Trabajo Final de Lenguaje de Programacion

Integrante: Jose Ramon Gascon Enriquez

El proyecto utiliza múltiples paradigmas de programación para gestionar juegos y sesiones de manera eficiente:

1. \*\*Orientado a Objetos (POO)\*\*

- \*\*Clases y objetos\*\*: `Juego`, `SesionJuego` y `AgendaJuegos` encapsulan datos (atributos) y comportamientos (métodos).

- \*\*Encapsulación\*\*: Los atributos se modifican mediante métodos como `editar\_juego`.

2. \*\*Procedimental\*\*

- \*\*Funciones independientes\*\*: En `logic.py`, funciones como `generar\_torneo` realizan tareas específicas.

- \*\*Flujo secuencial\*\*: El menú en `main.py` sigue un enfoque paso a paso.

3. \*\*Funcional\*\*

- \*\*Listas por comprensión\*\*: Usadas en `buscar\_juegos\_por\_categoria` para filtrar datos.

- \*\*Inmutabilidad\*\*: `generar\_torneo` devuelve una lista nueva sin modificar la original.

4. \*\*Asíncrono\*\*

- \*\*Async/Await\*\*: Operaciones de E/S en `storage.py` (ej. `guardar\_juegos`) evitan bloqueos.

5. \*\*Modular\*\*

- \*\*Separación por responsabilidades\*\*: Módulos como `models.py` (datos) y `logic.py` (reglas) mantienen el código organizado.

\*\*Conclusión\*\*:

La combinación de paradigmas permite un diseño flexible: POO para modelar entidades, procedimental para el flujo, funcional para procesamiento de datos, asíncrono para eficiencia y modular para mantenibilidad.