

Diseño de una arquitectura de Data Lake para integración de múltiples fuentes de datos y análisis de casos de uso mediante dashboards en Power BI

*Luna. S, Pazto. I, Mata. J y Vela. D
ESFOT, Escuela Politécnica Nacional
Quito, Ecuador*

Resumen— El creciente volumen de datos generados diariamente representa una oportunidad para extraer valor mediante técnicas analíticas. Este proyecto plantea diseñar una arquitectura de Data Lake para consolidar información desde al menos 12 fuentes, que incluyen sitios web, scrapers y repositorios públicos. La solución hará uso de bases de datos SQL y NoSQL para el almacenamiento, así como de Microsoft Azure para la integración de datos, que se conectará a Power BI para el análisis visual. Se desarrollarán 5 casos de estudio centrados en temáticas como juegos online, eventos mundiales y ciencia, contruidos dashboards con métricas e indicadores clave. Se incorporarán visualizaciones geoespaciales. De esta manera, se implementa una arquitectura Big Data escalable para explotar el valor de una diversa cantidad de fuentes de información.

I.INTRODUCCIÓN

LA gran cantidad de datos generados a diario a través de múltiples fuentes como archivos estáticos, web scraping, entre otros, representa una oportunidad para extraer información

valiosa mediante técnicas de análisis y visualización.

En este contexto, la implementación de una arquitectura de Data Lake resulta apropiada para la recolección e integración de grandes volúmenes de datos no estructurados provenientes de extensas fuentes.

El presente proyecto tiene como objetivo diseñar una arquitectura de este tipo que nos permita almacenar datos de al menos doce fuentes que pueden variar, incluyendo orígenes de repositorios públicos, web scraping, archivos csv y json, etc. La arquitectura propuesta tendrá, como mínimo, el uso de tres bases de datos SQL y tres NoSQL para almacenar la información, así como un concentrador de datos, en este proyecto se hace uso de Microsoft Azure, que se conectará a Power BI para el análisis de la data y su visualización gráfica.

Se propone desarrollar cinco casos de estudio focalizados en las temáticas: Juegos en línea por países, películas, eventos y noticias mundiales, conciertos a nivel mundial y ciencia en Ecuador. De cada caso de estudio se construye un dashboard que exponga las principales métricas e indicadores. De igual forma se incorporan visualizaciones con capacidades de geolocalización.

II.GLOSARIO DE TÉRMINOS Y HERRAMIENTAS

Data Lake: Repositorio centralizado para almacenar grandes volúmenes de datos en su formato nativo. Permite cargar datos estructurados, no estructurados y semiestructurados desde varias fuentes.

Azure Data Lake Gen 2: Servicio de almacenamiento de gran escala en la nube de Microsoft Azure.

MySQL: Sistema gestor de bases de datos relacional de código abierto.

SQL Server: Sistema gestor de bases de datos relacional desarrollado por Microsoft.

PostgreSQL: Sistema gestor de bases de datos relacional de código abierto.

MongoDB Compass: Interfaz gráfica de usuario para trabajar con bases de datos MongoDB.

CSV: Formato de archivo separado por comas, usado para almacenar datos tabulares.

JSON: Formato de intercambio de datos ligero basado en el lenguaje de programación web JavaScript.

Python: Lenguaje de programación de alto nivel y multiparadigma enfocado a la programación funcional, automatización de tareas y análisis de datos.

Datasets: Conjunto de datos, ordenado bajo un sistema de almacenamiento que otorga los lineamientos principales de búsqueda o directorio de la información que se quiere trabajar.

III.OBJETIVOS

Objetivo general

Diseñar una arquitectura de Data Lake para la integración de fuentes de datos, su análisis y visualización de indicadores y métricas mediante dashboards en Power BI.

Objetivos específicos

Seleccionar al menos 12 fuentes de datos a integrar en la solución respecto a los temas propuestos.

Realizar la limpieza de datos respectiva a cada fuente.

Definir cinco casos de estudio teniendo en cuenta los temas seleccionados y la coherencia de los datasets.

Concentrar todos los datos en un repositorio centralizado.

Identificar los índices y métricas a evaluar para cada caso de estudio conforme a los datos que le corresponden.

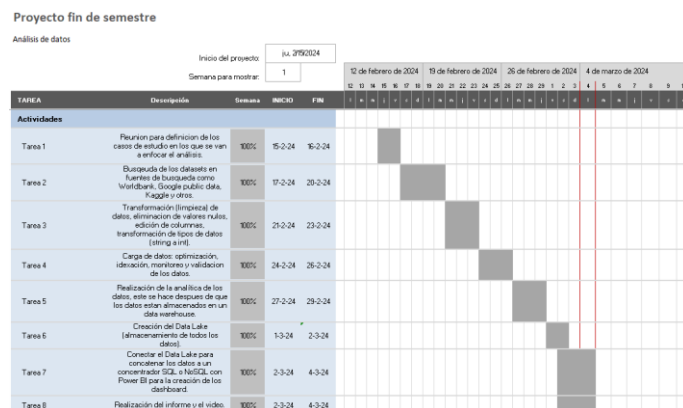
Generar el dashboard correspondiente a cada caso de estudio haciendo uso de Power BI

IV.DESCRIPCIÓN DEL EQUIPO DE TRABAJO Y ACTIVIDADES REALIZADAS POR CADA UNO

El equipo de trabajo estará compuesto por estudiantes de la carrera de Desarrollo de software, de tercer semestre. Las actividades realizadas por cada miembro del equipo incluirán la identificación de fuentes de datos, el diseño de la arquitectura de Data Lake, la implementación del concentrador de datos en MongoDB, SQL Server y MySQL, la extracción de datos, el análisis de la información, visualización de los dashboard en Power BI y la elaboración del informe final.

V.CRONOGRAMA

En la siguiente Ilustración 1 cronograma, se muestra la planificación del proyecto en base a un cronograma Gantt.



VI.RECURSOS Y HERRAMIENTAS UTILIZADAS

Fuentes de los datos:

- Repositorios públicos, estos brindan la ventaja de acceso a datos abiertos, estos se los puede usar sin restricciones, además de que cuenta con una diversidad de datos y son confiables y transparentes.
- Web scraping, la información que proporciona es en tiempo real, brinda asimismo personalizarlos.
- Archivos CSV, su manipulación los vuelve ampliamente compatible con diferentes aplicaciones y programas, lo que permite que en su análisis se identifiquen tendencias y patrones en los datos.

- Archivos JSON, al tener una estructura estandarizada permite organizar datos de manera eficiente y legible, igualmente estos datos pueden ser compartidos fácilmente y utilizados en diferentes aplicaciones y plataformas.

