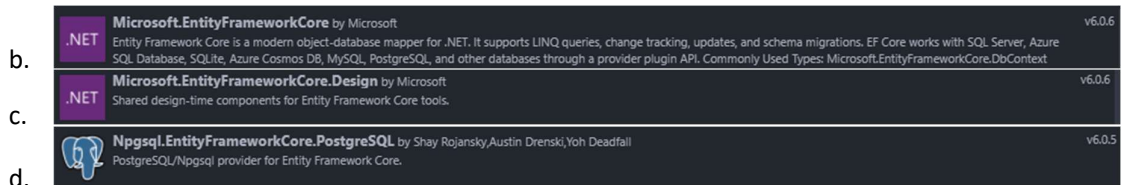
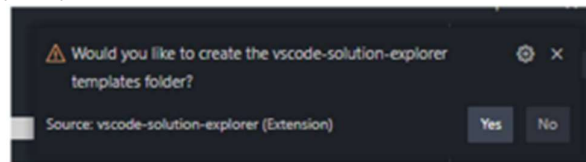


## API

1. Nueva terminal
  - a. dotnet new sln
2. En solution Explorer
  - a. Add new project
  - b. ASP .NET Core Web API
  - c. C#
  - d. Nombre Proyecto API
3. Instalar Extenciones con Nuget
  - a. Ctrl + shift +P



4. Cerrar VSCode, abrir de nuevo, clickar en barra navegación izquierda de solución (icono) y aceptar el cartel



- a.
5. Agregar Carpeta Comandos
6. Agregar Carpeta Resultados
7. Crear Carpeta Models
  - a. Abrir terminal
  - b. cd API
  - c. dotnet ef dbcontext scaffold "Server=localhost; Database=Prog3-11- 05; Port=5432; User Id=ClaseProg3; Password=123456;" Npgsql.EntityFrameworkCore.PostgreSQL --output-dir Models
8. Configurar Program.cs
  - a. Buscar en github
9. Configurar appSettings.json
  - a. Buscar en github
10. Crear Controlador para Usuarios
  - a. En carpeta Controllers crear UsuarioController.cs
  - b. Copiar de Github
11. Crear Resultados para Usuarios
  - a. Crear archivo ResultadoBase.cs
    - i. Copiar de Github
  - b. En carpeta Resultados crear Carpeta Usuarios
    - i. Crear archivo ResultadoUsuarioGet.cs
    - ii. Copiar de Github
12. Crear Comandos para Usuarios
  - a. En carpeta Comandos crear carpeta Usuarios
    - i. ComandoUsuarioPost.cs
    - ii. Copiar de Github

# ANGULAR

1. Crear nuevo Proyecto de Angular
  - a. En consola: `ng new nombreProyecto`
2. Levantar proyecto (dentro de carpeta del Proyecto):
  - a. En Consola: `cd Frontend`
  - b. `ng serve`
3. Colocar en `src/environments/environment.ts` las urls de API

```
export const environment = {  
  production: false,  
  urlAPI: 'http://localhost:3000/'  
};
```

- a.
4. En `src/index.html` importar Bootstrap y otras librerías

```
1 <!doctype html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="utf-8">  
5   <title>Frontend</title>  
6   <base href="/">  
7   <meta name="viewport" content="width=device-width, initial-scale=1">  
8   <link rel="icon" type="image/x-icon" href="favicon.ico">  
9   <!-- CSS only -->  
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0vHe/477x120vuzKqH+D8NvFBL000ltPr141jfhkMutt1p6p4vWor" crossorigin="anonymous">  
11  <!-- JavaScript Bundle with Popper -->  
12  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-pprn3073KE6169e4/SPyIo9T9XG737q7Xlglq2k2142k9162rx141k2" crossorigin="anonymous"></script>  
13 </head>  
14 <body>  
15   <app-root></app-root>  
16 </body>  
17 </html>
```

- a.
5. En `src/app/app.component.html` dejar lo necesario

```
1  
2 <span>{{ title }} app is running!</span>  
3 <router-outlet></router-outlet>
```

- a.
6. En `src/app/app.component.ts` programamos la funcionalidad del componente

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  title = 'FrontendAngular';  
}
```

- a.
7. Definir las redirecciones en `src/app/app.routing.module.ts`, de modo que según la ruta específico que componente me va a tener que abrir

```
const routes: Routes = [  
  {  
    path: 'login',  
    component: LoginComponent  
  },  
  {  
    path: 'listas',  
    component: ListasComponent  
  },  
  {  
    path: 'create',  
    component: CrearUsuarioComponent  
  },  
  {  
    path: 'editar/:id',  
    component: EditarUsuarioComponent  
  }  
];
```

- a.
- b. Buscar en Github

8. En src/app/app.module.ts declaramos todos los servicios que van a ser inyectados en nuestro código

```
@NgModule({
  declarations: [
    AppComponent,
    ParrafoEjemploComponent,
    ListasComponent,
    LoginComponent,
    EditarUsuarioComponent,
    CrearUsuarioComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule
  ],
  providers: [
    UsuarioProvider,
    RolesProvider
  ],
  bootstrap: [AppComponent]
})
```

- a.
  - b. Buscar en Github e Importar
9. Crear carpeta de Providers en src
    - a. Crear usuarioProvider.ts
    - b. Copiar código de Github
    - c. Importar Provider en appComponent
  10. Crear carpeta de Interfaces en src/app
    - a. Crear Usuario.ts como interfaz
    - b. Copiar código de Github
  11. En src/app/app.component.ts
    - a. Programación del componente principal que se inyecta en el Index.html
  12. En src/app/app.component.html
    - a. Definir barra de navegación con los respectivos enlaces de las rutas definidas en el App Routing
  13. Creación de Componentes
    - a. ng generate component nombreComponente
    - b. Copiar de Github