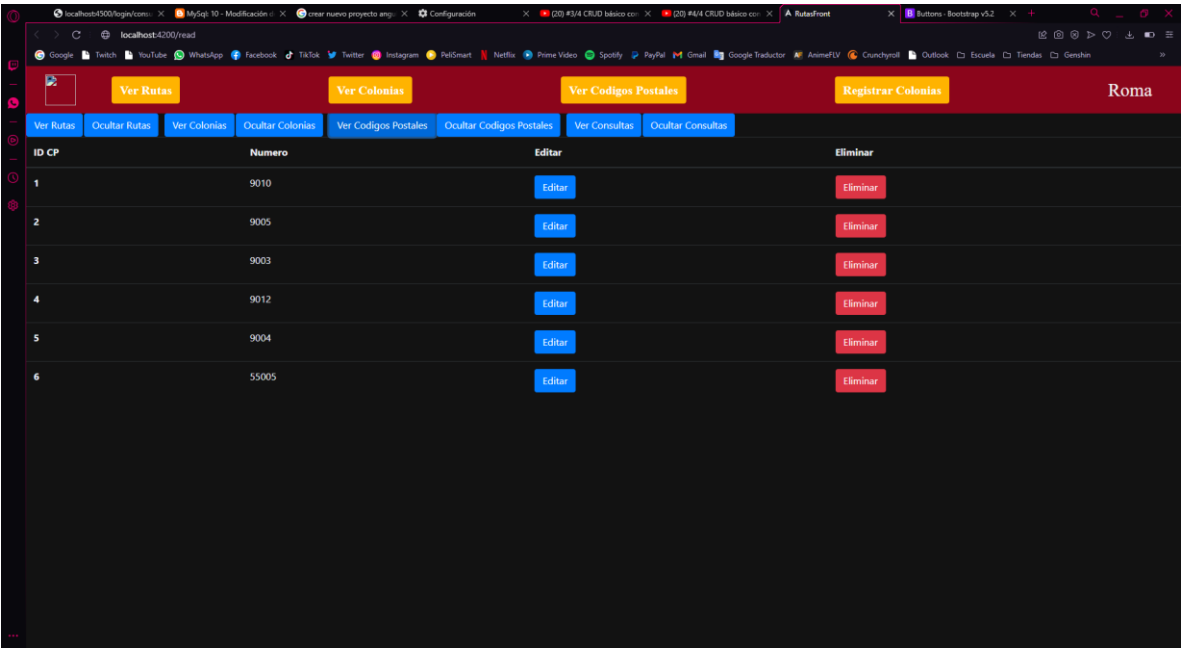
```
39 <table class="table" *ngIf="bcolonias">
40   <thead>
41     <tr>
42       <th scope="col">ID Colonia</th>
43       <th scope="col">Nombre</th>
44       <th scope="col">Editar</th>
45       <th scope="col">Eliminar</th>
46     </tr>
47   </thead>
48   <tbody class="table-group-divider">
49     <tr *ngFor="let col of listarColonias">
50       <td>{{col.idColonias}}</td>
51       <td>{{col.Nombre}}</td>
52       <td>
53         <button type="button" class="btn btn-primary">Editar</button>
54       </td>
55       <td>
56         <button type="button" class="btn btn-danger">Eliminar</button>
57       </td>
58     </tr>
59   </tbody>
60 </table>
61
62 <table class="table" *ngIf="bcps">
63   <thead>
64     <tr>
65       <th scope="col">ID CP</th>
66       <th scope="col">Numero</th>
67       <th scope="col">Editar</th>
68       <th scope="col">Eliminar</th>
69     </tr>
70   </thead>
71   <tbody class="table-group-divider">
72     <tr *ngFor="let cps of listarCps">
73       <td>{{cps.idCP}}</td>
74       <td>{{cps.Numero}}</td>
75       <td>
76         <button type="button" class="btn btn-primary">Editar</button>
77       </td>
78     </tr>
79   </tbody>
80 </table>
```



Ahora veremos como queda en nuestro *Front-End*

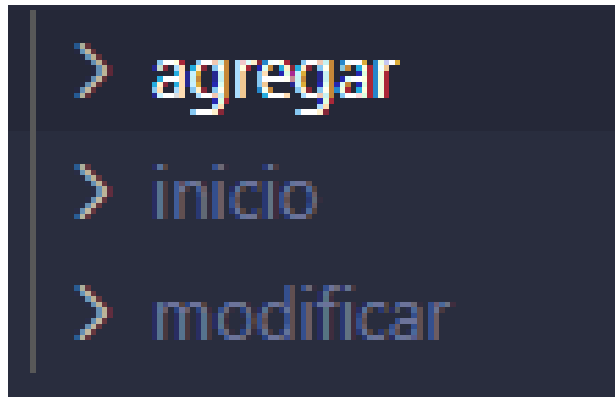






Hernández Cervantes Geovanni

Para la realización de esta practica primeramente tendremos que tener 3 componentes que nos permitaran agregar, modificar y mostrar todos nuestros paquetes



Podemos ver que tenemos de componentes *que son agregar, inicio y modificar*.

2.- Después crearemos nuestro *service* que tendremos nuestros métodos en el, para crear el service ingresamos el siguiente comando: **ng g s service/Paquetes-service**

Cuando se haya creado este servicio se deberá de crear rutas que permitan navegar por ellas para así crear los html correspondientes.

```
const routes: Routes = [  
  {path:'',redirectTo:'/inicio',pathMatch:'full'},  
  {path:'inicio',component: InicioComponent},  
  {path:'add',component: AgregarComponent},  
  {path:'edit/:id',component: ModificarComponent}  
];
```

3.-Se deberá de agregar las librerias poder realizar las consultas e importar el HttpClientModule y el Forms para poder visualizar los formularios.

```
import {HttpClientModule} from '@angular/common/http';  
import {FormsModule} from '@angular/forms';
```

4.- Ahora en nuestro servicio deberemos de importar lo que hemos agregado en nuestro app module , que fue el httpClient

