



# Trabajo de desarrollo

## Desarrollo de Aplicación de Consola

---

# Informe de Desarrollo de Aplicación de Consola

## Portada

**Título:** Desarrollo de Aplicación de Consola para Gestión de Clínica

**Autores:**

- Luis Alfredo Cereño Campos
- Fábian Antonio Fariás Cubillos
- Camilo Roberto Maldonado Valderrama
- Alejandro Isaac Sanchez Candia

**Fecha:** 20-04-2025

## Introducción

Este informe detalla el proceso de desarrollo de una aplicación de consola en Python para la gestión de una clínica, utilizando una base de datos PostgreSQL. El objetivo principal fue diseñar e implementar un sistema que permita la gestión eficiente de pacientes, médicos, camas y exámenes en un entorno clínico.

## Desarrollo

### Objetivos de la Tarea

El objetivo principal de esta tarea fue diseñar e implementar una aplicación de consola utilizando Python y PostgreSQL para gestionar una clínica. Esto incluye:

- Diseñar un modelo relacional de la base de datos.
- Crear la base de datos en un servidor PostgreSQL.
- Generar un script para poblar la base de datos.
- Implementar una aplicación de consola con diversas funcionalidades.

### Prerrequisitos

Para desarrollar esta tarea, se requirió:

- Conocimiento del modelo relacional y normalización.
- Comprensión de SQL y su sintaxis.
- Capacidad de programación en Python.
- Instalación de Miniconda con las librerías necesarias.
- Instalación de PostgreSQL en el computador.

### Instrucciones para la Elaboración

La tarea se realizó en equipo de 4 estudiantes, siguiendo estos pasos:

1. **Revisión del caso:** Se analizó el caso de la clínica para entender las relaciones entre pacientes, médicos, camas, habitaciones y exámenes.
2. **Diseño del modelo relacional:** Se diseñó un modelo relacional que reflejara las entidades y relaciones descritas en el caso.
3. **Creación de la base de datos:** Se creó la base de datos en PostgreSQL siguiendo el modelo diseñado.
4. **Script de población:** Se generó un script SQL para insertar al menos 10 registros por tabla.

5. **Implementación de la aplicación de consola:** Se desarrolló una aplicación en Python con las siguientes funcionalidades:

- Menú de acceso a funcionalidades.
- Mostrar pacientes de la clínica.
- Mostrar detalle de un paciente por RUT.
- Cambiar a un paciente de cama.
- Cambiar a un paciente de médico.
- Crear camas y habitaciones.
- Desplegar todas las tablas (nueva opción).

