

Asignación No. 5: Diseño Centrado en el Usuario y Prototipado de Baja Fidelidad

Jaime Alberto Silva Rodriguez

Broward International University – BIU

Master of Science in Computer Software Engineering

Human-Computer Integration and Digital Citizenship

Doctor Walter Marcelo Fuertes

Bogotá, Colombia

abril de 2025

Resumen

Este presenta los diagramas de diseño para la aplicación web de MediSync, nuestro Sistema de Gestión de Citas de Especialidades Médicas centrado en HCI.

Introducción

Antes de iniciar un proceso de desarrollo, es necesario enfrentar la fase de diseño, HCI va hasta la implementación, las pruebas y la mejora continua, sin embargo, su fase más eficiente según la visión del autor se da en esta etapa del desarrollo, abordemos los diferentes aspectos relevantes.

Diseño Conceptual

El diseño conceptual describe la estructura general de la aplicación desde la perspectiva del usuario, incluyendo cómo interactúan con ella y cómo se organizan las funcionalidades.

- **Metáfora o Analogía:** Puedes presentar la aplicación como una "agenda médica digital". Esto significa que los usuarios (pacientes, médicos, administradores) pueden programar, consultar y gestionar citas de manera similar a una agenda física, pero con las ventajas de la automatización, como notificaciones y accesibilidad en línea.
- **Modelo de Interacción:** La interacción será a través de una interfaz web intuitiva con formularios, botones y menús claros. Por ejemplo, tareas como "Programar cita" o "Consultar historial" se realizarán mediante clics y entradas de datos.
- **Estructura General:** Define las secciones principales de la aplicación:
 - **Inicio de Sesión:** Para autenticar usuarios.
 - **Página Principal:** Con accesos rápidos a las funcionalidades clave.
 - **Gestión de Pacientes:** Registro y consulta de pacientes.
 - **Gestión de Médicos:** Registro y consulta de médicos.
 - **Programación de Citas:** Para agendar citas.
 - **Historial de Citas:** Para revisar citas pasadas.

Modelo Relacional de la Base de Datos

Entidades principales derivadas del diagrama

A partir del diagrama de componentes y los requerimientos funcionales, las entidades centrales son:

Tabla 1 Componentes clave del modelo entidad relación

Entidad	Descripción
Paciente	Persona que agenda y recibe atención médica.
Médico	Profesional que brinda atención, con especialidad y horario.

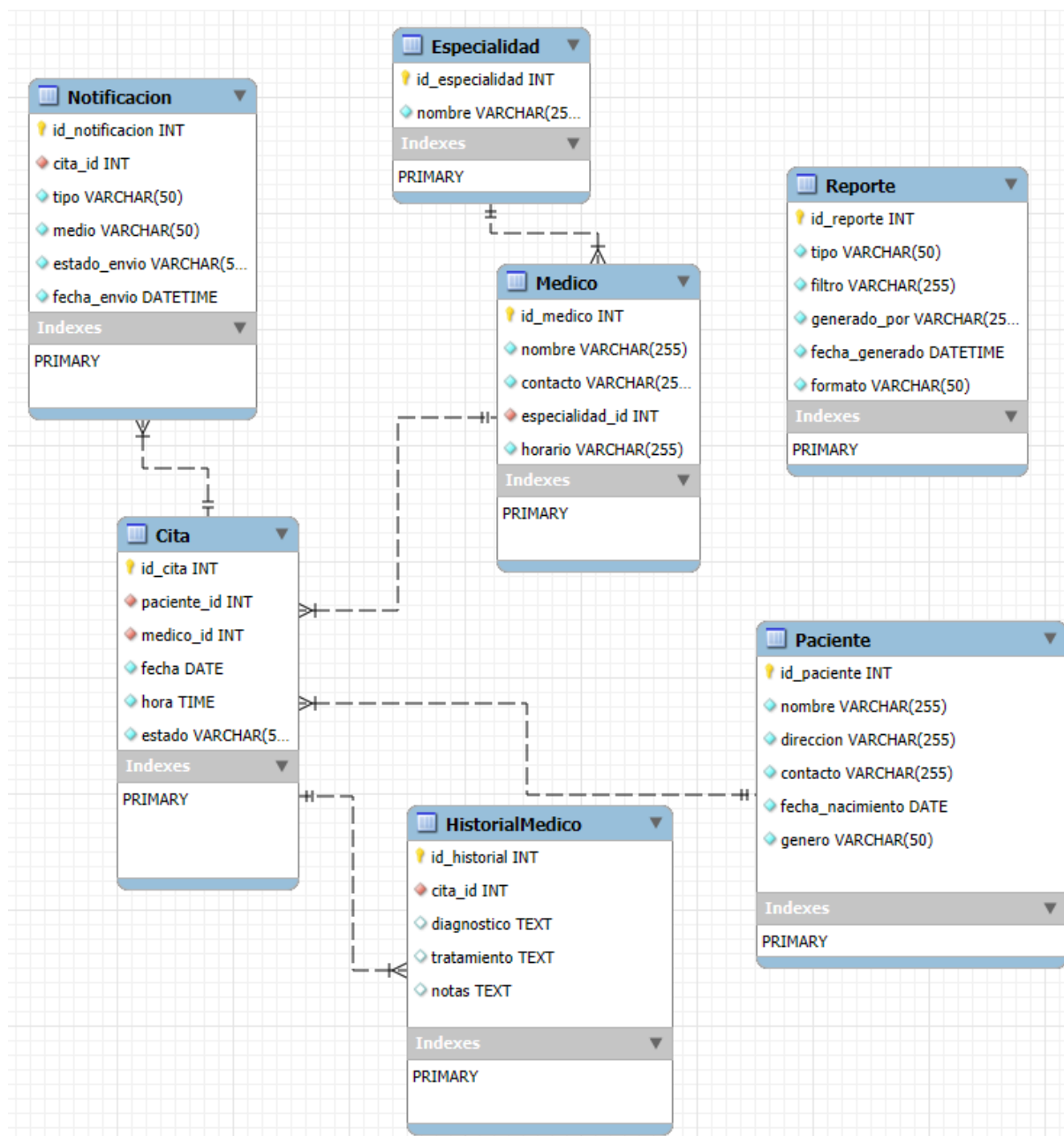
Especialidad	Área médica asignada a médicos y usada en citas.
Cita	Encuentro entre paciente y médico en fecha/hora específica.
Historial Médico	Registro clínico de citas, diagnósticos y tratamientos.
Notificación	Mensaje automático enviado a paciente o médico.
Reporte	Consolidado de información sobre citas.

