**Proyecto de clase de Python aplicado.**

**Fundamentos de Programación**

**Nombre del Proyecto:** Predicción de popularidad de videojuegos mediante análisis de datos en Python

**1. Información General**

* **Nombre estudiantes:**
  + **Juan Alejandro Castillo Lopez**
  + **Samuel Londoño Mayorga**
  + **Jhosef Alejandro Rojas Duran**
  + **Juan Sebastián Carrillo**
* **Curso / Grupo: SA1**
* **Fecha de entrega:**
* **Profesor:**
  + **Pablo Carreño**

**2. Título del Proyecto**

**Predicción de popularidad de videojuegos mediante análisis de datos en Python**

**3. Descripción del Proyecto**

* **Propósito:**  
  Analizar datos de videojuegos (precio, calificación, año de publicación, etc.) para generar un índice de popularidad.
* Aprender a manejar estructuras de datos, análisis estadístico y visualización con Python.
* **Público objetivo:**  
  Desarrolladores de videojuegos, analistas de mercado y jugadores interesados en tendencias.
* **Resultado esperado:**  
  Un programa en Python que calcule y muestre un índice de popularidad para una lista de videojuegos, permitiendo ordenar y comparar títulos fácilmente.

**4. Objetivos**

**General:**

* Construir un modelo que permita estimar la popularidad de un videojuego en función de sus características.
* Aplicar librerías como Pandas, Matplotlib y Seaborn para el análisis y visualización de datos.
* Desarrollar un sistema flexible que se adapte a distintos formatos de datos y detecte errores automáticamente.

**Específicos:**

* Utilizar estructuras de datos como listas y diccionarios para almacenar información.
* Aplicar las librerías para manejo y análisis de datos tabulares.
* Implementar normalización de datos y cálculo de índices ponderados.
* Practicar modularización mediante funciones.
* Presentar resultados de forma clara y ordenada.

