



Intro Angular

Angular es un framework de desarrollo web de código abierto, mantenido por Google, que permite crear aplicaciones web de una sola página (SPA) de manera eficiente y escalable. En este conjunto de laboratorios, exploraremos los conceptos básicos de Angular, desde la instalación y configuración del entorno de desarrollo hasta la creación de componentes interactivos y el uso de directivas estructurales.



**by Esteban
Calabria**

Instalación y Configuración

Instalar Node.js

El primer paso es instalar Node.js, que proporciona el entorno de ejecución JavaScript necesario para Angular. Puedes descargarlo e instalarlo desde el sitio web oficial de Node.js.

Instalar Angular CLI

Una vez que tengas Node.js instalado, puedes proceder a instalar el Angular CLI, que es una herramienta de línea de comandos que facilita la creación y el desarrollo de aplicaciones

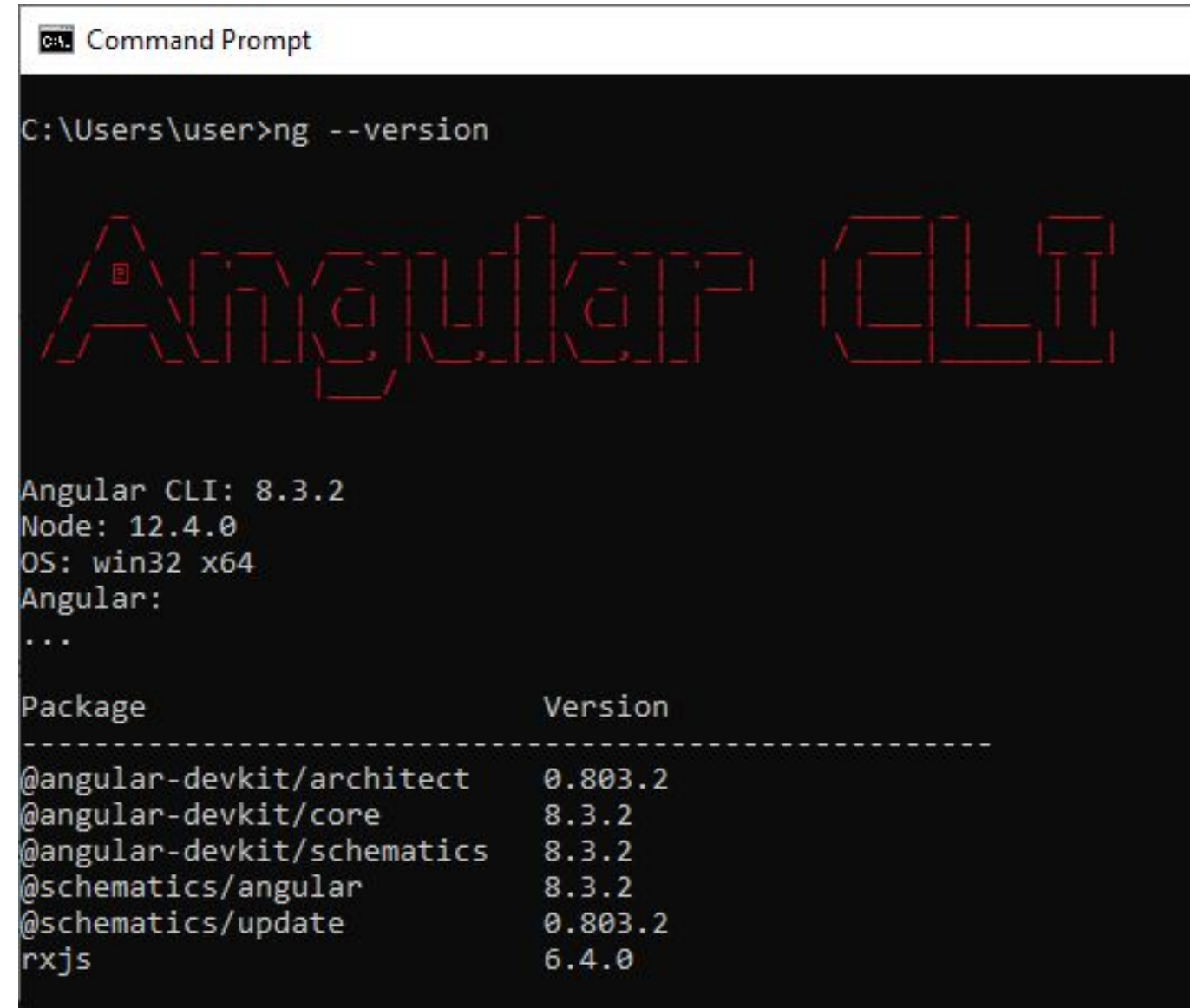
Crear una Nueva Aplicación

Usando el Angular CLI, puedes crear una nueva aplicación Angular con un solo comando. Esto generará la estructura básica de la aplicación y te permitirá comenzar a desarrollar de inmediato.



