

Caso Agregar con Angular – parte 1



Primera parte del caso de agregar producto en Angular

En esta guía, agregaremos la funcionalidad de navegación para la vista de **Agregar Producto** en Angular. Se explicará cómo crear el componente, configurar la ruta y modificar la navegación en la barra de menú.

Paso 1: Crear el componente `AgregarProductoComponent`

Ejecuta el siguiente comando en la terminal dentro del proyecto Angular:

```
ng g c agregar Producto --skip-tests
```

Explicación del comando:

- `ng g c agregar Producto`:
 - **Genera un componente nuevo** llamado `AgregarProductoComponent`.
 - `g` es de generate y `c` de component
- `--skip-tests`:

- Evita la creación del archivo de pruebas unitarias, ya que en este paso no lo necesitamos.

Después de ejecutar el comando, Angular generará los siguientes archivos:

```
/src/app/agregar Producto/
└── agregar Producto.component.ts
└── agregar Producto.component.html
└── agregar Producto.component.css (si se usa CSS global)
```

Paso 2: Agregar la ruta en `app.routes.ts`

Código actualizado de `app.routes.ts`

```
import { Routes } from '@angular/router';
import { ProductoListaComponent } from './producto-lista/producto-lista.component';
import { AgregarProductoComponent } from './agregar producto/agregar producto.component';

export const routes: Routes = [
  { path: 'productos', component: ProductoListaComponent },
  { path: '', redirectTo: 'productos', pathMatch: 'full' },
  { path: 'agregar producto', component: AgregarProductoComponent }
];
```

Explicación de los cambios:

1. Importamos `AgregarProductoComponent`:
 - Esto es necesario para que Angular reconozca el nuevo componente dentro del sistema de rutas.
 2. Se agregó la nueva ruta `{ path: 'agregar producto', component: AgregarProductoComponent }`:
 - Ahora, cuando el usuario navegue a `http://localhost:4200/agregar producto`, Angular renderizará `AgregarProductoComponent`.
 - Se mantiene la estructura de enrutamiento utilizando `Routes`.
-

Paso 3: Modificar la navegación en `app.component.html`

Código actualizado de `app.component.html`

```
<div class="container">
  <nav class="navbar navbar-expand-lg navbar-dark"
  style="background-color: #8c0aeef;">
```

```

<div class="container-fluid">
  <a class="navbar-brand"
    routerLink="productos">Sistema de Inventarios</a>
  <button class="navbar-toggler" type="button"
    data-bs-toggle="collapse" data-bs-target="#navbarNav"
    aria-controls="navbarNav" aria-expanded="false"
    aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page"
          routerLink="productos"
          routerLinkActive="active">Inicio</a>
      </li>
      <li class="nav-item">
        <a class="nav-link"
          routerLink="agregar-producto"
          routerLinkActive="active">Agregar Producto</a>
      </li>
    </ul>
  </div>
</div>
</nav>
</div>

<div class="container">
  <router-outlet></router-outlet>
</div>

```

Explicación de los cambios:

1. Cambio de `href` por `routerLink` en el enlace "Agregar Producto":

```
<a class="nav-link" routerLink="agregar-producto"
  routerLinkActive="active">Agregar Producto</a>
```

- ¿Por qué usar `routerLink` en lugar de `href`?

- `href` provoca una recarga completa de la página, lo que no es ideal en una aplicación de una sola página (SPA).
- `routerLink` permite la navegación interna sin recargar la página, lo que hace que la aplicación sea más rápida y fluida.
- `routerLinkActive="active"` resalta el enlace cuando la ruta está activa.

