

Práctica 5: Aplicación de Gestión de Tareas con Navegación y Carga Diferida

Objetivo:

El objetivo de esta práctica es que los estudiantes desarrollen desde cero una aplicación móvil utilizando Ionic y Angular para gestionar tareas. Los estudiantes deberán ser capaces de implementar conceptos aprendidos durante el curso, tales como:

- Creación de páginas para la visualización, creación y edición de tareas.
- Implementación de la navegación entre estas páginas utilizando rutas.
- Uso de lazy loading para optimizar la carga de las páginas.
- Gestión de tareas a través de un servicio que interactúe con el almacenamiento local (localStorage) para persistencia de datos.
- Diseño de una interfaz de usuario sencilla pero funcional y atractiva.

Descripción paso a paso de la práctica:

1. Inicialización del Proyecto

El proyecto debe llamarse **"TareaApp"** y debe ser una aplicación en blanco para partir de una base sencilla, sin ninguna plantilla preconfigurada. La estructura del proyecto deberá tener carpetas y archivos organizados siguiendo las mejores prácticas de desarrollo con Ionic y Angular.

2. Creación de Páginas

A continuación, los estudiantes deben crear las siguientes páginas en su proyecto:

- **Página Home (Inicio):**

Esta página será la principal y mostrará una lista de todas las tareas guardadas en el almacenamiento local. Cada tarea debe tener un título, una descripción y una fecha de vencimiento. La página también debe permitir la navegación hacia la página de creación de una nueva tarea y hacia la página de detalles de una tarea específica.

- **Página Tarea (Crear Tarea):**

Esta página debe permitir al usuario ingresar una nueva tarea. El formulario debe incluir campos para el nombre de la tarea, una breve descripción y la fecha

de vencimiento. Es necesario implementar validaciones para que no se puedan enviar campos vacíos o incorrectos. Al guardar una nueva tarea, la información debe ser almacenada en el almacenamiento local.

- **Página Tarea Detalles (Detalles de Tarea):**

Esta página se debe mostrar cuando el usuario haga clic en una tarea específica desde la página de inicio. En esta página se deben mostrar los detalles completos de la tarea seleccionada (nombre, descripción y fecha de vencimiento). Además, se debe permitir al usuario editar estos datos y guardarlos nuevamente en el almacenamiento local.

3. Creación de un Servicio de Gestión de Tareas

Los estudiantes deben crear un servicio llamado **"TareaService"** que se encargue de gestionar todas las operaciones relacionadas con las tareas. Este servicio debe ser responsable de:

- **Obtener todas las tareas** almacenadas en el almacenamiento local.
- **Agregar una nueva tarea** al almacenamiento local.
- **Actualizar una tarea existente.**
- **Eliminar una tarea.**

El servicio debe utilizar **localStorage** para almacenar y recuperar las tareas, asegurándose de que los datos sean persistentes entre sesiones.

4. Implementación de la Navegación entre Páginas

Los estudiantes deben configurar la navegación entre las páginas utilizando rutas. Esto implica configurar el sistema de navegación de la aplicación para que:

- Desde la página **Home** (Inicio), los usuarios puedan ir a la página **Tarea** para crear una nueva tarea o a la página **Tarea Detalles** para editar una tarea existente.
- En la página **Tarea Detalles**, los usuarios deben poder editar los datos de la tarea seleccionada y guardarlos de nuevo en el almacenamiento local.

5. Lazy Loading

La página **Tarea** y la página **Tarea Detalles** deben implementarse con lazy loading. Esto significa que estas páginas no se cargarán hasta que el usuario navegue a ellas, lo que mejora el rendimiento de la aplicación al reducir el tiempo de carga inicial.

6. Diseño de la Interfaz de Usuario (UI)

- **Página Home:**

Debe mostrar una lista de tareas utilizando una lista (por ejemplo, una lista con elementos de tareas). Cada tarea debe tener un botón o enlace que permita navegar a la página de detalles para editarla. Además, debe haber un botón que redirija a la página para crear una nueva tarea.

- **Página Tarea:**

Esta página debe contener un formulario sencillo para crear una nueva tarea, con campos para el nombre, la descripción y la fecha de vencimiento. Debe contar con un botón para guardar la tarea y otro para cancelar la operación.

- **Página Tarea Detalles:**

Aquí se deben mostrar los detalles de la tarea seleccionada. Los usuarios deben poder editar los campos de nombre, descripción y fecha, y un botón debe permitir guardar los cambios. El diseño debe ser claro y amigable para el usuario.

7. Validaciones de Formulario

La página **Tarea** debe contar con validaciones de los datos ingresados:

- El campo de nombre de la tarea no debe estar vacío.
- El campo de descripción no debe estar vacío.
- La fecha de vencimiento debe ser una fecha futura (no se deben permitir fechas pasadas).

Si los datos son válidos, la tarea se debe agregar al almacenamiento local. Si los datos no son válidos, se debe mostrar un mensaje de error.

8. Pruebas y Depuración

Una vez que la aplicación esté completa, los estudiantes deben probarla en el navegador o en un dispositivo móvil para asegurarse de que:

- Las tareas se guardan correctamente en el almacenamiento local.
- Las páginas de la aplicación se navegan correctamente, incluyendo las páginas de creación y edición de tareas.

- El proceso de edición de tareas funciona correctamente y se persiste la información después de que la aplicación se cierre y se vuelva a abrir.

9. Entregables de la práctica

1. **Código fuente completo del proyecto**, subido a un repositorio de GitHub.
2. **Capturas de pantalla** mostrando la aplicación corriendo en un dispositivo móvil, mostrando:
 - La lista de tareas en la página Home.
 - El formulario de creación de una nueva tarea en la página Tarea.
 - La vista de detalles y edición de una tarea en la página Tarea Detalles.
3. **Reporte técnico** que explique:
 - La estructura y organización del proyecto.
 - El uso de localStorage para persistencia de datos.
 - El uso de lazy loading en las páginas Tarea y Tarea Detalles.
 - Las validaciones implementadas en el formulario de creación de tareas.
4. **Mostrar la aplicación funcionando de manera local desde su PC.**