

Base de datos MediCare Desk (Mysql)



Presentado por:

Yamid Alfonso Gonzalez Torres

Jenny Catherine Herrera Garzon

Edwin Andres Marin Vanegas

Diego Steven Pinzon Yossa

Profesor:

Oscar Eduardo Alvarez Rodriguez

**Universidad Nacional de Colombia
Facultad de Ingeniería
Ingeniería de software
2025**



1. Descripción General del Sistema

Este documento describe la estructura y el propósito de cada componente de la base de datos del sistema **MediCareDesk**, diseñada en SQLite para ejecución local. Está orientada a personas sin conocimientos técnicos profundos.

Consideraciones generales

La base de datos está diseñada para uso local mediante SQLite. Incluye restricciones de integridad referencial (**PRAGMA foreign_keys = ON**) y emplea **CHECK** para validar algunos campos. La lógica de negocio originalmente implementada mediante procedimientos en MySQL fue migrada a funciones en Python. Se utilizan vistas para facilitar el acceso a información relevante desde el sistema sin realizar consultas SQL complejas en la interfaz.

2. Tablas principales

Cuidador

Almacena la información de los cuidadores autorizados para acceder al sistema. Cada cuidador tiene un nombre, una relación con el paciente, datos de contacto, correo electrónico único y contraseña.

Paciente

Registra los datos personales del paciente, incluyendo nombre, edad, género, contacto de emergencia y observaciones clínicas. También indica si el paciente está activo y guarda la fecha de registro.

Medicamento

Contiene los medicamentos disponibles para asignación. Incluye nombre, principio activo, indicaciones, contraindicaciones, presentación (como comprimidos, jarabe, etc.), laboratorio y fecha de caducidad.



3. Tablas funcionales

Tratamiento

Define un tratamiento general asignado a un paciente. Es una entidad base que se conecta con los medicamentos a través de la tabla intermedia.

Tratamiento_Medicamento

Relaciona cada tratamiento con los medicamentos que lo componen, incluyendo la dosis, frecuencia, vía de administración, rango de fechas, estado del tratamiento y hora preferida para la toma.

Toma

Registra cada una de las tomas generadas automáticamente por el sistema a partir de la frecuencia y duración del tratamiento. Incluye fecha, hora programada y estado (programada, tomada, omitida).

Bitacora_Eventos

Permite registrar eventos relevantes en la evolución del paciente (por ejemplo, vómitos, caídas, somnolencia), ingresados por los cuidadores.

Paciente_Cuidador

Tabla intermedia que permite relacionar múltiples pacientes con múltiples cuidadores. Facilita el control de acceso y la consulta por paciente asociado.

4. Vistas definidas

Vista_Pacientes

Ofrece un resumen general de los pacientes activos del sistema para visualización en la interfaz.

Vista_Medicamentos

Facilita la consulta y presentación de los medicamentos disponibles sin exponer todos los campos internos.

Vista_Tomas_Hoy

Filtra las tomas programadas exclusivamente para la fecha actual, usadas en la vista de seguimiento diario.



Vista_Historial_Tomas_Paciente

Permite visualizar el historial completo de tomas realizadas, omitidas o pendientes de un paciente determinado.

Vista_Alertas_Tomas

Filtra aquellas tomas que no han sido marcadas como completadas y están próximas a su hora límite. Se usa para emitir alertas.

Vista_General_Tratamientos

Muestra una consolidación de los tratamientos activos, incluyendo datos del paciente, medicamento, frecuencia y estado.

5. Observaciones finales

- Las operaciones complejas como generación automática de tomas o actualización de estado de una toma se trasladaron al backend en Python.
- El uso de vistas permite una separación entre lógica de presentación y estructura interna de la base de datos.
- El modelo está pensado para escalar y ser mantenido fácilmente por desarrolladores con distintos niveles de experiencia.