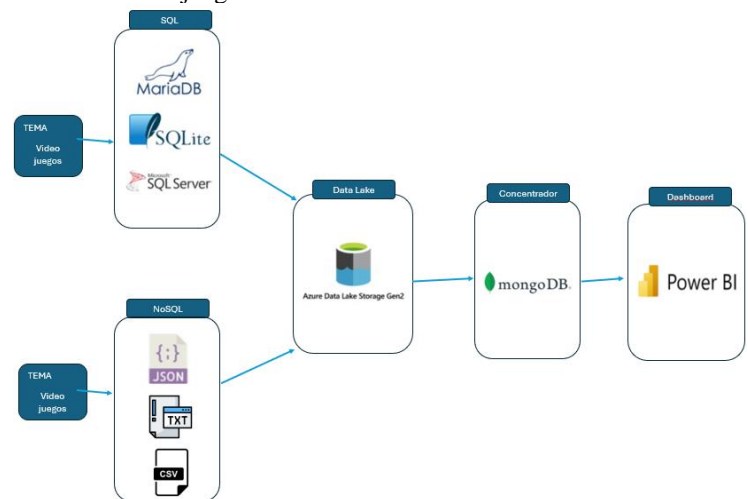
Análisis de los datos:

- MySQL, es una base de datos relacional que se utiliza principalmente para almacenar y analizar datos estructurados.
- Azure Data Lake Gen2, permite almacenar grandes volúmenes de datos no estructurados para su posterior transformación y análisis.
- SQL Server, se especializa en almacenar y analizar datos estructurados.
- MongoDB, es una base de datos NoSQL que se especializa en el almacenamiento y análisis de datos no estructurados o semiestructurados, esto lo hace ideal para trabajar con datos menos predecibles y más flexibles.

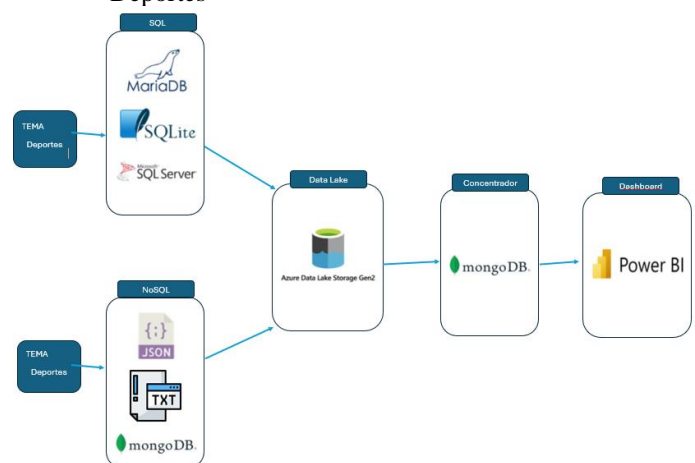
VII.ARQUITECTURA DE DATOS

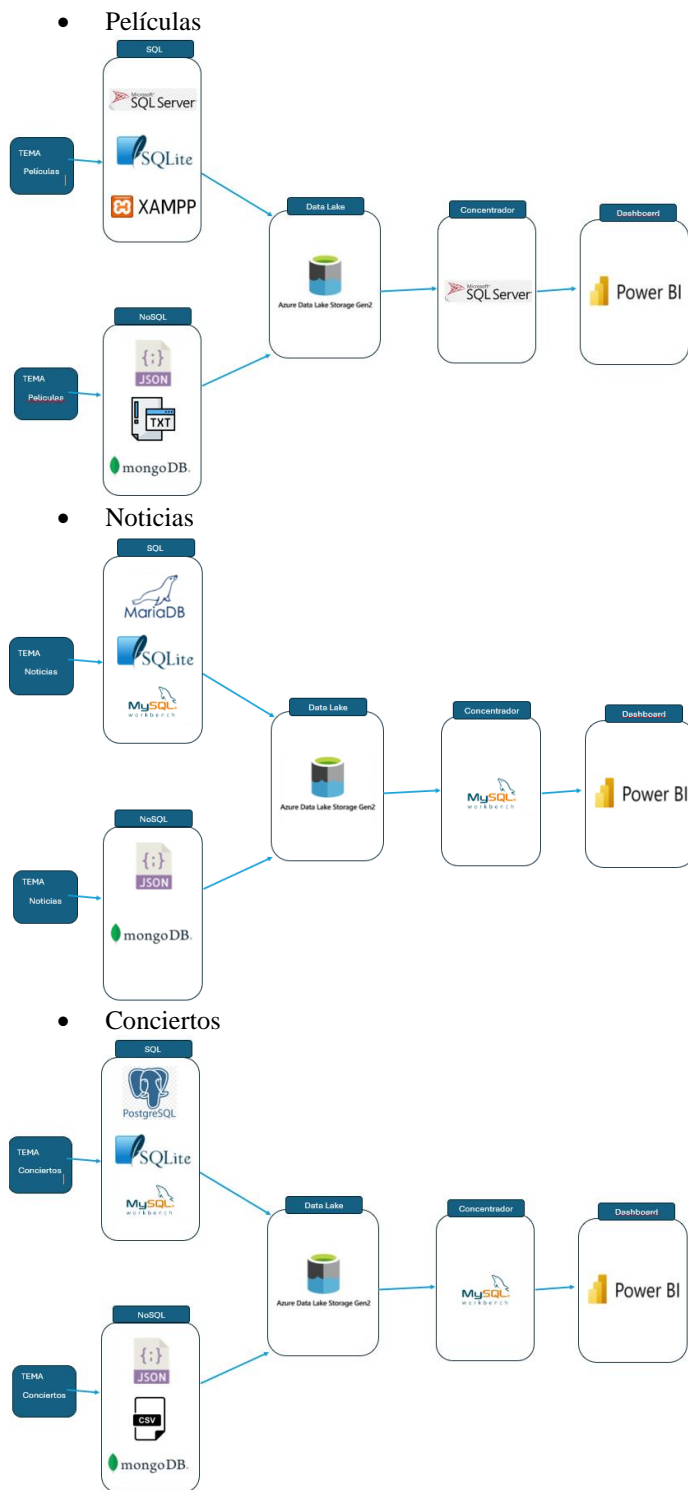
A continuación, se presenta la arquitectura de datos, en base a los casos de estudio seleccionados, los cuales son:

- Video juegos



- Deportes





VIII. EXTRACCIÓN DE DATOS

Fuentes:

<https://archive.ics.uci.edu/>
<https://www.opendatane트워크.com/>
<https://data.fivethirtyeight.com/>
<https://www.reddit.com/r/datasets/>
<https://beta.data.gov.sg/datasets>
<https://www.kaggle.com/datasets/fronkongames/steam-games-dataset>
<https://www.kaggle.com/datasets/arnabchaki/popular-video-games-1980-2023>

<https://www.kaggle.com/datasets/sidtwr/videogames-sales-dataset>

<https://www.kaggle.com/datasets/gpreda/bbc-news>

<https://data.world/robertjoellewis/film-subtitles>

<https://data.world/crawlfeeds/cnbc-news-dataset>

<https://data.world/opensnippets/cnn-news-dataset>

<https://data.world/sports/olympics>

• Web Scrapping

<https://www.metacritic.com/browse/game/>

<https://www.metacritic.com/browse/movie/>

<https://concertful.com/top?page=1>

deportes2: <https://www.kaggle.com/datasets/mcarujo/fifa-world-cup-2022-catar>

deportes1:

<https://www.kaggle.com/datasets/stefanoleone992/ea-sports-fc-24-complete-player-dataset>

deportes3:

<https://www.kaggle.com/datasets/bhanupratapbiswas/olympic-data>

deportes4: <https://www.kaggle.com/datasets/brenda89/fifa-world-cup-2022>

deportes5:

<https://www.kaggle.com/datasets/datasciencedonut/olympic-swimming-1912-to-2020>

deportes6:

<https://www.kaggle.com/datasets/mohsenzergani/data-1-video-juegos6>

<https://www.kaggle.com/datasets/arnabchaki/popular-video-games-1980-2023>

IV. ANÁLISIS DE INFORMACIÓN

• NOTICIAS

- NOTICIAS 1: Mandar a Mysql Workbench

The screenshot shows the MySQL Workbench interface. The top panel displays a query script for creating a database engine and inserting data from a dataset named 'primer_noticia'. The middle panel shows the 'Query 1' window with the executed query: `use primer_noticia; select * from noticia_1;`. The bottom panel shows the 'Result Grid' with a table of news items, including columns for IDLink, Title, and Headline. The table contains 10 rows of data, including news about Obama, the Chinese economy, and Finland's GDP.

✓ **Limpieza de datos**

1. Leer el dataset

```
data1 = pd.read_csv('noticia1.csv')
data1
```

2. Ver los tipos de datos

```
data1.dtypes
```

```
IDLink      float64
Title       object
Headline    object
Source      object
Topic       object
PublishDate object
SentimentTitle float64
SentimentHeadline float64
Facebook    int64
GooglePlus  int64
LinkedIn    int64
dtype: object
```

3. Rellenar los strings con 'sin información' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data1.loc[:, data1.dtypes == object] = data1.loc[:, data1.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data1.loc[:, data1.dtypes != object] = data1.loc[:, data1.dtypes != object].fillna(0)
```

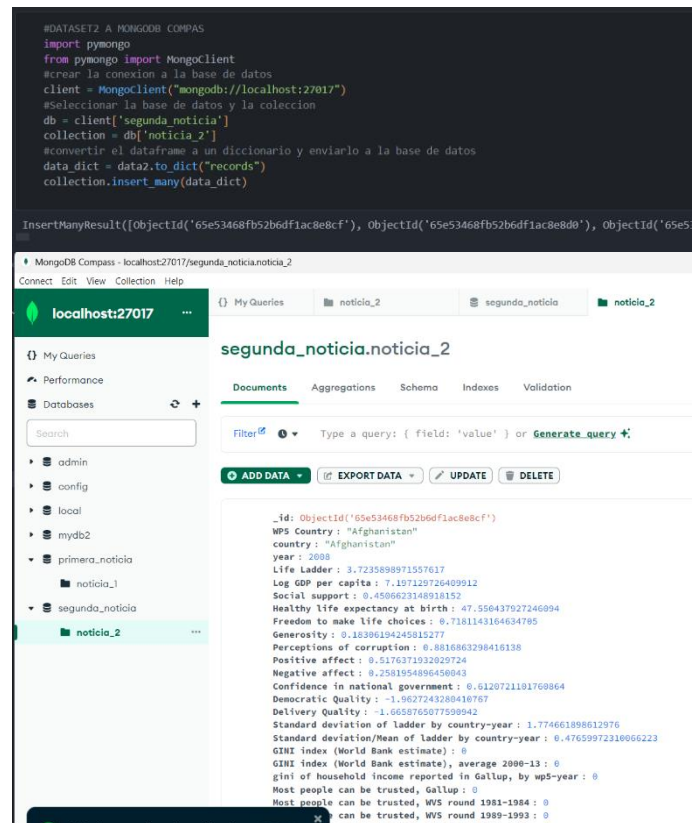
4. Eliminar registros duplicados

```
#eliminar duplicados
data1 = data1.drop_duplicates()
data1
```

5. Exportar el dat set

```
#SEGUNDA DATA SET
data2 = pd.read_excel('noticia2.xlsx')
data2
```

- NOTICIAS 2: Enviar a Monfo db compas



1. Leer el dataset

```
#SEGUNDA DATA SET
data2 = pd.read_excel('noticia2.xlsx')
data2
```

2. Rellenar los strings con 'sin infromacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data2.loc[:, data2.dtypes == object] = data2.loc[:, data2.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data2.loc[:, data2.dtypes != object] = data2.loc[:, data2.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
data2 = data2.drop_duplicates()
data2
```

4. Exportar el dat set

```
#exportar
data2.to_excel('noticias_2.xlsx', index=False)
```

- NOTICIA 3: Se encuentra en formato JSON

NOTICIAS 4: Enviar SQLite

```
#DATASET4 A SQLite
import sqlite3
import pandas as pd

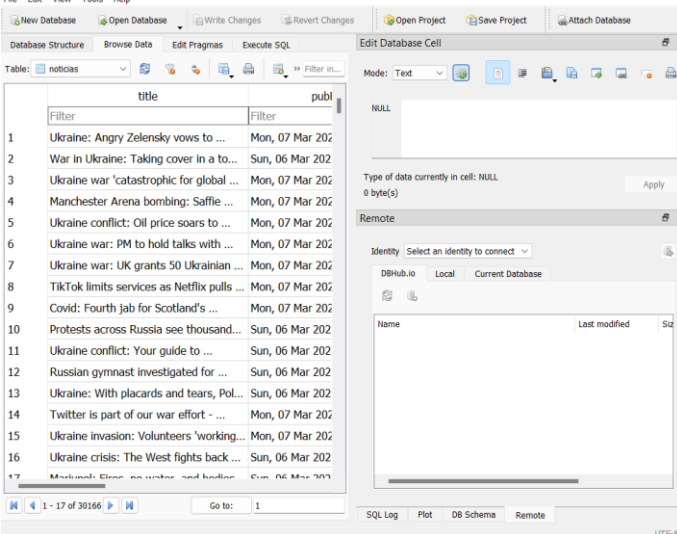
# Carga tu dataset en un DataFrame
data4 = pd.read_csv('noticias_4.csv')

# Crea una conexión a la base de datos SQLite
conn = sqlite3.connect('cuarta_noticia.db')

# Escribe el DataFrame a una tabla en SQLite
data4.to_sql('noticias', conn, if_exists='replace', index=False)

# No olvides cerrar la conexión
conn.close()
```

