

Mitigación



Se hizo esta mitigación para tratar de vincular en front con el back sin problemas de CORS.

Angular:

1-

Instalar Node.js

<https://nodejs.org/es>

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.3324]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Cinti>node -v
v18.17.1

C:\Users\Cinti>
```

2-

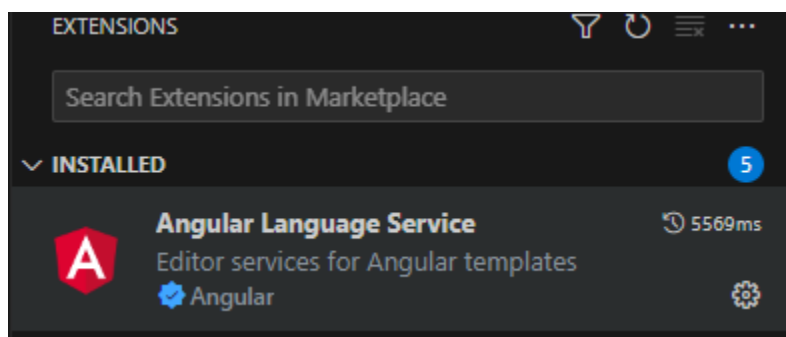
Instalar Visual Studio Code



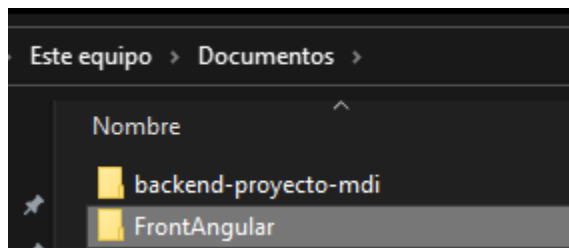
<https://code.visualstudio.com/download>

3-Opcionar

Dentro del Visual Studio Code instalar la extensión de Angular



4-Crear una carpeta donde sea, en mi caso le puse “FrontAngular”



5-Abrir la carpeta en el Visual Studio Code.

6-Abrir la consola



```
1-npm install -g @angular/cli
```

```
2-ng new pruebaNet
```

```
3-cd pruebaNet
```

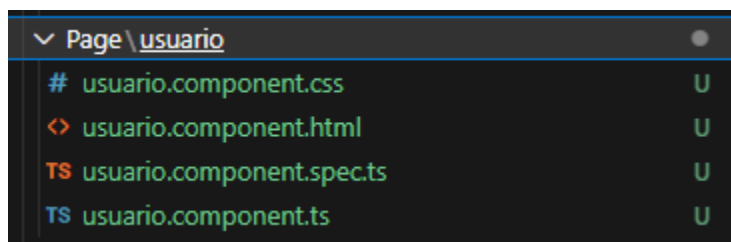
```
4-ng add @angular/material
```

```
5-ng serve
```

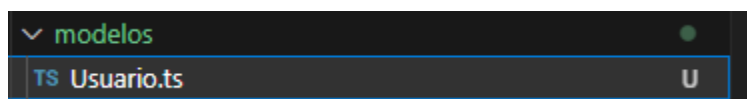
7- Dentro del proyecto a la altura de app crear 2 carpetas, una llamada ‘Modelos’, otra ‘Servicios’, y otra llamada ‘Page’.

8- Desde la consulta nos movemos usan cd hasta estar dentro de la carpeta Page y creamos un componente, la idea es que todos los componentes queden dentro de la carpeta para tener un orden.

