**1. Criterios para decidir entidades y relaciones**

Para definir las entidades y relaciones, nos enfocamos en los procesos principales que se dan en una clínica, como la atención médica, la gestión de pacientes y el control de diagnósticos y medicamentos. A partir de esto, identificamos entidades como *pacientes*, *médicos*, *citas*, *diagnósticos*, *medicamentos*, *exámenes* y *facturas*. Las relaciones se diseñaron a partir de cómo estas entidades interactúan: por ejemplo, un paciente agenda una cita con un médico, y esa cita puede generar un diagnóstico que a su vez puede estar vinculado a medicamentos o exámenes. Se tomaron decisiones para abstraer aspectos complejos, como el detalle exacto de procedimientos médicos, para centrarnos en una estructura de base de datos funcional y escalable.

**2. Adecuación de claves primarias y foráneas**

Las claves primarias fueron definidas correctamente para asegurar la unicidad de los registros en cada tabla. Las claves foráneas permitieron establecer relaciones claras entre las tablas y garantizar la integridad referencial. Por ejemplo, *id\_paciente* en la tabla de *citas* permite asegurar que solo se registren citas para pacientes existentes. Estas claves también facilitaron la creación de consultas con múltiples JOINs, lo que ayudó a obtener información más completa y estructurada de los datos.

**3. Aplicación de la normalización**

Aplicamos los tres primeros niveles de normalización (1FN, 2FN y 3FN) para mejorar la organización de los datos:

* **1FN**: Se eliminó la duplicidad de columnas y todos los valores se mantuvieron atómicos.
* **2FN**: Separación de datos que dependían solo de una parte de una clave compuesta.
* **3FN**: Separación de atributos que dependían transitivamente de otras columnas, como las especialidades de los médicos.  
  Esto permitió reducir la redundancia y mejorar la consistencia de los datos. La principal desventaja fue que, al estar tan normalizada, algunas consultas complejas requerían varios JOINs, lo cual podría afectar el rendimiento si el volumen de datos aumenta.

**4. Restricciones y reglas del negocio**

Implementamos varias restricciones y reglas directamente en la base de datos para garantizar la calidad e integridad de los datos:

* **CHECK** para validar ciertos campos, como los estados de las citas.
* **DEFAULT** en campos como la fecha de registro de pacientes.
* **NOT NULL** para asegurarnos de que campos críticos como nombre o correo no quedaran vacíos.
* **UNIQUE** en campos que requieren unicidad, como el correo electrónico de los médicos.
* **Claves foráneas** para asegurar que no se puedan insertar datos inconsistentes.
* **Triggers** para automatizar acciones, como actualizar el stock de medicamentos o calcular totales en facturas.  
  Estas restricciones reflejan reglas del negocio y ayudan a automatizar procesos importantes, minimizando errores manuales.

**5. Ventajas y desventajas en consultas complejas**

El modelo relacional fue bastante flexible para realizar consultas complejas con múltiples filtros y agrupaciones. Por ejemplo, pudimos generar reportes sobre diagnósticos frecuentes o ingresos mensuales. Las relaciones bien definidas y el uso de claves foráneas facilitaron estas consultas. Como desventaja, algunas consultas con múltiples JOINs resultaron pesadas, lo cual podría representar un problema de rendimiento con bases de datos de gran tamaño.

**6. Cambios para escalar a producción**

Para escalar este modelo a un entorno de producción, haríamos varios ajustes:

* **Volumen de datos**: Aplicar particionamiento en tablas grandes como *citas* o *facturas*.
* **Rendimiento**: Crear índices en columnas clave como *fecha\_hora* o *id\_paciente*, y considerar desnormalizar ciertas tablas para consultas críticas.
* **Integridad**: Añadir mecanismos de auditoría para monitorear cambios en datos sensibles.
* **Escalabilidad**: Evaluar la posibilidad de migrar a un sistema distribuido si el crecimiento de los datos lo requiere.  
  Además, implementar cachés para consultas frecuentes puede reducir el tiempo de respuesta en reportes complejos.