

Colección de videojuegos

Autor: Jose Adrián Aglio Gascón

Fecha: 24/05/2024

Indice

2.Introducción

3.Herramientas y Métodos

4.Perspectiva Estática

5.Perspectiva Dinámica

6.Conclusiones

7.Bibliografía y Webgrafía

2.Introducción

En este proyecto se va a realizar una aplicación donde podremos llevar una gestión de una colección de videojuegos para llevar un control sobre ellos, pudiendo introducir tanto datos del videojuego, como el género al que pertenece y la plataforma en la que esta disponible, y creando un usuario para cada cliente teniendo cada uno su propia colección.

3.Herramientas y Métodos

Para realizar este proyecto he utilizado:

- MongoDB para poder conectar la base de datos con la aplicación.
- ChatGPT para escribir el código con el lenguaje de programación Python para conectar la base de datos.
- VisualStudio Code para conectar la base de datos y poder modificarla con el código de Python.
- Draw.io para crear el modelo entidad-relación de la base de datos.
- Umbrello para crear el caso de uso y el diagrama de clases para la aplicación.

4.Perspectiva Estática

-E/R(Entidad-Relación)

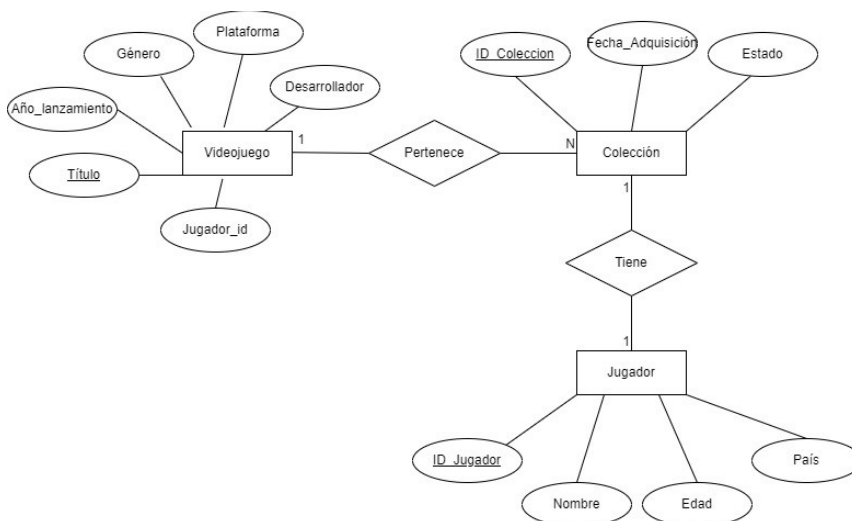


Fig 1.1

En la **Figura 1.1** tenemos el diagrama Entidad-Relación de la base de datos teniendo como claves principales Título, en la tabla Videojuego, ID_Colección en la tabla Colección, y ID_Jugador en la tabla Jugador.

-Paso a tablas

```
VIDEOJUEGO:  
Título+Año_lanzamiento+Género+Plataforma+Desarrollador  
+Jugador_id  
Clau Aliena: Jugador_id → Jugador(ID)  
COLECCIÓN: ID_Coleccion+Fecha_Adquisición+Estado  
JUGADOR: ID_Jugador+Nombre+Edad+País  
Pertenece: Título_v+ID_col  
Clau Aliena: Título_v → Videojuego(Título)  
ID_col → Colección(ID_Coleccion)  
Tiene: ID_col+ID_jug  
Clau Aliena: ID_jug → Jugador(ID_Jugador)
```

Fig 1.2

En la **Figura 1.2** tenemos el paso a tablas del diagrama E-R donde se indican los atributos y sus claves principales y claves externas.

-DDL

DDL, o Data Definition Language, es un conjunto de comandos utilizados para definir y modificar la estructura de las bases de datos.

Por ejemplo tenemos estas lineas:

```
self.collection = self.db['jugadores']
```

Esta linea crea la colección “jugadores”.

```
self.videojuegos = self.db['videojuegos']
```

Esta linea crea la colección “videojuegos”.

-DML

DML, o Data Manipulation Language, es un conjunto de comandos utilizados para manipular los datos almacenados en una base de datos.

Por ejemplo tenemos estas lineas:

```
self.collection.insert_one({"ID": siguiente_id, "Nombre": nombre, "Edad": edad, "Pais": pais})
```

Esta linea inserta un nuevo documento en la colección “jugadores”.

```
self.videojuegos.insert_one({"Titulo": titulo, "Año_Lanzamiento": año_lanzamiento, "Genero": genero, "Desarrollador": desarrollador, "Plataforma": plataforma, "Jugador_ID": self.jugador_id})
```

Esta linea inserta un nuevo documento en la colección “videojuegos”.

```
self.colecciones.insert_one({"Fecha_Adquisicion": fecha_adquisicion, "Estado": estado, "Jugador_ID": self.jugador_id})
```

Esta linea inserta un nuevo documento en la colección “colección”.

