

Entrega 1 – Proyecto Final: Base de Datos Consultorio Médico

Alumna:	Micaela Rombolá
Fecha:	12/08/2025
Materia:	SQL

1) Introducción

El presente trabajo describe el desarrollo de una base de datos relacional destinada a la administración de un consultorio médico. El sistema se diseñó para organizar la información de los pacientes, médicos, especialidades, consultorios y turnos, con el objetivo de garantizar la integridad de los datos, optimizar el flujo de trabajo y reducir errores administrativos. La base de datos fue concebida teniendo en cuenta criterios de normalización y escalabilidad para adaptarse a futuras necesidades.

2) Objetivos

Funcionales

- Permitir el registro, actualización y consulta de pacientes, médicos, especialidades, consultorios y turnos.
- Evitar duplicidades mediante claves únicas en DNI y matrícula profesional.
- Facilitar la búsqueda de turnos filtrando por fecha, médico, especialidad o consultorio.
- Generar estadísticas como cantidad de turnos por especialidad o estado.

Técnicos

- Implementar claves primarias y foráneas, índices y restricciones para mantener la integridad referencial.
- Diseñar un modelo en tercera forma normal (3FN) para evitar redundancia y mejorar la eficiencia.
- Preparar índices para optimizar consultas sobre campos de búsqueda frecuente.

Estratégicos

- Servir como base para la incorporación de módulos de facturación, gestión de usuarios o historial médico.
- Facilitar la toma de decisiones basada en datos, a través de reportes claros y confiables.

3) Situación Problemática

- Antes de la implementación, el consultorio gestionaba la información en hojas de cálculo y registros físicos.
- Pérdida de información por errores humanos o falta de respaldo.
 - Asignación duplicada de turnos.
 - Datos inconsistentes en nombres, documentos y matrículas.
 - Dificultad para acceder al historial de un paciente o consultar la agenda de un médico.

La base de datos propuesta resuelve estas deficiencias mediante relaciones bien definidas, integridad referencial y backups automáticos.

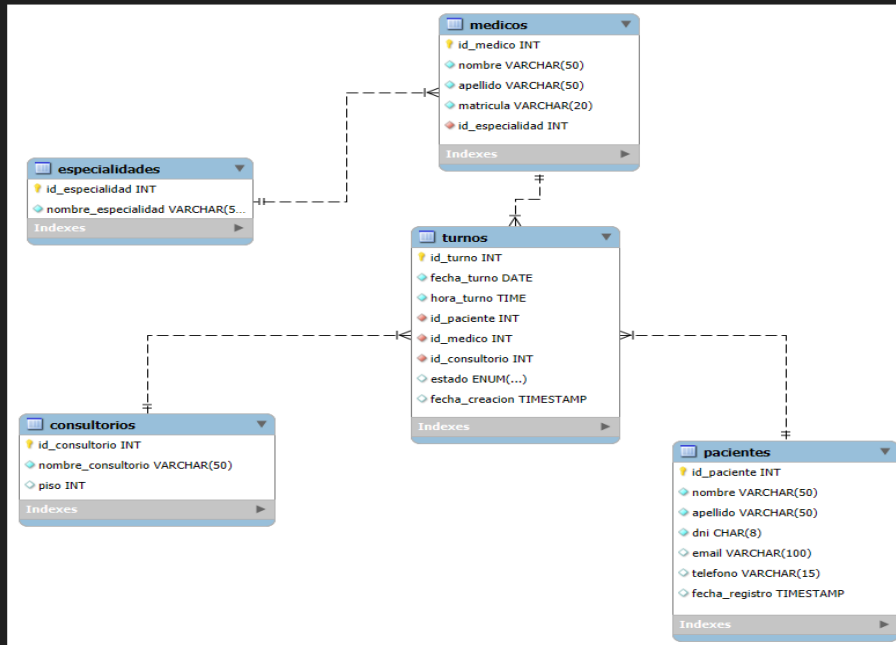
4) Modelo de Negocio

El consultorio cuenta con un equipo de médicos, cada uno perteneciente a una especialidad específica. Los pacientes solicitan turnos que se asignan a un médico y un consultorio en una fecha y hora determinada. Los turnos pueden tener estado Pendiente, Confirmado o Cancelado.

