

# Aplicaciones del lenguaje Python para el análisis de tendencias en el sector de los videojuegos

Rafael Zúñiga Vindas  
Universidad Nacional de Costa Rica,  
Heredia, Costa Rica  
rafael.zuniga.vindas@est.una.ac.cr

Luis Fernández Serrano  
Universidad Nacional de Costa Rica,  
Heredia, Costa Rica  
luis.fernandez.serrano@est.una.ac.cr

Daniel Gómez Mena  
Universidad Nacional de Costa Rica,  
Heredia, Costa Rica  
daniel.gomez.mena@est.una.ac.cr

Emmanuel Pérez Sánchez  
Universidad Nacional de Costa Rica,  
Heredia, Costa Rica  
emmanuel.perez.sanchez@est.una.ac.cr

**Resumen**—*El presente documento tiene la intención de mostrar la aplicabilidad de las numerosas funciones que ofrece el lenguaje Python a través del uso de librerías como Pandas, Mathlib y Seaborn para poder manejar estructuras de datos, realizar cálculos estadísticos y a través de múltiples tipos de gráficos exponer los resultados obtenidos en el análisis orientado al mundo de los videojuegos realizado a partir del conjunto de datos recuperado de la plataforma Kaggle, con base en la información obtenida del sitio Backlogg.*

**Abstract**—*This document intends to show the applicability of the numerous functions offered by the Python language through the use of libraries such as Pandas, Matplotlib and Seaborn, to be able to manage data structures, perform statistical calculations and, through multiple types of graphics, present the results obtained in the analysis oriented to the world of video games, carried out from the data set recovered from the Kaggle platform, based on the information obtained from the Backlogg site.*

## I. INTRODUCCION

En esta era digital, el extenso flujo de datos ha cambiado radicalmente la manera en que se comprenden y evalúan diversos aspectos de la sociedad. En particular, en el mundo de los videojuegos, este se ha convertido en un campo de estudio crucial, donde datos como la preferencia,

comportamiento y tendencias de los consumidores se han vuelto de suma importancia.

Por lo tanto, el presente documento se enfoca en explorar a detalle este mundo, centrándose en variables cruciales como la calificación, fecha de lanzamiento, equipo desarrollador, reseñas, género y la participación de los usuarios, medida en términos de cuántos juegos están en sus listas de deseos, cuántos lo poseen, pero no lo juegan y cuántos están jugando actualmente. Utilizando poderosas funcionalidades del lenguaje de programación Python, aprovechando las bibliotecas Pandas, Mathlib y Seaborn, con el fin de revelar patrones, tendencias y relaciones ocultas en este vasto conjunto de información.

## II. MARCO TEORICO

A continuación, se presentan algunas referencias sobre el lenguaje Python que permiten sustentar su versatilidad como lenguaje multipropósito, capaz de suplir diferentes necesidades en una amplia gama de aplicaciones y disciplinas, en este caso, para el análisis de datos.

### A. Diseño de un sistema de administración de cursos online basado en Python

En mayo de 2023, T. Li presentó en New York un artículo en las actas de EBIMCS que se centra en la creación de un sistema de gestión de cursos en línea usando Python como base tecnológica. [1]

### B. *QuaPy un Framework de Python para Cuantificación.*

Fue publicado en 2021 por Alejandro Moreo, Andrea Esuli y Fabrizio Sebastiano. El framework QuaPy es una herramienta de código abierto diseñada para la cuantificación, que proporciona implementaciones de una serie de métodos de referencia y cuantificación avanzados, rutinas para la selección de modelos orientados a la cuantificación, medidas de evaluación ampliamente aceptadas y protocolos de evaluación sólidos que se utilizan habitualmente en el campo. [2]

### C. *Implementación de análisis de algoritmos de encriptación y desencriptación basados en Python.*

Publicado en el año 2023 por Kai Xiao, el artículo habla sobre lo difícil que es garantizar la seguridad y estabilidad de la transmisión de datos después del cifrado de clave utilizando el cifrado de contraseña unidireccional tradicional y el cifrado de contraseña bidireccional. Por lo que, propone un algoritmo de cifrado híbrido de RSA, DES y BASE64 desarrollado en lenguaje Python. [3]

### D. *Python para análisis de datos, aplicaciones científicas y técnicas.*

Publicado en 2019 a través de la Conferencia Internacional Amity sobre Inteligencia Artificial (AICAI), el artículo habla sobre las características del lenguaje de programación Python y luego discute las razones detrás del hecho de que Python es uno de los lenguajes de programación que más rápido crecen en el campo del análisis de datos. Esto se debe principalmente a su baja curva de aprendizaje, amplia cantidad de librerías disponibles y capacidad para ser un lenguaje multipropósito en la creación de aplicaciones, investigación y desarrollo científico. [4]

### E. *Una descripción general y comparación de bibliotecas Python gratuitas para minería de datos y análisis de Big Data.*

Publicado en 2019 este artículo habla sobre la comparación de distintas bibliotecas de análisis de Big Data y minería de datos, usando el lenguaje de

Python que últimamente debido a sus características se ha aumentado la cantidad de bibliotecas gratuitas para su uso. [5]

### F. *“Diseño de algoritmo de mejora de imágenes médicas basado en Python.*

Este artículo comparte la importancia hoy en día para el área de salud de la precisión necesaria en el reconocimiento de imágenes médicas actuales, habla sobre la opción de combinar varios métodos de mejora en la imagen con el software de Python con este tienen la ventaja de que el algoritmo es más realista a la hora de brindar el resultado de la imagen. [6]

### G. *Diseño de un sistema de operación y mantenimiento automático VMware vSphere basado en Python.*

En el año de 2018 se llevó a cabo la creación de un sistema de Operación y Mantenimiento Automático de VMware vSphere basado en Python, Este trabajo se presentó en el marco de la Conferencia Internacional de Sistemas Mecatrónicos Avanzados en Zhengzhou. En este se menciona como el uso de Python para dicho sistema resulta en un consumo más eficiente de recursos en comparación con Java. [7]

