Una caricatura de una persona

Descripción generada automáticamente con confianza baja

**Escuela Superior Politécnica de Chimborazo**

**Facultad de Informática y Electrónica**

**Integrantes:**

Kevin Bedón – 6859

Arleth Cáceres – 6861

Erika Estrada – 7131

Alexis Rivera – 6901



**Semestre:**

Octavo “1”

**Tema:**

Historia Técnicas y Product Backlog

**Carrera:**

Software

**Asignatura:**

Aplicaciones Informáticas II

Noviembre 2023

**Historias Técnicas**

|  |  |
| --- | --- |
| Número: HT01 | |
| Nombre de la historia: Implementación del Diseño Responsive | |
| Prioridad: Alta | **Estimación de esfuerzo:** 100 horas |
| Descripción: Implementar el diseño responsive para garantizar una experiencia de usuario óptima en una variedad de dispositivos y tamaños de pantalla, para asegurar que la aplicación sea accesible y fácil de usar en diferentes entornos. | |
| Criterios de aceptación:   * Utilizar emuladores de dispositivos en navegadores web o herramientas para previsualizar la adaptabilidad en diferentes dispositivos * Adaptarse a diferentes resoluciones de pantalla, asegurando una experiencia visual coherente y agradable. | |
| Tareas:   * Realizar un análisis detallado de los estilos CSS existentes y determinar áreas que necesitan ajustes para lograr un diseño responsive. * Adaptar los estilos existentes utilizando técnicas como media queries, flexbox y/o grid layout para mejorar la adaptabilidad en diferentes dispositivos. * Probar la adaptabilidad en emuladores de dispositivos y navegadores web para verificar la apariencia y funcionalidad en varios tamaños de pantalla. * Realizar ajustes iterativos según sea necesario para mejorar la experiencia de usuario en dispositivos específicos. | |
| Dependencias: Integración del frontend | |

|  |  |
| --- | --- |
| Número: HT02 | |
| Nombre de la historia: Configuración de la Base de Datos | |
| Prioridad: Alta | **Estimación de esfuerzo:** 40horas |
| Descripción: Configurar una base de datos para almacenar y gestionar datos de manera eficiente, para establecer un sólido sistema de almacenamiento que respalde la funcionalidad de la aplicación. | |
| Criterios de aceptación:   * La base de datos debe estar instalada y configurada correctamente en el servidor designado. | |
| Tareas:   * Selección del Sistema de Gestión de Base de Datos * Configuración de la instancia de Base de Datos | |
| Dependencias: Servidor | |

|  |  |
| --- | --- |
| Número: HT03 | |
| Nombre de la historia: Configuración de Strapi como CMS | |
| Prioridad: Alta | **Estimación de esfuerzo:** 60horas |
| Descripción: Configurar Strapi como sistema de gestión de contenido (CMS) para administrar dinámicamente los datos de la aplicación, para establecer una infraestructura robusta que permita la administración eficiente y segura de contenido. | |
| Criterios de aceptación:   * Strapi debe estar instalado y configurado correctamente en el servidor designado. * Los usuarios y roles deben tener permisos configurados adecuadamente. | |
| Tareas:   * Descargar e instalar la última versión estable de Strapi en el servidor designado, asegurando que se esté utilizando la versión más actualizada del CMS. * Realizar la configuración inicial de Strapi, estableciendo la conexión con la base de datos y definiendo las credenciales de administrador. | |
| Dependencias: Motor de Base de Datos | |

|  |  |
| --- | --- |
| Número: HT04 | |
| Nombre de la historia: Modelado de Datos | |
| Prioridad: Alta | **Estimación de esfuerzo:** 40horas |
| Descripción: Definir la estructura y relaciones de la información que se gestionará en la aplicación, para asegurar una gestión eficiente y coherente de la información, lo que facilita el desarrollo de funcionalidades y la adaptabilidad a futuros cambios. | |
| Criterios de aceptación:   * Definir las tablas necesarias para almacenar la información, cada tabla debe tener sus columnas que representen los atributos de los datos. * Establecer las relaciones entre las tablas, cada tabla con sus respectivas claves primarias y foráneas. * Aplicar los principios de normalización, eliminando redundancias y duplicación de datos. | |
| Tareas:   * Identificar Entidades Principales * Configurar Campos y Tipos de Datos * Establecer Relaciones * Validar el Modelado de Datos | |
| Dependencias:   * Motor de gestión de Base de Datos * Strapi | |

|  |  |
| --- | --- |
| Número: HT05 | |
| Nombre de la historia: Configuración del Repositorio del Proyecto | |
| Prioridad: Alta | **Estimación de esfuerzo:** 20horas |
| Descripción: Configuración del repositorio del proyecto, fase crucial para establecer un entorno de desarrollo estructurado y colaborativo, garantizando una gestión eficiente del código fuente y facilitando la colaboración entre los miembros del equipo. | |
| Criterios de aceptación:   * El repositorio debe ser creado con la estructura adecuada para el proyecto. * Se deben establecer las ramas principales y secundarias según las mejores prácticas de desarrollo. * Los colaboradores deben tener los permisos adecuados para acceder y contribuir al repositorio. | |
| Tareas:   * Creación del Repositorio * Estructura del Repositorio * Configuración de Ramas * Gestión de Colaboradores | |
| Dependencias:   * Visual Studio Code * Git | |