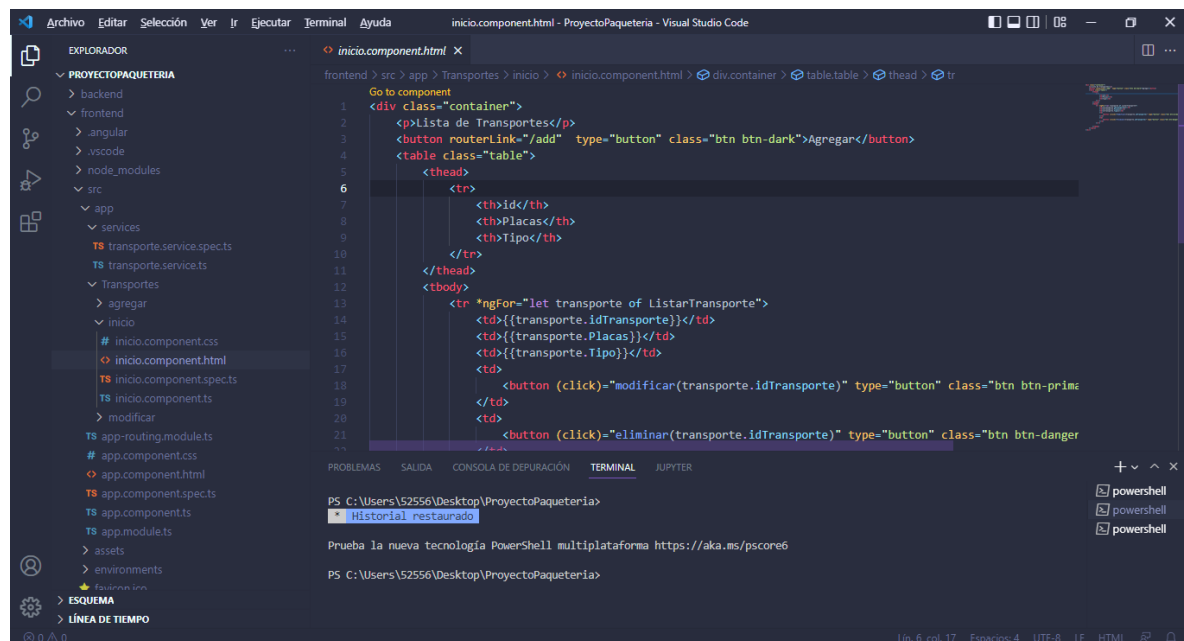
```
import { HttpClient } from '@angular/common/http'
```

5.- Una vez importado podemos agregar una url de nuestro servicio en donde se encuentra corriendo exactamente para poder conectar la base de datos a nuestro frontend.

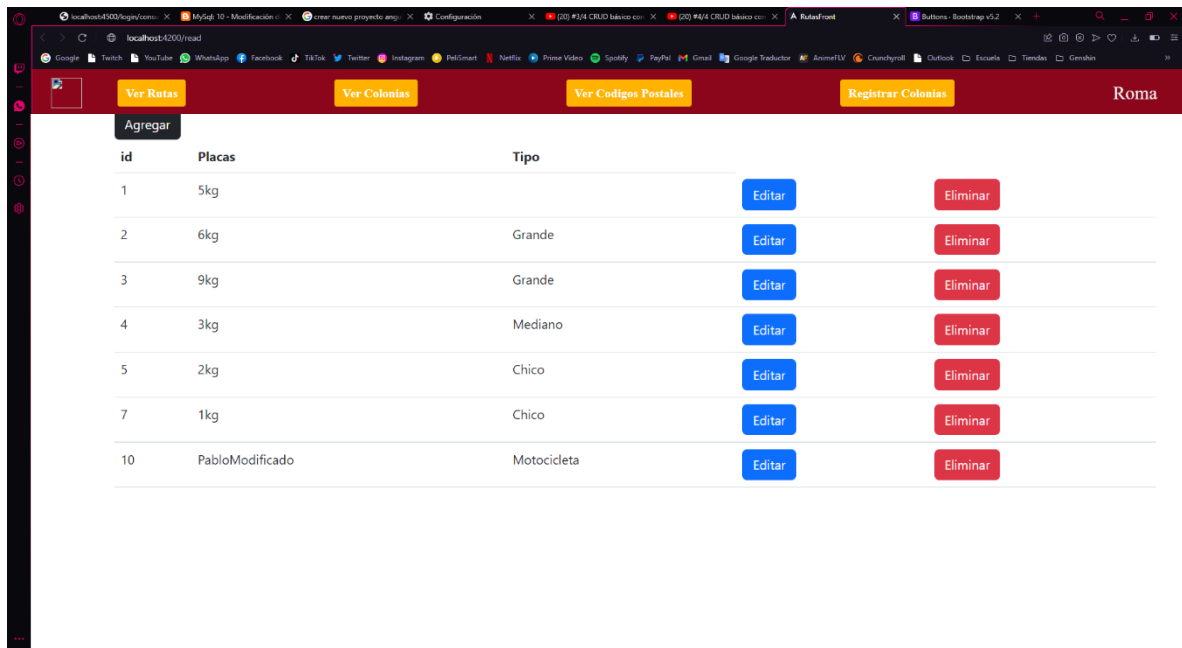
```
url="http://localhost:4500/paqueteria/paquetes;  
constructor(private http: HttpClient) { }
```

6. Ahora deberemos de crear un método Get en nuestro archivo inicio.ts y tendremos que inicializar el servicio y utilizar el método que estamos llamando para que realice la consulta y nos muestre los datos en el frontend

8. Una vez realizado todo esto, ya podremos agregar estilo a nuestro HTML para que nos lo muestre de forma correcta nuestros datos de la base de datos, para ello modificamos nuestro inicio.html y quedaría así.

