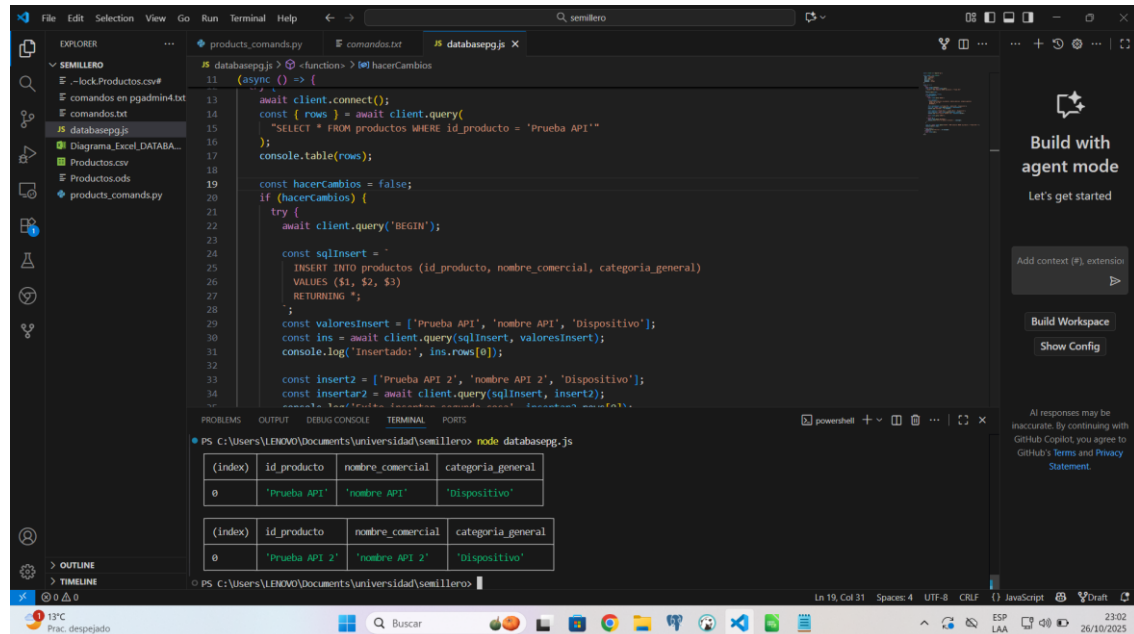


Jhonathan David Guevara Ramirez
SEMILLERO

PRUEBA DE API CONEXIÓN CON BASE DE DATOS



```

11 (async () => {
12   await client.connect();
13   const ( rows ) = await client.query(
14     "SELECT * FROM productos WHERE id_producto = 'Prueba API'"
15   );
16   console.table(rows);
17
18   const hacerCambios = false;
19   if (hacerCambios) {
20     try {
21       await client.query('BEGIN');
22
23       const sqlInsert = `
24         INSERT INTO productos (id_producto, nombre_comercial, categoria_general)
25         VALUES ($1, $2, $3)
26         RETURNING *;
27       `;
28       const valoresInsert = ['Prueba API', 'nombre API', 'Dispositivo'];
29       const ins = await client.query(sqlInsert, valoresInsert);
30       console.log('Insertado:', ins.rows[0]);
31
32       const insert2 = ['Prueba API 2', 'nombre API 2', 'Dispositivo'];
33       const insertar2 = await client.query(sqlInsert, insert2);
34     } catch (error) {
35       console.log('Error:', error);
36     }
37   }
38 })();

```

```

PS C:\Users\LEHMO\Documents\universidad\semillero> node databasepg.js

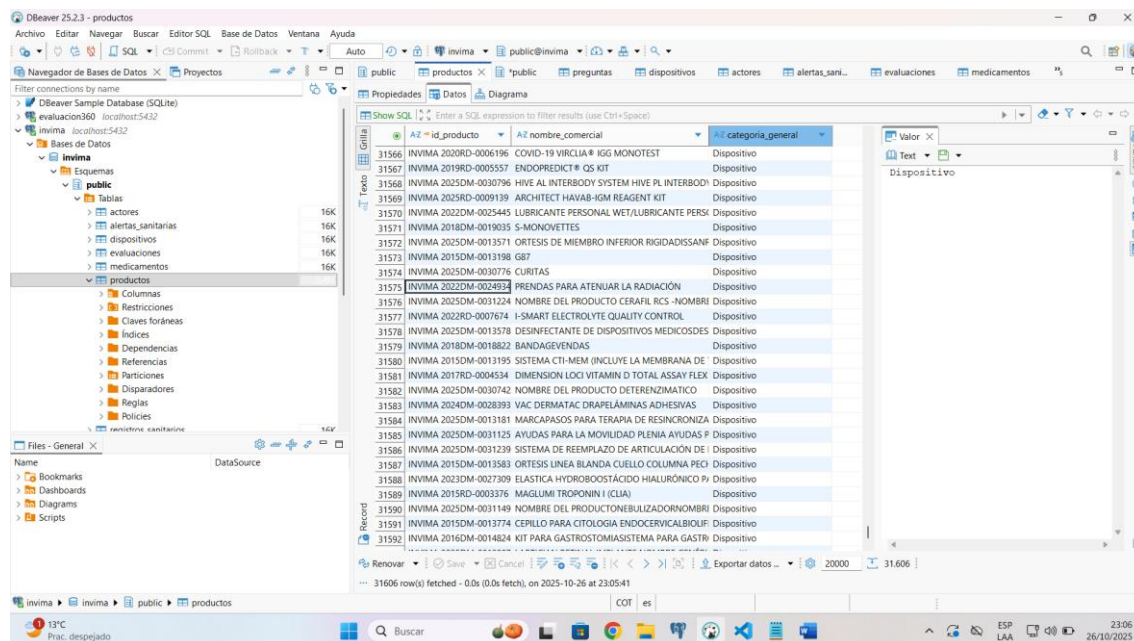
```

(index)	id_producto	nombre_comercial	categoria_general
0	'Prueba API'	'nombre API'	'Dispositivo'

(index)	id_producto	nombre_comercial	categoria_general
0	'Prueba API 2'	'nombre API 2'	'Dispositivo'

BASE DE DATOS ALIMENTADA PRIMERA FASE DE PRODUCTOS ALREDEDOR DE
31606 PRODUCTOS

ENFOQUE A FASE 2, 3, 4 Y 5



DBeaver 25.2.3 - productos

Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda

Navegador de Bases de Datos: public, productos, public, preguntas, dispositivos, actores, alertas_sani..., evaluaciones, medicamentos

Propiedades: Datos, Diagrama

Show SQL: Enter a SQL expression to filter results (use Ctrl+Space)