Flujo resumido:

1. Solicitud de turno.
2. Asignación de médico, especialidad y consultorio.
3. Confirmación o cancelación.
4. Atención del paciente.
5. Registro y análisis de la información.

5) Diagrama Entidad-Relación



6) Listado de Tablas y Campos

Especialidades

- id_especialidad (PK)
- nombre_especialidad (UNIQUE) - Nombre único de la especialidad médica.

Pacientes

- id_paciente (PK)
- nombre, apellido
- dni (UNIQUE), email, telefono
- fecha_registro (DEFAULT CURRENT_TIMESTAMP)

Consultorios

- id_consultorio (PK)
- nombre_consultorio, piso

Médicos

- id_medico (PK), nombre, apellido
- matricula (UNIQUE)
- id_especialidad (FK → Especialidades)

Turnos

- id_turno (PK), fecha_turno, hora_turno
- id_paciente (FK), id_medico (FK), id_consultorio (FK)
- estado (ENUM)
- fecha_creacion (DEFAULT CURRENT_TIMESTAMP)

7) Abreviaturas de Campos

Campo	Abrev.	Nombre completo
id_especialidad	IDES	ID de especialidad
nombre_especialidad	NOMESP	Nombre de la especialidad
id_paciente	IDP	ID del paciente
nombre	NOM	Nombre
apellido	APE	Apellido
dni	DNI	Documento Nacional de Identidad
email	EMAIL	Correo electrónico
telefono	TEL	Teléfono
fecha_registro	FREG	Fecha de registro
id_consultorio	IDC	ID de consultorio
nombre_consultorio	NOMC	Nombre del consultorio
piso	PISO	Piso
id_medico	IDM	ID del médico
matricula	MAT	Matrícula profesional
fecha_turno	FET	Fecha del turno
hora_turno	HOT	Hora del turno
estado	EST	Estado del turno
fecha_creacion	FCRE	Fecha de creación

8) Script SQL (Extracto)

```
-- Creación de la base de datos ---
CREATE DATABASE ConsultorioMedico;
USE ConsultorioMedico;

-- Tabla: Especialidades ---
CREATE TABLE Especialidades (
    id_especialidad INT AUTO_INCREMENT PRIMARY KEY,
    nombre_especialidad VARCHAR(50) NOT NULL UNIQUE COMMENT 'Nombre único de la especialidad médica'
);

-- Tabla: Pacientes ---
CREATE TABLE Pacientes (
    id_paciente INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    dni CHAR(8) UNIQUE NOT NULL COMMENT 'Documento Nacional de Identidad',
    email VARCHAR(100),
    telefono VARCHAR(15),
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT 'Fecha de alta del paciente'
);

-- Tabla: Consultorios ---
CREATE TABLE Consultorios (
    id_consultorio INT AUTO_INCREMENT PRIMARY KEY,
    nombre_consultorio VARCHAR(50) NOT NULL,
    piso INT COMMENT 'Número de piso donde se ubica'
);

-- Tabla: Medicos ---
CREATE TABLE Medicos (
    id_medico INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    matricula VARCHAR(20) UNIQUE NOT NULL COMMENT 'Número de matrícula profesional',
    id_especialidad INT NOT NULL,
    FOREIGN KEY (id_especialidad) REFERENCES Especialidades(id_especialidad)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

-- Tabla: Turnos ---
CREATE TABLE Turnos (
    id_turno INT AUTO_INCREMENT PRIMARY KEY,
    fecha_turno DATE NOT NULL,
    hora_turno TIME NOT NULL,
    id_paciente INT NOT NULL,
    id_medico INT NOT NULL,
    id_consultorio INT NOT NULL,
    estado ENUM('Pendiente', 'Confirmado', 'Cancelado') DEFAULT 'Pendiente',
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_paciente) REFERENCES Pacientes(id_paciente)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_medico) REFERENCES Medicos(id_medico)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_consultorio) REFERENCES Consultorios(id_consultorio)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

9) Enlace al repositorio en GitHub

Ver en GitHub: <https://github.com/Micaelarombola/ConsultoriomedicoRombola>