**Historias Técnicas**

|  |  |  |
| --- | --- | --- |
| 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 2022/2023 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) o herramientas equivalentes. * 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 | Creación de proyecto API y configuración inicial de acceso a base de datos | |
| Descripción | | Crear un proyecto API REST con ASP.NET Core, configurarlo 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. * API responde a consultas básicas. |
| Tareas | | * Crear proyecto API en Visual Studio. * Instalar EF Core y configurar DbContext. * Crear modelo de datos * Crear controlador básico que devuelva datos de prueba. * Probar la API localmente |

|  |  |  |
| --- | --- | --- |
| 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. |