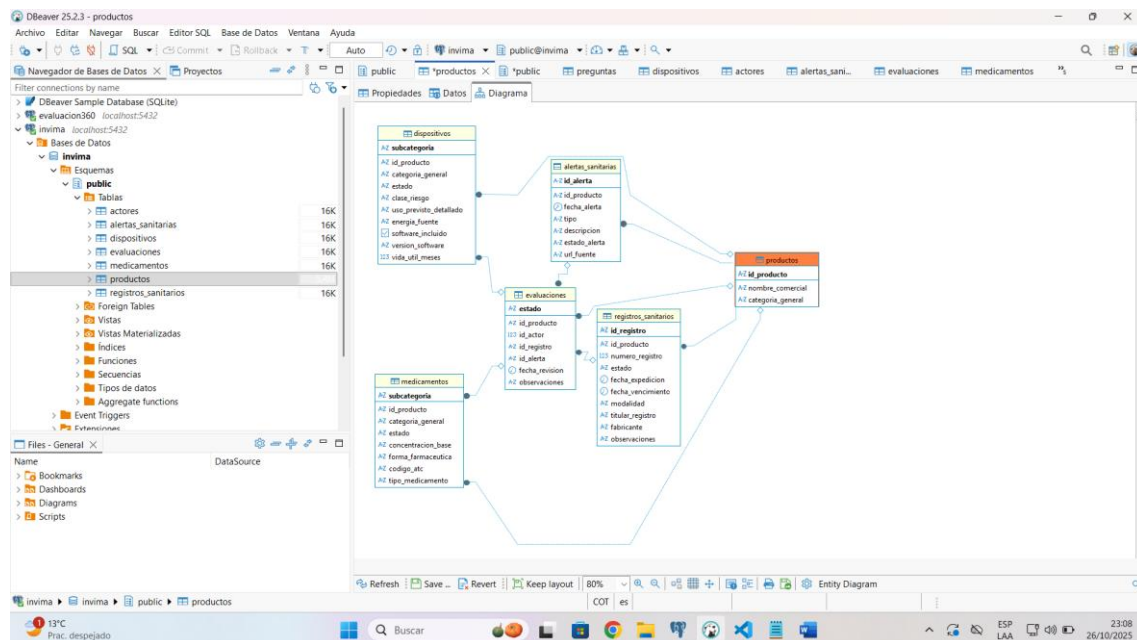
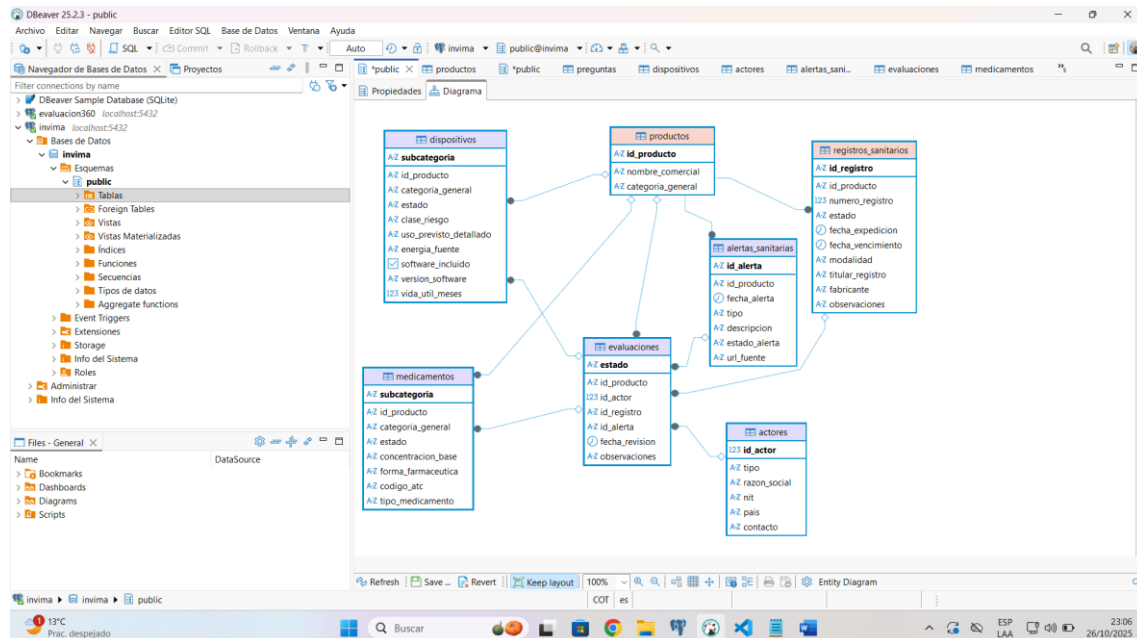
id_producto	nombre_comercial	categoria_general
31566	INVIMA 2020RD-0006196 COVID-19 VIRCLIA® IGG MONOTEST	Dispositivo
31567	INVIMA 2019RD-0005557 ENDOPROTECT® QS KIT	Dispositivo
31568	INVIMA 2025DM-0003796 HIVE AL INTERBODY SYSTEM HIVE PL INTERBOO	Dispositivo
31569	INVIMA 2025RD-0009139 ARCHITECT HAVAB-HGM REAGENT KIT	Dispositivo
31570	INVIMA 2025DM-0025445 LUBRICANTE PERSONAL WET/LUBRICANTE PERS	Dispositivo
31571	INVIMA 2018DM-0019035 S-MONOVETTES	Dispositivo
31572	INVIMA 2025DM-0013571 ORTESIS DE MIEMBRO INFERIOR RIGIDADISSANF	Dispositivo
31573	INVIMA 2015DM-0013198 G87	Dispositivo
31574	INVIMA 2025DM-0030776 CURITAS	Dispositivo
31575	INVIMA 2025DM-0024934 PRENDAS PARA ATENUAR LA RADIACIÓN	Dispositivo
31576	INVIMA 2025DM-0031224 NOMBRE DEL PRODUCTO CERAFIL RCS -NOMBRI	Dispositivo
31577	INVIMA 2022RD-0007674 I-SMART ELECTROLYTE QUALITY CONTROL	Dispositivo
31578	INVIMA 2025DM-0013578 DESINFECTANTE DE DISPOSITIVOS MEDICOSDES	Dispositivo
31579	INVIMA 2018DM-0018802 BANDAGEVIDENDAS	Dispositivo
31580	INVIMA 2015DM-0013195 SISTEMA CTI-MEM (INCLUYE LA MEMBRANA DE	Dispositivo
31581	INVIMA 2017RD-0004534 DIMENSION LOC VITAMIN D TOTAL ASSAY FLEX	Dispositivo
31582	INVIMA 2025DM-0030742 NOMBRE DEL PRODUCTO DETERENZIMATICO	Dispositivo
31583	INVIMA 2024DM-0028393 VAC DERMATAC DRAPELÁMNAS ADHESIVAS	Dispositivo
31584	INVIMA 2025DM-0013181 MARCAPASOS PARA TERAPIA DE RESINCRONIZA	Dispositivo
31585	INVIMA 2025DM-0031125 AYUDAS PARA LA MOVILIDAD PLENA AYUDAS P	Dispositivo
31586	INVIMA 2025DM-0013239 SISTEMA DE REEMPLAZO DE ARTICULACIÓN DE	Dispositivo
31587	INVIMA 2025DM-0013583 ORTESIS LINEA BLANDA CUELLO LUMÍNICO PEO	Dispositivo
31588	INVIMA 2023DM-0027309 ELASTICA HYDROBOOSTÁCIDO HALURÓNICO P	Dispositivo
31589	INVIMA 2015RD-0003376 MAGLUMI TROPONIN I (CLIA)	Dispositivo
31590	INVIMA 2025DM-0031149 NOMBRE DEL PRODUCTO NEBULIZADOR NOMBRI	Dispositivo
31591	INVIMA 2015DM-0013774 CEPILLO PARA CITOLOGIA ENDOCERVICALBIOLF	Dispositivo
31592	INVIMA 2016DM-0014824 KIT PARA GASTROSTOMIASISTEMA PARA GASTR	Dispositivo

