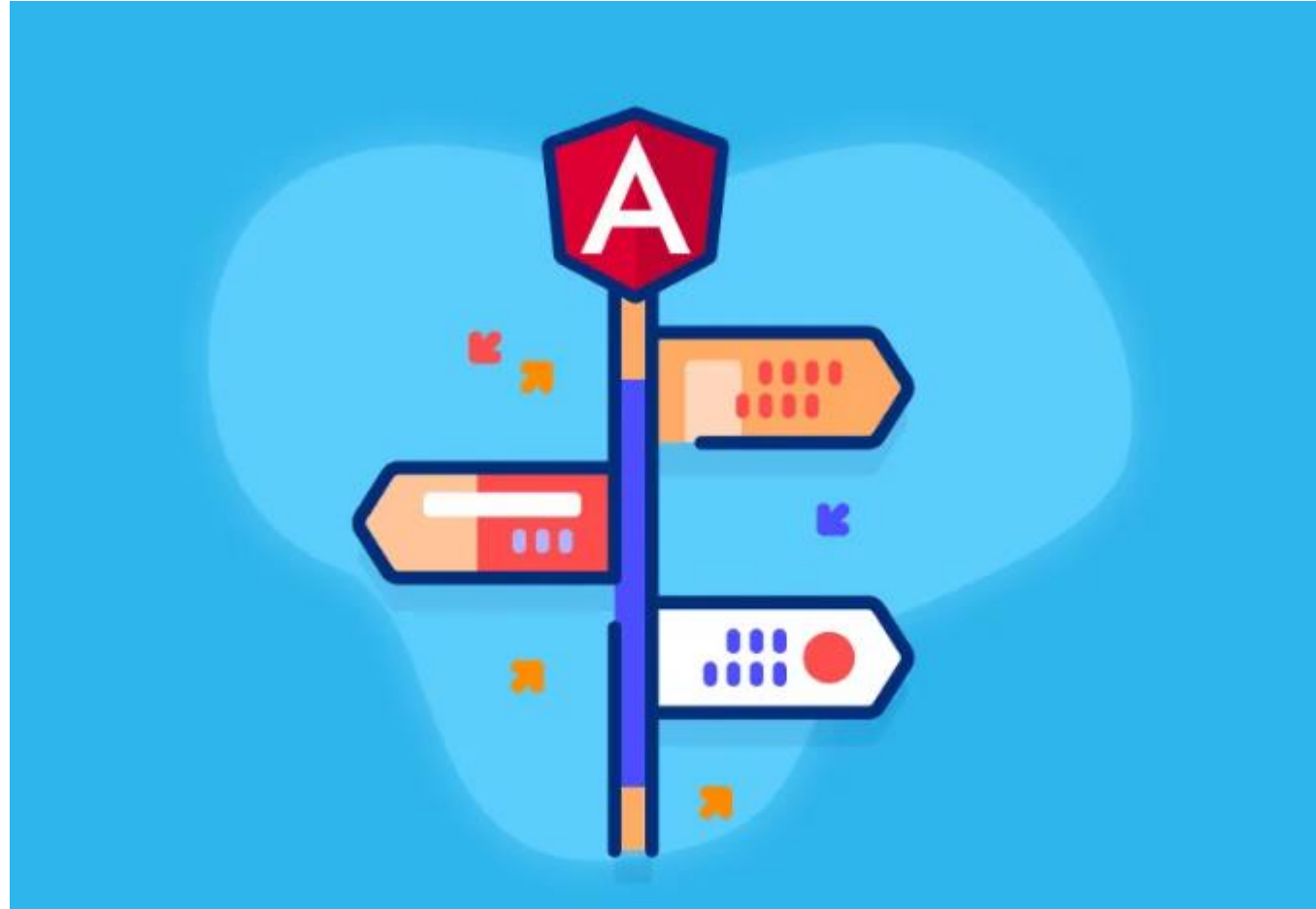


# Enrutamiento de Angular



# ¿Qué son las rutas en Angular?

¿Las **rutas en Angular** son una forma de **navegar entre diferentes vistas de los componentes** de una aplicación **SPA** (Single Page Application), **sin recargar la página**.





Las rutas en Angular son como un sistema de **GPS** para tu proyecto.  
Tú defines qué **camino** lleva a qué **pantalla**, y Angular se encarga de mostrarla  
sin recargar la página completa

# ¿Cómo funcionan las rutas?



1. Tú defines un arreglo de rutas en tu app (como un mapa).
2. Angular analiza la URL del navegador.
3. Si coincide con alguna ruta definida, muestra el componente asignado en el lugar que pongas la etiqueta **<router-outlet>**.

# Ejemplo

Supón que tienes dos componentes: **InicioComponent** y **ContactoComponent**.

