

Reto: Listado de Productos en Angular



Reto: Crear un Listado Dinámico de Productos en Angular

En este reto, vas a modificar la aplicación de Tienda Online de Productos para mostrar un listado dinámico de productos. También implementarás un formulario para agregar nuevos productos al listado y mostrarás los productos existentes en una lista ordenada.

Resultado del Listado de Productos de la Tienda Online

Tienda Online

Listado de Productos

Pantalón, \$130
Camisa, \$80
Playera, \$50

Agregar Nuevo Producto

Nota Importante: Código del Modelo `Producto`

En este reto NO utilizaremos el componente de producto, ya que lo usaremos en lecciones posteriores, sin embargo para este reto debes definir la clase de modelo `Producto`, que representa cada producto en la lista, e importar esta clase para que pueda ser utilizada en el componente de listado de productos:

```
export class Producto {  
  constructor(  
    public descripcion: string,  
    public precio: number  
  ) {}  
}
```

Descripción del Ejercicio:

Se te solicita desarrollar modificar el componente de listado-productos que cumpla con las siguientes funcionalidades:

- Listado de Productos:**
 - Mostrar una lista de productos que incluya una descripción y un precio para cada producto.
 - La lista de productos debe ser dinámica, es decir, debe actualizarse automáticamente cuando se agregue un nuevo producto.
- Formulario para Agregar Productos:**
 - Crear un formulario que permita al usuario ingresar la descripción y el precio de un nuevo producto.
 - El formulario debe validar que ambos campos (descripción y precio) estén completos y que el precio sea un valor positivo antes de agregar el producto al listado.
- Resultado Esperado:**
 - Cuando se ingresa un nuevo producto en el formulario y se hace clic en "Agregar", el producto debe añadirse al listado y los campos del formulario deben restablecerse.
 - El listado de productos debe mostrar todos los productos existentes, incluidos los recién agregados.

Resultado del Reto:

1. Estructura del HTML del Componente

Este es el código HTML del componente `listado-productos.component.html`:

```
<div class="container mt-3">  
  <h3 class="text-warning">Listado de Productos</h3>
```

```

<ul class="list-group w-50 mx-auto">
  @for (producto of productos; track producto) {
    <li class="list-group-item text-center">
      {{ producto.descripcion }},
      ${{ producto.precio }}
    </li>
  }
</ul>

<h5 class="text-success mt-3">Agregar Nuevo Producto</h5>

<form class="d-flex justify-content-between align-items-center my-4 w-50 mx-auto">
  <input type="text" id="descripcion" name="descripcion"
    placeholder="Descripción Producto" class="form-control me-2 w-75"
    autocomplete="off"
    [(ngModel)]="descripcionInput" />

  <input type="number" id="precio" name="precio"
    placeholder="Precio" class="form-control me-2 w-25"
    [(ngModel)]="precioInput" />

  <button type="submit" class="btn btn-primary"
    (click)="agregarProducto()">Agregar</button>
</form>
</div>

```

2. Código del Componente listado-productos-component.ts

Este es el código TypeScript para el componente ListadoProductosComponent, que manejará la lógica de la aplicación:

```

import { Component } from '@angular/core';
import { ProductoComponent } from '../producto/producto.component';
import { Producto } from '../producto/producto.model';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-listado-productos',
  standalone: true,
  imports: [ProductoComponent, FormsModule],
  templateUrl: './listado-productos.component.html',
  styleUrls: ['./listado-productos.component.css'],
})
export class ListadoProductosComponent {

```

```
productos: Producto[] = [  
  new Producto('Pantalón', 130.0),  
  new Producto('Camisa', 80.0),  
  new Producto('Playera', 50.0),  
];  
  
descripcionInput: string = '';  
precioInput: number | null = null;  
  
agregarProducto() {  
  // Validación simple para evitar productos sin descripción o con precio cero  
  if (this.descripcionInput.trim() === '' || this.precioInput == null ||  
      this.precioInput <= 0) {  
    console.log('Debe ingresar una descripción y un precio válidos');  
    return;  
  }  
  
  const producto = new Producto(this.descripcionInput, this.precioInput);  
  this.productos.push(producto);  
  
  // Limpia los campos de entrada después de agregar el producto  
  this.descripcionInput = '';  
  this.precioInput = 0;  
}  
}
```

Resultado Esperado

Al finalizar este ejercicio, tu componente `listado-productos-component` debería mostrar un listado de productos dinámico, con un formulario que permita agregar nuevos productos al listado. Cada vez que se agregue un producto, el listado debe actualizarse automáticamente y los campos del formulario deben restablecerse.



Tienda Online

Listado de Productos

Pantalón, \$130
Camisa, \$80
Playera, \$50

Agregar Nuevo Producto

Descripción Producto Precio **Agregar**

Puntos clave de este reto

- **Validación del Formulario:** Asegúrate de que el usuario no pueda agregar productos con descripciones vacías o precios inválidos.
- **Interactividad:** El listado debe reflejar automáticamente los cambios cuando se agregan nuevos productos.
- **Limpieza del Formulario:** Después de agregar un producto, el formulario debe restablecerse para permitir la entrada de un nuevo producto sin tener que borrar manualmente los campos.

Con este reto reforzarás todos los temas estudiados hasta el momento, como es trabajar con formularios, validar entradas de usuario, y manejar colecciones dinámicas en Angular y TypeScript.

Saludos!

Ing. Ubaldo Acosta

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