Actividad 1

Universidad Internacional de Valencia.

Maestría Oficial en Desarrollo de Aplicaciones y Servicios Web.

Análisis de datos web

Edgar Gamarra

William Forero

2022.

Introducción:

Para la realización del trabajo se utilizaron herramientas vistas en clase y adicional se investigó más información en internet.

Objetivos:

En esta actividad se pretende usar todos los conocimientos aprendidos, de igual manera es necesario realizar diferentes consultas en las herramientas disponibles (buscadores de internet, artículos, libros) para lograr dar cumplimiento a los puntos planteados para el ejercicio.

I. Contexto de uso:

COVID 19 en Colombia en los departamentos de Antioquia, Cundinamarca y Cesar.

Por medio de los cuadros de mandos se quiere en primer lugar validar el total de casos reportados en los tres departamentos de Colombia: Cundinamarca, Medellín y Cesar.

Validar los casos reportados por: sexo (F, M), su estado actual: Fallecido, Recuperado, leve, cantidad por edades.

El pico más alto de contagios reportados.

II. Fuentes de datos:

Fuentes de datos usadas de datos abiertos de Colombia:

https://www.datos.gov.co/

- Datos Antioquia:

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Casos-positivos-de-Covid-19-en-el-departamento-de-/w3du-c2j6

- Datos Cundinamarca:

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Vista-Casos-Positivos-COVID-19-Cundinamarca/rik9-88u9

- Datos Cesar:

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Casos-positivos-de-COVID-19-en-el-Departamento-del/uspf-j4t8

III. Limpieza de datos

Ya con los datos csv descargados, se procede a utilizar Google Colab el cual nos brinda un ambiente en la nube para ejecutar código Python, utilizado para realizar las respectivas operaciones de limpieza y trasformación de los datos.

Google colab brinda dos maneras de cargar los archivos:

1-Carga manual de los archivos por la opción de subir archivos que proporciona el aplicativo

2-Por medio de código (utilizada en este caso)

```
#CODIGO PARA CARGA DE ARCHIVOS A GOOGLE COLAB
from google.colab import files
uploaded = files.upload()
```

Esto nos permite cargar en el navegador los archivos que se van a utilizar para la ETL

```
#IMPORTAMOS LAS LIBRERIAS A UTILIZAR
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Invocamos las librerías que necesitamos utilizar y como buena práctica le asignamos un alias más corto para su posterior uso.

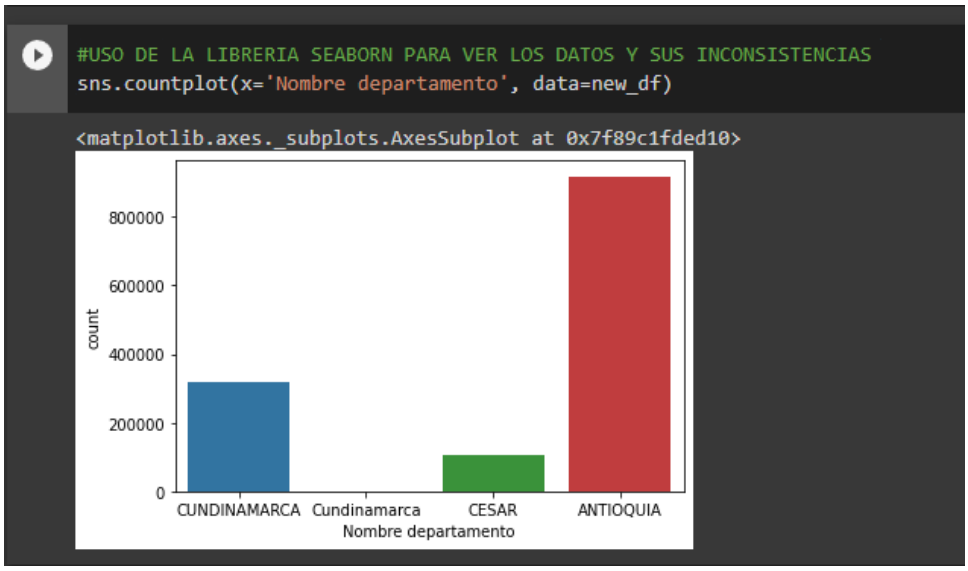
```
#LECTURA DE ARCHIVOS CSV
cundinamarca = pd.read_csv('/content/Vista_Casos_Positivos_COVID-19_Cundinamarca.csv')
antioquia = pd.read_csv('/content/Casos_positivos_de_Covid-19_en_el_departamento_de_Antioquia.csv')
cesar = pd.read_csv('/content/Casos_positivos_de_COVID-19_en_el_Departamento_del_Cesar.csv')
```

Posteriormente se leen los archivos csv cargados y se empieza con el tratamiento de los datos.

Ya con los datos leídos por Python lo primero que se realiza es la concatenación de los datos ya que se confirma que manejan la misma cantidad de columnas.

Después de este proceso se empieza a realizar la limpieza de los datos.

Inicialmente se revisan que en los datos de las diferentes columnas que se van a utilizar no se encuentren variables diferentes:



Se observa que para el caso de CUNDINAMARCA los datos no están unificados

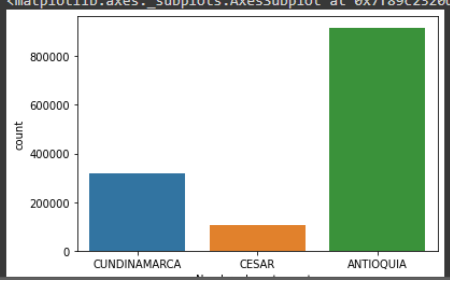
```
#REVISAMOS LOS VALORES QUE CONTENGAN LA CADENA DE TEXTO INDICADA
dpto_data = new_df['Nombre departamento'].str.contains('Cund', case = False)
dpto_data
```

