



Ejercicio de Navidad - Entrega: 11 de Enero de 2026

Objetivo

Ampliar la funcionalidad de la aplicación implementando un sistema de roles de usuario y gestión de imágenes de perfil.

Requisitos del Ejercicio

1. Usuario Administrador

Debes implementar un sistema de roles de usuario que permita distinguir entre usuarios normales y administradores.

Tareas a realizar:

- **Modificar el modelo de usuario:** Añadir un campo `rol` a la estructura de usuario que pueda tener los valores '`usuario`' o '`administrador`'.
- **Crear usuario administrador:** Asegúrate de que exista al menos un usuario con rol '`administrador`' en la base de datos. Puedes hacerlo:
 - Modificando directamente la base de datos, o
 - Añadiendo una funcionalidad de registro especial para administradores, o
 - Creando un script de inicialización
- **Modificar el servicio de autenticación:** Actualizar AuthService para que maneje correctamente el rol del usuario autenticado.
- **Modificar la navegación:** En el componente NavComponent, añadir lógica para que cuando un usuario con rol '`administrador`' haga login, aparezca un nuevo enlace en el menú de navegación que permita acceder al CRUD de usuarios.

2. CRUD de Usuarios

Debes implementar un CRUD completo para la gestión de usuarios, siguiendo el mismo patrón que los CRUDs ya implementados en el proyecto (como MisPersonajes, FiltrarPersonajes, etc.).

Estructura requerida:

- **Componente de formulario:** Un componente para crear/editar usuarios (similar a `miPersonaje.component.ts`)
- **Componente de listado:** Un componente para mostrar la lista de usuarios con opciones de editar y eliminar (similar a `listadoMisPersonajes.component.ts`)
- **Servicio:** Un servicio para gestionar las operaciones CRUD de usuarios (crear, leer, actualizar, eliminar)
- **Interfaz:** Definir la interfaz TypeScript para el modelo de usuario
- **Ruta protegida:** Crear una nueva ruta `/usuarios` que solo sea accesible para administradores



Funcionalidades del CRUD:

- Listar todos los usuarios
- Crear nuevo usuario
- Editar usuario existente
- Eliminar usuario
- Validación de formularios
- Manejo de errores

3. Gestión de Imágenes de Usuario

Debes añadir la capacidad de gestionar imágenes de perfil para los usuarios.

Tareas a realizar:

- **Modificar el modelo de usuario:** Añadir un campo `imagen` (o `avatar`) a la estructura de usuario que almacene la URL o ruta de la imagen.
- **Actualizar el formulario de registro:** Modificar el componente `RegistroComponent` para incluir un campo de carga de imagen.
- **Actualizar el formulario de edición de usuario:** En el CRUD de usuarios, añadir la capacidad de subir/cambiar la imagen de perfil.
- **Mostrar imágenes en la interfaz:**
 - Mostrar la imagen del usuario en el menú de navegación (junto al nombre)
 - Mostrar las imágenes en el listado de usuarios del CRUD
 - Mostrar la imagen en el formulario de edición
- **Gestión de archivos:** Decidir cómo gestionar las imágenes:
 - Opción 1: Almacenar como base64 en la base de datos
 - Opción 2: Subir a un servicio de almacenamiento (recomendado para producción)
 - Opción 3: Almacenar en la carpeta `assets` del proyecto (más simple para desarrollo)

Recomendación: Para este ejercicio, puedes usar la opción de base64 o almacenamiento local en `assets`, ya que es más simple de implementar.



Entregables

1. **Código fuente completo** del proyecto con todas las modificaciones
2. **README actualizado** explicando las nuevas funcionalidades implementadas
3. **Capturas de pantalla** mostrando:
 - o Login como administrador
 - o Menú de navegación con el enlace al CRUD de usuarios
 - o Listado de usuarios
 - o Formulario de edición de usuario con imagen
 - o Usuario con imagen en el menú de navegación

Fecha de Entrega

11 de Enero de 2026

Notas Importantes

- Asegúrate de que el proyecto compile sin errores
- Prueba todas las funcionalidades antes de entregar
- Mantén la estructura y convenciones del proyecto original
- Documenta cualquier decisión de diseño importante en comentarios del código
- Si encuentras problemas técnicos, documenta cómo los resolviste

Pistas y Ayuda

- Revisa cómo están implementados los CRUDs existentes
(MisPersonajes, FiltrarPersonajes)
- El AuthService ya maneja signals, úsalos para gestionar el rol del usuario
- Para los guards, puedes crear un nuevo AdminGuard o modificar el AuthGuard existente
- Para la carga de imágenes, puedes usar `<input type="file">` y FileReader para convertir a base64
- Recuerda actualizar las interfaces TypeScript cuando modifiques el modelo de datos

¡Mucha suerte con el ejercicio, y... si tenéis cualquier duda, ya sabéis como contactarme! 🎄🎅
Siiiiii, da igual si es navidad, nochebuena o nochevieja 😊