31606 row(s) fetched - 0.0s (0.0s fetch), on 2025-10-26 at 23:05:41

Jhonathan David Guevara Ramirez
SEMILLERO

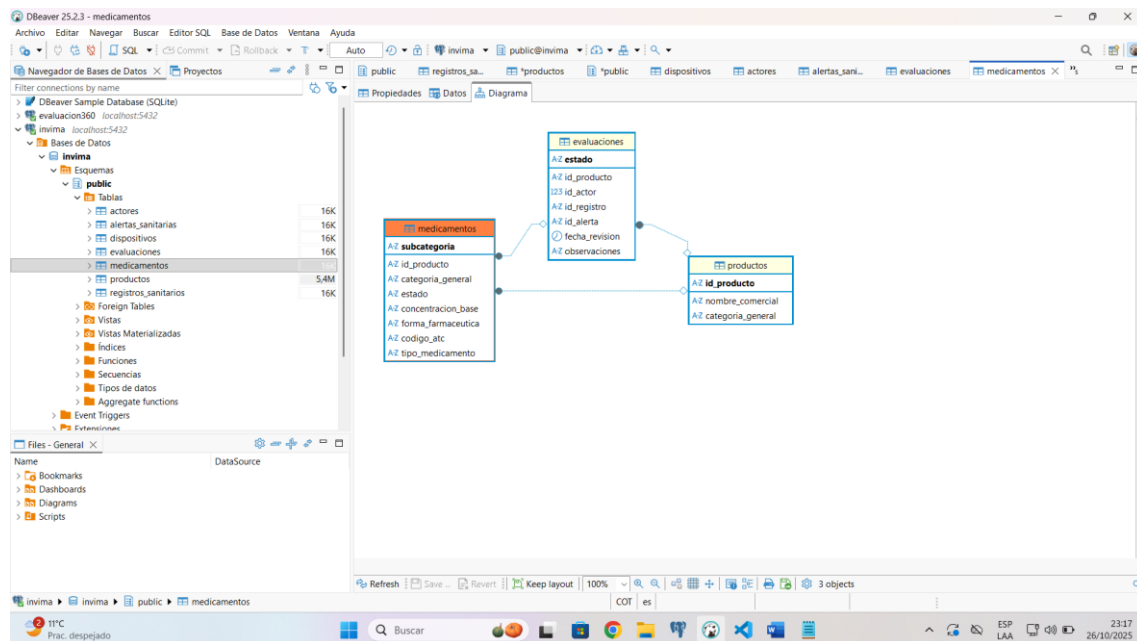
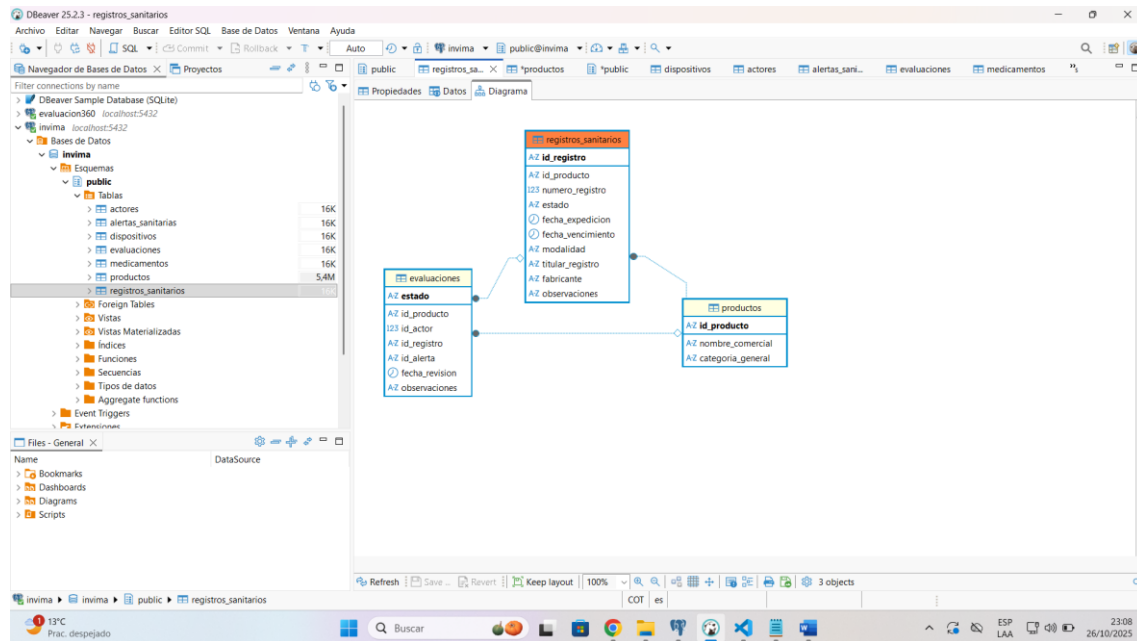
ESQUEMAS DE LA BASE DE DATOS FASE 1

ENFOQUE A FASE 2, 3, 4



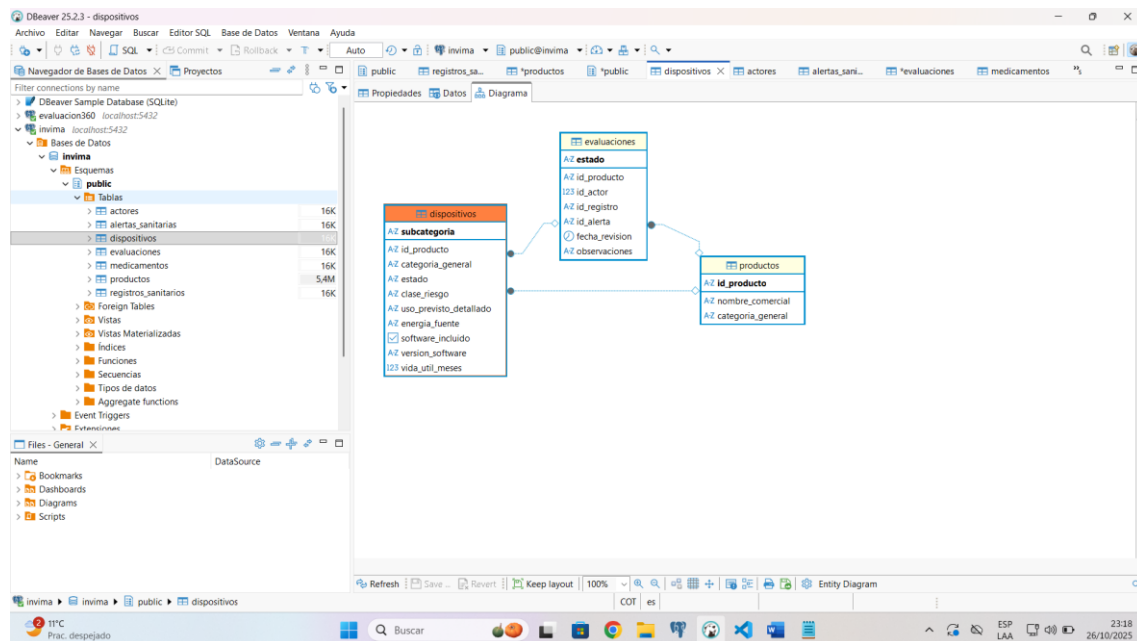
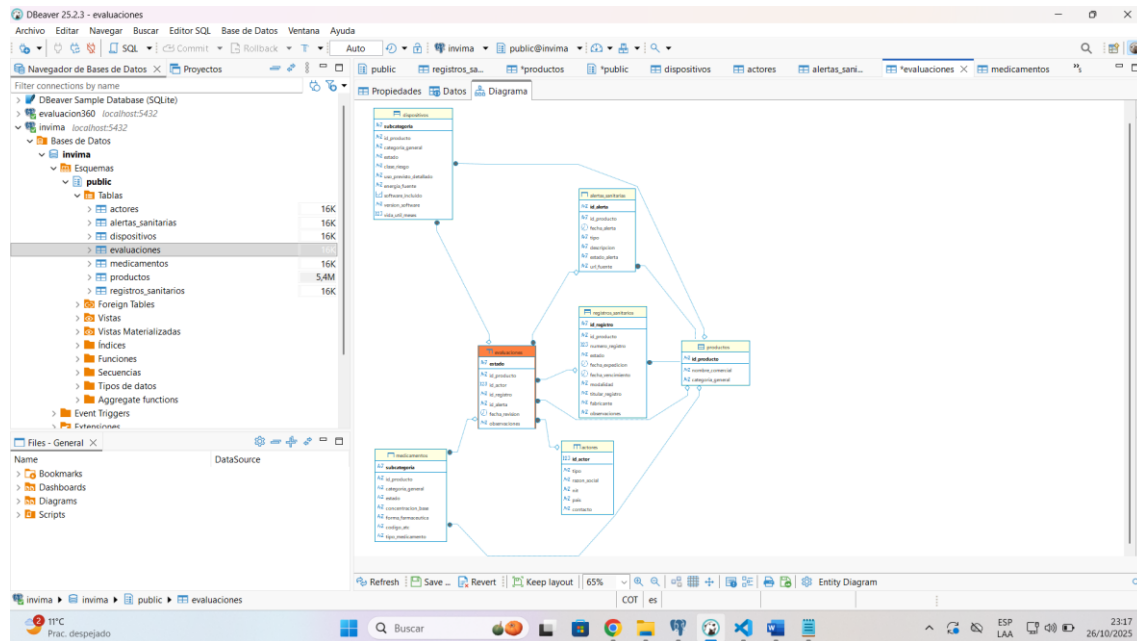
Jhonathan David Guevara Ramirez

