

Reporte de Laboratorio Nro.1

Jhandry Zambrano^{L00380367}

Universidad de las Fuerzas Armadas
jvzambrano3@dominio.ec

Tema: ETL (archivos planos)

Resumen

En el siguiente reporte de laboratorio se llevo a cabo un ciclo ETL. Se implemento conocimientos teóricos y prácticos donde se hace uso de Python para archivos planos. Este proceso nos permite mover datos de varias fuentes, formatearlos, limpiarlos y finalmente cargarlos. Se pretende aprender cada una de las funcionalidades del proceso ETL. Para verificar que los funciones y comandos funcionan se comprueba cuando se extraen los documentos. En contexto se va a realizar un análisis enfocado en los pasos de un proceso ETL. En efecto se investiga acerca de la carga incremental y su importancia. También se aprende acerca del mapeo en ETL. Y finalmente la funcionalidad y uso de librerías que se usan para desarrollar uno de los procesos ETL, es decir el funcionamiento de cada librería usada.

1. Introducción

En el presente laboratorio se va a realizar el proceso ETL donde conlleva extraer, transformar o limpiar y cargar o entregar. El objetivo de ETL es extraer los datos de varios sistemas transformarlos y cargarlos acorde a las necesidades de la empresa en algún data warehouse. Esto es muy importante aprender porque se aplica cuando una empresa cambia de sistema. En consecuencia, al almacenar los datos aporta a la toma de decisiones.

El proceso de extracción es donde se puede cargar los datos. La transformación es el segundo paso donde se ajustan los datos acordes a las necesidades estableciendo reglas que mejoran la calidad de los datos. Y la carga de datos, es el ultimo paso donde se establece el destino de los datos sólidos. Por lo tanto, aprender ETL no es cuestión de solamente ejecutar código sino saber lo que se está realizando realmente .

Varias empresas hacen uso de ETL, esto se da por los cambios que con el transcurso del tiempo la empresa necesita. Usado para migrar desde sistemas obsoletos hacia los actualizados. Esto puede incluir una gran ventaja a la empresa ya que le permite saber algunas falencias y poder llegar a la toma de decisiones. Esto le permite solventar problemas optimizando tiempo y ganancias cumpliéndose los objetivos de la empresa.

2. Método

Ciclo ETL

ETL es el proceso que permite extraer, transformar y cargar datos de un lugar a otro. Lo que este proceso hace es tomar los datos, transformarlos, limpiarlos y establecerlos en algún otro lugar. ETL es usado por empresas para migrar de sistemas o posteriormente establecer una toma de decisiones que permita realizar mejoras a la empresa. Para empezar con este proceso primero importamos cada una de las funciones y módulos que requiere como se muestra en la figura 10.

```
[1]: import glob
import pandas as pd
import xml.etree.ElementTree as ET
from datetime import datetime
```

Figura 1: Importación de librerías [?].

En consecuencia se hace uso del comando `extract` para extraer los datos de varias fuentes. Esta función en sí permite cargar los archivos esto se realiza acorde a los datos que se vayan a extraer. Hacemos uso de CSV, JSON, XML. Esto se lo realiza con el fin de tener listos nuestros datos para pasar al siguiente paso el cual es transformar. Recordar que el orden de cada fila es el mismo en el que se agregó las filas a los datos.

```
def extract_from_csv(file_to_process):
    dataframe = pd.read_csv(file_to_process)
    return dataframe

def extract_from_csv(file_to_process):
    dataframe = pd.read_csv(file_to_process)
    return dataframe

def extract_from_json(file_to_process):
    dataframe = pd.read_json(file_to_process, lines=True)
    return dataframe

def extract_from_xml(file_to_process):
    dataframe = pd.DataFrame(columns=['car_model', 'year_of_manufacture', 'price', 'fuel'])
    tree = ET.parse(file_to_process)
    root = tree.getroot()
    for person in root:
        car_model = person.find("car_model").text
        year_of_manufacture = int(person.find("year_of_manufacture").text)
        price = float(person.find("price").text)
        fuel = person.find("fuel").text
        dataframe.append([car_model, year_of_manufacture, price, fuel])
    return dataframe
```

Figura 2: Proceso de extracción de datos.