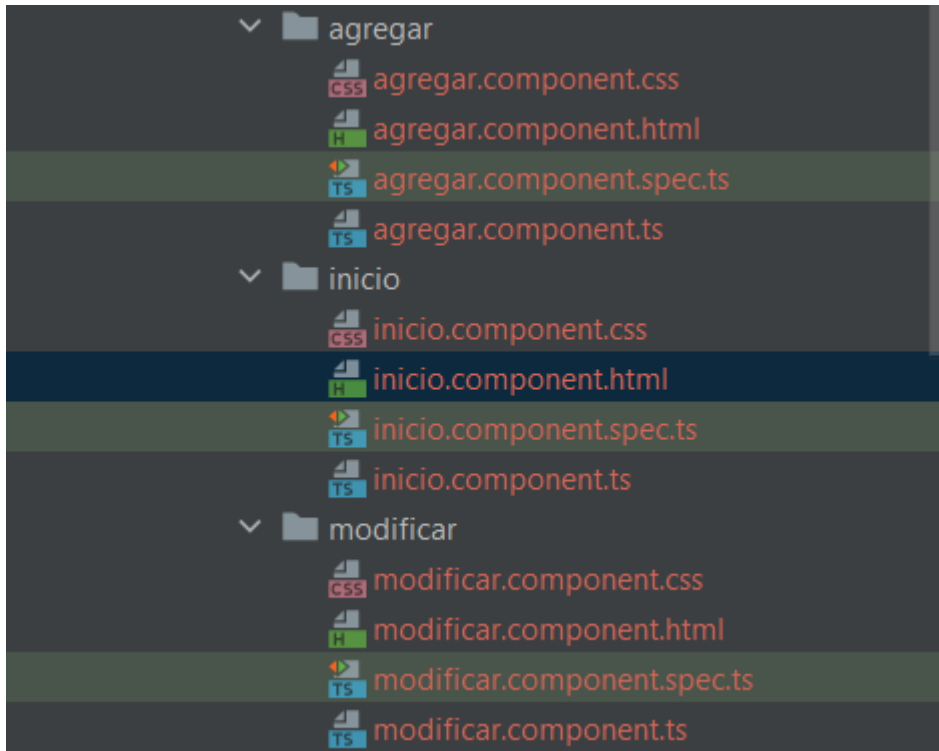


Ahora veremos como queda en nuestro *Front-End*

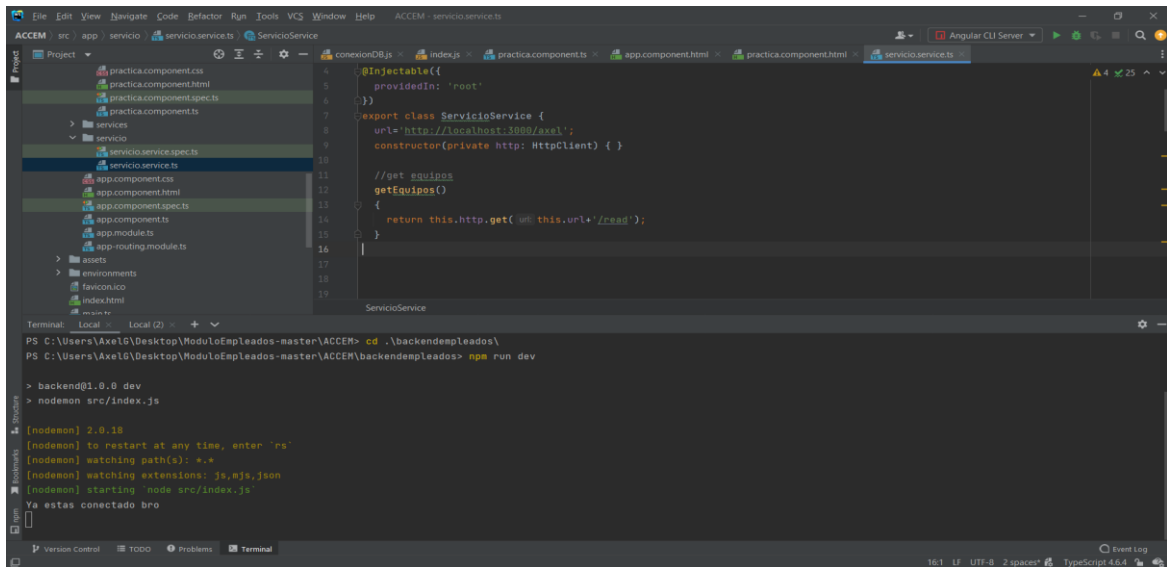


Rodriguez Fuentes Axel Roman

Primeramente haremos nuestro componentes que serán el agregar, modificar y hacer consultas.



2.- Después crearemos nuestro servicio donde contendrán los métodos para poder hacer las altas y bajas.



```

@Injectable({
  providedIn: 'root'
})
export class ServicioService {
  url = 'http://localhost:3000/axel';
  constructor(private http: HttpClient) {}

  //get equipos
  getEquipos() {
    return this.http.get(`${this.url}/read`);
  }
}

