

Departamento de Ciencias de la Computación (DCCO)

Carrera de Software

A&D Sw

Perfil del Proyecto

Presentado por: Escobar Rivadeneira Isaac, Mortensen

Franco Eduardo, Ponce Arguello Diego (Grupo 3)

Tutor académico: Ing. Jenny A Ruiz R

Ciudad: Quito

Fecha: 10/05/2025

Índice

Pág.

PERFIL DE PROYECTO

IDEs y Herramientas de Desarrollo	8
1. Visual Studio Code (VS Code)	9
2. Laravel (Framework PHP)	9
3. React (Biblioteca JavaScript para Frontend)	10
4. MySQL/MariaDB (Base de Datos Relacional)	11
5. Herramientas de Control de Versiones (Git, GitHub/GitLab)	11
REQ001: Autenticación de los usuarios	12
REQ002: Visualización de notas	13
REQ003: Registro de Calificaciones	13
REQ004: Memoria Académica	14
REQ005: Registro de Modificaciones	14
REQ006: Control de acceso por perfil	15
8.1 Viabilidad Humana	16
8.2 Viabilidad Tecnológica	17
9.1 Conclusiones	18
9.2 Recomendaciones	18

1. Introducción

En la actualidad, la gestión de las notas en las instituciones educativas enfrenta varios desafíos, especialmente en la Unidad Educativa Mahanaym. A pesar de contar con una plataforma de calificación, no está totalmente adaptada a las constantes actualizaciones de las guías educativas y a las necesidades de personalización de la institución. Esta situación dificulta el seguimiento académico en tiempo real y complica la administración eficiente de las calificaciones. Este documento de especificación de requisitos de software detalla el desarrollo de un sistema de gestión de calificaciones diseñado específicamente para satisfacer las necesidades de la Unidad Educativa de Mahanaim. El objetivo del sistema es permitir el registro, actualización y consulta de las calificaciones, asegurando la integridad de la información y facilitando el acceso a profesores, alumnos y padres de familia. A través de este sistema, buscamos optimizar la organización académica de la institución, adaptándonos a los lineamientos ministeriales y ofreciendo una herramienta segura y flexible para mejorar el proceso de calificación y reporte.

2. Planteamiento del trabajo

2.1 Formulación del problema

La Unidad Educativa Mahanaym actualmente utiliza una plataforma de calificaciones que cumple con los requerimientos establecidos por el Ministerio de Educación. Sin embargo, esta herramienta no permite una adaptación suficiente a los procesos internos, criterios personalizados y objetivos específicos de la institución. Esta situación limita la autonomía en la gestión académica, impide una personalización adecuada del seguimiento estudiantil y restringe el desarrollo de funcionalidades propias que optimicen la experiencia tanto para el personal docente como para los estudiantes y sus familias.

Ante esta necesidad, se plantea el desarrollo de un sistema de gestión de calificaciones propio, diseñado a medida de la realidad institucional de Mahanaym. Este sistema permitirá mayor flexibilidad, control y autonomía en el manejo de calificaciones, manteniendo el cumplimiento de los lineamientos ministeriales, pero al mismo tiempo fortaleciendo los procesos educativos internos con una herramienta más cercana, adaptable y eficiente.

2.2 Justificación

La presente propuesta surge de la necesidad institucional de contar con una herramienta tecnológica que se ajuste específicamente a los procesos, objetivos y estructura organizativa de la Unidad Educativa Mahanaym. Si bien la plataforma actualmente utilizada cumple con los lineamientos establecidos por el Ministerio de Educación, su carácter general y limitado no permite implementar criterios de

evaluación propios, personalizar reportes académicos, ni integrar funcionalidades adicionales que respondan a las dinámicas particulares del plantel.

Desarrollar un sistema de gestión de calificaciones propio permitirá a la institución ejercer un mayor control sobre el manejo académico, optimizar la organización interna, facilitar la toma de decisiones pedagógicas basadas en datos precisos y actualizados, y mejorar la experiencia de los usuarios directos, como docentes, estudiantes y representantes. Además, este sistema contribuirá a una mayor eficiencia en el seguimiento académico y a la generación de reportes adaptados a las necesidades reales de la institución, fortaleciendo así la calidad educativa ofrecida.

En este sentido, el proyecto responde a una necesidad concreta de la Unidad Educativa Mahanaym y representa una inversión estratégica en tecnología educativa alineada con sus prioridades institucionales.