Instalar Angular

- Verificar si esta nodejs instalado
 - En una consola cmd ejecutar
 - `node -version`
- Instalar Node JS
 - Ir a la pagina <https://nodejs.org/en> y descargar el instalador
- Instalar Angular CLI
 - `npm install -g @angular/cli`
- Verificar la instalación
 - Abrir una consola cmd nueva y ejecutar el comando
 - `Ng -version`



```
Command Prompt
C:\Users\user>ng --version

Angular CLI
Angular CLI: 8.3.2
Node: 12.4.0
OS: win32 x64
Angular:
...

Package                                Version
-----
@angular-devkit/architect              0.803.2
@angular-devkit/core                   8.3.2
@angular-devkit/schematics             8.3.2
@schematics/angular                   8.3.2
@schematics/update                     0.803.2
rxjs                                   6.4.0
```

Crear Proyecto Angular

Ejecuta el siguiente comando para crear un nuevo proyecto Angular:

```
ng new nombre-proyecto
```

Navega al directorio del proyecto:

```
cd nombre-proyecto
```

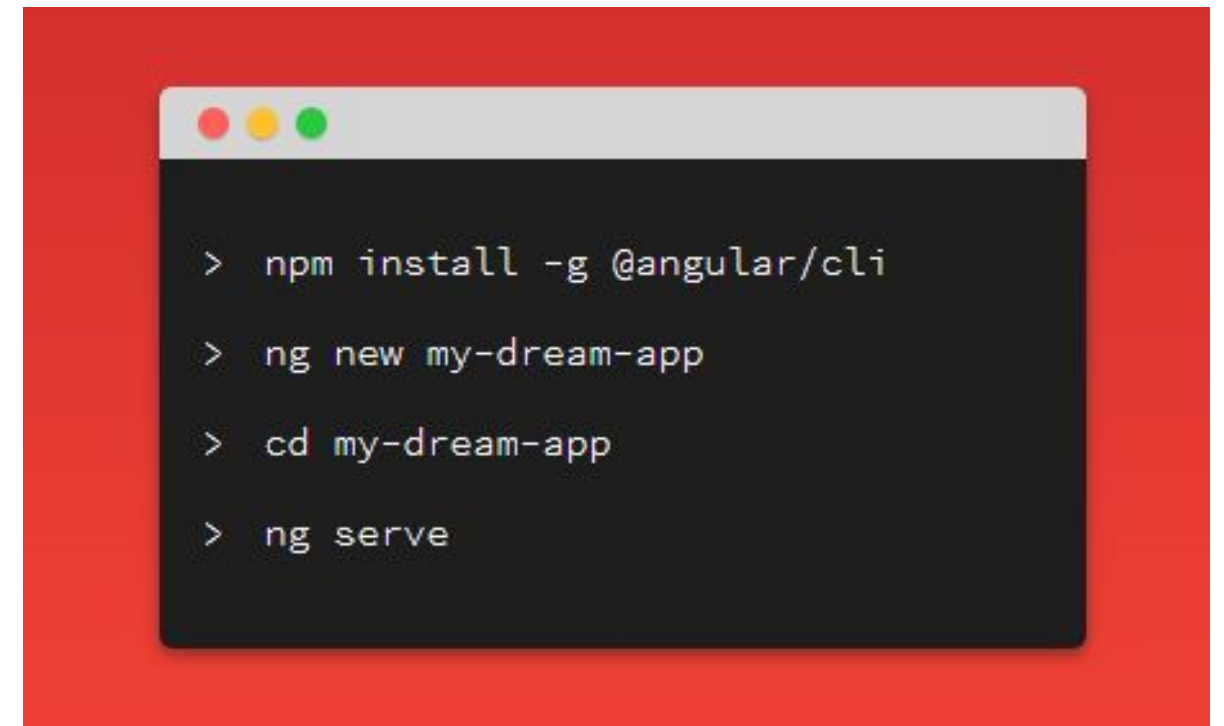
Ejecuta el servidor de desarrollo:

```
ng serve
```

Abre tu navegador web y visita

```
http://localhost:4200/
```

para ver tu nueva aplicación Angular en funcionamiento



Conceptos Básicos de Angular

1

Componentes

Los componentes son la unidad básica de una aplicación Angular. Cada componente tiene su propio template HTML, lógica de negocio en TypeScript y estilos CSS.

