**Historias Técnicas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 23/04/2025 | 1.0 | Cristhian Chimbo | - | Primera versión de historias técnicas |
| 24/04/2025 | 1.1 | Cristhian Chimbo | - | Cambios en las historias técnicas |
| 24/04/2025 | 2.0 | Cristhian Chimbo | - | Separación de desarrollo e implementación final |
| 13/06/2025 | 3.0 | Cristhian Chimbo | - | Fusión de HT 11, 14, 15 en la HT 11 |
| 13/06/2025 | 4.0 | Cristhian Chimbo | - | Fusión de HT 12, 13 en la HT 12 |
| 16/06/2025 | 5.0 | Cristhian Chimbo | - | Cambio de orden e ingreso de HT 14 – HT 15 |

|  |  |  |
| --- | --- | --- |
| HT-001 | Configuración del entorno de desarrollo local para Blazor Server | |
| Descripción | | Configurar correctamente el entorno de desarrollo local para trabajar el proyecto Blazor Server, con todas las herramientas necesarias estén instaladas y funcionando. |
| Criterios de Aceptación | | * Visual Studio configurado con soporte para Blazor Server. * SDK de .NET instalado y funcionando. * El proyecto se puede compilar y ejecutar en local sin errores. |
| Tareas | | * Instalar Visual Studio 2022 o Visual Studio Code con extensiones de C#. * Instalar el SDK de .NET (versión específica, por ejemplo, .NET 8). * Compilar y correr el proyecto localmente para verificar que funciona. |

|  |  |  |
| --- | --- | --- |
| HT-002 | Configuración del repositorio para despliegue de Blazor. | |
| Descripción | | Como desarrollador quiero configurar el repositorio GitHub para el proyecto Blazor Server, incluyendo todos los archivos para su funcionamiento. |
| Criterios de Aceptación | | * El proyecto tiene una rama main organizada. * Los commits se realizan frecuentemente. |
| Tareas | | * Crear repositorio nuevo en GitHub * Crear la rama main * Subir el proyecto Blazor Server. |

|  |  |  |
| --- | --- | --- |
| HT-003 | Instalación y configuración de SQL Server Express para entorno local | |
| Descripción | | Instalar SQL Server Express y asegurar que la instancia esté en ejecución, configurada correctamente y lista para ser usada en el proyecto Blazor Server. |
| Criterios de Aceptación | | * SQL Server Express está instalado exitosamente. * Se puede acceder a la instancia usando SQL Server Management Studio (SSMS). * La instancia local permite crear y administrar bases de datos. |
| Tareas | | * Descargar e instalar SQL Server Express Edition. * Descargar e instalar SQL Server Management Studio (SSMS) para administrar la base de datos. * Configurar la instancia para permitir conexiones locales. * Verificar que el servicio de SQL Server esté corriendo. * Crear una base de datos para pruebas iniciales. |

|  |  |  |
| --- | --- | --- |
| HT-004 | Configuración de la API REST y configuración inicial de acceso a base de datos | |
| Descripción | | Configurar el proyecto API REST con ASP.NET Core, para acceder a la base de datos SQL Server usando Entity Framework Core, y preparar la base para exponer servicios que serán consumidos por la aplicación Blazor Server. |
| Criterios de Aceptación | | * Proyecto API creado (ASP.NET Core Web API). * Conexión a SQL Server configurada (DbContext). * Primera entidad y primer controlador creados. |
| Tareas | | * Crear proyecto API en Visual Studio. * Instalar EF Core y configurar DbContext. * Crear modelo de datos |

|  |  |  |
| --- | --- | --- |
| HT-005 | Diseño e implementación inicial de la base de datos | |
| Descripción | | Definir las tablas principales y relaciones de la base de datos SQL Server, crear el esquema inicial y asegurarse que sea funcional para la aplicación. |
| Criterios de Aceptación | | * Las tablas están correctamente creadas en SQL Server. * Se definen columnas, tipos de datos y relaciones. * Se crean los scripts o migraciones necesarias. * La base de datos es accesible sin errores desde la API. |
| Tareas | | * Diseñar el modelo de datos (entidades y relaciones). * Crear scripts SQL o configuraciones de Entity Framework Core. * Ejecutar migraciones o scripts para construir la base de datos. * Verificar que la API puede acceder a la base de datos correctamente. |

