**Evaluación : “Rick and Morty Data Manager”**

**Objetivo:**

Desarrollar una aplicación en Python que permita gestionar usuarios y personajes del universo de **Rick and Morty**, aplicando:

* Programación orientada a objetos (POO)
* Conexión a base de datos (SQLite)
* Operaciones CRUD
* Consumo de API externa
* Deserialización de JSON
* Registro de datos en base de datos

**Enunciado General**

Desarrolla una aplicación en Python que permita:

1. Registrar usuarios con nombre y email.
2. Conectarse a la API oficial de Rick and Morty: https://rickandmortyapi.com/
3. Consultar personajes desde la API y almacenarlos en una base de datos SQLite.
4. Implementar un CRUD completo para los personajes almacenados.
5. Mostrar los personajes guardados y permitir su edición o eliminación.

**Parte 1: Programación Orientada a Objetos (25 pts)**

Crea las siguientes clases:

* Usuario: con atributos id, nombre, email
* Personaje: con atributos id, nombre, especie, origen, estado
* RickAndMortyAPI: clase para consumir y deserializar datos de la API

**Parte 2: Base de Datos y CRUD (25 pts)**

Usa SQLite para:

* Crear las tablas usuarios y personajes
* Implementar funciones para:
  + Crear, leer, actualizar y eliminar personajes
  + Registrar usuarios

**Parte 3: Consumo de API Externa (25 pts)**

* Usa requests para consumir la API de Rick and Morty.
* Obtén una lista de personajes (por ejemplo, los primeros 10).

**Parte 4: Deserialización y Registro (25 pts)**

* Deserializa el JSON recibido de la API.
* Registra los personajes en la base de datos.

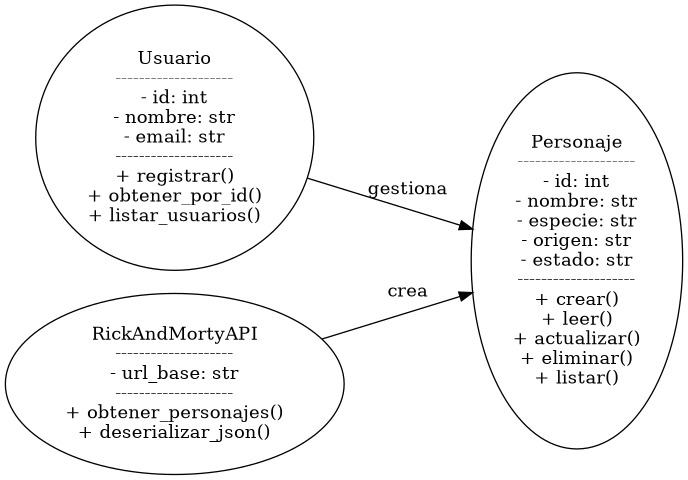
**Considerar en la implementación**

* Completar funciones adicionales como edición de personajes
* Agregar validaciones o interfaz de usuario (por consola)
* Extender la clase RickAndMortyAPI para obtener episodios o ubicaciones (universos)
* Crear un menú interactivo para navegar entre opciones (por consola)

**Evaluación**

| **Criterio** | **Puntaje** |
| --- | --- |
| Clases y POO | 25 pts |
| CRUD con SQLite | 25 pts |
| Consumo de API externa | 25 pts |
| Deserialización y registro de datos | 25 pts |
| **Total** | 100 pts |

**UML:**



**Clases y Detalles:**

**Usuario**

* **Atributos**:
  + id: Identificador único
  + nombre: Nombre del usuario
  + email: Correo electrónico
* **Métodos**:
  + registrar()
  + obtener\_por\_id()
  + listar\_usuarios()

**Personaje**

* **Atributos**:
  + id: Identificador del personaje
  + nombre: Nombre del personaje
  + especie: Especie del personaje
  + origen: Lugar de origen
  + estado: Estado (vivo, muerto, desconocido)
* **Métodos**:
  + crear(), leer(), actualizar(), eliminar(), listar()

**RickAndMortyAPI**

* **Atributos**:
  + url\_base: URL base de la API
* **Métodos**:
  + obtener\_personajes()
  + deserializar\_json()

**Relaciones:**

* Usuario **gestiona** Personaje
* RickAndMortyAPI **crea** Personaje a partir de datos externos