

## Prueba Técnica Detallada

### Sección 1: Creación de una Aplicación CRUD Simple

#### Objetivo:

Desarrollar una aplicación Angular que permita a los usuarios gestionar una lista de productos, incluyendo las operaciones de Crear, Leer, Actualizar y Eliminar (CRUD).

#### Requerimientos Funcionales:

##### 1. Listado de Productos:

- Crear un componente llamado ProductListComponent que muestre una tabla con la lista de productos.
- Cada fila debe mostrar el nombre del producto, la descripción, el precio y un botón de "Editar" y otro de "Eliminar".
- Implementar paginación en la tabla para que solo se muestren 10 productos por página.

##### 2. Creación y Edición de Productos:

- Crear un componente llamado ProductFormComponent que incluya un formulario para crear o editar un producto.
- El formulario debe contener los siguientes campos:
  - Nombre del producto (requerido, mínimo 3 caracteres).
  - Descripción (opcional).
  - Precio (requerido, numérico, mayor a 0).
- Al guardar el formulario:
  - Si es un nuevo producto, agregarlo a la lista.
  - Si es una edición, actualizar el producto existente en la lista.

- Implementar validaciones de formularios y mostrar mensajes de error debajo de cada campo si no cumple con los criterios.

### 3. Eliminación de Productos:

- Al hacer clic en el botón "Eliminar" en la tabla de productos, mostrar un diálogo de confirmación.
- Si el usuario confirma, eliminar el producto de la lista.
- Si el usuario cancela, cerrar el diálogo sin eliminar el producto.

### 4. Vista Detallada del Producto:

- Crear un componente llamado ProductDetailComponent que muestre la información detallada del producto seleccionado.
- Mostrar todos los campos del producto, incluyendo un botón "Volver" para regresar al listado.

## Requerimientos Técnicos:

### 1. Uso de Servicios:

- Implementar un servicio llamado ProductService que maneje la lógica de gestión de productos.
- El servicio debe incluir métodos para obtener la lista de productos, agregar un nuevo producto, actualizar un producto existente y eliminar un producto.
- Inicialmente, almacena los productos en un array dentro del servicio.

### 2. Enrutamiento:

- Configurar las rutas de la aplicación utilizando el módulo de enrutamiento de Angular (RouterModule).
- Utilizar el RouterLink para la navegación entre componentes.

### 3. Formulario Reactivo:

- Implementar los formularios de creación y edición utilizando formularios reactivos (FormGroup y FormControl).

- Asegúrate de incluir validaciones en los campos del formulario.

#### 4. Paginación:

- Implementar paginación en el ProductListComponent utilizando Angular Material o una solución personalizada.

#### 5. Estilo y Diseño:

- Aplicar un diseño básico utilizando CSS o Angular Material para asegurar que la interfaz sea clara y fácil de usar.

---

## Sección 2: Optimización de la Aplicación

### Objetivo:

Optimizar la aplicación CRUD desarrollada en la Sección 1 para mejorar su rendimiento y escalabilidad.

### Requerimientos Funcionales:

#### 1. Lazy Loading:

- Modificar la configuración de enrutamiento para implementar Lazy Loading en los módulos de creación/edición y detalle de productos.
- Asegúrate de que los módulos se carguen solo cuando el usuario navega a esas rutas.

#### 2. Change Detection Strategy:

- En el ProductListComponent, cambiar la estrategia de detección de cambios a OnPush.
- Verificar que la tabla de productos se siga actualizando correctamente cuando se agreguen, editen o eliminen productos.

#### 3. Guard para la Ruta de Edición:

- Implementar un guard CanDeactivate en la ruta de edición (/products/:id/edit).

- El guard debe preguntar al usuario si desea abandonar la página sin guardar los cambios cuando se detecten modificaciones no guardadas en el formulario.

---

## Sección 3: Pruebas Unitarias

### Objetivo:

Escribir pruebas unitarias para asegurar la calidad del código en la aplicación CRUD desarrollada.

### Requerimientos Técnicos:

#### 1. Prueba del ProductService:

- Escribir pruebas unitarias que verifiquen:
  - La correcta adición de un producto.
  - La actualización de un producto existente.
  - La eliminación de un producto.
  - La obtención de la lista de productos.
- Mockear cualquier dependencia externa (por ejemplo, si se usara una API real).

#### 2. Prueba del ProductListComponent:

- Verificar que el componente lista correctamente los productos.
- Verificar que la paginación funciona correctamente.
- Probar la interacción del usuario, como la eliminación de un producto y la navegación a la página de edición.

#### 3. Prueba del ProductFormComponent:

- Verificar que las validaciones del formulario funcionan correctamente.
- Probar que el formulario se inicializa con los valores correctos cuando se edita un producto.



- Probar el comportamiento al intentar guardar un formulario con datos inválidos.

---

## Entrega y Evaluación

### Entrega:

- Subir su código a un repositorio en GitHub o GitLab, incluyendo instrucciones claras sobre cómo instalar las dependencias y ejecutar la aplicación.
- Debe también incluir instrucciones sobre cómo ejecutar las pruebas unitarias.