

Plan de Proyecto

“LevelUp”

Profesor: Patricio Soto

Integrantes:

- Antonia Rojas
- Millaray Ibarra
- Millaray Currihual

Índice

Índice.....	2
Problema.....	3
Solución.....	4
Beneficiarios.....	5
Alcance del Proyecto.....	6
Problema, solución propuesta y contexto del proyecto.....	6
Funciones esperadas del software a desarrollar.....	6
Rendimiento esperado del software a desarrollar.....	6
Restricciones del proyecto.....	6
Metodología.....	7
Requerimientos.....	8
Diagrama de Clases.....	10
Diagrama de Base de Datos.....	11
Casos de Usos.....	12
1. Actor: Estudiante.....	12
2. Actor: Docente.....	12
3. Actor: Administrador.....	13
4. Casos de uso comunes.....	13
Diagrama de Casos de Uso General.....	14
Diagrama de Casos de Uso Estudiante.....	15
Diagrama de Casos de Uso Administrador.....	17
Definición de roles y responsabilidades.....	18
Tecnologías.....	19
Arquitectura.....	20
Estructura de Desglose de trabajo.....	21
Definición de Diagrama Matriz EDT.xlsx.....	21
Diccionario EDT.....	21
Costo por Fase y por Rol.....	22
Cronograma.....	23
Costos.....	25
Riesgos.....	26
Pruebas.....	28
Lecciones aprendidas.....	29

Problema

Uno de los principales desafíos en la educación básica es la dificultad para reforzar y complementar los aprendizajes adquiridos en el aula, especialmente cuando los estudiantes avanzan a ritmos diferentes y tienen necesidades particulares. En muchas ocasiones, los contenidos entregados en clase no son suficientes para asegurar una comprensión sólida, lo que puede generar brechas de conocimiento y pérdida de motivación.

Este fenómeno afecta principalmente a los estudiantes, quienes requieren espacios adicionales de práctica y recursos dinámicos que refuercen lo aprendido en el colegio. A menudo, los métodos tradicionales no logran ofrecer este refuerzo de manera adecuada. Además, impacta a los docentes, quienes carecen de herramientas eficaces para dar un seguimiento al progreso de sus estudiantes.

La importancia de resolver este problema radica en que disponer de un sistema complementario al proceso escolar permite reforzar los aprendizajes de manera dinámica, contribuyendo a mejorar la comprensión, motivar a los estudiantes y aumentar su participación en el proceso educativo. Al hacerlo, se logra una educación más inclusiva y efectiva, permitiendo a los estudiantes superar las barreras del aprendizaje y alcanzar su máximo potencial.

Solución

La solución propuesta consiste en el desarrollo de “LevelUp”, una aplicación web diseñada para complementar el trabajo realizado en el aula y reforzar los aprendizajes de los estudiantes de educación básica. Su propósito principal es brindar un espacio digital de apoyo donde los contenidos escolares puedan ser repasados y consolidados de forma práctica y dinámica.

La plataforma integrará herramientas interactivas, actividades lúdicas y sistemas de seguimiento académico, que permitirán a los estudiantes reforzar los contenidos vistos en clases, mientras docentes podrán monitorear el progreso y apoyar en las áreas que presenten mayor dificultad.

Para fomentar la motivación, se implementará un sistema de recompensas gamificado, en el que los estudiantes podrán realizar actividades, conseguir logros al completar tareas, subir de rango en función de su progreso. Además, contarán con un ranking o tabla de clasificación que promueva la competencia saludable, junto con premios virtuales y reconocimientos por sus logros durante el proceso de aprendizaje.

Por otra parte, los docentes dispondrán de funcionalidades específicas, como herramientas para crear actividades interactivas, la capacidad de asignar actividades, así como la visibilidad de ranking globales del rendimiento de los estudiantes, que facilitarán la toma de decisiones pedagógicas y la mejora continua.

De esta manera, LevelUp no reemplaza al sistema escolar, sino que lo complementa, ofreciendo una experiencia inclusiva, motivadora y efectiva. Con ello, se espera mejorar la comprensión y el rendimiento académico, incrementar la motivación y confianza de los estudiantes, y aprovechar la tecnología como apoyo al proceso educativo.

Beneficiarios

Este problema afecta principalmente a niños y niñas de educación básica, quienes se encuentran en una etapa formativa clave para el desarrollo de sus habilidades y hábitos de aprendizaje. En general, son estudiantes que muestran curiosidad natural y disposición para aprender, pero que a menudo enfrentan dificultades de concentración, diferencias en los ritmos de aprendizaje y falta de apoyo adecuado. Entre sus comportamientos más comunes se observan la desmotivación frente a contenidos poco atractivos, la frustración al no comprender un tema al mismo ritmo que sus compañeros y la dependencia del acompañamiento constante de docentes o apoderados. Sus principales frustraciones se relacionan con sentirse rezagados respecto a sus pares, tener limitaciones en el acceso a recursos educativos dinámicos y experimentar clases poco interactivas. Los estudiantes prefieren actividades entretenidas y accesibles que les permitan aprender de forma dinámica, avanzar con mayor motivación y sentirse apoyados en su proceso escolar.

Alcance del Proyecto

Problema, solución propuesta y contexto del proyecto

El proyecto busca reforzar y complementar los aprendizajes adquiridos en la educación básica mediante una aplicación web, que funcione como un apoyo al trabajo realizado en el aula. Esta solución se enmarca en la transformación digital educativa, donde la tecnología permite ofrecer contenidos dinámicos y atractivos que favorecen la comprensión, la motivación y la participación de los estudiantes, contribuyendo a hacer más efectivo el proceso de enseñanza-aprendizaje.

Funciones esperadas del software a desarrollar

- Registro y gestión de usuarios (estudiantes, docentes y administradores).
- Creación de perfiles de aprendizaje
- Acceso a contenidos interactivos y dinámicos.
- Sistema de recompensas y logros
- Herramientas específicas para docentes que permitan diseñar actividades, evaluar progresos y gestionar reconocimientos.

Rendimiento esperado del software a desarrollar

- Carga rápida de contenidos (menos de 3 segundos por acción).
- Escalabilidad para múltiples usuarios simultáneos.
- Disponibilidad mínima del 99%.
- Acceso desde distintos dispositivos (PC, tablets, celulares).
- Seguridad en los datos personales y académicos.

Restricciones del proyecto

- El alcance inicial se limita a una aplicación web
- Dependencia de conexión a internet.
- Se priorizará el uso de tecnologías de código abierto para optimizar costos.
- El proyecto se enfocará exclusivamente en educación básica.

Metodología

La metodología se implementará bajo un enfoque tradicional de cascada, dividiendo el desarrollo del proyecto en fases secuenciales. En la fase inicial, se realizará un análisis exhaustivo del problema de baja retención estudiantil y del aprendizaje. Esto guiará la planificación y el diseño del sistema, definiendo claramente los objetivos, requerimientos y la arquitectura de la plataforma.

En la fase de diseño, se establecerán las funcionalidades principales. Se utilizarán tecnologías como Django y SQL para el backend, y HTML, CSS y JavaScript para el frontend, con el objetivo de desarrollar una plataforma que complemente el trabajo en el aula y refuerce los aprendizajes de los estudiantes de manera dinámica y motivadora.