✓ 6.2s



	title	publ
1	Ukraine: Angry Zelensky vows to ...	Mon, 07 Mar 202
2	War in Ukraine: Taking cover in a to...	Sun, 06 Mar 202
3	Ukraine war 'catastrophic for global ...	Mon, 07 Mar 202
4	Manchester Arena bombing: Saffie ...	Mon, 07 Mar 202
5	Ukraine conflict: Oil price soars to ...	Mon, 07 Mar 202
6	Ukraine war: PM to hold talks with ...	Mon, 07 Mar 202
7	Ukraine war: UK grants 50 Ukrainian ...	Mon, 07 Mar 202
8	TikTok limits services as Netflix pulls ...	Mon, 07 Mar 202
9	Covid: Fourth jab for Scotland's ...	Mon, 07 Mar 202
10	Protests across Russia see thousand...	Sun, 06 Mar 202
11	Ukraine conflict: Your guide to ...	Sun, 06 Mar 202
12	Russian gymnast investigated for ...	Sun, 06 Mar 202
13	Ukraine: With placards and tears, Pol...	Sun, 06 Mar 202
14	Twitter is part of our war effort - ...	Mon, 07 Mar 202
15	Ukraine invasion: Volunteers 'working...	Mon, 07 Mar 202
16	Ukraine crisis: The West fights back ...	Sun, 06 Mar 202
17	Musical: Elton, number and bodie...	Sun, 06 Mar 202

1. Leer el dataset

```
#cuarto dataset
data4 = pd.read_csv('noticia4.csv')
data4
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data4.loc[:, data4.dtypes == object] = data4.loc[:, data4.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data4.loc[:, data4.dtypes != object] = data4.loc[:, data4.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
data4 = data4.drop_duplicates()
data4
```

4. Exportar el dat set

```
#exportar
data4.to_csv('noticias_4.csv', index=False)
```

- NOTICIAS 5: Se encuentra en formato JSON

- NOTICIA 6: Enviar a Maria DB

1. Leer el dataset

```
#leer dataset
import pandas as pd
data6 = pd.read_csv('noticia6.csv')
```

✓ 2.8s

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data6.loc[:, data6.dtypes == object] = data6.loc[:, data6.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data6.loc[:, data6.dtypes != object] = data6.loc[:, data6.dtypes != object].fillna(0)
```

✓ 0.0s Python

3. Eliminar registros duplicados

```
data6 = data6.drop_duplicates()
```

✓ 0.0s

4. Exportar el dat set

```
data6.to_csv('noticias_6.csv', index=False)
```

✓ 0.2s Python

• PELICULAS

- PELICULAS 1: Enviar SQLserver

1. Leer el dataset

```
data1 = pd.read_csv('peliculas1.csv')
data1
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data1.loc[:, data1.dtypes == object] = data1.loc[:, data1.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data1.loc[:, data1.dtypes != object] = data1.loc[:, data1.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
#eliminar duplicados
data1 = data1.drop_duplicates()
data1
```

Python

4. Exportar el dat set

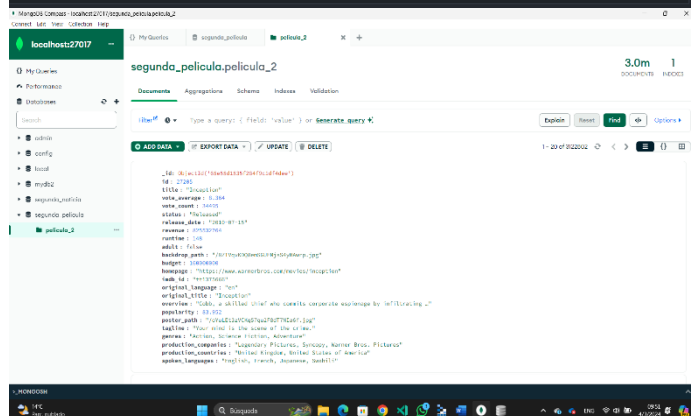
```
data1.to_csv('pelicula_1.csv', index=False)
```

Python

- PELICULAS2: Enviar a MONGO DB

```
#DATASET2 A MONGODB
import pymongo
from pymongo import MongoClient
#crear la conexion a la base de datos
client = MongoClient("mongodb://localhost:27017")
#Seleccionar la base de datos y la coleccion
db = client['segunda_pelicula']
collection = db['pelicula_2']
#convertir el dataframe a un diccionario y enviarlo a la base de
datos
data_dict = data2.to_dict("records")
collection.insert_many(data_dict)
```

Python



1. Leer el dataset

```
#segunda dataset
data2 = pd.read_csv('peliculas2.csv')
data2
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin informacion'
data2.loc[:, data2.dtypes == object] = data2.loc[:, data2.
dtypes == object].fillna('sin informacion')

# Rellenar los valores nulos en columnas numéricas con 0
data2.loc[:, data2.dtypes != object] = data2.loc[:, data2.
dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data2 = data2.drop_duplicates()
data2
```

Python

4. Exportar el dat set

```
data2.to_csv('pelicula_2.csv', index=False)
```

Python

- PELICULAS 3: Transformar de txt a csv

```
#transformar peliculas_3 a txt
# Escribe el DataFrame a un archivo TXT
data3.to_csv('pelicula_3.txt', index=False, sep='\t')
```

Python

1. Leer el dataset

```
#tercer dataset
data3 = pd.read_csv('peliculas3.csv')
data3
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin informacion'
data3.loc[:, data3.dtypes == object] = data3.loc[:, data3.
dtypes == object].fillna('sin informacion')

# Rellenar los valores nulos en columnas numéricas con 0
data3.loc[:, data3.dtypes != object] = data3.loc[:, data3.
dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data3 = data3.drop_duplicates()
data3
```

Python

4. Exportar el dat set

```
data3.to_csv('pelicula_3.csv', index=False)
```

Python

- PELICULA 4: formato csv

1. Leer el dataset

```
#cuarto dataset
data4 = pd.read_excel('peliculas4.xlsx')
data4
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin informacion'
data4.loc[:, data4.dtypes == object] = data4.loc[:, data4.
dtypes == object].fillna('sin informacion')

# Rellenar los valores nulos en columnas numéricas con 0
data4.loc[:, data4.dtypes != object] = data4.loc[:, data4.
dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados


```
data4 = data4.drop_duplicates()
data4
```

Python

4. Exportar el dat set

```
data4.to_excel('pelicula_4.xlsx', index=False)
```

Python

- Pelicula 5: Enviar el csv a sqlite
- 1. Leer el dataset

```
#DATA SET 5
data5 = pd.read_csv('pelicula5.csv')
data5
```

Python

- 2. Rellenar los strings con 'sin informacion' y numero con 0

```
#eliminar datos duplicados
data5 = data5.drop_duplicates()
data5
```

Python

- 3. Eliminar registros duplicados

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data5.loc[:, data5.dtypes == object] = data5.loc[:, data5.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data5.loc[:, data5.dtypes != object] = data5.loc[:, data5.dtypes != object].fillna(0)
```

Python

- 4. Exportar el dat set

```
#exportar
data5.to_csv('pelicula_5.csv', index=False)
```

Python

```
#DATASET 5 A SQLITE
#Concierto 1 de csv a SQLITE
import sqlite3
import pandas as pd

