

Trabajo Práctico de Angular: Gestión de Personajes de Cómic

Objetivo: Crear una aplicación en Angular para gestionar una lista de personajes de cómic. La aplicación permitirá visualizar, agregar, editar y eliminar personajes utilizando “@Input”, “@Output”, “FormsModule”, data binding, rutas y componentes compartidos.

Requisitos

1. Componentes principales:

- “AppComponent”: Componente principal que contiene el menú de navegación.
- “CharacterListComponent”: Muestra una lista de personajes de cómic. Utiliza un componente hijo “CharacterCardComponent” para cada personaje.
- “CharacterDetailComponent”: Muestra los detalles de un personaje seleccionado. Permite editar la información del personaje.
- “CharacterFormComponent”: Formulario para agregar o editar un personaje. Debe manejar tanto la creación como la edición de personajes.
- “SharedModule”: Un módulo que contiene el componente “CharacterCardComponent” y otros posibles componentes reutilizables.

2. Rutas:

- La aplicación debe manejar al menos tres rutas:
 - “/characters”: Página principal que muestra la lista de personajes.
 - “/characters/:id”: Página de detalles de un personaje, donde se podrán ver y editar los datos del personaje seleccionado.
 - “/add-character”: Página que contiene el formulario para agregar un nuevo personaje.

3. Data binding:

- Utiliza two-way data binding para manejar los formularios de creación y edición de personajes.

4. @Input y @Output:

- El componente “CharacterCardComponent” debe recibir el personaje como “@Input”.
- Utiliza “@Output” en “CharacterCardComponent” para emitir un evento al hacer clic en “Eliminar” o “Ver detalles”, permitiendo al componente padre (“CharacterListComponent”) gestionar la eliminación o la navegación a la página de detalles.

5. FormsModule:

- El formulario en “CharacterFormComponent” debe utilizar el FormsModule para gestionar la creación y edición de personajes, validando los datos antes de permitir el envío del formulario.

6. Servicios:

- Crea un servicio “CharacterService” que maneje la lógica de negocio de la aplicación, con métodos para:
 - Obtener la lista de personajes.
 - Obtener un personaje específico por su “id”.
 - Agregar un nuevo personaje.
 - Editar un personaje existente.
 - Eliminar un personaje.

7. Modelo de datos:

- El modelo de datos para los personajes debe incluir al menos los siguientes atributos:
 - “id”: Un identificador único.
 - “name”: Nombre del personaje.
 - “alias”: Alias o apodo del personaje.
 - “powers”: Lista de habilidades o poderes del personaje.
 - “description”: Una breve descripción del personaje.

Pasos sugeridos para la implementación:

1. Crear el proyecto Angular y configurar las rutas principales de la aplicación.
2. Crear los componentes mencionados (“CharacterListComponent”, “CharacterDetailComponent”, “CharacterFormComponent”, “CharacterCardComponent”).

3. Configurar el “FormsModule” y utilizar data binding en el formulario para la creación y edición de personajes.
4. Crear y configurar el servicio “CharacterService” para manejar la lógica de datos.
5. Implementar la funcionalidad de @Input y @Output en los componentes correspondientes para pasar datos entre componentes padres e hijos.
6. Diseñar una interfaz básica para la visualización de la lista de personajes, el formulario de creación/edición y los detalles de los personajes.
7. Agregar validaciones en el formulario y mostrar mensajes de error en caso de datos inválidos.
8. Probar la navegación entre las rutas de la aplicación para asegurarte de que el flujo sea el esperado.