Fuente: Elaboración propia.

Relaciones clave

- Un **paciente** puede tener muchas **citas**.
- Un **médico** puede tener muchas **citas** y está vinculado a una **especialidad**.
- Una **cita** tiene un **diagnóstico/tratamiento**, que se convierte en parte del **historial médico**.
- **Notificaciones** están asociadas a **citas**.
- **Reportes** se generan a partir de **citas**, agrupadas por filtros.

Ilustración 1 Modelo Entidad - Relación de MediSync



Fuente: Elaboración propia. Elaborado en MySQL Workbench

Diagrama de casos de uso

Actores identificados:

- **Administrador**
- **Administrativo**
- **Paciente**
- **Médico**
- **Sistema** (como actor secundario para automatización)

Casos de uso principales:

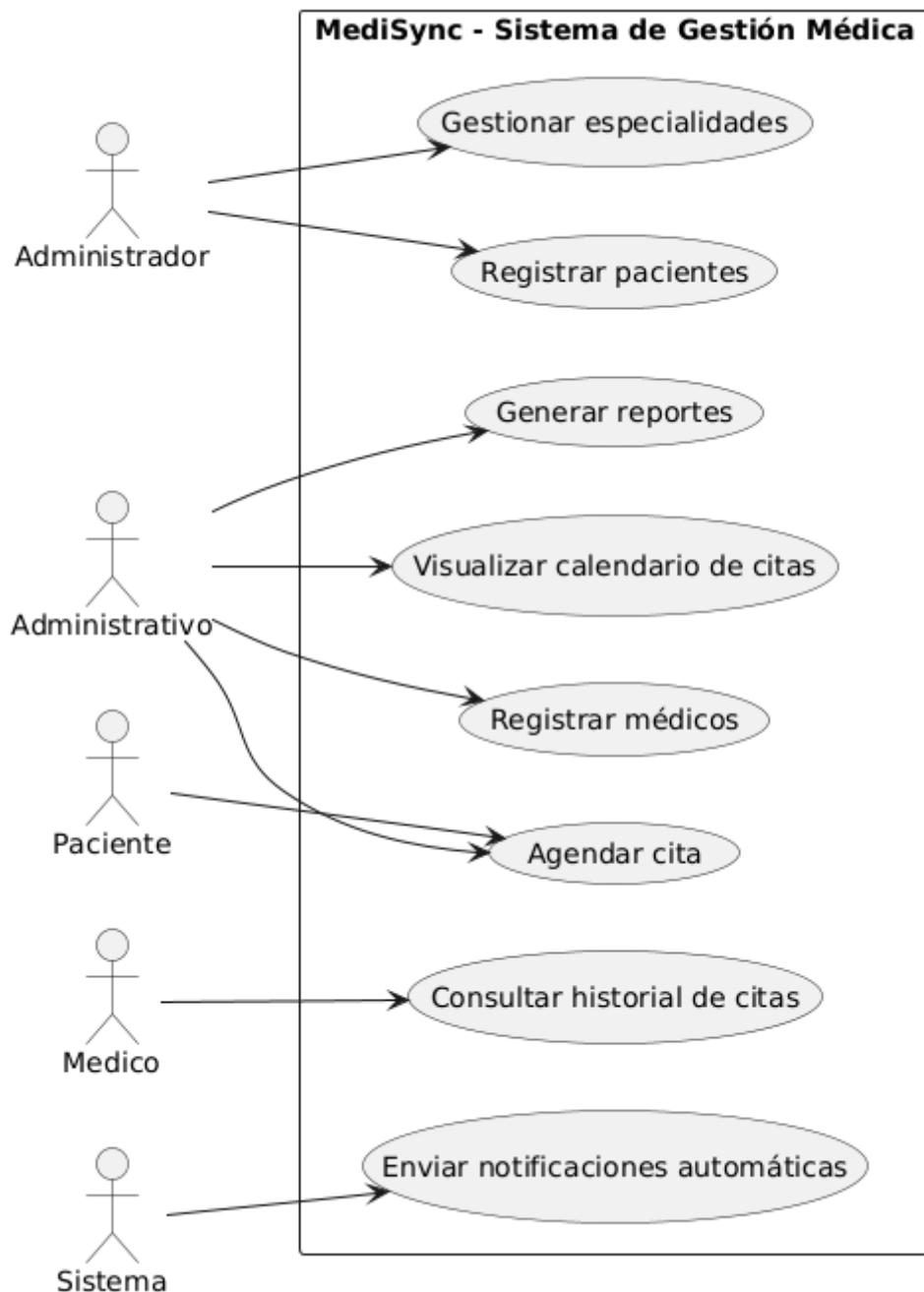
Tabla 2 Resumen de los casos de uso por actor

Actor	Casos de Uso
Administrador	<ul style="list-style-type: none">- Registrar pacientes- Gestionar especialidades
Administrativo	<ul style="list-style-type: none">- Registrar médicos- Agendar citas- Visualizar calendario de citas- Generar reportes
Paciente	<ul style="list-style-type: none">- Agendar citas
Médico	<ul style="list-style-type: none">- Consultar historial de citas
Sistema	<ul style="list-style-type: none">- Enviar notificaciones

Fuente: Elaboración propia.

