Ana María Acosta Pacheco

Informe de proceso ETL al utilizar la API de personajes de Rick and Morty

Proceso ETL

El proceso de extracción, transformación y carga de datos (ETL) se utilizó para recopilar, procesar y cargar datos de la API Rick and Morty haciendo uso de un programa escrito en Python y la herramienta Power BI. El proceso se dividió en 3 pasos que son listados a continuación:

1. **Extracción de Datos**

   En este paso, se importó la biblioteca 'requests' para hacer una solicitud GET a la API de Rick and Morty (https://rickandmortyapi.com/api/character) y recuperar los datos de los personajes en formato JSON. Es importante recalcar que el proceso se llevo a cabo usando un ciclo para recuperar todas los datos en las diferentes páginas en las que estaban contenidos.

2. **Transformación, Limpieza y Acondicionamiento de Datos**

   El uso de la biblioteca Pandas fue fundamental en este paso ya que se utilizó para crear un dataFrame, con el cual se pudo visualizar de mejor forma que categorías contiene cada personaje, su contenido de estas y el tipo de datos con los que se trabajaría. Una vez realizado lo anterior se eligieron las columnas 'status', 'species', 'type', 'gender', 'location' y 'created' que fueron consideradas como valiosas para el análisis de los datos. Se precindió de las los campos 'nombre, ' id', 'origen', 'image', 'episode' y 'url'.

   A continuación, se explican las razones detrás de estas decisiones:

   Campos conservados

   status: La columna "status" permite conocer si un personaje está vivo, muerto o en estado desconocido. Esta información es fundamental para comprender la situación de los personajes y puede ser relevante para analizar patrones o tendencias en relación con el estado de los personajes a lo largo de la serie.

   species: "species" describe la especie del personaje, ya sea humano, alienígena u otra categoría. Esta información puede ser útil a la hora de identificar la diversidad de especies en la serie y a explorar posibles correlaciones entre la especie y otros atributos.

   type: Esta categoría se incluyó para comprender la variedad de tipos en cada especie y para explorar cualquier patrón o tendencia que pueda surgir al analizar estos.

   gender: Aquí se identifica el género del personaje, ya sea masculino, femenino, sin genero o desconocido. La distribución de genero puede ser relevante para analizar posibles temas relacionados con el género y su relación con la ubicación, status, localización ,etc. Estos datos serian super útiles si fuéramos una empresa de ventas en línea y quisiéramos recomendarle productos a los personajes de la serie 🤑

   location: Describe la ubicación actual del personaje o la última ubicación donde se le vio. Al conservar esta información, se puede analizar dónde se encuentran los personajes, algún tipo

de especie en específico o algún genero de personajes, y si hay ubicaciones recurrentes en la serie.

created: Contiene la fecha y hora en que se creó el registro del personaje. Se utilizo solo la parte del año de la fecha, esto para ver que tipos de personajes fueron introducidos en los años que ha estado activa la serie. Se pueden realizar análisis de series temporales o para explorar la evolución de los personajes a lo largo del tiempo. *Aunque el verdadero motivo de conservar ese campo fue para demostrar la habilidad en poder extraer solo el año de la fecha y poder usarla en el tablero BI.*

Campos no conservados

nombre e id: Estos campos son únicos para cada personaje y, aunque son valiosos para identificarlos individualmente, no aportan una perspectiva analítica significativa en relación con otros atributos. Por lo tanto, se optó por centrarse en características más descriptivas y que permitieran realizar un análisis más amplio de los datos.

origin: Describe la locación de origen del personaje, incluida una URL que apunta a detalles específicos. Aunque si bien el origen de los personajes puede ser relevante para tener en cuenta donde se originan más especies y sus tipos, esta información no se consideró relevante para los propósitos del tablero, en cambio se enfoca más en la ubicación actual de los personajes.

image, episode y url: Estos campos contienen información sobre la imagen del personaje, los episodios en los que aparece y las URL asociadas. Las imágenes y url realmente no aportan nada al análisis de los datos en este caso, por otro lado si bien los episodios podrían aportar algunos datos como en que episodio sale algún tipo de especie en específico, entre otras cosas, no se consideraron relevantes como los campos si conservados.

Por otro lado, es importante mencionar la transformación de la columna "type" ya que esta contaba con cadenas vacías como datos y se decidió reemplazarlos con la etiqueta "Unknown" utilizando el método replace().

Además, se llevaron a cabo transformaciones para extraer solo el nombre de los campos ‘origin’ y ‘location’, eliminando los detalles innecesarios de la ubicación como url de esta, así como se extrajo solo el año del campo ‘created’. Esto permitió dejar los datos en condiciones óptimas para el análisis, asegurando consistencia y precisión en los resultados.

3. Carda de Datos

Una vez finalizado el proceso de transformación y limpieza se guardaron los datos en un archivo CSV, creando así una representación persistente para que estos sean fáciles de acceder para diferentes herramientas de análisis, incluyendo Power BI. Los datos quedaron disponibles en un formato tabular y estructurado, facilitando su carga y manipulación futura.

