Universidad Peruana Los Andes

Facultad de Ingeniería

Escuela Profesional de Ingeniería de Sistemas y Computación

Curso: Base de Datos II

Docente: Raúl Fernández Bejarano

Estudiante: Sarmiento Mosquera Yeims Abraham

V Ciclo - Código: s03807f

Huancayo – 2025

Documento de Diseño Lógico

Definición de la Arquitectura Seleccionada

Arquitectura: Cliente-Servidor de Tres Capas.

Justificación: Permite una mejor escalabilidad (añadir más usuarios o funciones sin saturar la base de datos), seguridad (la lógica de negocio actúa como intermediario) y mantenibilidad (cada capa puede ser actualizada independientemente). Es ideal para un hospital donde el *servidor de base de datos* es el centro crítico de la operación.

Componentes Fundamentales Explicados

1. **Servidor de Base de Datos:** Contiene la información crítica (Expedientes Electrónicos, Citas, Facturación). Garantiza consistencia y disponibilidad.
2. **Servidor de Aplicaciones (Lógica de Negocio):** Gestiona la validación de datos, las transacciones médicas, y la comunicación entre la interfaz de usuario y la base de datos.
3. **Clientes (Estaciones de Trabajo/Móviles):** Puntos de acceso para el personal médico.

Justificación en Relación con los Requisitos del Sistema

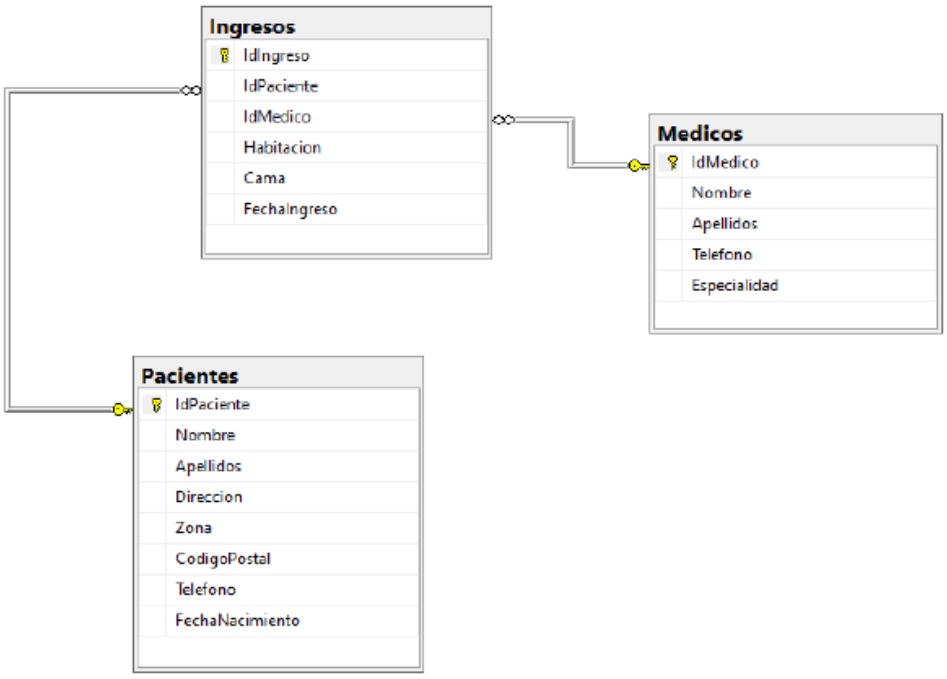
- **Requisito de Seguridad:** La arquitectura de tres capas centraliza la lógica de negocio y los datos, facilitando la aplicación de políticas de acceso estricto (ej: sólo médicos pueden ver diagnósticos, sólo facturación puede ver pagos).
- **Requisito de Concurrencia:** Al separar la lógica del servidor de datos, múltiples usuarios (médicos, administrativos) pueden acceder y actualizar registros (citas, historial) simultáneamente sin degradar el rendimiento del servidor de la base de datos.