Entonces en tu archivo de rutas **app.routes.ts** completas tu arreglo

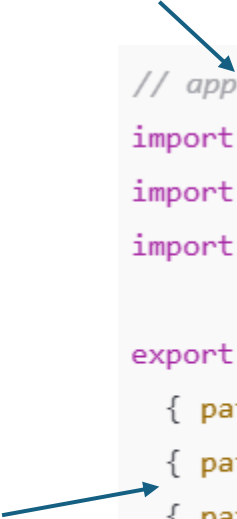
```
const routes: Routes = [  
  { path: 'inicio', component: InicioComponent },  
  { path: 'contacto', component: ContactoComponent },  
  { path: '', redirectTo: 'inicio', pathMatch: 'full' },  
  { path: '**', redirectTo: 'inicio' }  
];
```

**Esto significa:**

- **/inicio** → muestra vista del componente InicioComponent
- **/contacto** → muestra vista de componente ContactoComponent
- **/** redirige a /inicio
- Cualquier ruta inválida (**/algoqueNoExiste**) también redirige a /inicio

# ¿Que necesitas para implementar rutas en Angular?

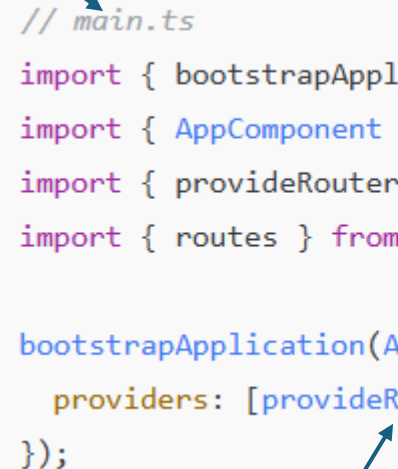
## 1 Definir rutas



```
// app.routes.ts
import { Routes } from '@angular/router';
import { HomeComponent } from './home.component';
import { ContactoComponent } from './contacto.component';

export const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'contacto', component: ContactoComponent },
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: '**', redirectTo: 'home' }
];
```

## 2. Configurar el main.ts con provideRouter



```
// main.ts
import { bootstrapApplication } from '@angular/platform-browser';
import { AppComponent } from './app/app.component';
import { provideRouter } from '@angular/router';
import { routes } from './app.routes';

bootstrapApplication(AppComponent, {
  providers: [provideRouter(routes)]
});
```

3. En el archivo ts del componente que controla el menú, importar el uso de RouterLink y RouterOutlet. Por ejemplo en el archivo ts del componente app.

```
ts

import { Component } from '@angular/core';
import { RouterOutlet, RouterLink } from '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet, RouterLink],
  templateUrl: './app.component.html'
})
export class AppComponent { }
```

4. En el archivo html del componente que controla el menú, colocar el atributo routerLink y la etiqueta router-outlet. Por ejemplo en el archivo html del componente app.

```
html

<h1>Mi App</h1>

<nav>
  <a routerLink="/home">Inicio</a>
  <a routerLink="/contacto">Contacto</a>
</nav>

<router-outlet></router-outlet>
```

## Sobre el uso del routerLink en el html del componente

```
html

<h1>Mi App</h1>

<nav>
  <a routerLink="/home">Inicio</a>
  <a routerLink="/contacto">Contacto</a>
</nav>

<router-outlet></router-outlet>
```



### Diferencia entre routerLink="/ruta" y routerLink="ruta"

#### Sintaxis

routerLink="/home"

#### Significado

Ruta **absoluta**

routerLink="home"

Ruta **relativa** al  
path actual

#### Resultado real

Ir directamente a /home desde cualquier parte de la app. cuando quieres ir siempre a una ruta específica de nivel raíz.

Se concatena al path actual. Ej: Si estás en /admin, va a /admin/home cuando estás navegando **dentro de rutas hijas** o componentes que están anidados.



Las rutas en Angular van mucho más allá de solo hacer clic en un menú.  
Son una **pieza clave en la arquitectura de una SPA moderna**



¿En que más puedo usar  
las rutas de Angular?

## Uso de las rutas en Angular

#	Uso	¿Para qué sirve?	Ejemplo de uso	Situación práctica real
1	<b>Navegación programática</b>	Cambiar de vista desde código TypeScript	<code>this.router.navigate(['/home'])</code>	Redirigir al usuario después de iniciar sesión
2	<b>Parámetros en la URL</b>	Mostrar info dinámica usando la URL	<code>/producto/42,</code> <code>route.snapshot.params['id']</code>	Ver detalles de un producto o editar un registro
3	<b>Guards (protección)</b>	Controlar el acceso a rutas	<code>canActivate: [AuthGuard]</code>	Bloquear acceso a usuarios no logueados o no autorizados
4	<b>Lazy loading</b>	Cargar módulos solo cuando se necesitan	<code>loadChildren: () =&gt;</code> <code>import('./admin.routes')</code>	Mejorar rendimiento inicial de la app
5	<b>Rutas hijas</b>	Crear secciones internas dentro de una vista principal	<code>children: [{ path: 'editar',</code> <code>component:</code> <code>EditarComponent }]</code>	Submenús, paneles en dashboard
6	<b>Múltiples outlets</b>	Mostrar varios componentes al mismo tiempo	<code>&lt;router-outlet</code> <code>name="popup"&gt;&lt;/router-</code> <code>outlet&gt;</code>	Mostrar un modal o panel lateral sin salir de la página