```
0          True
1          True
2          True
3          True
4          True
          ...
918277    False
918278    False
918279    False
918280    False
918281    False
Name: Nombre departamento, Length: 1342979, dtype: bool
```

```
[17] #REEMPLAZAMOS LOS DATOS CON LA CADENA DE TEXTO INDICADA
new_df['Nombre departamento'] = np.where(dpto_data, 'CUNDINAMARCA',new_df['Nombre departamento'])
```

```
[18] sns.countplot(x='Nombre departamento', data=new_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f89c2320dd0>
```



Con este proceso ya se han unificado los valores, esta misma operación se realiza con cada una de las columnas que se van a utilizar.

Después de realizar este proceso se determina que las columnas que se van a usar para el datawarehouse son las siguientes:

Fecha_notificacion

Nombre_departamento

Edad

Sexo

Tipo_contagio

Ubicación_caso

Estado

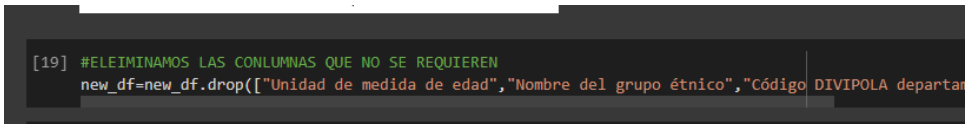
Recuperado

Tipo recuperacion

```
[37] #AGREGAMOS UNA NUEVA COLUMNA LLAMADA COLOMBIA CON VALOR COLOMBIA PARA POSTERIORMENTE EN DATA STUDIO GENERAR UN MAPA
new_df["PAIS"]="COLOMBIA"
```

Y se crea una nueva columna para especificar el país al que pertenecen los departamentos para un posterior gráfico en data Studio.

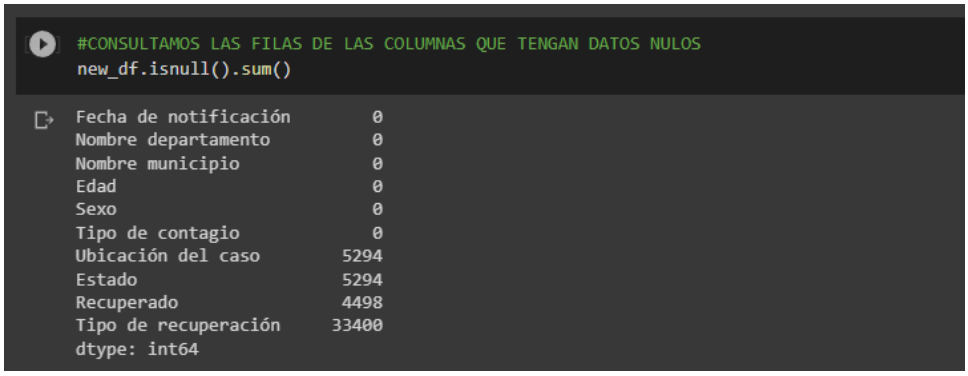
Luego de que se definen las columnas a utilizar, se eliminan las que no se requieren. Para ello se utiliza lo siguiente:



```
[19] #ELEIMINAMOS LAS CONLUMNAS QUE NO SE REQUIEREN
     new_df=new_df.drop(["Unidad de medida de edad","Nombre del grupo étnico","Código DIVIPOLA departar
```

De esta manera ya tendremos filtrados solo las columnas que necesitamos.

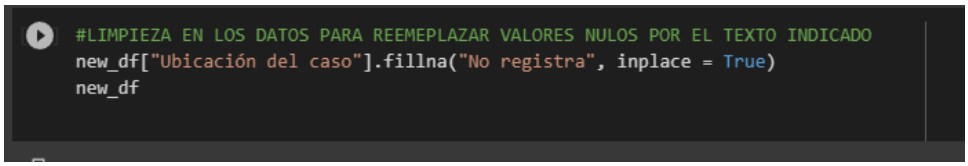
Se continúa con el proceso de limpieza, para este caso se revisan las filas de las columnas que tengan datos nulos:



```
#CONSULTAMOS LAS FILAS DE LAS COLUMNAS QUE TENGAN DATOS NULOS
new_df.isnull().sum()
```

```
Fecha de notificación        0
Nombre departamento          0
Nombre municipio             0
Edad                         0
Sexo                         0
Tipo de contagio             0
Ubicación del caso        5294
Estado                    5294
Recuperado                4498
Tipo de recuperación     33400
dtype: int64
```

Como no son pocos los datos nulos se optan por reemplazar el valor nulo por el texto no registra como se indica a continuación:



```
#LIMPIEZA EN LOS DATOS PARA REEMEPLAZAR VALORES NULOS POR EL TEXTO INDICADO
new_df["Ubicación del caso"].fillna("No registra", inplace = True)
new_df
```

Al finalizar la revisión y al terminar de reemplazar los datos nulos se ejecuta nuevamente el método isnull() buscando datos nulos.

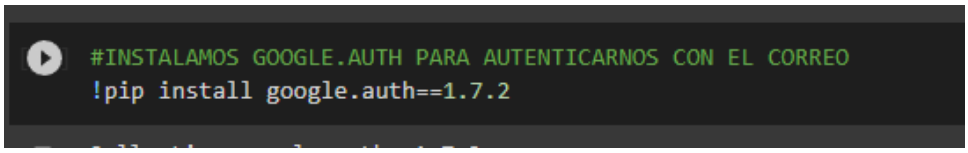
```
#CONSULTAMOS NUEVAMENTE PARA VALIDAR QUE YA NO EXISTAN DATOS NULOS
new_df.isnull().sum()
```

