



**MACC**  
Matemáticas Aplicadas y  
Ciencias de la Computación



Universidad del  
**Rosario**

# PROYECTO FINAL INGENIERÍA DE DATOS

2022-1

Sofía Ochoa Gutiérrez  
Mariana Cadena Moreno  
Valentina Pabón León

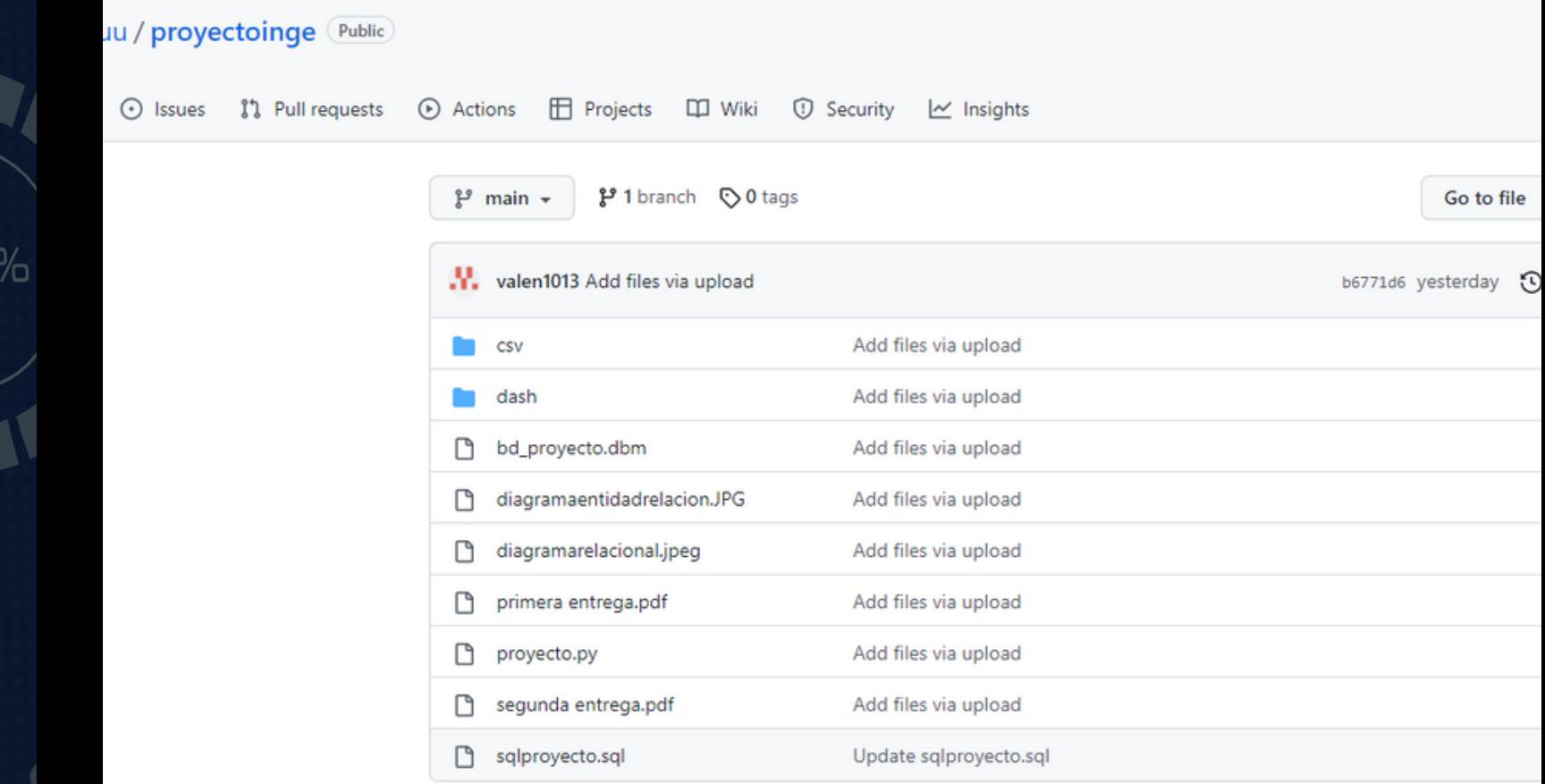
# REPOSITORIO

<https://github.com/Meiiyuu/proyectoinge>

third quarter

second quarter

first quarter



Meiiyuu / proyectoinge Public

Issues Pull requests Actions Projects Wiki Security Insights

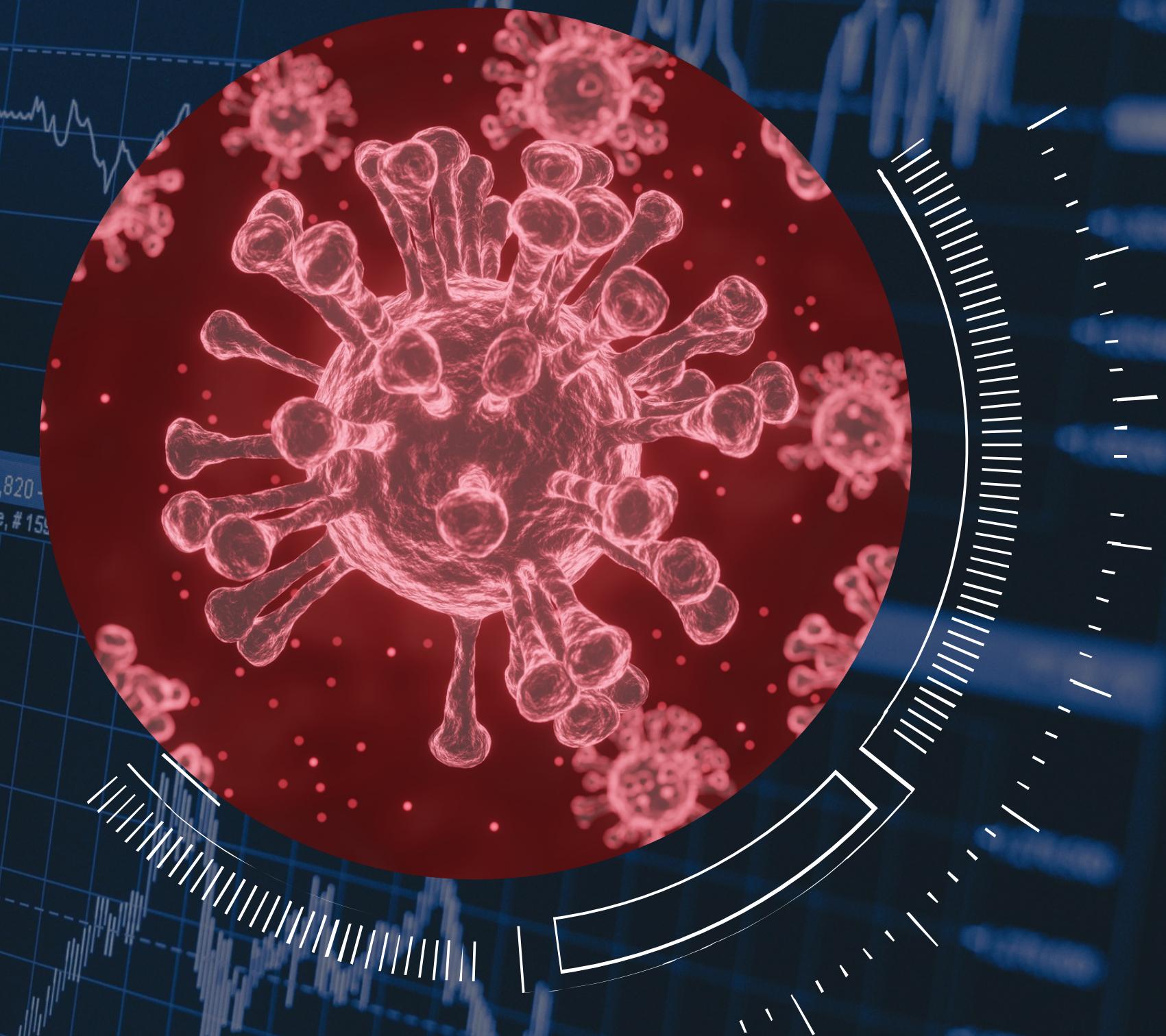
main · 1 branch · 0 tags

valen1013 Add files via upload b6771d6 yesterday

csv	Add files via upload
dash	Add files via upload
bd_proyecto.dbm	Add files via upload
diagramaentidadrelacion.JPG	Add files via upload
diagramarelacional.jpeg	Add files via upload
primera entrega.pdf	Add files via upload
proyecto.py	Add files via upload
segunda entrega.pdf	Add files via upload
sqlproyecto.sql	Update sqlproyecto.sql

# FUENTE DE DATOS

Registro de los casos positivos de COVID-19 en Colombia en el departamento de Sucre hasta el mes de mayo del año 2022. Datos proporcionados por el Instituto Nacional de Salud.