# Carga tu dataset en un DataFrame
data5 = pd.read_csv('pelicula_5.csv')

# Crea una conexión a la base de datos SQLite
conn = sqlite3.connect('quinta_pelicula.db')

# Escribe el DataFrame a una tabla en SQLite
data5.to_sql('peliculas', conn, if_exists='replace', index=False)

# No olvides cerrar la conexión
conn.close()
```

- Pelicula 6: Enviar a xampp

- 1. Leer el dataset

```
#DATASET 6
data6 = pd.read_csv('peliculas6.csv')
```

Python

- 2. Rellenar los strings con 'sin informacion' y numero con 0

```
#eliminar duplicados
data6 = data6.drop_duplicates()
data6
```

Python

- 3. Eliminar registros duplicados

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data6.loc[:, data6.dtypes == object] = data6.loc[:, data6.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data6.loc[:, data6.dtypes != object] = data6.loc[:, data6.dtypes != object].fillna(0)
```

Python

- 4. Exportar el dat set

```
#exportar
data6.to_csv('películas_6.csv', index = False)
```

Python

- CONCIERTOS
- CONCIERTO1: Enviar A SQLITE

```
#Concierto 1 de csv a SQLITE
import sqlite3
import pandas as pd

# Carga tu dataset en un DataFrame
data1 = pd.read_csv('concierto_1.csv')

# Crea una conexión a la base de datos sqlite
conn = sqlite3.connect('primer_concierto.db')

# Escribe el DataFrame a una tabla en sqlite
data1.to_sql('concierto', conn, if_exists='replace', index=False)

# No olvides cerrar la conexión
conn.close()
```

Python

ID	Date and Time	Genre	Location	Artist
1	08/21/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
2	08/22/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
3	08/23/2008 11:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
4	08/24/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
5	08/25/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
6	08/26/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
7	08/27/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
8	08/28/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
9	08/29/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
10	08/30/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
11	08/31/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
12	09/01/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
13	09/02/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
14	09/03/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
15	09/04/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
16	09/05/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
17	09/06/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
18	09/07/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
19	09/08/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
20	09/09/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden

1. Leer el dataset

```
data1 = pd.read_csv('concierto1.csv')
data1
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin informacion'
data1.loc[:, data1.dtypes == object] = data1.loc[:, data1.dtypes == object].fillna('sin informacion')

# Rellenar los valores nulos en columnas numéricas con 0
data1.loc[:, data1.dtypes != object] = data1.loc[:, data1.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data1 = data1.drop_duplicates()
data1
```

Python

4. Exportar el dat set

```
data1.to_csv('concierto_1.csv', index=False)
```

Python

- CONCIERTO 2: Enviar a postgrade

1. Leer el dataset

```
#DATASET 2
import pandas as pd
data2 = pd.read_csv('concierto2.csv')
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin informacion'
data2.loc[:, data2.dtypes == object] = data2.loc[:, data2.dtypes == object].fillna('sin informacion')

# Rellenar los valores nulos en columnas numéricas con 0
data2.loc[:, data2.dtypes != object] = data2.loc[:, data2.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data2 = data2.drop_duplicates()
data2
```

Python

4. Exportar el dat set

```
data2.to_csv('concierto_2.csv', index=False)
```

Python

- CONCIERTO 3: Enviar a Mysql

```
#Mandar el dataset 3 a mysql
from sqlalchemy import create_engine
#crear la conexion a la base de datos
engine = create_engine('mysql+pymysql://root:123456@localhost:3307/concierto3')
#enviar el dataframe a la base de datos
data3.to_sql('concierto_3', con=engine, if_exists='replace', index=False)
```

ID	Date and Time	Genre	Location	Artist
1	08/21/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
2	08/22/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
3	08/23/2008 11:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
4	08/24/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
5	08/25/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
6	08/26/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
7	08/27/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
8	08/28/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
9	08/29/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
10	08/30/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
11	08/31/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
12	09/01/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
13	09/02/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
14	09/03/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
15	09/04/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
16	09/05/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
17	09/06/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
18	09/07/2008 10:00:00 AM	Pop	Madison Square Garden	Madison Square Garden
19	09/08/2008 03:00:00 PM	Pop	Madison Square Garden	Madison Square Garden
20	09/09/2008 07:00:00 PM	Pop	Madison Square Garden	Madison Square Garden

1. Leer el dataset

```
#DATASET 3
import pandas as pd
data3 = pd.read_csv('conciertos3.csv', encoding='latin1')
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0


```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data3.loc[:, data3.dtypes == object] = data3.loc[:, data3.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data3.loc[:, data3.dtypes != object] = data3.loc[:, data3.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

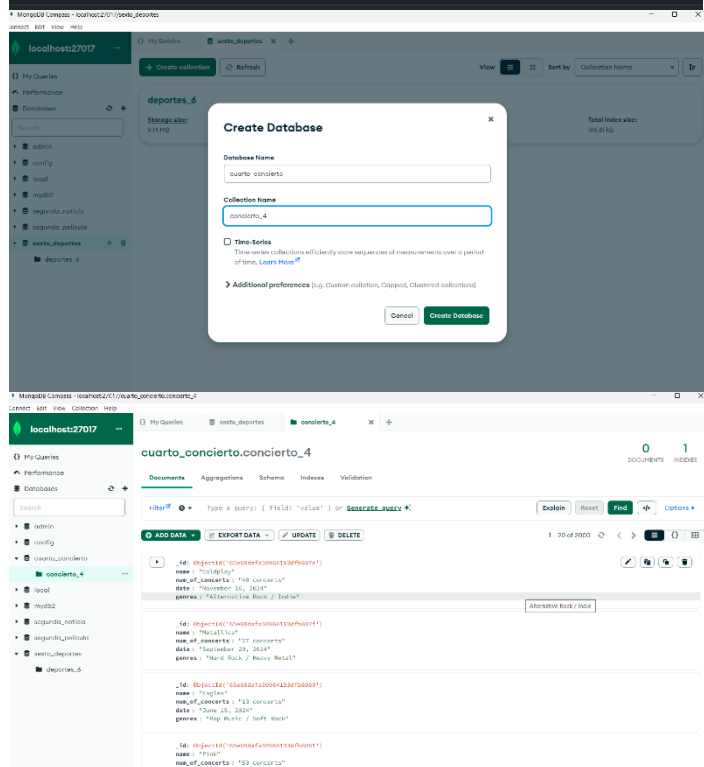
```
#eliminar datos duplicados
data3 = data3.drop_duplicates()
data3
```

4. Exportar el dat set

```
#exportar
data3.to_csv('concierto_3.csv', index=False)
```

- CONCIERTO 4: Enviar a MONGO DB

```
#Cuarto dataset4 enviar a mongo DB
import pymongo
from pymongo import MongoClient
#crear la conexion a la base de datos
client = MongoClient("mongodb://localhost:27017")
#Seleccionar la base de datos y la coleccion
db = client['cuarto_concierto']
collection = db['concierto_4']
#convertir el dataframe a un diccionario y enviarlo a la
base de datos
data_dict = data4.to_dict("records")
collection.insert_many(data_dict)
```



