

Prueba técnica

Sección 1: Procesamiento y transferencia de datos

Objetivo: Crear un proceso con las herramientas disponibles por el usuario

Los ejercicios de programación tienen que incluir los procedimientos de instalación y ejecución de las herramientas a utilizar y los scripts que realizaran los procedimientos. Se puede realizar a través de Docker. Pueden incluir pruebas unitarias o de integración. Se puede compartir por Github o cualquier repositorio o en un zip.

Nota: Junto con esta guía te compartimos 1 data set con la información sobre las compras de dos compañías ficticias que procesan con nosotros.

1.1 Carga de información

La información proporcionada se debe de cargar en alguna base de datos. Puede ser estructurada o no estructurada. Ejemplo: MySQL, Postgres, MongoDB, etc.

Incluye comentarios del por qué elegiste ese tipo de base de datos

Este código se utiliza para gestionar la inserción del dataset (archivo CSV) en la base de datos MySQL. Se emplea MySQL debido a que el dataset proporcionado muestra claramente una relación de muchos a muchos en los datos. Por lo tanto, es más conveniente emplear una base de datos relacional que permita gestionar y mantener de manera eficiente estas relaciones entre los registros.

- *Se utiliza Pandas para leer el archivo CSV, lo que facilita la manipulación y extracción de datos desde un archivo plano.*

- *La conexión con la base de datos MySQL se establece utilizando SQLAlchemy.*

```
PS C:\GITHUB\P_Tecnica_NT> docker exec -it Seccion1_Container bash
root@ab7435120d66:/usr/src/app# cd Seccion1
root@ab7435120d66:/usr/src/app/Seccion1# python 1_Cargar_Informacion.py
Datos insertados en la tabla Cargo.
root@ab7435120d66:/usr/src/app/Seccion1#
```

```
15 |
| 2fa2c8e8d608da51b6eca7674fae189a16b410 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 373.82 | paid | 2019-03-17 | 2019-03-
17 |
| 1b6978412f5c804d6f99ecd513986f33b28ba22c | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 3 | voided | 2019-03-30 | NULL
|
| 4c61cd79789096bfb9a50521eb78de37a490fa5b | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 58.56 | paid | 2019-04-18 | 2019-04-
18 |
| 79d5c18b3a64bb0d1165cac713beac9d2a23ae36 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 164.78 | refunded | 2019-04-12 | 2019-04-
12 |
| ef9ec85887418d2ccf220686312cfc4c615665d5 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 23.05 | paid | 2019-02-20 | 2019-02-
20 |
| 50bb3950ecdce94ca3e07853003a11a413508da36 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 20.5 | paid | 2019-03-19 | 2019-03-
19 |
| 1b2c4e503a4a952064b70369cf8e72937c720474 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 81.42 | paid | 2019-03-10 | 2019-03-
10 |
| 4e85c4eac968c9465fcd34bcd4968ac019fa850 | MiPasajefy | cbf1c8b09cd5b549416d49d220a40cbd317f952e | 33.45 | paid | 2019-03-16 | 2019-03-
17 |
+-----+-----+-----+-----+-----+-----+-----+-----+
---+
```

1.2 Extracción

Se debe de realizar un procedimiento de extracción de la información anterior por medio de algún lenguaje de programación que permita procesarlo. El formato final de la información extraída puede ser CSV, Avro, parquet o el que se considere más adecuado.

Agrega comentarios acerca del por qué tuviste que utilizar el lenguaje y el formato que elegiste. También platicamos si te encontraste con algún reto a la hora de extraer la información.

Este código se utiliza para extraer datos desde una tabla en una base de datos MySQL y exportarlos a un archivo CSV.

