

Universidad del Valle de Guatemala
Facultad de Ingeniería
Bases de datos 1



Proyecto 3

Dilary Sarahí Cruz López – 231010

Pablo André Orellana Mijangos – 20555

Reflexión Dilary

1. ¿Qué criterios usamos para definir entidades y relaciones?

Primero identificamos los elementos clave en una clínica: pacientes, médicos, citas, historial, especialidades y medicamentos. Nos enfocamos en lo necesario para registrar y dar seguimiento a la atención médica. Dejamos fuera procesos como facturación o manejo de insumos, porque iban más allá del alcance del proyecto. Las relaciones las hicimos pensando en cómo se conectan las cosas en la vida real, pero cada cita pertenece a un solo médico y un solo paciente.

2. ¿Qué tan bien funcionaron las claves primarias y foráneas?

Funcionaron muy bien. Usamos identificadores tipo SERIAL como claves primarias, lo cual fue útil para mantener orden y evitar duplicados. Las foráneas ayudaron a conectar bien todo: pacientes con citas, médicos con especialidades, etc. Esto hizo que las consultas fueran más fáciles y seguras, porque nos aseguramos que no quedaran registros sueltos ni datos rotos.

3. ¿Aplicamos normalización? ¿Fue útil?

Sí, aplicamos la normalización hasta 3FN. Eso nos ayudó a eliminar duplicación de datos como repetir especialidades en varias citas y a mantener consistencia. Lo bueno fue que las tablas quedaron más limpias y entendibles. Lo malo es que para hacer algunas consultas complejas tuvimos que usar varios joins, pero nada grave.

4. ¿Qué restricciones del negocio implementamos directamente en la BD?

Usamos varias:

- NOT NULL en campos importantes como nombre del paciente o fecha de cita.
- CHECK para validar opciones, Tipo; sexo: Masculino/Femenino/Otro)
- DEFAULT para algunos campos no obligatorios
- UNIQUE en correos electrónicos
- Claves foráneas para mantener integridad
- Y un TRIGGER que actualiza automáticamente el historial del paciente cuando se registra una nueva cita.

Todo esto lo hicimos para asegurar que la base de datos no acepte datos erróneos o incompletos.

5. ¿Ventajas y desventajas del modelo para consultas complejas?

Una ventaja grande fue que la estructura estaba clara, así que pudimos hacer consultas con filtros y joins sin mucho problema. También, buscar pacientes que han sido atendidos por cierto médico en cierto mes fue bastante directo. Como desventaja, algunas consultas sí se volvieron largas por la cantidad de tablas conectadas, pero eso es normal en bases bien normalizadas. Aun así, el modelo fue flexible y escalable.

6. ¿Qué cambiaríamos si esto fuera para producción?

Si fuera un sistema real, mejoraríamos varias cosas:

- Agregaríamos índices para acelerar consultas
- Validaríamos que no haya solapamiento de citas, o sea más lógica en triggers o funciones.

- Revisaríamos los tipos de datos para que ocupen el espacio justo
- Y tal vez usaríamos particiones o técnicas más avanzadas si el volumen de datos crece mucho.

Reflexión Pablo

1. Criterios utilizados para definir entidades y relaciones

El equipo decidió modelar un sistema para el control de una clínica médica, tomando en cuenta los procesos más relevantes: registro de pacientes, control de citas médicas y asignación de médicos a especialidades. Se identificaron entidades clave como paciente, médico, cita y especialidad, así como relaciones 1:N y N:M representadas correctamente con tablas de cruce. Se buscaron simplificar algunos aspectos sin perder el enfoque en la funcionalidad real del sistema, cuidando que cada tabla cumpliera una función específica.

2. Evaluación de las claves primarias y foráneas

Las claves primarias fueron definidas como enteros autoincrementales para garantizar su unicidad y facilitar las referencias. Las claves foráneas permitieron conectar eficientemente las tablas de manera lógica, asegurando la integridad referencial. Estas decisiones facilitaron la construcción de consultas complejas y permitieron mantener una estructura organizada y coherente en la base de datos.

3. Aplicación de la normalización

El equipo aplicó las tres primeras formas normales para garantizar la integridad y eficiencia de la base de datos. Se eliminaron redundancias, se separaron atributos multivaluados y se aseguraron dependencias funcionales correctas. Esto permitió tener un modelo optimizado, fácil de mantener y escalable. La normalización también contribuyó a que las consultas se pudieran construir de manera clara y lógica.

4. Implementación de restricciones y reglas de negocio

Desde la estructura de la base de datos se incluyeron restricciones como NOT NULL, CHECK, UNIQUE y DEFAULT para asegurar la validez de los datos. También se implementaron varios triggers que permiten automatizar acciones como la validación de datos o la actualización de registros relacionados. Estas reglas fueron integradas directamente en la base para reforzar la lógica del negocio y reducir errores del usuario.

5. Evaluación del modelo al realizar consultas complejas

Durante la construcción de los reportes, el equipo comprobó que el modelo diseñado respondía correctamente a consultas con múltiples filtros, agrupaciones y ordenamientos. Gracias al uso adecuado de relaciones y claves, fue posible generar reportes con filtros por fecha, médico, especialidad y sexo del paciente. Sin embargo, se identificó que el rendimiento podría verse afectado si el volumen de datos aumentara considerablemente, lo cual se tendría que atender en una versión más robusta del sistema.

Universidad del Valle de Guatemala

Facultad de Ingeniería

Bases de datos 1



6. Cambios que se realizarían si el sistema se llevara a producción

En caso de implementar el sistema en un entorno real, se considerarían mejoras como el uso de índices para optimizar las búsquedas, control de sesiones de usuario y mecanismos de autenticación. Además, se reforzaría la seguridad en la conexión a la base de datos y se optimizaría la experiencia del usuario en la interfaz gráfica. También se podría implementar paginación en los reportes y agregar más filtros avanzados para agilizar el análisis de datos.

Este proyecto representó una experiencia completa que integró el modelado lógico, la programación SQL, y la conexión de la base de datos con una aplicación web dinámica. El trabajo en equipo permitió distribuir tareas y resolver problemas de forma colaborativa, resultando en un sistema funcional que cumplió con todos los requerimientos planteados.

Link de github

<https://github.com/dils1809/Proyecto-3---BD>