Documentación Funcional

Ingesta de Datos de Ligas de Fútbol

Bootcamp de Ingeniería de Datos de Código Facilito

Esta documentación está pensada para:

- El cliente
- El usuario en general.

Autor / Data Engineer : Nahuel Raúl Lopez

Para mayor información técnica, ver el README en el repositorio de GitHub:
https://github.com/Nahuel-DevOne/data-engineering-cf

2024

Proyecto Final
Ingeniería de datos

Bootcamp de Código Facilito

Índice

3. Información del proyecto
4. Desarrollo del proyecto
5. Tecnologías y configuraciones
6. Extracción y transformación
7. Carga y configuración
8. Resultado final
9. Agradecimientos y contacto

Información del proyecto

El proyecto es un ETL (Extract, Transform, Load por sus siglas en inglés) que consiste en extraer, transformar y cargar datos de distintas ligas de fútbol de Europa, dejando data disponible para un posterior trabajo de analítica de datos. No es la intención de este proyecto resolver una problemática en específico, sino que el fin es realizar este proceso llevando a cabo las mejores practicas de Ingeniería de datos utilizando varias de las herramientas y tecnologías vistas en el bootcamp, mostrando la vital importancia del rol de un ingeniero de datos para cualquier trabajo posterior de data.
Se muestra a continuación los pasos del ETL sobre los que se apoya el proyecto, desde el inicio hasta el final:

Proceso de ETL

Extracción

Se utiliza Pandas, una librería de Python, para la extracción de los datos de distintas páginas web

Transformación

También se utiliza Pandas para la transformación de los datos.

Carga

Se orquesta el proceso con Airflow, para llevar los datos limpios y transformados a una tabla en Snowflake. En este punto los datos están listos para cualquier trabajo posterior de analítica de datos.

Desarrollo del proyecto

Si bien el proceso de ingeniería de datos se refiere y da lugar a la importancia del ETL, para lograr tener la data disponible y poder posteriormente trabajar cualquier caso de analítica, existen otros desarrollos e implementaciones para que esto pueda llevarse a cabo.
Por empezar, este proyecto utiliza WSL (Windows Subsystem for Linux) para correr un entorno de linux en windows que permite trabajar con varias de las tecnologías utilizadas de una forma muy eficiente.
A la vez se utiliza Docker (una tecnología de contenedores que encapsula y hace portable los códigos a cualquier otra máquina) en conjunto con Astro CLI para levantar y correr el servicio de Airflow (el orquestador que lleva el manejo de los procesos del ETL). Este orquestador, una vez configurado, es responsable de automatizar el proceso de extracción de los datos en distintas páginas web, de la transformación de los mismos con Pandas y de la carga de los datos limpios y transformados a una tabla de Snowflake, donde la data queda finalmente disponible.

Proceso de desarrollo

Tecnologías y configuraciones

Para poder iniciar este proyecto se requiere de WSL que se instala por terminal. Este sirve para correr un entorno de Linux en Windows, que permite trabajar distintos procesos y hacer instalaciones de una forma mucho más eficiente, teniendo en cuenta que la mayoría de los servidores del mundo utilizan Linux.
Al contar con WSL, se instala Astro CLI por terminal (en mi caso usé la distro de Ubuntu). A su vez, es necesario contar con Docker que en conjunto con Astro CLI va a permitir levantar y correr la UI (User Interface) de Airflow con unos sencillos comandos también por terminal.
Ya levantado y en ejecución, se configura la automatización de los distintos procesos que se van a tener en cuenta para el ETL. Esto permite que un proceso manual que uno debería estar haciendo reiteradas veces para actualizar la información disponible en Snowflake (nuestro almacén de datos), se haga automáticamente con Airflow, que vela por repetir estos procesos que hacemos una vez pero buscamos automatizar.
Para poder usar Snowflake, es necesario contar con una cuenta en la misma plataforma.

WSL
Permite instalar fácilmente Astro CLI por terminal, lo que agiliza el desarrollo.

Astro CLI
Permite levantar y correr en conjunto con Docker, el orquestador Airflow.

Docker
Es necesario para poder levantar una imagen de Airflow.