Una vez extraído los datos de varias fuentes llamamos a la función. Para realizar esto se llama a la función CSV, JSON, XML. A continuación se muestra el llamado en la figura 3.

```
def extract():
    extracted_data = pd.DataFrame(columns=['car_model', 'year_of_manufacture', 'price', 'fuel'])
    #for csv files
    for csvfile in glob.glob("dealership_data/*.csv"):
        extracted_data = extracted_data.append(extract_from_csv(csvfile), ignore_index=True)
    #for json files
    for jsonfile in glob.glob("dealership_data/*.json"):
        extracted_data = extracted_data.append(extract_from_json(jsonfile), ignore_index=True)
    #for xml files
    for xmlfile in glob.glob("dealership_data/*.xml"):
        extracted_data = extracted_data.append(extract_from_xml(xmlfile), ignore_index=True)
    return extracted_data
```

Figura 3: `Extract function()`.

Ahora se realiza el proceso de transformación de la fase. El uso de la función `transform` lo que realmente hace es transformar o convertir las columnas después devuelva los resultados modificado de las variables. Esto permite ver que convierte la columna en metros por ende redondeará hasta 2 decimales. A continuación se muestra los comandos ejecutándose.

```
def transform(data):
    data['price'] = round(data.price, 2)
    return data
```

Figura 4: Uso de la función transformar.

Después de haber cargado y modificado mediante la transformación de datos pasamos a la fase de carga y registro. Para cargar los datos se debe establecer la entrada del registro. Para ello se escribirá una función de registro

```
def load(targetfile, data_to_load):
    data_to_load.to_csv(targetfile)
```

Figura 5: Cargar y registro.

Se puede saber en que tiempo se adjunta, es decir en que momento inicia y finaliza. Después de haber establecido el código para que realice todos los procesos se llama a todas las funciones. A continuación se muestra en la siguiente figura

```
def log(logfile, message):
    timestamp_format = '%H:%M:%S-%h-%d-%Y'
    #Hour-Minute-Second-MonthName-Day-Year
    now = datetime.now() # get current timestamp
    timestamp = now.strftime(timestamp_format)
    with open(logfile, "a") as f:
        f.write('[' + timestamp + ']: ' + message + '\n')
    print(message)
```

Figura 6: Uso de la función log.

Para la ejecución del proceso ETL realizamos un llamado a la función. Los datos se transfieren y se cargan en el destino. Por lo tanto se debe tener en cuenta que antes y después se agrega la hora, inicio y finalización.

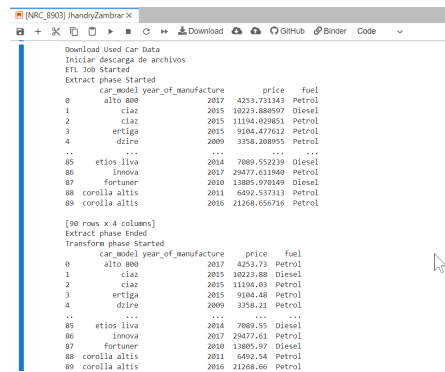
```
logfile = "dealership_logfile.txt" # all event logs will be stored
targetfile = "dealership_transformed_data.csv" # transformed data is stored

log(logfile, "Download Used Car Data")
get_data()
log(logfile, "ETL Job Started")
log(logfile, "Extract phase Started")
extracted_data = extract()
print(extracted_data)
#
log(logfile, "Extract phase Ended")
log(logfile, "Transform phase Started")
transformed_data = transform(extracted_data)
print(transformed_data)
log(logfile, "Transform phase Ended")
log(logfile, "Load phase Started")
load(targetfile, transformed_data)
log(logfile, "Load phase Ended")
log(logfile, "ETL Job Started")
```

Figura 7: Ejecución del proceso ETL.