```

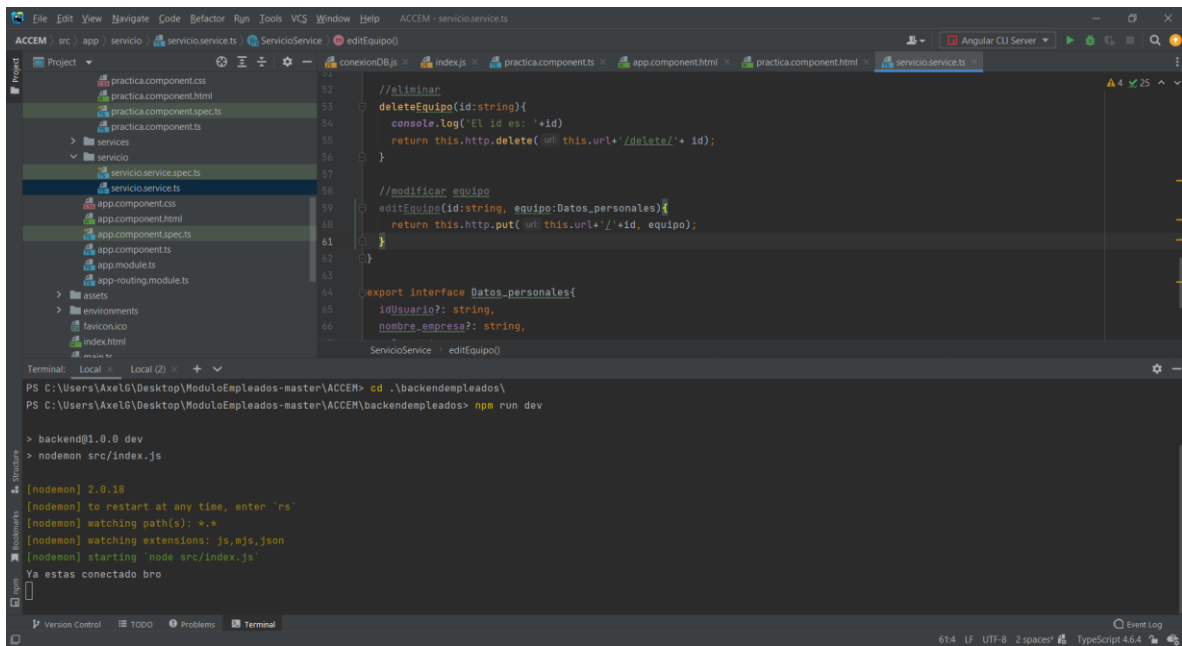
```

PS C:\Users\AxelG\Desktop\ModuloEmpleados-master\ACCEN> cd .\backendeempleados\
PS C:\Users\AxelG\Desktop\ModuloEmpleados-master\ACCEN\backendeempleados> npm run dev

> backend@1.0.0 dev
> nodemon src/index.js

[nodemon] 2.0.18
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Ya estas conectado bro

```



```

//eliminar
deleteEquipo(id:string){
  console.log('El id es: ' + id);
  return this.http.delete(`${this.url}/delete/${id}`);
}

//modificar equipo
editEquipo(id:string, equipo:Datos_personales){
  return this.http.put(`${this.url}/${id}`, equipo);
}

```

```

PS C:\Users\AxelG\Desktop\ModuloEmpleados-master\ACCEN> cd .\backendeempleados\
PS C:\Users\AxelG\Desktop\ModuloEmpleados-master\ACCEN\backendeempleados> npm run dev

> backend@1.0.0 dev
> nodemon src/index.js

[nodemon] 2.0.18
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Ya estas conectado bro

```

3.- Después en nuestros componentes de altas y bajas agregaremos los comandos para su correcto funcionamiento

```

ruta.post( path: '/agregar', handlers: (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> )=>{
    const {idUserio, nombre_empresa, cp, Correo, Contraseña} = req.body
    let sql = 'insert into Datos_personales( idUsuario, nombre_empresa, cp, Correo, Contraseña) VALUES ('${idUserio}', '${nombre_empresa}', '${cp}', '${C
    conexion.query(sql,(err, rows, fields)=>{
        if(err) throw err
        else{
            res.json( body: {status: 'usuario agregado'})
        }
    })
});

ruta.delete( path:('/:id', handlers: (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> )=>{
    const{id} = req.params
    let sql = 'delete from datos_personales where idUsuario = '${id}:'
    conexion.query(sql, (err, rows, fields)=>{
        if(err) throw err
        else{
            res.json( body: {status: 'Usuario eliminado'})
        }
    })
});

```

4.- En nuestro HTML donde recopilamos los datos y las tablas de los datos.

```

1 <div *ngIf="robot == true">
2   <p></p>
3 </div>
4 <div *ngIf="robot == false">
5   <p></p>
6 </div>
7
8
9
10
11 <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #80518; font-size:12pt; color:#f2f2">
12   <div class="container-fluid" style="background-color: #80518; color:#f2f2">
13     <a class="navbar-brand" routerLink="/welcome" style="color:#f2f2">
14       
15     </a>
16     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
17       <span class="navbar-toggler-icon"></span>
18     </button>
19     <button routerLink="/profile" class="btn btn-dark-success me-2" type="button" style="background-color: #80518; color:#f2f2">Haz un envio</button>
20     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
21       <span class="navbar-toggler-icon"></span>
22     </button>
23     <button routerLink="/packages" class="btn btn-dark-success me-2" type="button" style="background-color: #80518; color:#f2f2">Ver envios</button>
24   </div>
25 </nav>
26
27 <div>
28   <p></p>
29 </div>
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
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

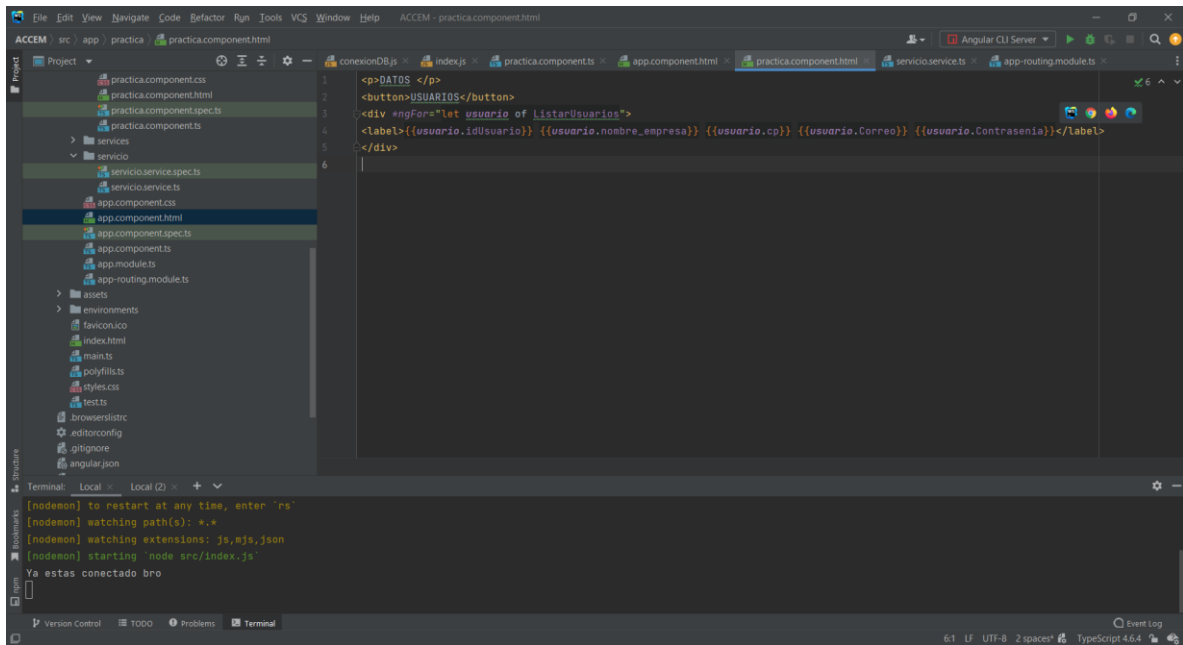
```

