

# Formulario para Editar con Angular



## Tercera parte de la edición de un producto en Angular

En esta tercera parte, agregaremos **un formulario de edición usando Bootstrap** y configuraremos **Two-Way Binding** con `ngModel` en **Angular**. Además, prepararemos el método `onSubmit()`, que implementaremos más adelante.

---

### Paso 1: Crear el formulario de edición en `editar-producto.component.html`

#### Código actualizado

```
<div class="container">  
  <div class="container text-center" style="margin: 30px">  
    <h3>Editar Producto</h3>  
  </div>  
  
<form (ngSubmit)="onSubmit()">  
  <div class="mb-3">
```

```

        <label for="descripcion"
      class="form-label">Descripción del Producto</label>
      <input type="text" class="form-control"
      id="descripcion" name="descripcion" required="true"
      [(ngModel)]="producto.descripcion">
    </div>
    <div class="mb-3">
      <label for="precio" class="form-label">Precio</label>
      <input type="number" step="any" class="form-control"
      id="precio" name="precio" [(ngModel)]="producto.precio">
    </div>
    <div class="mb-3">
      <label for="existencia" class="form-label">Existencia</label>
      <input type="number" class="form-control"
      id="existencia" name="existencia"
      [(ngModel)]="producto.existencia">
    </div>
    <div class="text-center">
      <button type="submit"
      class="btn btn-warning btn-sm me-3">Editar</button>
      <a href="/" class="btn btn-danger btn-sm">Regresar</a>
    </div>
  </form>

</div>

```

## Explicación del código

### 1. Uso de `ngSubmit="onSubmit()"` en el formulario:

```
<form (ngSubmit)="onSubmit()">
```

- Evita la recarga de la página y ejecuta la función `onSubmit()` cuando se presiona el botón "Editar".

### 2. Uso de `ngModel` en los campos del formulario:

- `[(ngModel)]="producto.descripcion"`
- `[(ngModel)]="producto.precio"`
- `[(ngModel)]="producto.existencia"`
- Esto permite que los datos del producto se muestren en el formulario y puedan ser modificados en tiempo real.

### 3. Botones estilizados con Bootstrap:

- Botón "Editar":

```
<button type="submit" class="btn btn-warning btn-sm me-3">Editar</button>
```

- `btn-warning`: Color amarillo para indicar una acción de edición.

- btn-sm: Tamaño más pequeño.
- m-3: Margen derecho para separación.
- Botón "Regresar":

```
<a href="/" class="btn btn-danger btn-sm">Regresar</a>
```

- btn-danger: Color rojo para indicar una acción de salida o cancelación.

---

## Paso 2: Modificar `editar-producto.component.ts` para importar `FormsModule`

### Código actualizado

```
import { Component, inject } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { Producto } from '../producto';
import { ProductoService } from '../producto.service';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-editar-producto',
  imports: [FormsModule],
  templateUrl: './editar-producto.component.html'
})
export class EditarProductoComponent {
  producto: Producto = new Producto();
  id: number;

  private productoServicio = inject(ProductoService);
  private ruta = inject(ActivatedRoute);

  ngOnInit() {
    this.id = this.ruta.snapshot.params['id'];
    this.productoServicio.obtenerProductoPorId(this.id).subscribe(
      {
        next: (datos) => this.producto = datos
      ,
        error: (errores: any) => console.log(errores)
      }
    );
  }
}
```

```

onSubmit(){
    //editar producto
}
}

```

## Explicación de los cambios

### 1. Se importó `FormsModule` en imports: `[FormsModule]`:

```
imports: [FormsModule]
```

- **FormsModule es necesario para que ngModel funcione** y permita el Two-Way Binding.
- Sin esto, Angular no reconocerá `[ngModel]` en el formulario y mostrará errores.

### 2. Método `ngOnInit()`:

```

ngOnInit() {
    this.id = this.ruta.snapshot.params['id'];
    this.productoServicio.obtenerProductoPorId(this.id).subscribe(
        {
            next: (datos) => this.producto = datos,
            error: (errores: any) => console.log(errores)
        }
    );
}

```

- **Obtiene el ID del producto desde la URL** (`/editar-producto/:id`).
- **Llama al servicio obtenerProductoPorId(id)** para recuperar el producto a editar.
- **Muestra los datos en el formulario automáticamente.**

### 3. Método `onSubmit()` (Aún no implementado):

```

onSubmit() {
    // editar producto (se implementará en el siguiente paso)
}

```

- Este método se ejecutará cuando el usuario presione el botón "Editar".
- En el siguiente paso, implementaremos la lógica para actualizar el producto en el backend.

## Paso 3: Probar la funcionalidad en el navegador

### Pasos para probar

#### 1. Ejecutar la aplicación Angular:

<https://www.globalmentoring.com.mx>

```
ng serve -o
```

2. Abrir el navegador en `http://localhost:4200/productos`.
3. Hacer clic en el botón "Editar" de un producto:

The screenshot shows a web browser window with the URL `localhost:4200/productos`. The page title is "InventarioApp". The main content is a table titled "Sistema de Inventarios" with the following data:

ID	Descripción del Producto	Precio	Existencias	Acciones
1	Pantalon Rojo M	500	100	<button>Editar</button>
2	Blusa Amarilla L	300	50	<button>Editar</button>
3	Vaso Tequilero	200	30	<button>Editar</button>
4	Florero L	350	23	<button>Editar</button>

- La URL cambiará a algo como:

<http://localhost:4200/editar-producto/1>

The screenshot shows a web browser window with the URL `localhost:4200/editar-producto/1`. The page title is "InventarioApp". The main content is a form titled "Editar Producto" with the following fields:

- Descripción del Producto: Pantalon Rojo M
- Precio: 500
- Existencia: 100

At the bottom of the form are two buttons: "Editar" (in yellow) and "Regresar" (in red).

- Los **campos del formulario** se llenarán automáticamente con la información del producto seleccionado.

4. Modificar los valores en el formulario y presionar "Editar":
    - o Aún no se guardan los cambios, pero Angular debe permitir modificar los valores.
  5. Revisar la consola del navegador para confirmar que no hay errores.
- 

## Conclusión

- Se creó el formulario de edición usando Bootstrap.
- Se configuró Two-Way Binding con `[ngModel]` para sincronizar el formulario con el objeto `producto`.
- Se importó `FormsModule` en `EditarProductoComponent` para habilitar `ngModel`.
- El botón "Editar" ya carga los datos del producto en el formulario.
- El método `onSubmit()` está definido pero aún no implementado.

Con estos cambios, la pantalla de edición ya permite **cargar y modificar los datos de un producto**, aunque aún no se guardan los cambios. En la siguiente parte, implementaremos la actualización del producto en el backend. 

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)