3. Sistema de Objetivos

3.1. Objetivo General

Desarrollar un sistema propio de gestión de calificaciones para la Unidad Educativa Mahanaym, mediante una plataforma web segura y personalizable, que permita el registro, actualización y consulta en tiempo real de los resultados académicos de los estudiantes, garantizando la integridad de la información, el cumplimiento de las directrices ministeriales y facilitando la toma de decisiones pedagógicas.

3.2. Objetivos Específicos (03)

3.2.1. Implementar un módulo de autenticación y control de acceso personalizado:

Implantar el módulo de autenticación y control de acceso personalizado mediante un sistema de roles y permisos granulares configurables en la plataforma web para garantizar la seguridad de los datos académicos y restringir funcionalmente el acceso de docentes, estudiantes, padres y administradores según sus necesidades.

Paquetes de trabajo:

- Análisis de roles y permisos (tutor académico, profesor, estudiante, padre de familia).
- Diseño de la base de datos de usuarios y perfiles.
- Desarrollo backend (API de autenticación, hashing, gestión de sesiones).
- Desarrollo frontend (formularios de login, panel de administración de usuarios).

3.2.2. Desarrollar el módulo de registro y gestión de calificaciones con parámetros personalizables:

Desarrollar el repositorio de registro y gestión de calificaciones con parámetros de evaluación personalizables empleando formularios dinámicos y lógica de negocio configurable en el backend para facilitar la entrada, modificación y cálculo automático de notas conforme a las políticas internas y ministeriales.

Paquetes de trabajo:

- Definición de tipologías de evaluación y configuración de porcentajes.
- Modelado de datos académicos (entidades: estudiante, asignatura, evaluación).
- Implementación de interfaces para ingreso y edición de notas.

3.2.3. Generar un sistema de reportes académicos en tiempo real:

Generar el sistema de informes académicos en tiempo real utilizando librerías de exportación a PDF y Excel integradas en el frontend para proporcionar a docentes, directivos y familias reportes claros, actualizados y descargables que apoyen la toma de decisiones pedagógicas y administrativas.

Paquetes de trabajo:

- Especificar campos obligatorios, encabezados, pies de página y estilos corporativos.
- Crear endpoints en el backend para consulta de calificaciones y métricas académicas..
- Optimizar consultas SQL o Eloquent para agrupar y filtrar datos según curso, periodo y rol (docente, familia, directivo).

4. Alcance

En este proyecto, el sistema propio de gestión de calificaciones de la Unidad Educativa Mahanaym cubrirá de manera integral las siguientes funcionalidades:

4.1.1. Gestión de usuarios y roles

- Registro y mantenimiento de perfiles (tutor académico, docente, estudiante, representante).
- Control de acceso granular según permisos definidos (creación, edición, consulta).

4.1.2. Registro y actualización de calificaciones

- Formulario dinámico para ingreso masivo o individual de notas.
- Historial de modificaciones con sello temporal y usuario responsable.

4.1.3. Consulta en tiempo real

- Panel de seguimiento para docentes, alumnos y padres, con vistas filtrables por curso, materia y período.
- Búsqueda avanzada (por estudiante, por asignatura).

4.1.4. Interfaz administrativa

- Panel de administración para parametrización global (periodos lectivos, asignaturas, porcentajes).

4.2. Límites del proyecto

- No incluye integración automática con otras plataformas externas (p. ej., sistemas bancarios o de pago).
- El desarrollo de módulos como Inventario y Aula Virtual se considera alcance futuro; esta especificación solo define la base de datos y servicios que podrán reutilizarse en esos módulos.
- No contempla funcionalidad offline; el acceso está diseñado para operar en línea sobre la red institucional.

5. Marco Teórico

En este proyecto, el desarrollo del sistema de gestión de calificaciones para la Unidad Educativa Mahanaym se llevará a cabo utilizando una metodología ágil, basada en un enfoque de trabajo iterativo y flexible que permite la adaptación a los requerimientos del cliente (Pressman, 2014). Para facilitar este desarrollo, se emplearán diversas herramientas de software y entornos de desarrollo (IDEs) que permitirán a los miembros del equipo optimizar su tiempo y rendimiento, asegurar la calidad del producto y cumplir con los plazos establecidos. A continuación, se explican las herramientas y tecnologías clave que se utilizarán a lo largo de este proceso.