```
Fecha de notificación    0
Nombre departamento      0
Nombre municipio         0
Edad                     0
Sexo                     0
Tipo de contagio         0
Ubicación del caso       0
Estado                   0
Recuperado               0
Tipo de recuperación     0
dtype: int64
```

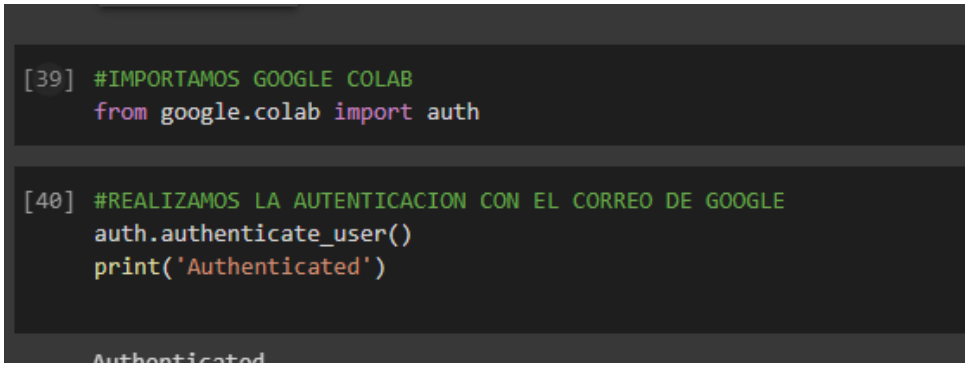
Finalmente, nuestra fuente de datos está lista para ser cargada al repositorio de datos en este caso se utilizó Bigquery por la facilidad que permite para representar los datos en data Studio.

IV ingesta en Bigquery

Inicialmente se debe instalar google.auth para autenticación con el correo electrónico

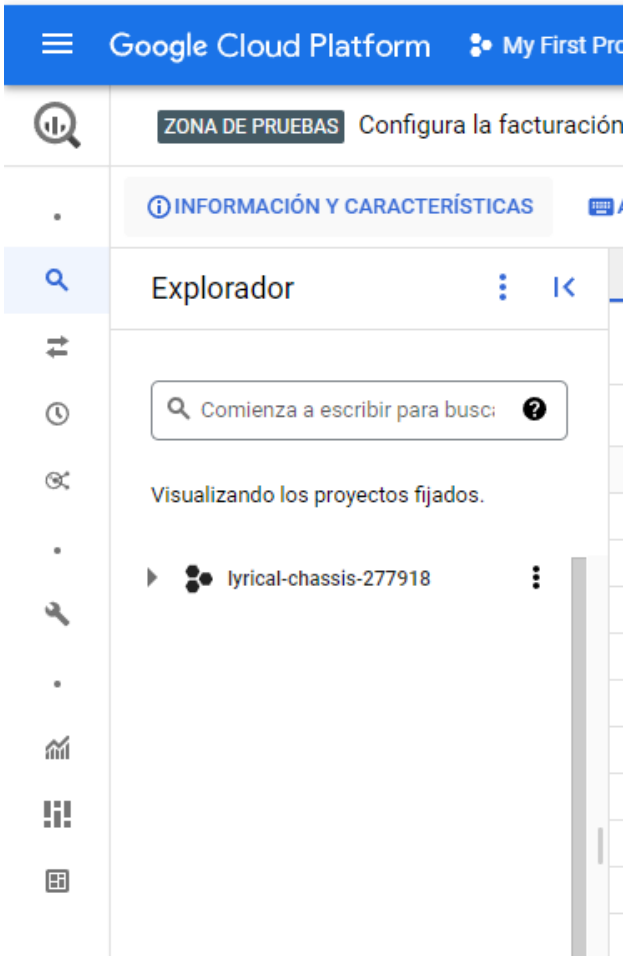


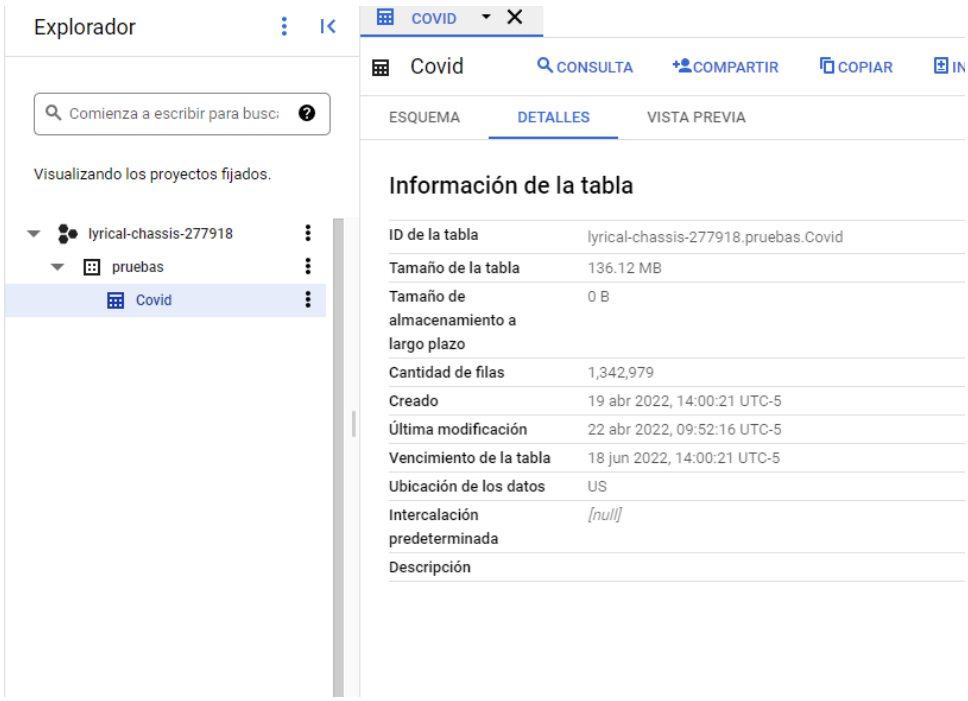
```
#INSTALAMOS GOOGLE.AUTH PARA AUTENTICARNOS CON EL CORREO
!pip install google.auth==1.7.2
```