3. Results and Analysis

El proceso de extracción, transformación y carga se ejecuto en el metodo donde secuencialmente se establece el proceso. Al ir ejecutando cada uno de los procesos podemos darnos cuenta prácticamente el orden en que se maneja el algoritmo. Por lo tanto, a continuación tenemos detalladamente la ejecución ETL, donde nos muestra un bosquejo claro de lo que realizó el algoritmo. A continuación como se muestra en la figura 8.



```
[NRC_8903] handyZambrax
Download used Car Data
Iniciar descarga de archivos
ETL Job Started
Extract phase Started
car_model year_of_manufacture price fuel
0 alto 800 2017 4253.73 Petrol
1 ciaz 2015 10223.00 Diesel
2 ciaz 2015 11194.03 Petrol
3 ertiga 2015 9104.48 Petrol
4 dfire 2009 3358.21 Petrol
.. ..
85 etios livia 2014 7889.55 Diesel
86 innova 2017 29477.61 Petrol
87 fortuner 2010 13805.97 Diesel
88 corolla altis 2011 6402.54 Petrol
89 corolla altis 2016 21268.66 Petrol

[90 rows x 4 columns]
Extract phase Ended
Transform phase Started
car_model year_of_manufacture price fuel
0 alto 800 2017 4253.73 Petrol
1 ciaz 2015 10223.00 Diesel
2 ciaz 2015 11194.03 Petrol
3 ertiga 2015 9104.48 Petrol
4 dfire 2009 3358.21 Petrol
.. ..
85 etios livia 2014 7889.55 Diesel
86 innova 2017 29477.61 Petrol
87 fortuner 2010 13805.97 Diesel
88 corolla altis 2011 6402.54 Petrol
89 corolla altis 2016 21268.66 Petrol
```

Figura 8: Ejecución del proceso ETL.

Al lograr extraer los datos prodemos ver el proceso de carga. Este se guarda en el origen del destino. Este archivo es un archivo de tipo csv. A continuación se muestra los datos generados en la figura 9.

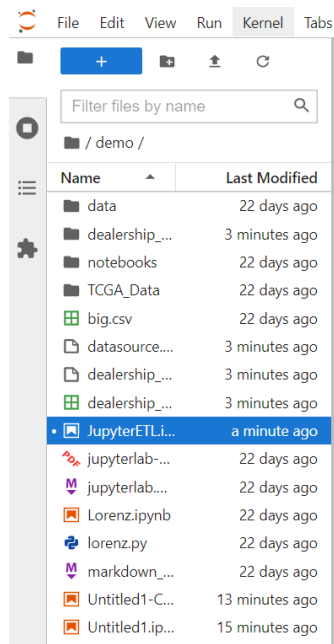
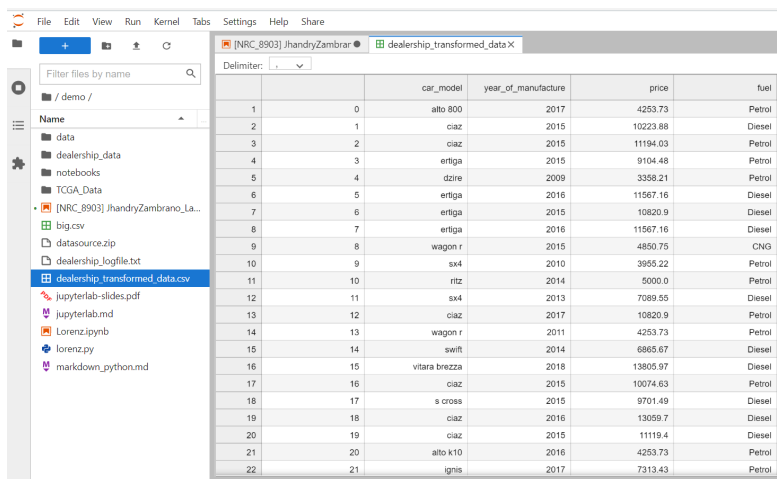


Figura 9: Proceso ETL generado.

A continuación se muestra los archivos generados después de la ejecución. Donde muestra realmente que el algoritmo cumple con su función del proceso ETL. Se generó un archivo CSV y uno .db. A continuación se muestra en la figura 10.



		car_model	year_of_manufacture	price	fuel
1	0	alto 800	2017	4253.73	Petrol
2	1	ciaz	2015	10223.88	Diesel
3	2	ciaz	2015	11194.03	Petrol
4	3	ertiga	2015	9104.48	Petrol
5	4	dzire	2009	3358.21	Petrol
6	5	ertiga	2016	11567.16	Diesel
7	6	ertiga	2015	10820.9	Diesel
8	7	ertiga	2016	11567.16	Diesel
9	8	wagon r	2015	4850.75	CNG
10	9	sv4	2010	3955.22	Petrol
11	10	rtz	2014	5000.0	Petrol
12	11	sv4	2013	7089.55	Diesel
13	12	ciaz	2017	10820.9	Petrol
14	13	wagon r	2011	4253.73	Petrol
15	14	swift	2014	6865.67	Diesel
16	15	vitara brezza	2018	13905.97	Diesel
17	16	ciaz	2015	10074.63	Petrol
18	17	s cross	2015	9701.49	Diesel
19	18	ciaz	2016	13059.7	Diesel
20	19	ciaz	2015	11119.4	Diesel
21	20	alto k10	2016	4253.73	Petrol
22	21	ignis	2017	7313.43	Petrol

Figura 10: .