### H. *¿Serpientes en el paraíso?: Prácticas de codificación inseguras relacionadas con Python en Stack Overflow.*

En el año 2019 se presentó un análisis en “IEEE/ACM 16th International Conference on Mining Software Repositories”, celebrada en Montreal, Canadá. El estudio examina prácticas de codificación inseguras relacionadas con Python en Stack Overflow y todos los peligros que esto puede significar para los usuarios. [8]

### I. *Juegos de mesa en la clase de informática para mejorar el conocimiento de los estudiantes sobre el lenguaje de programación Python*

En el año 2018 el autor D. B. Jordaan redactó este artículo en el que relaciona los juegos de mesa con el aprendizaje del lenguaje de programación

Python ya que este recurso ayuda a fomentar el trabajo en equipo, la comunicación y al mismo tiempo les da confianza a los estudiantes. Este estudio dio como resultado una gran mejora en su aprendizaje de Python ya que era una forma entretenida de aprender. [9]

### J. *Análisis de rendimiento de distintas distribuciones de Python y TensorFlow para utilizar eficientemente el CPU en HPC Cluster.*

En este artículo, publicado por K. G. Gupta, S. K. Maity, A. Das y S. Wandhekar en el año 2021, se muestra un gran estudio acerca del uso de Python y TensorFlow, que es una librería de software para “machine learning” e inteligencia artificial para entrenar redes neuronales y se hace énfasis en la comparación de diferentes distribuciones para ver cuál es la óptima en HPC Cluster (computadoras de alto rendimiento para manejar gran cantidad de datos). Se logró aumentar 7 veces el rendimiento utilizando Intel Distribution y la aceleración en 15 veces utilizando CPUs de distintas arquitecturas. [10]

## III. DESCRIPCIÓN DE PÁGINA

La información se obtuvo de la plataforma Kaggle y Backloggd, ofrece una visión completa de 1,109 videojuegos. Este análisis explora diversas dimensiones, desde calificaciones y fechas de lanzamiento hasta desarrolladores, revisiones y géneros. Además, se profundiza en la participación de los usuarios, incluyendo cuántos juegos están en listas de deseos, cuántos se poseen, pero no se han jugado y cuántos se están jugando actualmente.

## IV. DESCRIPCIÓN DE PROCESAMIENTO DE LA INFORMACIÓN

Para la obtención de los datos para analizar y crear las gráficas se utilizó la librería Pandas para leer los datos del archivo CSV y Matplotlib para el diseño de los gráficos. A continuación, se presenta la sintaxis de código Python para dar solución a los casos escogidos.

A. *Obtener los 5 primeros juegos que fueron elaborados por el equipo desarrollador “Capcom”.*

Para la solución, se filtra la información según el nombre del equipo desarrollador en este caso “Capcom”, donde se recorre la lista con 1109 juegos buscando la similitud según la columna llamada Team.

```
def busca_retorna_por_equipo(df, equipo):
    filas = []
    for indice, i in enumerate(df['Team']):
        for j in i:
            if(j == equipo):
                filas.append(df.iloc[indice])
    resultado = pd.DataFrame(filas)
    resultado.reset_index(drop = True, inplace = True)
    return(resultado)
```

### Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist
Resident Evil Village	6/5/2021	[Capcom]	3.9	1800	venture, Shoo	9900	333	2800	2800
Resident Evil 2	25/1/2019	com Developm	4.3	2500	venture, Shoo	1500	585	3600	1900
Devil May Cry 5	8/3/2019	com Developm	4.3	1900	venture, Braw	1200	524	4100	2400
Resident Evil 4	11/1/2005	pcom Product	4.3	2300	[Shooter]	1400	360	300	1200
Resident Evil 7: Biohazard	24/1/2017	[Capcom]	3.9	1800	venture, Shoo	1300	326	3100	1330

**Tabla 1. Los 5 primeros juegos elaborados por Capcom.**

**Fuente: Backlogg.**

Se puede obtener que 4 de los juegos superan el rating de 4.0 y pertenecen al género de Shooter.

B. *Obtener los 5 primeros juegos que pertenecen al género “RPG”.*

Para la solución, se filtra la información según el nombre del género del juego en este caso “RPG”, donde se recorre la lista con 1109 juegos buscando la similitud según la información en la columna llamada Genres.

```
def busca_retorna_por_genero(df, genero):
    filas = []
    for indice, i in enumerate(df['Genres']):
        for j in i:
            if(j == genero):
                filas.append(df.iloc[indice])
    resultado = pd.DataFrame(filas)
    resultado.reset_index(drop = True, inplace = True)
    return(resultado)
```

### Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist
Elden Ring	25/2/2022	Entertainment	4.5	3900	[Adventure, RPG]	1700	3800	4600	4800
Hades	10/12/2019	pergiant Gam	4.3	2900	enture, Brawler, Indie,	2100	3200	6300	3600
of Zelda: Breath	3/3/2017	do EPD Produ	4.4	4300	[Adventure, RPG]	3000	2500	500	2600
Undertale	15/9/2015	[tobyfox, 8-4]	4.2	3500	, Indie, RPG, Turn Base	2800	679	4900	1800
Omori	25/12/2020	MOCAT, PLAYH	4.2	1600	, Indie, RPG, Turn Base	7200	1100	4500	3800

**Tabla 2 Los 5 primeros juegos de género RPG.**

**Fuente: Backlogg**

Se puede obtener que 5 de los juegos superan el rating de 4.2 lo que se considera alto.

