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**
- **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:

  que se utilizó para manejo de datos

  → Matplotlib y Seaborn:

  para poder visualizar las gráficas.

  → Tkinter:

  esta se usa para la interfaz gráfica.

  → Os y Sys:

  para gestión de archivos y errores.

6. Arquitectura o estructura del programa

El proyecto fue diseñado siguiendo una arquitectura modular, lo que significa que el código se dividió en bloques o funciones independientes, cada una encargada de una tarea específica dentro del flujo general del programa. Esta estructura facilita la lectura, mantenimiento y escalabilidad del código, ya que permite realizar modificaciones o agregar nuevas funciones sin afectar el resto del sistema.

Entre los módulos principales se destacan:

- Carga y validación de datos: Esta parte del código se encarga de leer el archivo CSV que contiene la información de los videojuegos, verificando su existencia y

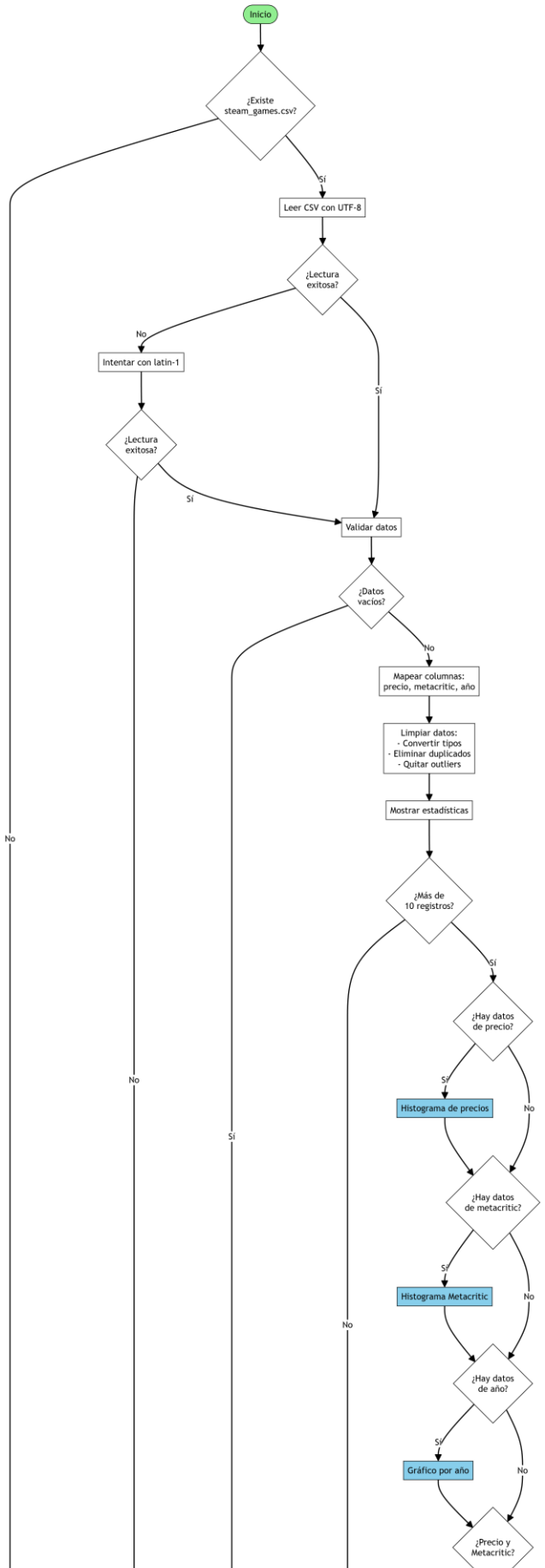
formato antes de procesarlo. Además, se implementan controles de validación para asegurar que las columnas esenciales (como precio, metacritic y año_de_publicacion) estén presentes y contengan datos válidos. En caso contrario, se generan mensajes de advertencia o se detiene el proceso de manera controlada.

- Análisis estadístico: En este módulo se realizan operaciones descriptivas como promedios, conteos, valores máximos y mínimos, así como análisis de correlación entre variables relevantes. Esto permite obtener una visión general del comportamiento del conjunto de datos y detectar posibles tendencias o anomalías.