2

Directivas

Las directivas son clases que agregan comportamiento adicional a los elementos de tu aplicación Angular. Existen directivas de atributo y directivas estructurales.

3

Binding de Datos

Angular proporciona diferentes formas de enlazar datos entre el componente y la vista, como la interpolación de cadenas, el enlace de propiedades y el enlace de eventos.

re properties of class Point

draw a line comes here
of class by using 'this' operat
this.a + ', b: ' + this.b);

Point, point2: Point) => {
calculate length of line comes he

s of a and b

properties to set, like...

all method to draw a line

Trabajando con Componentes

Creación de Componentes

Puedes crear nuevos componentes utilizando el Angular CLI. Esto generará automáticamente los archivos necesarios, como el archivo TypeScript, el template HTML y los estilos CSS.

Comunicación entre Componentes

Los componentes pueden comunicarse entre sí a través de la entrada (input) y la salida (output). Esto permite que los componentes se reutilicen y se integren de manera modular.

Ciclo de Vida de los Componentes

Angular proporciona varios métodos de ciclo de vida que te permiten ejecutar código en diferentes etapas del ciclo de vida de un componente, como la inicialización, los cambios en las propiedades y la destrucción.

Directivas Estructurales

1

***ngIf**

La directiva *ngIf te permite mostrar u ocultar elementos condicionalmente en tu plantilla HTML.

2

***ngFor**

La directiva *ngFor te permite iterar sobre una lista de elementos y generar una plantilla para cada uno de ellos.

3

***ngSwitch**

La directiva *ngSwitch te permite mostrar diferentes elementos en función de una expresión.

NgIf

```
<div *ngIf="mostrarElemento">
```

Este elemento se mostrará si mostrarElemento es verdadero.

```
</div>
```


NgFor

```
<ul>
```

```
  <li *ngFor="let item of listaDeItems">
```

```
    {{ item }}
```

```
  </li>
```

```
</ul>
```


Enlace de Datos



Interpolación de Cadenas

La interpolación de cadenas te permite mostrar el valor de una propiedad del componente en la plantilla HTML.



Enlace de Propiedades

El enlace de propiedades te permite establecer el valor de un atributo HTML a partir de una propiedad del componente.



Enlace de Eventos

El enlace de eventos te permite escuchar y responder a eventos desencadenados por los elementos de la plantilla HTML.

Interpolación de Cadenas

{{double-moustache}}

```
<h1>{{ titulo }}</h1>
```

```
<p>Bienvenido(a), {{ nombre }}</p>
```

Enlace de Propiedades

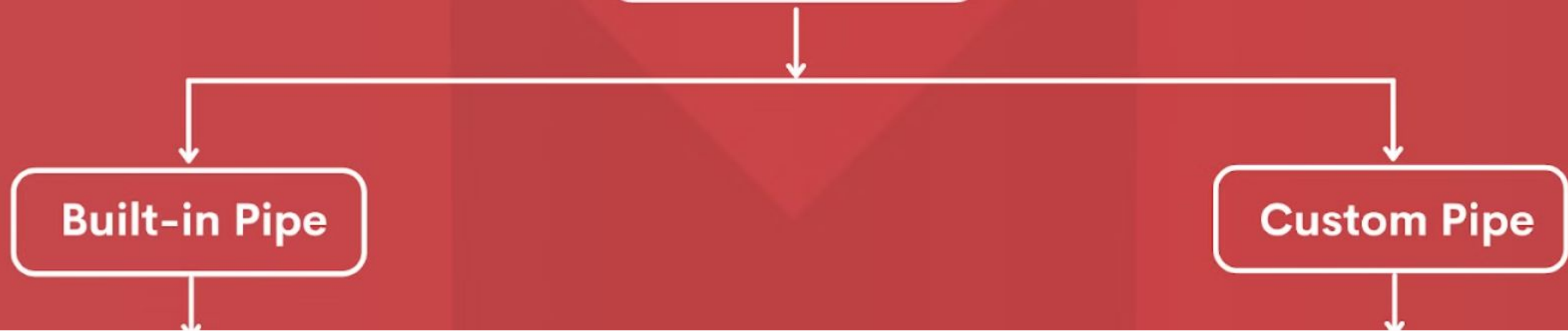
```
<button [disabled]="botonDeshabilitado">Enviar</button>
```

```
<img [src]="imagenURL" alt="Mi imagen">
```

Enlace de Eventos

```
<button [disabled]="botonDeshabilitado">Enviar</button>
```

```
<img [src]="imagenURL" alt="Mi imagen">
```



Pipes



Transformación de Datos

Los pipes te permiten transformar los datos que se muestran en la plantilla HTML, como formatear fechas, números y más.