|  |  |
| --- | --- |
| Número: HT06 | |
| Nombre de la historia: Implementación de Pruebas Unitarias | |
| Prioridad: Alta | **Estimación de esfuerzo:** 30 horas |
| Descripción: Establecer un conjunto de pruebas unitarias, para garantizar la calidad y el rendimiento de las funciones individuales del sistema. | |
| Criterios de aceptación:   * Seleccionar un marco de pruebas unitarias adecuado para el entorno de desarrollo (por ejemplo, JUnit, NUnit, pytest). * Definir una estrategia de cobertura de pruebas que identifique las funciones y componentes clave que se deben probar. * Desarrollar pruebas unitarias para las funciones y métodos críticos del sistema, asegurando la cobertura definida. | |
| Tareas:   * Definición de Estrategia de Pruebas * Implementación de Pruebas Unitarias Iniciales * Configuración de Reportes de Cobertura | |
| Dependencias:   * Código implementado | |

|  |  |
| --- | --- |
| Número: HT07 | |
| Nombre de la historia: Implementación de Documentación de Código | |
| Prioridad: Baja | **Estimación de esfuerzo:** 30 horas |
| Descripción: Implementar documentación de código, para que los desarrolladores puedan comprender, mantener y colaborar de manera efectiva en el desarrollo del proyecto. | |
| Criterios de aceptación:   * Definir estándares claros para la documentación de código, estableciendo prácticas uniformes en la nomenclatura y estructura de la documentación. * Documentar módulos y funciones clave que son fundamentales para la comprensión del sistema, proporcionando descripciones claras de su propósito y funcionamiento. * Incluir comentarios en el código fuente para explicar secciones específicas, decisiones de diseño o implementaciones particulares. | |
| Tareas:   * Definición de Estándares de Documentación. * Identificación de Módulos y Funciones Clave. * Inclusión de Comentarios en Código | |
| Dependencias:   * Código base | |

|  |  |
| --- | --- |
| Número: HT08 | |
| Nombre de la historia: Implementación de Login Seguro | |
| Prioridad: Alta | **Estimación de esfuerzo:** 20 horas |
| Descripción: Implementar un proceso de inicio de sesión seguro que proteja la autenticación de los usuarios y evite posibles vulnerabilidades. | |
| Criterios de aceptación:   * Implementar un algoritmo de hash seguro para almacenar las contraseñas de los usuarios de forma segura en la base de datos * Implementar un algoritmo de hash seguro para almacenar las contraseñas de los usuarios de forma segura en la base de datos | |
| Tareas:   * Implementación de Hashing de Contraseñas * Configuración de Política de Contraseñas | |
| Dependencias:   * Integración en el BackEnd | |

|  |  |
| --- | --- |
| Número: HT09 | |
| Nombre de la historia: Desarrollo de la Capa de Presentación | |
| Prioridad: Alta | **Estimación de esfuerzo:** 160 horas |
| Descripción: Implementar interfaces de usuario (UI) interactivas y estéticamente agradables que reflejen las funcionalidades ofrecidas por la lógica de negocio. | |
| Criterios de aceptación:   * Desarrollar interfaces de usuario que sean intuitivas y fáciles de usar para mejorar la experiencia del usuario. * Integrar las interfaces de usuario con los servicios backend para permitir la interacción y el intercambio eficiente de datos. | |
| Tareas:   * Codificar la lógica de negocio según los requisitos establecidos para respaldar las funcionalidades de la interfaz de usuario. * Implementar la configuración necesaria en la interfaz de usuario para acceder y gestionar la base de datos desde la capa de datos. * Crear y diseñar interfaces de usuario que reflejen de manera efectiva las funcionalidades de la aplicación y sigan las mejores prácticas de diseño. * Establecer una conexión efectiva entre las interfaces de usuario y los servicios backend para garantizar un intercambio de datos fluido. | |
| Dependencias:   * Integración en el FrontEnd | |

|  |  |
| --- | --- |
| Número: HT10 | |
| Nombre de la historia: Desarrollo de la Capa Lógica de Negocio | |
| Prioridad: Alta | **Estimación de esfuerzo:** 160 horas |
| Descripción: Desarrollar la lógica de negocio de la aplicación, incluyendo procesos como la autenticación, gestión de cursos y evaluaciones. | |
| Criterios de aceptación:   * Implementar un sistema seguro de autenticación de usuarios * Desarrollar la lógica para la creación, modificación y eliminación de cursos * Codificar la lógica relacionada con la creación y evaluación de actividades de aprendizaje. | |
| Tareas:   * Codificar el proceso de autenticación de usuarios, asegurando la seguridad y validez de las credenciales. * Implementar la lógica para la creación, modificación y eliminación de cursos, incluyendo la validación de datos. * Codificar la lógica relacionada con la creación y evaluación de actividades de aprendizaje, asegurando la coherencia y precisión en los resultados. * Desarrollar el mecanismo de comunicación instantánea (chat) entre usuarios, garantizando la entrega y recepción eficientes de mensajes. | |
| Dependencias:   * Integración en el BackEnd | |

|  |  |
| --- | --- |
| Número: HT11 | |
| Nombre de la historia: Desarrollo de la Capa de Acceso a Datos | |
| Prioridad: Alta | **Estimación de esfuerzo:** 160 horas |
| Descripción: Desarrollar la lógica de negocio de la aplicación, incluyendo procesos como la autenticación, gestión de cursos y evaluaciones. | |
| Criterios de aceptación:   * Configurar la conexión segura y eficiente con la base de datos. * Verificae las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para las entidades principales de la aplicación. | |
| Tareas:   * Implementar la configuración para acceder y gestionar la base de datos desde la capa de datos. | |
| Dependencias:   * Integración en el BackEnd | |