- Visualización de resultados: Una vez validados y analizados los datos, se generan gráficos dinámicos e interpretativos (por ejemplo, histogramas, diagramas de dispersión o gráficos de barras) que permiten visualizar relaciones entre variables como precio, puntuación o año de publicación. Las visualizaciones se ejecutan solo si los datos son válidos, evitando errores o bloqueos en la ejecución.

- **Diagrama de flujo:**

📊 Diagrama de Flujo - Análisis Steam Games



- **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 proyecto se centró en el desarrollo de un programa en Python diseñado para analizar datos de videojuegos obtenidos a partir de un archivo CSV. Se implementaron procesos de limpieza de datos, identificación de columnas relevantes y generación de estadísticas y visualizaciones que facilitaron la comprensión de las tendencias del mercado.
- **Fragmentos de código relevantes comentados:**

```python
# Verificar existencia del archivo
if not os.path.exists(CSV_PATH):
    print(f"Error: no se encontró el archivo '{CSV_PATH}'")
    sys.exit(1)

# Intentar leer con distintos encodings
try:
    df = pd.read_csv(CSV_PATH, encoding='utf-8')
except Exception:
    df = pd.read_csv(CSV_PATH, encoding='latin-1')

# Limpieza de precios
df['precio'] = df['precio'].astype(str).str.replace(r'[^\d\.,-]', '', regex=True)
df['precio'] = df['precio'].str.replace(',', '.', regex=False)
df['precio'] = pd.to_numeric(df['precio'], errors='coerce')

# Extraer año de publicación
possible_dates = pd.to_datetime(df['año_de_publicacion'], errors='coerce', dayfirst=True)
df['año_de_publicacion'] = possible_dates.dt.year
```

- **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.

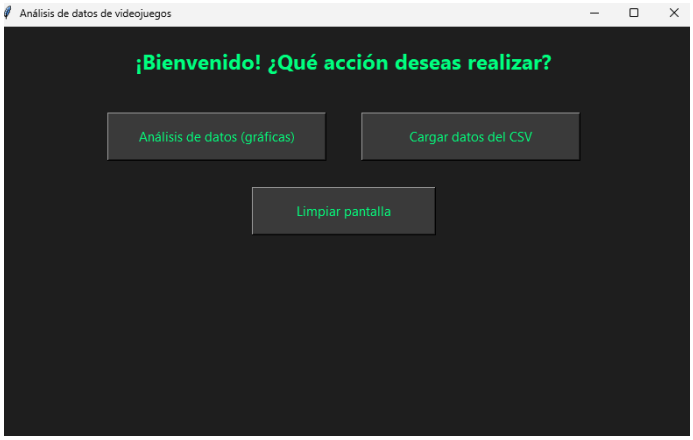
8. Pruebas y Resultados

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

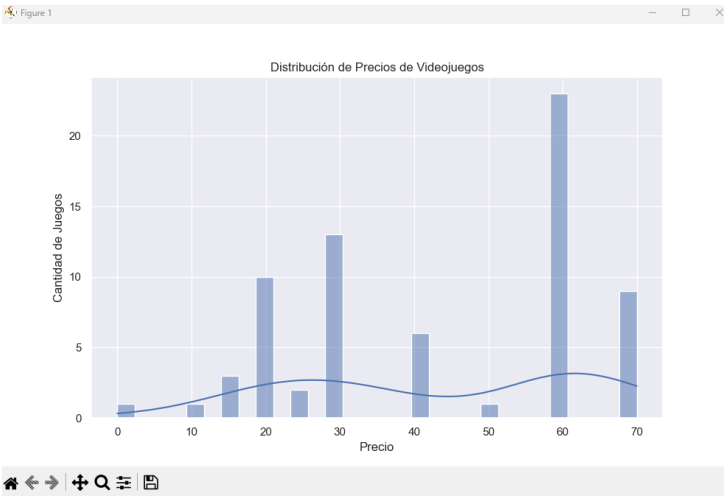
  Se probaron casos con datos incompletos para verificar la tolerancia a errores.

