espoch

Tabla de contenido

Resumen Ejecutivo

1 Alcance del Proyecto

Resumen

Este proyecto consiste en el desarrollo de una aplicación web para la gestión clínica y administrativa del Laboratorio Clínico de baja complejidad "La Inmaculada", con el objetivo de automatizar procesos como el registro de pacientes, emisión de órdenes médicas, gestión de resultados clínicos, manejo de inventario de reactivos, cuentas por cobrar y convenios con médicos.

Problema resuelto

El laboratorio realiza actualmente sus procesos de forma manual, lo que genera ineficiencia, errores en registros, falta de trazabilidad, y pérdida de tiempo en la atención al paciente. Esta situación impacta negativamente tanto en la calidad del servicio como en la gestión administrativa interna.

Ámbito de implantación

La solución será implementada en el Laboratorio Clínico La Inmaculada, ubicado en el cantón Guano, provincia de Chimborazo – Ecuador, y operará desde un entorno web alojado en la nube, accesible desde computadoras del personal del laboratorio.

Orientación de la aplicación

Está dirigida al personal interno del laboratorio, incluyendo administradores, recepcionistas y laboratoristas. Los principales beneficios son:

- Mayor eficiencia en el flujo de trabajo
- Reducción de errores humanos
- Mejora en el control de pagos, insumos y resultados
- Mejor atención al paciente