Terminal: Local x Local (2) x +

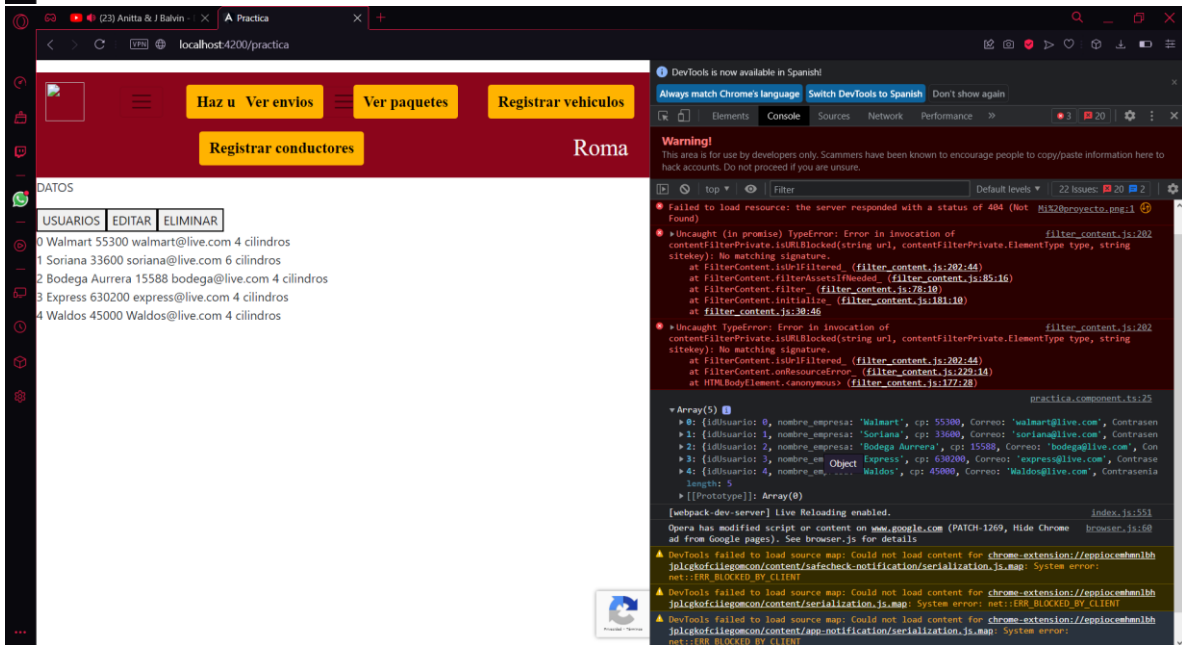
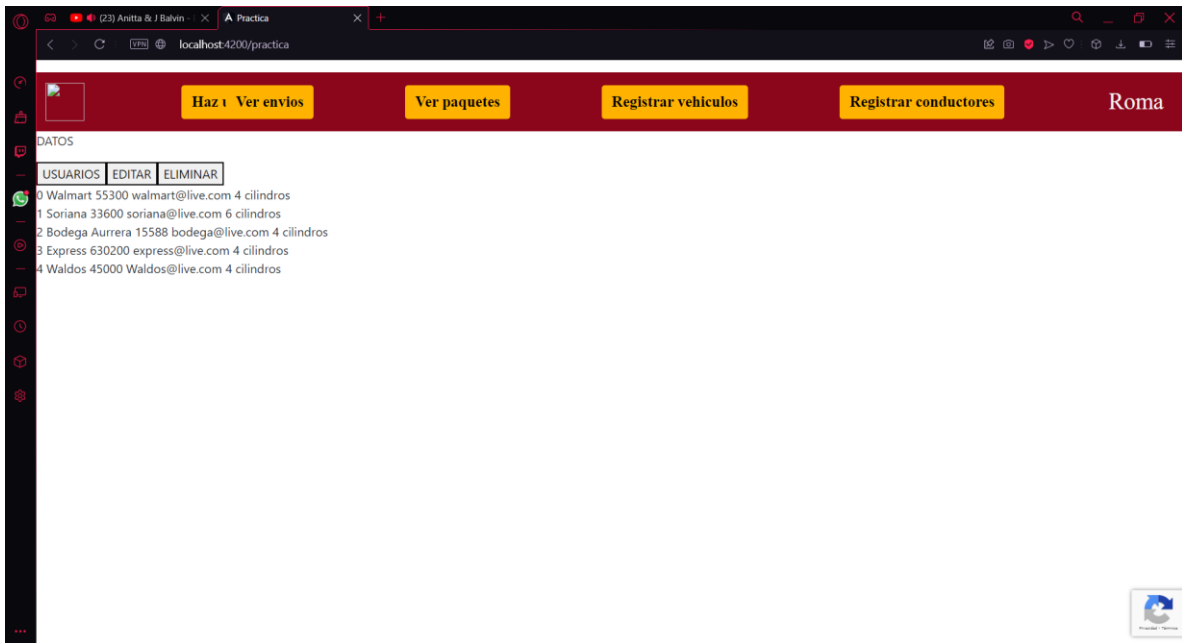
```

[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Ya estas conectado bro