**5. Requisitos**

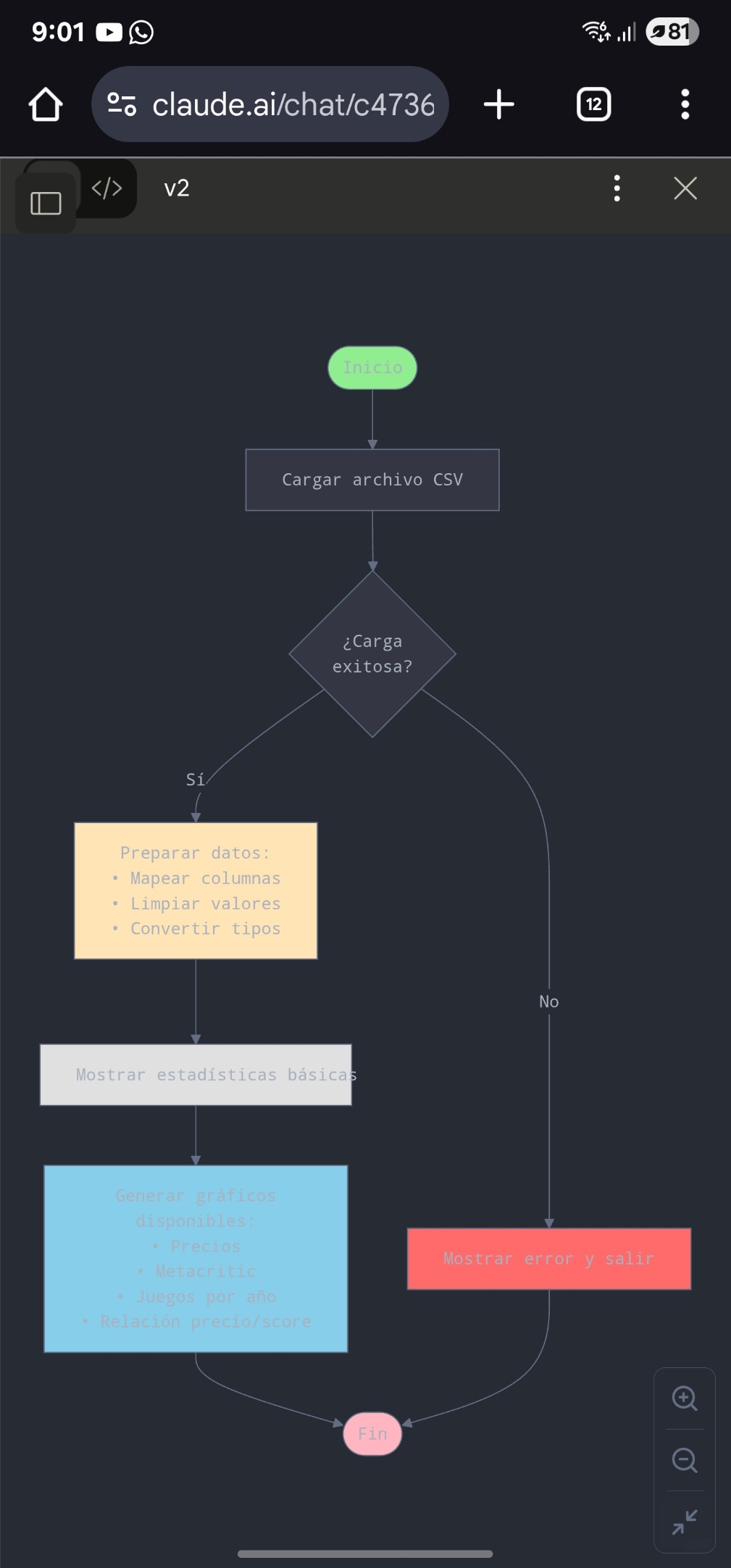
* Python
* **Librerías utilizadas:**   
  Pandas, para manejo de datos  
  Matplotlib y Seaborn para visualización

Tkinter incluye interfaz gráfica

Os y Sys para gestión de archivos y errores

**6. Diseño del Proyecto**

* **Arquitectura o estructura del programa:** El proyecto se estructuró de forma modular, dividiendo el código en funciones específicas para carga de datos, análisis estadístico y visualización.
* Se emplean condicionales para manejar errores y verificar la existencia de columnas clave (precio, metacritic, año\_de\_publicacion).
* Las visualizaciones se generan solo si hay datos válidos, lo que mejora la funcionalidad del programa.
* **Diagrama de flujo**:



* **Interfaz:**
* Las gráficas se presentan en ventanas emergentes usando Matplotlib y Seaborn, con estilos personalizados y títulos explicativos.
* Se diseñó una interfaz gráfica sencilla con Tkinter que permite:

Cargar el archivo de datos.

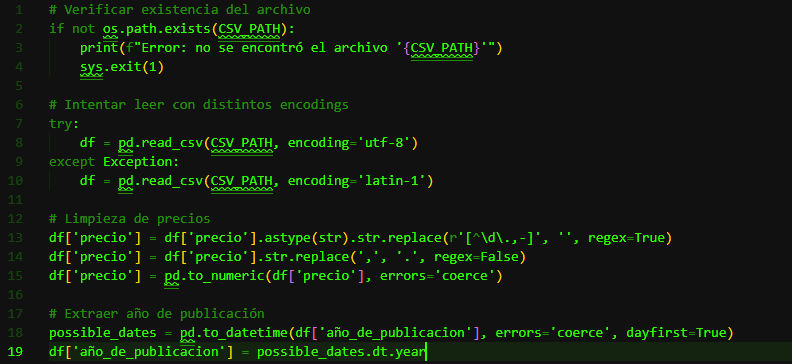
Visualizar gráficos generados con Matplotlib y Seaborn.

Mostrar el índice de popularidad calculado.