```
ng generate component usuario
```



9-En la carpeta Model crear un archivo llamado usuario.ts, en ese archivo se declara un modelo para la class Usuario para usar en otros lados.

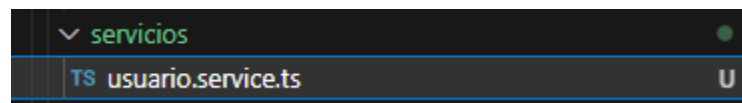


```

1  export class Usuario {
2      public id: number;
3      public name: string;
4      public correo: string;
5      public edad: number;
6
7      constructor(id: number, name: string, correo: string, edad: number) {
8          this.id = id;
9          this.name = name;
10         this.correo = correo;
11         this.edad = edad;
12     }
13 }

```

10- En la carpeta Servicios crear un archivo, aquí va la conexión.

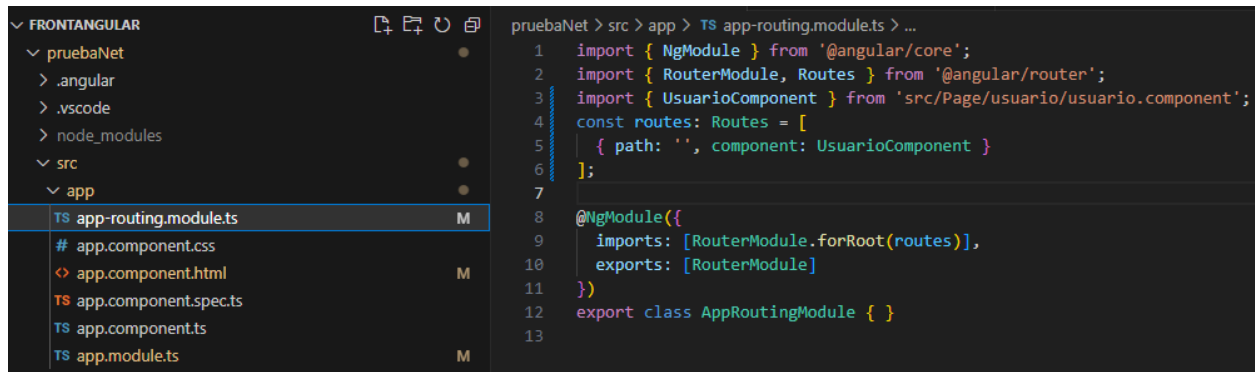


```

pruebaNet > src > servicios > TS usuario.service.ts > UsuarioService > constructor
1  import { Injectable } from '@angular/core';
2  import { catchError, Observable, throwError } from 'rxjs';
3  import { HttpClient, HttpResponse } from '@angular/common/http';
4  import { Usuario } from 'src/modelos/Usuario';
5  @Injectable({
6      providedIn: 'root'
7  })
8
9  export class UsuarioService {
10
11      url: string = "https://localhost:7166/usuario";
12      constructor(private http: HttpClient) { }
13
14      getUsuario() {
15          return this.http.get<Usuario[]>(this.url + '/listar');
16      }
17
18
19  }

```

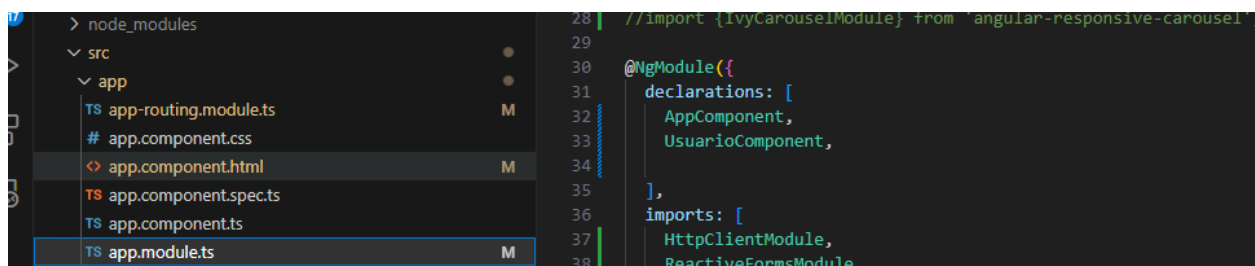
11- En el archivo app-routing.module.ts, es para administrar las rutas, en nuestro caso puse que cuando path es "" se dirija a mi componente Usuario.



The screenshot shows the VS Code interface. On the left, the file explorer displays the project structure: **FRONTANGULAR** > **pruebaNet** > **src** > **app**. The file **app-routing.module.ts** is selected. On the right, the code editor shows the following TypeScript code:

```
pruebaNet > src > app > TS app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { UsuarioComponent } from 'src/Page/usuario/usuario.component';
4  const routes: Routes = [
5    { path: '', component: UsuarioComponent }
6  ];
7
8  @NgModule({
9    imports: [RouterModule.forRoot(routes)],
10   exports: [RouterModule]
11 })
12 export class AppRoutingModule { }
```

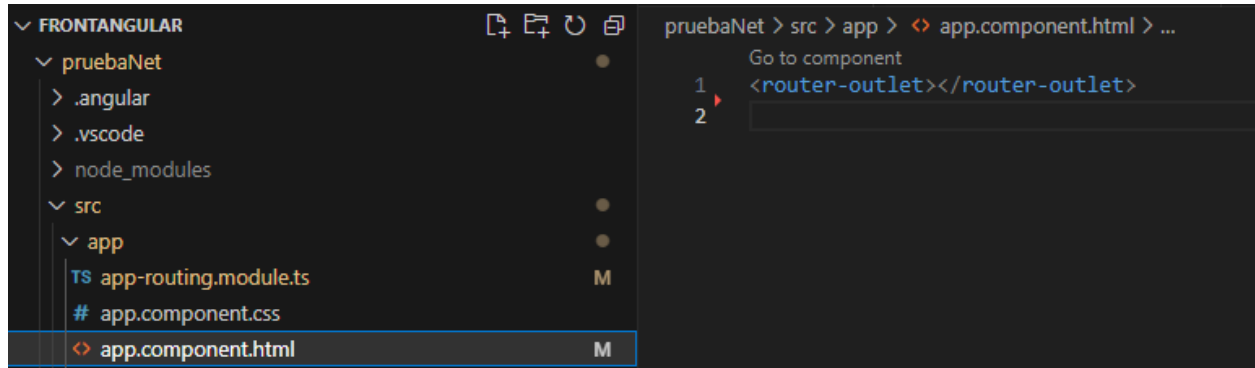
12- En el archivo app.module.ts es importante declarar nuestro nuevo componente.



The screenshot shows the VS Code interface. On the left, the file explorer shows the same project structure, but now **app.module.ts** is selected. On the right, the code editor shows the following TypeScript code:

```
28 //import { IvyCarouselModule } from 'angular-responsive-carousel';
29
30 @NgModule({
31   declarations: [
32     AppComponent,
33     UsuarioComponent,
34   ],
35   imports: [
36     HttpClientModule,
37     ReactiveFormsModule,
38
```

13- En el archivo app.component.html es importante la línea que se ve



The screenshot shows the VS Code interface. On the left, the file explorer shows the same project structure, but now **app.component.html** is selected. On the right, the code editor shows the following HTML code:

```
pruebaNet > src > app > <> app.component.html > ...
Go to component
1  <router-outlet></router-outlet>
2
```

14-En nuestro componente, colocamos un `ngOnInit` para que podamos ver lo que nos traemos del back al cargar la pagina.

```

1  import { Component, OnInit } from '@angular/core';
2  import { UsuarioService } from 'src/servicios/usuario.service';
3  import { Usuario } from 'src/modelos/Usuario';
4
5  @Component({
6    selector: 'app-usuario',
7    templateUrl: './usuario.component.html',
8    styleUrls: ['./usuario.component.css']
9  })
10 export class UsuarioComponent implements OnInit {
11
12   public usuario: Usuario[] = [];
13   constructor(private service: UsuarioService) { }
14
15   ngOnInit(): void {
16     this.service.getUsuario().subscribe((Data) => this.usuario = Data);
17   }
18 }
19

```

15-Para la parte que se ve, el archivo html usa componentes de angular Material, usando un ngFor podemos recorrer e imprimir cada datos del array de Json que manda el back.