1. Leer el dataset

```
#DATASET 4
import pandas as pd
data4 = pd.read_csv('conciertos4.csv')
```

2. Rellenar los strings con 'sin infromacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data4.loc[:, data4.dtypes == object] = data4.loc[:, data4.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data4.loc[:, data4.dtypes != object] = data4.loc[:, data4.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
#eliminar datos duplicados
data4 = data4.drop_duplicates()
data4
```

4. Exportar el dat set

```
#exportar
data4.to_csv('concierto_4.csv', index=False)
```

- DEPORTES
- Deportes 1 json

Transformar de csv a json

```
#DATASET 1 A JSON
data1.to_json('deportes_1.json', orient='records')
```

1. Leer el dataset

```
data1 = pd.read_csv('deportes1.csv')
data1
```

2. Rellenar los strings con 'sin infromacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data1.loc[:, data1.dtypes == object] = data1.loc[:, data1.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data1.loc[:, data1.dtypes != object] = data1.loc[:, data1.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
data1 = data1.drop_duplicates()
data1
```

4. Exportar el dat set

```
data1.to_csv('deportes_1.csv', index=False)
```

- Deportes 2: enviar Maria db

```
# Exportar el archivo limpio a MariaDB
from sqlalchemy import create_engine

engine = create_engine('mysql+pymysql://root:12345@localhost:3308/deportes_2')

data2.to_sql('deportes_2', con=engine, if_exists='replace', index=False)
```

[illegible]

1. Leer el dataset

```
#DATSET 2
data2 = pd.read_csv('deportes2.csv')
data2
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data2.loc[:, data2.dtypes == object] = data2.loc[:, data2.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data2.loc[:, data2.dtypes != object] = data2.loc[:, data2.dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
data2 = data2.drop_duplicates()
data2
```

4. Exportar el dat set como xlsx

```
data2.to_excel('deportes_2.xlsx', index=False)
```

- Deportes 3: enviar a SQLITE

```
#DATASET 3 A SQLITE
#Concierto 1 de csv a SQLITE
import sqlite3
import pandas as pd
```

Address bar: http://www.olympic.org/... 2015-2016 Winter season results

Page title: 2015-2016 Winter season results

Table columns: ID, Name, Sex, Age, Height, Weight, Team, RUC, Gender, Year, Season, City, etc.

Table data (first 10 rows):

ID	Name	Sex	Age	Height	Weight	Team	RUC	Gender	Year	Season	City
1	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
2	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
3	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
4	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
5	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
6	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
7	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
8	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
9	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi
10	A. B. B. B. B.	M	24	180	80	USA	2015	Winter	2015	Winter	Sochi

Page footer: 2015-2016 Winter season results

1. Leer el dataset

```
#DATASET 3
data3 = pd.read_csv('deportes3.csv')
data3
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data3.loc[:, data3.dtypes == object] = data3.loc[:, data3.
dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data3.loc[:, data3.dtypes != object] = data3.loc[:, data3.
dtypes != object].fillna(0)
```

3. Eliminar registros duplicados

```
data3 = data3.drop_duplicates()
data3
```

4. Exportar el dat set

```
data3.to_excel('deportes 3.xlsx', index=False)
```

- Deportes 4: enviar al SQL server

1. Leer el dataset

```
#DATASET 4
data4 = pd.read_csv('deportes4.csv')
data4
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data4.loc[:, data4.dtypes == object] = data4.loc[:, data
dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data4.loc[:, data4.dtypes != object] = data4.loc[:, data
dtypes != object].fillna(0)
```

- ### 3. Eliminar registros duplicados

```
data4 = data4.drop_duplicates()
data4
```

- #### 4. Exportar el dat set

```
data4.to_excel('deportes_4.xlsx', index=False)
```

- Deportes 5 Elastic search

[illegible]

- ## 1. Leer el dataset

```
#DATASET 5
data5 = pd.read_csv('deportes5.csv')
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con 'sin información'
data5.loc[:, data5.dtypes == object] = data5.loc[:, data5.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas numéricas con 0
data5.loc[:, data5.dtypes != object] = data5.loc[:, data5.dtypes != object].fillna(0)
```

- ### 3. Exportar el dat set

```
#exportar
data5.to_csv('deportes_5.csv', index=False)
```

Python

- Deportes 6: Enviar a Mongo db

```
#DATASET6 A MONGODB COMPAS
import pymongo
from pymongo import MongoClient
#crear la conexion a la base de datos
client = MongoClient("mongodb://localhost:27017")
#Seleccionar la base de datos y la coleccion
db = client['sexto_deportes']
collection = db['deportes_6']
#convertir el dataframe a un diccionario y enviarlo a la base de
datos
data_dict = data6.to_dict("records")
collection.insert_many(data_dict)
```

```
InsertManyResult([ObjectId('65e629ba208f868d0fbf99dd'), ObjectId('65e629
```

[illegible]

- ## 1. Leer el dataset

```
#DATASET 6
data6 = pd.read_csv('deportes6.csv')
```

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de texto con
'sin información'
data6.loc[:, data6.dtypes == object] = data6.loc[:, data6.
dtypes == object].fillna('sin información')
```

```
# Rellenar los valores nulos en columnas numéricas con 0
data6.loc[:, data6.dtypes != object] = data6.loc[:, data6.
dtypes != object].fillna(0)
```

Python

- ### 3. Eliminar registros duplicados

```
#eliminar datos duplicados
data6 = data6.drop_duplicates()
data6
```

Python

- #### 4. Exportar el dat set

```
#exportar
data6.to_csv('deportes_6.csv', index=False)
```

Python

- VIDEOJUEGOS
- Videojuego 1: En formato csv
- 1. Leer el dataset

```
#primer dataset
data1 = pd.read_csv('videojuegos1.csv')
data1
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data1.loc[:, data1.dtypes == object] = data1.loc[:, data1.dtypes == object].fillna('sin información')

# Rellenar los valores nulos en columnas
numéricas con 0
data1.loc[:, data1.dtypes != object] = data1.loc[:, data1.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data1 = data1.drop_duplicates()
data1
```

Python

4. Exportar el dat set

```
data1.to_csv('videojuegos_1.csv', index=False)
```

Python

Video juego2: Formato json
Trasformar a json

```
#trasnformar videojuegos 2 a JSON
data2.to_json('videojuegos_2.json',
orient='records')
```

Python

1. Leer el dataset

```
data2 = pd.read_csv('videojuegos2.csv')
data2
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data2.loc[:, data2.dtypes == object] = data2.loc[:, data2.dtypes == object].fillna('sin información')
```

```
# Rellenar los valores nulos en columnas
numéricas con 0
data2.loc[:, data2.dtypes != object] = data2.loc[:, data2.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data2 = data2.drop_duplicates()
data2
```

Python

4. Exportar el dat set