```
[39] #IMPORTAMOS GOOGLE COLAB
     from google.colab import auth

[40] #REALIZAMOS LA AUTENTICACION CON EL CORREO DE GOOGLE
     auth.authenticate_user()
     print('Authenticated')
```

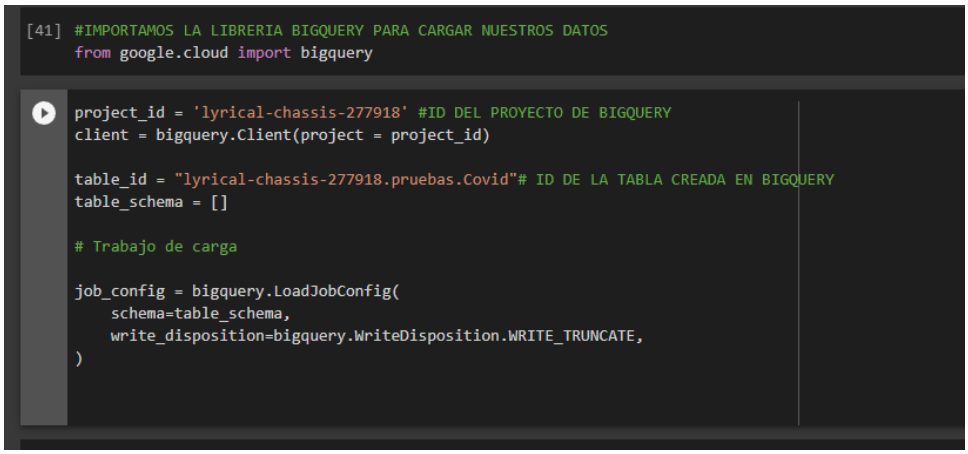
En bigquery creamos una cuenta gratuita y también un proyecto para la carga de los datos





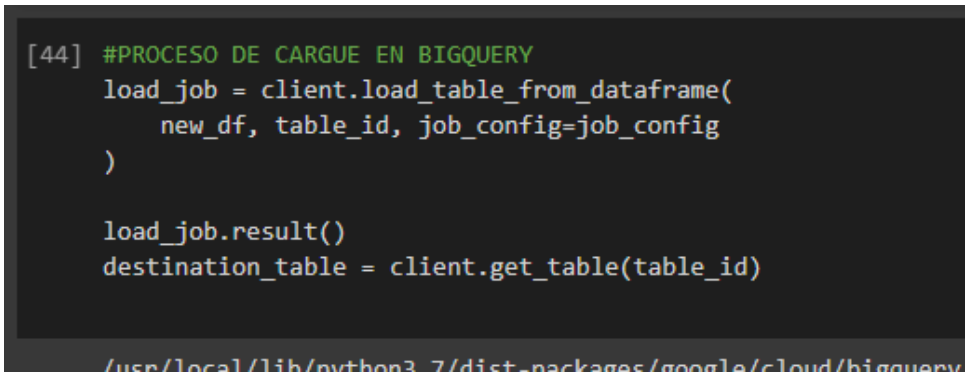
Se deben tener presente los datos como nombre del proyecto y Id de la tabla para la carga de los datos y posteriormente en Google Colab se ejecutan las siguientes líneas de código:

Se llama a la librería de bigquery para acceder al proyecto creado:



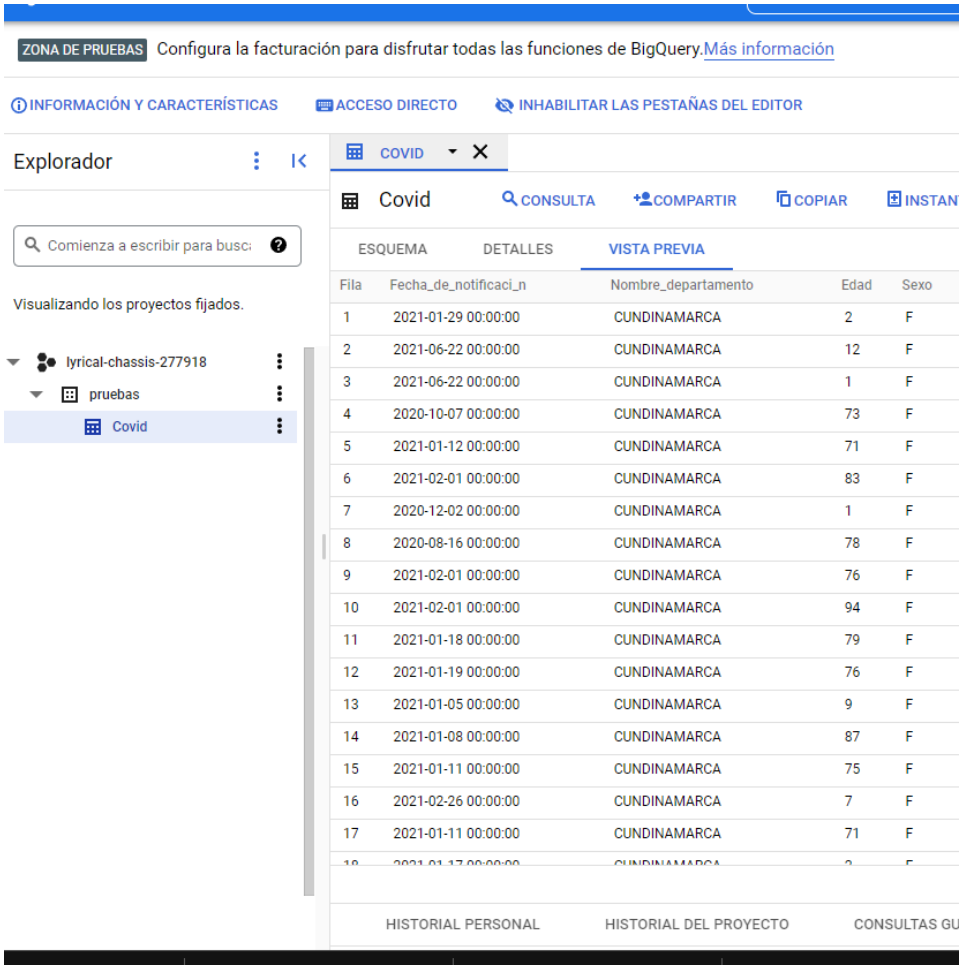
```
[41] #IMPORTAMOS LA LIBRERIA BIGQUERY PARA CARGAR NUESTROS DATOS
     from google.cloud import bigquery

