PROYECTO FINAL BASES DE DATOS

SISTEMA DE GESTION HOSPITALARIA CON USO DE BASES DE DATOS DISTRIBUIDAS

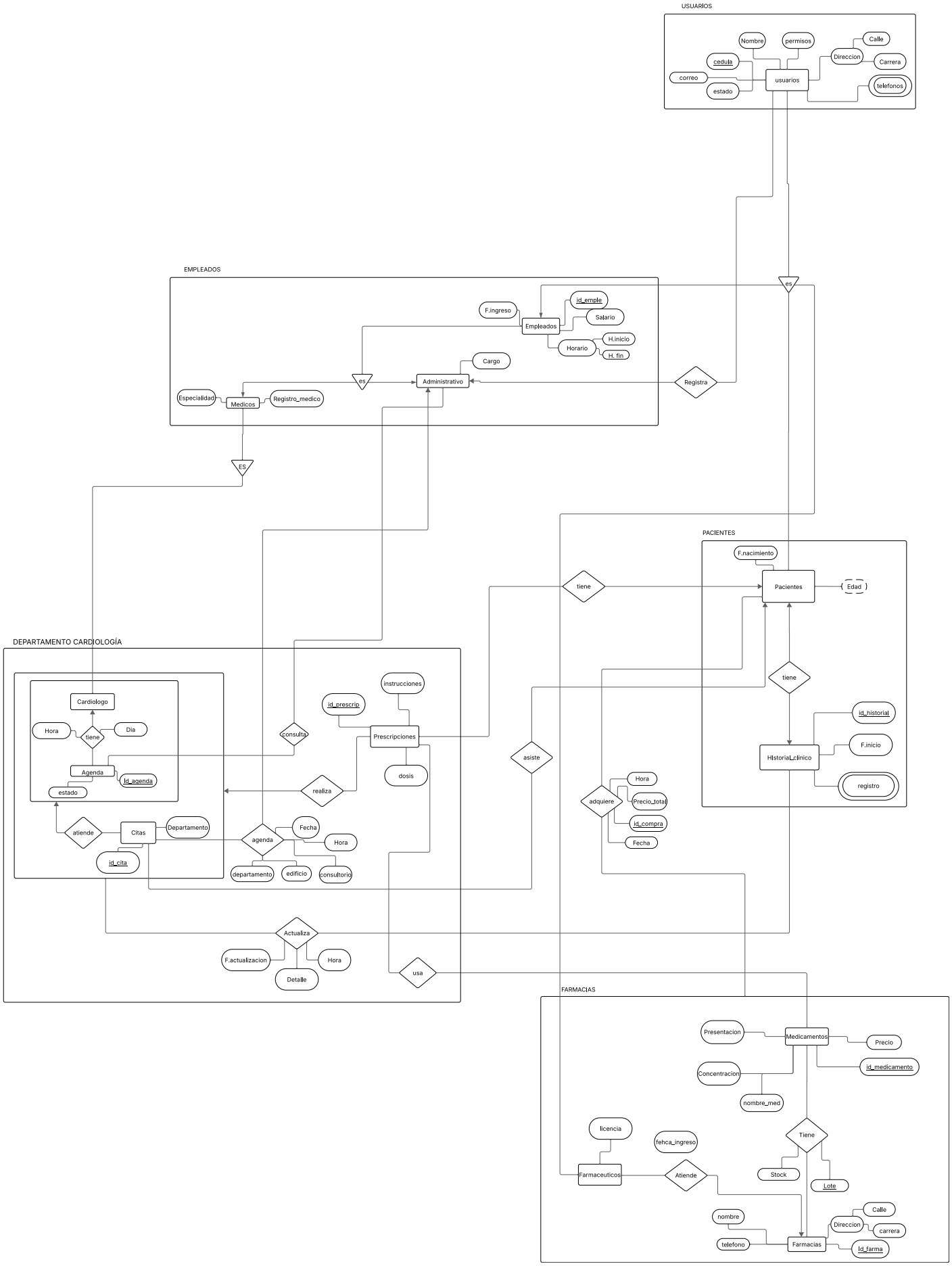
MATERIA:
BASES DE DATOS

PROFESOR:
DANIEL DAVID LEAL LARA

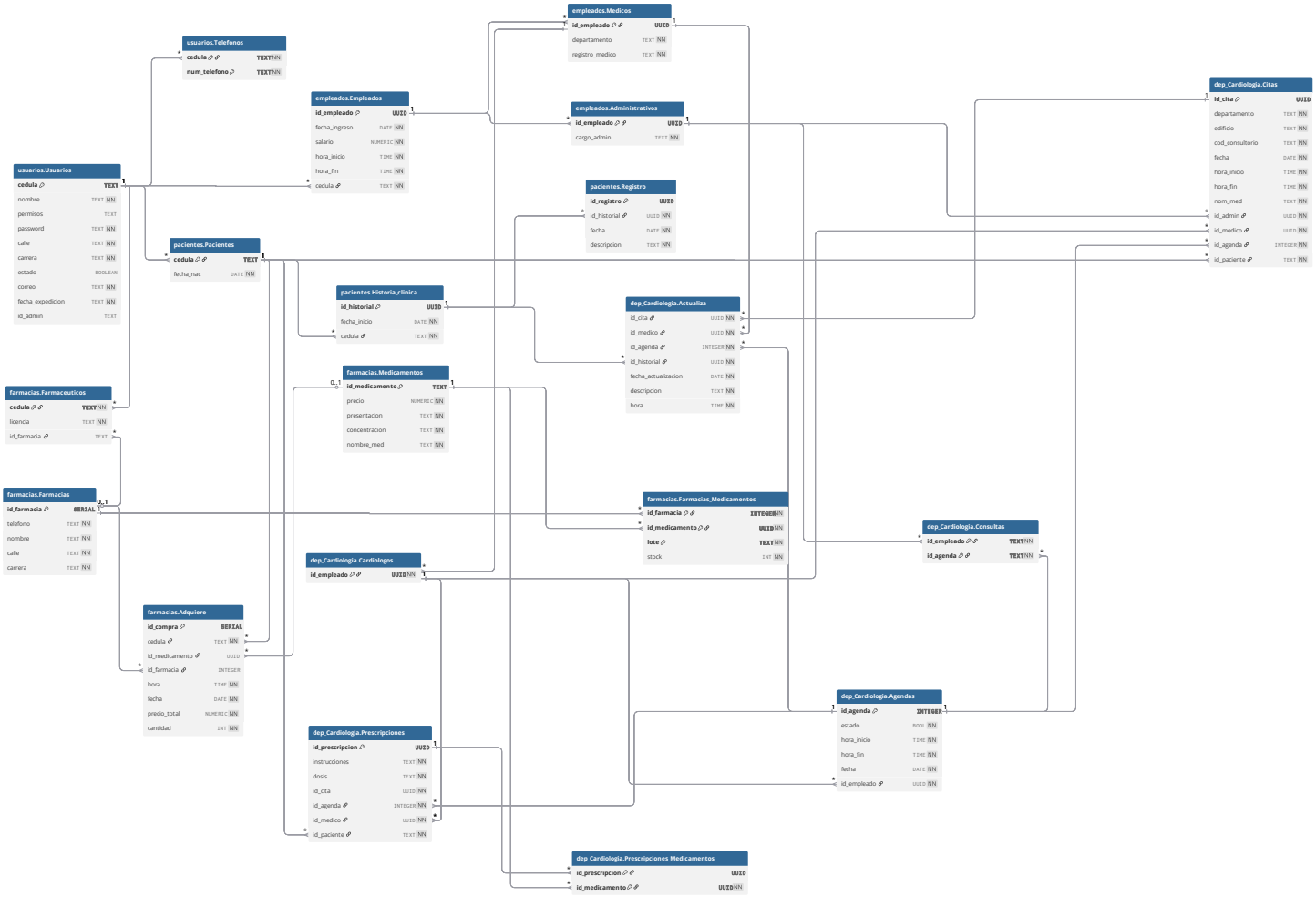
INTEGRANTES:
Juan David Zárate Moya - 20222020184
Ammi Susana Pineda Guzmán - 20221020064
Edgar Alejandro Mora Chala – 20222020005

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

DIAGRAMA ENTIDAD-RELACION:



MODELO RELACIONAL:



Diccionario de datos:

ESQUEMA: Usuarios

Tabla	PK	FK	Campo	Tipo de Dato	Nulo	Descripción
Usuarios	✔		cedula	TEXT	No	Identificador único del usuario
			nombre	TEXT	No	Nombre completo del usuario
			permisos	TEXT	Sí	Nivel de permisos del usuario (default 'user')
			password	TEXT	No	Contraseña
			calle	TEXT	No	Dirección: calle
			carrera	TEXT	No	Dirección: carrera
			estado	BOOLEAN	Sí	Estado activo/inactivo (default TRUE)
			correo	TEXT	No	Correo electrónico, único
			id_admin	TEXT	Sí	Referencia opcional a un administrativo
Telefonos	✔	✔	cedula	TEXT	No	Relación con Usuarios
	✔		num_telefono	TEXT	No	Teléfono del usuario

ESQUEMA: Pacientes

Tabla	PK	FK	Campo	Tipo de Dato	Nulo	Descripción
Pacientes	✔	✔	cedula	TEXT	No	FK a usuarios.cedula
			fecha_nac	DATE	No	Fecha de nacimiento
Historia_clinica	✔		id_historial	UUID	No	ID único de historial clínico
			fecha_inicio	DATE	No	Fecha de creación del historial
		✔	cedula	TEXT	No	Paciente propietario del historial
Registro	✔		id_registro	UUID	No	Identificador del registro clínico
		✔	id_historial	UUID	No	Historial clínico al que pertenece
			fecha	DATE	No	Fecha del registro médico
			descripcion	TEXT	No	Detalle clínico registrado

ESQUEMA: Empleados

Tabla	PK	FK	Campo	Tipo de Dato	Nulo	Descripción
Empleados	✔		id_empleado	UUID	No	Identificador único del empleado
			fecha_ingreso	DATE	No	Fecha de inicio laboral
			salario	NUMERIC	No	Sueldo del empleado
			hora_inicio	TIME	No	Hora de entrada
			hora_fin	TIME	No	Hora de salida
		✔	cedula	TEXT	No	FK a usuarios.cedula
Medicos	✔	✔	id_empleado	UUID	No	Referencia a empleados (es médico)
			departamento	TEXT	No	Departamento médico asignado
			registro_medico	TEXT	No	Número de licencia médica
Administrativos	✔	✔	id_empleado	UUID	No	Referencia a empleados (es administrativo)
			cargo_admin	TEXT	No	Rol administrativo

ESQUEMA: Farmacias

Tabla	PK	FK	Campo	Tipo de Dato	Nulo	Descripción
Farmacias	✔		id_farmacia	SERIAL	No	Identificador de farmacia
			telefono	TEXT	No	Teléfono de contacto
			nombre	TEXT	No	Nombre de la farmacia
			calle	TEXT	No	Dirección: calle
			carrera	TEXT	No	Dirección: carrera
Farmaceuticos	✔		cedula	TEXT	No	Cédula del farmaceútico
			licencia	TEXT	No	Licencia profesional (única)
		✔	id_farmacia	TEXT	Sí	FK a Farmacias
Medicamentos	✔		id_medicamento	TEXT	No	Identificador único del medicamento
			precio	NUMERIC	No	Precio unitario
			presentacion	TEXT	No	Forma de presentación
			concentracion	TEXT	No	Concentración del principio activo
			nombre_med	TEXT	No	Nombre comercial
Farmacias_Medicamentos	✔	✔	id_farmacia	INTEGER	No	FK a Farmacias
	✔	✔	id_medicamento	UUID	No	FK a Medicamentos
	✔		lote	TEXT	No	Lote del medicamento
			stock	INT	No	Cantidad disponible
Adquiere	✔		id_compra	SERIAL	No	ID de compra
			cedula	TEXT	No	Paciente que realiza la compra
		✔	id_medicamento	UUID	Sí	Medicamento adquirido
		✔	id_farmacia	INTEGER	Sí	Farmacia asociada
			hora	TIME	No	Hora de la compra
			fecha	DATE	No	Fecha de la compra
			precio_total	NUMERIC	No	Valor total
			cantidad	INT	No	Unidades adquiridas