```
data2.to_csv('videojuegos_2.csv', index=False)
```

Python

- Video juego 3: Formato txt
- 1. Leer el dataset

```
#tercer data set
data3 = pd.read_csv('videojuegos3.csv')
data3
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data3.loc[:, data3.dtypes == object] = data3.loc[:, data3.dtypes == object].fillna('sin información')
```

```
# Rellenar los valores nulos en columnas
numéricas con 0
data3.loc[:, data3.dtypes != object] = data3.loc[:, data3.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data3 = data3.drop_duplicates()
data3
```

Python

4. Exportar el dat set

```
data3.to_csv('videojuegos_3.csv', index=False)
```

Python

- Video juego 4: Enviar a sql server
- 1. Leer el dataset

```
#cuarta data set
data4 = pd.read_csv('videojuegos4.csv')
data4
```

Pyti

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data4.loc[:, data4.dtypes == object] = data4.loc
[:, data4.dtypes == object].fillna('sin
información')

# Rellenar los valores nulos en columnas
numéricas con 0
data4.loc[:, data4.dtypes != object] = data4.loc
[:, data4.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
data4 = data4.drop_duplicates()
data4
```

Python

4. Exportar el dat set

```
data4.to_csv('videojuegos_4.csv', index=False)
```

Python

- Video juego 5: Enviar APOSTGRADE

1. Leer el dataset

```
#DATA SET 5
data5 = pd.read_csv('videojuegos5.csv')
data5
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data5.loc[:, data5.dtypes == object] = data5.loc
[:, data5.dtypes == object].fillna('sin
información')

# Rellenar los valores nulos en columnas
numéricas con 0
data5.loc[:, data5.dtypes != object] = data5.loc
[:, data5.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
#eliminar datos duplicados
data5 = data5.drop_duplicates()
data5
```

Python

4. Exportar el dat set

```
#exportar
data5.to_csv('videojuegos_5.csv', index=False)
```

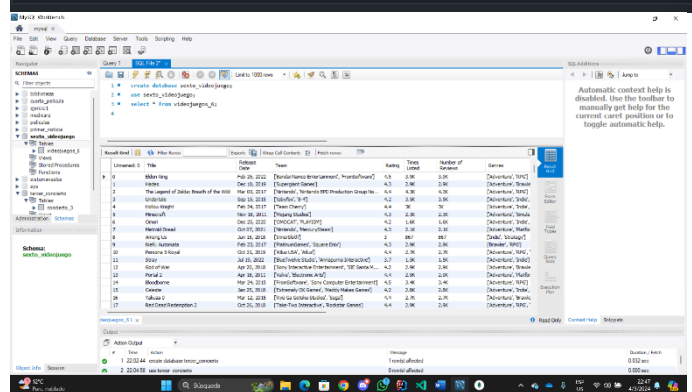
Python

- Video juego 6: Enviar a my sql workbench

```
#enviar el dataset 6 a mysql
from sqlalchemy import create_engine
#crear la conexion a la base de datos
engine = create_engine('mysql+pymysql://
root:1726405390@localhost:3307/sexta_videojuego')
#enviar el dataframe a la base de datos
data6.to_sql('videojuegos_6', con=engine,
if_exists='replace', index=False)
```

✓ 1.2s

Python



1. Leer el dataset

```
#eliminar duplicados
data6 = data6.drop_duplicates()
data6
```

Python

2. Rellenar los strings con 'sin informacion' y numero con 0

```
# Rellenar los valores nulos en columnas de
texto con 'sin información'
data6.loc[:, data6.dtypes == object] = data6.loc
[:, data6.dtypes == object].fillna('sin
información')

# Rellenar los valores nulos en columnas
numéricas con 0
data6.loc[:, data6.dtypes != object] = data6.loc
[:, data6.dtypes != object].fillna(0)
```

Python

3. Eliminar registros duplicados

```
#eliminar duplicados
data6 = data6.drop_duplicates()
data6
```

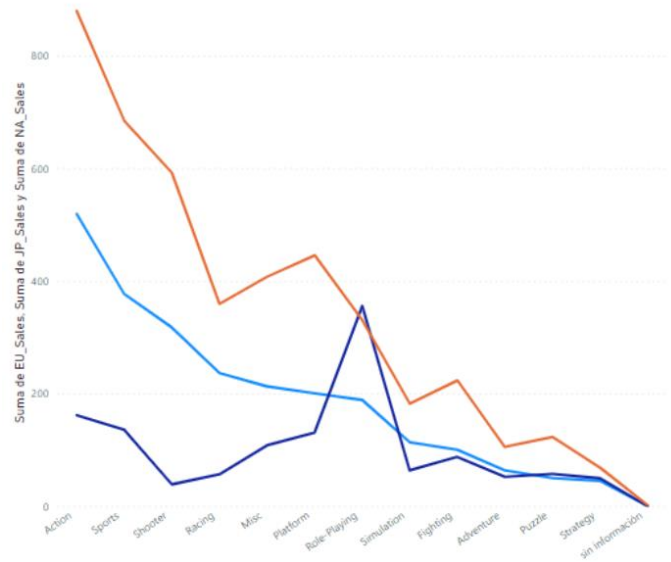
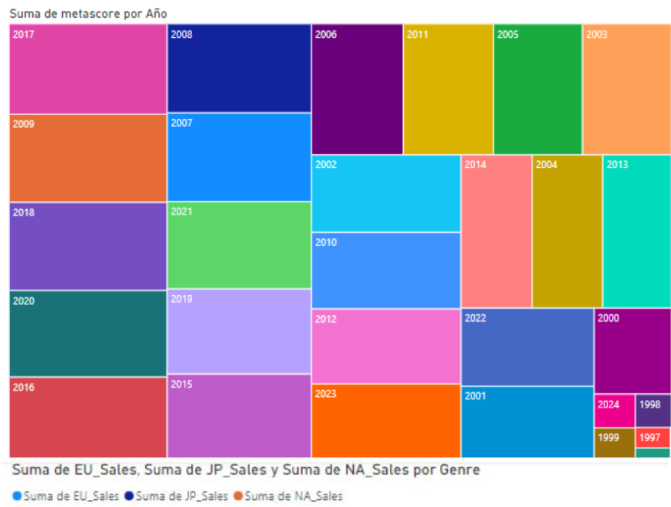
Python