Una vez diseñado el sistema, se pasará a la fase de desarrollo, donde se implementarán las funcionalidades definidas. Posteriormente, se llevarán a cabo pruebas unitarias, de integración y de rendimiento para asegurar que la plataforma funcione correctamente y cumpla con los requisitos establecidos.

Finalmente, en la fase de cierre, la plataforma será lanzada y se pondrán en marcha procesos de monitoreo y soporte continuo. Esto permitirá realizar actualizaciones periódicas y mejorar la plataforma de acuerdo con los comentarios de los usuarios, con el objetivo de incrementar la retención y motivación estudiantil a lo largo del tiempo.

Requerimientos

Registro y autenticación de usuarios

- Creación de cuentas para estudiantes, docentes y administradores.
- Inicio de sesión seguro con validación de credenciales.
- Recuperación de contraseña.

Gestión de perfiles

- Edición de información personal y académica.

Sistema de recompensa

- Los estudiantes podrán realizar actividades que les haya asignado su docente.
- Visualizar actividades según asignatura.
- Los estudiantes podrán subir de rango o alcanzar logros según las actividades realizadas.
- La plataforma mostrará un ranking o tabla de clasificación para fomentar la competencia saludable.
- Los estudiantes completarán actividades diarias para reforzar conceptos y mantener su motivación.

Funcionalidades para docentes

- Herramientas para crear actividades interactivas y minijuegos.
- Visualizar su lista de actividades creadas.
- Podrá editar y eliminar actividades.
- Asignar actividades a cursos y/o alumnos.
- Visibilidad de ranking global de los estudiantes.

Funcionalidades para administradores

- Gestión de usuarios, asignaturas, cursos, asignaciones y matrículas.
- Editar perfiles

Interfaz intuitiva

- Diseño adaptable a distintos dispositivos (PC, tablet, móvil).
- Navegación simple y clara para los usuarios.

Seguridad y privacidad

- Gestión de permisos y roles (estudiantes, docentes y administradores).
- Protección de datos personales y académicos.

Escalabilidad y rendimiento

- Soporte para un número creciente de usuarios activos.
- Optimización del tiempo de carga de la plataforma.

Mantenimiento y actualización

- Implementación de mejoras continuas en funcionalidades.
- Corrección de errores y actualizaciones periódicas.

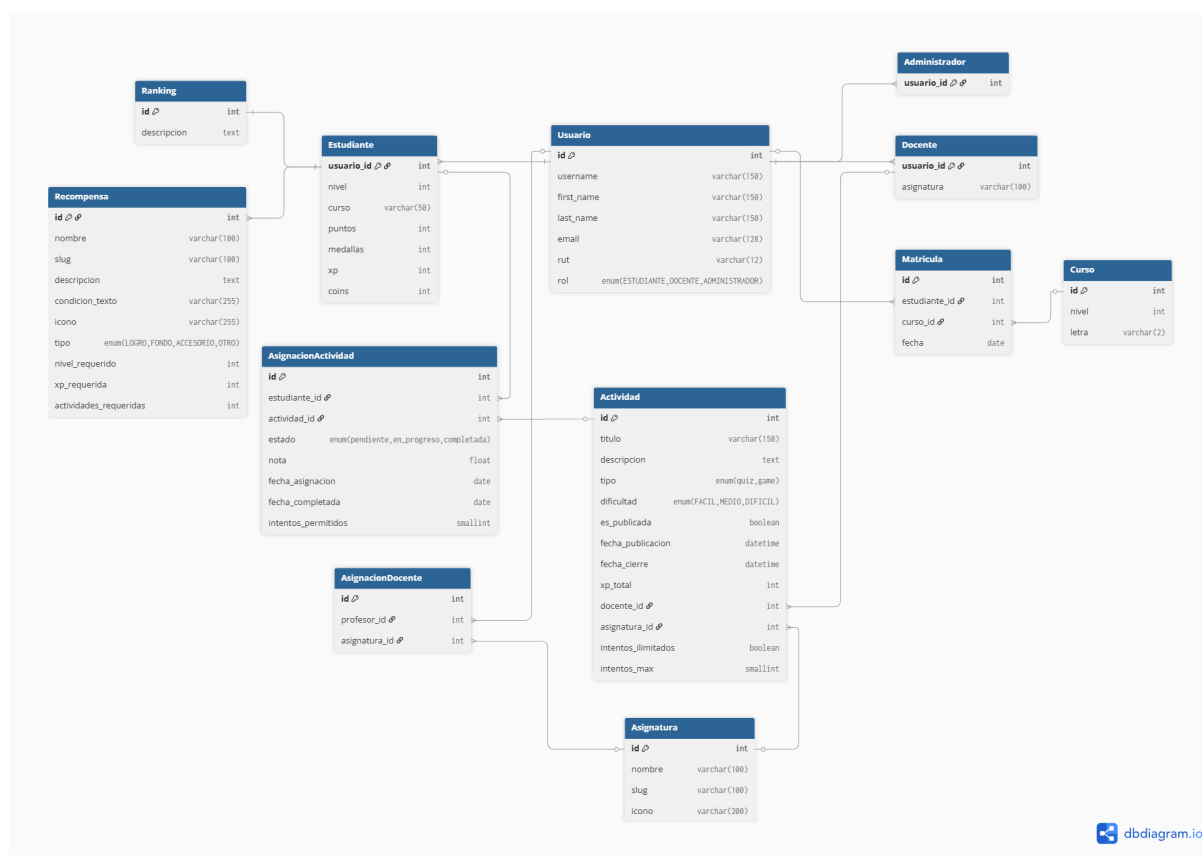
```

classDiagram
    class Usuario {
        +id: int
        +username: varchar(150)
        +first_name: varchar(150)
        +last_name: varchar(150)
        +email: varchar(128)
        +rut: varchar(12)
        +rol: Rol
        +autenticar(email, password): bool
        +esEstudiante(): bool
        +esDocente(): bool
        +esAdmin(): bool
        +getNombreCompleto(): string
    }
    class Administrador {
        +usuario_id: int
        +gestionarUsuarios(): void
        +gestionCurso(): void
        +gestionarAsignaturas(): void
        +gestionarMatriculas(): void
        +gestionarAsignaciones(): void
    }
    class Docente {
        +usuario_id: int
        +asignatura: varchar(100)
        +crearActividad(titulo, tipo): Actividad
        +asignarActividad(a: Actividad, curso: Curso): void
        +editarActividad(a: Actividad): void
        +eliminarActividad(a: Actividad): void
    }
    class AsignacionDocente {
        +id: int
        +profesor_id: int
        +asignatura_id: int
        +asignar(): void
        +remover(): void
    }
    class Asignatura {
        +id: int
        +nombre: varchar(100)
        +slug: varchar(100)
        +icono: varchar(200)
        +obtenerActividades(): List<Actividad>
    }
    class Actividad {
        +id: int
        +titulo: varchar(150)
        +descripcion: text
        +tipo: TipoActividad
        +dificultad: Dificultad
        +es_publicada: boolean
        +fecha_publicacion: datetime
        +fecha_cierre: datetime
        +xp_total: int
        +intentos_ilmitados: boolean
        +intentos_max: smallint
        +publicar(): void
        +cerrar(): void
        +estaAbierta(fecha): bool
        +calcularXPObtenida(est: Estudiante): int
    }
    class AsignacionActividad {
        +id: int
        +estudiante_id: int
        +actividad_id: int
        +estado: EstadoAsignacion
        +nota: float
        +fecha_asignacion: date
        +fecha_completada: date
        +intentos_permitidos: smallint
        +puedeIntentar(est: Estudiante): bool
        +registrarIntento(est: Estudiante, score: float): void
        +marcarCompletada(nota: float): void
    }
    class Ranking {
        +id: int
        +descripcion: text
        +calcularPosiciones(): void
        +obtenerPosicion(est: Estudiante): int
    }
    class Estudiante {
        +usuario_id: int
        +nivel: int
        +curso: varchar(50)
        +puntos: int
        +medallas: int
        +xp: int
        +coins: int
        +verActividadesAsignadas(): List<Actividad>
        +iniciarActividad(a: Actividad): void
        +verResultados(a: Actividad): void
        +obtenerXPTotal(): int
        +verRanking(): Ranking
        +subirRango(): void
    }
    class Recompensa {
        +id: int
        +nombre: varchar(100)
        +slug: varchar(100)
        +descripcion: text
        +condicion_texto: varchar(255)
        +icono: varchar(255)
        +tipo: TipoRecompensa
        +nivel_requerido: int
        +xp_requerida: int
        +actividades_requeridas: int
        +cumpleCondicion(est: Estudiante): bool
    }
    class Curso {
        +id: int
        +nivel: int
        +letra: varchar(2)
        +obtenerEstudiantes(): List<Estudiante>
    }
    class Matricula {
        +id: int
        +estudiante_id: int
        +curso_id: int
        +fecha: date
        +registrari(): void
        +cambiarCurso(nuevo: Curso): void
    }

    Usuario <|-- Administrador
    Usuario <|-- Docente
    Usuario <|-- Estudiante
    Administrador --> AsignacionDocente : crea
    Docente --> AsignacionDocente : crea
    AsignacionDocente -- "0..*" --> "0..*" Asignatura
    AsignacionDocente -- "0..*" --> "0..*" Actividad
    Asignatura -- "0..*" --> "0..*" Actividad
    Asignatura -- "1" --> "0..*" AsignacionActividad
    Actividad -- "1" --> "0..*" AsignacionActividad
    Estudiante -- "1" --> "0..*" AsignacionActividad
    Estudiante -- "1" --> "0..*" Ranking
    Estudiante -- "1" --> "0..*" Recompensa
    Estudiante -- "1" --> "0..*" Matricula
    Estudiante -- "1" --> "0..*" Curso
    Ranking -- "1" --> "0..*" Estudiante : incluye
    Recompensa -- "0..*" --> "0..*" AsignacionActividad
    Matricula -- "0..*" --> "0..*" AsignacionActividad