ESQUEMA: Dep_Cardiologia/Neurologia

Tabla	PK	FK	Campo	Tipo de Dato	Nulo	Descripción
[Especialistas]	✔		id_empleado	UUID/TEXT	No	Médico del departamento
Agendas	✔		id_agenda	INTEGER/TEXT	No	Identificador único de agenda
			estado	BOOLEAN	No	Estado de la agenda (activa/inactiva)
			hora_inicio	TIME	No	Hora de inicio de la jornada
			hora_fin	TIME	No	Hora de finalización
			fecha	DATE	No	Fecha del turno
		✔	id_empleado	UUID/TEXT	No	Médico asignado
Consultas	✔		id_empleado	TEXT	No	Médico que realiza la consulta
	✔	✔	id_agenda	TEXT	No	Agenda correspondiente
Citas	✔		id_cita	UUID/TEXT	No	Cita médica
			departamento	TEXT	No	Departamento
			edificio	TEXT	No	Edificio donde se da la consulta
			cod_consultorio	TEXT	No	Número de consultorio
			fecha	DATE	No	Fecha de la cita
			hora_inicio	TIME	No	Hora de inicio
			hora_fin	TIME	No	Hora de fin
			nom_med	TEXT	No	Nombre del médico
			id_admin	UUID/TEXT	No	Administrativo asignado
		✔	id_medico	UUID/TEXT	No	Médico asignado
		✔	id_agenda	INTEGER/TEXT	No	Agenda asignada
			id_paciente	TEXT	No	Paciente que asiste
Prescripciones	✔		id_prescripcion	UUID/TEXT	No	Identificador único
			instrucciones	TEXT	No	Instrucciones médicas
			dosis	TEXT	No	Dosis del medicamento
		✔	id_cita	UUID/TEXT	No	Cita relacionada
		✔	id_agenda	INTEGER/TEXT	No	Agenda de referencia
		✔	id_medico	UUID/TEXT	No	Médico que prescribe
			id_paciente	TEXT	No	Paciente
Prescripciones_Medicamentos	✔	✔	id_prescripcion	UUID/TEXT	No	FK a Prescripción
	✔		id_medicamento	UUID/TEXT	No	Medicamento recetado
Actualiza		✔	id_cita	UUID/TEXT	No	Cita actualizada
			id_medico	UUID/TEXT	No	Médico que actualiza
		✔	id_agenda	INTEGER/TEXT	No	Agenda correspondiente
			id_historial	UUID/TEXT	No	Historial que se actualiza
			fecha_actualizacion	DATE	No	Fecha de actualización
			descripcion	TEXT	No	Descripción del evento clínico
			hora	TIME	No	Hora de la actualización

Descripción de la distribución y asignación de la información entre los diferentes departamentos:

La gestión de la información en el sistema se estructura para asegurar una distribución y asignación eficiente de los datos, garantizando que cada departamento acceda únicamente a la información relevante para sus operaciones.

Acceso a Información Interna del Departamento: Cada departamento contará con acceso exclusivo a la información que le es propia. Esto incluye:

- Citas médicas programadas.
- Agendas médicas específicas del personal asignado.
- Prescripciones generadas dentro del departamento.
- Actualizaciones realizadas en los historiales clínicos por sus médicos.
- Identificación de los médicos encargados de atender dicho departamento.

Acceso a Información Externa (Consulta): En cuanto a la información externa, los departamentos tendrán una capacidad de consulta limitada y específica para fomentar la colaboración sin comprometer la seguridad o la relevancia de los datos. Podrán consultar únicamente:

- El listado de medicamentos disponibles en la base de datos general.
- Los historiales clínicos de los pacientes que están siendo atendidos por el propio departamento.

De esta forma, se establece un mecanismo de "comunicación" indirecta y controlada entre los distintos departamentos, permitiendo la visualización de los registros guardados en el historial clínico de los pacientes, lo cual es fundamental para una atención integral y coordinada.