SEMILLERO

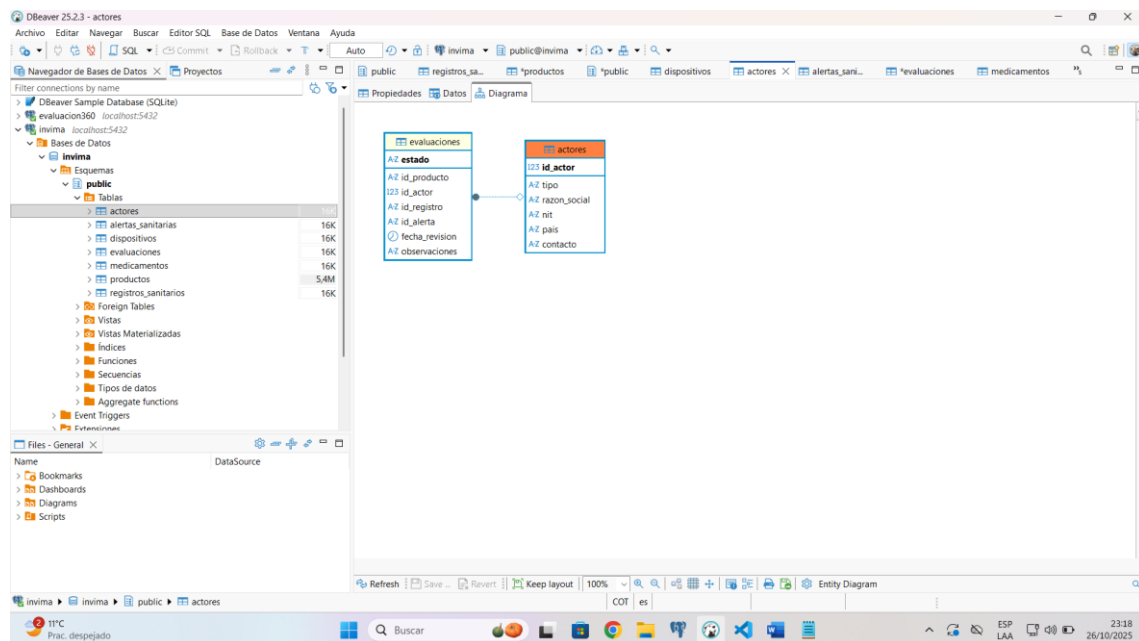
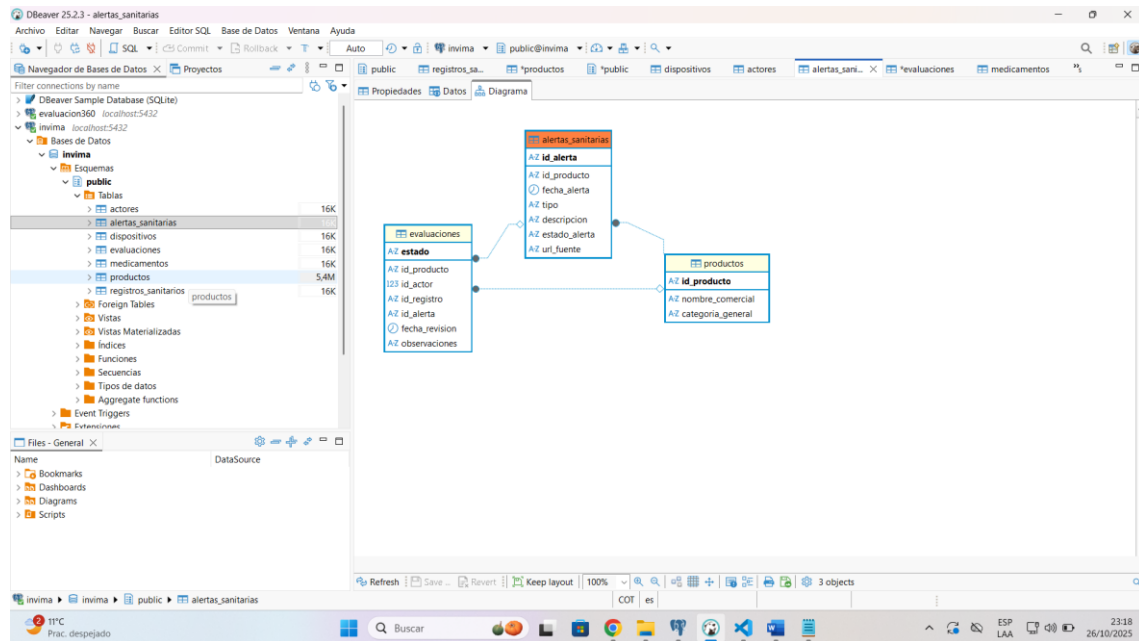


Jhonathan David Guevara Ramirez

SEMILLERO



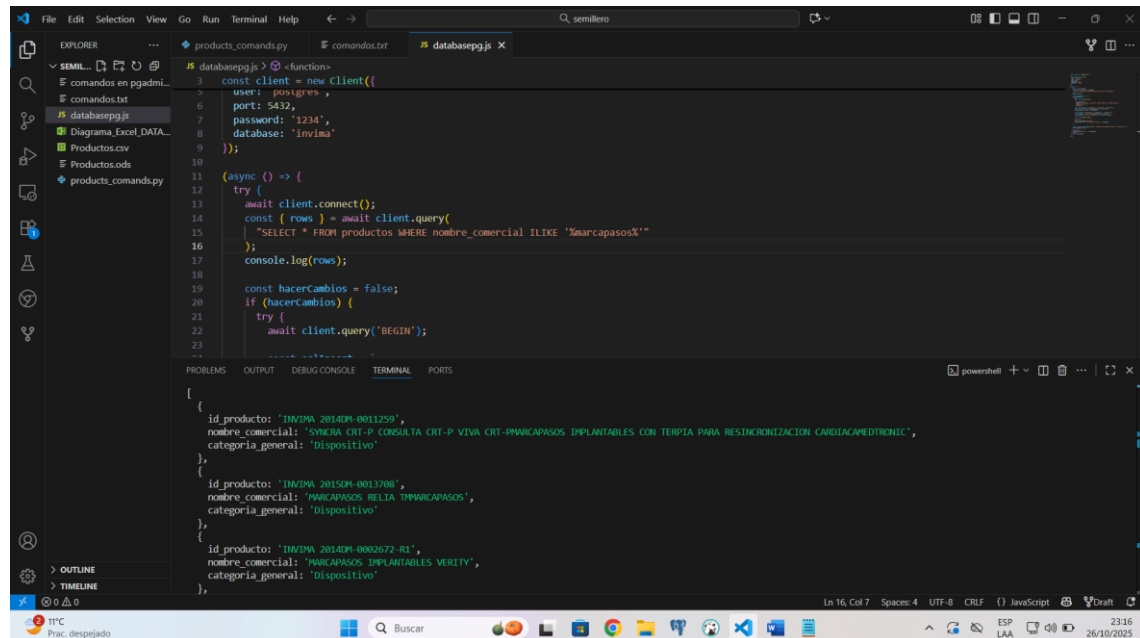
Jhonathan David Guevara Ramirez SEMILLERO