C. *Retorna un df con la información de los títulos ganadores del premio "Game of the Year".*

En esta función se utiliza una lista con los títulos de los videojuegos ganadores del premio juego del año para así retornar todos sus datos respectivos.

```
def obtener_titulos_premiados(df):
    df = df.loc[df['Title'].isin(games_of_the_year)]
    df.reset_index(drop = True, inplace = True)
    return(df)
```

Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist
Elden Ring	25/2/2022	Entertainment	4.5	3900	Adventure, RPG	1700	3800	4600	4800
Legend of Zelda: Breath of the Wild	3/3/2017	EPD Product	4.4	4300	Adventure, RPG	3000	2500	500	2600
God of War	20/4/2018	Entertainment	4.2	2900	Action, Brawler	2100	1100	4800	2600
Seikiro: Shadows Die Twice	22/3/2019	Software, Activision	4.4	2300	Action, Brawler	1400	919	4800	3400
The Last of Us Part II	19/6/2020	Sony Interactive	4.1	2500	Action, Shooter	1300	472	2800	2500
The Witcher 3: Wild Hunt	19/5/2015	CD Projekt	4.3	200	Adventure, RPG	1700	1200	6100	1600
Overwatch	24/5/2016	Blizzard Entertainment	3.2	957	Shooter, Strategy	1900	436	411	214
It Takes Two	25/3/2021	Studios, Electronic Arts	4.2	1300	Action, Platform	8100	999	2500	2100

**Tabla 3 Los juegos ganadores del premio "Game of the Year".**

**Fuente: Backlogg**

Podemos observar que la mayoría de los juegos que obtuvieron el premio de juego del año tienen en común que pertenecen al género de videojuegos de aventura.

D. *Determina la matriz de correlaciones para las columnas con valores numéricos en el df.*

En esta función se hace énfasis en los aspectos numéricos de cada juego según sus datos, para poder establecer una matriz de correlación que sea más fácil de entender.

```
def obtener_matriz_correlacion(df):
    df = df.select_dtypes(include = 'number')
    return(df.corr())
def clasificar_matriz_correlacion(valor):
    if valor > 0.1:
        return "Positiva"
    elif valor < 0:
        return "Negativa"
    else:
        return "Nula"
def limpiar_diagonal(df):
    filas, columnas = df.shape
    for i in range(min(filas, columnas)):
        df.iat[i, i] = "-"
    return(df)
```

Resultado:

La salida resultante de esta función se muestra en el apartado de análisis de datos en el gráfico 1

Podemos deducir de esta tabla de correlación que de acuerdo con sus valores numéricos la mayoría de los datos muestran valores positivos entre sí, esto dando a entender que tuvieron un buen desempeño en el juego.

E. *Retorna los equipos con más de 2000 jugadores que poseen el juego, pero no lo han jugado*

Esta función nos brinda información sobre los juegos que cumplen las condiciones de que más de 2000 jugadores lo compraran, pero no lo han jugado después de la compra en base a una compañía desarrolladora de videojuegos pertenezcan dichos títulos.

```
def juegos_inactivos_por_equipo(df, equipo):
    df = busca_retorna_por_equipo(df, equipo)
    df = df.loc[(df['Backlogs'] >= 2000)]
    df.reset_index(drop = True, inplace = True)
    return(df)
```

Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist	Diferencia
God of War	20/4/2018	Entertainment	4.2	2900	Action, Brawler	2100	1100	4800	2600	2700
Marvel's Spider-Man	7/9/2018	Sony Interactive	4.1	2500	Action, Brawler	2100	577	2900	2200	800
The Last of Us Part II	19/6/2020	Sony Interactive	4.1	2500	Action, Shooter	1300	472	2800	2500	1500
Death Stranding	8/11/2019	Entertainment	3.9	1600	Action, RPG, SH	9400	934	4800	2400	-4600
Marvel's Spider-Man: Miles Morales	12/11/2020	Sony Interactive	3.9	1600	Action, Brawler	9900	372	2800	2800	-7100
Shadow of the Colossus	6/2/2018	Entertainment	4.1	1100	Action, Platform	7800	242	3200	1600	-4600
Horizon Forbidden West	24/2/2022	Sony Interactive	3.8	850	Adventure, RPG	2700	561	2100	2300	-600
Detroit: Become Human	25/5/2018	Sony Interactive	3.5	100	Action, Puzzle	1100	216	2200	1300	1100
Demon's Souls	12/11/2020	Sony Interactive	3.9	880	Adventure, RPG	4100	403	2400	2100	-1700
Uncharted 4: A Thief's End	10/9/2016	Sony Interactive	4.1	1800	Action, Shooter	1400	177	2700	1200	1500
Ghost of Tsushima	17/7/2020	Productions, Sony	4.1	1400	Action, Brawler	8500	688	3500	3100	-9000
The Last Guardian	6/12/2016	Sony Interactive	3.7	658	Action, Platform	3200	83	2300	1300	-900

**Tabla 4 Los equipos con más de 2000 jugadores que no poseen el juego, pero no lo han jugado.**

**Fuente: Backlogg**

En conclusión, podemos observar que de acuerdo con la compañía seleccionada se encuentran varios juegos que actualmente los usuarios a pesar de haber comprado nunca jugaron y están de una forma inactiva esto con el fin de analizar el rendimiento de los juegos en términos de jugadores.

F. *Obtiene los juegos que son activamente jugados por al menos 1500 jugadores.*

Esta función nos brinda información sobre los juegos que cumplen las condiciones de que más de 1500 jugadores lo jueguen de forma activa en base a genero de videojuegos.

```
def juegos_activos_por_genero_del_juego(df, genero):
    df = busca_retorna_por_genero(df, genero)
    df = df.loc[(df['Playing'] >= 1500)]
    return(df)
```

## Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist
Elden Ring	25/2/2022	FromSoftware	4.5	3900	[Adventure, RPG]	1700	3800	4600	4800
Hades	10/12/2019	[Supergiant Games]	4.3	2900	[Adventure, RPG]	2100	3200	6300	3600
Legend of Zelda: Breath of the Wild	3/3/2017	Nintendo EPD Production Group	4.4	4300	[Adventure, RPG]	3000	2500	500	2600
Persona 5 Royal	31/10/2019	[Atlus USA, Atlus]	4.4	2700	[RPG, Turn Based]	1200	2300	5100	300
Yakuza 0	12/3/2015	Ryū Ga Gotoku Studios, Sega	4.4	2700	[Action, Brawler, RPG, Shoot]	1500	1800	6400	200
Red Dead Redemption 2	26/10/2018	Rockstar Games	4.4	2900	[Adventure, RPG, Shoot]	1900	1700	5500	2900
Pokemon Legends: Arceus	28/1/2022	[Nintendo, Game Freak]	3.7	1600	[RPG, Turn Based]	9100	1600	2500	2100
Cyberpunk 2077	9/12/2020	[CD Projekt RED]	3.3	1500	[Adventure, RPG, Shoot]	1300	1500	4700	2900
Persona 4 Golden	15/6/2012	[NIS America, Atlus]	4.2	1900	[RPG, Simulator, Visual Novel]	1300	1500	5200	2200
Genshin Impact	28/9/2020	[miHoYo, HoYoverse]	2.6	653	[Adventure, RPG]	1400	2700	1300	280

**Tabla 5 Los juegos que son activamente jugados por al menos 1500 jugadores.**

*Fuente: Backlogg'd*

En conclusión, podemos observar que de acuerdo con el género de videojuegos seleccionado se encuentran varios juegos que actualmente tienen una buena cantidad de jugadores activos, esto con el fin de analizar el rendimiento de los juegos en términos de jugadores.

### G. Mostrar los mejores juegos del género RPG de cada año desde el 2000.

Para realizar este procedimiento primero fue necesario tomar la fecha de lanzamiento de los videojuegos y hacer el cambio a formato fecha. Luego de eso se busca el juego que tuvo la mayor cantidad de calificaciones y se toman únicamente los juegos posteriores al año 2000. Una vez realizado este procedimiento se ordenan los juegos por fecha de lanzamiento ascendentemente.

```
def juegos_rating_superior(df, genero):
    df = busca_retorna_por_genero(df, genero)
    df = df.loc[(df['Rating'] >= 4.0)]
    df['Release Date'] = pd.to_datetime(df['Release Date'], format='%d/%m/%Y')
    df['Year'] = df['Release Date'].dt.year
    grupos = df.groupby('Year')
    filas_seleccionadas = []
    for year, grupo in grupos:
        fila_max_reviews = grupo[grupo['Reviews'] == grupo['Reviews'].max()]
        filas_seleccionadas.append(fila_max_reviews)
    df = pd.concat(filas_seleccionadas)
    fecha_actual = pd.Timestamp.now()
    fecha_limite = fecha_actual - pd.DateOffset(years=23)
    df = df[df['Release Date'] >= fecha_limite]
    df = df.sort_values(by='Release Date', ascending=True)
    df.reset_index(drop = True, inplace = True)
    return df
```

## Resultado:

Title	Release Date	Team	Rating	Reviews	Genres	Plays	Playing	Backlogs	Wishlist	Year
Final Fantasy X	19/7/2001	Square Electronic Arts, Square	4.1	1500	[Adventure, RPG]	8500	330	2500	990	2001
Elder Scrolls III: Morrowind	1/5/2002	Bethesda Game Studios, ZeniMax	4	549	[Adventure, RPG]	3700	150	1700	487	2002
Megami Tensei: Nocturne	20/2/2003	[Atlus, Ghostlight Ltd.]	4.2	1100	[RPG]	4100	358	2500	1300	2003
Mario: The Thousand-Year Door	22/7/2004	Nintendo Intelligent Systems Co.	4.4	1300	[Adventure, RPG, Turn Based]	6700	187	2900	1500	2004
Kingdom Hearts II	22/12/2005	Square Enix, Square Enix	4.3	1200	[Adventure, RPG]	9600	137	1600	652	2005
Mother 3	20/4/2006	[HAL Laboratory, Nintendo]	4.5	1600	[G, Turn Based Strategy]	5700	287	3600	1900	2006
BioShock	21/8/2007	[2K Games, 2K Boston]	4	1800	[Adventure, RPG, Shoot]	1700	339	400	1200	2007
Pokemon Platinum	13/9/2008	[Nintendo, Game Freak]	4.1	1400	[RPG, Turn Based]	1400	254	1300	461	2008
Demon's Souls	5/2/2009	[FromSoftware, Atlus]	4.1	1200	[Adventure, RPG]	5500	155	1700	1200	2009
Red Dead Redemption	18/5/2010	Rockstar Games, Rockstar North	4.2	1500	[Adventure, RPG, Shoot]	1300	218	2400	1600	2010
Dark Souls	22/9/2011	FromSoftware, Bandai Namco Entertainment	4.3	2100	[Adventure, RPG]	1400	447	2500	1400	2011
Persona 4 Golden	15/6/2012	[NIS America, Atlus]	4.2	1900	[RPG, Simulator, Visual Novel]	1300	1500	5200	2200	2012
Legend of Zelda: A Link Between Worlds	22/11/2013	Nintendo EAD Software Development	4.1	1300	[Adventure, Puzzle, RPG]	1000	168	2400	1100	2013
Lisa	15/12/2014	[DingoLing]	4.3	761	[Adventure, Indie, RPG]	2900	143	2100	1300	2014
Undertale	15/9/2015	[tobyfox, 84]	4.2	3500	[Adventure, Indie, RPG, Turn Based]	2800	679	4900	1800	2015
Dark Souls III	24/3/2016	FromSoftware, Bandai Namco Entertainment	4.2	2400	[Adventure, RPG]	1900	851	4300	2200	2016
Legend of Zelda: Breath of the Wild	3/3/2017	Nintendo EPD Production Group	4.4	4300	[Adventure, RPG]	3000	2500	500	2600	2017
God of War	20/4/2018	Sony Interactive Entertainment, Santa Monica Studio	4.2	2900	[Adventure, Brawler, RPG]	2100	1100	4800	2600	2018
Red Dead Redemption 2	26/10/2018	Rockstar Games	4.4	2900	[Adventure, RPG, Shoot]	1900	1700	5500	2900	2018
Hades	10/12/2019	[Supergiant Games]	4.3	2900	[Adventure, Brawler, Indie]	2100	3200	6300	3600	2019
Yakuza: Like a Dragon	16/12/2020	Sega, Ryū Ga Gotoku Studios	4.3	1600	[RPG, Turn Based]	600	847	4800	2800	2020
Omori	25/12/2020	[OMOCAT, PLAYISM]	4.2	1600	[Adventure, Indie, RPG, Turn Based]	7200	1100	4500	3800	2020
Replicant ver.1.224744875	22/4/2021	[Toylogic, Square Enix]	4.2	1500	[Adventure, RPG]	6700	880	4100	3700	2021
Elden Ring	25/2/2022	FromSoftware, Bandai Namco Entertainment	4.5	3900	[Adventure, RPG]	1700	3800	4600	4800	2022
Legend of Zelda: Tears of the Kingdom	12/5/2023	Nintendo EPD Production Group	4.5	581	[Adventure, RPG]	72	6	1600	5400	2023