IDEs y Herramientas de Desarrollo

En este proyecto se emplearán entornos de desarrollo integrados (IDEs) y herramientas específicas que facilitarán tanto la programación como el diseño y la implementación del sistema. La elección de herramientas adecuadas es crucial para garantizar la eficiencia y la calidad del software, tal como señala Sommerville (2011).

1. Visual Studio Code (VS Code)

Visual Studio Code es el IDE principal que se empleará para el desarrollo del sistema. Esta herramienta es ligera pero poderosa, y se adapta a una amplia variedad de lenguajes de programación, incluidos JavaScript, PHP y HTML/CSS, que son los lenguajes clave utilizados en este proyecto.

Según Microsoft (2023), VS Code se destaca por su **versatilidad**, **facilidad de uso** y **potente integración con Git**, lo que permite una colaboración eficiente entre desarrolladores.

- Razón de elección:
 - Versatilidad: Soporta una gran cantidad de extensiones que permiten agregar funcionalidades específicas para los lenguajes utilizados en el proyecto, como el soporte para PHP, JavaScript y React.
 - Facilidad de uso: Su interfaz limpia y modular permite una rápida adaptación por parte de los desarrolladores, aumentando la productividad.
 - Integración con Git: VS Code permite una integración directa con Git, lo que facilita la colaboración entre los miembros del equipo y la gestión del control de versiones.

2. Laravel (Framework PHP)

Laravel es un framework backend basado en PHP que proporciona un conjunto robusto de herramientas para el desarrollo de aplicaciones web seguras y escalables (Stauffer, 2019).

Entre sus ventajas destacan su arquitectura MVC, su sistema integrado de autenticación y control de acceso y su ORM Eloquent, que facilita la interacción con bases de datos relacionales como MySQL/MariaDB.

- Razón de elección:
 - Arquitectura MVC: Laravel sigue el patrón de diseño Modelo-Vista-Controlador (MVC), lo que facilita la separación de la lógica de negocio, la interfaz de usuario y el acceso a los datos.
 - Seguridad: Ofrece características de seguridad como la protección contra CSRF, XSS y SQL Injection, lo que garantiza que el sistema sea seguro desde el punto de vista de la gestión de calificaciones sensibles.
 - Eloquent ORM: Laravel viene con un ORM (Object-Relational Mapper) llamado Eloquent, que facilita la interacción con bases de datos, permitiendo una integración fluida entre el sistema y el modelo de datos académico.
 - Autenticación y roles: Laravel tiene un sistema de autenticación integrado, lo que permite implementar el control de acceso de usuarios (docentes, estudiantes, administradores) de forma rápida y segura.

3. React (Biblioteca JavaScript para Frontend)

React, desarrollado inicialmente por Facebook, es una biblioteca que permite construir interfaces de usuario dinámicas y reactivas mediante un sistema de componentes reutilizables (Griffiths, 2020).

Su uso en este proyecto permitirá mejorar la eficiencia en la actualización de datos gracias a su virtual DOM y fomentar una experiencia de usuario fluida.

- Razón de elección:

- Desempeño: Gracias al Virtual DOM, React es altamente eficiente en la actualización de las interfaces de usuario, lo que garantiza una experiencia fluida para los usuarios al interactuar con las calificaciones y los reportes académicos.
- Reutilización de componentes: La estructura basada en componentes permite a los desarrolladores crear módulos independientes que pueden ser reutilizados en diferentes partes del sistema, lo que reduce el tiempo de desarrollo y mejora la mantenibilidad.
- Comunidad activa: React tiene una comunidad muy activa que constantemente desarrolla nuevas herramientas y recursos que pueden ser utilizados para mejorar el proyecto.

4. MySQL/MariaDB (Base de Datos Relacional)

Para la gestión de los datos académicos, se utilizará una base de datos relacional MySQL o MariaDB. Ambas tecnologías ofrecen **estabilidad, rendimiento y escalabilidad**, características esenciales en sistemas de gestión de información educativa (DuBois, 2013).

Además, Laravel posee integración nativa con MySQL, lo cual facilita su conexión al sistema desarrollado.