project_id = 'lyrical-chassis-277918' #ID DEL PROYECTO DE BIGQUERY
client = bigquery.Client(project = project_id)

table_id = "lyrical-chassis-277918.pruebas.Covid"# ID DE LA TABLA CREADA EN BIGQUERY
table_schema = []

# Trabajo de carga

job_config = bigquery.LoadJobConfig(
    schema=table_schema,
    write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE,
)
```

Y finalmente se ejecuta el script para carga de los datos:



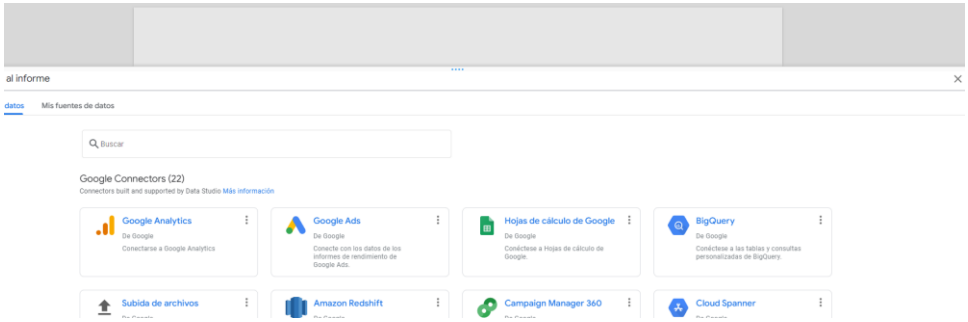
```
[44] #PROCESO DE CARGUE EN BIGQUERY
     load_job = client.load_table_from_dataframe(
         new_df, table_id, job_config=job_config
     )

