



# DESARROLLO DE SOFTWARE FRONTEND III

MARTINEZ COSIO JOSE ALFREDO

Aarón Hernández García

Práctica con enrutamiento en  
Angular (Proyecto sencillo)

05/08/2025

# Proyecto Angular Propuesto: “Mi Sitio Simple”

## 1. Crear el proyecto y configurar el enrutamiento

```
ng new mi-sitio-simple --routing  
cd mi-sitio-simple
```

## 2. Crear componentes

```
ng generate component home  
ng generate component perfil  
ng generate component login  
ng generate component not-found
```

## 3. Crear un guardia de ruta (AuthGuard)

```
ng generate guard auth
```

## 4. Configurar las rutas en app-routing.module.ts

```
import { NgModule } from '@angular/core';  
import { RouterModule, Routes } from '@angular/router';  
import { HomeComponent } from './home/home.component';  
import { PerfilComponent } from './perfil/perfil.component';  
import { LoginComponent } from './login/login.component';  
import { NotFoundComponent } from './not-found/not-found.component';  
import { AuthGuard } from './auth.guard';  
  
const routes: Routes = [  
  { path: '', component: HomeComponent },  
  { path: 'login', component: LoginComponent },  
  { path: 'perfil/:usuario', component: PerfilComponent, canActivate: [AuthGuard] },  
  { path: '**', component: NotFoundComponent }  
];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule { }
```

## 5. Implementar navegación entre rutas (routerLink)

En app.component.html:

```
<nav>
  <a routerLink="/">Inicio</a>
  <a routerLink="/login">Login</a>
  <a [routerLink]="['/perfil', 'aaron']">Perfil</a>
</nav>

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

## 6. Usar parámetros en la ruta perfil/:usuario

En perfil.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-perfil',
  templateUrl: './perfil.component.html'
})
export class PerfilComponent implements OnInit {
  usuario: string = '';

  constructor(private route: ActivatedRoute) {}

  ngOnInit(): void {
    this.usuario = this.route.snapshot.paramMap.get('usuario') || '';
  }
}
```

## 7. Crear el guardia auth.guard.ts

```
import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(private router: Router) {}

  canActivate(): boolean {
    const autenticado = false; // simula si el usuario está logueado
    if (!autenticado) {
      this.router.navigate(['/login']);
      return false;
    }
    return true;
  }
}
```

## 8. Conclusión del documento

El enrutamiento en Angular permite organizar la navegación de la aplicación de forma modular y escalable. El uso de rutas con parámetros facilita el paso de datos entre páginas, mientras que los guardias (guards) agregan control sobre el acceso a ciertas vistas. Estas herramientas permiten crear experiencias más seguras, dinámicas y estructuradas.