Una vez están clasificados y definidos los casos de uso principales, se procede a la generación del diagrama de casos de uso definido en la Ilustración 2.

Ilustración 2 Diagrama de casos de uso de MediSync



Fuente: Elaboración propia.

Diseño de la Arquitectura de software

Estilo arquitectónico propuesto: Arquitectura en Capas (N-Tier)

Esta arquitectura organiza el sistema en capas independientes, lo cual facilita la separación de responsabilidades y el mantenimiento.

Capas:

1. **Capa de Presentación (Frontend)**
2. **Capa de Aplicación / Lógica de Negocio**

3. **Capa de Servicios / Integración**
4. **Capa de Persistencia / Acceso a Datos**
5. **Base de Datos**

Componentes por Capa

Presentación (Frontend Web)

- Interfaces para:
 - Pacientes (agendamiento)
 - Médicos (historial de citas)
 - Administrativos (reportes, calendarios)
 - Administrador (gestión de usuarios y especialidades)
- Validaciones básicas en el cliente
- Acceso mediante autenticación

Aplicación / Lógica de Negocio (Backend)

- Módulo de gestión de pacientes
- Módulo de gestión de médicos
- Módulo de citas médicas
- Módulo de historial médico
- Módulo de reportes
- Módulo de notificaciones
- Módulo de autenticación/autorización

Servicios / Integraciones

- Servicio de envío de correos (SMTP)
- Servicio de envío de SMS (Twilio o similar)
- API de Google Calendar (solo lectura)
- API RESTful para clientes web y móviles