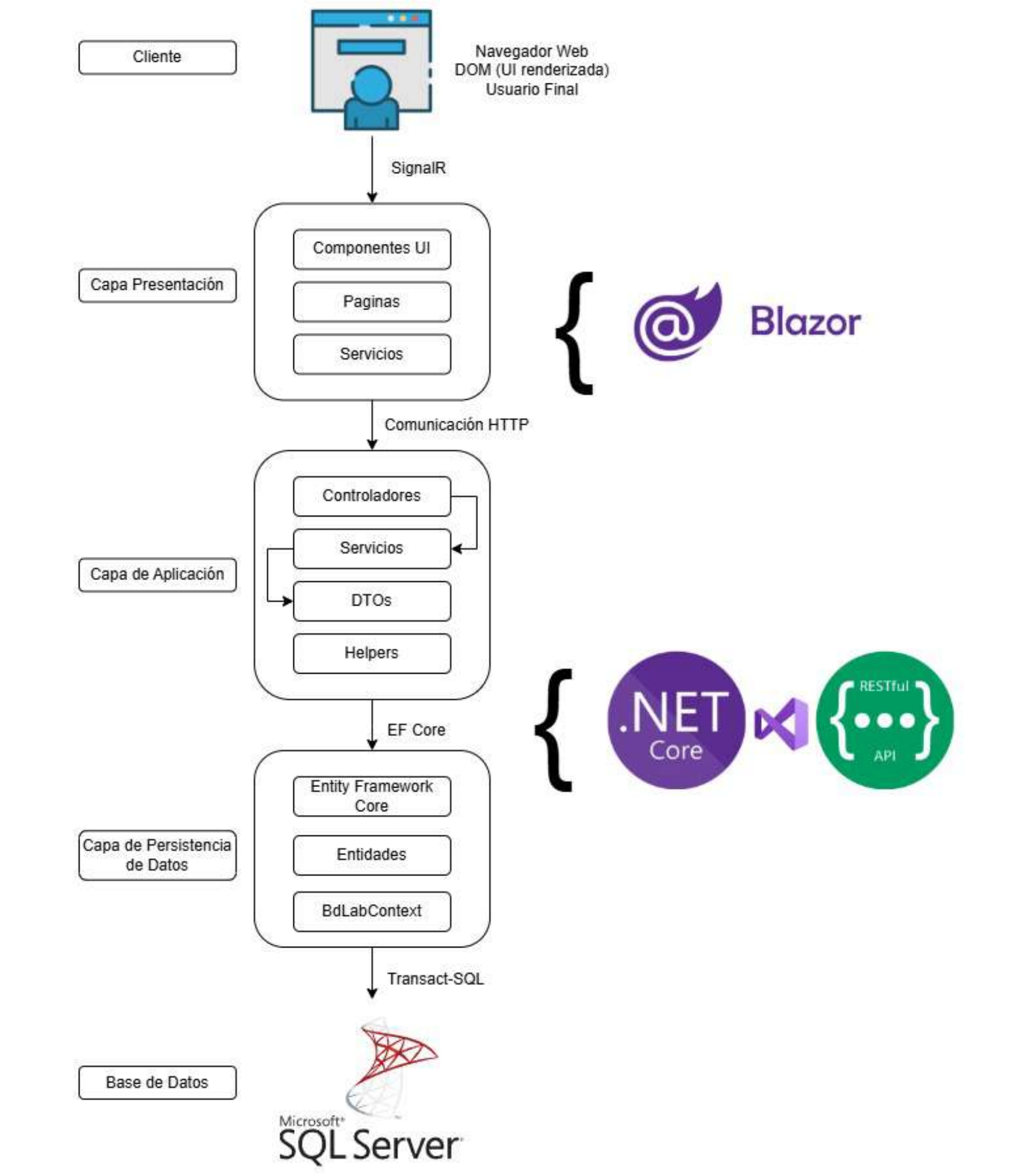
Situaciones abordadas

- Eliminación del registro manual de pacientes, órdenes y pagos
- Automatización del ingreso y validación de resultados clínicos
- Control preciso del stock de reactivos y su uso en exámenes
- Gestión centralizada de convenios médicos y comisiones
- Emisión automatizada de comprobantes y reportes en PDF

2 Diseño de la Aplicación (Arquitectura)

Diagrama de arquitectura de la solución

Arquitectura en capas distribuida (Blazor Server + API REST + EF Core)



Descripción general de los componentes

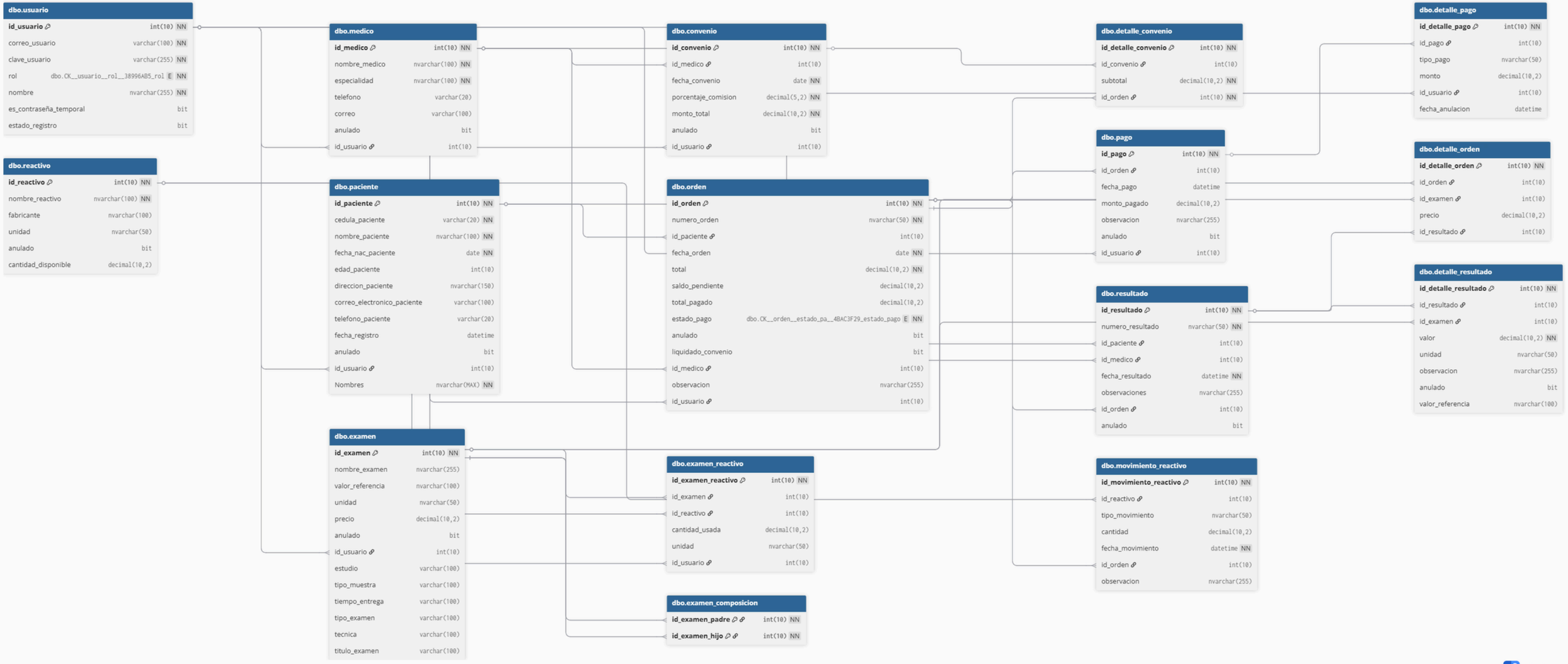
La aplicación está diseñada siguiendo una arquitectura en capas distribuida, que separa claramente la interfaz de usuario, la lógica de negocio, la lógica de acceso a datos y la base de datos. A continuación se describen los componentes clave:

1. **Cliente / Navegador Web**
   - El usuario accede a la aplicación a través de un navegador web.
   - La interfaz está construida con Blazor Server, lo cual permite renderizado en tiempo real mediante SignalR.
2. **Capa de Presentación (Blazor Server)**
   - Contiene Componentes UI, Páginas y Servicios que definen la experiencia visual y lógica del usuario.
   - Se comunica mediante llamadas HTTP con la API.
   - Utiliza servicios inyectables (IApiService) para interactuar con los controladores de la API REST.
3. **Capa de Aplicación (API REST ASP.NET Core)**
   - Aquí residen los Controladores que exponen endpoints HTTP.
   - Incluye Servicios, DTOs (Data Transfer Objects) y Helpers, que manejan la lógica de negocio y transformación de datos.
   - Expone funcionalidades a través de endpoints RESTful.
4. **Capa de Persistencia de Datos (Entity Framework Core)**
   - Utiliza Entity Framework Core (EF Core) como ORM para mapear las entidades del dominio a la base de datos.
   - Contiene las Entidades y el DbContext (BdLabContext).
   - Encapsula la lógica de acceso a datos.
5. **Base de Datos (SQL Server)**
   - Sistema de gestión de bases de datos relacional utilizado es Microsoft SQL Server.
   - Almacena la información estructurada de pacientes, órdenes, resultados, usuarios, etc.
   - La comunicación con EF Core se realiza mediante Transact-SQL.

3 Diseño de Datos

Diagrama de la base de datos

Diseño de Base de Datos



El diseño de la base de datos se ha elaborado bajo un enfoque relacional en SQL Server. A continuación se explican las entidades principales y sus relaciones:

Explicación concisa de las entidades principales y relaciones

Entidades clave

- usuario: Registra los datos de acceso (correo, contraseña, rol) y estado de los usuarios del sistema.
- paciente: Almacena la información personal de los pacientes que acceden a servicios del laboratorio.
- medico: Contiene los datos de los médicos que remiten órdenes y están asociados a convenios.
- orden: Representa una orden médica generada para un paciente. Se enlaza a resultados, pagos y convenios.
- detalle_orden: Relaciona una orden con los exámenes solicitados y su respectivo precio.
- examen: Lista de exámenes clínicos disponibles en el laboratorio, con atributos técnicos.
- resultado: Registra los resultados obtenidos en los análisis clínicos de una orden.
- detalle_resultado: Detalla el valor, unidad y observaciones para cada examen realizado.
- pago: Representa pagos asociados a órdenes médicas, permitiendo múltiples pagos por orden.
- detalle_pago: Divide un pago en partes para registrar tipo de pago, monto y fecha.
- convenio: Almacena convenios entre médicos y el laboratorio (porcentaje de comisión, monto total).
- detalle_convenio: Subdivide convenios para asociarlos a órdenes y cálculos individuales.
- reactivo: Contiene el inventario de reactivos del laboratorio (unidad, fabricante, stock).
- movimiento_reactivo: Registra ingresos y egresos de reactivos, vinculados a órdenes.
- examen_reactivo: Define qué reactivos y qué cantidades se usan en cada examen.
- examen_composicion: Permite jerarquizar exámenes (ej. exámenes compuestos de otros).

Relaciones principales

- Un paciente puede tener muchas órdenes.
- Una orden contiene múltiples detalle_orden (exámenes).

- Una orden puede tener varios pagos y a su vez, estos pueden tener detalle_pago.
- Cada orden genera un resultado global, que se desglosa en varios detalle_resultado.
- Un médico puede estar vinculado a muchos convenios.
- Los convenios pueden aplicarse a múltiples órdenes, a través de detalle_convenio.
- Cada examen puede usar uno o más reactivos (examen_reactivo).
- Cada movimiento de reactivo puede estar relacionado con una orden, por trazabilidad.

4 Planificación de Ejecución (Product Backlog)

Versión final del Product Backlog

El backlog se compone de un conjunto de historias de usuario (HU) y tareas técnicas (HT) priorizadas de acuerdo con la lógica del flujo funcional de la aplicación: desde la configuración técnica inicial, pasando por la gestión de pacientes, exámenes, órdenes, resultados, pagos, convenios y despliegue.

Se organizó en 10 sprints, con estimaciones en horas para las HU y objetivos concretos por iteración.

Pila del Producto

Fecha	Versión	Autor	Organización	Descripción
23/04/2025	1.0	Cristhian Chimbo	-	Primera versión de la pila del producto
24/04/2025	2.0	Cristhian Chimbo	-	Adicción de los Sprint
24/04/2025	2.1	Cristhian Chimbo	-	Replanificación sprint 9 y 10
20/05/2025	3.0	Cristhian Chimbo	-	Ingreso de HU 23-28
13/06/2025	4.0	Cristhian Chimbo	-	Ingreso de HU 29-33
16/06/2025	4.1	Cristhian Chimbo	-	Modificación HU 16: Stock Reactivos
20/06/2025	5.0	Cristhian Chimbo	-	Reorganización y depuración del Sprint 10 con historias técnicas fusionadas
20/06/2025	6.0	Cristhian Chimbo	-	Ingreso de HT-014 y 15

ID	Tarea	Est (Horas)	Sprint	Inicio	Fin
HT-001	Configuración del entorno local	-	1	24 abril	30 abril