Se realiza una consulta sobre la tabla 'Cargo' para recuperar los registros almacenados, luego se convierte estos datos en un DataFrame de Pandas y, finalmente, se exportan a un archivo CSV.

```
Datos extraídos con éxito:
      id      name      company_id  amount  status  created_at  paid_at
0  48ba4bdfb56ceebb32f2bd0263e759be942af3d  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  3.00  voided  2019-03-19  None
1  05fc6f5ac66b6ee7e4253aa5d0c2299eb47aaaf4  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  3.00  pending_payment  2019-05-06  None
2  2cdce231c1fc6a2061bfa2f1d978351fe217245d  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  3.00  voided  2019-02-22  None
3  81633ba310a50b673efd469c37139576982901aa  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  102.61  paid  2019-02-27  2019-02-27
4  6ccfc4c24e788e4bca448df343698782db6b0c0b  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  184.49  paid  2019-02-05  2019-02-05
...      ...      ...      ...      ...      ...      ...
9995  4907c36c037bfa4874047e14f722329016dc3908  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  59.88  paid  2019-03-14  2019-03-14
9996  ef9ec85887418d2ccf220686312cfc4c615665d5  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  23.05  paid  2019-02-20  2019-02-20
9997  50bb3950ecde94ca3e07853003a11a413508da36  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  20.50  paid  2019-03-19  2019-03-19
9998  1b2c4e503a4e952064b70369cf8e72937c720474  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  81.42  paid  2019-03-10  2019-03-10
9999  4e85c4eac968c9465fc8d34bcd4968ac019fa850  MiPasajefy  cbf1c8b09cd5b549416d49d220a40cbd317f952e  33.45  paid  2019-03-16  2019-03-17

[10000 rows x 7 columns]
Datos exportados exitosamente a datos_extraccion.csv.
root@ab7435120d66: /usr/src/app/Seccion1#
```

```
extraccion_datos.py  3_Transformacion_datos.py  datos_transformados.csv  datos_extraccion.csv x
```

```
id,name,company_id,amount,status,created_at,paid_at
48ba4bdfb56ceebb32f2bd0263e759be942af3d,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,3.0,voided,2019-03-19,
05fc6f5ac66b6ee7e4253aa5d0c2299eb47aaaf4,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,3.0,pending_payment,2019-05-06,
2cdce231c1fc6a2061bfa2f1d978351fe217245d,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,3.0,voided,2019-02-22,
81633ba310a50b673efd469c37139576982901aa,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,102.61,paid,2019-02-27,2019-02-27
6ccfc4c24e788e4bca448df343698782db6b0c0b,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,184.49,paid,2019-02-05,2019-02-05
...
4907c36c037bfa4874047e14f722329016dc3908,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,59.88,paid,2019-03-14,2019-03-14
ef9ec85887418d2ccf220686312cfc4c615665d5,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,23.05,paid,2019-02-20,2019-02-20
50bb3950ecde94ca3e07853003a11a413508da36,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,20.50,paid,2019-03-19,2019-03-19
1b2c4e503a4e952064b70369cf8e72937c720474,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,81.42,paid,2019-03-10,2019-03-10
4e85c4eac968c9465fc8d34bcd4968ac019fa850,MiPasajefy,cbf1c8b09cd5b549416d49d220a40cbd317f952e,33.45,paid,2019-03-16,2019-03-17
```

1.3 Transformación

Se propone el siguiente esquema para la información

Cargo
id varchar(24) NOT NULL
company_name varchar(130) NULL
company_id varchar(24) NOT NULL
amount decimal(16,2) NOT NULL
status marchar(30) NOT NULL
created_at timestamp NOT NULL
updated_at timestamp NULL

Realiza las transformaciones necesarias para que la información extraída cumpla con el esquema. Puedes realizarlas con el lenguaje de programación de tu preferencia.

Incluye comentarios acerca de que transformaciones tuviste que realizar y que retos te encontraste en la implementación de estos mecanismos de transformación.

Este script realiza la transformación y ajuste de datos extraídos desde un archivo CSV. Las transformaciones incluyen la generación de nuevos IDs, la limitación de nombres y cantidades, la normalización de estados y fechas, y el manejo de valores nulos. Finalmente, los datos ajustados se exportan a un nuevo archivo CSV.

Estas transformaciones fueron varios retos ya que requirieron un enfoque cuidadoso en la validación y manipulación de datos, combinando lógica flexible con restricciones claras para asegurar que la transformación fuera efectiva y segura (sin eliminar registros).

- `id varchar(24) NOT NULL`
- `company_name varchar(130) NULL`
- `company_id varchar(24) NOT NULL`
- `status varchar(30) NOT NULL`
 - Se encontró, dificultades de que había algunos id en nulos, entonces se tuvo que generar un nuevo id para no perder el registro, también se añadió o se redujo la cantidad de caracteres por id para normalizar, también se verifico si en los ids hay caracteres inválidos.
- `amount decimal(16,2) NOT NULL`
 - Se encontró dificultades de transformación ya que había cantidades muy grandes o con notación científica, por lo tanto se tuvo que reducir estas cantidades en decimales o ponerlo en 0 si no tiene un tamaño suficiente, esto para no perder el registro.
- `created_at timestamp NOT NULL`
- `updated_at timestamp NULL`
 - Con las fechas las dificultades encontradas era darle el formato correcto de timestamp ya que solo había fecha y se tenía que poner la hora.

```
Terminal Local (3) x Local x + ↗
root@ab7435120d66:/usr/src/app/Seccion1# python 3_Transformacion_datos.py
Transformando... Espere un momento.
Datos transformados exportados a datos_transformados.csv.
root@ab7435120d66:/usr/src/app/Seccion1#
```