```



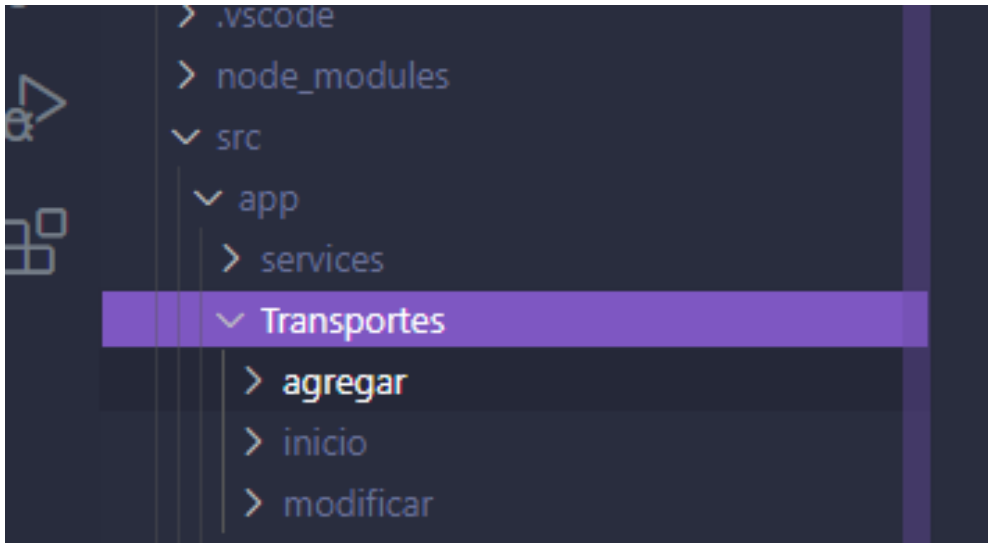
RESULTADOS FRONTEND.



Ruiz Teodocio Jose Pablo – Registrar Transporte

Bueno primeramente para mostrar los datos de nuestra base de datos en nuestro Frontend deberemos de realizar lo siguiente

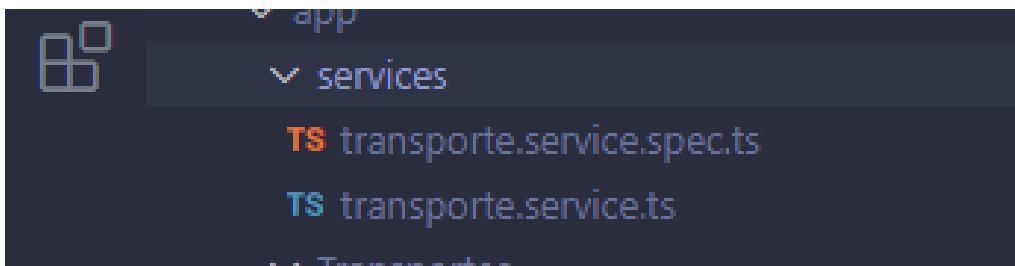
1.- Debemos de tener nuestros componentes que se encargaran para altas, modificaciones y consultas



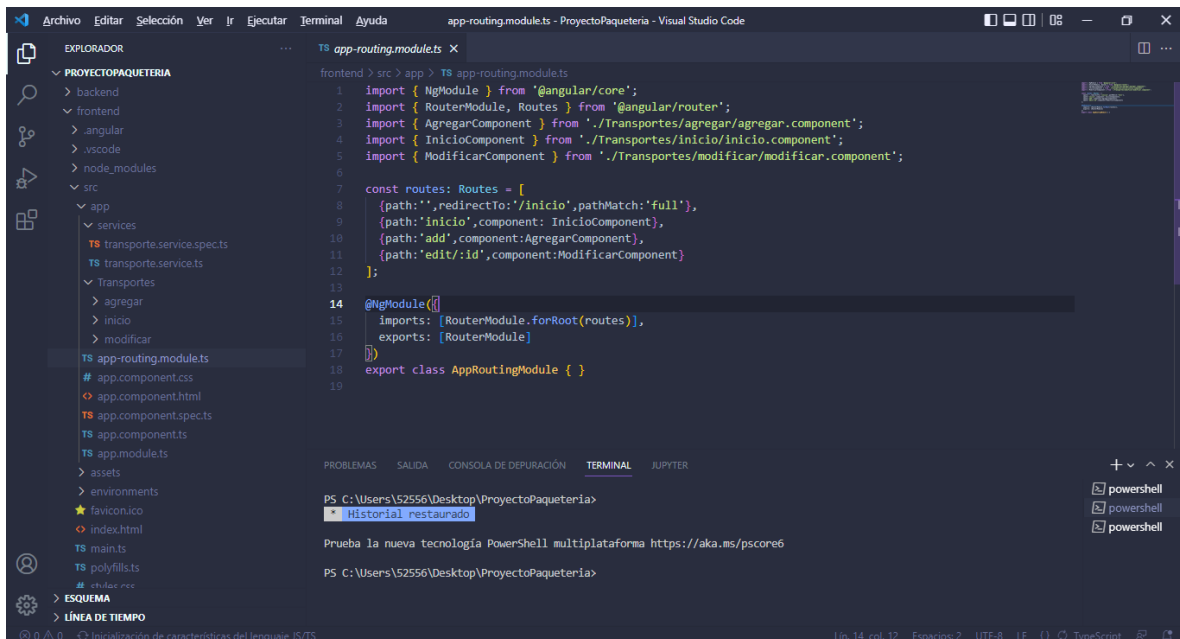
Podemos ver que tenemos de componentes *que son agregar, inicio y modificar. Estos componentes nos servirán para ahí colocar las rutas que nos permitan realizar dichos cambios.*

2.- Después crearemos nuestro *service* que tendremos nuestros métodos en el, para crear el service ingresamos el siguiente comando:

ng g s service/Transporte-service:

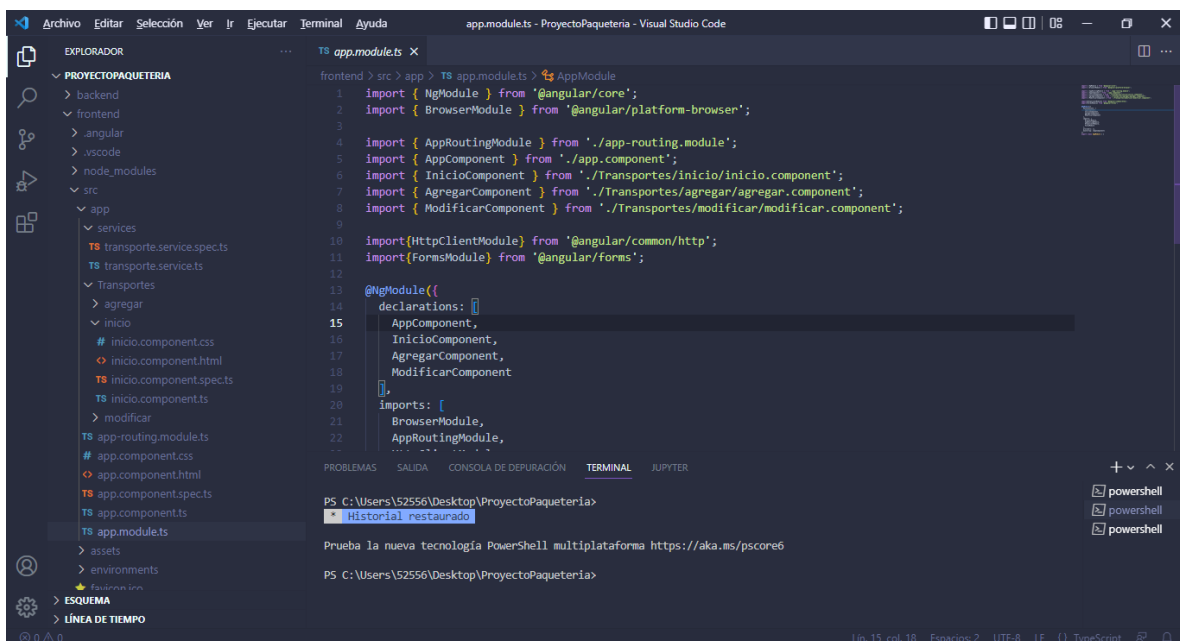


3.- Una vez creado esto debemos de verificar nuestras rutas que tenemos en nuestro backend, para poder realizar el código en nuestro ts.



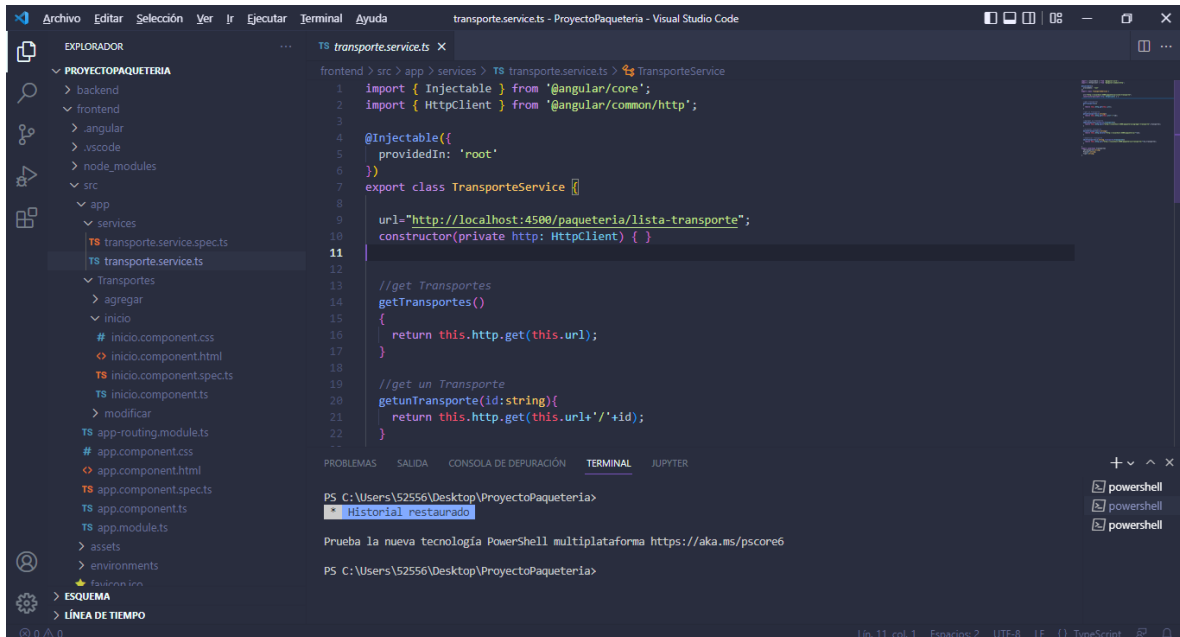
```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { AgregarComponent } from '../Transportes/agregar/agregar.component';
4 import { InicioComponent } from '../Transportes/inicio/inicio.component';
5 import { ModificarComponent } from '../Transportes/modificar/modificar.component';
6
7 const routes: Routes = [
8   {path:'',redirectTo:'/inicio', pathMatch:'full'},
9   {path:'inicio', component: InicioComponent},
10  {path:'add', component: AgregarComponent},
11  {path:'edit/:id', component: ModificarComponent}
12 ];
13
14 @NgModule({
15   imports: [RouterModule.forRoot(routes)],
16   exports: [RouterModule]
17 })
18 export class AppRoutingModule { }
```

4.- Ahora deberemos de agregar código en nuestro app.module.ts, este código será para poder realizar las consultas e importar el HttpClientModule y el Forms para poder visualizar los formulario, todo esto para utilizar api rest.