Ana María Acosta Pacheco

Desafíos Enfrentados

En primera instancia al recuperar los datos de la API e imprimirlos era algo que no se podía leer de una manera fluida, se tuvo que implementar una función llamada 'jprint' para poder darle cierto formato a la impresión de los datos y ver con lo que se iba a trabajar, en especial que campos contenía cada personaje y ver la estructura de los datos. También se realizo una suma de campos para comprobar que los datos de los 826 personajes estuvieran completos.

Al momento de cargar los datos a power BI fue cuando me percaté que en el campo 'type' muchos personajes tenían una cadena vacia como dato, decidí regresar a mi código en Python y remplazar estas cadenas vacías por 'unknown' ya que con esa palabra se expresaban muchos otros datos de otras columnas y quise mantener ese estilo (más porque mi primer pensamiento fue poner 'No especifica' 😳).

Power BI era una herramienta de la que había oído hablar bastante pero no había manipulado por mi cuenta, al principio creé varias gráficas que eran bastante sencillas y realmente la información que presentaban nunca termino de convencerme, así que me puse en modo Tony Stark y me hice experta en una noche en Power BI, bueno, lo suficiente para aprender cosas super valiosas y mostrar relaciones importantes entre los datos. No fue la tarea más fácil del mundo (más porque por alguna razón me pide iniciar sesión cada 20 minutos) y sé que aún Power BI tiene mucho que enseñarme creo que se completó la misión satisfactoriamente.

Lecciones aprendidas

Realmente las lecciones aprendidas desde que apliqué a la vacante fueron muchas, en la entrevista me sentía muy nerviosa (hasta le cambié el nombre a Sebastián jeje una disculpa), sin embargo con el transcurso de la platica y su calma me di cuenta que no había nada que temer en estos procesos.

Por otro lado, durante el proceso ETL, pude reafirmar la importancia de la limpieza y el acondicionamiento riguroso de los datos antes de realizar cualquier análisis o creación de modelo posterior. La necesidad de comprender la estructura de los datos que entrega una API y tomar decisiones que demuestren una coherencia al no tomar en cuenta ciertas columnas o al llenar valores faltantes siento que fue una habilidad clave en la manipulación de datos.

Al usar power BI como herramienta de visualización pude descubrir y reafirmar la potencia de las herramientas de visualización interactiva que ofrece. Sin embargo, también tuve que tomar en cuenta que es importante evitar la sobrecarga de información en un tablero. Opté por visualizaciones claras y concisas que comunicaran de manera efectiva los patrones y tendencias más importantes sin abrumar al usuario con demasiados detalles.

Aprendí mucho más este fin de semana realizando esta tarea (más que con algunos profesores de la carrera) y eso es algo que atesoraré mucho de esta experiencia. Estoy lista para los desafíos que vengan y para darlo todo si me dan la oportunidad.

Anexos

Capturas de pantalla de la creación del tablero.

La primera decisión del tablero fue crear una segmentación de datos basada en las especies como se muestra en la siguiente figura:

Figura 1. Filtro de segmentación de datos por especie

Posteriormente se eligieron dos de los datos que se consideraron que aportan más y son más relevantes de visualizar que son la distribución de Genero y Estado de vida como se muestran en las figuras 2 y 3 respectivamente.

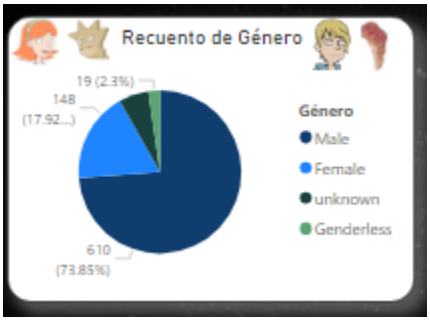

Figura 2. Recuento de genero

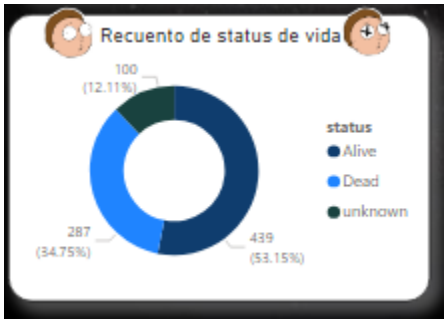

Figura 3. Recuento por status de vida

Un conteo con el total de personajes como se muestra en la figura a continuación es una de las estrellas del tablero, refleja cuantos personajes hay en total según los filtros aplicados.

Figura 4. Cantidad de personajes totales por filtro aplicado

Al tener tantos planetas, universos y personajes de diversas especies como lo es el de Rick and Morty se tomó la decisión de mostrar la relación entre estos. En la figura 5 se puede observar en el eje Y las ubicaciones con más especies, se utilizo un filtro para mostrar solo el top 3 de ubicaciones donde el recuentro de especies es mayor debido a que existen muchas localizaciones con datos como '1' o '0' lo cual implicaba un impedimento de una correcta visualización de los datos.