Ejecutar la predicción para nuevos videojuegos ingresados manualmente.

**7. Desarrollo**

* El desarrollo del proyecto consistió en crear un programa en Python capaz de analizar datos de videojuegos extraídos desde un archivo CSV. Se implementaron procesos para limpiar la información, identificar columnas clave y generar estadísticas y visualizaciones que permitieran entender mejor las tendencias del mercado.
* **Fragmentos de código relevantes comentados:**



* **Descripción de las funciones principales:**

**Verificación del archivo CSV**: Antes de iniciar el análisis, el programa comprueba si el archivo steam\_games.csv existe en el directorio actual. Si no se encuentra, se muestra un mensaje de error y se detiene la ejecución para evitar fallos posteriores.

**Mapeo de columnas clave**: El programa no depende de nombres fijos de columnas. En su lugar, utiliza un sistema de alias que identifica variantes comunes como price, final\_price, release\_date, y las mapea a los nombres esperados (precio, metacritic, año\_de\_publicacion).

**Limpieza y normalización de datos:** Los valores de la columna precio se limpian eliminando símbolos de moneda y comas, y se convierten a tipo numérico. La columna metacritic también se convierte a valores numéricos, y en el caso de año\_de\_publicacion, se extrae el año desde fechas completas usando pd.to\_datetime() o se convierte directamente si ya está en formato numérico.

**Análisis estadístico:** Si las columnas precio y metacritic están disponibles, se genera un resumen estadístico con describe(), mostrando métricas como media, desviación estándar, mínimo, máximo y percentiles.

**Visualización de datos:** Se generan distintos tipos de gráficos con Matplotlib y Seaborn:

* Histogramas para visualizar la distribución de precios y puntuaciones.
* Gráfico de barras para mostrar la cantidad de juegos publicados por año.
* Línea de tendencia que representa el promedio de puntuación Metacritic por año.
* Diagrama de dispersión que muestra la relación entre precio y puntuación.

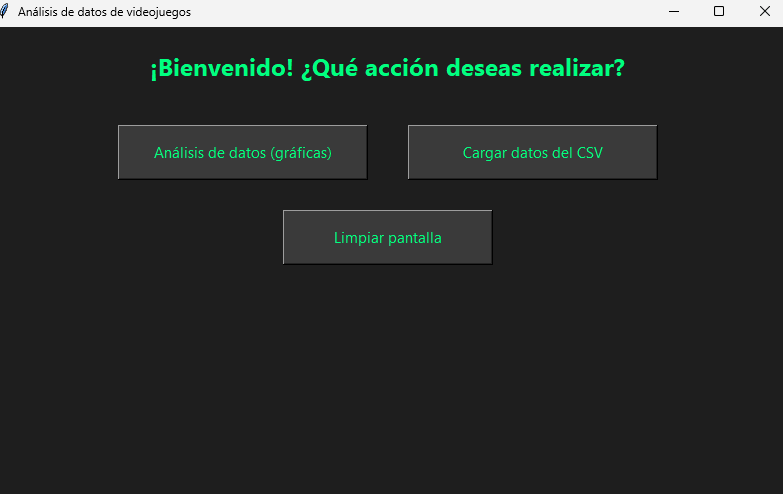
**8. Pruebas y Resultados**

* Se ejecutó el script con distintos archivos CSV para validar la flexibilidad del mapeo de columnas.

Se probaron casos con datos incompletos para verificar la tolerancia a errores.