```

1  <mat-card-header *ngFor="let x of usuario">
2    <mat-card class="example-card">
3      <mat-card-header>
4        <div mat-card-avatar class="example-header-image"></div>
5        <mat-card-title>{{ x.name }}</mat-card-title>
6      </mat-card-header>
7      
8      <mat-card-content>
9        <p>{{ x.correo }}</p>
10     </mat-card-content>
11     <mat-card-actions>
12       <button mat-button>LIKE</button>
13       <button mat-button>SHARE</button>
14     </mat-card-actions>
15   </mat-card>

```

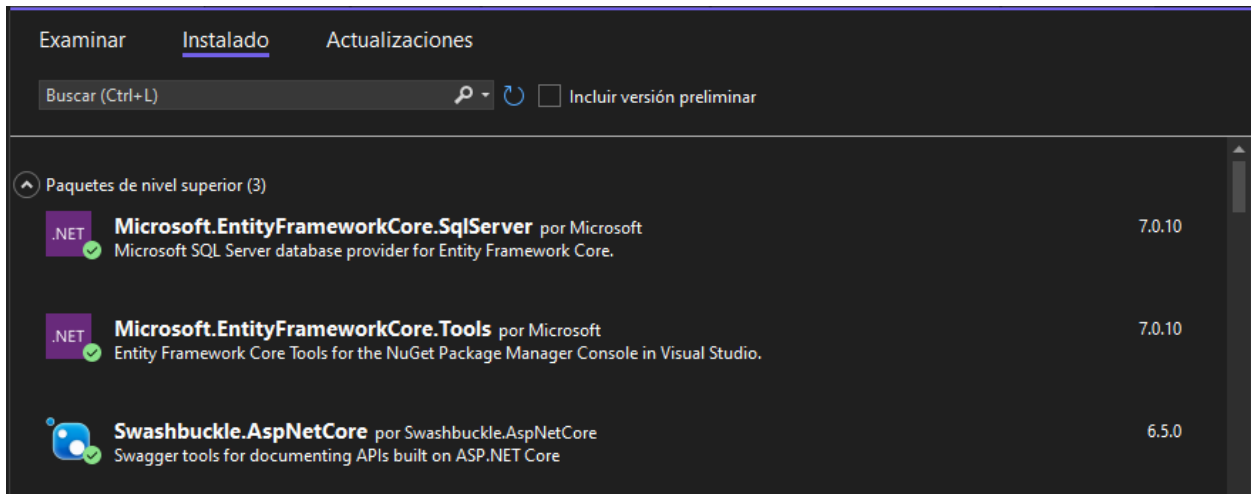
Angular Material: <https://material.angular.io/>

.NET

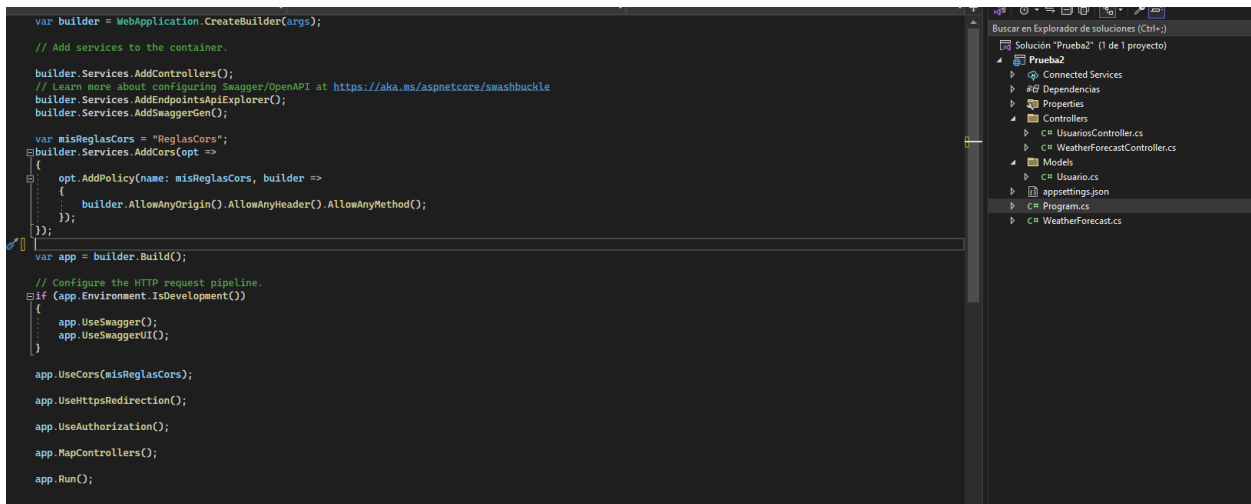
Como crear una Web Api

Importantes:

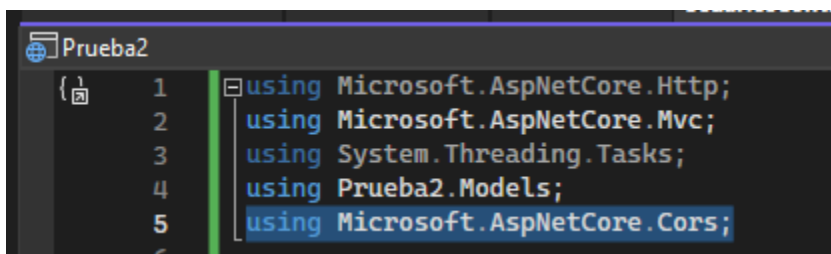
1-Instalar



2- En el archivo Program.cs ya la parte de CORS



3- En los controladores es importante incluir el using señalado y colocar la etiqueta de la línea 11.



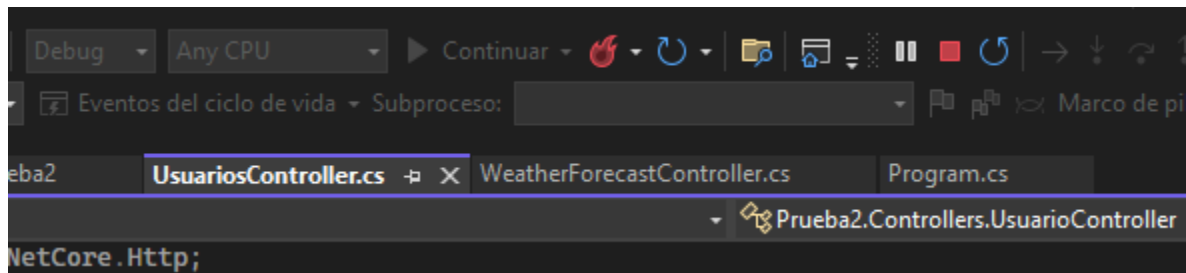
```