4. Discusión

Un proceso ETL en si, es extraer, transformar y cargar, como muestran sus siglas en ingles. Entonces ETL es una técnica que permite extraer datos de varias fuentes y permite poder transformarlos de forma útil para finalmente cargarlo en otro sistema. Esto se realiza con el objetivo de que sea accesible los datos acordes a las necesidades de la empresa. Además, sirve para que posteriormente la empresa al tener los datos puede establecer la toma de decisiones acorde a los datos obtenidos mediante ETL. Por lo tanto, ETL se basa en 3 pasos principales como muestran sus siglas en ingles (Extract, Transform, Load) [2].

Extraer

Permite extraer los datos del origen. Esta fase puede tener archivos relacionales o no relacionales. Por ende, la extracción convierte los datos a un formato listo para continuar con el siguiente proceso.

Transformar

En esta fase es donde se ejecutan los cambios y transformaciones. Permite seleccionar, traducir códigos, establecer valores libres, unir datos, calcular totales, dividir columnas y muchas mas funcionalidades. Es prácticamente donde se desarrollan todos los cambios que requieren pequeñas manipulaciones.

Cargar

Permite cargar todos los datos ya establecidos en la fase anterior. Carga los datos en los destinatarios [4].

Al hablar de ETL es necesario saber que es cargar incremental. La carga incremental es una técnica que ayuda a optimizar tiempo en la carga de datos modificados. También optimiza los recursos al borrar las cargas de los datos cuando se actualizan en el repositorio. Esto influye en el mapeo de ETL. Es un proceso donde se obtiene los datos, pueden ser uno o varios archivos

del origen. Esto se lo hace coincidir desde su extracción hacia su destino. Es bastante usado en cualquier proceso de datos.

Al desarrollar el laboratorio se hizo uso de varias librerías necesarias para el proceso ETL. Como por ejemplo pandas que permite leer y escribir los datos en varios formatos como, por ejemplo: CSV, Excel, sql entre otros. También permite seleccionar y realizar el filtrado de tablas y demás funcionalidades necesarias al momento de realizar un ETL. Glob permite encontrar las rutas. Request permite hacer peticiones http y demás funcionalidades [3].

5. Conclusión

ETL es uno de los procesos que en sí, permiten mover datos entre los sistemas de una organización o empresa. Su importancia radica en que permite extraer los datos, transformarlos, limpiarlos y finalmente cargarlos.

Se logro verificar como se manejan los procesos ETL. El poder realizar un proceso ETL aporta bastante a nuestro conocimiento debido a que las empresas realiza estos procesos y que mejor que tener noción de como realizarlo. Así se puede llegar a una toma de decisiones, donde se ayude a optimizar recursos y generar ganancias en cualquier empresa.

Para desarrollar este laboratorio se lo realizo mediante el uso de Jupyter notebooks. Esta herramienta nos permitió ver con claridad los procesos realizados. Al igual que diferenciar los procesos de datos de las diferentes librerías de python. Esto permitio satisfacer el proceso ETL y finalmente obtener archivos planos como: csv,xml y json cumpliendo los objetivos principales.

Referencias

- [1] Bioinformatics at COMAV. pandas — bioinformatics at comav 0.1 documentation. <https://bioinf.comav.upv.es/courses/linux/python/pandas.html>, Marzo 2020. (Accessed on 01/12/2022).
- [2] Elena Bello. Guía de procesos etl: Qué son, cómo usarlos y herramientas clave. <https://www.iebschool.com/blog/que-son-los-procesos-etl-big-data/>, Enero 2022. (Accessed on 01/12/2022).
- [3] Juan Lozano. Python requests. la librería para hacer peticiones http en python. <https://j2logo.com/python/python-requests-peticiones-http/>, Septiembre 2018. (Accessed on 01/12/2022).
- [4] wikipedia. Extract, transform and load - wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Extract,_transform_and_load, Noviembre 2012. (Accessed on 01/12/2022).