# REGLAS DE NEGOCIO

01

Cada caso tiene asignado un número de registro único.

02

Se pueden presentar varios casos en un municipio.

03

Un grupo étnico puede tener varios casos de contagios de Covid-19.

04

En más de un caso se pueden presentar las mismas características del contagio (estado).

05

Cada municipio tiene un código único.

06

Si el paciente se recupera, no habrá fecha de muerte.

07

La fecha de muerte y de recuperación pueden tener o no datos puesto que, si el paciente no murió, fecha\_muerte será nula, y si murió fecha\_recuperación será nula.

08

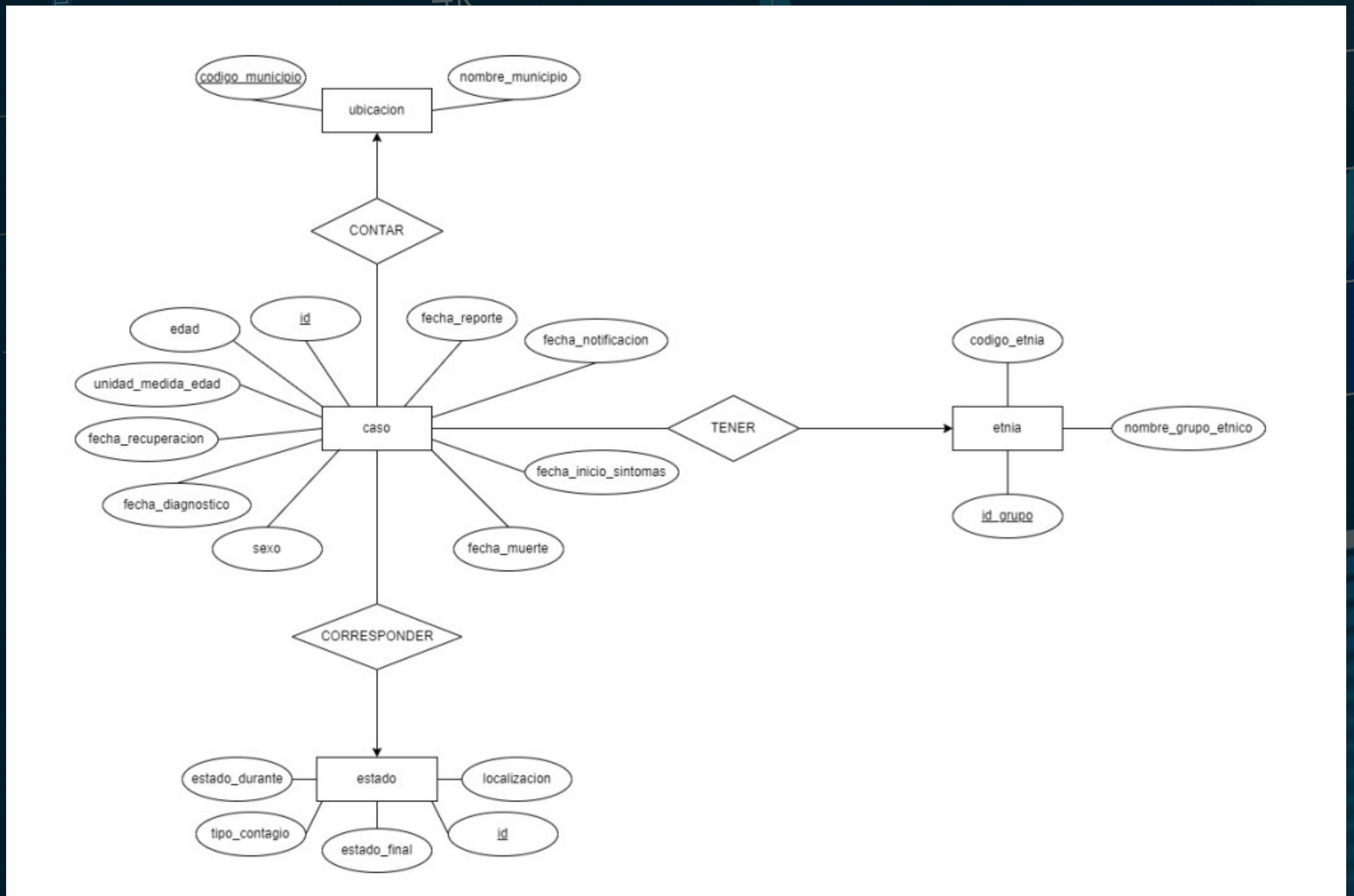
estadotiene un identificador único.