Pipes Integrados

Angular proporciona una variedad de pipes integrados, como `UpperCasePipe`, `LowerCasePipe`, `DatePipe` y `CurrencyPipe`, entre otros.

Pipes Personalizados

También puedes crear tus propios pipes personalizados para adaptarlos a tus necesidades específicas.

Pipes

<p>Fecha actual: {{ fechaActual | date }}</p>

<p>Fecha con formato personalizado: {{ fechaActual | date:'dd/MM/yyyy' }}</p>

<p>Número: {{ numero }}</p>

<p>Número con decimales: {{ numero | number:'1.2-3' }}</p>

<p>Cadena en mayúsculas: {{ cadena | uppercase }}</p>

<p>Cadena en minúsculas: {{ cadena | lowercase }}</p>

<p>Cadena truncada: {{ cadenaLarga | slice:0:10 }}</p>

<p>Moneda: {{ cantidad | currency:'USD' }}</p>

<p>JSON: {{ objeto | json }}</p>

Servicios

Inyección de Dependencias

Angular utiliza la inyección de dependencias para proporcionar servicios a los componentes.

Comunicación entre Componentes

Los servicios pueden ser utilizados para compartir datos y lógica entre diferentes componentes, facilitando la comunicación entre ellos.

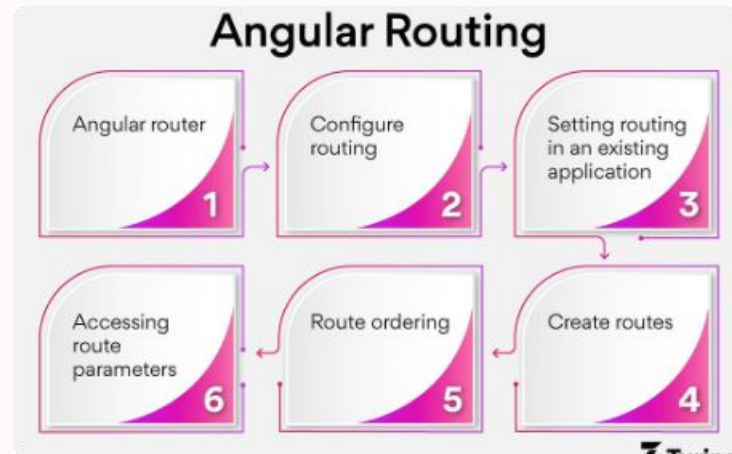
Acceso a Datos

Los servicios son ideales para encapsular la lógica de acceso a datos, ya sea de una API, una base de datos o cualquier otra fuente de datos.

Pruebas Unitarias

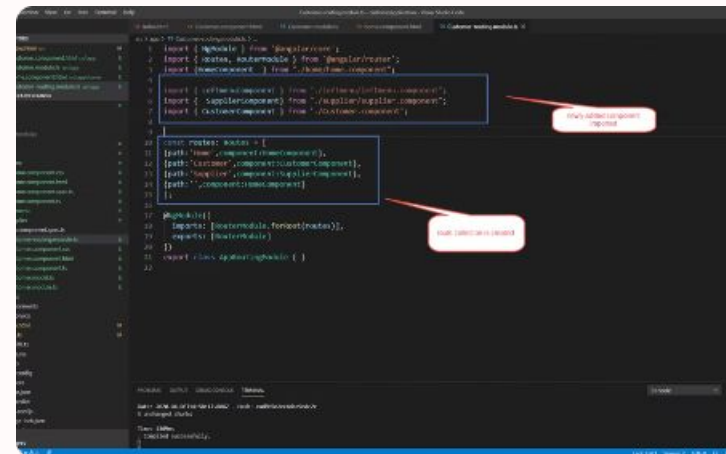
Al separar la lógica en servicios, es más fácil realizar pruebas unitarias y mantener la calidad de tu aplicación.

Enrutamiento



Configuración de Rutas

El enrutamiento en Angular te permite definir las diferentes rutas de tu aplicación y asociarlas a componentes específicos.



Navegación entre Rutas

Angular proporciona herramientas para navegar entre las diferentes rutas de tu aplicación, como el uso de enlaces y el enrutamiento programático.



Parámetros de Ruta

Las rutas pueden incluir parámetros que se pueden utilizar para pasar información entre componentes y mejorar la experiencia del usuario.