**Tabla 6 Los mejores juegos del género RPG de cada año desde el 2000-2023.**

*Fuente: Backlogg'd*

Como resultado se obtienen todos los juegos RPG que han tenido mayor calificación por los usuarios desde el año 2000 y de hecho, uno de los juegos con la calificación más alta es Elden Ring, el cual fue galardonado como juego del año en el 2022.

### H. Encontrar el juego con el mayor rating para cada equipo desarrollador.

En este caso, como algunos juegos cuentan con una lista de equipo desarrolladores, se utiliza explode() para separar cada uno de los equipos y se recorren todos los datos para encontrar el juego con mayor calificación. Luego se agrupan los datos por equipo desarrollador con groupby('Team').

```
def juego_mayor_rating_por_equipo(df):
    df = df[['Title', 'Team', 'Rating', 'Reviews']]
    juegos_con_mayor_rating = []
    equipos = df['Team'].explode().unique()
    for e in equipos:
        juegos_de_equipo = df[df['Team'] == e].apply(lambda x:
            juego_mayor_rating =
            juegos_de_equipo[juegos_de_equipo['Rating'] ==
            juegos_de_equipo['Rating'].max()]
            juego_mayor_rating.loc[juego_mayor_rating.index, 'Team']
            = e
            juegos_con_mayor_rating.append(juego_mayor_rating)
    df = pd.concat(juegos_con_mayor_rating)
    grupos = df.groupby('Team')
    filas_seleccionadas = []
    for team, grupo in grupos:
        fila_equipo = grupo[grupo['Reviews'] == grupo['Reviews'].max()]
        filas_seleccionadas.append(fila_equipo)
    df = pd.concat(filas_seleccionadas)
    df.reset_index(drop = True, inplace = True)
    return df
```



Resultado:

Title	Team	Rating	Reviews
Vagrant Story	re Product Developr	4	503
nineko: When They Cry Ch	07th Expansion	4.6	324
n To Die: Death Game By	OUP Games	4.1	221
Children of Morta	11 bit studios	3.4	153
Wanted: Dead	110 Industries	2.6	81
...	...	...	...
Signalis	rose-engine	4.2	820
a: Four Rhythms Across th	sprite	3.9	32
Pathologic 2	tinyBuild	4.3	513
Deltarune: Chapter 2	tobyfox	4.3	737
Tales of Zestiria	tri-Crescendo	2.7	273

**Tabla 7 Los juegos con mayor rating de cada equipo desarrollador.**  
*Fuente: Backlogg*

Se obtiene un nuevo dataframe con una lista de todos los equipos y su juego mejor calificado.

### *I. Busca al juego con el rating más alto de cada año.*

Con esta función se busca obtener todos los juegos mejor calificados de cada año. En primer lugar se toma la columna de la fecha de lanzamiento y se cambia a formato de fecha para después tomar la fecha actual con `Timestamp.now()` y con `DateOffset(years=año)` se envían por parámetro la cantidad de años que se quieren analizar. También se hace uso de `idxmax()` para encontrar al juego con la calificación más alta.

```
def busca_retorna_por_rating(df, año):
    df['Release Date'] = pd.to_datetime(df['Release Date'],
    format='%d/%m/%Y')
    hoy = pd.Timestamp.now()
    df_filtrado = df[df['Release Date'] >= hoy -
    pd.DateOffset(years=año)].copy()
    df_filtrado['Year'] = df_filtrado['Release
    Date'].dt.year
    resultados =
    df_filtrado.loc[df_filtrado.groupby('Year')['Rating'].idx
    max()]
    resultados.reset_index(drop = True, inplace = True)
    return resultados[['Year', 'Title', 'Rating']]
busca_retorna_por_rating(datos_games, 5)
```

Resultado:

Year	Title	Rating
2013	Final Fantasy X HD Remaster	4.3
2014	Resident Evil 4: Ultimate HD Edition	4.4
2015	Bloodborne: The Old Hunters	4.6
2016	Tokyo Necro	4.6
2017	The Great Ace Attorney 2: Resolve	4.6
2018	Hollow Knight: Voidheart Edition	4.5
2019	Outer Wilds	4.6
2020	Disco Elysium: The Final Cut	4.6
2021	Final Fantasy XIV: Endwalker	4.6
2022	Dwarf Fortress	4.6
2023	Hitman World of Assassination	4.6

**Tabla 8 Los juegos con mayor rating desde el 2013.**  
*Fuente: Backlogg*

Con esta implementación, se pueden observar los juegos con la calificación más alta y se puede concluir que desde hace 5 años el juego mejor valorado ha tenido una nota de 4.6 y no hay juegos con una nota mayor a esta.