Persistencia / DAO

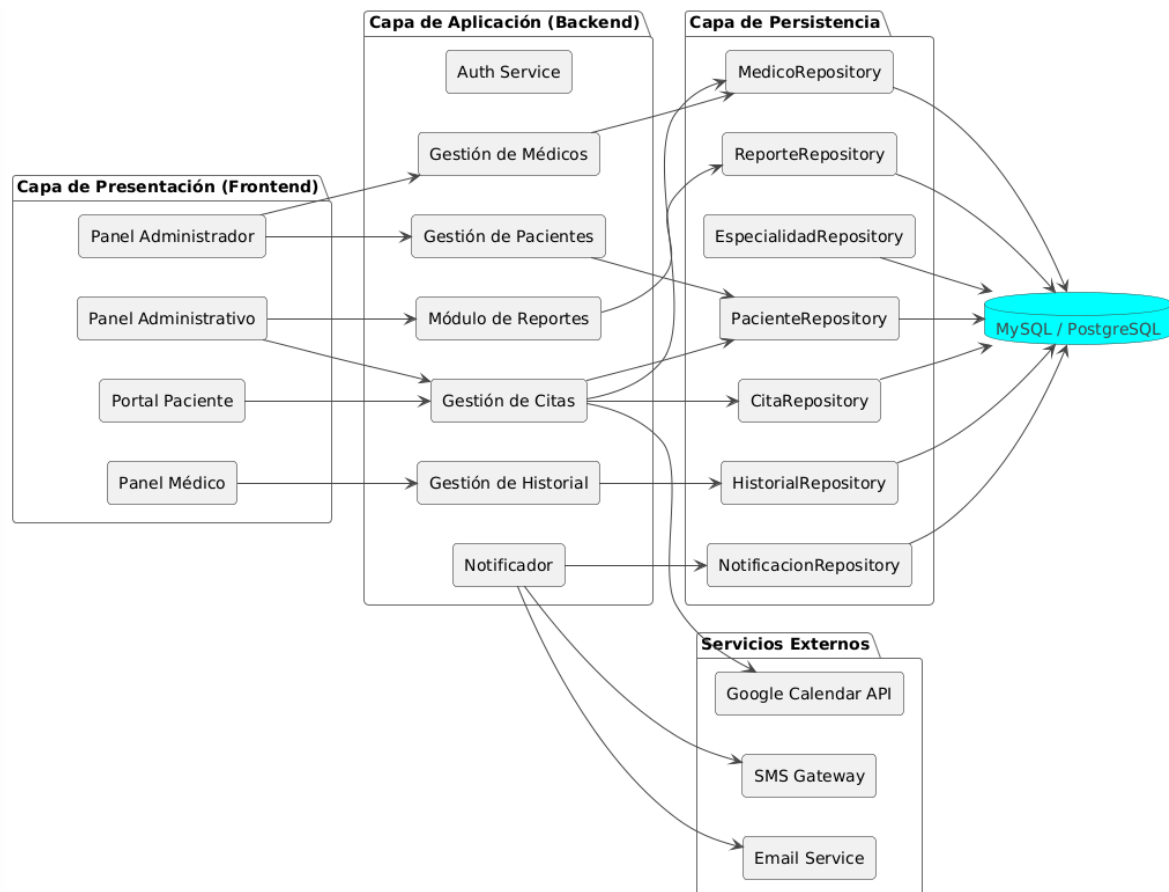
- Repositorios para cada entidad:
 - PacienteRepository

- MedicoRepository
 - CitaRepository
 - HistorialRepository
 - EspecialidadRepository
 - NotificacionRepository
 - ReporteRepository
- ORM (por ejemplo, Hibernate, Sequelize, etc.)

Base de Datos Relacional

- MySQL / PostgreSQL
- Modelo entidad-relación (ya definido)

