

**Procesamiento de datos del api de SpaceflightNews:**

El presente documento describe el proceso implementado para la extracción, procesamiento almacenamiento, modelado de Datawarehouse de los datos obtenidos del api de SpaceflightNews. Todo este proceso esta orquestad mediante Apache Airflow y culmina con el poblado de las dimensiones y tabla de hechos utilizando el servicio de RDS (MySQL).

**Arquitectura del proceso ETL implementado:**

El proceso implementado es orquestado a través de un DAG en pache Airflow llamado DAG\_INETUM\_ETL, el cual sigue la siguiente secuencia de tareas:

Start >> extract\_data >> preprocess\_data\_task >> load\_data\_task >> clear\_data\_task>> submit\_glue\_crwal >> submit\_glue\_job >> end

Cada tarea contiene un propósito clave los cuales son:

* Extract\_data: este proceso se encarga de la extracción de los datos mediante la clase ExtractAPI el cual se encarga de hacer la petición al api de SpaceflightNews.
* preprocess\_data\_task: Esta tarea se encarga de crear un nuevo campo “type” en los articulo extraído con el fin de identificar el tipo de documento que se va procesar si es article, blog o info extraído del API. Además, se encarga de la eliminación de los datos duplicados de los artículos extraídos esto mediante la implementación de la librería pandas, donde los datos extraídos son convertidos en un dataframe y luego se utiliza la función drop\_duplicates el cual elimina los datos duplicados. También esta tarea se encarga de almacenar en un archivo temporal los datos extraídos en formato json.
* load\_data\_task: Esta tarea es la encargada leer el archivo json que fue almacenados en el archivo temporal y seguido a esto se procede a almacenar los datos en un bucket de S3.
* clear\_data\_task: Este proceso se encarga de limpiar los archivos temporales utilizados en apache Airflow en las tareas anteriores
* submit\_glue\_crwal: Una vez los archivos son cargados al bucket de S3, esta tarea se encarga de ejecutar un crawler el cual rastrea la metadata de los archivos almacenados en el bucket de S3, una vez esta metadata es extraida se almacena n la base de datos del datacatog de AWS Glue, este proceso es invocado desde el operador de apach Airflow GlueCrawlerOperator
* submit\_glue\_job : Esta tarea se encarga de invocar el job que fue construido en la plataforma de AWS Glue, mediante el operador GlueJobOperator. Donde este Job se encarga de invocar un proceso ETL el cual es el encargado de leer los datos almacenados en el bucket de S3, esto mediante la metadata extraída por el crawler y almacenada en el datacatalog. Seguido a esto se realiza el procesamiento de la data donde se extraen palabras claves, clasifica el texto y se extraen las dimensiones de cada artículo, se realiza un proceso de limpieza de data y por ultimo se extrae ñla información que va a poblar las dimensiones de autor, category, site y el poblado de la tabla de hechos, generando así un Datawarhouse con esquema estrella

**Proceso Glue:**

El Job está construido en el servicio de AWS Glue el cual esta construido sobre apache Spark. Este Job se encarga de invocar un script creado en Spark el cual se encarga de:

* Lectura de los datos que están almacenados en un formato json en el bucket de S3, Seguido a esto los datos son leídos en un formato DynamicDatframe y se convierte en un dataframe de Spark.
* Se realiza un aplanado de los datos, donde se aplanan los datos que se encuentra en formato array dentro del dataframe leído, este aplanamiento consiste en extraer en columnas separadas aquellos datos que están dentro de un array, en nuestro caso el campo autor es un array el cual almacena el nombre del autor y las redes sociales del autor con este aplanamiento el nombre del autor queda en un campo separado.
* Procesamiento nlp, se realiza la implementación de la librería de spacy la cual se encarga atravez de una serie de funciones de: la clasificación de los artículos, extracción de palabras claves y extracción de entidades.

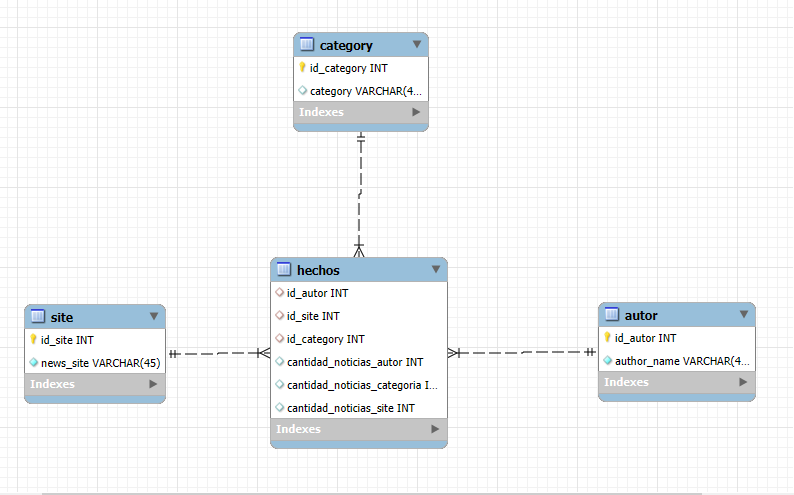
Esta librería al no ser nativa de Glue se instala en el job desencadenador.

* Generación de un dataframe limpio, realizado el procesamiento nlp se procede a la construcción de un dataframe limpio y consolidado con la información relevante para el modelado del Datawarehouse.
* Obtenido el dataframe final se procede al poblado de las dimensiones y tabla de hechos teniendo como base el dataframe final que cuenta con una data limpia y estructurada.

**Tecnologías implementadas:**

* Lenguaje de programación Python
* Apache Airflow: orquestador de flujos de trabajo
* Apache Spark en Aws Glue: Sistema de procesamiento de datos
* Bucket de S3: Capa de almacenamiento de data estructurada, semiestructurada y no estructurada
* RDS: Capa de almacenamiento y modelado para el Datawarehouse
* Spacy: Librería utilizada para el procesamiento de lenguaje natural(NLP)

**Esquema Datawarehouse:**



En el modelamiento del Datawarehouse se implementa un modelo estrella el cual cuenta con 3 dimensiones site, category y autor. De igual manera se implementa una tabla de hechos la cual contiene las medidas de cantidad de noticias por autor, cantidad de noticias por categoría y cantidad de noticias por site, estas métricas se plantean teniendo en cuenta los tipos de datos de relevancia extraídos del API

Este modelado se realiza sobre una RDS MySQL, en la cual se optimiza el procesamiento mediante la implementación de índices en las llaves primarias de las dimensiones y las llaves foráneas de la tabla de hechos.