HT-002	Configuración del repositorio	-		24 abril	30 abril
HT-003	Instalación y configuración de SQL Server Express	-		24 abril	30 abril
HT-005	Diseño e implementación de base de datos	-	2	1 mayo	7 mayo
HT-004	Creación de proyecto API y acceso a base de datos	-		1 mayo	7 mayo
HT-006	Separación de capas	-		1 mayo	7 mayo
HT-007	Configuración de consumo de API	-		1 mayo	7 mayo
HT-008	Primer servicio Blazor para API	-	3	8 mayo	14 mayo
HT-009	Integración de ASP.NET Identity	-		8 mayo	14 mayo
HU-001	Inicio de sesión (Login)	5		8 mayo	14 mayo
HU-002	Registro manual de nuevo paciente	5	4	15 mayo	21 mayo
HU-003	Preregistro de datos del paciente	6		15 mayo	21 mayo
HU-022	Ingreso de nuevo examen	5		15 mayo	21 mayo
HU-023	Edición de examen	4		15 mayo	21 mayo
HU-024	Visualización de exámenes registrados	3	5	22 mayo	28 mayo
HU-004	Selección de exámenes	8		22 mayo	28 mayo
HU-005	Registro de pago	6		22 mayo	28 mayo
HU-006	Confirmación y guardado de orden	4		22 mayo	28 mayo
HU-007	Impresión de orden médica	5	6	29 mayo	4 junio
HU-008	Visualización de órdenes	3		29 mayo	4 junio
HU-009	Detalle de orden médica	4		29 mayo	4 junio
HU-025	Ingreso de nuevo médico	4		29 mayo	4 junio
HU-026	Visualización de médicos registrados	3		29 mayo	4 junio
HU-027	Edición de datos de médico	4	7	5 junio	11 junio
HU-010	Ingreso de resultados de exámenes	8		5 junio	11 junio
HU-011	Impresión de resultados	5		5 junio	11 junio
HU-012	Anulación de orden	4		5 junio	11 junio
HU-013	Anulación de resultado	4		5 junio	11 junio
HU-014	Visualización de cuentas por cobrar	2	8	12 junio	18 junio
HU-015	Pago de saldo pendiente	4		12 junio	18 junio
HU-028	Creación de reactivos	3		12 junio	18 junio
HU-029	Visualización de reactivos	2		12 junio	18 junio

HU-030	Edición de reactivos	2		12 junio	18 junio
HU-016	Ingreso de Stock de reactivos	4		12 junio	18 junio
HU-032	Ingreso de asociación entre examen - reactivo	3		12 junio	18 junio
HU-033	Visualización de asociación examen- reactivo	2		12 junio	18 junio
HU-017	Egreso de reactivos por exámenes	5		12 junio	18 junio
HU-031	Egreso de reactivos por otros factores	3		12 junio	18 junio
HU-018	Visualización de convenios médicos	2	9	12 junio	18 junio
HU-019	Ingreso de convenios con médicos	4		19 junio	25 junio
HU-020	Cálculo de pago por orden médica	2		19 junio	25 junio
HU-021	Generación de pagos a médicos	6		19 junio	25 junio
HT-014	Mejora y modificación de la interfaz de usuario en Blazor Server	-	10	26 junio	2 julio
HT-015	Verificación interna de funcionalidades antes del despliegue	-		26 junio	2 julio
HT-010	Configuración de CI/CD con GitHub Actions			26 junio	2 julio
HT-011	Despliegue completo en Azure (App + Dominio + Configuración de entorno)	-		26 junio	2 julio
HT-012	Azure SQL Database y migración de datos	-		26 junio	2 julio
HT-013	Configuración de backup automático en Azure SQL Database	-		26 junio	2 julio

Dinámica de evolución del Backlog

Durante el desarrollo del proyecto, el backlog tuvo una evolución iterativa y adaptativa, con los siguientes ajustes relevantes:

Versión 2.0 y 2.1: Se añadieron los sprints como estructura organizativa y se replanificaron los Sprint 9 y 10 para optimizar la carga de trabajo.

Versión 3.0 y 4.0: Se incorporaron nuevas historias de usuario relacionadas con exámenes, médicos y convenios (HU-23 a HU-33).

Versión 4.1: Se ajustó la historia HU-016 para mejorar el flujo de registro de stock de reactivos.

Versión 5.0: Sprint 10 fue reorganizado; se fusionaron historias técnicas redundantes (HT-11, 14 y 15).

Versión 6.0: Se integraron las tareas técnicas finales de mejora de interfaz y verificación antes de despliegue.

5 Seguimiento del Proyecto (Burndown Chart)

El gráfico de Burndown Chart refleja el progreso del proyecto a lo largo de 10 sprints (del 24 de abril al 2 de julio de 2025), abarcando un total de 60 tareas entre historias de usuario y tareas técnicas.