Ilustración 3 Diagrama de componentes del sistema a desarrollar

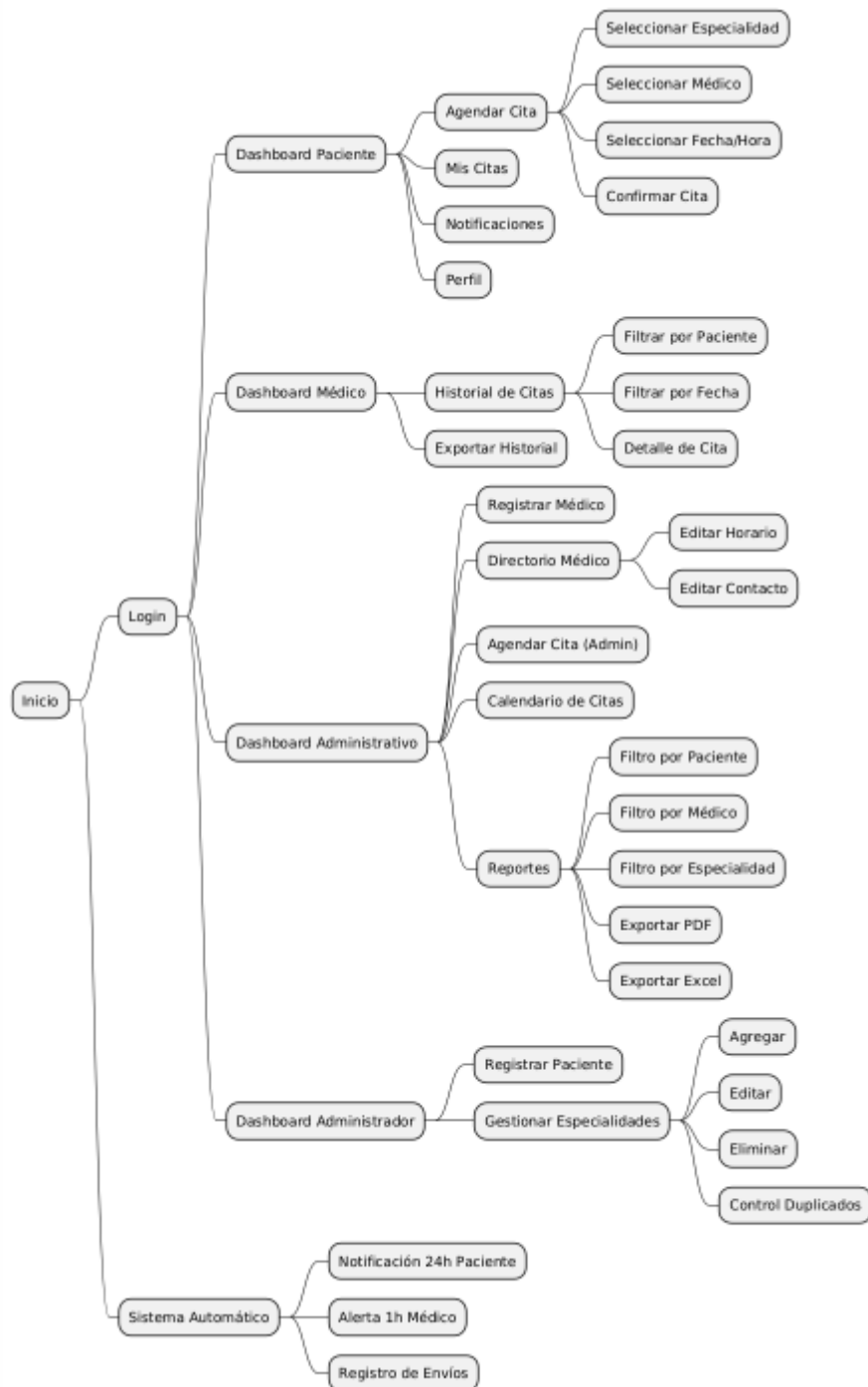


Fuente: Elaboración propia.

Mapa de navegación

Se procede a presentar el mapa de navegación ajustado, respecto a la anterior entrega.

Ilustración 4 Mapa de navegación en modelo de mapa mental



Fuente: Elaboración propia.

De esta manera, se presenta la base para iniciar la fase de implementación.

Conclusión

La creación del modelo entidad-relación hasta el mapa de navegación ha sido un proceso clave para estructurar el sistema de gestión médica de manera efectiva. Partiendo de la definición de entidades como pacientes, médicos, citas y especialidades, se estableció una relación clara entre ellas, lo que facilitó la comprensión de los flujos de datos y las operaciones necesarias para gestionar las interacciones. Con el modelo ER como base, fue posible derivar un diagrama de casos de uso que cubriera los requisitos de los usuarios del sistema, tales como pacientes, médicos y administradores, asegurando que todas las funcionalidades clave fueran bien documentadas. Posteriormente, el diseño de la arquitectura de software y el mapa de navegación permitió dar forma a la interfaz del sistema, estableciendo una estructura jerárquica clara para los diferentes roles y asegurando que los usuarios pudieran navegar fácilmente a través de las diversas funcionalidades del sistema.

Este enfoque integrado desde la base de datos hasta la interfaz de usuario asegura que el sistema esté bien organizado y sea intuitivo para todos los usuarios. Además, la creación del mapa de navegación, organizado verticalmente, proporciona una forma clara y eficiente de visualizar cómo los usuarios interactuarán con el sistema en función de sus roles, lo que a su vez facilita la implementación del sistema y su uso. Este proceso asegura que tanto la funcionalidad como la usabilidad sean prioritarias, haciendo que el sistema sea robusto y fácil de mantener a largo plazo.

Referencias

Sommerville, I. (2011). Software Engineering (9th ed.). Boston: Addison-Wesley.

Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). New York: McGraw-Hill.