```

Diagrama de Base de Datos



Casos de Usos

El diagrama representa los principales casos de uso del sistema LevelUP, organizados según los tres actores que interactúan con la plataforma: Estudiante, Docente y Administrador. Cada actor tiene acceso a funcionalidades específicas, aunque existen acciones comunes como la autenticación y la gestión básica del perfil.

1. Actor: Estudiante

El estudiante puede autenticarse en el sistema e ingresar a su portal personal. Dentro de este entorno, tiene acceso a funcionalidades orientadas a la visualización de información y a la gestión de su perfil. Entre las acciones principales se encuentran:

- Iniciar sesión y cerrar sesión
- Acceder al Portal Estudiante
- Cambiar contraseña
- Ver actividades disponibles
- Visualizar su ranking, recompensas obtenidas y rangos alcanzados.

Estas funciones permiten al estudiante realizar un seguimiento de su progreso dentro de la plataforma.

2. Actor: Docente

El docente también puede ingresar a la plataforma mediante autenticación y cuenta con un conjunto de funcionalidades orientadas a la gestión académica. Entre sus acciones se encuentran:

- Acceder al Portal Docente.
- Gestionar actividades, lo que incluye su creación, edición o actualización.
- Visualizar ranking de los estudiantes.

Además, comparte con los demás actores la capacidad de iniciar, cerrar sesión y cambiar contraseña.

3. Actor: Administrador

El administrador posee el nivel más amplio de acceso dentro del sistema, permitiéndole gestionar la estructura operativa de la plataforma. Sus funciones incluyen:

- Acceder al Portal Administrador.
- Gestión de usuarios, asignaturas, cursos, asignaciones y matrículas.
- Editar perfiles
- Acciones comunes como iniciar sesión, cerrar sesión y cambiar contraseña.

Su rol está enfocado en el mantenimiento general del sistema y en la administración de todos los elementos que componen la plataforma educativa.

4. Casos de uso comunes

Todos los actores comparten ciertos casos de uso fundamentales para la interacción con el sistema:

- Iniciar sesión
- Cerrar sesión
- Cambiar contraseña

Diagrama de Casos de Uso General

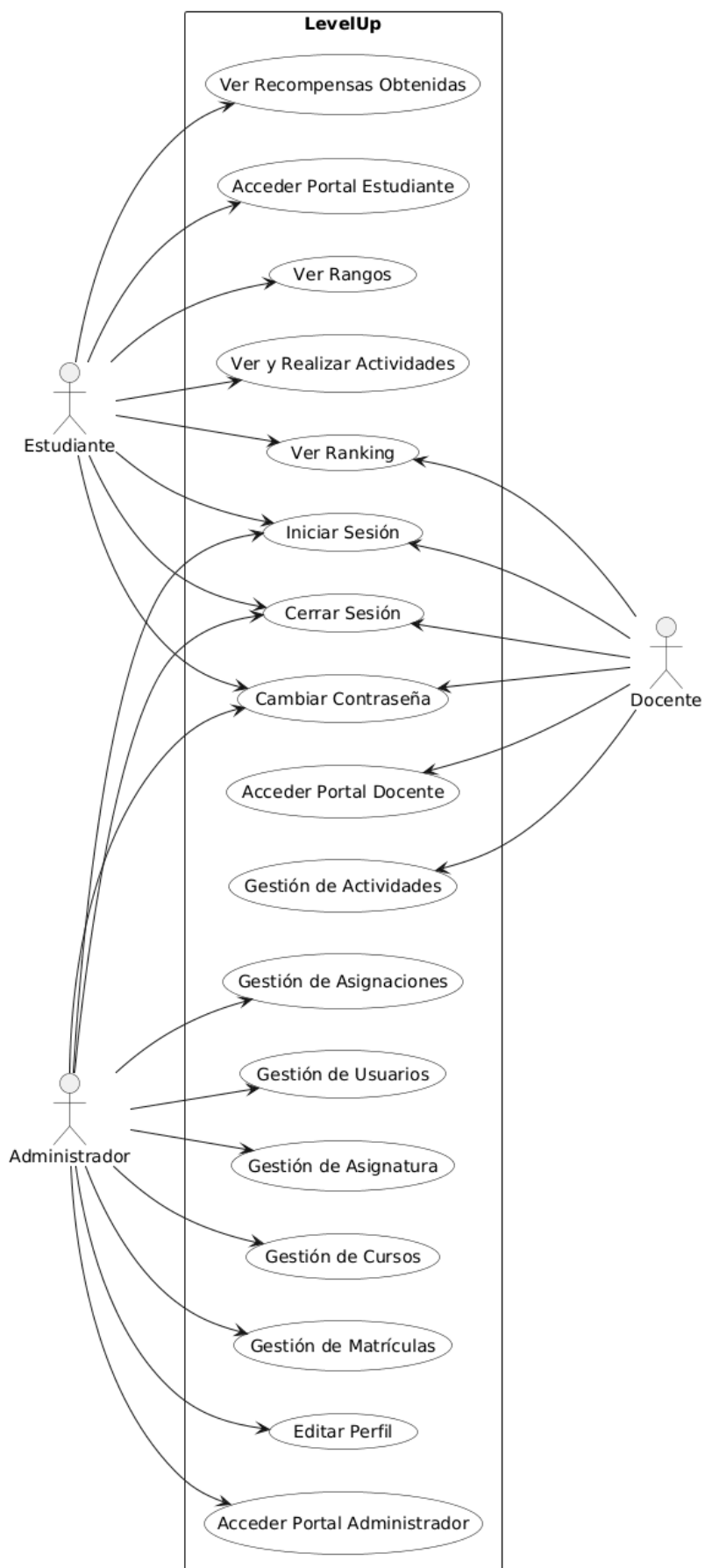


Diagrama de Casos de Uso Estudiante

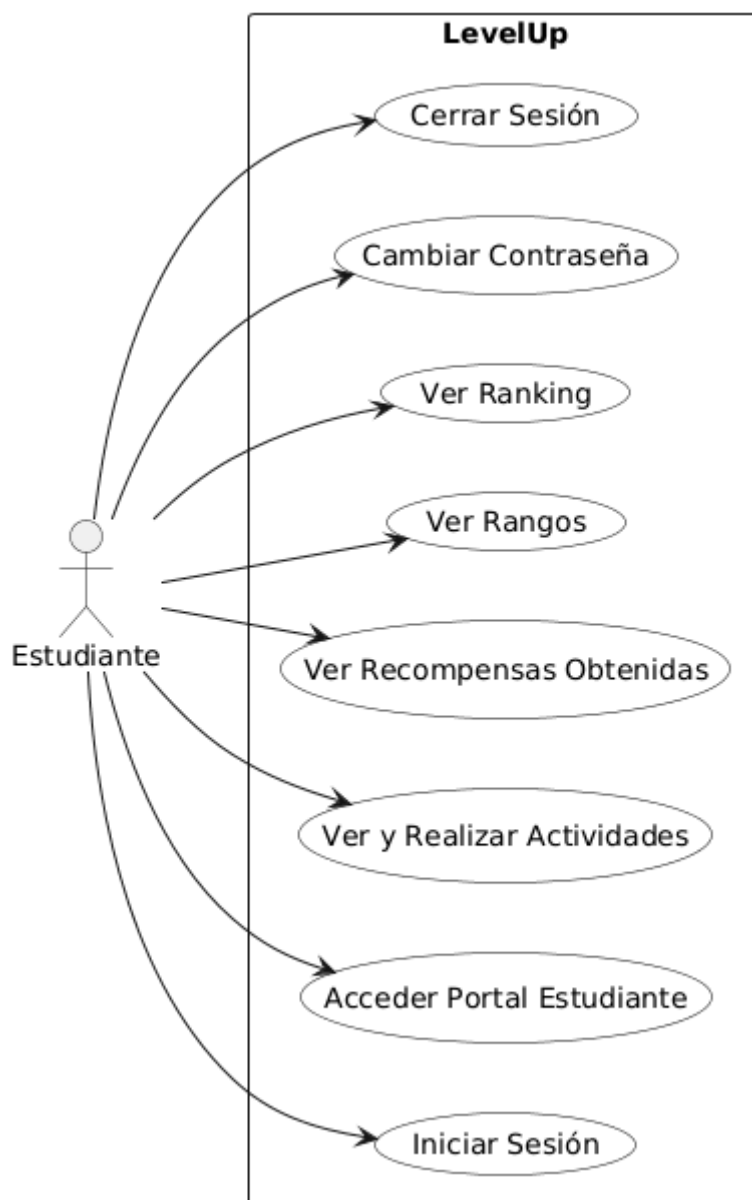


Diagrama de Casos de Uso Docente

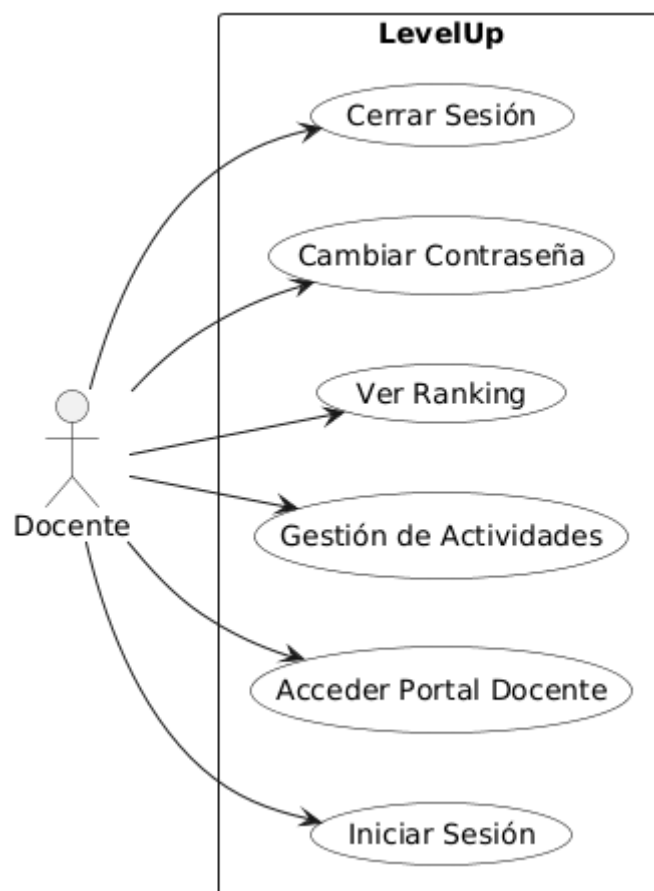
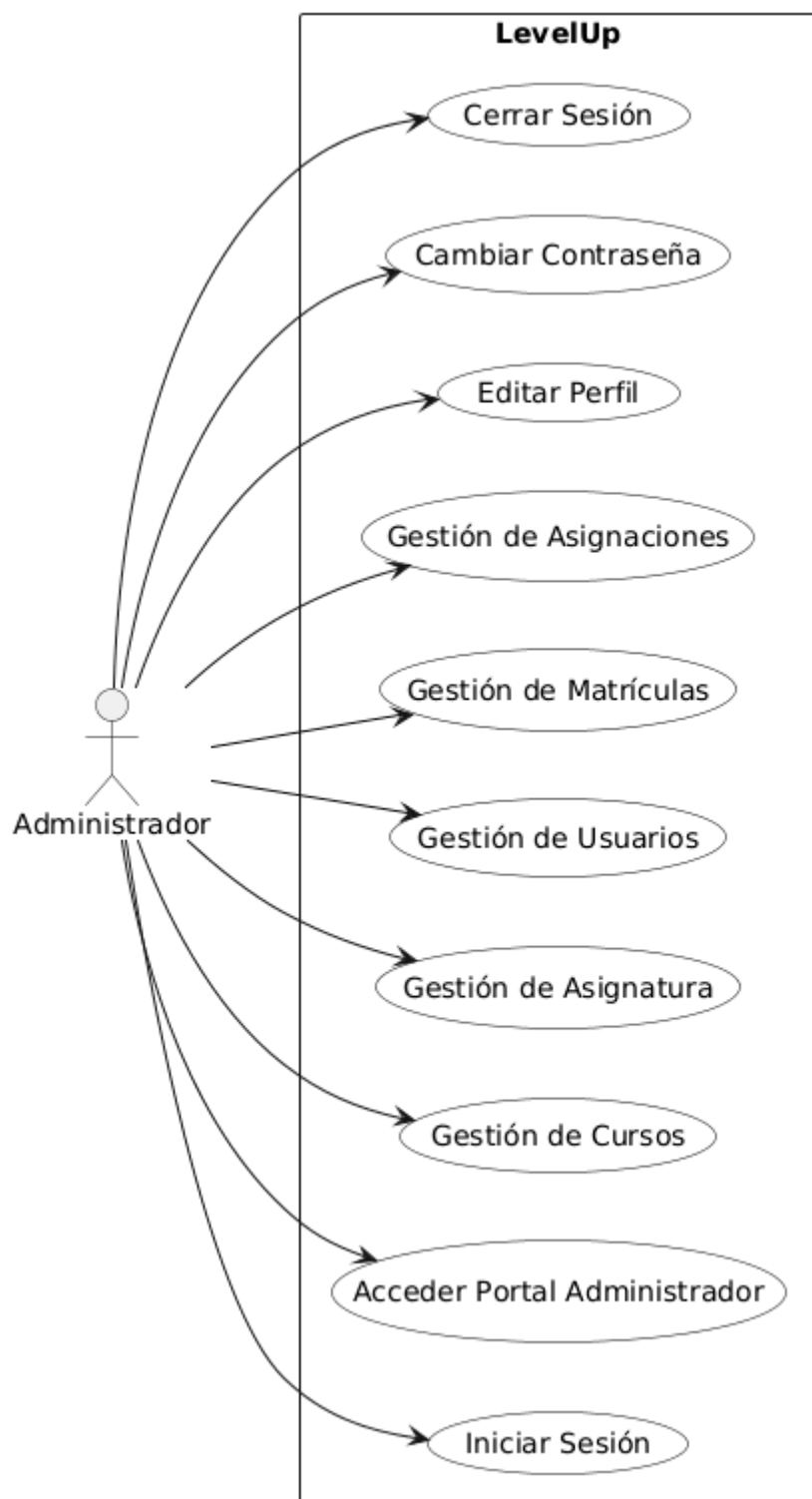


Diagrama de Casos de Uso Administrador



Definición de roles y responsabilidades

Rol	Responsabilidades
Jefe de Proyecto	Responsable de la planificación, coordinación general y cumplimiento de objetivos.
Analista Funcional	Levantar, analizar y documentar los requerimientos del sistema. Asegurar que la solución cumpla con las necesidades del negocio. Validar funcionalidades y mantener trazabilidad entre requerimientos y resultados del desarrollo.
Desarrollador	Desarrollar la lógica del sistema y la base de datos. Implementar la interfaz gráfica y conectarla con el backend. Garantizar el buen funcionamiento, seguridad y compatibilidad en distintos dispositivos.
Diseñador de UX/UI	Diseñar la interfaz gráfica de la plataforma con criterios de usabilidad y accesibilidad. Crear prototipos y wireframes de la solución.
Tester/QA	Diseñar y ejecutar pruebas del sistema. Detectar errores y validar que el software cumpla los requerimientos definidos.

Matriz RACI [x Matriz de Responsabilidades RACI.xlsx](#)

Tecnologías

El desarrollo de aplicación web educativa se llevará a cabo utilizando un conjunto de tecnologías open source que permiten garantizar escalabilidad, facilidad de mantenimiento, bajo costo y un entorno flexible para la construcción del prototipo. A continuación, se detallan las herramientas seleccionadas según su propósito dentro del proyecto:

1. Back-end

- Django (Python): framework que facilita el desarrollo ágil de aplicaciones web seguras y escalables. Permite la integración de la lógica de negocio, la gestión de usuarios y la comunicación con la base de datos.
- SQL (SQLite): sistema gestor de bases de datos relacional que asegura integridad, consistencia y eficiencia en el manejo de la información académica (usuarios, progreso, recursos).

2. Front-end

- HTML, CSS y JavaScript: tecnologías estándar para el desarrollo de interfaces web, garantizando compatibilidad multiplataforma y accesibilidad.
- Frameworks/librerías complementarias: uso de Bootstrap para optimizar la maquetación y diseño responsivo.

3. Entorno de desarrollo

- Visual Studio Code: entorno de desarrollo integrado (IDE) ligero y altamente extensible.
- Git y GitHub: sistema de control de versiones y plataforma colaborativa para la gestión del código, ramas de desarrollo y seguimiento de cambios.

4. Diseño y experiencia de usuario (UX/UI)

- Figma: herramienta de diseño colaborativo que facilita la creación de prototipos, wireframes e interfaces intuitivas para los usuarios de la plataforma.
- Tiled: editor de mapas 2D open source utilizado para crear el mapa del minijuego. En Tiled se definen capas del escenario en archivos fuente XML (bloques, decoraciones, etc) y luego estos mapas se exportan a formato JSON. Dichos archivos JSON se integran con el motor del minijuego en la aplicación.

5. Gestión del proyecto

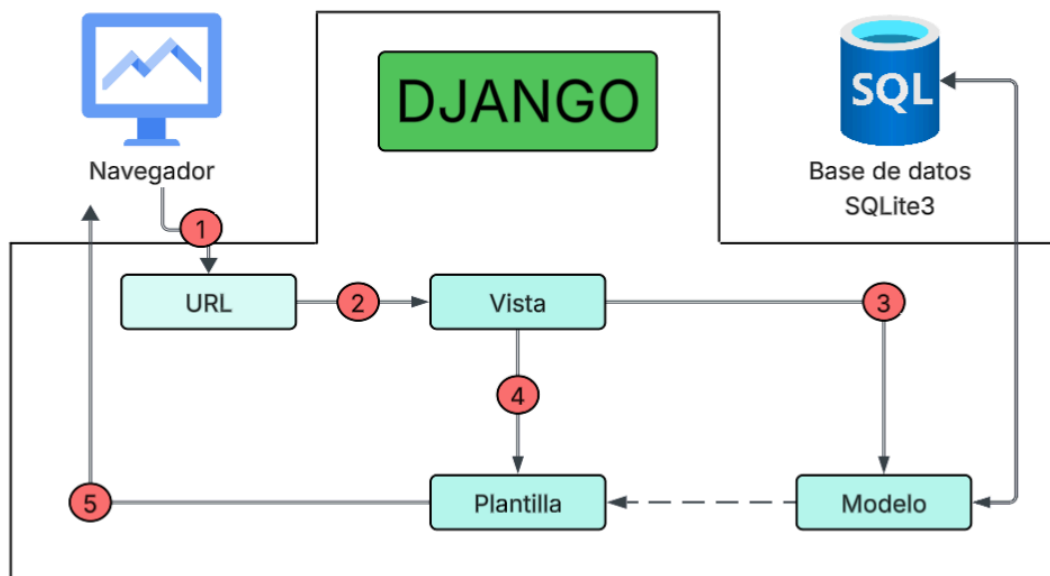
- Herramientas de gestión: Trello para organizar tareas, asignar responsables y dar seguimiento al avance del proyecto.

