**Proyecto Prueba CUN**

Este documento detalla paso a paso el proceso de construcción de un pipeline ETL y la creación de un tablero de visualización usando datos de COVID-19, desplegados sobre Google Cloud Platform (GCP) y visualizados mediante Looker Studio.

**1. Requisitos previos**

**Herramientas necesarias:**

* Cuenta activa en **Google Cloud Platform (GCP)**.
* Permisos para crear una instancia **Cloud SQL (MySQL)**.
* Acceso a **Google Colab** o entorno local con Python 3 y librerías necesarias.
* Archivos .csv en formato separado por ; con los siguientes nombres:
  + cases.csv
  + gender.csv
  + municipality.csv
  + status.csv
  + type\_contagion.csv
  + Department.csv

**2. Crear base de datos en Cloud SQL (GCP)**

**Pasos:**

1. Inicia sesión en tu consola de GCP.
2. Dirígete a **SQL > Crear instancia**.
3. Elige **MySQL** como motor.
4. Configura:
   * Nombre: myinstance
   * Usuario: root
   * Contraseña: 1234
   * Base de datos: covid\_db
5. Activa el **acceso público** temporal:
   * En la instancia, ve a "CONEXIONES" > "Direcciones IP autorizadas".
   * Agrega tu IP actual.
6. Copia la IP externa de tu instancia (ej. 34.10.20.53).

**3. Subir los datos a MySQL usando Google Colab**

**Pasos:**

1. Abre Google Colab.
2. Crea una nueva libreta y copia el siguiente código:

!pip install pandas pymysql

from google.colab import files

import pandas as pd

from sqlalchemy import create\_engine

# Conexión a MySQL

user = "root"

password = "1234"

host = "34.10.20.53" # cambia por tu IP

port = "3306"

database = "covid\_db"

engine = create\_engine(f"mysql+pymysql://{user}:{password}@{host}:{port}/{database}")

# Archivos a subir

csv\_files = ['Department.csv', 'cases.csv', 'gender.csv', 'municipality.csv', 'status.csv', 'type\_contagion.csv']

# Subir archivos

print("Selecciona los archivos CSV:")

uploaded = files.upload()

# Subir a MySQL

for filename in uploaded.keys():

if filename in csv\_files:

df = pd.read\_csv(filename, sep=';')

df.columns = df.columns.str.replace(' ', '\_').str.replace('[^A-Za-z0-9\_]+', '', regex=True)

table\_name = filename.replace('.csv', '')

df.to\_sql(name=table\_name, con=engine, if\_exists='replace', index=False)

print(f"Tabla '{table\_name}' creada en la base '{database}'")

**4. Crear tabla consolidada**

1. En la misma libreta, ejecuta el siguiente bloque:

# Leer tablas desde MySQL

dfs = {}

for file in csv\_files:

table\_name = file.replace('.csv', '')

dfs[file] = pd.read\_sql\_table(table\_name, engine)

# Renombrar columnas para evitar colisiones

dfs['gender.csv'].rename(columns={'name': 'gender\_name'}, inplace=True)

dfs['municipality.csv'].rename(columns={'name': 'municipality\_name'}, inplace=True)

dfs['status.csv'].rename(columns={'name': 'status\_name'}, inplace=True)

dfs['type\_contagion.csv'].rename(columns={'name': 'type\_contagion\_name'}, inplace=True)

dfs['Department.csv'].rename(columns={'name': 'department\_name'}, inplace=True)

# Consolidar con JOINs

consolidated\_df = dfs['cases.csv'] \

.merge(dfs['gender.csv'], on='id\_gender', how='left') \

.merge(dfs['municipality.csv'], on='id\_municipality', how='left') \

.merge(dfs['status.csv'], on='id\_status', how='left') \

.merge(dfs['type\_contagion.csv'], on='id\_type', how='left') \

.merge(dfs['Department.csv'], on='id\_department', how='left')

# Subir tabla consolidada

table\_name\_consolidated = 'consolidada'

consolidated\_df.to\_sql(name=table\_name\_consolidated, con=engine, if\_exists='replace', index=False)

print(f"Tabla '{table\_name\_consolidated}' creada correctamente")

**5. Conectar Looker Studio a MySQL**

**Pasos:**

1. Abre [Looker Studio](https://lookerstudio.google.com/).
2. Crea un nuevo reporte.
3. Agrega una nueva fuente de datos:
   * Tipo: **MySQL**
   * Host: 34.10.20.53
   * Puerto: 3306
   * Usuario: root
   * Contraseña: 1234
   * Base de datos: covid\_db
4. Selecciona la tabla consolidada.
5. Agrega y configura visualizaciones:
   * KPI: Total contagios
   * KPI: Total recuperados (filtrando por status = recuperado)
   * KPI: Fallecidos
   * Promedio de días de recuperación (campo calculado)
   * % de fallecidos por rango de edad (campo calculado con CASE)
   * Filtro por fecha (usa date\_symptom o date\_diagnosis)

**7. Resultado Final**

* Base de datos en MySQL GCP con tabla consolidada disponible.
* Tablero en Looker Studio público y funcional con todos los KPIs solicitados.
  1. **Posibles errores comunes y cómo solucionarlos**

### Error: Can't connect to MySQL server on 'localhost'

**Solución:**

* Verifica que estás usando la IP pública de GCP, no localhost.
* Revisa que el puerto 3306 esté habilitado en las reglas de firewall.

### Error: FileNotFoundError: No such file or directory

**Solución:**

* Asegúrate de que los archivos CSV estén subidos correctamente al entorno de ejecución.

### Error al subir archivos CSV por separador

**Solución:**

* Revisa que el separador correcto sea ; (usa sep=';' en pd.read\_csv).

### Columnas con nombres name duplicados

**Solución:**

* Cambiar el nombre de las columnas name al hacer merge:

.rename(columns={'name': 'gender\_name'})

### Error en el dashboard: tabla consolidada no visible

**Solución:**

* Verifica que consolidada se haya cargado correctamente a MySQL.
* Revisa permisos y credenciales en Looker Studio.