SCRIPTS:

DDL:

```
CREATE DATABASE gestion_usuarios;
-- ==========================
-- ESQUEMA: usuarios
-- ==========================

    CREATE SCHEMA usuarios;

    CREATE TABLE usuarios.Usuarios (
        cedula TEXT PRIMARY KEY,
        nombre TEXT NOT NULL,
        permisos TEXT DEFAULT 'user',
        password TEXT NOT NULL,
        calle TEXT NOT NULL,
        carrera TEXT NOT NULL,
        estado BOOLEAN DEFAULT TRUE,
        correo TEXT NOT NULL UNIQUE,
        fecha_expedicion TEXT NOT NULL,
        id_admin TEXT NULL
    );
```

```sql
CREATE TABLE usuarios.Telefonos (
    cedula TEXT NOT NULL,
    num_telefono TEXT NOT NULL,
    PRIMARY KEY (cedula, num_telefono),
    FOREIGN KEY (cedula) REFERENCES usuarios.Usuarios(cedula)
);

-- ==========================
-- ESQUEMA: pacientes
-- ==========================

CREATE SCHEMA pacientes;

CREATE TABLE pacientes.Pacientes (
    cedula TEXT PRIMARY KEY REFERENCES usuarios.Usuarios(cedula),
    fecha_nac DATE NOT NULL
);

CREATE TABLE pacientes.Historia_clinica (
    id_historial UUID PRIMARY KEY,
    fecha_inicio DATE NOT NULL,
    cedula TEXT NOT NULL,
    FOREIGN KEY (cedula) REFERENCES pacientes.Pacientes(cedula)
);

CREATE TABLE pacientes.Registro (
    id_registro UUID PRIMARY KEY,
    id_historial UUID NOT NULL,
    fecha DATE NOT NULL,
    descripcion TEXT NOT NULL,
    FOREIGN KEY (id_historial) REFERENCES pacientes.Historia_clinica(id_historial)
);

-- ==========================
-- ESQUEMA: empleados
-- ==========================
CREATE SCHEMA IF NOT EXISTS empleados;

CREATE TABLE empleados.Empleados (
    id_empleado UUID PRIMARY KEY,
    fecha_ingreso DATE NOT NULL,
    salario NUMERIC NOT NULL,
    hora_inicio TIME NOT NULL,
    hora_fin TIME NOT NULL,
    cedula TEXT NOT NULL,
    FOREIGN KEY (cedula) REFERENCES usuarios.Usuarios(cedula)
);

CREATE TABLE empleados.Medicos (
    id_empleado UUID PRIMARY KEY,
```

```
        departamento TEXT NOT NULL,
        registro_medico TEXT NOT NULL,
        FOREIGN KEY (id_empleado) REFERENCES empleados.Empleados(id_empleado)
    );

    CREATE TABLE empleados.Administrativos (
        id_empleado UUID PRIMARY KEY,
        cargo_admin TEXT NOT NULL,
        FOREIGN KEY (id_empleado) REFERENCES empleados.Empleados(id_empleado)
    );
```

```
CREATE DATABASE gestion_depCardio;
    -- ==========================
    -- ESQUEMA: departamento
    -- ==========================

    CREATE SCHEMA dep_Cardiologia;

    CREATE TABLE dep_Cardiologia.Cardiologos(
        id_empleado UUID NOT NULL PRIMARY KEY -- PRIMARY KEY REFERENCES
empleados.Medicos(id_empleado)
    );

    CREATE TABLE dep_Cardiologia.Agendas (
        id_agenda INTEGER PRIMARY KEY,
        estado BOOL NOT NULL,
        hora_inicio TIME NOT NULL,
        hora_fin TIME NOT NULL ,
        fecha DATE NOT NULL,
        id_empleado UUID NOT NULL,
        FOREIGN KEY (id_empleado) REFERENCES dep_Cardiologia.Cardiologos(id_empleado)
    );

    CREATE TABLE dep_Cardiologia.Consultas (
        id_empleado TEXT NOT NULL,
        id_agenda TEXT NOT NULL,
        PRIMARY KEY (id_empleado, id_agenda),
        -- FOREIGN KEY (id_empleado) REFERENCES empleados.Administrativos(id_empleado),
        FOREIGN KEY (id_agenda) REFERENCES dep_Cardiologia.Agendas(id_agenda)
    );

    CREATE TABLE dep_Cardiologia.Citas (
        id_cita UUID PRIMARY KEY,
        departamento TEXT NOT NULL,
        edificio TEXT NOT NULL,
        cod_consultorio TEXT NOT NULL,
        fecha DATE NOT NULL,
        hora_inicio TIME NOT NULL,
        hora_fin TIME NOT NULL,
        nom_med TEXT NOT NULL,