### *J. Encontrar los juegos en los cuales los jugadores inactivos son mayores a los jugadores totales.*

En primer lugar, se crea una columna llamada “Diferencia” la cual consiste en la resta entre los jugadores inactivos y los jugadores totales. Luego de eso se toman los juegos los cuales tienen una mayor cantidad de jugadores inactivos. Se puede ajustar el parámetro de la cantidad de juegos que se desean recibir.

```
def comparacion_backlogs_plays(df, cantidadJuegos):
    df['Diferencia'] = df['Backlogs'] - df['Plays']
    juegos_con_diferencia_positiva = df[df['Diferencia']
    > 0]
    if not juegos_con_diferencia_positiva.empty:
        print("Juegos que se han comprado y no jugado con
        comparación a la cantidad de veces jugado:")
        return(juegos_con_diferencia_positiva[['Title',
        'Backlogs', 'Plays', 'Diferencia']]).head(cantidadJuegos)
    else:
        return("No hay juegos con más Backlogs que Plays
        en el archivo.")
```

## Resultado:

Title	Backlogs	Plays	Genres	Rating	Diferencia
Elden Ring	4600	1700	[Adventure, RPG]	4.5	2900
Hades	6300	2100	enture, Brawler, Indie,	4.3	4200
Undertale	4900	2800	, Indie, RPG, Turn Base	4.2	2100
Hollow Knight	8300	2100	dventure, Indie, Platfor	4.4	6200
NieR: Automata	6200	1800	[Brawler, RPG]	4.3	4400
Persona 5 Royal	5100	1200	ure, RPG, Turn Based S	4.4	3900
God of War	4800	2100	adventure, Brawler, RP	4.2	2700
Portal 2	3900	2900	ure, Platform, Puzzle, S	4.4	1000
Bloodborne	5600	1700	[Adventure, RPG]	4.5	3900
Celeste	5900	2000	dventure, Indie, Platfor	4.2	3900
Yakuza 0	6400	1500	ture, Brawler, RPG, Sim	4.4	4900
Dead Redemption	5500	1900	adventure, RPG, Shoot	4.4	3600
Super Mario Odyssey	2900	2500	[Adventure, Platform]	4.2	400
Hi-Fi Rush	1500	300	ture, Brawler, Music, Pl	4.3	1200
Gear Rising: Reveng	4200	1400	ure, Brawler, Shooter, S	4.1	2800
Grand Theft Auto V	3200	3000	[Adventure, Shooter]	3.8	200
Cyberpunk 2077	4700	1300	adventure, RPG, Shoot	3.3	3400
Marvel's Spider-Man	2900	2100	[Adventure, Brawler]	4.1	800
Dark Souls III	4300	1900	[Adventure, RPG]	4.2	2400
Super Mario 64	2500	2100	[Adventure, Platform]	4.1	400

**Tabla 9** Los juegos en los cuales los jugadores inactivos son mayores a los jugadores totales.

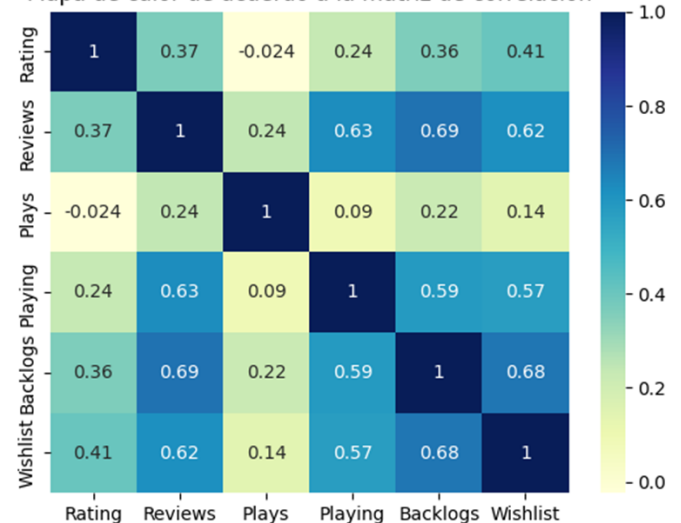
**Fuente:** Backlogg

Se puede observar que el juego con la mayor cantidad de jugadores inactivos es “Hollow Knight”. La razón de esto puede ser la gran dificultad que tiene el videojuego que hace que los jugadores dejen de jugarlo.

## V. ANÁLISIS DE DATOS

Tomando en cuenta el método del punto D, se utiliza la función `sns.heatmap`, de esta manera generar un mapa de calor según una matriz de correlación tomando en cuenta sólo los datos numéricos del conjunto de datos, en este caso serían solo las columnas Rating, Reviews, Plays, Playing, Backlogs y Wishlist, con el fin de saber si la clasificación es positiva, negativa o nula, como se logra ver en el Gráfico 1.

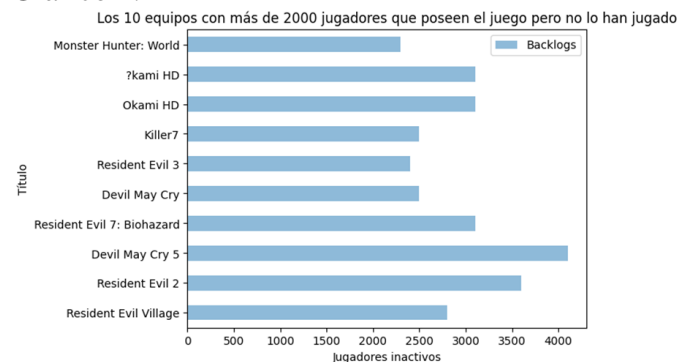
Mapa de calor de acuerdo a la matriz de correlación



**Gráfico 1** Mapa de calor según la matriz de correlación.

**Fuente:** Backlogg

Inicialmente, según la descripción del procesamiento de la información, de acuerdo a lo explicado en el método del punto E que nos ayuda a ver los juegos con más de 2000 jugadores que compraron el juego en su momento pero no han jugado después de la compra, se procede a utilizar la función `plot.barh()`, de esta manera generar un gráfico horizontal donde se visualiza la información del resultado de dicho método, agregando identificadores como `plt.title()` para modificar el título del gráfico, así como la etiqueta `plt.xlabel()` y `plt.ylabel()` para modificar los nombres que se encuentran en la parte inferior y en la parte izquierda del gráfico, como se logra ver en el Gráfico 2.