|  |  |
| --- | --- |
| HT-006 | Separación de capas en la arquitectura |
| Descripción | Separar el proyecto en capas: presentación (UI), lógica de negocio y acceso a datos, para mejorar mantenimiento y escalabilidad. |
| Criterios de Aceptación | * Cada capa está en un proyecto distinto. * No hay dependencias cruzadas incorrectas. |
| Tareas | * Crear proyectos independientes en la solución. * Configurar inyección de dependencias. |

|  |  |  |
| --- | --- | --- |
| HT-007 | Configuración de consumo de API desde Blazor Server | |
| Descripción | | Configurar el proyecto Blazor Server para consumir datos de una API externa que accede a la base de datos, usando servicios basados en HttpClient. |
| Criterios de Aceptación | | * Se configura un servicio HttpClient para realizar peticiones a la API. * El proyecto puede consumir datos de prueba de la API (GET, POST). * Se implementa un servicio Blazor que maneja la comunicación con la API. |
| Tareas | | * Configurar HttpClient en Program.cs. * Crear un servicio Blazor que realice solicitudes HTTP (IApiService y ApiService). * Probar que Blazor Server obtiene información de la API. |

|  |  |  |
| --- | --- | --- |
| HT-008 | Creación del primer servicio en Blazor Server para consumir la API | |
| Descripción | | Implementar un servicio en Blazor Server que consuma los datos de la API mediante HttpClient, y lo exponga a los componentes de la aplicación. |
| Criterios de Aceptación | | * Servicio IApiService y ApiService creados. * Los componentes de Blazor pueden obtener datos desde la API. * Manejo básico de errores en llamadas HTTP implementado. |
| Tareas | | * Crear una carpeta Services en Blazor Server. * Crear interfaces y clases de servicio para consumir la API. * Inyectar el servicio en componentes Blazor. |

|  |  |
| --- | --- |
| HT-009 | Integración de ASP.NET Identity en Blazor Server |
| Descripción | Implementar autenticación y autorización basada en roles (administrador, laboratorista, recepcionista) usando ASP.NET Identity con base de datos SQL. |
| Criterios de Aceptación | * Flujo de login y logout funcional. * Usuarios y roles gestionables desde la base de datos. |
| Tareas | * Añadir Identity al proyecto. * Crear migraciones para esquema de usuarios. * Proteger páginas según roles. |

|  |  |
| --- | --- |
| HT-010 | Configuración de CI/CD con GitHub Actions |
| Descripción | Implementar pipeline de integración y despliegue continuo para automatizar compilación, pruebas y publicación. |
| Criterios de Aceptación | * Push a rama principal desencadena despliegue exitoso. * Logs disponibles en GitHub. |
| Tareas | * Probar con cambios reales. |

|  |  |
| --- | --- |
| HT-011 | Despliegue completo en Azure (App + Dominio + Configuración de entorno) |
| Descripción | Como desarrollador quiero desplegar la aplicación Blazor Server en Azure App Service, asociar un dominio personalizado y configurar los parámetros de entorno de forma segura, para tener una versión funcional en producción. |
| Criterios de Aceptación | * La app es accesible desde una URL pública con dominio personalizado. * La configuración sensible (como cadenas de conexión) no está expuesta. * La app corre en modo producción y responde correctamente. |
| Tareas | * Crear recurso Azure App Service para aplicaciones .NET. * Publicar la app desde Visual Studio o GitHub Actions. * Configurar dominio personalizado en App Service. * Configurar certificado SSL y redirección automática a HTTPS. * Crear appsettings.Production.json y almacenar secretos en Azure (Application Settings). * Verificar acceso correcto y comportamiento de la app en modo producción. |