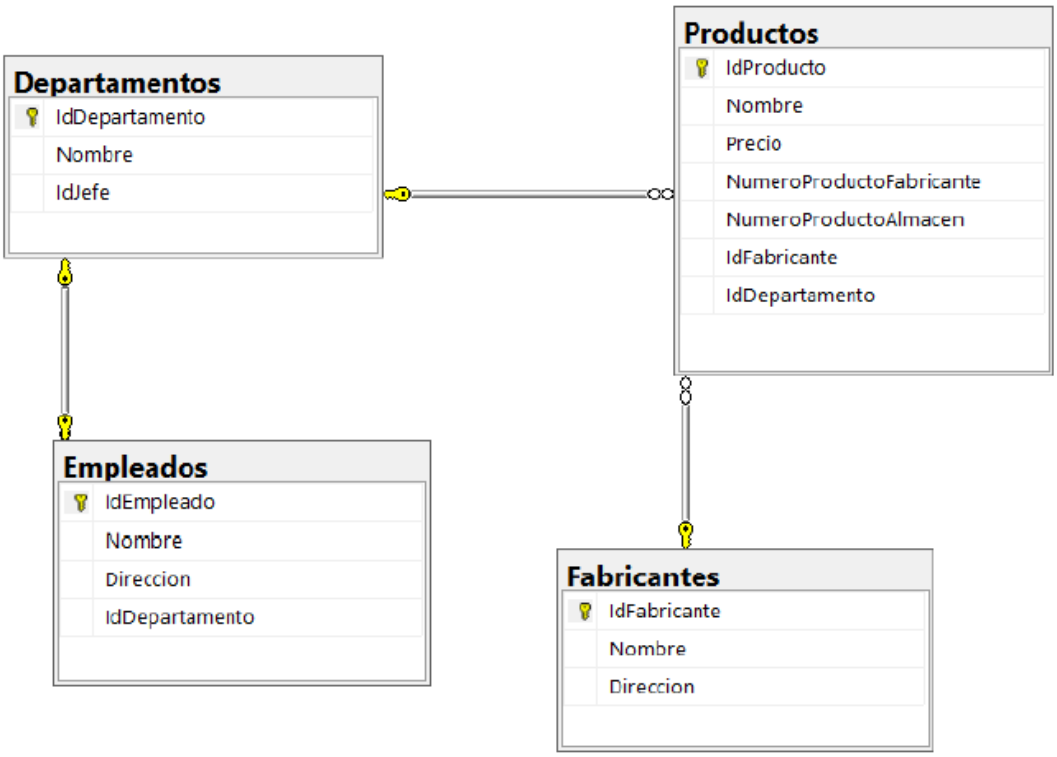
Para que el sistema de gestión hospitalaria funcione, necesitamos al menos estas 4 entidades principales:

1. Médicos (El recurso humano que proporciona el servicio).
2. Pacientes (Los usuarios que reciben la atención).
3. Citas (La relación de tiempo y espacio entre el médico y el paciente).
4. Expediente Médico (El historial de la atención del paciente).

Relaciones Clave:

1. **Médicos : Citas** → **1:N** (Un médico puede tener muchas citas).
2. **Pacientes : Citas** → **1:N** (Un paciente puede tener muchas citas).
3. **Pacientes : Expediente_Médico** → **1:N** (Un paciente puede tener varios registros en su expediente).
4. **Médicos : Expediente_Médico** → **1:N** (Un médico genera muchos registros de expedientes).





Modelos Conceptual, Lógico y Físico

Modelo	Descripción	En el caso del Hospital
Conceptual	Describe la realidad del negocio. (Independiente de la tecnología).	Las 4 entidades que definimos: **Médicos, Pacientes, Citas, Expediente_Médico**, y sus relaciones.
Lógico	Estructura los datos en tablas, atributos y claves. (Independiente del SGBD).	**Las Tablas y Atributos** detallados en el punto 4 de la Evidencia 2.
Físico	Implementación real. (Dependiente del SGBD, ej: MySQL).	Definición de los **tipos de datos** y **restricciones** específicos (Ej: `Nombre` como `VARCHAR(50)`, `id_Medico` como `INT PRIMARY KEY`).

Justificación de normalización

Justificación: Normalización (ej. 3NF) es esencial para evitar la redundancia de datos, anomalías de inserción/actualización/eliminación y mantener la integridad de los datos.

- Ejemplo: En tu diseño, la información del Médico (Nombre, Especialidad) se almacena solo una vez en la tabla Médicos. En la tabla Citas, solo se guarda la clave foránea id_Medico. Si no

normalizamos y guardamos el nombre del médico en la tabla Citas, tendríamos que actualizar cientos de registros de citas si el médico cambia su nombre o especialidad.