**Gráfico 2** Los equipos con más de 2000 jugadores que poseen el juego, pero no lo han jugado

**Fuente:** Backlogg

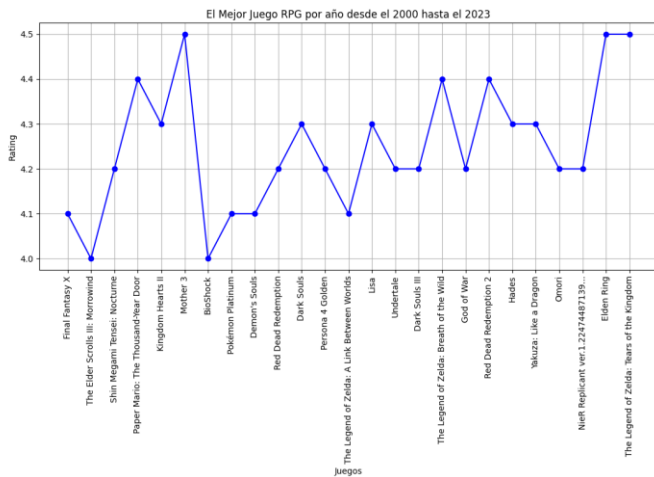
Inicialmente, según la Descripción del procesamiento de la Información, de acuerdo a lo explicado en el método del punto F, se procede a

utilizar la función `plot.barh()`, de esta manera generar un gráfico horizontal donde se visualiza la información del resultado de dicho método que nos ayuda a ver los juegos con más de 1500 jugadores activos actualmente, agregando identificadores como `plt.title()` para modificar el título del gráfico, así como la etiqueta `plt.xlabel()` y `plt.ylabel()` para modificar los nombres que se encuentran en la parte inferior y en la parte izquierda del gráfico, como se logra ver en el Gráfico 3.



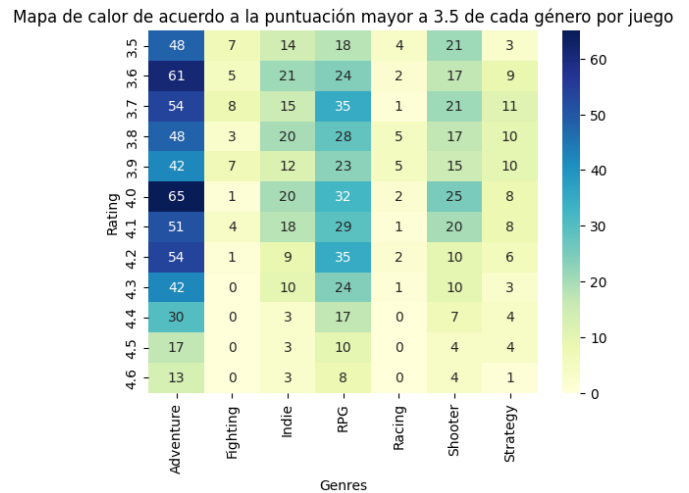
**Gráfico 3 Juegos con más de 1500 jugadores activos por género del juego**  
**Fuente: Backlogg**

Con respecto a los resultados que arroja la función número 7, se puede observar cómo únicamente en 3 ocasiones se ha llegado a la calificación más alta y desde la primera vez que sucedió, en el 2006, pasaron 16 años para que eso volviera a suceder, como se logra ver en el Gráfico 4.



**Gráfico 4 Los mejores juegos RPG por año desde el 2000 hasta el 2023**  
**Fuente: Backlogg**

Este mapa de calor muestra los datos comparando el “Rating” con los géneros de videojuegos que se agregaron en la lista. Se pueden observar la cantidad de juegos que poseen cierta calificación en cada uno de los géneros. Para lograr esto se busca que el rating sea mayor a 3.5 llegue hasta el máximo de calificación que tiene un juego, en este caso es 4.6, como se logra ver en el Gráfico 5.



**Gráfico 5 Mapa de calor de acuerdo con la puntuación mayor a 3.5 de cada género por juego**  
**Fuente: Backlogg**

## VI. CONCLUSIONES Y RECOMENDACIONES

El uso de Python como lenguaje de programación y el entorno de desarrollo Google Colab fueron una configuración muy eficiente a la hora de desarrollar el análisis. El lenguaje es una herramienta muy poderosa y sencilla de usar. El ambiente de desarrollo es ligero, funcional y permite impulsar el trabajo colaborativo al permitir programar de manera conjunta. La integración con Google Drive facilitó el almacenamiento y el envío de los archivos relacionados con el proyecto, brindando un ambiente de trabajo fluido. Esta combinación no solo optimizó la eficiencia durante el desarrollo, sino que también mejoró significativamente la gestión y colaboración del equipo de trabajo.

Los resultados obtenidos con base en las funciones y gráficos permitieron conocer los géneros de videojuegos más populares, el comportamiento que presentaban los jugadores con los juegos que adquirían o deseaban y la adaptabilidad de los equipos desarrolladores para entregar productos alineados a las tendencias que



existían en la industria del videojuego en un momento determinado.

El conjunto de datos cuenta con un importante volumen de datos, en su mayoría numéricos, relacionados principalmente con la manera en la que los usuarios de la página Backloggd interactuaban con sus videojuegos favoritos. El resto de los datos involucraban todos los detalles relacionados con el desarrollo y publicación de cada videojuego, entre ellos: los equipos desarrolladores involucrados, fecha de salida y género.

Las funciones permitieron encontrar relaciones importantes entre variables numéricas, la más importante de ellas fue que la buena calificación de un videojuego influye en el deseo de los usuarios por adquirirlo, y este deseo se manifiesta aún más si se da el caso donde la buena aceptación del producto se evidencia por parte de una gran cantidad de usuarios. Se evidencia cómo ciertos videojuegos conocidos por su alto nivel de complejidad han sido adquiridos, pero no se suelen jugar activamente, se asume que los jugadores están más dispuestos a disfrutar de experiencias menos demandantes, porque estos juegos como el conocido “Hollow Knight”, que posee buen rating entra dentro de este grupo.