Snowflake
El almacén de los datos, donde se alojarán limpios y transformados

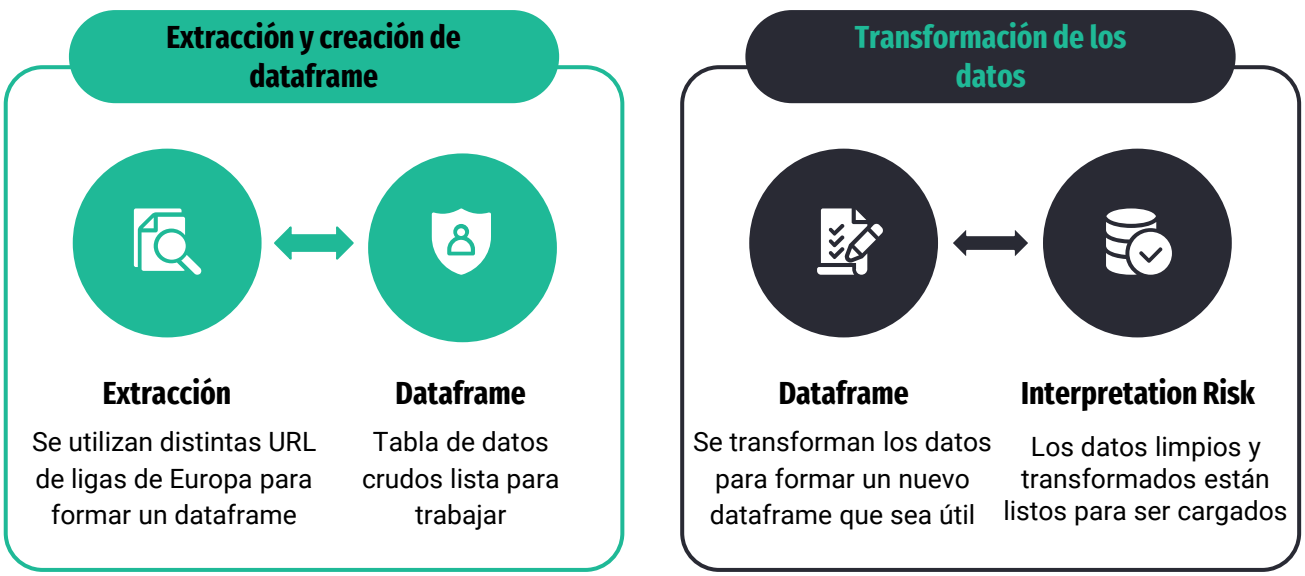
Extracción

Utilizando Python y la librería de Pandas, se genera en un notebook, una tabla de datos (dataframe) con los datos de las ligas de Europa, usando la URL de cada liga y de su respectivo país.
De esta manera, se extraen los datos en un notebook de forma que luego se pueden trabajar para ser transformados.

Transformación

Ya cargado el dataframe, con los datos en crudo, se limpia y se transforma en datos útiles con Python y Pandas.
Se realizar un proceso llamado feature Engineering (o ingeniería de variables) para dejar los datos en el formato que se necesitan para ser cargados.
Una vez realizado esto, los datos se encuentran listos para la carga en Snowflake.