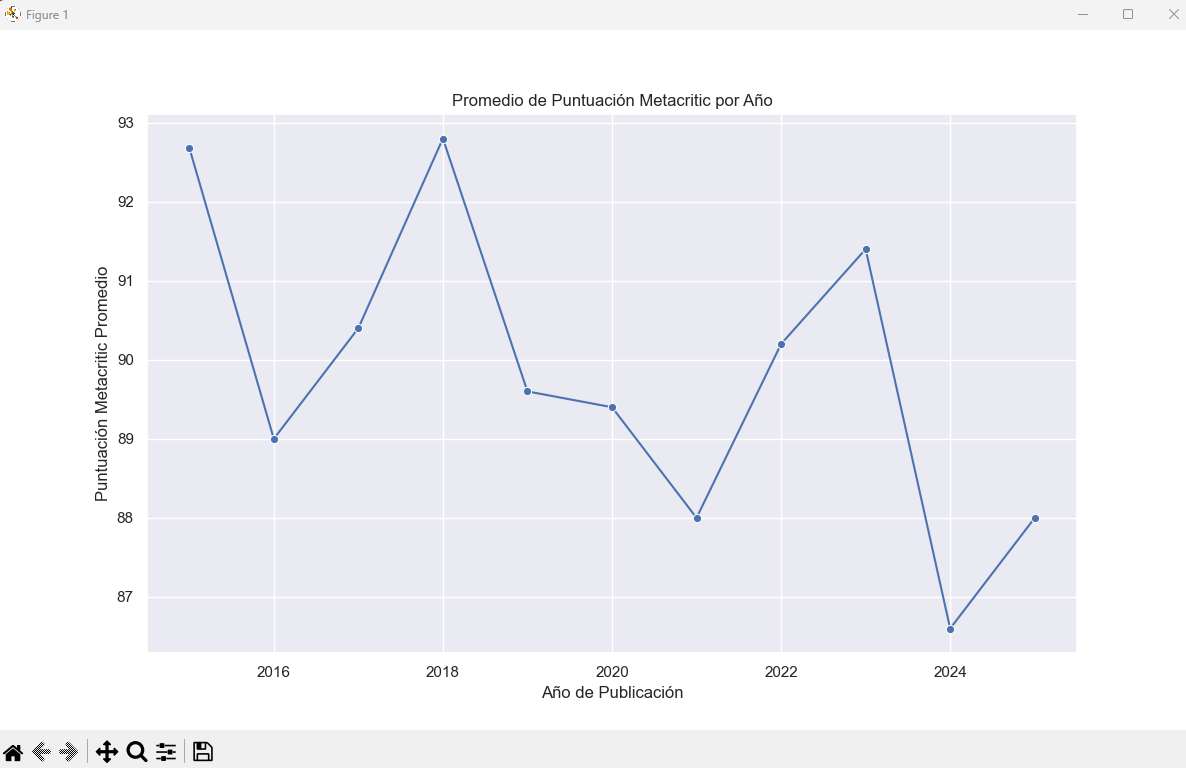
Se observó el comportamiento de las gráficas con distintos rangos de precios y puntuaciones.

* **Capturas de pantalla con ejemplos de ejecución y resultados obtenidos:**









* Se logró visualizar la distribución de precios y puntuaciones con claridad.
* Se identificaron tendencias como el aumento de lanzamientos en ciertos años y la relación inversa entre precio y puntuación en algunos casos.
* El programa demostró ser adaptable a distintos formatos de datos.

**9. Conclusiones**

* La validación de columnas y formatos es esencial para evitar fallos en el análisis.
* Las visualizaciones son clave para interpretar tendencias y relaciones entre variables.
* **Dificultades encontradas:**

En algunos entornos, las gráficas no se mostraban correctamente o bloqueaban la ejecución.

Algunos errores o advertencias eran difíciles de entender por lo que se utiliza el print() para dejar mensajes claros de cuál es el error que daba.

* **Posibles mejoras o ideas futuras:**

Permitir exportar resultados y gráficas en formatos PDF o Excel.

Agrupar videojuegos por género (acción, estrategia, simulación, etc.) y analizar tendencias específicas dentro de cada grupo.

Añadir una función que permita comparar dos o más videojuegos en base a sus métricas y mostrar cuál es más popular o mejor valorado.