Extraccion_datos.py 3_Transformacion_datos.py datos_transformados.csv x datos_extraccion.csv

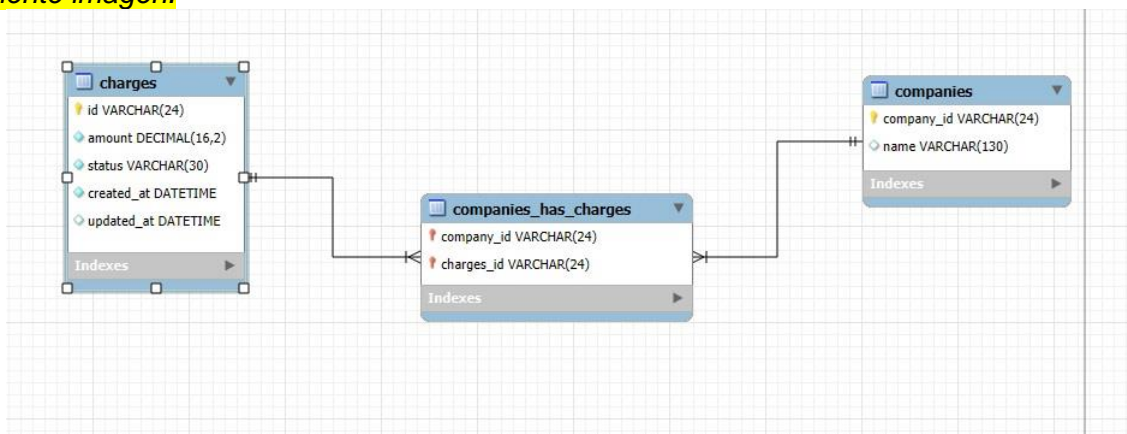
```
1 |id,name,company_id,amount,status,created_at,updated_at
2 |48ba4bdfb56ceebb32f2bd0,MiPasajefy,cbf1c8b09cd5b549416d49d2,3.0,voided,2019-03-19 00:00:00,2025-01-12 02:56:06
3 |05fc6f5ac66b6ee7e4253aa5,MiPasajefy,cbf1c8b09cd5b549416d49d2,3.0,pending_payment,2019-05-06 00:00:00,2025-01-12 02:56:06
4 |2cdce231c1fc6a2061bfa2f1,MiPasajefy,cbf1c8b09cd5b549416d49d2,3.0,voided,2019-02-22 00:00:00,2025-01-12 02:56:06
5 |81633ba310a50b673efd469c,MiPasajefy,cbf1c8b09cd5b549416d49d2,102.61,paid,2019-02-27 00:00:00,2019-02-27 00:00:00
6 |6ccfc4c24e788e4bca448df3,MiPasajefy,cbf1c8b09cd5b549416d49d2,184.49,paid,2019-02-05 00:00:00,2019-02-05 00:00:00
7 |b25f2ff15c24ea881e676be7,MiPasajefy,cbf1c8b09cd5b549416d49d2,85.27,pending_payment,2019-01-04 00:00:00,2025-01-12 02:56:06
8 |4f9db76960dcdf6b7e2d7d71,MiPasajefy,cbf1c8b09cd5b549416d49d2,41.0,pre_authorized,2019-05-04 00:00:00,2025-01-12 02:56:06
9 |7e96211c7d133a6227e809ab,MiPasajefy,cbf1c8b09cd5b549416d49d2,3.0,pending_payment,2019-01-23 00:00:00,2025-01-12 02:56:06
10 |6ee515d5f0a1995731fb3105,MiPasajefy,cbf1c8b09cd5b549416d49d2,41.61,voided,2019-05-03 00:00:00,2025-01-12 02:56:06
```

1.4 Dispersión de la información

Se debe de utilizar una base de datos MySQL, Postgres, MongoDB, etc. En esta base se va a crear un esquema estructurado basado en el ejercicio anterior, pero debemos de crear una tabla llamada charges donde tendremos la información de las transacciones y otra llamada companies donde incluiremos la información de las compañías. Estas tablas deberán de estar relacionadas. Cargaremos la información del dataset en estas dos tablas.

Incluye el diagrama de base de datos resultado de este ejercicio.