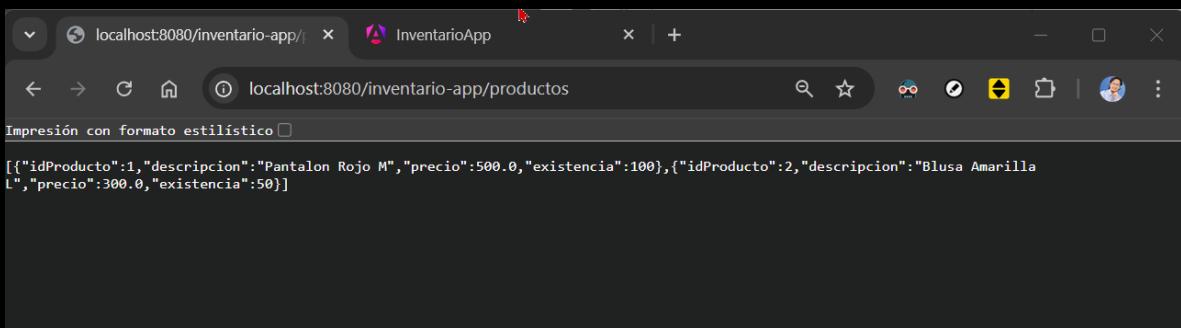
2. Uso de `<router-outlet>`:

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

- Este marcador de posición, como ya sabemos, indica **dónde se deben renderizar los componentes** según la ruta activa.
 - Cuando el usuario navegue a `agregar-producto`, Angular insertará dinámicamente el `AgregarProductoComponent` en este lugar.
-

Paso 4: Probar la navegación

1. Ya debe estar ejecutándose la aplicación de Spring Boot (BackEnd) en la ruta:
<http://localhost:8080/inventario-app/productos>



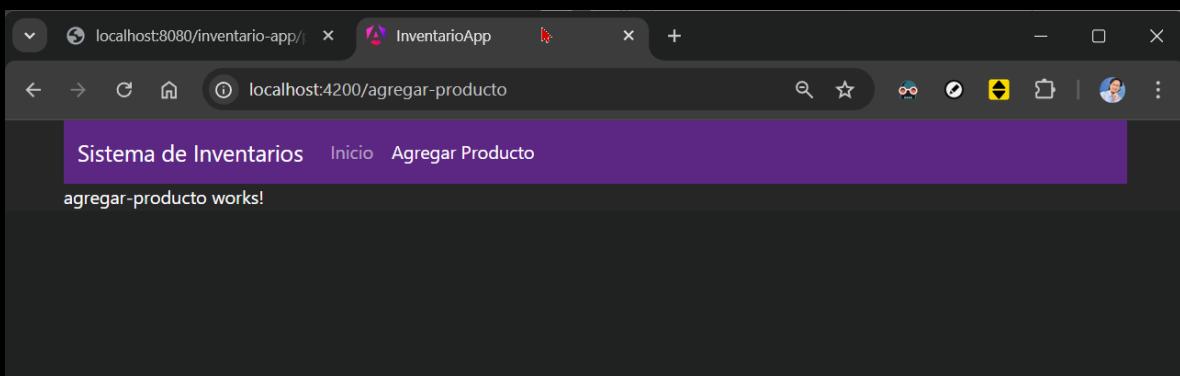
2. Ejecutar la aplicación Angular:

```
ng serve -o
```

3. Abrir el navegador en <http://localhost:4200/>

4. Hacer clic en el enlace "Agregar Producto":

- Se debería navegar a <http://localhost:4200/agregar-producto>.
- La página debería mostrar el contenido de `AgregarProductoComponent`.
- Esta es la primera parte, en el siguiente video veremos cómo agregar el contenido de este componente para poder agregar un nuevo producto.



Conclusión

- Se creó el componente `AgregarProductoComponent` con su gen.
- Se configuró la ruta en `app.routes.ts` para manejar `/agregar-producto`.
- Se modificó `app.component.html` para usar `routerLink` en lugar de `href`.
- Se usó `<router-outlet>` para cargar dinámicamente los componentes según la ruta.

Con estos cambios, la aplicación ahora permite navegar a la vista de "Agregar Producto" sin recargar la página, mejorando la experiencia de usuario. 

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](#)