Arquitectura

La plataforma LevelUp se desarrolla utilizando el framework Django, siguiendo el patrón MVT (Model-View-Template), lo que permite una clara separación entre la lógica de negocio, la gestión de datos y la presentación de la información.

- **Modelo (Model):** Gestiona la estructura de los datos y la persistencia en la base de datos SQLite3, incluyendo usuarios, cursos, actividades y recompensas.
- **Vista (View):** Contiene la lógica de negocio, procesa las solicitudes del usuario, aplica validaciones y controla el flujo de la aplicación.
- **Plantilla (Template):** Define la interfaz web, presentando los datos de forma clara y adaptativa para estudiantes, docentes y administradores.

Este enfoque modular asegura un desarrollo ordenado y escalable, facilita el mantenimiento del sistema, mejora la seguridad y permite que los diferentes componentes puedan evolucionar de manera independiente. La arquitectura MVT junto con SQLite3 permite un rápido prototipado y pruebas eficientes, manteniendo la integridad de los datos y un control efectivo sobre los permisos y roles dentro de la plataforma.



Estructura de Desglose de trabajo

Definición de Diagrama x Matriz EDT.xlsx

EDT - Matriz Estructura de descomposición de tareas						
	DIAS HÁBILES	HORAS POR ACTIVIDAD O ENTREGABLE				
Fase de Planificación	13	JP	AF	DU	D	QA
Kick Off	1	3	2	2	2	1
Elaboración del plan de proyecto	4	20	16	9	2	2
Elaboración del registro de requerimientos del software (ERS)	3	12	9	4	2	2
Identificación de requerimientos	2	12	8	6	2	2
Definición de roles y responsabilidades	2	4	5	3	1	1
Análisis de riesgos inicial	1	5	4	2	2	1
Fase de Análisis y diseño	5	JP	AF	DU	D	QA
Modelado de casos de uso	1	3	5	3	3	2
Diagrama de clases	1	3	4	3	3	1
Diagrama de base de dato	1	3	4	3	3	1
Diseño UX/UI	1	3	5	6	2	2
Revisión de requerimientos	1	4	5	4	2	2
Fase de Desarrollo	53	JP	AF	DU	D	QA
Configuración del entorno de desarrollo	2	1	1	1	6	2
Implementación del registro y autenticación de usuarios.	5	7	6	8	24	1
Desarrollo de gestión de perfiles	6	9	11	16	36	4
Desarrollo de módulo para crear cursos y asignaturas	3	4	6	5	15	2
Desarrollo/Diseño de juego educativo	13	12	20	25	62	4
Desarrollo de módulo para crear y gestionar actividades.	22	12	15	40	124	13
Enlazar actividades/juegos con preguntas de docente	16	18	32	30	88	10
Implementación del sistema de recompensas y ranking	4	6	7	12	24	3
Fase de Pruebas	8	JP	AF	DU	D	QA
Preparación de pruebas (plan/casos/datos/entorno)	1	2	2	1	2	5
Pruebas unitarias (cobertura + calidad)	1	1	0	0	5	3
Pruebas de integración	1	1	1	1	5	4
Pruebas de sistema y regresión	1	1	1	2	3	4
Pruebas de seguridad	1	0	1	1	2	2
Pruebas de compatibilidad y rendimiento	1	1	1	1	3	2
Corrección de errores, ajustes finales y re-pruebas	2	2	4	2	12	6
Cierre	5	JP	AF	DU	D	QA
Ajustes finales	4	5	7	3	24	18
Documentación final	4	8	12	8	8	8
Presentación final y cierre del proyecto.	1	1	1	1	1	1

Diccionario EDT.

DICCIONARIO EDT			
ROL	SIGLA	NOMBRE	COSTO x HORA
Jefe Proyecto	JP	Antonia Rojas	\$10.688
Analista Funcional	AF	Antonia Rojas	\$7.994
Diseñador de UX/UI	DU	Millaray Ibarra	\$6.394
Desarrollador	D	Millaray Currihual	\$7.326
Tester/QA	QA	Millaray Ibarra	\$7.762

Costo por Fase y por Rol

COSTO POR FASE			
Fase de Planificación	\$1.266.952	COSTO HH POR ROL	
Fase de Análisis y Diseño	\$633.690	Jefe Proyecto	\$10.688
Fase de Desarrollo	\$5.476.134	Analista Funcional	\$7.994
Fase de Pruebas	\$652.840	Desarrollador	\$7.326
Fase de Despliegue y Cierre	\$837.572	Diseñador de UX/UI	\$6.394
TOTAL HH FASES	\$8.867.188	Tester/QA	\$7.762

Cronograma

Fase 1: Planificación (11/08 – 27/08, 2 semanas)

1. Kick-off del proyecto: reunión inicial para definir objetivos y alcance.
2. Elaboración del plan de proyecto: cronograma, recursos, costos, riesgos.
3. Elaboración del registro de requerimientos del software (ERS).
4. Identificación de requerimientos: levantamiento y documentación de requerimientos funcionales y no funcionales (usuarios, sistema de recompensas, docentes, actividades diarias).
5. Definición de roles y responsabilidades: asignación de tareas y matriz R.A.C.I.
6. Análisis de riesgos inicial: identificar riesgos de desarrollo, implementación y operación.

Fase 2: Análisis y diseño (28/08 – 03/09, 1 semana)

1. Modelado de casos de uso: diagramas de interacción entre actores y sistema.
2. Diagrama de clases: definir estructura de datos y relaciones.
3. Diagrama de base de datos: tablas, relaciones, llaves primarias/foráneas.
4. Diseño UX/UI: wireframes, prototipos de interfaz, diseño responsive.
5. Revisión de requerimientos: aseguramiento de que todos los requisitos estén contemplados en el diseño.