4. Exportar el dat set

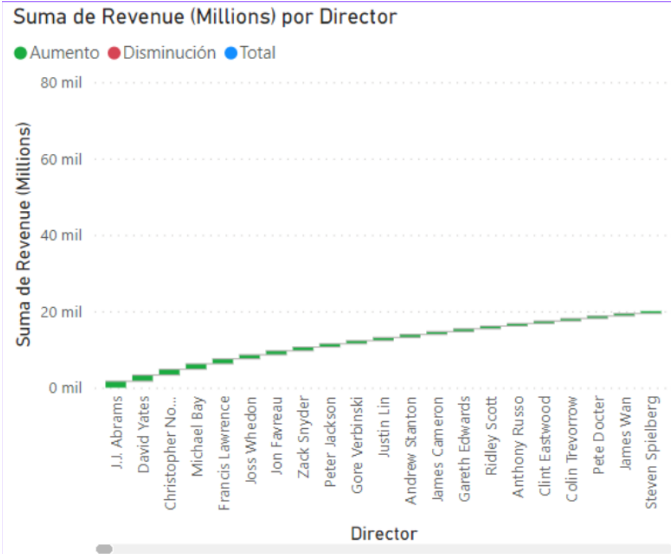
V. VISUALIZACIÓN Y RESULTADOS OBTENIDOS

- Video juegos

Años con mejores puntuaciones generales

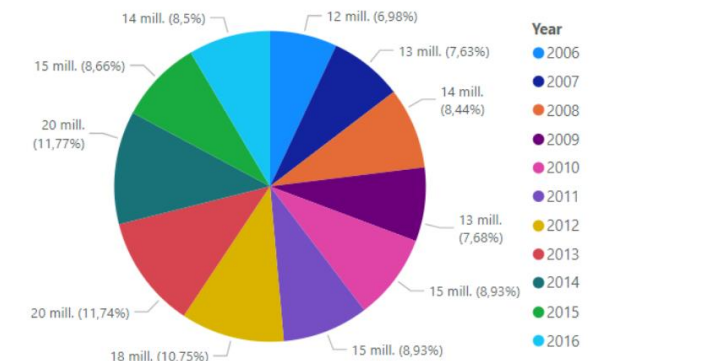


- Películas

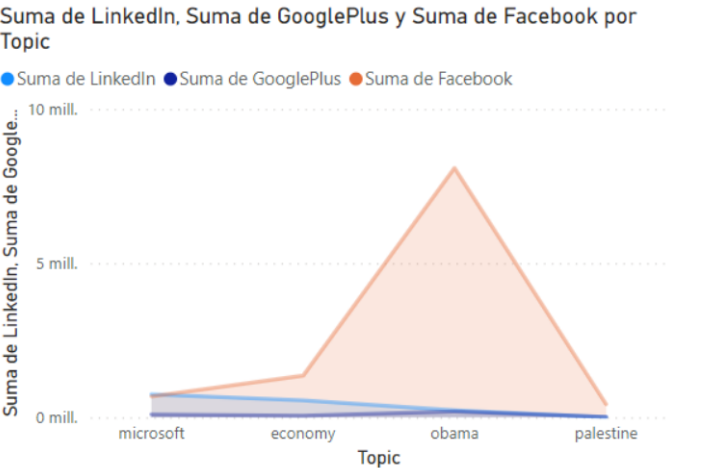
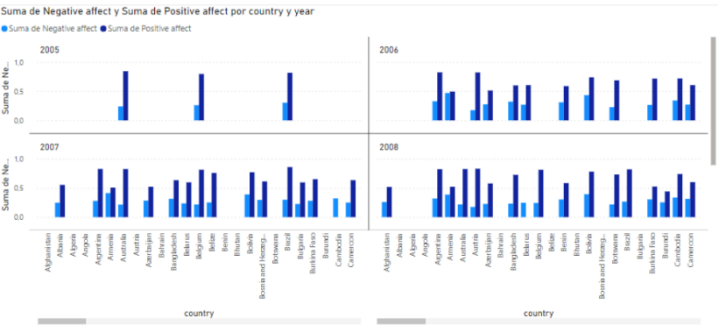


Numero de votantes de películas por año

Suma de Votes por Year



- Noticias



- Deportes



VI.CONCLUSIONES Y RECOMENDACIONES

- Se implementó una arquitectura de Data Lake que permitió la recolección, integración y almacenamiento de grandes volúmenes de datos.
- La arquitectura de Data Lake utiliza Microsoft Azure como plataforma en la nube, integrando con las bases de datos.
- Se identificaron analizaron los índices y métricas relevantes para cada uno de los casos de estudio.
- Se realizaron los dashboard respectivos.
- Realizar una buena investigación para la recolección de datasets sea eficiente, tenga calidad y sea veraz.
- Automatizar en la medida de lo posible los procesos de recolección, limpieza y carga de datos en el Data Lake con el fin de optimizar el flujo de trabajo.

VII.DESAFÍOS Y PROBLEMAS ENCONTRADOS

El desafío principal al momento de realizar este proyecto fue la búsqueda de datasets. En algunos casos, fue difícil encontrar datasets que se ajustaran a las necesidades específicas del proyecto, lo que requirió un esfuerzo adicional por parte del equipo para buscar alternativas y adaptar los datos disponibles. ENLACE de github del proyecto.

RECONOCIMIENTOS

Se agradece a la ingeniera por su dedicación y enseñanza durante el desarrollo del proyecto. Su guía y apoyo fueron fundamentales para el éxito del equipo en la implementación de la arquitectura de Data Lake y la realización de los casos de estudio.

LINK

- Presentacion Canva
<https://www.canva.com/design/DAF-mLei9Tc/QyR2WtXOtgbpGrgfb5Apyw/edit>
- Drive
https://epnecuador-my.sharepoint.com/:f/g/personal/john_mata_epn_edu_ec/EpAoqD2iXDNBncR7EtkT5CwBkHC5_DsrG_QKQNJ5att6xA?e=jn3a3R
- Video
<https://drive.google.com/file/d/1SpNsZ7fbf-KET4xFkJ7inhcdvoFTJ1XY/view?usp=sharing>

REFERENCIAS

[1] "Tutorial: Introducción a la creación en el servicio Power BI - Power BI". Microsoft Learn: Build skills that open doors in your career. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://learn.microsoft.com/es-es/power-bi/fundamentals/service-get-started>

[2] "Steam Games Dataset". Kaggle: Your Machine Learning and Data Science Community. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.kaggle.com/datasets/fronkongames/steam-games-dataset>

[3] "What Movies to Watch Right Now - Metacritic". Movie Reviews, TV Reviews, Game Reviews, and Music Reviews - Metacritic. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.metacritic.com/browse/movie/>

[4] "Open Data Network". Open Data Network. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.opendatanetwork.com/>

[5] "FIFA complete player dataset". Kaggle: Your Machine Learning and Data Science Community. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.kaggle.com/datasets/mohsenzergani/data-1>

[6] "Linguistic data of 32k film subtitles with meta-data - dataset by robertjoellewis". The Data Catalog Platform | data.world. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://data.world/robertjoellewis/film-subtitles>

[7] "CNBC news dataset - dataset by crawlfeeds". The Data Catalog Platform | data.world. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://data.world/crawlfeeds/cnbc-news-dataset>

[8] "CNN news dataset - dataset by opensnippets". The Data Catalog Platform | data.world. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://data.world/opensnippets/cnn-news-dataset>

[9] "What Movies to Watch Right Now - Metacritic". Movie Reviews, TV Reviews, Game Reviews, and Music Reviews - Metacritic. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.metacritic.com/browse/movie/>