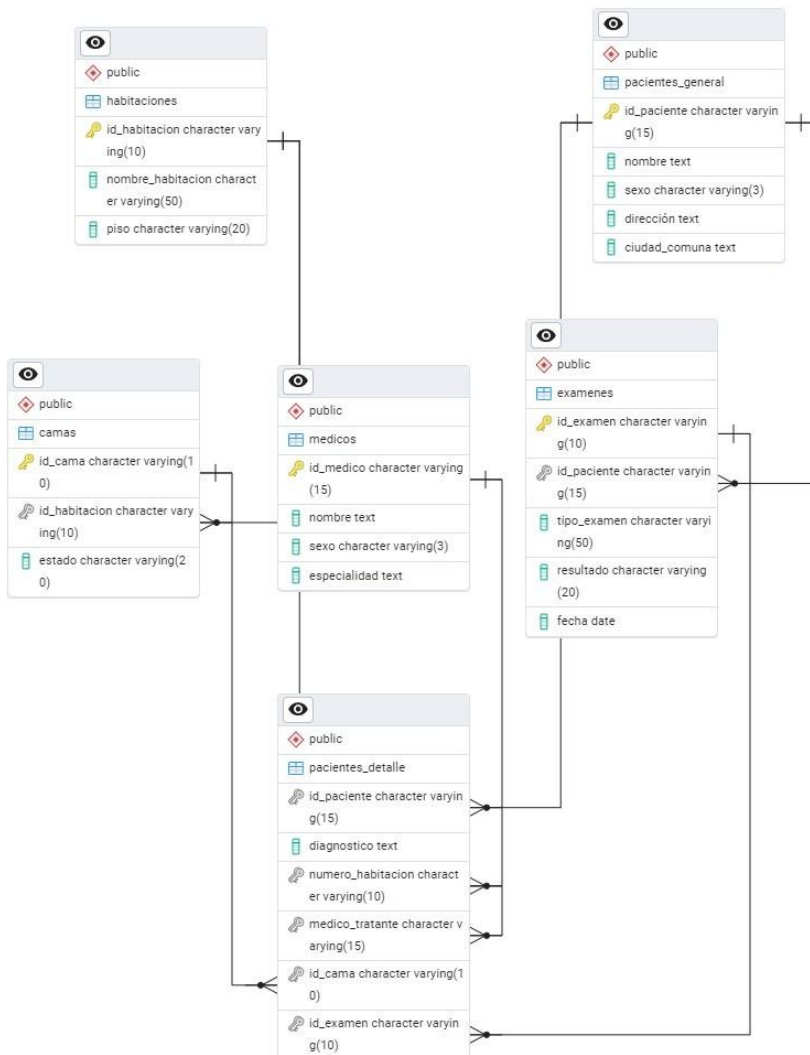
## Modelo Relacional

El modelo relacional diseñado para la clínica incluye las siguientes tablas:

- **habitaciones:** id\_habitacion, nombre\_habitacion, piso
- **camas:** id\_cama, id\_habitacion, estado
- **pacientes\_general:** id\_paciente, Nombre, Sexo, Dirección, Ciudad\_Comuna
- **medicos:** id\_medico, Nombre, Sexo, Especialidad
- **exámenes:** id\_examen, id\_paciente, tipo\_examen, resultado, fecha
- **pacientes\_detalle:** id\_paciente, Diagnostico, numero\_habitacion, medico\_tratante, id\_cama, id\_examen

## Diagrama Entidad-Relación

A continuación se presenta el diagrama Entidad-Relación del modelo de datos diseñado:



## Código SQL

```
DROP TABLE IF EXISTS pacientes_detalle;
DROP TABLE IF EXISTS examenes;
DROP TABLE IF EXISTS medicos;
DROP TABLE IF EXISTS pacientes_general;
DROP TABLE IF EXISTS camas;
DROP TABLE IF EXISTS habitaciones;

CREATE TABLE habitaciones (
    id_habitacion VARCHAR(10) PRIMARY KEY,
    nombre_habitacion VARCHAR(50),
    piso VARCHAR(20)
);

CREATE TABLE camas (
    id_cama VARCHAR(10) PRIMARY KEY,
    id_habitacion VARCHAR(10) REFERENCES habitaciones(id_habitacion),
    estado VARCHAR(20)
);

CREATE TABLE pacientes_general (
    id_paciente VARCHAR(15) PRIMARY KEY,
    Nombre TEXT,
    Sexo VARCHAR(3),
    Dirección TEXT,
    Ciudad_Comuna TEXT
);

CREATE TABLE medicos (
    id_medico VARCHAR(15) PRIMARY KEY,
    Nombre TEXT,
    Sexo VARCHAR(3),
    Especialidad TEXT
);

CREATE TABLE examenes (
    id_examen VARCHAR(10) PRIMARY KEY,
    id_paciente VARCHAR(15) REFERENCES pacientes_general(id_paciente),
    tipo_examen VARCHAR(50),
    resultado VARCHAR(20),
    fecha DATE
);

CREATE TABLE pacientes_detalle (
    id_paciente VARCHAR(15) REFERENCES pacientes_general(id_paciente),
    Diagnostico TEXT,
    numero_habitacion VARCHAR(10) REFERENCES habitaciones(id_habitacion),
    medico_tratante VARCHAR(15) REFERENCES medicos(id_medico),
    id_cama VARCHAR(10) REFERENCES camas(id_cama),
    id_examen VARCHAR(10) REFERENCES examenes(id_examen)
);

-- Insertar habitaciones
INSERT INTO habitaciones VALUES ('101001', 'habitacion_101001', 'piso_1');
INSERT INTO habitaciones VALUES ('101002', 'habitacion_101002', 'piso_1');
INSERT INTO habitaciones VALUES ('101003', 'habitacion_101003', 'piso_1');
INSERT INTO habitaciones VALUES ('101004', 'habitacion_101004', 'piso_2');
```

```

INSERT INTO habitaciones VALUES ('101005', 'habitacion_101005', 'piso_2');
INSERT INTO habitaciones VALUES ('101006', 'habitacion_101006', 'piso_2');
INSERT INTO habitaciones VALUES ('101007', 'habitacion_101007', 'piso_3');
INSERT INTO habitaciones VALUES ('101008', 'habitacion_101008', 'piso_3');
INSERT INTO habitaciones VALUES ('101009', 'habitacion_101009', 'piso_3');
INSERT INTO habitaciones VALUES ('101010', 'habitacion_101010', 'piso_4');

-- Insertar camas
INSERT INTO camas VALUES ('C0001', '101009', 'Mantenimiento');
INSERT INTO camas VALUES ('C0002', '101001', 'Ocupada');
INSERT INTO camas VALUES ('C0003', '101006', 'Ocupada');
INSERT INTO camas VALUES ('C0004', '101009', 'Disponible');
INSERT INTO camas VALUES ('C0005', '101001', 'Ocupada');
INSERT INTO camas VALUES ('C0006', '101001', 'Ocupada');
INSERT INTO camas VALUES ('C0007', '101010', 'Disponible');
INSERT INTO camas VALUES ('C0008', '101002', 'Ocupada');
INSERT INTO camas VALUES ('C0009', '101010', 'Disponible');
INSERT INTO camas VALUES ('C0010', '101006', 'Mantenimiento');
INSERT INTO camas VALUES ('C0011', '101002', 'Mantenimiento');
INSERT INTO camas VALUES ('C0012', '101009', 'Disponible');
INSERT INTO camas VALUES ('C0013', '101003', 'Mantenimiento');
INSERT INTO camas VALUES ('C0014', '101001', 'Mantenimiento');
INSERT INTO camas VALUES ('C0015', '101010', 'Mantenimiento');
INSERT INTO camas VALUES ('C0016', '101009', 'Disponible');
INSERT INTO camas VALUES ('C0017', '101006', 'Disponible');
INSERT INTO camas VALUES ('C0018', '101009', 'Ocupada');
INSERT INTO camas VALUES ('C0019', '101004', 'Disponible');
INSERT INTO camas VALUES ('C0020', '101002', 'Mantenimiento');

-- Insertar pacientes_general
INSERT INTO pacientes_general VALUES ('5701779-1', 'Gonzalez Gonzalez Maria Teres');
INSERT INTO pacientes_general VALUES ('19995187-4', 'Gonzalez Gonzalez Miguel Ang');
INSERT INTO pacientes_general VALUES ('17475679-1', 'Gonzalez Gonzalez Nicole Est');
INSERT INTO pacientes_general VALUES ('5203909-6', 'Gonzalez Gonzalez Sara Isabel');
INSERT INTO pacientes_general VALUES ('15603250-6', 'Gonzalez Gonzalez Servando F');
INSERT INTO pacientes_general VALUES ('17083096-2', 'Perez De Arce Perez Pia Cons');
INSERT INTO pacientes_general VALUES ('4004562-7', 'Perez De Arce Perez Yolanda A');
INSERT INTO pacientes_general VALUES ('10369669-0', 'Perez De Tudela Perez Dean G');
INSERT INTO pacientes_general VALUES ('4660993-K', 'Perez Perez Alfredo Haroldo D');
INSERT INTO pacientes_general VALUES ('5093926-K', 'Perez Perez Alice Rosa Oriana');

-- Insertar medicos
INSERT INTO medicos VALUES ('8050976-6', 'Torres Torres Yorka Guacolda Del Buen C');
INSERT INTO medicos VALUES ('20353187-7', 'Torres Torres Abner Tomas Exequiel', '');
INSERT INTO medicos VALUES ('21419633-6', 'Torres Torres Alexander Ariel Timo', '');
INSERT INTO medicos VALUES ('16084382-9', 'Torres Torres Alfredo Miguel Angel', '');
INSERT INTO medicos VALUES ('16976065-9', 'Torres Torres Angela Esparta Jossabeth');
INSERT INTO medicos VALUES ('18665639-3', 'Garcia Garcia Alba Laureana Aracely', '');
INSERT INTO medicos VALUES ('17287413-4', 'Garcia Garcia Alejandro Juan Alexis', '');
INSERT INTO medicos VALUES ('4302025-0', 'Garcia Garcia Antonio Segundo Panfilo', '');
INSERT INTO medicos VALUES ('5038526-4', 'Garcia Garcia Aurelia Maria Del Carmen', '');
INSERT INTO medicos VALUES ('19001977-2', 'Garcia Garcia Ayleen Suiland Marcela', '');

-- Insertar examenes
INSERT INTO examenes VALUES ('E0001', '17475679-1', 'Resonancia', 'Pendiente', '2');
INSERT INTO examenes VALUES ('E0002', '5093926-K', 'Hemograma', 'Pendiente', '202');
INSERT INTO examenes VALUES ('E0003', '10369669-0', 'Tomografía', 'Pendiente', '2');
INSERT INTO examenes VALUES ('E0004', '10369669-0', 'Tomografía', 'Normal', '2024');
INSERT INTO examenes VALUES ('E0005', '4004562-7', 'Hemograma', 'Pendiente', '202');
INSERT INTO examenes VALUES ('E0006', '10369669-0', 'Hemograma', 'Pendiente', '20');

```

```

INSERT INTO examenes VALUES ('E0007', '19995187-4', 'Hemograma', 'Anormal', '202
INSERT INTO examenes VALUES ('E0008', '17475679-1', 'Resonancia', 'Pendiente', '2
INSERT INTO examenes VALUES ('E0009', '19995187-4', 'Resonancia', 'Anormal', '202
INSERT INTO examenes VALUES ('E0010', '17475679-1', 'Tomografía', 'Anormal', '202

-- Insertar pacientes_detalle
INSERT INTO pacientes_detalle VALUES ('19995187-4', 'Gripe', '101007', '16976065-
INSERT INTO pacientes_detalle VALUES ('17475679-1', 'Apendicitis', '101002', '186
INSERT INTO pacientes_detalle VALUES ('15603250-6', 'Fractura', '101002', '186656
INSERT INTO pacientes_detalle VALUES ('17475679-1', 'Apendicitis', '101004', '160
INSERT INTO pacientes_detalle VALUES ('4004562-7', 'Covid-19', '101003', '1866563
INSERT INTO pacientes_detalle VALUES ('17475679-1', 'Covid-19', '101001', '214196
INSERT INTO pacientes_detalle VALUES ('4004562-7', 'Fractura', '101005', '1697606
INSERT INTO pacientes_detalle VALUES ('5203909-6', 'Fractura', '101007', '2035318
INSERT INTO pacientes_detalle VALUES ('15603250-6', 'Fractura', '101009', '160843
INSERT INTO pacientes_detalle VALUES ('19995187-4', 'Gripe', '101003', '21419633-

```

## Código Python

A continuación se presenta el código Python desarrollado para la aplicación de consola, actualizado con la opción para desplegar todas las tablas:

```

import psycopg2
from tabulate import tabulate

# Configuración de la conexión a la base de datos
conn = psycopg2.connect(
    dbname="clinica",
    user="postgres",
    password="your_password",
    host="localhost",
    port="5432"
)
cur = conn.cursor()

def mostrar_pacientes():
    cur.execute("""
SELECT pg.id_paciente, pg.Nombre, pd.Diagnostico, m.Nombre AS medico_tratante
FROM pacientes_general pg
JOIN pacientes_detalle pd ON pg.id_paciente = pd.id_paciente
JOIN medicos m ON pd.medico_tratante = m.id_medico
JOIN camas c ON pd.id_cama = c.id_cama
JOIN habitaciones h ON c.id_habitacion = h.id_habitacion
""")
    pacientes = cur.fetchall()
    print(tabulate(pacientes, headers=["RUT", "Nombre", "Diagnóstico", "Médico Tr

def mostrar_detalle_paciente(rut):
    cur.execute("""
SELECT pg.id_paciente, pg.Nombre, pd.Diagnostico, m.Nombre AS medico_tratante
        e.tipo_examen, e.resultado, h.nombre_habitacion, c.id_cama
FROM pacientes_general pg
JOIN pacientes_detalle pd ON pg.id_paciente = pd.id_paciente
JOIN medicos m ON pd.medico_tratante = m.id_medico
JOIN camas c ON pd.id_cama = c.id_cama
JOIN habitaciones h ON c.id_habitacion = h.id_habitacion