09

etnia tiene un identificador único.



# DIAGRAMA ENTIDAD- RELACION

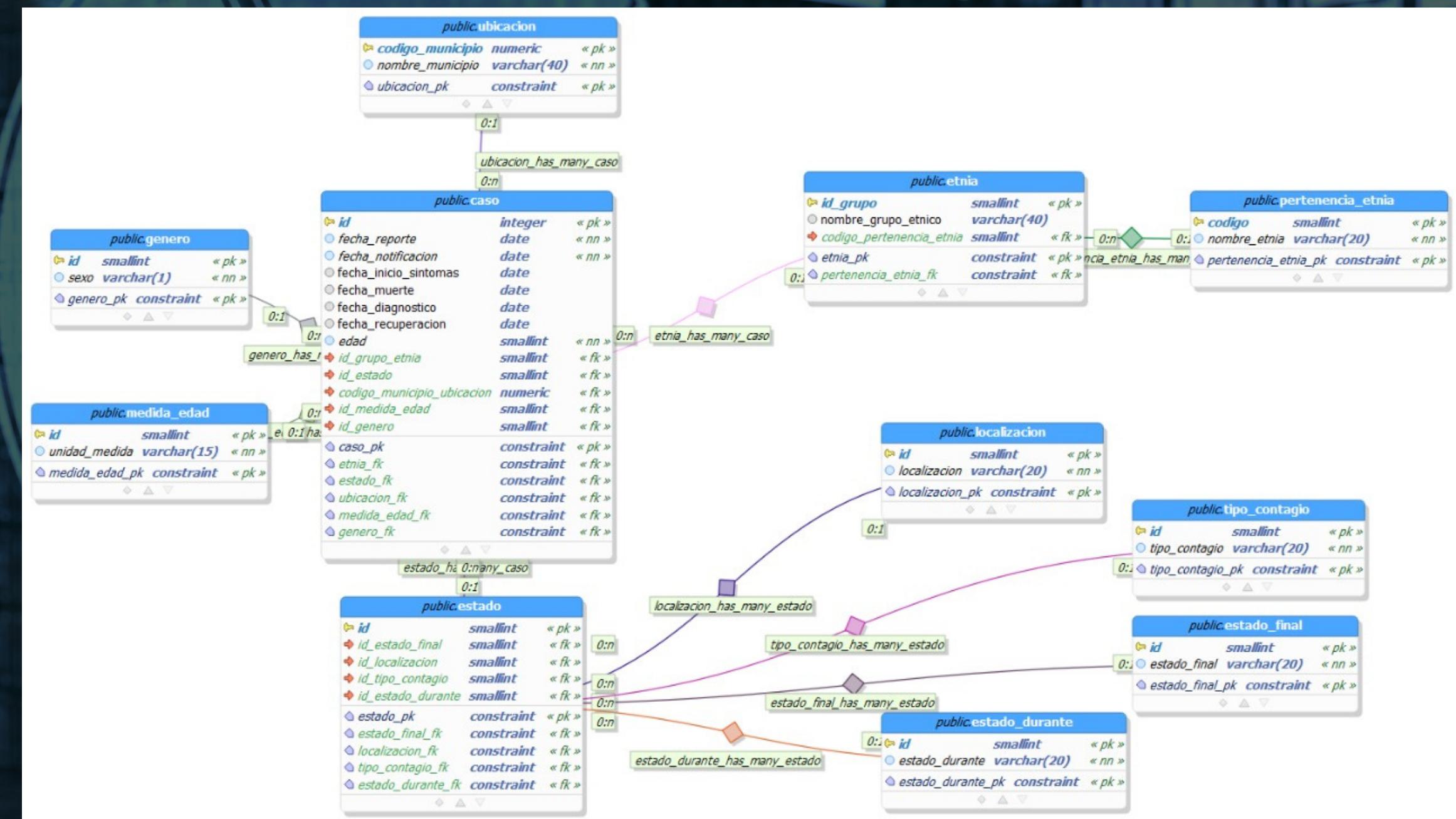


12 13 14 15 16 17 18 19

4k



# DIAGRAMA RELACIONAL



# BASE DE DATOS EN POSTGRESQL

## INSERCIÓN DE DATOS

```
233 COPY pertenencia_etnia FROM 'C:\csv\pertenencia_etnia.csv' WITH delimiter ';' CSV HEADER;
234 COPY etnia FROM 'C:\csv\etnia.csv' WITH delimiter ';' CSV HEADER;
235 COPY localizacion FROM 'C:\csv\localizacion.csv' WITH delimiter ';' CSV HEADER;
236 COPY tipo_contagio FROM 'C:\csv\tipo_contagio.csv' WITH delimiter ';' CSV HEADER;
237 COPY estado_final FROM 'C:\csv\estado_final.csv' WITH delimiter ';' CSV HEADER;
238 COPY estado_durante FROM 'C:\csv\estado_durante.csv' WITH delimiter ';' CSV HEADER;
239 COPY estado FROM 'C:\csv\estado.csv' WITH delimiter ';' CSV HEADER;
240 COPY medida_edad FROM 'C:\csv\medida_edad.csv' WITH delimiter ';' CSV HEADER;
241 COPY genero FROM 'C:\csv\genero.csv' WITH delimiter ';' CSV HEADER;
242 COPY ubicacion FROM 'C:\csv\ubicacion.csv' WITH delimiter ';' CSV HEADER;
243 COPY caso FROM 'C:\csv\caso.csv' WITH delimiter ';' CSV HEADER;
```

# BASE DE DATOS EN POSTGRESQL

## CONEXION Y CONSULTA CON PYTHON

```
import psycopg2
import pandas as pd

engine = None
resultDataFrame = None
try:
    print("Connecting...\n")

    engine = psycopg2.connect(user="postgres",
                               password="123456789",
                               database="proyecto",
                               host="localhost",
                               port="5432")

    tablas=["caso","estado","ubicacion","genero","medida_edad","localizacion", "etnia", "pertenencia_etnia", "tipo_contagio", "estado_final", "estado_durante"]
    for t in tablas:
        query = 'select * from {}'.format(t)
        print(t)
        tabla = pd.read_sql_query(query, engine)
        print(tabla)
        print()
        print()

    print("Disconnecting...\n")

except(Exception, psycopg2.DatabaseError) as error:
    print("Error:", error)
finally:
    if engine is not None:
        engine.close()
```

# DASH

```
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import pandas as pd