|  |  |
| --- | --- |
| HT-012 | Azure SQL Database y migración de datos |
| Descripción | Como desarrollador quiero crear la base de datos en Azure SQL y migrar los datos del entorno local, para que la aplicación en producción tenga acceso a la información real desde la nube. |
| Criterios de Aceptación | * La base de datos está disponible en Azure SQL. * Los datos locales han sido migrados correctamente. * La API y la app pueden conectarse a la base sin errores. |
| Tareas | * Crear instancia de Azure SQL Database. * Configurar firewall para permitir el acceso seguro. * Migrar estructura y datos desde SQL local usando .bacpac o Data Migration Assistant. * Actualizar la cadena de conexión en la aplicación. * Validar integridad y acceso a los datos desde la app. |

|  |  |
| --- | --- |
| HT-013 | Configuración de backup automático en Azure SQL Database |
| Descripción | Como desarrollador quiero asegurar copias de seguridad automáticas en la base de datos de Azure, para garantizar la recuperación en caso de pérdida de datos. |
| Criterios de Aceptación | * Los backups automáticos están habilitados. * Se ha verificado al menos una restauración de prueba. * Las políticas de retención están configuradas según los requerimientos del proyecto. |
| Tareas | * Revisar la política de retención predeterminada de Azure SQL. * Ajustar la configuración si se requiere mayor tiempo de retención. * Probar una restauración parcial o total de un backup. * Documentar la política de backup implementada. |

|  |  |
| --- | --- |
| HT-014 | Mejora y modificación de la interfaz de usuario en Blazor Server |
| Descripción | Como desarrollador quiero mejorar la interfaz de usuario de los módulos de la aplicación Blazor Server, aplicando estilos consistentes, diseño responsivo, iconografía y tarjetas visuales, para mejorar la experiencia del usuario final. |
| Criterios de Aceptación | * Se implementa diseño responsivo usando Bootstrap 5. * Se incluyen íconos visuales (FontAwesome o Bootstrap Icons) donde sea necesario. * Los formularios y tablas usan tarjetas y estilos consistentes. * Se mejora la estética de al menos tres páginas: órdenes, exámenes y convenios. |
| Tareas | * Revisar cómo se ven actualmente las pantallas de todos los módulos * Organizar los contenidos para que se vean más limpios y ordenados. * Usar cajas o recuadros para separar bien la información (como tarjetas visuales). * Agregar pequeños íconos que ayuden a entender rápidamente qué hace cada botón. * Asegurar que las pantallas se vean bien tanto en computador como en celular. * Comprobar que todo siga funcionando correctamente después de los cambios visuales. |

|  |  |
| --- | --- |
| HT-015 | Verificación interna de funcionalidades antes del despliegue |
| Descripción | Como desarrollador quiero verificar que todas las funcionalidades principales de la aplicación funcionan correctamente en el entorno de pre-producción, para asegurar que el sistema está listo para su despliegue en producción sin errores críticos. |
| Criterios de Aceptación | * Se prueban todos los módulos: login, órdenes, exámenes, pacientes, resultados, convenios y reactivos. * La comunicación entre el frontend (Blazor Server) y la API es fluida y sin errores. * Los datos se registran, consultan y modifican correctamente en la base de datos. * No hay errores visibles en consola del navegador ni en logs del backend. * La interfaz se muestra correctamente en diferentes dispositivos y navegadores. |
| Tareas | * Probar acceso con usuarios de distintos roles (administrador, laboratorista, recepcionista). * Validar operaciones de creación, edición y visualización en todos los módulos. * Revisar funcionamiento de formularios, validaciones, botones e íconos. * Asegurar que no haya errores en red (HTTP 400/500), consola o base de datos. * Comprobar diseño responsivo en PC, móvil y tablet. * Documentar cualquier fallo o comportamiento inesperado y corregir antes del despliegue. |