     load_job.result()
     destination_table = client.get_table(table_id)
```

En la opción de vista previa ya se pueden ver el Datawarehose con sus respectivos datos

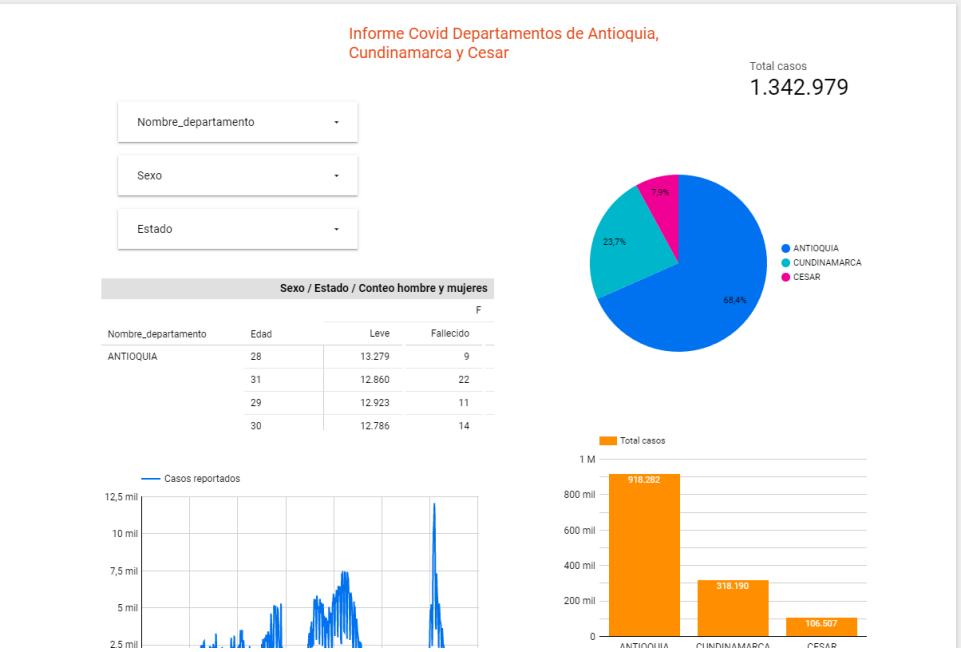


Fila	Fecha_de_notificaci_n	Nombre_departamento	Edad	Sexo
1	2021-01-29 00:00:00	CUNDINAMARCA	2	F
2	2021-06-22 00:00:00	CUNDINAMARCA	12	F
3	2021-06-22 00:00:00	CUNDINAMARCA	1	F
4	2020-10-07 00:00:00	CUNDINAMARCA	73	F
5	2021-01-12 00:00:00	CUNDINAMARCA	71	F
6	2021-02-01 00:00:00	CUNDINAMARCA	83	F
7	2020-12-02 00:00:00	CUNDINAMARCA	1	F
8	2020-08-16 00:00:00	CUNDINAMARCA	78	F
9	2021-02-01 00:00:00	CUNDINAMARCA	76	F
10	2021-02-01 00:00:00	CUNDINAMARCA	94	F
11	2021-01-18 00:00:00	CUNDINAMARCA	79	F
12	2021-01-19 00:00:00	CUNDINAMARCA	76	F
13	2021-01-05 00:00:00	CUNDINAMARCA	9	F
14	2021-01-08 00:00:00	CUNDINAMARCA	87	F
15	2021-01-11 00:00:00	CUNDINAMARCA	75	F
16	2021-02-26 00:00:00	CUNDINAMARCA	7	F
17	2021-01-11 00:00:00	CUNDINAMARCA	71	F

Para finalizar se crea un proyecto en data Studio y se selecciona bigquery como fuente de datos:



Ya con la conexión se crea el panel de gráficos con la información necesaria:



Se agrega url de visualización:

https://datastudio.google.com/embed/reporting/f353e3d2-fa3e-4a4e-8a54-93ae82f1f21c/page/wt1qC

Conclusiones

1- El total de casos repostados es de 1.342.979 sumando los tres departamentos.

2- Antioquia es el departamento con más casos reportados.

3- En fallecimientos los hombres tienen una cantidad superior al de las mujeres.

4- en las edades superiores a los 69 años en donde se refleja mayor cantidad de muertes por COVID

5- Se confirma que el mayor pico reportado es en enero del 2022, lo más probable es que esto se deba a las festividades de fin de año y vacaciones.