```

```

LEFT JOIN examenes e ON pd.id_examen = e.id_examen
WHERE pg.id_paciente = %s
ORDER BY e.fecha DESC
LIMIT 1
""" , (rut,))
detalle = cur.fetchone()
if detalle:
    print(tabulate([detalle], headers=["RUT", "Nombre", "Diagnóstico", "Médico"])
else:
    print("Paciente no encontrado.")

def cambiar_cama(rut, nueva_cama):
    cur.execute("UPDATE pacientes_detalle SET id_cama = %s WHERE id_paciente = %s")
    conn.commit()
    print("Cama del paciente actualizada.")

def cambiar_medico(rut, nuevo_medico):
    cur.execute("UPDATE pacientes_detalle SET medico_tratante = %s WHERE id_paciente = %s")
    conn.commit()
    print("Médico del paciente actualizado.")

def crear_cama_y_habitacion(id_habitacion, nombre_habitacion, piso, id_cama, estado):
    cur.execute("INSERT INTO habitaciones (id_habitacion, nombre_habitacion, piso) VALUES (%s, %s, %s)"
                (id_habitacion, nombre_habitacion, piso))
    cur.execute("INSERT INTO camas (id_cama, id_habitacion, estado) VALUES (%s, %s, %s)"
                (id_cama, id_habitacion, estado))
    conn.commit()
    print("Cama y habitación creadas.")

def mostrar_todas_tablas():
    # Mostrar todas las filas de cada tabla para que el usuario revise la data
    tablas = ["habitaciones", "camas", "pacientes_general", "medicos", "examenes"]
    for tabla in tablas:
        print("\n--- Datos de la tabla:", tabla, "---")
        cur.execute("SELECT * FROM " + tabla + ";")
        datos = cur.fetchall()
        if datos:
            for fila in datos:
                print(fila)
        else:
            print("No hay datos en la tabla", tabla)
    print("-" * 40)

def menu():
    while True:
        print("\n--- Menú de Gestión de Clínica ---")
        print("1. Mostrar pacientes")
        print("2. Mostrar detalle de un paciente")
        print("3. Cambiar paciente de cama")
        print("4. Cambiar paciente de médico")
        print("5. Crear camas y habitaciones")
        print("6. Desplegar todas las tablas")
        print("0. Salir")

        opcion = input("Seleccione una opción: ")

        if opcion == '1':
            mostrar_pacientes()
        elif opcion == '2':
            rut = input("Ingrese el RUT del paciente: ")
            mostrar_detalle_paciente(rut)

```

```
elif opcion == '3':
    rut = input("Ingrese el RUT del paciente: ")
    nueva_cama = input("Ingrese el ID de la nueva cama: ")
    cambiar_cama(rut, nueva_cama)
elif opcion == '4':
    rut = input("Ingrese el RUT del paciente: ")
    nuevo_medico = input("Ingrese el ID del nuevo médico: ")
    cambiar_medico(rut, nuevo_medico)
elif opcion == '5':
    id_habitacion = input("Ingrese el ID de la habitación: ")
    nombre_habitacion = input("Ingrese el nombre de la habitación: ")
    piso = input("Ingrese el piso: ")
    id_cama = input("Ingrese el ID de la cama: ")
    estado = input("Ingrese el estado de la cama: ")
    crear_cama_y_habitacion(id_habitacion, nombre_habitacion, piso, id_ca
elif opcion == '6':
    mostrar_todas_tablas()
elif opcion == '0':
    break
else:
    print("Opción no válida. Intente de nuevo.")

if __name__ == "__main__":
    menu()
    cur.close()
    conn.close()
```

## Conclusiones

El desarrollo de esta aplicación de consola ha permitido aplicar de forma práctica conocimientos en modelado de bases de datos, SQL y programación en Python. Se logró cumplir con los objetivos propuestos: diseñar un modelo relacional coherente con el caso de la clínica, implementar la base de datos en PostgreSQL y desarrollar una aplicación funcional para la gestión de pacientes, médicos, camas y exámenes. Además, el trabajo en equipo facilitó la integración de ideas y la resolución de problemas en un entorno colaborativo.

## Bibliografía

- Documentación oficial de PostgreSQL.
- Documentación oficial de Python.
- Materiales complementarios proporcionados por el curso.