Fase 3: Desarrollo (04/09 – 17/11, 7 semanas)

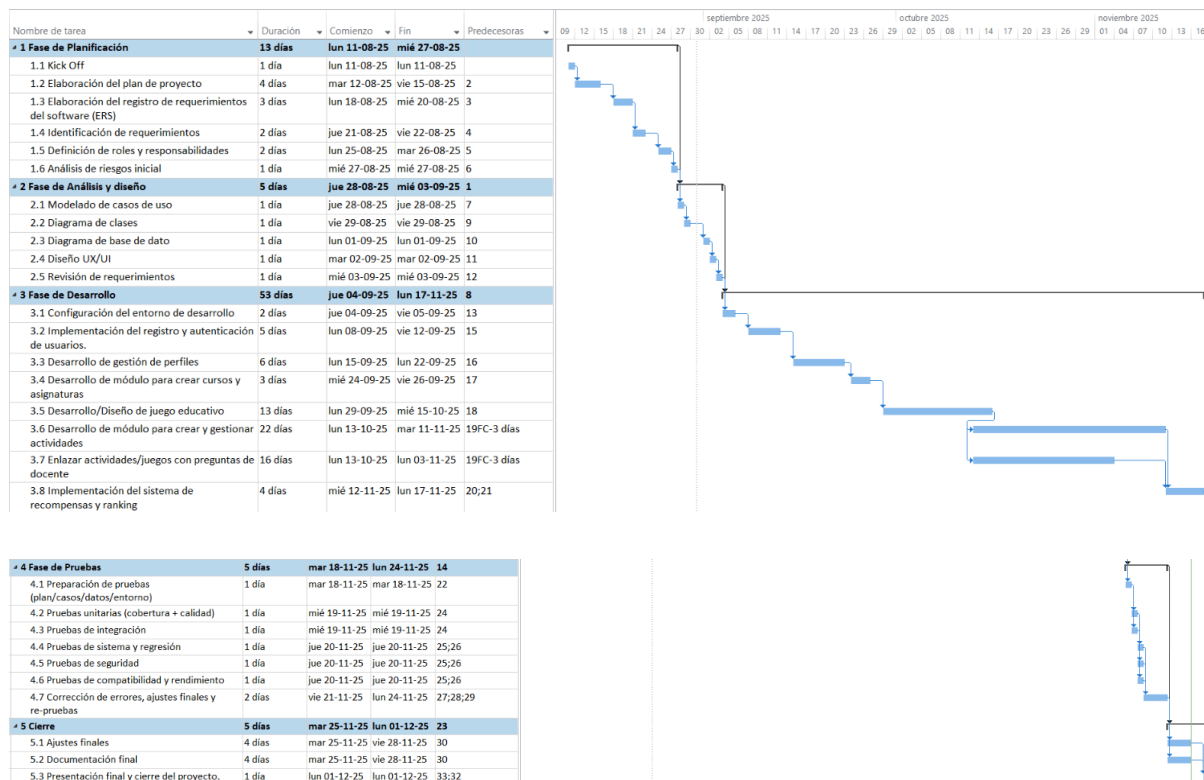
1. Configuración del entorno de desarrollo: Django, SQL, front-end (HTML/CSS/JS).
2. Implementación del registro y autenticación de usuarios.
3. Desarrollo de gestión de perfiles
4. Desarrollo de módulo para crear cursos y asignaturas
5. Desarrollo /Diseño de juego educativo
6. Desarrollo de módulo para crear y gestionar actividades
7. Enlazar actividades/juegos con preguntas de docente
8. Implementación del sistema de recompensas y ranking

Fase 4: Pruebas y ajustes (18/11 – 24/11, 1 semana)

1. Preparación de pruebas (plan/casos/datos/entorno)
2. Pruebas unitarias (cobertura + calidad)
3. Pruebas de integración: verificar que todos los módulos funcionen juntos correctamente.
4. Pruebas de sistema y regresión
5. Pruebas de seguridad
6. Pruebas de compatibilidad y rendimiento
7. Corrección de errores y ajustes finales y repruebas.

Fase 5: Cierre (25/11 – 01/12, 1 semana)

1. Ajustes finales
2. Documentación final
3. Presentación final y cierre del proyecto



Costos

Presupuesto de Proyecto

Líder del Proyecto: Antonia Rojas
 Fecha de Inicio: [15/08/2025]

[AM]

% Reserva de Contingencia

15%

Total					Presupuesto	Reservas	Total
					11.838.158	1.775.724	13.613.882
Código	Tarea / Actividad	Elemento	Tipo de Recurso	Tipo de Unidades	Unidades	Tasa	Presupuesto
1.1	LevelUp						11.838.158
1.1.1	[Fase de Planificación]						3.651.942
		Jefe de Proyecto	Labor (Personal)	Hora	56	\$10.688	\$598.528
		Analista Funcional	Labor (Personal)	Hora	44	\$7.994	\$351.736
		Diseñador de UX/UI	Labor (Personal)	Hora	26	\$6.394	\$166.244
		Desarrollador	Labor (Personal)	Hora	11	\$7.326	\$80.586
		Tester/QA	Labor (Personal)	Hora	9	\$7.762	\$69.858
		[Equipos de Comunicación]	Materiales	Pago Único	1	\$280.000	\$280.000
		[Equipos de Computación]	Materiales	Pago Único	1	\$2.100.000	\$2.100.000
		[Redes y Conectividad]	Materiales	Unidad	1	\$4.990	\$4.990
1.1.2	[Fase de Análisis y diseño]						\$633.690
		Jefe de Proyecto	Labor (Personal)	Hora	16	\$10.688	\$171.008
		Analista Funcional	Labor (Personal)	Hora	23	\$7.994	\$183.862
		Desarrollador	Labor (Personal)	Hora	13	\$7.326	\$95.238
		Tester/QA	Labor (Personal)	Hora	8	\$7.762	\$62.096
		Diseñador de UX/UI	Labor (Personal)	Hora	19	\$6.394	\$121.486
		[Equipos de Comunicación]	Materiales	Pago Único	0	\$280.000	\$0
		[Equipos de Computación]	Materiales	Pago Único	0	\$2.100.000	\$0
		[Redes y Conectividad]	Materiales	Unidad	0	\$4.990	\$0
1.1.3	[Fase de Desarrollo]						\$5.967.124
		Jefe de Proyecto	Labor (Personal)	Hora	69	\$10.688	\$737.472
		Analista Funcional	Labor (Personal)	Hora	98	\$7.994	\$783.412
		Desarrollador	Labor (Personal)	Hora	379	\$7.326	\$2.776.554
		Diseñador de UX/UI	Labor (Personal)	Hora	137	\$6.394	\$875.978
		Tester/QA	Labor (Personal)	Hora	39	\$7.762	\$302.718
		[Servidores]	Materiales	Pago Único	1	\$360.000	\$360.000
		[Equipos de Computación]	Materiales	Pago Único	0	\$2.100.000	\$0
		[Sistemas de Almacenamiento]	Materiales	Unidad	1	\$74.990	\$74.990
		[Figma Professional]	Licencias	Unidad	1	\$16.000	\$16.000
		[Redes y Conectividad]	Materiales	Unidad	0	\$4.990	\$0
		[Licencia ChatGPT-5]	Licencias	Mensual	2	\$20.000	\$40.000
1.1.4	[Fase de Pruebas]						\$747.830
		Jefe de Proyecto	Labor (Personal)	Hora	8	\$10.688	\$85.504
		Desarrollador	Labor (Personal)	Hora	32	\$7.326	\$234.432
		Diseñador de UX/UI	Labor (Personal)	Hora	8	\$6.394	\$51.152
		Analista Funcional	Labor (Personal)	Hora	10	\$7.994	\$79.940
		Tester/QA	Labor (Personal)	Hora	26	\$7.762	\$201.812
		[Servidores]	Materiales	Pago Único	0	\$360.000	\$0
		[Equipos de Computación]	Materiales	Pago Único	0	\$2.100.000	\$0
		[Sistemas de Almacenamiento]	Materiales	Unidad	1	\$74.990	\$74.990
		[Redes y Conectividad]	Materiales	Unidad	0	\$4.990	\$0
		[Licencia ChatGPT-5]	Licencias	Mensual	1	\$20.000	\$20.000
1.1.5	[Cierre]						\$837.572
		Jefe de Proyecto	Labor (Personal)	Hora	14	\$10.688	\$149.632
		Desarrollador	Labor (Personal)	Hora	33	\$7.326	\$241.758
		Diseñador de UX/UI	Labor (Personal)	Hora	12	\$6.394	\$76.728
		Analista Funcional	Labor (Personal)	Hora	20	\$7.994	\$159.880
		Tester/QA	Labor (Personal)	Hora	27	\$7.762	\$209.574
		[Servidores]	Materiales	Pago Único			
		[Equipos de Computación]	Materiales	Pago Único	0	\$2.100.000	\$0
		[Sistemas de Almacenamiento]	Materiales	Unidad	0	\$74.990	\$0
		[Redes y Conectividad]	Materiales	Unidad	0	\$4.990	\$0

