

# Escritos de entregables

## Justificación de la Normalización (hasta BCNF)

El diseño de la base de datos se ha sometido a un proceso de normalización para asegurar la integridad de los datos, reducir la redundancia y evitar anomalías de actualización, inserción y borrado.

- **Primera Forma Normal (1FN):** El esquema cumple con la 1FN. Cada columna en cada tabla contiene valores atómicos (indivisibles) y cada registro es único, identificado por su llave primaria. No existen grupos de repetición; por ejemplo, los medicamentos de una receta no se almacenan en una sola columna, sino en una tabla separada (receta\_items).
- **Segunda Forma Normal (2FN):** El esquema cumple con la 2FN. Todas las tablas están en 1FN y cada atributo no-clave depende funcionalmente de la totalidad de la llave primaria. En las tablas con llaves primarias simples (medicos, pacientes, recetas, etc.), esta forma se satisface automáticamente. En la tabla medico\_paciente, que tiene una llave candidata compuesta (medico\_id, paciente\_id), el único atributo no-clave (asociacion\_fecha) depende de ambos, representando el momento exacto de la asociación.
- **Tercera Forma Normal (3FN):** El esquema cumple con la 3FN. Está en 2FN y no existen dependencias transitivas. Es decir, ningún atributo no-clave depende de otro atributo no-clave. Por ejemplo, en la tabla recetas, los detalles del médico (como su nombre) no se almacenan directamente; en su lugar, se usa una llave foránea (medico\_id). Esto evita que la información del médico se repita y dependa transitivamente de la receta a través del ID del médico.
- **Forma Normal de Boyce-Codd (BCNF):** El esquema cumple con la BCNF, una versión más estricta de la 3FN. Para cada dependencia funcional no trivial  $X \rightarrow Y$ ,  $X$  es una superllave (una llave candidata o un superconjunto de ella). En todas las tablas del modelo, los únicos determinantes de atributos son las llaves candidatas (como id en la mayoría de las tablas, o el par (medico\_id, paciente\_id) en medico\_paciente). No existen dependencias donde un atributo no-clave determine a otro, por lo que el diseño satisface los requisitos de BCNF, garantizando un nivel muy alto de integridad y eficiencia.

- Planeación SCRUM

- Product Backlog

- Este es un listado priorizado de las funcionalidades y requisitos del sistema.

		Prioridad
•	• User Story	
•	• Como médico, quiero iniciar sesión de forma segura para acceder a mis datos.	• Alta
•	• Como médico, quiero ver una lista de todos mis pacientes asignados para tener una vista general.	• Alta
•	• Como médico, quiero poder agregar un nuevo paciente al sistema y asignármelo.	• Alta
•	• Como médico, quiero poder dar de alta a un paciente (desvincularlo) cuando ya no lo atienda.	• Mediana
•	• Como médico, quiero crear una receta médica para un paciente	• Alta

seleccionado.		
•	• Como médico, quiero buscar medicamentos de una base de datos al crear una receta.	• Alta
•	• Como médico, quiero que los nuevos medicamentos se guarden automáticamente si no existen.	• Media
•	• Como médico, quiero que la receta generada se descargue como un archivo PDF profesional.	• Alta
•	• Como médico, quiero poder cerrar mi sesión de forma segura.	• Alta
<ul style="list-style-type: none"> <li>• Planeación de Sprints</li> <li>• Se proponen dos Sprints para desarrollar las funcionalidades clave.</li> <li>• Sprint 1: "Core y Gestión de Pacientes" (Duración: 1 semana)</li> <li>• Objetivo: Implementar la autenticación básica y la gestión completa de pacientes.</li> <li>• Tareas:</li> <li>• Configuración del Proyecto: Inicializar Node.js, Express, Sequelize y estructura de carpetas. (Asignado: Backend Dev)</li> <li>• US-1 (Login): Crear modelo de datos, controlador y ruta para autenticación de médicos. (Asignado: Backend Dev)</li> <li>• US-1 (Login): Diseñar y desarrollar el formulario de inicio de sesión. (Asignado: Frontend Dev)</li> <li>• US-2 (Ver Pacientes): Crear endpoint para obtener la lista de pacientes de un médico. (Asignado: Backend Dev)</li> <li>• US-2 (Ver Pacientes): Desarrollar el dashboard principal que muestra la lista de pacientes. (Asignado: Frontend Dev)</li> <li>• US-3 (Agregar Paciente): Crear endpoint y formulario para registrar nuevos pacientes. (Asignado: Fullstack Dev)</li> <li>• US-4 (Dar de Alta): Implementar endpoint y botón para desvincular un paciente. (Asignado: Fullstack Dev)</li> <li>• US-9 (Logout): Implementar la funcionalidad de cierre de sesión. (Asignado: Backend Dev)</li> <li>• Sprint 2: "Creación y Emisión de Recetas" (Duración: 1 semana)</li> <li>• Objetivo: Desarrollar el flujo completo de creación de recetas, desde el formulario hasta la generación del PDF.</li> <li>• Tareas:</li> <li>• US-5 (Crear Receta): Diseñar el formulario de creación de recetas. (Asignado: Frontend Dev)</li> <li>• US-6 (Buscar Medicamentos): Crear endpoint para obtener la lista de medicamentos. (Asignado: Backend Dev)</li> <li>• US-6 (Buscar Medicamentos): Implementar el buscador con autocompletado en el frontend. (Asignado: Frontend Dev)</li> <li>• US-5 y US-7 (Guardar Receta): Crear el endpoint que procesa y guarda la receta, incluyendo nuevos medicamentos. (Asignado: Backend Dev)</li> <li>• US-8 (Generar PDF): Crear la plantilla HTML para la receta imprimible. (Asignado: Frontend Dev)</li> <li>• US-8 (Generar PDF): Implementar la lógica para obtener los datos y usar html2pdf para la descarga. (Asignado: Fullstack Dev)</li> <li>• Pruebas E2E: Realizar pruebas completas del flujo de creación de recetas. (Asignado: QA/Dev)</li> </ul>		

**Tablero de Progreso (Kanban)**

To Do (Pendiente)	In Progress (En Progreso)	Done (Hecho)
		Sprint 1: Configuración del Proyecto
		Sprint 1: US-1 (Login Backend)
		Sprint 1: US-1 (Login Frontend)
		Sprint 1: US-2 (Ver Pacientes Backend)
		Sprint 1: US-2 (Ver Pacientes Frontend)
		Sprint 1: US-3 (Agregar Paciente)
		Sprint 1: US-4 (Dar de Alta)
		Sprint 1: US-9 (Logout)
		Sprint 2: US-5 (Diseño Formulario Receta)
		Sprint 2: US-6 (Backend Medicamentos)
		Sprint 2: US-6 (Frontend Buscador)
		Sprint 2: US-5 y US-7 (Guardar Receta)
		Sprint 2: US-8 (Plantilla PDF)
		Sprint 2: US-8 (Lógica PDF)
		Sprint 2: Pruebas E2E
(vacío)	(vacío)	