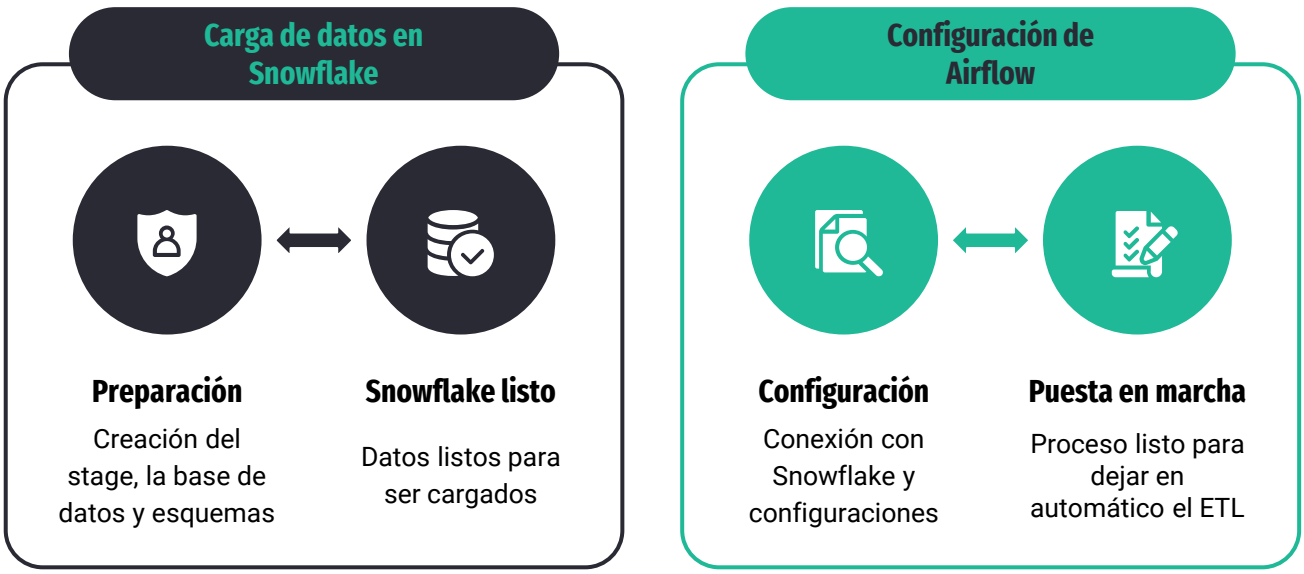
Proceso de extracción y transformación



Carga y configuración

Se trabaja en el almacén de los datos, Snowflake. Para ello se crea el stage (que es un área de almacenamiento temporal para archivos de datos antes de cargarlos en tablas) donde se pasa la base de datos con sus esquemas.
Luego se crea la tabla de fútbol “football_leagues”.
En este punto Airflow está funcional y listo para conectarse con Snowflake.
Se realiza la configuración de conexión con Snowflake y se generan las variables de entorno para proteger los datos sensibles de la misma.
A su vez, se configura la automatización, para repetir este proceso de ETL algunas veces en la semana.

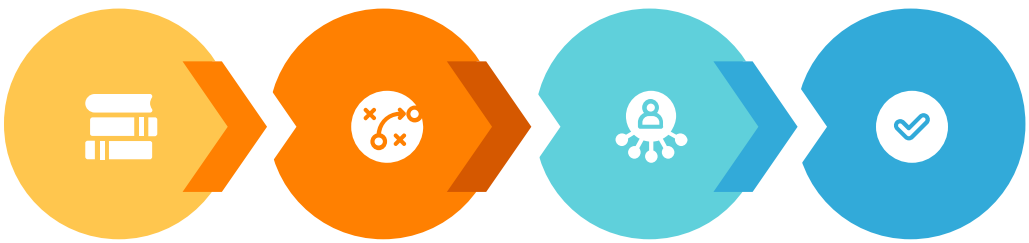
Carga de datos y configuración de Airflow



Resultado final

Al finalizar el proceso de ETL, quedan cargados en una tabla de Snowflake todos los datos limpios y transformados listos para ser utilizados.
La ventaja de estar usando Airflow para orquestar el proceso, es que este trabajo queda automatizado y los datos en Snowflake van a ser actualizados periódicamente, según como se haya configurado.
Esto se supone fundamental, en un contexto donde es necesario contar con los datos los más actualizados posibles, muchas veces para informes y toma de decisiones.

El proceso que lleva a la data final



Todas las instalaciones y configuraciones necesarias para poder trabajar.

Extracción de los datos con un script de Python utilizando la librería de Pandas.

Transformación de los datos con un script de Python, utilizando Pandas.

Carga de los datos limpios y transformados en Snowflake, nuestro Datawarehouse.

Agradecimientos y contacto

Agradecimientos

En especial, agradecimiento a Código Facilito por la excelente experiencia del bootcamp, la entrega y el compromiso permanente con toda su comunidad, que se extiende más allá del ámbito académico y son un ejemplo de plataforma educativa a nivel internacional.
Agradecimientos al líder del bootcamp Sergio Sánchez por diseñar los contenidos, y a todos los profesores y el equipo que conforman Código facilito, brindando su apoyo y conocimiento para lograr la mejor experiencia educativa.

Sobre el autor

Para conocerme más o contactarme, puedes visitar mi **Linktree** y buscarme en mis redes. ¡Siéntete libre de ponerte en contacto!

- Soy de Buenos Aires (Argentina).
- Estudio Ingeniería en sistemas.
- Me gusta leer y estudiar sobre diversos temas, explorar nuevas tecnologías y resolver problemas.
- Me desempeño como Data Engineer en NTT Data, pero en mi trabajo diario hago tanto ingeniería de datos como ciencia de datos y machine learning.
- Tengo formación en desarrollo full Stack con JavaScript y Python, pero por mi trabajo utilizo diariamente tecnologías orientadas para data como Python, numpy, pandas, Scikit learn, SQL, tecnologías de Big Data como PySpark y Microsoft Azure para trabajar en la nube, entre otras.

Desarrollado con 💚 por **Nahuel DevOne**