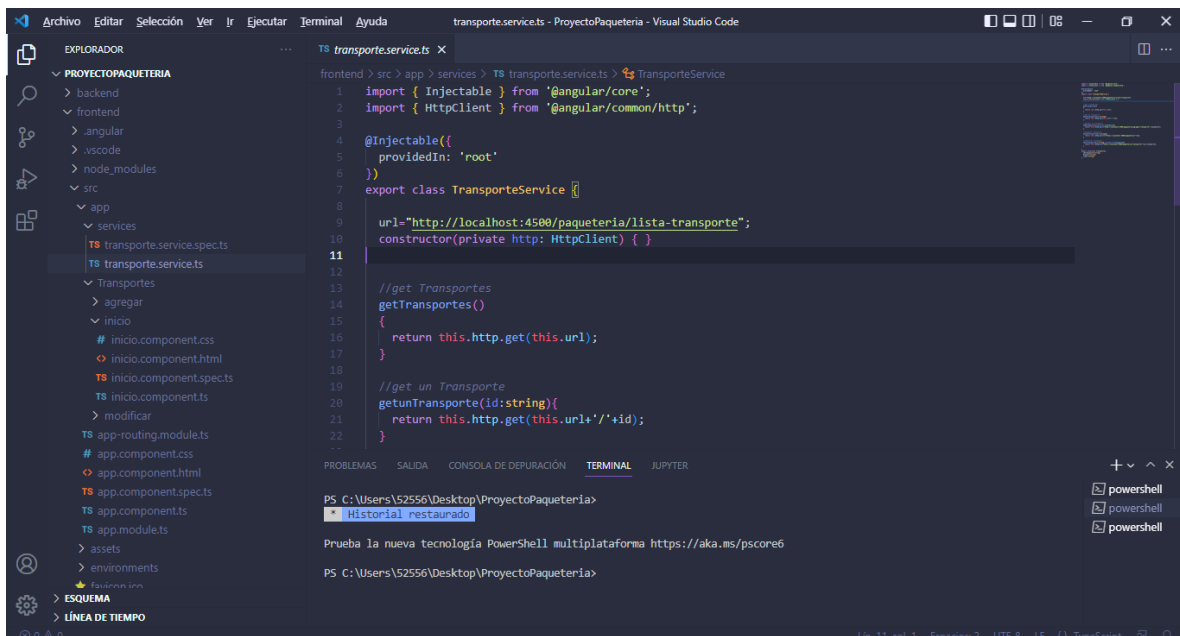
```
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { InicioComponent } from '../Transportes/inicio/inicio.component';
7 import { AgregarComponent } from '../Transportes/agregar/agregar.component';
8 import { ModificarComponent } from '../Transportes/modificar/modificar.component';
9
10 import { HttpClientModule } from '@angular/common/http';
11 import { FormsModule } from '@angular/forms';
12
13 @NgModule({
14   declarations: [
15     AppComponent,
16     InicioComponent,
17     AgregarComponent,
18     ModificarComponent
19   ],
20   imports: [
21     BrowserModule,
22     AppRoutingModule,
23     HttpClientModule,
24     FormsModule
25   ],
26   providers: [],
27   bootstrap: [AppComponent]
28 })
29 export class AppModule { }
```

5.- Ahora en nuestro servicio deberemos de importar lo que hemos agregado en nuestro app module , que fue el httpClient



```
transport.service.ts
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class TransporteService {
8
9   url="http://localhost:4500/paqueria/lista-transporte";
10   constructor(private http: HttpClient) { }
11
12   //get Transportes
13   getTransportes()
14   {
15     return this.http.get(this.url);
16   }
17
18   //get un Transporte
19   getunTransporte(id:string){
20     return this.http.get(this.url+'/'+id);
21   }
22 }
--
```

6.- Una vez importado podemos agregar una url de nuestro servicio en donde se encuentra corriendo exactamente para poder conectar la base de datos a nuestro frontend.



```
transport.service.ts
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class TransporteService {
8
9   url="http://localhost:4500/paqueria/lista-transporte";
10   constructor(private http: HttpClient) { }
11
12   //get Transportes
13   getTransportes()
14   {
15     return this.http.get(this.url);
16   }
17
18   //get un Transporte
19   getunTransporte(id:string){
20     return this.http.get(this.url+'/'+id);
21   }
22 }
--
```

7. Ahora deberemos de crear un método Get que nos permita mostrar los datos al hacer un llamado, para ello utilice el método getTransporte(), que nos va retornar el url que nosotros le asignemos.

```

getTransportes()
{
    return this.http.get(this.url);
}

```

Este método se va llamar en nuestro archivo inicio.ts y tendremos que inicializar el servicio y utilizar el método que estamos llamando para que realice la consulta y nos muestre los datos en el frontend.

```

//variable
ListarTransporte: Transporte[];
constructor(private TransporteService: TransporteService, private router:Router) { }

ngOnInit(): void {
    this.listarTransporte();
}

listarTransporte()
{
    this.TransporteService.getTransportes().subscribe(
        res=>{
            console.log(res)
            this.ListarTransporte=<any>res;
        },
        err=>console.log(err)
    )
}

```

7. Una vez realizado todo esto, ya podremos agregar estilo a nuestro HTML para que nos lo muestre de forma correcta nuestros datos de la base de datos, para ello modificamos nuestro inicio.html y quedaría así.

```

<div class="container">
    <p>Lista de Transportes</p>
    <button routerLink="/add" type="button" class="btn btn-dark">Agregar</button>
    <table class="table">
        <thead>
            <tr>
                <th>id</th>
                <th>Placas</th>
                <th>Tipo</th>
            </tr>
        </thead>
        <tbody>
            <tr *ngFor="let transporte of ListarTransporte">
                <td>{{transporte.idTransporte}}</td>
                <td>{{transporte.Placas}}</td>
                <td>{{transporte.Tipo}}</td>
                <td>
                    <button (click)="modificar(transporte.idTransporte)" type="button" class="btn btn-primary">Modificar</button>
                    <button (click)="eliminar(transporte.idTransporte)" type="button" class="btn btn-danger">Eliminar</button>
                </td>
            </tr>
        </tbody>
    </table>
</div>

```

Ahora veremos como queda en nuestro *Front-End*