import psycopg2
from dash import Dash, dcc, html

def CasosRecuperadosByEtnias():
    return """select pe.nombre_etnia, count(c.id)
        from etnia e INNER JOIN caso c ON (e.id_grupo = c.id_grupo_etnia)
        INNER JOIN estado es ON (c.id_estado = es.id)
        INNER JOIN estado_final ef ON (es.id_estado_final = ef.id)
        inner join pertenencia_etnia as pe on (e.codigo_pertenencia_etnia=pe.codigo)
    WHERE ef.estado_final = 'Recuperado'
    group by (pe.nombre_etnia)
    order by (count(c.id))"""

def ProbablesRecuperadosByEdad():
    return """select c.edad, ef.estado_final, count(c.id)
        from caso c INNER JOIN estado es ON (c.id_estado = es.id)
        INNER JOIN estado_final ef ON (es.id_estado_final = ef.id)
    group by (c.edad, ef.estado_final)
    order by edad asc"""

def TiempoRecuperacionByEdad():
    return """select c.edad, AVG(extract(DAY FROM age(c.fecha_recuperacion,c.fecha_inicio_sintomas))) as tiempo_recuperacion, count(c.id)
        from caso c INNER JOIN medida_edad m ON (c.id_medida_edad = m.id)
    where m.unidad_medida = 'Anios'
    group by(edad)
    ORDER BY edad ASC"""

def ContagiosByMunicipio():
    return """select u.nombre_municipio, ef.estado_final, count(c.id)
        from ubicacion u INNER JOIN caso c ON (u.codigo_municipio = c.codigo_municipio_ubicacion)
        INNER JOIN estado es ON (c.id_estado = es.id)
        INNER JOIN estado_final ef ON (es.id_estado_final = ef.id)
    where ef.estado_final = 'Recuperado' or ef.estado_final = 'Fallecido'
    group by(u.nombre_municipio, ef.estado_final)
    order by(count(c.id)) desc"""

def RecuperadosFallecidosByAnho():
    return """select count(c.id), ef.estado_final, EXTRACT(YEAR FROM fecha_reporte) as anio
        from caso c INNER JOIN estado es ON (c.id_estado = es.id)
        INNER JOIN estado_final ef ON (es.id_estado_final = ef.id)
    where ef.estado_final = 'Recuperado' or ef.estado_final = 'Fallecido' and
    EXTRACT(YEAR FROM fecha_reporte) = 2020 or EXTRACT(YEAR FROM fecha_reporte) = 2021
    group by(ef.estado_final, anio)"""
```