Por otro lado, también mostraron cómo los videojuegos más galardonados en la ceremonia de premiación anual más importante en el área pertenecían a los géneros de rol y aventura, y en promedio, su calificación de parte de los jugadores era bastante alta. No obstante, los resultados también muestran que no necesariamente el videojuego mejor calificado de ese año fue el ganador, pero en su mayoría sí que llegaron a ser nominados o hasta premiados en otras subcategorías.

Uno de los objetivos de los gráficos que surgieron a raíz de los resultados de las funciones fue ilustrar el volumen de jugadores que está jugando activamente a determinado videojuego. Al observar cada gráfico se evidencia cómo juegos gratuitos como “Genshin Impact” o de alta calificación como “Elden Ring” entran en la categoría de juegos activos, y por otra parte, aquellos exclusivos para una plataforma en específico, por ejemplo, el ecosistema PlayStation es donde emergen los juegos inactivos.

El último gráfico, que se trata de un mapa de calor, permitió conocer la cantidad de juegos con

mejor calificación dentro de los géneros más influyentes, donde reaparece el género de aventura, una vez más cobrando protagonismo, seguido por los juegos de rol en segundo lugar. Queda claro que estos son los géneros que más destacan sobre el resto a la hora de generar productos en su mayoría bien calificados y recibidos tanto por jugadores como por la crítica.

Durante todo el análisis se llegó a soñar con la idea de poder contar con una o varias columnas relacionadas con los ingresos por ventas que obtuvo cada videojuego para poder encontrar correlaciones más claras y lograr incrementar los índices de precisión de todos los resultados obtenidos. El hecho de poder condensar, junto con el conjunto de datos existente, información tan valiosa como los ingresos por ventas de un videojuego, cantidad de preventas y hasta el gasto publicitario serviría no solo para cubrir lo mencionado con relación a la precisión obtenida en las conclusiones, si no, que incluso abre la posibilidad de poder expandir mucho más el alcance del análisis.

Mejorar los datos de las tablas de datos puede resultar tediosa, ya que implica tratar con diversos desafíos como valores nulos, cambiar nombres de variables, ajustar formatos de fechas y modificar el tipo de separador decimal, entre otros aspectos. Una vez la tabla se encuentra depurada (si es que lo necesita) es fundamental invertir el tiempo necesario en comprender el significado de cada variable para lograr interpretaciones precisas, aún más si se da el caso donde existe ausencia de una documentación adecuada en la fuente de información. Debido a la existencia de tantas fuentes, es importante informarse al respecto de herramientas especializadas para la búsqueda de conjuntos de datos como la base de datos SIDUNA, la cual sirvió para poder encontrar los datos para realizar el análisis.

La convergencia entre los objetivos de aprendizaje relacionados con el curso y la selección de un tema familiar para los miembros del grupo permitió visualizar la industria desde una nueva perspectiva, ya que ahora se debía tomar el tiempo para sacar las conclusiones basadas en datos, convertir una de las formas de entretenimiento más convencionales en objeto de estudio y comprender parte del criterio que utilizan los diligentes de la industria para tomar decisiones sobre estos productos y servicios.

## VII. REFERENCIAS

- [1] T. Li, "Design of online course management system based on Python," in Proceedings of the 2022 5th International Conference on E-Business, Information Management and Computer Science, in EBIMCS '22. New York, NY, USA: Association for Computing Machinery, May 2023, pp. 207–211. doi: 10.1145/3584748.3584783
- [2] A. Moreo, A. Esuli, and F. Sebastiani, "QuaPy: A Python-Based Framework for Quantification," in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event Queensland Australia: ACM, Oct. 2021, pp. 4534–4543. doi: 10.1145/3459637.3482015.
- [3] K. Xiao, "Implementation Analysis of Encryption and Decryption Algorithm Based on Python Language," in 2022 7th International Conference on Systems, Control and Communications, Chongqing China: ACM, Oct. 2022, pp. 1–5. doi: 10.1145/3575828.3575829.
- [4] A. Nagpal and G. Gabrani, "Python for Data Analytics, Scientific and Technical Applications," in 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates: IEEE, 2019, pp. 140–145. doi: 10.1109/AICAI.2019.8701341.
- [5] I. Stancin and A. Jovic, "An overview and comparison of free Python libraries for data mining and big data analysis," in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia: IEEE, 2019, pp. 977–982. doi: 10.23919/MIPRO.2019.8757088.
- [6] H. Zhou and S. Wu, "Design of medical image enhancement algorithm based on Python," in 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China: IEEE, Jan. 2021, pp. 482–485. doi: 10.1109/ICPECA51329.2021.9362581.
- [7] F. Liu and Z. Yang, "Design of VMware vSphere Automatic Operation and Maintenance System Based on Python," in 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), Zhengzhou: IEEE, 2018, pp. 283–286. doi: 10.1109/ICAMechS.2018.8506789.
- [8] A. Rahman, E. Farhana, and N. Imtiaz, "Snakes in Paradise?: Insecure Python-Related Coding Practices in Stack Overflow," in 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), Montreal, QC, Canada: IEEE, 2019, pp. 200–204. doi: 10.1109/MSR.2019.00040.
- [9] D. B. Jordaan, "Board Games in the Computer Science Class to Improve Students' Knowledge of the Python Programming Language," in 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Plaine Magnien: IEEE, 2018, pp. 1–5. doi: 10.1109/ICONIC.2018.8601207..
- [10] K. G. Gupta, S. K. Maity, A. Das, and S. Wandhekar, "Performance analysis of different distribution of Python and TensorFlow to efficiently utilize CPU on HPC Cluster," in 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa: IEEE, Dec. 2021, pp. 1–6. doi: 10.1109/ICECET52533.2021.9698764.