**Proyecto de Ingesta y Clasificación de Artículos de SpaceflightNews**

Este proyecto permite la extracción, procesamiento, almacenamiento, clasificación de artículos, extracción de entidades (compañías, personas, lugares), extracción de palabras claves del API SpaceflightNews todo esto mediante la implementación de un procesamiento de lenguaje natural orquestado en un flujo de datos en Databricks el cual esta orquestado por Apache Airflow mediante un DAG el cual desencadenara un Job para el procesamiento en Databricks

Este proyecto permite una solución automatizada y escalable para manejar grandes volúmenes de datos provenientes del api de SpaceflightNews esto mediante la implementación de tecnologías como:

* Python
* NLP (Procesamiento de Lenguaje Natural)
* Apache Airflow
* Apache Spark (Databricks)
* Unity Catalog
* Datalake

Este proyecto facilitará el manejo de volúmenes de información del API de SpaceflightNews mediante una arquitectura distribuida garantizando seguridad en el proceso y un DataGovernanza atravez de unity catalog y el Datalake utilizado de los datos utilizados.

La solución del presente proyecto esta orquestada por la tecnología de apache Airflow el cual es el encargado de orquestar las diferentes tareas en el DAG, con el fin de realizar un procesamiento eficiente y monitoreado.

**Pruebas Unitarias:**

No se realiza proceso de pruebas unitarias por falta de conocimiento técnico de como deben desarrollarse y que deben incluir para obtener una aprobación cercana al 80%

**Scripts SQL:**

Los scripts de SQL que se aplican al DataWareHouse están incluidos en el notebook de Spark, de igual manera se agrega en el presente documento:

* Tendencia de temas por mes

SELECT

    DATE\_TRUNC('month', df.published\_at) AS month,

    c.category,

    COUNT(\*) AS tendencia\_count

FROM df\_final df

JOIN autor.autors.category c ON (df.category = c.category)

GROUP BY month, c.category

ORDER BY month, tendencia\_count DESC;

* Fuentes mas influyentes

%sql

SELECT

s.news\_site,

COUNT(\*) AS count\_sources

FROM df\_final df

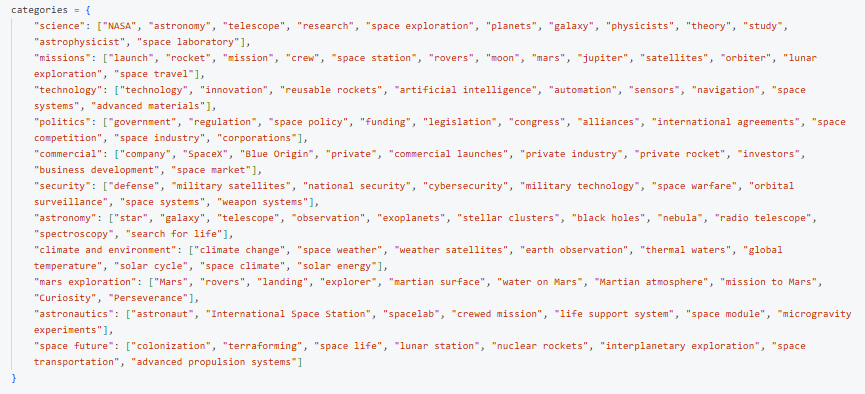
JOIN autor.autors.site s ON (df.news\_site = s.news\_site)

GROUP BY s.news\_site

ORDER BY count\_sources DESC;

**Uso de modelos LLM en el desarrollo de la prueba**

En la presente prueba se utilizo el modelo de openai gpt-4º, esto en los siguientes procesos:

* Análisis del Api de SpaceflightNews para la extracción del catálogo de categorías para el proceso de clasificación de los artículos.   
  
* Aplanado de campos tipo Array en un dataframe con el fin de extraer el nombre del autor de cada articulo:



* Implementación de la librería de logging en Python:  
  