

Documentación del Proyecto ETL con CSV de Nvidia

Tu Nombre

28 de agosto de 2025

Índice general

| | | |
|----------|--|----------|
| 1 | Introducción | 2 |
| 2 | Dataset de Nvidia | 2 |
| 2.1 | Descripción | 2 |
| 2.2 | Columnas del dataset | 2 |
| 2.3 | Ejemplo de filas | 2 |
| 3 | Herramientas utilizadas | 2 |
| 4 | Estructura del proyecto | 3 |
| 5 | Flujo ETL | 3 |
| 5.1 | Extract | 3 |
| 5.2 | Transform | 3 |
| 5.3 | Load | 3 |
| 6 | Prompts de ChatGPT más relevantes | 3 |
| 7 | Resultados | 4 |
| 8 | Conclusiones | 4 |

1 Introducción

El presente proyecto consiste en un flujo ETL (**Extract, Transform, Load**) para procesar datos históricos de acciones de Nvidia. El objetivo principal fue leer un archivo CSV, limpiar y transformar los datos, y finalmente generar un nuevo CSV listo para análisis posteriores.

2 Dataset de Nvidia

El proyecto utiliza un conjunto de datos históricos de las acciones de Nvidia.¹

2.1 Descripción

Este dataset contiene información diaria de las acciones de Nvidia desde 1999 hasta 2024. Es útil para análisis financieros, predicciones de precios y estudios de tendencias del mercado.

2.2 Columnas del dataset

- **Date:** Fecha de la transacción.
- **Open:** Precio de apertura de la acción.
- **High:** Precio máximo alcanzado durante el día.
- **Low:** Precio mínimo alcanzado durante el día.
- **Close:** Precio de cierre de la acción.
- **Adj Close:** Precio de cierre ajustado por dividendos y splits.
- **Volume:** Número de acciones negociadas.

2.3 Ejemplo de filas

| Date | Open | High | Low | Close | Adj Close | Volume |
|------------|--------|--------|--------|--------|-----------|--------|
| 2024-12-31 | 150.00 | 155.00 | 149.00 | 153.00 | 153.00 | 200000 |
| 2024-12-30 | 148.50 | 151.00 | 147.00 | 150.50 | 150.50 | 180000 |

3 Herramientas utilizadas

- **Python 3.12** para la implementación del flujo ETL.
- **Pandas** para manipulación y transformación de datos.
- **Vim** como editor de código.
- **Git y GitHub** para control de versiones y gestión de ramas (main, develop, release).
- **ChatGPT** para apoyo en la programación, depuración de errores y recomendaciones de flujo ETL.

¹<https://www.kaggle.com/datasets/ranugadisansagamage/nvidia-stock>

4 Estructura del proyecto

- Config/config.py
- Extract/extractor.py
- Extract/Files/Nvidia.csv
- Transform/transformer.py
- Load/loader.py
- main.py

5 Flujo ETL

5.1 Extract

Se implementó la clase `Extractor` en `Extract/extractor.py`, que lee el CSV original (`Nvidia.csv`) y retorna un `DataFrame` de `Pandas`.

5.2 Transform

La clase `Transformer` en `Transform/transformer.py` realiza:

- Conversión de la columna `Date` a tipo `datetime64[ns]`.
- Ordenación por fecha.
- Relleno de valores nulos.
- Cálculo de nuevas columnas:

- `Daily Change = Close - Open`
- `Percent Change = $\frac{Close - Open}{Open} \times 100$`

5.3 Load

La clase `Loader` en `Load/loader.py` guarda el `DataFrame` final en un nuevo CSV llamado `Nvidia_clean.csv` en la carpeta `Extract/Files/`.

6 Prompts de ChatGPT más relevantes

- Depurar errores de rutas y lectura de CSV.
- Ajustar transformaciones de `pandas` para evitar errores con `datetime64`.
- Comandos de `Git` para el trabajo con ramas (`main`, `develop`, `release`).

7 Resultados

Al ejecutar el script principal:

```
python3 main.py
```

Se obtiene:

- Archivo limpio: Extract/Files/Nvidia_clean.csv
- Columnas originales más Daily Change y Percent Change
- 5872 filas procesadas correctamente

8 Conclusiones

El proyecto demuestra cómo construir un flujo ETL modular y reproducible usando Python. El uso de ChatGPT facilitó la depuración y la optimización de rutas y transformaciones. La gestión de ramas en Git permite un desarrollo ordenado y preparación de releases.