```

```sql
        id_admin UUID NOT NULL,
        id_medico UUID NOT NULL,
        id_agenda INTEGER NOT NULL,
        id_paciente TEXT NOT NULL,
        FOREIGN KEY (id_agenda) REFERENCES dep_Cardiologia.Agendas(id_agenda),
        -- FOREIGN KEY (id_admin) REFERENCES empleados.Administrativos(id_empleado),
        FOREIGN KEY (id_medico) REFERENCES dep_Cardiologia.Cardiologos(id_empleado)
        -- FOREIGN KEY (id_paciente) REFERENCES pacientes.Pacientes(cedula)
    );

    CREATE TABLE dep_Cardiologia.Prescripciones (
        id_prescripcion UUID PRIMARY KEY,
        instrucciones TEXT NOT NULL,
        dosis TEXT NOT NULL,
        id_cita UUID NOT NULL,
        id_agenda INTEGER NOT NULL,
        id_medico UUID NOT NULL,
        id_paciente TEXT NOT NULL,

        FOREIGN KEY (id_agenda) REFERENCES dep_Cardiologia.Agendas(id_agenda),
        FOREIGN KEY (id_medico) REFERENCES dep_Cardiologia.Cardiologos(id_empleado)
        -- FOREIGN KEY (id_paciente) REFERENCES pacientes.Pacientes(cedula)
    );

    CREATE TABLE dep_Cardiologia.Prescripciones_Medicamentos (
        id_prescripcion UUID REFERENCES dep_Cardiologia.Prescripciones(id_prescripcion),
        id_medicamento UUID NOT NULL, -- REFERENCES farmacias.Medicamentos(id_medicamento),
        PRIMARY KEY (id_prescripcion,id_medicamento)
    );

    CREATE TABLE dep_Cardiologia.Actualiza (
        id_cita UUID NOT NULL,
        id_medico UUID NOT NULL,
        id_agenda INTEGER NOT NULL,
        id_historial UUID NOT NULL,
        fecha_actualizacion DATE NOT NULL,
        descripcion TEXT NOT NULL,
        hora TIME NOT NULL,
        FOREIGN KEY (id_cita) REFERENCES dep_Cardiologia.Citas(id_cita),
        -- FOREIGN KEY (id_medico) REFERENCES empleados.Medicos(id_empleado),
        FOREIGN KEY (id_agenda) REFERENCES dep_Cardiologia.Agendas(id_agenda)
        -- FOREIGN KEY (id_historial) REFERENCES pacientes.Historia_clinica(id_historial)
    );
```

```sql
CREATE DATABASE gestion_farmacias;

CREATE SCHEMA IF NOT EXISTS farmacias;


    CREATE TABLE farmacias.Farmacias (
```

```sql
        id_farmacia SERIAL PRIMARY KEY,
        telefono TEXT NOT NULL,
        nombre TEXT NOT NULL,
        calle TEXT NOT NULL,
        carrera TEXT NOT NULL
    );
    CREATE TABLE farmacias.Farmaceuticos(
        cedula TEXT NOT NULL PRIMARY KEY,
        licencia TEXT NOT NULL UNIQUE,
        id_farmacia TEXT REFERENCES farmacias.Farmacias(id_farmacia)
        --FOREIGN KEY (cedula) REFERENCES usuarios.usuarios(cedula),
    );


    CREATE TABLE farmacias.Medicamentos (
        id_medicamento TEXT PRIMARY KEY,
        precio NUMERIC NOT NULL,
        presentacion TEXT NOT NULL,
        concentracion TEXT NOT NULL,
        nombre_med TEXT NOT NULL
    );

    CREATE TABLE farmacias.Farmacias_Medicamentos (
        id_farmacia INTEGER NOT NULL,
        id_medicamento UUID NOT NULL,
        lote TEXT NOT NULL,
        stock INT NOT NULL,
        PRIMARY KEY (id_farmacia, id_medicamento, lote),
        FOREIGN KEY (id_farmacia) REFERENCES farmacias.Farmacias(id_farmacia),
        FOREIGN KEY (id_medicamento) REFERENCES farmacias.Medicamentos(id_medicamento)
    );

    CREATE TABLE farmacias.Adquiere (
        id_compra SERIAL PRIMARY KEY,
        cedula TEXT NOT NULL, -- REFERENCES pacientes.Pacientes(cedula),
        id_medicamento UUID REFERENCES farmacias.Medicamentos(id_medicamento),
        id_farmacia INTEGER REFERENCES farmacias.Farmacias(id_farmacia),
        hora TIME NOT NULL,
        fecha DATE NOT NULL,
        precio_total NUMERIC NOT NULL,
        cantidad INT NOT NULL
    );
```