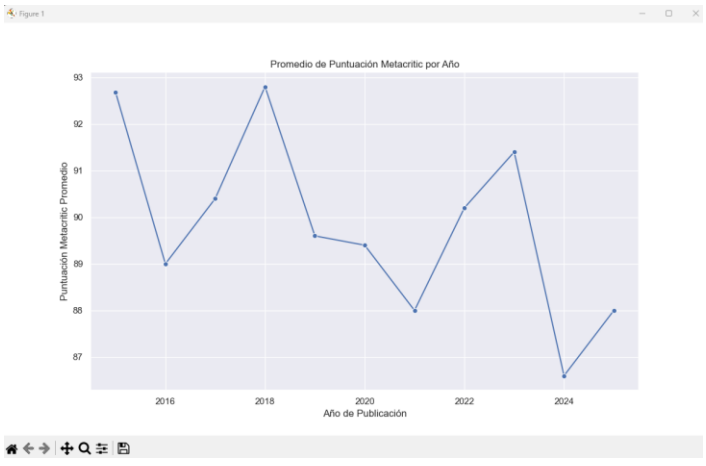
  Se observó el comportamiento de las gráficas con distintos rangos de precios y puntuaciones.

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



titulo	año_de_publicacion	precio	metacritic
Metal Gear Solid V: The Phantom Pain	2015	19.99	93
The Witcher 3: Wild Hunt	2015	29.99	92
Bloodborne	2015	19.99	92
Undertale	2015	9.99	92
Uncharted 4: A Thief’s End	2015	19.99	93
Overwatch	2015	0.00	90
Dark Souls III	2015	29.99	89
The Legend of Zelda: Breath of the Wild	2015	59.99	97
Super Mario Odyssey	2015	59.99	97
Divinity: Original Sin II	2015	14.99	93
Persona 5	2015	19.99	93
Red Dead Redemption 2	2015	59.99	97
God of War	2015	29.99	94
Super Smash Bros. Ultimate	2015	69.99	92
Forza Horizon 4	2015	29.99	92
Resident Evil 2	2015	19.99	93
Sekiro: Shadows Die Twice	2015	29.99	91
Final Fantasy XIV: Shadowbringers	2015	19.99	91
Monster Hunter World – Iceborne	2015	29.99	90
Dark Souls III	2016	59.99	89
Inside	2016	19.99	91
Stardew Valley	2016	14.99	89
Titanfall 2	2016	29.99	89
Doom (2016)	2016	29.99	87
Hollow Knight	2017	14.99	90
The Legend of Zelda: Breath of the Wild	2017	59.99	97
Cuphead	2017	19.99	88
Nier: Automata	2017	39.99	88
Horizon Zero Dawn	2017	29.99	89
Celeste	2018	19.99	94
Red Dead Redemption 2	2018	59.99	97
God of War	2018	39.99	94
Monster Hunter: World	2018	29.99	90
Return of the Obra Dinn	2018	19.99	89
Sekiro: Shadows Die Twice	2019	59.99	90
Control	2019	39.99	85
Resident Evil 2 (Remake)	2019	39.99	93
Disco Elysium	2019	39.99	91
Fire Emblem: Three Houses	2019	59.99	89
Hades	2020	24.99	93
The Last of Us Part II	2020	59.99	93
Ghost of Tsushima	2020	59.99	83
Doom Eternal	2020	59.99	88
Animal Crossing: New Horizons	2020	59.99	90
It Takes Two	2021	39.99	89





- 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 resulta fundamental para garantizar la integridad de los datos y evitar fallos durante el proceso de análisis. Un control adecuado en esta etapa inicial permite detectar inconsistencias, valores nulos o tipos de datos incorrectos antes de generar cálculos o visualizaciones. Esto asegura que los resultados obtenidos sean confiables y representativos de la realidad.

Las visualizaciones de datos desempeñan un papel clave en la interpretación de tendencias, correlaciones y patrones ocultos dentro del conjunto de información. A través de gráficos y diagramas adecuados, es posible comprender de manera más intuitiva el comportamiento de variables como popularidad, precios, valoraciones y géneros de videojuegos, facilitando la toma de decisiones o el planteamiento de nuevas hipótesis.

- **Dificultades encontradas:**

No guarda resultados

- Los análisis se muestran por consola o en pantalla, pero no se exportan a PDF, SCV o imagen. Si se cierra el programa, se pierde toda la información procesada.

  GITHUB:

  https://github.com/ZedEnjoyer/Proyecto-Analisis-de-Estadisticas-Videojuegos-AEV-.git