Jhonathan David Guevara Ramirez
SEMILLERO

CONSULTAS OFICIALES DESDE LA API A LA BASE DE DATOS FASE 1

ENFOQUE FASE 2, 3, 4



```

const client = new Client({
  user: 'postgres',
  port: 5432,
  password: '1234',
  database: 'invima'
});

(async () => {
  try {
    await client.connect();
    const { rows } = await client.query(
      "SELECT * FROM productos WHERE nombre_comercial ILIKE '%marcapasos%'"
    );
    console.log(rows);

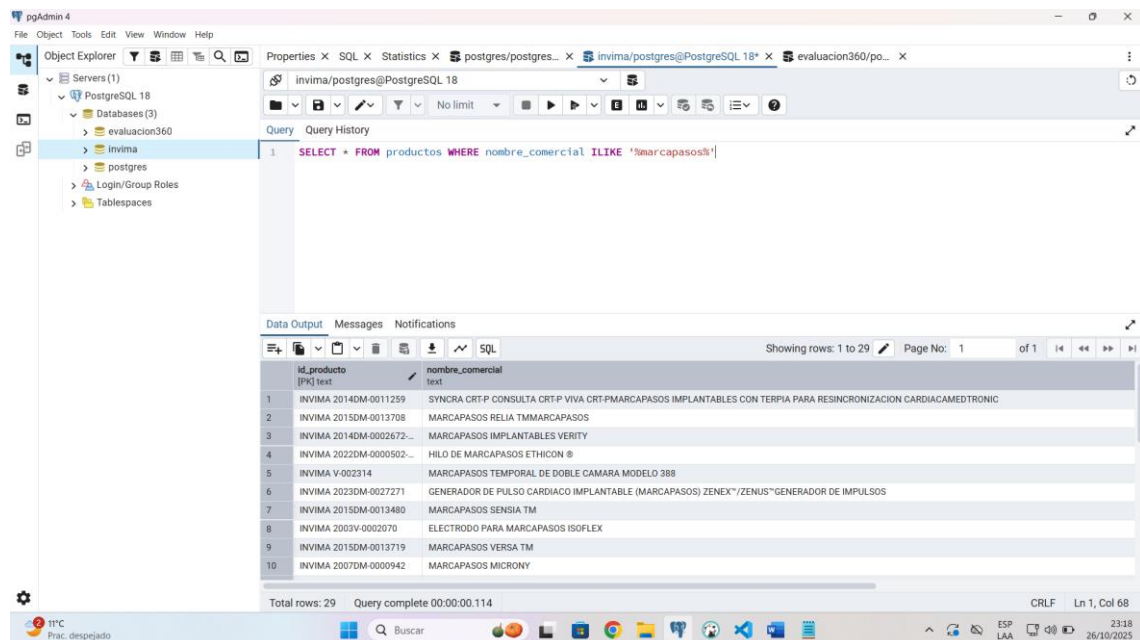
    const hacerCambios = false;
    if (hacerCambios) {
      try {
        await client.query('BEGIN');
      } catch (error) {
        console.log(error);
      }
    }
  } catch (error) {
    console.log(error);
  }
})();

```

```

[
  {
    id_producto: 'INVIMA 2014DM-0011259',
    nombre_comercial: 'SYNCR CRT-P CONSULTA CRT-P VIVA CRT-PMARCAPASOS IMPLANTABLES CON TERPIA PARA RESINCRONIZACION CARDIACAMEDTRONIC',
    categoria_general: 'Dispositivo'
  },
  {
    id_producto: 'INVIMA 2015DM-0013708',
    nombre_comercial: 'MARCAPASOS RELIA TMMARCAPASOS',
    categoria_general: 'Dispositivo'
  },
  {
    id_producto: 'INVIMA 2014DM-0002672-R1',
    nombre_comercial: 'MARCAPASOS IMPLANTABLES VERITY',
    categoria_general: 'Dispositivo'
  }
]

```

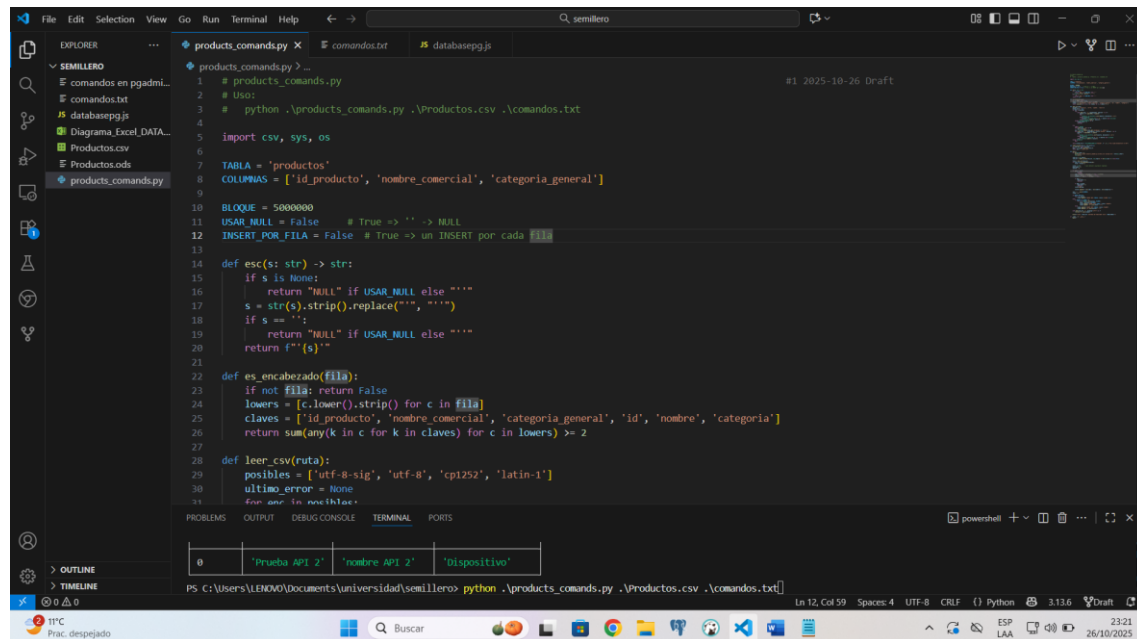


id_producto [PK] text	nombre_comercial text
INVIMA 2014DM-0011259	SYNCR CRT-P CONSULTA CRT-P VIVA CRT-PMARCAPASOS IMPLANTABLES CON TERPIA PARA RESINCRONIZACION CARDIACAMEDTRONIC
INVIMA 2015DM-0013708	MARCAPASOS RELIA TMMARCAPASOS
INVIMA 2014DM-0002672-R1	MARCAPASOS IMPLANTABLES VERITY
INVIMA 2022DM-0000502-...	HILO DE MARCAPASOS ETHICON ®
INVIMA V-002314	MARCAPASOS TEMPORAL DE DOBLE CAMARA MODELO 388
INVIMA 2023DM-0027271	GENERADOR DE PULSO CARDIACO IMPLANTABLE (MARCAPASOS) ZENEX™/ZENUS™GENERADOR DE IMPULSOS
INVIMA 2015DM-0013480	MARCAPASOS SENSIA TM
INVIMA 2003V-0002070	ELECTRODO PARA MARCAPASOS ISOFLX
INVIMA 2015DM-0013719	MARCAPASOS VERSA TM
INVIMA 2007DM-0000942	MARCAPASOS MICRONY

