

# Aplicación juego de Rol

## 1. Estructura

### Interfaz con Tabs:

- **Tab de Creación de Personajes:** aquí se podrá crear un personaje nuevo.
- **Tab de Lista y Detalles de Personaje:** aquí se podrá ver la lista de personajes y los detalles de uno una vez seleccionado.
- **Tab de Inventario:** aquí se podrá ver los objetos y gestionar el equipamiento.

## 2. Creación de Personaje

- Los personajes tendrán los siguientes atributos:
  - **Nombre:** Texto introducido por el usuario.
  - **Clase:** Combobox con las distintas clases.
  - **Género:** Selector de género con radiobutton.
  - **Atributos (Fuerza, Inteligencia, Destreza, Resistencia):** Cada atributo se va a configurar mediante sliders.
  - **Foto del Personaje**

## 3. Objetos

- **Atributos:** Tendrá los mismos atributos que un personaje y se sumarán cuando se equipen
- **Tipo de Objeto:** Cada objeto tendrá un tipo (Espada, Casco, Botas, etc.).
- El personaje podrá **equipar y desequipar objetos**. Solo se puede equipar un objeto de cada tipo (ej, un solo casco a la vez).
- Un personaje podrá tener muchos objetos con el mismo nombre como objetos repetidos.
- Habrá un máximo de 20 objetos en el inventario.

## 4. Diseño

- La interfaz debe ser intuitiva y bonita, utilizando **paneles y grids** para organizar los elementos.

## 5. Base de Datos

- Toda la información deberá almacenarse en una base de datos MySQL.
- La aplicación debe tener las funcionalidades CRUD (Crear, Leer, Actualizar y Borrar).

## Clases:

La estructura de la aplicación deberá incluir las siguientes clases:

- Clase personaje con una lista de objetos y sus atributos
- Clase objetos con los atributos, su nombre y si está equipado
- Clase repositorio en la que se llevarán a cabo todos los accesos a la base de datos para separarlo de la lógica de aplicación.

## Base de datos:

```
CREATE DATABASE JuegoRoIDB; USE JuegoRoIDB; CREATE TABLE Personaje ( id INT
PRIMARY KEY AUTO_INCREMENT, nombre VARCHAR(100) NOT NULL, clase
VARCHAR(50) NOT NULL, genero VARCHAR(20) NOT NULL, fuerza INT CHECK (fuerza
BETWEEN 0 AND 100), inteligencia INT CHECK (inteligencia BETWEEN 0 AND 100),
destreza INT CHECK (destreza BETWEEN 0 AND 100), resistencia INT CHECK
(resistencia BETWEEN 0 AND 100), foto VARCHAR(255) ); CREATE TABLE Objeto ( id INT
PRIMARY KEY AUTO_INCREMENT, nombre VARCHAR(100) NOT NULL, tipo
VARCHAR(50) NOT NULL, fuerza INT DEFAULT 0 CHECK (fuerza BETWEEN 0 AND 100),
inteligencia INT DEFAULT 0 CHECK (inteligencia BETWEEN 0 AND 100), destreza INT
DEFAULT 0 CHECK (destreza BETWEEN 0 AND 100), resistencia INT DEFAULT 0 CHECK
(resistencia BETWEEN 0 AND 100) ); CREATE TABLE Inventario ( personaje_id INT,
objeto_id INT, equipado BOOLEAN DEFAULT FALSE, PRIMARY KEY (personaje_id,
objeto_id), FOREIGN KEY (personaje_id) REFERENCES Personaje(id) ON DELETE
CASCADE, FOREIGN KEY (objeto_id) REFERENCES Objeto(id) ON DELETE CASCADE );
```

Para equipar un objeto haremos una consulta en la que comprobaremos si ya hay algún objeto equipado cuyo tipo coincida con el del objeto a equipar.

La tabla Inventario hará de tabla intermedia para asociar personajes a varios objetos.