9
10
11 [EnableCors("ReglasCors")]
12 [Route("usuario")]
13 [ApiController]
14 0 referencias
15 public class UsuarioController : ControllerBase
16 {
17     [HttpGet]
18     [Route("listar")]
19     0 referencias
20     public dynamic listaUsuarios()
21     {

```

Ejecución:

1- Ejecutar desde el visual el proyecto .NET



2-Desde la consola del Visual Studio Code ejecuta para levantar el Front.


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Cinti\OneDrive\Documentos\FrontAngular> cd pruebaNet
PS C:\Users\Cinti\OneDrive\Documentos\FrontAngular\pruebaNet> ng serve
? Port 4200 is already in use.
Would you like to use a different port? Yes
✓ Browser application bundle generation complete.

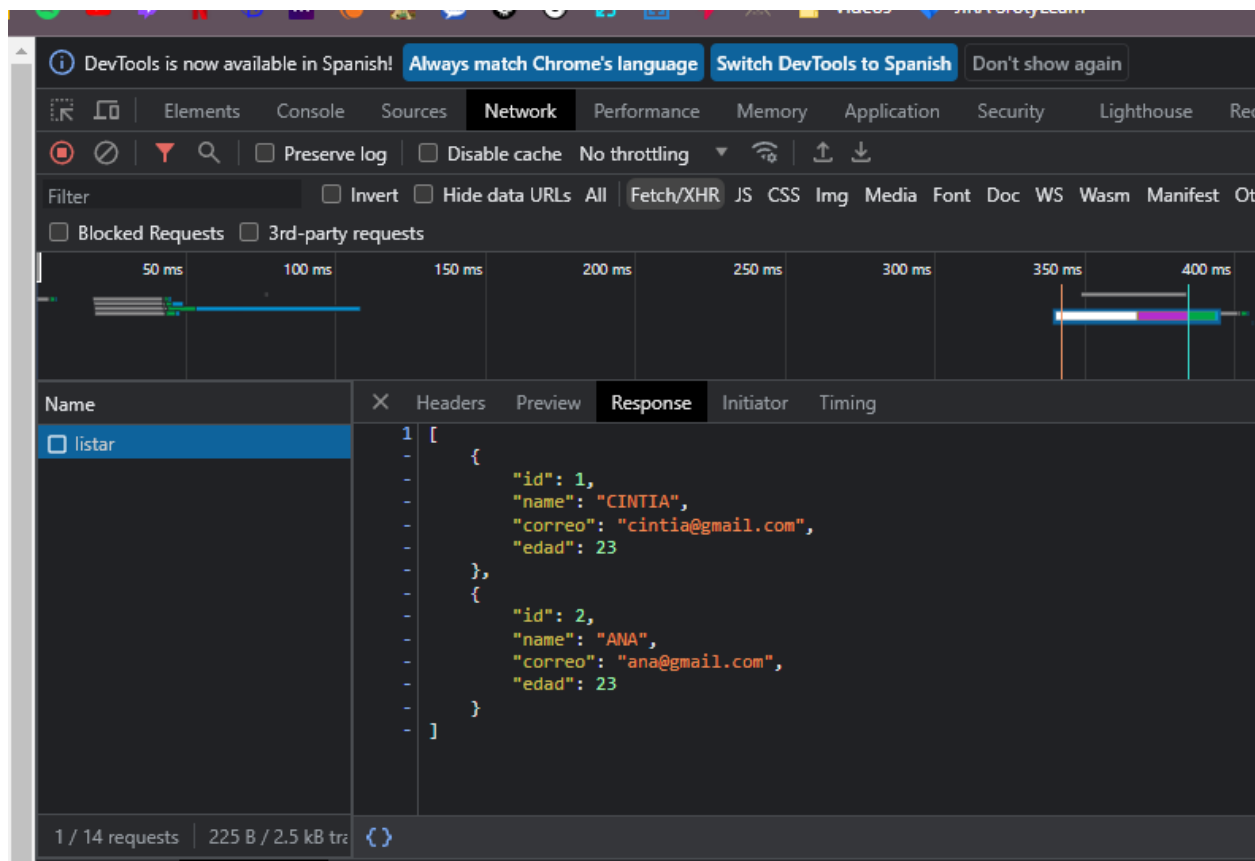
Initial Chunk Files | Names | Raw Size
vendor.js           | vendor | 5.15 MB |
polyfills.js        | polyfills | 333.18 kB |
styles.css, styles.js | styles | 315.83 kB |
main.js             | main | 20.81 kB |
runtime.js          | runtime | 6.52 kB |
| Initial Total | 5.81 MB

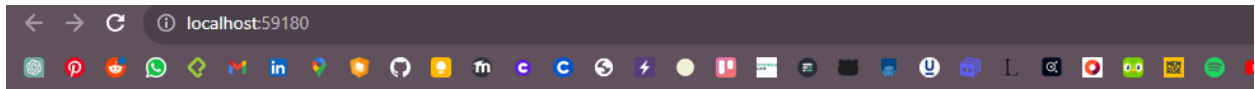
Build at: 2023-09-03T00:28:26.611Z - Hash: dd1db0cc80cc7ce5 - Time: 10278ms

** Angular Live Development Server is listening on localhost:59180, open your browser on http://localhost:59180/ **

✓ Compiled successfully.
```

3- Ya se puede observar desde angular lo que obtuvo del back





**CINTIA**



cintia@gmail.com

LIKE SHARE

**ANA**



ana@gmail.com