Jhonathan David Guevara Ramirez
SEMILLERO

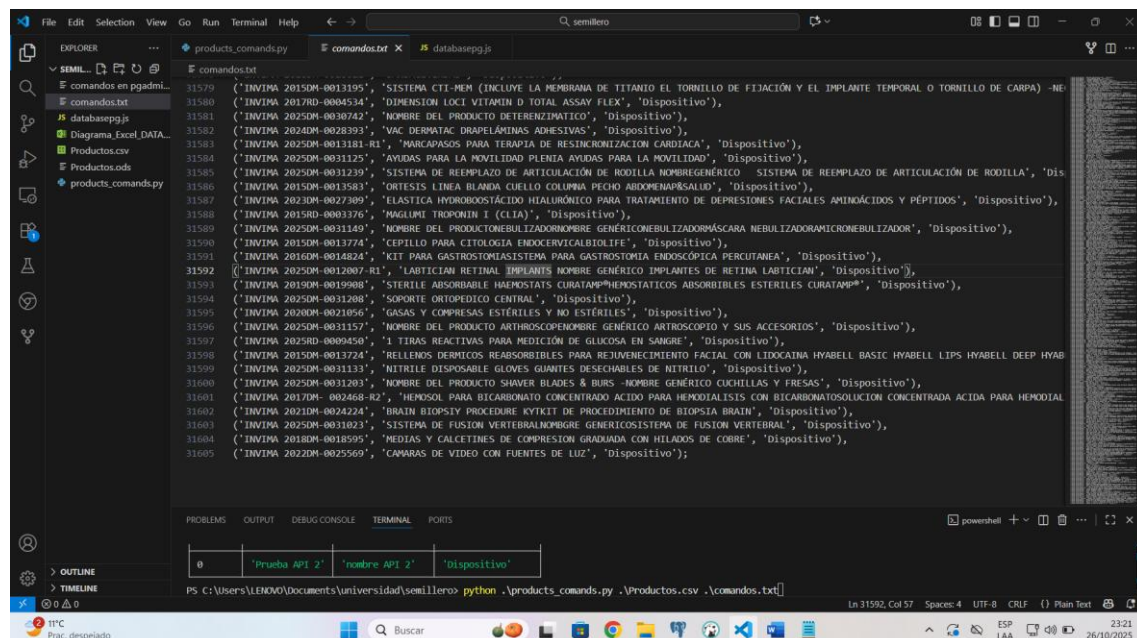
SCRIPT DE PYTHON CAPAZ DE COGER TODO LO DEL CSV Y CONVERTIRLO EN
COMANDOS PGADMIN O MYSQL, PRIMER INDICIO HACIA PRODUCTOS FASE 1

ENFOQUE A ALIMENTAR DATABASE FASE 2,3,4



```

1 # products_comands.py
2 # Uso:
3 # python .\products_comands.py .\Productos.csv .\comandos.txt
4
5 import csv, sys, os
6
7 TABLA = 'productos'
8 COLUMNAS = ['id_producto', 'nombre_comercial', 'categoria_general']
9
10 BLOQUE = 5000000
11 USAR_NULL = False # True -> '' -> NULL
12 INSERT_POR_FILA = False # True -> un INSERT por cada fila
13
14 def esc(s: str) -> str:
15     if s is None:
16         return "NULL" if USAR_NULL else ""
17     s = str(s).strip().replace("'", "")
18     if s == "":
19         return "NULL" if USAR_NULL else ""
20     return f"'{s}'"
21
22 def es_encabezado(fila):
23     if not fila: return False
24     lowers = [c.lower().strip() for c in fila]
25     claves = ['id_producto', 'nombre_comercial', 'categoria_general', 'id', 'nombre', 'categoria']
26     return sum(any(k in c for k in claves) for c in lowers) >= 2
27
28 def leer_csv(ruta):
29     posibles = ['utf-8-sig', 'utf-8', 'cp1252', 'latin-1']
30     ultimo_error = None
31     for enc in posibles:
32         try:
33             with open(ruta, 'r', encoding=enc) as f:
34                 reader = csv.reader(f)
35                 filas = list(reader)
36                 return filas
37         except UnicodeDecodeError as e:
38             ultimo_error = e
39     raise ultimo_error
40
41 if __name__ == '__main__':
42     if len(sys.argv) != 3:
43         print("Uso: python .\products_comands.py .\Productos.csv .\comandos.txt")
44         sys.exit(1)
45     ruta_csv, ruta_txt = sys.argv[1], sys.argv[2]
46     filas = leer_csv(ruta_csv)
47     with open(ruta_txt, 'w') as f:
48         for i, fila in enumerate(filas):
49             if es_encabezado(fila):
50                 continue
51             if INSERT_POR_FILA:
52                 sql = f"INSERT INTO {TABLA} ({', '.join(COLUMNAS)}) VALUES ({', '.join(esc(c) for c in fila)})"
53             else:
54                 sql = f"INSERT INTO {TABLA} ({', '.join(COLUMNAS)}) VALUES ({', '.join(esc(c) for c in fila)})"
55             f.write(sql + "\n")
56             if i % BLOQUE == 0:
57                 print(f"Procesados {i} filas")
58     print(f"Se generaron {len(filas) - 1} comandos en {ruta_txt}")
  
```



```

31579 ('INVIMA 2015DM-0013195', 'SISTEMA CIT-MER (INCLUYE LA MEMBRANA DE TITANIO EL TORNILLO DE FIJACIÓN Y EL IMPLANTE TEMPORAL O TORNILLO DE CARPA) -ME-
31580 ('INVIMA 2017RD-0004534', 'DIMENSION LOC VITAMIN D TOTAL ASSAY FLEX', 'Dispositivo'),
31581 ('INVIMA 2025DM-0030742', 'NOMBRE DEL PRODUCTO DETERENZIMATICO', 'Dispositivo'),
31582 ('INVIMA 2024DM-0028393', 'VAC DERMATAC DRAPELÁMINAS ADHESIVAS', 'Dispositivo'),
31583 ('INVIMA 2025DM-0013181-R1', 'MARCAPASOS PARA TERAPIA DE RESINCRONIZACION CARDIACA', 'Dispositivo'),
31584 ('INVIMA 2025DM-0031125', 'AYUDAS PARA LA MOVILIDAD PLENIA AYUDAS PARA LA MOVILIDAD', 'Dispositivo'),
31585 ('INVIMA 2025DM-0031239', 'SISTEMA DE REEMPLAZO DE ARTICULACIÓN DE RODILLA NOMBREGENÉRICO SISTEMA DE REEMPLAZO DE ARTICULACIÓN DE RODILLA', 'Dis-
31586 ('INVIMA 2015DM-0013583', 'ORTESIS LÍNEA BLANDA CUELLO COLUMNA PECO ABDOMENAPRSALUD', 'Dispositivo'),
31587 ('INVIMA 2023DM-0027389', 'ELÁSTICA HYDROSTACÍDICO HIALURÓNICO PARA TRATAMIENTO DE DEPRESIONES FACIALES AMINOÁCIDOS Y PÉPTIDOS', 'Dispositivo'),
31588 ('INVIMA 2015RD-0003326', 'MAGLUMI TROPOMIN I (CLIA)', 'Dispositivo'),
31589 ('INVIMA 2025DM-0031149', 'NOMBRE DEL PRODUCTONEBULIZADORNOMBRE GENÉRICONEBULIZADORMÁSCARA NEBULIZADORAMICRONEBULIZADOR', 'Dispositivo'),
31590 ('INVIMA 2015DM-0013774', 'CEPILLO PARA CITOLOGÍA ENDOCERVICALBIOLIFE', 'Dispositivo'),
31591 ('INVIMA 2016DM-0014824', 'KIT PARA GASTROSTOMIASISTEMA PARA GASTROSTOMIA ENDOSCÓPICA PERCUTÁNEA', 'Dispositivo'),
31592 ('INVIMA 2025DM-0012007-R1', 'LABITICIAN RETINAL IMPLANTS NOMBRE GENÉRICO IMPLANTES DE RETINA LABITICIAN', 'Dispositivo'),
31593 ('INVIMA 2019DM-0019908', 'STERILE ABSORBABLE HAEMOSTATS CURATAMPHEMOSTATICS ABSORBIBLES ESTERILES CURATAMP*', 'Dispositivo'),
31594 ('INVIMA 2025DM-0031288', 'SOPORTE ORTOPÉDICO CENTRAL', 'Dispositivo'),
31595 ('INVIMA 2020DM-0021656', 'GASAS Y COMPRESAS ESTÉRILES Y NO ESTÉRILES', 'Dispositivo'),
31596 ('INVIMA 2025DM-0031157', 'NOMBRE DEL PRODUCTO ARTHROSCOPYNOMBRE GENÉRICO ARTHROSCOPIO Y SUS ACCESORIOS', 'Dispositivo'),
31597 ('INVIMA 2025RD-0009450', '1 TIRAS REACTIVAS PARA MEDICIÓN DE GLUCOSA EN SANGRE', 'Dispositivo'),
31598 ('INVIMA 2015DM-0013724', 'RELLENOS DERMICOS REABSORBIBLES PARA REJUVENECIMIENTO FACIAL CON LIDOCAINA HYABELL BASIC HYABELL LIPS HYABELL DEEP HYAB
31599 ('INVIMA 2025DM-0031133', 'NITRILE DISPOSABLE GLOVES GUANTES DESECHABLES DE NITRILLO', 'Dispositivo'),
31600 ('INVIMA 2025DM-0031203', 'NOMBRE DEL PRODUCTO SHAVER BLADES & BURS -NOMBRE GENÉRICO CUCHILLAS Y FRESAS', 'Dispositivo'),
31601 ('INVIMA 2017DM-002468-R2', 'HEMOSOL PARA BICARBONATO CONCENTRADO ACIDO PARA HEMODIALISIS CON BICARBONATOSOLUCION CONCENTRADA ACIDA PARA HEMODIAL
31602 ('INVIMA 2021DM-0024224', 'BRAIN BIOPSY PROCEDURE KYKIT DE PROCEDIMIENTO DE BIOPSIA BRAIN', 'Dispositivo'),
31603 ('INVIMA 2025DM-0031023', 'SISTEMA DE FUSION VERTEBRALNOMBRE GENÉRICOSISTEMA DE FUSION VERTEBRAL', 'Dispositivo'),
31604 ('INVIMA 2018DM-0018595', 'MEDIAS Y CALCETINES DE COMPRESION GRADUADA CON HILADOS DE COBRE', 'Dispositivo'),
31605 ('INVIMA 2022DM-0025569', 'CÁMARAS DE VIDEO CON FUENTES DE LUZ', 'Dispositivo');
  