ROLES Y USUARIOS:

```sql
-- Crear roles
CREATE ROLE admin_global LOGIN PASSWORD 'admin123';
CREATE ROLE paciente LOGIN PASSWORD 'paciente123';
CREATE ROLE medico LOGIN PASSWORD 'medico123';
CREATE ROLE admin_clinico LOGIN PASSWORD 'adminclinico123';
```

```sql
CREATE ROLE farmaceutico LOGIN PASSWORD 'farmaceutico123';
CREATE ROLE admin_farm LOGIN PASSWORD 'adminfarm123';

-- Esquema: usuarios
GRANT CONNECT ON DATABASE gestion_hospitalaria TO paciente, medico, admin_clinico,
admin_global;
GRANT USAGE ON SCHEMA usuarios TO paciente, medico, admin_clinico, admin_global;
GRANT SELECT, UPDATE ON usuarios.Usuarios TO paciente; -- solo su propio registro
GRANT SELECT ON usuarios.Usuarios TO medico, admin_clinico, admin_global;
GRANT INSERT, UPDATE, DELETE ON usuarios.Usuarios TO admin_clinico, admin_global;

-- Esquema: pacientes
GRANT USAGE ON SCHEMA pacientes TO paciente, medico, admin_clinico, admin_global;
GRANT SELECT ON pacientes.Pacientes TO paciente, medico, admin_clinico, admin_global;
GRANT INSERT, UPDATE ON pacientes.Historia_clinica TO medico;
GRANT SELECT ON pacientes.Historia_clinica TO medico, paciente;
GRANT INSERT ON pacientes.Registro TO medico;
GRANT SELECT ON pacientes.Registro TO medico, paciente;

-- Esquema: empleados
GRANT USAGE ON SCHEMA empleados TO medico, admin_clinico, admin_global;
GRANT SELECT ON empleados.Empleados TO medico, admin_clinico, admin_global;
GRANT INSERT, UPDATE, DELETE ON empleados.Empleados TO admin_clinico, admin_global;

-- Esquema: farmacias
GRANT USAGE ON SCHEMA farmacias TO paciente, farmaceutico, admin_farm, admin_global;
GRANT SELECT ON farmacias.Medicamentos TO paciente, farmaceutico, admin_farm;
GRANT SELECT, INSERT, UPDATE ON farmacias.Adquiere TO farmaceutico, admin_farm;
GRANT INSERT, UPDATE, DELETE ON farmacias.Farmacias_Medicamentos TO admin_farm;
GRANT INSERT, DELETE ON farmacias.Farmaceuticos TO admin_farm;

-- Esquemas: dep_Cardiologia, dep_Neurologia
GRANT USAGE ON SCHEMA dep_Cardiologia TO medico, admin_clinico, admin_global;
GRANT USAGE ON SCHEMA dep_Neurologia TO medico, admin_clinico, admin_global;

GRANT SELECT, INSERT, UPDATE ON dep_Cardiologia.Citas TO medico, admin_clinico;
GRANT SELECT, INSERT ON dep_Cardiologia.Prescripciones TO medico;
GRANT SELECT ON dep_Cardiologia.Prescripciones TO paciente;
GRANT INSERT ON dep_Cardiologia.Actualiza TO medico;

GRANT SELECT, INSERT, UPDATE ON dep_Neurologia.Citas TO medico, admin_clinico;
GRANT INSERT ON dep_Neurologia.Prescripciones TO medico;
GRANT SELECT ON dep_Neurologia.Prescripciones TO paciente;
GRANT INSERT ON dep_Neurologia.Actualiza TO medico;
```