- Razón de elección:
 - Estabilidad y rendimiento: MySQL/MariaDB son bases de datos probadas y muy utilizadas en entornos de producción, lo que garantiza una alta disponibilidad y fiabilidad para manejar los datos académicos de estudiantes, calificaciones y reportes.
 - Escalabilidad: Estas bases de datos pueden escalar para manejar un número creciente de usuarios y registros sin sacrificar el rendimiento.
 - Integración con Laravel: La integración con Laravel es directa y sencilla, ya que Laravel viene con soporte nativo para MySQL/MariaDB, lo que facilita la conexión entre el sistema y la base de datos.

5. Herramientas de Control de Versiones (Git, GitHub/GitLab)

Git es un sistema de control de versiones distribuido ampliamente adoptado en proyectos de software colaborativo (Chacon & Straub, 2014).

GitHub y GitLab serán las plataformas utilizadas para alojar el código fuente, permitiendo trabajar en ramas independientes, realizar revisiones de código y mantener un historial completo de cambios.

- Razón de elección:
 - Colaboración eficiente: Git permite que varios desarrolladores trabajen en paralelo sin interferir entre sí, facilitando la integración de cambios y mejoras al código del proyecto.
 - Control de versiones: Git mantiene un historial completo de los cambios realizados en el proyecto, lo que permite revertir cualquier error y mantener un registro detallado del proceso de desarrollo.

5.1 Metodología (Marco de trabajo 5W+2H)

La metodología 5W+2H es una estrategia de planificación que permite clarificar la estructura de un proyecto respondiendo a siete preguntas fundamentales: **qué, cómo, quién, cuándo, dónde, por qué y cuánto** (Kumar, 2005). En este caso, el marco de trabajo 5W+2H se aplicará al desarrollo del **sistema de gestión de calificaciones** para la Unidad Educativa Mahanaym. A continuación, se describe cómo se aplica este marco a través de una **matriz** y se explica cada una de las preguntas que conforman las 5W+2H utilizando la herramienta de Excel.

Esta estructuración detallada a nivel de tareas facilita la trazabilidad, el seguimiento y la verificación del cumplimiento de los objetivos de cada iteración, alineándose con las buenas prácticas de la gestión ágil de proyectos (Serrador & Pinto, 2015).

REQ001: Autenticación de los usuarios

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe permitir la identificación de los usuarios	Se ingresa nombre de usuario y contraseña para	Programador responsable: Isaac	Durante la fase inicial del desarrollo (Entrega: 2025-03-15)	Para asegurar que solo usuarios autorizados

mediante autenticación.	validar el acceso.			accedan a las funcionalidades del sistema.
-------------------------	--------------------	--	--	--

Tabla 1 Marco de trabajo 5W+2H (REQ001)

Se desarrolló una función de login donde los usuarios generales pueden ingresar sus credenciales. Se validan los datos contra la base registrada. Si los datos son incorrectos, el sistema solicita el reintento o el registro previo. Esto garantiza la seguridad del acceso y evita vulnerabilidades.

REQ002: Visualización de notas

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe registrar calificaciones diferenciadas por tipo de evaluación.	Se habilitarán campos específicos para ingresar el tipo de evaluación, nota, fecha y observaciones.	Programador responsable: Isaac	Fase de desarrollo medio (Entrega: 2025-03-15)	Para permitir el control detallado y estructurado de diferentes tipos de evaluación en el sistema.

Tabla 2 Marco de trabajo 5W+2H (REQ002)

Se diseñaron formularios dinámicos donde el administrador o docente puede registrar evaluaciones específicas (examen, tarea, lección) junto con sus respectivas notas. Este registro alimenta una base de datos ordenada que facilita el análisis posterior.

REQ003: Registro de Calificaciones

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe permitir el registro de calificaciones diferenciadas y ponderadas.	Seleccionando el tipo de evaluación y asignando notas que luego se ponderan automáticamente.	Programador responsable: Eduardo	Durante la mitad del ciclo de desarrollo (Entrega: 2025-03-15)	Para que los docentes puedan ingresar las notas considerando las ponderaciones de cada

				evaluación y evitar errores manuales.
--	--	--	--	---------------------------------------

Tabla 3 Marco de trabajo 5W+2H ((REQ003)

Se creó una funcionalidad donde el docente selecciona el estudiante, el tipo de evaluación, asigna la nota, y el sistema calcula automáticamente el porcentaje aplicable. Esto evita la manipulación manual de cálculos y reduce errores.

REQ004: Memoria Académica

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe generar boletines consolidados por materia, curso y estudiante.	Se recogerán los datos de las evaluaciones registradas y se generarán boletines e informes en PDF o similar.	Programador responsable: Eduardo	Fase de consolidación del sistema (Entrega: 2025-03-15)	Para brindar acceso claro y resumido del desempeño académico por trimestre a directivos, docentes y padres.