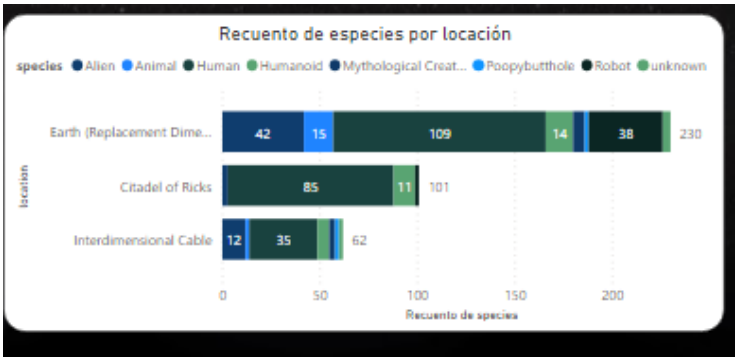

Figura 5. Top 3 del recuento de especies por locación

Un caso similar al anterior se puede observar en la figura 6 donde se muestra el top 3 de tipos de especie más relevante para cada especie.

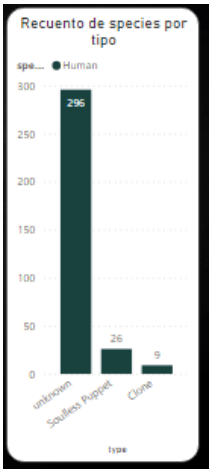

Figura 6. Top 3 del recuento de especies por tipo

Por último, se tiene el gráfico de fechas en las que se originaron los personajes, mostrando cuantos personajes fueron introducidos por año al universo de Rick y Morty como se muestra en la siguiente figura.

Figura 7. Recuento de personajes por fecha de introducción

Por último y más importante, las imágenes del tablero concluido ✨💥🎉

Figura 8. Tablero de visualización en Power BI

Ana María Acosta Pacheco

En la siguientes figuras se mostrarán ejemplos de la visualización de los datos al hacer uso de la parte interactiva del tablero.

Ejemplo 1: Se selección el botón de ‘Human’ para ver un panorama general de la raza humana

Ejemplo 2: Selecciónd especie ‘Alien’ y personajes de genero ‘Female’

Ejemplo 3: Selección de especie ‘Robot’ y estado ‘Muerto’:

Ejemplo 4: Selección de especie ‘Disease’ y año 2017

Código fuente del programa escrito en Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Aug  6 15:43:52 2023

@author: ana
"""

import pandas as pd
import requests
import json
from datetime import datetime

#Funcion para imprimir el objeto JSON y tener mejor visualización
def jprint(obj):
    #Crea una cadeda con formato del objeto JSON
    text = json.dumps(obj, sort_keys=True, indent=4)
    print(text)

#Funcion para hacer la solicitud de consulta 'request' de los datos
def get_characters():
    #URL de la API RickAndMorty
    url = "https://rickandmortyapi.com/api/character"
    #Primera página de personajes
    page = 1
    #Arreglo para de personajes
    characters = []
    #Ciclo para recuperar todas las paginas donde se encuentran los tados de
los personajes
    while page:
        # Hacer la solicitud a la API
        response = requests.get(url, params={'page': page})

        # Verificar el código de respuesta
        if response.status_code == 200:
            #obtener la respuesta de la API en formato json
            data = response.json()
            #Agregar los personajes de la página actual al total de
personajes
            newCharacters = data["results"]
            characters.extend(newCharacters)

            # Pasar a la siguiente página si es que existe
            if data["info"]["next"]:
```

```
                page += 1
            else:

                page = None
        #Manda mensaje del error presentado en caso que la solicitud no haya
sido exitosa
        else:
            print("Error: ", response.status_code)
            return []
    #Visualizacion de los datos de los personajes obtenidos
    jprint(characters)
    return characters

# Función para extraer el año de una fecha
def extract_year_from_date(date_str):
    date_obj = datetime.strptime(date_str, '%Y-%m-%dT%H:%M:%S.%fZ')
    return date_obj.year


#Se llama la funcion para obtener los datos de los personajes
characters = get_characters()

#Los datos obtenidos se colocan en un data frame de pandas
df = pd.DataFrame(characters)
print(df.shape)
#Se observan los nombres de las columnas del dataframe
#print(df.dtypes)

#Se sumarizan cuantos campos hay con datos faltantes
missing_values = df.isnull().sum()
# Imprimir el recuento de valores nulos en cada columna
print(missing_values)

#Se extraen solamente los datos que se consideran relevantes  para el
analisis de datos
df = df[["status", "species", "type", "gender", "location","created"]]

#Se extrae solo el nombre del objeto de 'localización', sus demas campos no
se consideran relevantes
df["location"] = df["location"].apply(lambda x: x["name"])
#Se extrae solo el año del objeto de 'created', sus demas campos no se
consideran relevantes
df['created'] = df['created'].apply(extract_year_from_date)

# Reemplazar valores de cadena vacia ('') del DataFrame por "Unknown"
```

```
df.replace('', 'unknown', inplace=True)

#Se guarda el dataFrame procesado en un archivo csv
df.to_csv('rick_and_morty_characters.csv', index=False)
```