-DQL

DQL, o Data Query Language, se utiliza específicamente para realizar consultas o consultas de datos en una base de datos. Aquí utilizamos el Select para elegir una tabla y poder modificar un atributo.

Por ejemplo tenemos estas lineas:

```
self.collection.find_one({"Nombre": nombre})
```

```
max_id_jugador = self.collection.find_one(sort=[("ID", -1)])
```

```
videojuegos = self.db['videojuegos'].find({"Jugador_ID": self.jugador_id})
```

Estas lineas realizan consultas a la base de datos para recuperar documentos que cumplan con ciertos criterios de búsqueda.

-DCL

DCL, o Data Control Language es un lenguaje proporcionado por el sistema de gestión de base de datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la base de datos.

Por ejemplo tenemos estas lineas:

```
import tkinter as tk
```

```
import tkinter.simpledialog as simpledialog
```

```

from tkinter import ttk, messagebox

from pymongo import MongoClient, errors

class RegistroJugador(tk.Toplevel)

class iniciarCuenta(tk.Toplevel)

class MenuDatos(tk.Toplevel)

```

Estas líneas definen las clases y las importaciones de módulos necesarios para el funcionamiento del programa.

5. Perspectiva Dinámica

-Sketch

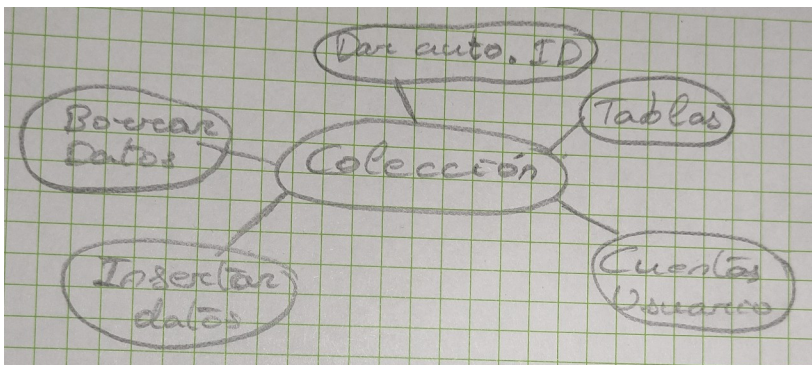


Fig 1.3

En la **Figura 1.3** tenemos un sketch con la idea principal del programa y sus funciones

-Casos de Uso



Fig 1.4

En la **Figura 1.4** tenemos el diagrama de los casos de uso con todas las funciones del programa, explicadas aquí:

-Registrar jugador:

Se utiliza para registrar un jugador proporcionándole una ID única y introduciendo su nombre, edad y país.

-Iniciar sesión:

Se utiliza para que el usuario entre con su cuenta particular donde tendrá su colección guardada en la base de datos.

-Agregar datos:

Tiene la función de agregar los datos que quiera el usuario a la base de datos.

-Ver videojuegos:

Se utiliza para que el usuario pueda ver que datos y que videojuegos tiene en su colección.

-Borrar videojuegos:

Se utiliza para borrar los videojuegos, junto con los otros datos, de la colección del usuario.

6.Conclusiones

-Resumen de los resultados obtenidos

Tenemos una aplicación con la que podemos registrarnos y introducir datos de los videojuegos que queramos junto con su fecha de adquisición y estado para nuestra colección, y también podemos ver la lista de videojuegos de nuestra colección que ya hayamos agregado iniciando sesión en nuestra cuenta.

-Reflexiones sobre el proceso y posibles mejoras futuras

Algunas de las posibles mejoras que se podrían implementar serían que a la hora de agregar videojuegos nos muestre un desplegable con algunos títulos de videojuegos, también añadir alguna foto a la colección, y poder hablar con otros usuarios para comparar colecciones.

7.Bibliografía y Webgrafía

Lo que se utilizó es ChatGPT de OpenAI para escribir el código y corregir algunos errores de la aplicación, draw.io para crear el modelo entidad-relación y umbrello para crear el diagrama de casos de uso.

ChatGPT. (s. f.). <https://chat.openai.com/>

draw.io - free flowchart maker and diagrams online. (s. f.). <https://app.diagrams.net/>

Umbrello Project - Welcome to Umbrello - The UML Modeller. (s. f.). <https://uml.sourceforge.io/>

Repositorio de GitHub

<https://github.com/JoseAdrianAG/MicroProyectoBD2>