Tabla 4 Marco de trabajo 5W+2H (REQ004)

A partir de los registros de notas, el sistema crea informes automáticos con promedios, detalles de cada materia y observaciones. Estos informes pueden ser descargados y enviados a los representantes para su seguimiento.

REQ005: Registro de Modificaciones

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe almacenar un historial de cambios de calificaciones.	Cada modificación de nota registrará fecha, hora, usuario responsable y descripción del cambio.	Programador responsable: Diego	Etapa avanzada del desarrollo (Entrega: 2025-03-15)	Para garantizar la trazabilidad, transparencia y seguridad en el proceso de gestión de calificaciones.

Tabla 5 Marco de trabajo 5W+2H (REQ005)

Se implementó un sistema de auditoría interna donde todo cambio queda registrado. Esto permite rastrear modificaciones y asegura la integridad de la información, vital en procesos educativos sensibles.

REQ006: Control de acceso por perfil

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?
El sistema debe restringir el acceso a funcionalidades según el rol del usuario (docente, estudiante, administrador, representante).	Se implementará control de permisos y roles asignados automáticamente tras el login.	Programador responsable: Diego	Durante la configuración de seguridad (Entrega: 2025-03-15)	Para limitar el acceso a información y acciones sensibles, aumentando la seguridad del sistema.

Tabla 6 Marco de trabajo 5W+2H (REQ006)

Los perfiles serán gestionados desde un módulo de administración donde cada usuario será etiquetado bajo un rol específico. Cada rol tendrá restricciones y accesos definidos para garantizar que la información confidencial no sea vista ni manipulada por usuarios no autorizados.

6. Ideas a Defender

En este proyecto se defienden cuatro ideas centrales que combinan los principios de Ingeniería de Software y de Programación para la Unidad Educativa Mahanaym: la adopción de una arquitectura modular en capas (presentación, lógica de negocio y datos) que facilite la mantenibilidad y escalabilidad del sistema; la parametrización dinámica de la configuración (tipos de evaluación, porcentajes y escalas) a través de fuentes externas (archivos o tablas de base de datos) para que el personal administrativo pueda ajustar reglas sin modificar código; un enfoque en la experiencia de usuario, validado mediante prototipos interactivos y recorridos de interfaz, que garantice flujos intuitivos para docentes, estudiantes y familias; y un esquema de seguridad y auditoría robusto

7. Resultados Esperados

Al finalizar el proyecto se dispondrá de una plataforma web operativa para gestionar calificaciones con control de acceso por roles, parámetros de evaluación; un código modular y documentado que facilite su mantenimiento y que garanticen la rápida adopción y configuración autónoma del sistema por parte del personal de la Unidad Educativa Mahanaym.

8. Viabilidad(Ej.)

Cantida d	Descripció n	Valor Unitari o (USD)	Valor Total (USD)
	Equipo en casa		
1	Laptop HP I7 RAM 12 GB 512 GB SD (o laptop propia o computadora de escritorio)	\$849.99	\$849.99
	Software		
1	Sistema operativo Windows 10/11 y Linux	0	0
1	Visual Studio Code	0	0
1	PHPmyadmin	0	0
	MongoDB	0	0
1	React	0	0
		TOTAL	\$849.99

Tabla 2 Presupuesto del proyecto

8.1 Viabilidad Humana

8.1.1 Tutor Empresarial

- **Persona:** Mgtr. Ing. Alfredo Escobar (Rector)
- **Responsabilidades:**
 - Validar requisitos de la institución
 - Facilitar el acceso a infraestructura y datos académicos.
 - Supervisar la alineación del proyecto con los objetivos institucionales.

8.1.2 Tutor Académico

- **Persona:** Ing. Jenny Ruiz (Docente líder del proyecto)

Responsabilidades:

- Asesorar en criterios pedagógicos y formatos de evaluación.
- Revisar y aprobar los diseños de reportes académicos.

8.1.3. Estudiantes

- **Responsabilidades:**

- Participar en sesiones de prueba del sistema (usabilidad y funcionalidad).
- Proporcionar retroalimentación sobre la claridad de reportes y accesibilidad.
- Validar la precisión de la consulta de sus calificaciones.