Presupuesto de un Proyecto: ☒ Presupuesto del Proyecto Actualizado.xlsx

Riesgos

Riesgo	Fase	Probabilidad	Impacto	Acción de mitigación
Retrasos en la planificación o definición de requerimientos	Planificación	Media	Alto	Reuniones periódicas, seguimiento de tareas con cronograma, uso de checklist de requerimientos.
Definición inadecuada de requisitos y necesidades durante la fase de planificación	Planificación	Media	Alto	Documentar claramente cada requerimiento, validar y aprobar formalmente antes de iniciar diseño.
Sobrecostos en el desarrollo del proyecto	Planificación / Desarrollo	Media	Alto	Control riguroso del presupuesto, estimaciones realistas, priorización de funcionalidades críticas y revisión de alcance en cada fase.
Diseño UX/UI poco intuitivo o no adaptable	Análisis y diseño	Media	Medio	Pruebas de usabilidad tempranas, feedback de usuarios, ajustes iterativos.
Errores en la configuración de roles y permisos (docentes/estudiantes/admins)	Desarrollo	Media	Alto	Revisar matriz de roles, pruebas de acceso diferenciadas, validación de reglas en QA.
Problemas en la integración front-end / back-end	Desarrollo	Media	Alto	Implementación por módulos, pruebas de integración continuas, documentación.
Fallos en el sistema de autenticación y seguridad	Desarrollo	Media	Alto	Implementar estándares de seguridad, pruebas de vulnerabilidad y validación de accesos con roles.
Retrasos en desarrollo por complejidad del sistema de recompensas	Desarrollo	Media	Medio	Priorizar funcionalidades críticas, desarrollar el sistema de recompensas en fases, pruebas parciales para reducir riesgos.
Errores en la autenticación del usuario durante la fase de implementación	Implementación	Media	Alto	Pruebas exhaustivas de login/logout, validación de credenciales y recuperación de contraseñas antes del despliegue.

Falta de pruebas de carga en el sistema durante la fase de implementación	Implementación	Media	Alto	Simular un alto número de usuarios concurrentes, realizar pruebas de rendimiento y optimizar consultas/servidor antes del lanzamiento.
Retrasos en la implementación, falta de seguimiento adecuado y fallos en la comunicación.	Implementación/Despliegue	Media	Alto	Reuniones de control de avances, asignación clara de responsables, herramientas de seguimiento.

Pruebas

Las pruebas del sistema se diseñaron para asegurar que la plataforma educativa gamificada cumpliera tanto los requerimientos funcionales como los no funcionales establecidos. Se evaluaron todos los módulos relevantes del MVP, incluyendo autenticación, registro y gestión de usuarios, actividades interactivas, sistema de recompensas, dashboard de seguimiento, administración de contenidos y seguridad.


Para ello se aplicaron distintos tipos de pruebas:


- **Unitarias**, orientadas a validar la lógica interna de funciones esenciales como validación de formularios, creación de usuarios, autenticación y operaciones CRUD.
- **Integración**, verificando la comunicación entre módulos como login–dashboard, actividades–recompensas y gestión de usuarios–roles.
- **Rendimiento y carga**, evaluando tiempos de respuesta y estabilidad con múltiples solicitudes concurrentes.
- **Usabilidad**, comprobando la claridad de la interfaz, accesibilidad y fluidez de navegación para estudiantes, docentes y administradores.
- Se emplearon técnicas de caja negra, evaluando entradas y salidas del sistema, y de caja blanca, revisando la correcta implementación de lógica y flujos internos.

Los criterios de aceptación definidos fueron:

- Ejecución satisfactoria del 100% de los casos de prueba.
- Corrección obligatoria de cualquier defecto crítico o alto.
- Verificación completa de los requerimientos del MVP.
- Aprobación de la funcionalidad por parte de los usuarios clave.
- Registro formal de evidencias y estabilidad general del sistema.

Entre los principales riesgos considerados estuvieron retrasos en la definición inicial del alcance, errores en la integración entre módulos, inconsistencias en permisos y autenticación, y falta de validación bajo carga. Para mitigarlos, se realizaron revisiones periódicas de avance, ajuste del cronograma, aplicación de pruebas iterativas y estrategias de contingencia para asegurar la continuidad del proyecto.

Plan de Prueba:  Copia de Plan de Pruebas.docx

Casos de Pruebas y métricas:  Pruebas

Lecciones aprendidas

Durante el desarrollo del proyecto LevelUp, el equipo se enfrentó a diversos desafíos que permitieron aprender y mejorar la gestión, el desarrollo y la calidad del producto. Entre los principales problemas se identificaron dificultades en la organización del equipo, que se solucionaron mediante la implementación de herramientas de gestión de tareas como Trello y GitHub, la definición de responsables por módulo y reuniones de sincronización periódicas.

Asimismo, se presentó incertidumbre respecto al manejo y consultas a la base de datos, mitigada mediante un flujo de control de versiones y la realización de respaldos antes de efectuar cambios importantes. La estructura inicial del código fuente requería limpieza y orden, por lo que se llevaron a cabo revisiones sistemáticas para eliminar elementos innecesarios y mantener un código más organizado y comprensible.

El diseño de la interfaz fue otro desafío, especialmente al intentar que fuera atractiva y lúdica; esto se abordó incorporando elementos visuales, recompensas, íconos y microinteracciones, aplicando principios de gamificación para mejorar la experiencia del usuario. Las modificaciones en el alcance del proyecto tuvieron como consecuencia ajustes en las tareas planificadas y retrasos en algunos entregables, enseñando la importancia de la flexibilidad y la adaptación continua.

La brecha de conocimientos en diseño y desarrollo de minijuegos se superó mediante un plan de aprendizaje enfocado, cursos cortos y la creación de micro-prototipos con retroalimentación constante. Finalmente, los cambios frecuentes en los requerimientos destacaron la necesidad de implementar iteraciones cortas tipo sprint y mantener una trazabilidad clara de todas las decisiones tomadas.

Estas experiencias permitieron al equipo adquirir aprendizajes valiosos en gestión de proyectos, desarrollo de software y diseño de interfaces educativas, estableciendo buenas prácticas que serán aplicables en futuros desarrollos.