```

Jhonathan David Guevara Ramirez
SEMILLERO

COMANDOS PARA PGADMIN DATABASE

-- Crear base de datos invima --

```
CREATE DATABASE invima;
```

-- DROP es para eliminar TABLAS Y BASES DE DATOS si la cago --

```
DROP TABLE productos;
```

```
DROP TABLE registros_sanitarios;
```

-- Crear la tabla productos lo mismo con las de abajo--

-- OJO QUE TENEMOS RESTRICCIONES ESTO PARA QUE NO LA EMBARREMOS --

```
CREATE TABLE productos (
```

```
    id_producto                text NOT NULL,
```

```
    nombre_comercial           text NOT NULL,
```

```
    categoria_general           text NOT NULL
```

```
);
```

```
CREATE TABLE registros_sanitarios (
```

```
    id_registro                text NOT NULL,
```

```
    id_producto                text NOT NULL,
```

```
    numero_registro            numeric CHECK (numero_registro>0) NOT NULL,
```

```
    estado                     text CONSTRAINT estado_valido
```

```
CHECK(estado IN ('vencido', 'no vencido')) DEFAULT 'no vencido',
```

```
    fecha_expedicion           DATE NOT NULL,
```

```
    fecha_vencimiento          DATE NOT NULL,
```

```
    modalidad                  text NOT NULL,
```

```
    titular_registro           text NOT NULL,
```

```
    fabricante                  text NOT NULL,
```



Jhonathan David Guevara Ramirez
SEMILLERO

```
observaciones          text NOT NULL

);

-- INSERTAR INFORMACION ESTO LO HAREMOS MAS ADELANTE CON UN SCRIPT
DE PYTHON QUE HAGA ESTO YA TENIENDO EL COMANDO BASE--

INSERT INTO productos (id_producto,    nombre_comercial, categoria_general)
VALUES

    ('10e372', 'NADOREX',    'Medicamento' ),

    ('12e518', 'Noxpirin Plus' , 'Medicamento' );

-- Actualizar datos

UPDATE productos SET nombre_comercial = 'Acetaminofén' WHERE id_producto =
'10e372';

-- BORRAR PRODUCTOS

DELETE FROM productos WHERE nombre_comercial = 'Acetaminofén';

DELETE FROM productos WHERE nombre_comercial = 'Noxpirin Plus';

--Consultar esto para que luego con la API se consulte breve

SELECT * FROM productos;

SELECT * FROM productos ORDER BY id_producto;

SELECT * FROM productos WHERE nombre_comercial = 'Acetaminofén' ORDER BY
id_producto;

-- ALTER osea cambios

-- QUITAR NO NULO
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
ALTER TABLE productos ALTER COLUMN id_producto DROP NOT NULL;
```

```
-- AGREGAR NO NULO
```

```
ALTER TABLE productos ALTER COLUMN id_producto SET NOT NULL;
```

```
-- CAMBIAR NOMBRE DE LA COLUMNA
```

```
ALTER TABLE productos RENAME COLUMN id_producto TO identificador;
```

```
-- RENOMBRAR UNA TABLA
```

```
ALTER TABLE productos_invima RENAME TO productos;
```

```
-- AÑADIR PRIMARY KEY
```

```
ALTER TABLE productos ADD PRIMARY KEY (id_producto);
```

```
-- AÑADIR UN FOREIGN KEY
```

```
ALTER TABLE registros_sanitarios ADD FOREIGN KEY (id_producto) REFERENCES  
productos(id_producto);
```

```
-- AÑADIR UN CHECK A CATEGORIA PRINCIPAL DE PRODUCTOS PARA EL CHECK  
DE MEDICAMENTOS Y ASI
```

```
ALTER TABLE productos ADD CONSTRAINT categoria_general_valido CHECK  
(categoria_general IN ('Medicamento','Dispositivo','Alimento','Cosmetico'));
```

```
ALTER TABLE registros_sanitarios ADD PRIMARY KEY (id_registro);
```

```
CREATE TABLE alertas_sanitarias (
```

```
    id_alerta          text NOT NULL PRIMARY KEY,
```

```
    id_producto        text NOT NULL REFERENCES productos(id_producto),
```

```
    fecha_alerta       DATE NOT NULL DEFAULT CURRENT_DATE,
```

```
    tipo               text CONSTRAINT tipo_valido CHECK (tipo IN  
( 'Medicamentos', 'Biológicos', 'Alimentos', 'Dispositivos', 'Cosméticos')) DEFAULT  
'No específico',
```

```
    descripcion        text NOT NULL,
```

```
    estado_alerta      text CONSTRAINT estado_alerta_valido CHECK  
(estado_alerta IN ('Abierto', 'En curso', 'Cerrado')) DEFAULT 'cerrado',
```



Jhonathan David Guevara Ramirez
SEMILLERO

```
url_fuente          text NOT NULL  
  
);
```

```
CREATE TABLE medicamentos (  
    id_producto          text REFERENCES  
productos(id_producto),  
    categoria_general    text CHECK (categoria_general IN  
('Medicamento')) DEFAULT 'Medicamento',  
    estado              text REFERENCES  
evaluaciones(estado),  
    subcategoria        text NOT NULL PRIMARY KEY,  
    concentracion_base  text NOT NULL,  
    forma_farmaceutica  text NOT NULL,  
    codigo_atc          text NOT NULL,  
    tipo_medicamento   text NOT NULL CHECK (tipo_medicamento IN  
('quimico','biologico','biotecnologico','fitoterapeutico','homeopatico','radiofarmaco'))  
);
```

```
CREATE TABLE dispositivos (  
    id_producto          text REFERENCES  
productos(id_producto),  
    categoria_general    text CHECK (categoria_general IN ('Dispositivo'))  
DEFAULT 'Dispositivo',  
    estado              text REFERENCES  
evaluaciones(estado),  
    subcategoria        text NOT NULL PRIMARY KEY,  
    clase_riesgo        text NOT NULL CHECK (clase_riesgo IN ('I','IIa','IIb','III')),
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
    uso_previsto_detallado text NOT NULL,  
    energia_fuente      text NOT NULL,  
    software_incluido   boolean NOT NULL,  
    version_software    text NOT NULL,  
    vida_util_meses     integer NOT NULL  
);
```

```
CREATE TABLE actores (  
    id_actor          numeric PRIMARY KEY CHECK (id_actor>0) NOT NULL,  
    tipo              text NOT NULL,  
    razon_social text NOT NULL,  
    nit               text NOT NULL,  
    pais              text NOT NULL,  
    contacto          text NOT NULL  
);
```

```
CREATE TABLE evaluaciones (  
    id_producto text REFERENCES productos(id_producto),  
    id_actor     numeric REFERENCES actores(id_actor),  
    id_registro  text REFERENCES registros_sanitarios(id_registro),  
    id_alerta    text REFERENCES alertas_sanitarias(id_alerta),  
    estado       text PRIMARY KEY NOT NULL CHECK (estado IN  
( 'buenas condiciones', 'en uso', 'no conforme', 'vencido', 'malas condiciones',  
'aceptable')) DEFAULT 'no conforme',  
    fecha_revision DATE NOT NULL,  
    Observaciones text
```