Viabilidad humana: Los roles están claramente definidos dentro de la organización, con disponibilidades horarias compatibles y sin requerir contrataciones externas, lo que garantiza un apoyo continuo y eficiente durante todo el ciclo de desarrollo.

8.2 Viabilidad Tecnológica

8.2.1 Hardware

	Requisitos mínimos	Disponibilidad
Memoria RAM	12 GB de RAM	Alta
Almacenamiento	10 GB de espacio de almacenamiento	Alta

Tabla 3 Requisitos de Hardware

8.2.2 Software

	Requisitos mínimos	Disponibilidad
Sistema Operativo	Windows 10/11 y Linux (dual-boot o máquinas virtuales).	Alta
IDE	Es recomendable Visual Studio Code sin embargo, cualquier IDE con esta funcionalidad funciona.	Alta
BASE DE DATOS	Es recomendable MySQL/MariaDB (phpMyAdmin) y MongoDB (para esquemas flexibles si se requiere).	Alta
FRAMEWORKS Y LIBRERÍAS	Es recomendable Laravel (o similar), React, jsPDF, SheetJS, Chart.js.	Alta

Tabla 4 Requisitos de Software

9. Conclusiones y recomendaciones

Este es uno de los capítulos fundamentales del documento. En él se trata en primer lugar de hacer una recapitulación del trabajo y un juicio crítico del mismo, tome en cuenta el cumplimiento de los objetivos mencionados anteriormente

9.1 Conclusiones

9.2 Recomendaciones

.

10. Planificación para el Cronograma:

Nombre de tarea	Duración	Comienzo	Fin	Nombres de los involucrados
Elaboración introducción	2 días	04/11/2024	06/11/2024	Isaac Escobar
Plantamiento del problema	1 día	05/11/2024	05/11/2024	Isaac Escobar
Objetivos	1 día	07/11/2024	07/11/2024	Isaac Escobar
Alcance	1 día	10/11/2024	10/11/2024	Isaac Escobar
Planificación del proyecto de fin de curso	n/a	n/a	n/a	n/a
Revisión del SRS	1 día	29/04/2025	01/05/2025	Isaac Escobar
Generación de la matriz	4 días	29/04/2025	06/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Creación del Foda	1 día	06/05/2025	06/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Entrevista del cliente	1 hora	11/05/2025	11/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Etapas de Requisitos				
Reunión con los expertos del tema asignado	0.5 día	09/05/2025	09/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Revisión del Perfil de proyecto	2 días	10/05/2025	11/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Elaboración de presentación para defensa del perfil del proyecto	2 días	12/05/2025	13/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Defensa del perfil del proyecto	1 día	13/05/2025	13/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Etapas de Analisis				
Sprint 0				
Definir reglas del Proyecto (Reunión)				
Matriz HU	3 horas	12/05/2025	12/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Historias de Usuario	1 hora	11/05/2025	11/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Perfilado del proyecto	1 día	10/05/2025	11/05/2025	Isaac Escobar, Eduardo Mortensen, Diego Ponce
Preguntas Entrevista (Entrevista)	30 minutos	11/05/2025	11/05/2025	Diego Ponce, Isaac Escobar, Eduardo Mortensen

Tabla 5 Cronograma del proyecto.

11. Referencias

- Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
<https://git-scm.com/book/en/v2>
- DuBois, P. (2013). *MySQL*. Addison-Wesley Professional.
- Griffiths, D. (2020). *React Cookbook: Recipes for Mastering the React Framework*. O'Reilly Media.
- Kumar, S. (2005). *Research Methodology: A Step-by-Step Guide for Beginners*. SAGE Publications.
- Microsoft. (2023). *Visual Studio Code Documentation*.
<https://code.visualstudio.com/docs>
- Pressman, R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040-1051.
<https://doi.org/10.1016/j.ijproman.2015.01.006>
- Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.

Anexos.

Anexo I. Crono

https://uespe-my.sharepoint.com/:x:/g/personal/iaescobar3_espe_edu_ec/EfqDvg5TsYBLoe-vL0WqSVkBQbsha_GS6dUTkD-nP8DNHg?e=C24KUH

Anexo II. MTZ de Historias de Usuarios

https://uespe-my.sharepoint.com/:x:/g/personal/iaescobar3_espe_edu_ec/EYJtTOX2z99FictT5z8Nz2EBmHGBgM7EaTkZKNtaaJZU-A?e=a7xsnY