VISTAS:

```sql
 ----- PACIENTES -----

-- Vista con la información básica de los pacientes.
```

```sql
CREATE OR REPLACE VIEW vista_info_paciente AS
SELECT
    u.cedula,
    u.nombre,
    u.correo,
    u.estado,
    p.fecha_nac
FROM usuarios.Usuarios u
JOIN pacientes.Pacientes p ON u.cedula = p.cedula;

-- Vista de historial clínico por paciente
CREATE MATERIALIZED VIEW vista_historial_paciente AS
SELECT
    h.cedula,
    h.id_historial,
    h.fecha_inicio,
    r.id_registro,
    r.fecha,
    r.descripcion
FROM pacientes.Historia_clinica h
LEFT JOIN pacientes.Registro r ON h.id_historial = r.id_historial;
 -- REFRESH MATERIALIZED VIEW vista_historial_paciente;

-- Tabla temporal de prescripciones (Cardiología + Neurología)
-- y agrega un campo para identificar el departamento.

CREATE OR REPLACE VIEW vista_prescripciones_paciente AS
SELECT
    id_prescripcion,
    id_cita,
    id_agenda,
    id_medico,
    id_paciente,
    instrucciones,
    dosis,
    'Cardiologia' AS departamento
FROM dep_Cardiologia.Prescripciones
UNION
SELECT
    id_prescripcion,
    id_cita,
    id_agenda,
    id_medico,
    id_paciente,
    instrucciones,
    dosis,
    'Neurologia' AS departamento
FROM dep_Neurologia.Prescripciones;

  ----- MEDICOS -----
```

```sql
-- Vista con el resumen de agendas de los médicos. (los horarios no cambian constantemente)
CREATE MATERIALIZED VIEW vista_agendas_medico AS
SELECT
    'Cardiologia' AS departamento,
    id_agenda,
    fecha,
    hora_inicio,
    hora_fin,
    estado,
    id_empleado
FROM dep_Cardiologia.Agendas
UNION
SELECT
    'Neurologia',
    id_agenda,
    fecha,
    hora_inicio,
    hora_fin,
    estado,
    id_empleado
FROM dep_Neurologia.Agendas;
-- REFRESH MATERIALIZED VIEW vista_agendas_medico;

-- Vista de citas medicas completas con información del paciente y médico. (Cambia
constantemente)
CREATE OR REPLACE VIEW vista_citas_medico AS
SELECT
    c.id_cita,
    c.departamento,
    c.fecha,
    c.hora_inicio,
    c.hora_fin,
    c.id_medico,
    c.id_paciente,
    p.nombre AS nombre_paciente
FROM dep_Cardiologia.Citas c
JOIN usuarios.Usuarios p ON p.cedula = c.id_paciente
UNION
SELECT
    c.id_cita,
    c.departamento,
    c.fecha,
    c.hora_inicio,
    c.hora_fin,
    c.id_medico,
    c.id_paciente,
    p.nombre
FROM dep_Neurologia.Citas c
JOIN usuarios.Usuarios p ON p.cedula = c.id_paciente;
```

```sql
  ---- ----- FARMACEUTICOS -----
-- Vista materializada de stock por farmacia (No toca revisar stock constantemente)
CREATE MATERIALIZED VIEW vista_stock_farmacia AS
SELECT
    f.id_farmacia,
    f.nombre AS nombre_farmacia,
    m.id_medicamento,
    m.nombre_med,
    fm.lote,
    fm.stock
FROM farmacias.Farmacias f
JOIN farmacias.Farmacias_Medicamentos fm ON f.id_farmacia = fm.id_farmacia
JOIN farmacias.Medicamentos m ON fm.id_medicamento = m.id_medicamento;
-- REFRESH MATERIALIZED VIEW vista_stock_farmacia;

-- Tabla temporal de compras realizadas por pacientes (Se usa en sesiones para filtrar por
fecha o paciente)
CREATE TEMP TABLE temp_compras_sesion AS
SELECT
    a.id_compra,
    a.cedula,
    u.nombre AS nombre_paciente,
    a.id_medicamento,
    m.nombre_med,
    a.cantidad,
    a.precio_total,
    a.fecha,
    a.hora,
    f.nombre AS farmacia
FROM farmacias.Adquiere a
JOIN farmacias.Medicamentos m ON m.id_medicamento = a.id_medicamento
JOIN farmacias.Farmacias f ON f.id_farmacia = a.id_farmacia
JOIN usuarios.Usuarios u ON u.cedula = a.cedula;
```