Jhonathan David Guevara Ramirez
SEMILLERO

);

CREATE TABLE

DROP TABLE evaluaciones;

DROP TABLE dispositivos;

DROP TABLE medicamentos;

--- ESTE COMANDO PARA HACER CONSULTAS MEJORES

SELECT * FROM productos WHERE nombre_comercial ILIKE '%marcapasos%'

-- SI INCLUYE EL NOMBRE MARCAPASOS CON MAYUSCULA O MINUSCULA
ENCUENTRA

SCRIPT DE PYTHON GENERADOR DE COMANDOS PGADMIN

products_comands.py

Uso:

python .\products_comands.py .\Productos.csv .\comandos.txt

import csv, sys, os

TABLA = 'productos'

COLUMNAS = ['id_producto', 'nombre_comercial', 'categoria_general']

Jhonathan David Guevara Ramirez
SEMILLERO

BLOQUE = 5000000

USAR_NULL = False # True => " -> NULL

INSERT_POR_FILA = False # True => un INSERT por cada fila

```
def esc(s: str) -> str:
```

```
    if s is None:
```

```
        return "NULL" if USAR_NULL else ""
```

```
    s = str(s).strip().replace("", "")
```

```
    if s == ":
```

```
        return "NULL" if USAR_NULL else ""
```

```
    return f"{{s}}"
```

```
def es_encabezado(fila):
```

```
    if not fila: return False
```

```
    lowers = [c.lower().strip() for c in fila]
```

```
    claves = ['id_producto', 'nombre_comercial', 'categoria_general', 'id', 'nombre',  
'categoria']
```

```
    return sum(any(k in c for k in claves) for c in lowers) >= 2
```

```
def leer_csv(ruta):
```

```
    posibles = ['utf-8-sig', 'utf-8', 'cp1252', 'latin-1']
```

```
    ultimo_error = None
```

```
    for enc in posibles:
```

```
        try:
```

```
            with open(ruta, 'r', encoding=enc, newline="") as f:
```

```
                muestra = f.read(4096); f.seek(0)
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
    try:

        dialect = csv.Sniffer().sniff(muestra, delimiters=";;\t|")

    except Exception:

        dialect = csv.excel_tab if '\t' in muestra else csv.excel

    return list(csv.reader(f, dialect))

except UnicodeDecodeError as e:

    ultimo_error = e

    continue

try:

    import chardet

    with open(ruta, 'rb') as fb:

        raw = fb.read()

    enc = (chardet.detect(raw).get('encoding') or 'latin-1')

    with open(ruta, 'r', encoding=enc, errors='replace', newline='') as f:

        muestra = f.read(4096); f.seek(0)

        try:

            dialect = csv.Sniffer().sniff(muestra, delimiters=";;\t|")

        except Exception:

            dialect = csv.excel_tab if '\t' in muestra else csv.excel

        return list(csv.reader(f, dialect))

except Exception:

    pass

raise ultimo_error or UnicodeDecodeError("decode", b"", 0, 1, "No se pudo  
decodificar el CSV")
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
def nombre_salida(ruta_in, ruta_out=None):  
    if ruta_out: return ruta_out  
    base, _ = os.path.splitext(ruta_in)  
    return base + '_insert.txt'  
  
def main():  
    if len(sys.argv) < 2:  
        print("Uso: python products_comands.py archivo.csv [salida.txt]",  
file=sys.stderr)  
        sys.exit(1)  
  
    ruta_csv = sys.argv[1]  
    ruta_txt = nombre_salida(ruta_csv, sys.argv[2] if len(sys.argv) >= 3 else None)  
  
    filas = leer_csv(ruta_csv)  
    if filas and es_encabezado(filas[0]):  
        filas = filas[1:]  
  
    vistos = set()    # para detectar id_producto repetidos  
    duplicadas = 0  
    valores = []  
  
    for fila in filas:  
        if not fila:  
            continue
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
idp = (fila[0] if len(fila) > 0 else "").strip()
nombre = fila[1] if len(fila) > 1 else ""
categoria = fila[2] if len(fila) > 2 else ""

if idp == "":
    duplicadas += 1
    continue

if idp in vistos:
    duplicadas += 1
    continue
vistos.add(idp)

valores.append(f"({esc(idp)}, {esc(nombre)}, {esc(categoria)})")

cols = ", ".join(COLUMNAS)
lineas = []

if INSERT_POR_FILA:
    for v in valores:
        lineas.append(f"INSERT INTO {TABLA} ({cols}) VALUES {v};")
else:
    if BLOQUE and BLOQUE > 0:
        for i in range(0, len(valores), BLOQUE):
            chunk = valores[i:i+BLOQUE]
            lineas.append(f"INSERT INTO {TABLA} ({cols}) VALUES")
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
        lineas.append(",\n".join(chunk) + ";")
    else:
        lineas.append(f"INSERT INTO {TABLA} ({cols}) VALUES")
        lineas.append(",\n".join(valores) + ";")

with open(ruta_txt, 'w', encoding='utf-8') as f:
    f.write("\n".join(lineas))

print(f"Listo: {ruta_txt} (omitidas por duplicado/ vacío: {duplicadas})")

if __name__ == "__main__":
    main()
```

API CONEXIÓN ENTRE BASE DE DATOS Y NODE JS LUEGO HACER WEB SERVICE

```
const { Client } = require('pg');

const client = new Client({
  host: 'localhost',
  user: 'postgres',
  port: 5432,
  password: '1234',
  database: 'invima'
});
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
(async () => {  
  try {  
    await client.connect();  
    const { rows } = await client.query(  
      "SELECT * FROM productos WHERE nombre_comercial ILIKE '%marcapasos%'"  
    );  
    console.log(rows);  
  
    const hacerCambios = false;  
    if (hacerCambios) {  
      try {  
        await client.query('BEGIN');  
  
        const sqlInsert = `  
          INSERT INTO productos (id_producto, nombre_comercial, categoria_general)  
          VALUES ($1, $2, $3)  
          RETURNING *;  
        `;  
        const valoresInsert = ['Prueba API', 'nombre API', 'Dispositivo'];  
        const ins = await client.query(sqlInsert, valoresInsert);  
        console.log('Insertado:', ins.rows[0]);  
  
        const insert2 = ['Prueba API 2', 'nombre API 2', 'Dispositivo'];  
        const insertar2 = await client.query(sqlInsert, insert2);  
        console.log('Exito insertar segunda cosa', insertar2.rows[0]);
```

Jhonathan David Guevara Ramirez
SEMILLERO

```
    await client.query('COMMIT');

    } catch (e) {
        await client.query('ROLLBACK');
        console.error('Error en cambios (rollback):', e.message);
    }
}

const r2 = await client.query("SELECT * FROM productos WHERE id_producto =
'Prueba API 2'");

console.table(r2.rows);

} catch (err) {
    console.error('DB error:', err.message);
} finally {
    await client.end();
}
})();
```