Este script realiza la conexión a una base de datos MySQL y lleva a cabo la creación de tablas necesarias para almacenar los cargos, compañías y las relaciones entre estas. Además, se leen datos desde un archivo CSV el cual fue ya previamente transformado (limpiado/normalizado) y se insertan en las tablas correspondientes. El diagrama de la BD se puede observar en el siguiente imagen.



```
Datos insertados en la tabla 'charges'.
Datos insertados en la tabla 'companies'.
Datos insertados en la tabla 'companies_has_charges'.
root@ab7435120d66:/usr/src/app/Seccion1#
```

```
| ffee16f1b4edfb13577d14b8 |      116.13 | paid      | 2019-03-14 00:00:00 | 2019-03-14 00:00:00 |
| recovered_eK00hi30DYAFe5 |      55.71 | paid      | 2019-02-13 00:00:00 | 2019-02-13 00:00:00 |
| recovered_hsp1vajt15470J |      66.16 | pending_payment | 2019-03-14 00:00:00 | 2025-01-12 02:56:06 |
| recovered_lpBJqMnKrmWCDj |      89.36 | paid      | 2019-04-17 00:00:00 | 2019-04-17 00:00:00 |
+-----+-----+-----+-----+-----+
10000 rows in set (0.02 sec)
```

```
+-----+-----+
| company_id | name |
+-----+-----+
| 8f642dc67fccf861548dfe1c | Muebles chidos |
| cbf1c8b09cd5b549416d49d2 | MiPasajefy |
+-----+-----+
```

```
| cbf1c8b09cd5b549416d49d2 | recovered_eK00hi30DYAFe5 |
| cbf1c8b09cd5b549416d49d2 | recovered_hsp1vajt15470J |
| cbf1c8b09cd5b549416d49d2 | recovered_lpBJqMnKrmWCDj |
+-----+-----+
9995 rows in set (0.01 sec)
```

```
mysql>
```

1.5 SQL

Diseña una vista en la base de datos MySQL, Postgres, MongoDB, etc., de las tablas donde cargamos la información transformada para que podamos ver el monto total transaccionado por día para las diferentes compañías

Este script realiza la conexión a una base de datos MySQL y crea una vista llamada 'daily_transaction_summary' que ofrece un resumen diario de las transacciones por compañía. Además, extrae los datos de la vista utilizando Pandas para su visualización en consola.

```
root@ab7435120d66:/usr/src/app/Seccion1# python 5_Vista_SQL.py
Vista 'daily_transaction_summary' creada exitosamente.
Resumen diario de transacciones por compañía:
      company_name transaction_date  total_amount_transacted
0      MiPasajefy      2019-01-01              4150.04
1      MiPasajefy      2019-01-02             16974.89
2  Muebles chidos      2019-01-02               70.03
3      MiPasajefy      2019-01-03             9934.66
4      MiPasajefy      2019-01-04             6349.69
..          ...          ...
199  Muebles chidos      2019-05-17               41.61
200      MiPasajefy      2019-05-18              6148.64
201      MiPasajefy      2019-05-19             5478.52
202  Muebles chidos      2019-05-19               75.48
203      MiPasajefy      2019-05-20             633.54

[204 rows x 3 columns]
root@ab7435120d66:/usr/src/app/Seccion1#
```

```
mysql> SELECT * FROM daily_transaction_summary;
+-----+-----+-----+
| company_name | transaction_date | total_amount_transacted |
+-----+-----+-----+
| MiPasajefy   | 2019-01-01       | 4150.04 |
| MiPasajefy   | 2019-01-02       | 16974.89 |
| Muebles chidos | 2019-01-02       | 70.03 |
| MiPasajefy   | 2019-01-03       | 9934.66 |
| MiPasajefy   | 2019-01-04       | 6349.69 |
| MiPasajefy   | 2019-01-05       | 5052.20 |
| Muebles chidos | 2019-01-05       | 132.77 |
| MiPasajefy   | 2019-01-06       | 3937.31 |
| Muebles chidos | 2019-01-06       | 68.15 |
| MiPasajefy   | 2019-01-07       | 26632.60 |
```

Sección 2: Creación de una API

Objetivo: Creación e implementación de una aplicación

Problema: Calcular el número faltante de un conjunto de los primeros 100 números naturales del cual se extrajo uno.

Especificaciones:

- La aplicación debe implementarse en el lenguaje de acuerdo con el perfil
- Se debe implementar una clase que represente al conjunto de los primeros 100 números
- La clase implementada debe tener el método Extract para extraer un cierto número deseado
- La clase implementada debe poder calcular qué número se extrajo y presentarlo
- Debe incluir validación del input de datos (número, número menor de 100)
- La aplicación debe poder ejecutarse con un argumento introducido por el usuario que haga uso de nuestra clase y muestre qué número pudo calcular que se extrajo ese número