

JUSTIFICACIÓN DEL DISEÑO DE LA BASE DE DATOS

Introducción

En este documento se presenta la justificación del diseño de la base de datos para el Sistema de Gestión de Hospital. El diseño busca garantizar la integridad, eficiencia y flexibilidad del sistema, asegurando que la base de datos esté normalizada al menos hasta la Tercera Forma Normal (3FN).

Objetivos del Diseño

Minimizar la redundancia de datos.

Evitar anomalías en las operaciones de inserción, actualización y eliminación.

Garantizar la integridad referencial entre las entidades.

Facilitar la escalabilidad y mantenimiento del sistema.

Entidades y Relaciones

El sistema se compone de tres entidades principales:

Paciente

Médico

Turno

Diagrama de relaciones

Paciente ||--o{ Turno : "tiene"

Medico ||--o{ Turno : "atiende"

Definición de Tablas

Tabla Paciente

Almacena información de los pacientes del hospital.

Campos:

id_paciente (INT AUTO_INCREMENT, PRIMARY KEY)

nombre (VARCHAR(50), NOT NULL)

apellido (VARCHAR(50), NOT NULL)

edad (INT)

direccion (VARCHAR(100))

telefono (VARCHAR(20))

email (VARCHAR(100))

Tabla Medico

Almacena información de los médicos del hospital.

Campos:

id_medico (INT AUTO_INCREMENT, PRIMARY KEY)

nombre (VARCHAR(50), NOT NULL)

apellido (VARCHAR(50), NOT NULL)

especialidad (VARCHAR(50))

telefono (VARCHAR(20))

email (VARCHAR(100))

Tabla Turno

Registra los turnos programados entre pacientes y médicos.

Campos:

id_turno (INT AUTO_INCREMENT, PRIMARY KEY)

id_paciente (INT, FOREIGN KEY → Paciente(id_paciente))

id_medico (INT, FOREIGN KEY → Medico(id_medico))

fecha_hora (DATETIME)

estado (VARCHAR(20))

Normalización de la Base de Datos

A continuación, se justificará que el diseño de la base de datos corresponde al menos a la Tercera Forma Normal (3FN).

Primera Forma Normal (1FN)

Una tabla está en 1FN si:

Todos los atributos son atómicos (no divisibles).

No hay grupos repetitivos.

Análisis:

Paciente: Todos los campos contienen datos atómicos, como nombre, apellido, etc.

Medico: Similarmente, todos los campos son atómicos.

Turno: Los campos representan datos indivisibles.

Conclusión: Todas las tablas están en 1FN.

Segunda Forma Normal (2FN)

Una tabla está en 2FN si:

Está en 1FN.

Todos los atributos que no son clave dependen completamente de la clave primaria.

Análisis:

Tabla Paciente

Clave primaria: id_paciente.

Todos los demás campos dependen totalmente de id_paciente.

Tabla Medico

Clave primaria: id_medico.

Los demás campos dependen totalmente de id_medico.

Tabla Turno

Clave primaria: id_turno.

Los campos id_paciente, id_medico, fecha_hora y estado dependen completamente de id_turno.

Conclusión: Todas las tablas están en 2FN.

Tercera Forma Normal (3FN)

Una tabla está en 3FN si:

Está en 2FN.

No hay dependencias transitivas entre los atributos que no son clave.

Análisis:

Tabla Paciente

No existen dependencias entre atributos no clave.

Dependencias funcionales:

id_paciente → nombre, apellido, edad, direccion, telefono, email

Tabla Medico

No existen dependencias entre atributos no clave.

Dependencias funcionales:

id_medico → nombre, apellido, especialidad, telefono, email

Tabla Turno

No existen dependencias transitivas.

Dependencias funcionales:

id_turno → id_paciente, id_medico, fecha_hora, estado

Conclusión: Todas las tablas están en 3FN.

Justificación Detallada

Tabla Paciente

Descripción: Almacena los datos personales de los pacientes.

Clave primaria: id_paciente.

Cada paciente es identificado de forma única por id_paciente.

Dependencias funcionales:

id_paciente → nombre, apellido, edad, direccion, telefono, email

Tabla Medico

Descripción: Almacena los datos personales y profesionales de los médicos.

Clave primaria: id_medico.

Cada médico es identificado de forma única por id_medico.

Dependencias funcionales:

id_medico → nombre, apellido, especialidad, telefono, email

Tabla Turno

Descripción: Registra los turnos programados entre pacientes y médicos.

Clave primaria: id_turno.

Claves foráneas:

id_paciente referencia a Paciente(id_paciente).

id_medico referencia a Medico(id_medico).

Dependencias funcionales:

id_turno → id_paciente, id_medico, fecha_hora, estado

Integridad Referencial

Las relaciones entre tablas se establecen mediante claves foráneas:

Turno.id_paciente referencia a Paciente.id_paciente.

Turno.id_medico referencia a Medico.id_medico.

Esto garantiza:

Consistencia: Un turno no puede existir sin un paciente y un médico válidos.

Integridad: Se evita la existencia de referencias huérfanas.

Ventajas del Diseño

Eliminación de Redundancia: Al estar en 3FN, se minimiza la duplicación de datos.

Flexibilidad: Fácil de extender si se requieren nuevas entidades o atributos.

Eficiencia en Consultas: La normalización facilita consultas más rápidas y eficientes.

Mantenimiento Simplificado: Las actualizaciones y eliminaciones afectan solo a las tablas pertinentes sin causar anomalías.

Conclusión

El diseño de la base de datos para el Sistema de Gestión de Hospital se ha realizado, asegurando que:

Las tablas están al menos en Tercera Forma Normal (3FN).

Se garantiza la integridad y consistencia de los datos.

El sistema es escalable y mantenible.

Este diseño sienta las bases para un sistema robusto que puede soportar las operaciones y funcionalidades requeridas por el proyecto, facilitando futuras ampliaciones o mejoras.