**1. Prerrequisitos**

Asegúrate de tener instalado lo siguiente:

* **Node.js y npm**: Angular necesita Node.js para su entorno de desarrollo. Descárgalo desde [Node.js](https://nodejs.org/).
* **Angular CLI (Command Line Interface)**: Instálalo globalmente usando npm.

npm install -g @angular/cli

### ****2. Crear un proyecto Angular****

1. **Generar un nuevo proyecto:**

Usa el comando ng new para crear un nuevo proyecto Angular.

ng new nombre-del-proyecto

**Opciones:**

* Te pedirá si deseas agregar routing. Responde Yes si necesitas navegación entre páginas.
* Selecciona el formato de estilos (CSS, SCSS, SASS, LESS).

**Accede al directorio del proyecto:**

cd nombre-del-proyecto

### ****3. Ejecutar el servidor de desarrollo****

Inicia el servidor para ver tu aplicación en acción.

ng serve

Luego, abre un navegador web y accede a http://localhost:4200. Aquí verás la aplicación base generada por Angular.

### ****4. Estructura del proyecto****

El proyecto Angular tiene una estructura organizada. Algunos de los directorios importantes son:

* **src/app/**: Contiene los componentes y la lógica de la aplicación.
* **src/assets/**: Para recursos estáticos como imágenes o archivos JSON.
* **src/environments/**: Configuración para diferentes entornos (desarrollo, producción).

### ****5. Crear un componente****

Los componentes son bloques fundamentales de Angular. Puedes crear uno con el CLI.

ng generate component nombre-componente

Esto crea un directorio con los archivos del componente:

* .component.ts: Lógica del componente.
* .component.html: Plantilla HTML.
* .component.css: Estilo del componente.
* .component.spec.ts: Pruebas unitarias.

Incluye automáticamente el componente en app.module.ts.

### ****6. Usar un componente****

1. En el archivo **app.component.html**, usa la etiqueta del componente:

<app-nombre-componente></app-nombre-componente>

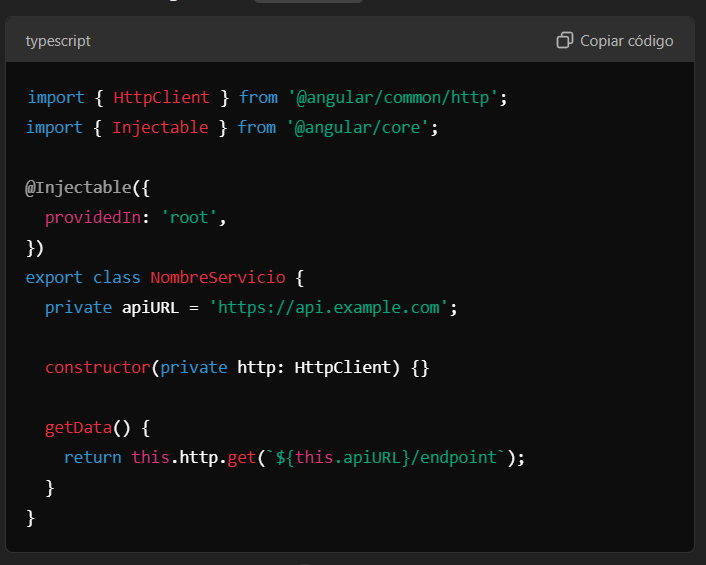
### ****7. Servicios y comunicación con APIs****

Para manejar datos o consumir APIs, utiliza **servicios**.

1. **Crear un servicio:**

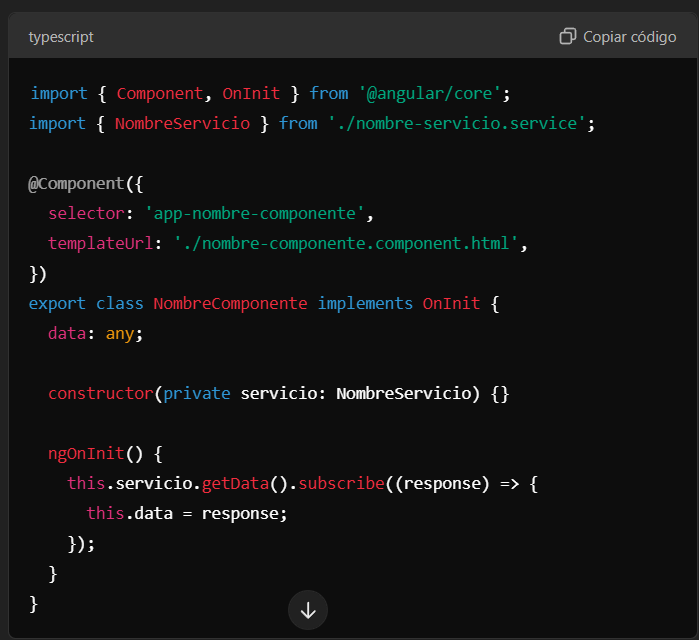
ng generate service nombre-servicio

**Lógica del servicio:** Edita el archivo .service.ts para manejar datos o llamadas HTTP. Angular usa HttpClient.



Usar el servicio en un componente:

Importa el servicio y usa su método.



### ****8. Crear rutas****

Si tu aplicación tiene varias páginas, utiliza el sistema de rutas de Angular.

1. Define rutas en app-routing.module.ts:

