



Universidad del Valle de
Guatemala
Facultad de Ingeniería
Ingeniería de Software I

EduVial: Aprendizaje para todos

Corte 3

Daniel Oswaldo Juárez Herrera – 23709

Dilary Sarahí Cruz López - 231010

Andrés Alberto Mazariegos Escobar – 21749

José André Estrada Contreras – 23659

Jonathan Alejandro Díaz tahuite – 23837

Resumen

El proyecto **"EduVial: Aprendizaje para todos"** tiene como objetivo desarrollar una plataforma interactiva de educación vial dirigida a personas que desean obtener su licencia de conducir en Guatemala. La falta de conocimiento sobre las leyes de tránsito y la ausencia de recursos accesibles para aprender y practicar de manera efectiva han sido identificados como factores clave en la alta tasa de accidentes de tránsito en el país. Según datos de la Superintendencia de Transporte Público y Tránsito (STPT, 2022), la falta de educación vial es una de las principales causas de infracciones y accidentes en Guatemala.

El proyecto busca abordar esta problemática mediante la creación de una aplicación móvil y una plataforma web que ofrezca módulos de aprendizaje interactivos, simulaciones de conducción y un sistema de recompensas para motivar a los usuarios. Los objetivos concretos incluyen identificar los principales obstáculos en el aprendizaje de normas de tránsito, categorizar a los diferentes tipos de usuarios afectados por la problemática y diseñar una solución que sea accesible y efectiva para todos los niveles socioeconómicos.

Introducción

El proyecto **"EduVial : Aprendizaje para todos"** está dirigido a personas que desean obtener su licencia de conducir en Guatemala, pero enfrentan barreras de acceso a información y práctica efectiva. La plataforma está diseñada para ser utilizada por conductores principiantes, avanzados, personas mayores, y aquellos que buscan reforzar sus conocimientos antes de presentar el examen de conducción. Además, se incluye un módulo de administración para gestionar el contenido y monitorear el progreso de los usuarios.

La idea central del proyecto es abordar la problemática de la falta de educación vial en Guatemala, donde el desconocimiento de las leyes de tránsito y la falta de habilidades prácticas de conducción han sido identificados como factores clave en la alta tasa de accidentes de tránsito. Actualmente, el proceso de obtención de licencias no garantiza que los conductores posean un conocimiento

sólido de las normativas viales, lo que resulta en infracciones, accidentes y pérdida de vidas.

Los objetivos generales del proyecto incluyen:

- Identificar los principales obstáculos en el aprendizaje de normas de tránsito en Guatemala.
- Diseñar una plataforma interactiva que facilite el aprendizaje teórico y práctico de la educación vial.
- Implementar un sistema de recompensas y comunidad para motivar a los usuarios a continuar aprendiendo.

Los objetivos específicos son:

- Describir la problemática actual de la educación vial en Guatemala.
- Presentar las soluciones propuestas, incluyendo la aplicación móvil y la plataforma web.
- Establecer los requisitos funcionales y no funcionales del sistema.

Entrevista 1 - Usuario sin login

- Fecha: 21 de Febrero de 2025
- Entrevistado: David Enrique Ozaeta Maldonado
- Objetivo: Analizar la percepción de usuarios sin cuenta sobre la educación vial y la utilidad de una aplicación.
- Hallazgos:
 - o Considera que la educación vial es importante, pero poco accesible.
 - o Está interesado en conocer las reglas de tránsito antes de empezar a manejar.

- Cree que una plataforma que ofrezca contenido gratuito ayudaría a generar más interés en la educación vial.

Entrevista 2 - Conductor principiante

- Fecha: 22 de Febrero de 2025
- Entrevistado: Mariana Fernanda Rojas Gonzales
- Objetivo: Identificar dificultades en el aprendizaje de las normas de tránsito.
- Hallazgos:
 - No conoce completamente las leyes de tránsito y considera que deberían ser más accesibles.
 - Opina que la falta de educación vial desde la infancia contribuye a un mal manejo en Guatemala.
 - Sugiere la creación de materiales didácticos interactivos para mejorar la enseñanza de las normas.

Entrevista 3 - Conductor avanzado

- Fecha: 21 de Febrero de 2025
- Entrevistado: Jesús Alberto Mazariegos Robledo
- Objetivo: Conocer sus necesidades de actualización en normas viales.
- Hallazgos:
 - Aprendió a manejar de forma empírica sin asistencia de una academia.
 - Considera que la corrupción en los exámenes de licencia afecta la educación vial.
 - Sugiere incluir la educación vial en colegios para mejorar la formación desde temprana edad.

Entrevista 4 - Administrador de contenido

- Fecha: 25 de febrero de 2025
- Entrevistado: Andrés Alberto Mazariegos Escobar
- Objetivo: Conocer las necesidades en la gestión de contenido educativo dentro de la plataforma.
- Hallazgos:
 - Destaca la importancia de mantener la información actualizada sobre normativas de tránsito.
 - Considera esencial una interfaz intuitiva para la administración del contenido.
 - Sugiere implementar reportes de uso para evaluar el impacto de las lecciones en los usuarios.

Prototipo realizado con la realimentación de los usuarios:

Lista de requisitos no funcionales

<i>Requisito no funcional</i>	<i>Categoría</i>	<i>Forma en la que se medirá su cumplimiento</i>
La aplicación debe mostrar imágenes claras de las señales de tránsito	Interfaz	Se verifica que las imágenes tengan una resolución mínima de 720p
La aplicación debe ser compatible con dispositivos Android a partir de la versión 8.0 (Oreo) en adelante	Compatibilidad	Utilizando emuladores de los distintos sistemas Android
La aplicación debe mostrar las notificaciones de cambio de ley, en un máximo de 1 hora después de ser legislada	Eficiencia	Se harán pruebas simuladas de estos casos

Los datos de los usuarios deben estar encriptados para su seguridad	Seguridad	Se harán pruebas de seguridad para verificar la confidencialidad
La aplicación debe de soportar un mínimo de 1000 usuarios a la vez	Escalabilidad	Se someterá a la aplicación a pruebas de estrés
La aplicación debe tener actualizadas las leyes de tránsito	Mantenimiento	Se verifica cada mes si las leyes han sufrido cambios
La aplicación debe tener una interfaz adaptable a distintos dispositivos	Compatibilidad	Se harán pruebas simuladas para teléfono, tabletas y computadoras
La aplicación debe permitir la autenticación mediante	Interfaz	Se realizarán pruebas de inicio de sesión con diferentes cuentas para verificar su funcionamiento.
La aplicación debe mantener una disponibilidad del 99% en los servidores.	Confiabilidad	Se monitoreará el tiempo de actividad del servidor
La aplicación debe permitir a los usuarios restablecer su contraseña en 2 pasos	Seguridad	Se realizarán pruebas para medir los pasos promedio del proceso de restablecimiento de contraseña.

Lista de requisitos funcionales

<i>Categoría</i>	<i>Requisito Funcional</i>
Gestión de Usuarios	El sistema debe permitir a los usuarios registrarse en la plataforma proporcionando un nombre, correo electrónico y contraseña
Gestión de Usuarios	El sistema debe permitir a los usuarios iniciar sesión con su correo electrónico y contraseña
Gestión de Usuarios	El sistema debe permitir a los usuarios editar su perfil nombre, foto de perfil, preferencias de aprendizaje, etc
Módulos de Aprendizaje	El sistema debe ofrecer módulos de aprendizaje interactivos sobre normas de tránsito, organizados por temas
Módulos de Aprendizaje	El sistema debe incluir simulaciones de conducción para que los usuarios practiquen situaciones reales de tránsito

Módulos de Aprendizaje	El sistema debe permitir a los usuarios avanzar en los módulos de aprendizaje de manera progresiva
Módulos de Aprendizaje	El sistema debe ofrecer materiales didácticos interactivos videos, imágenes, cuestionarios
Módulos de Aprendizaje	El sistema debe implementar un sistema de recompensas puntos, insignias, niveles para motivar a los usuarios.
Gestión de Contenido	El sistema debe permitir a los administradores agregar, editar y eliminar contenido educativo
Gestión de Contenido	El sistema debe permitir a los administradores actualizar las normativas de tránsito en tiempo real
Gestión de Contenido	El sistema debe notificar a los usuarios sobre cambios en las leyes de tránsito en un máximo de 1 hora después de ser actualizadas
Interfaz y Experiencia	El sistema debe mostrar imágenes claras y de alta resolución (mínimo 720p) de las señales de tránsito
Seguridad y Privacidad	El sistema debe encriptar los datos de los usuarios para garantizar su seguridad y privacidad

- Prototipos

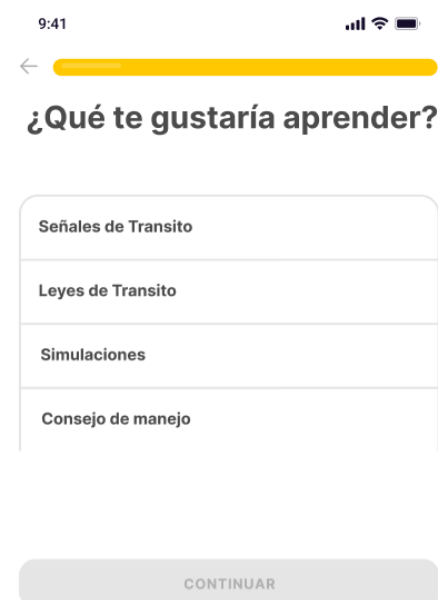
9:41



Modulo 1

Conoce las señales de transito

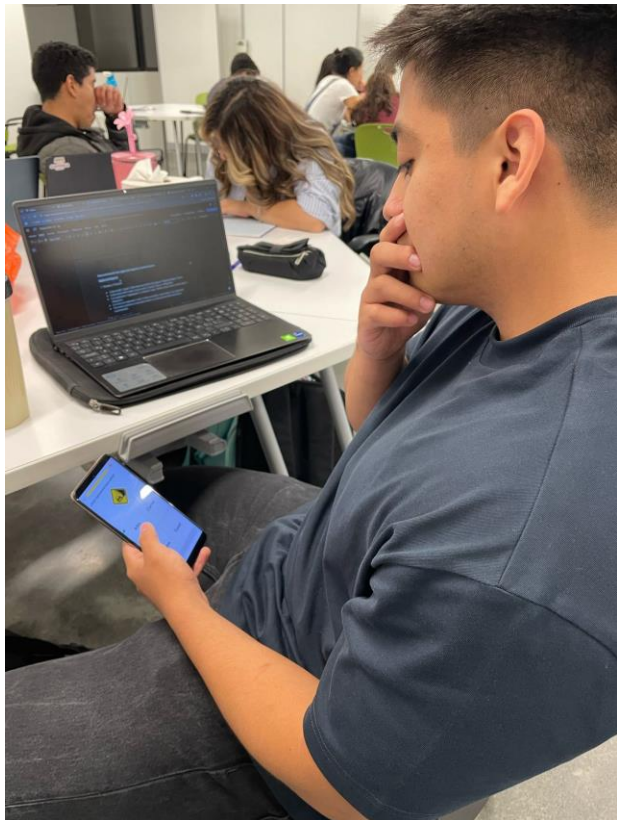




- Evidencias

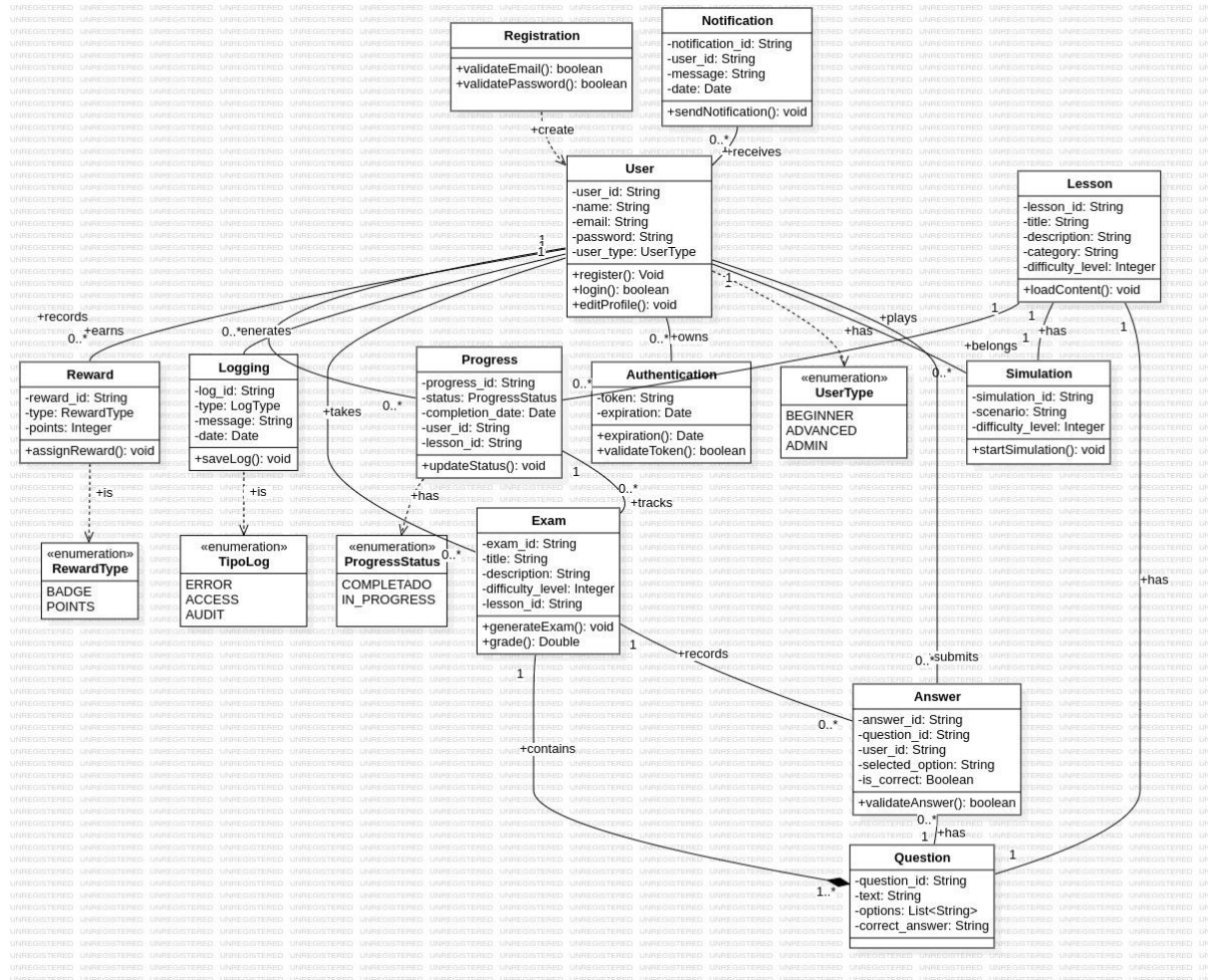






Clases Preliminares

- Diagramas de clases



- Descripción de clases

1. Registration (Registro)

- **Atributos:** Ninguno
- **Métodos:**
 - +validateEmail(): boolean - Valida el formato del correo electrónico
 - +validatePassword(): boolean - Valida que la contraseña cumpla con requisitos

2. User (Usuario)

- **Atributos:**
 - -user_id: String - Identificador único del usuario
 - -name: String - Nombre del usuario
 - -email: String - Correo electrónico
 - -password: String - Contraseña (almacenada de forma segura)
 - -user_type: UserType - Tipo de usuario (enum: BEGINNER, ADVANCED, ADMIN)
- **Métodos:**
 - +register(): Void - Registra un nuevo usuario
 - +login(): boolean - Inicia sesión
 - +editProfile(): void - Edita información del perfil

3. Notification (Notificación)

- **Atributos:**
 - -notification_id: String - Identificador único de la notificación
 - -user_id: String - Usuario destinatario
 - -message: String - Contenido del mensaje
 - -date: Date - Fecha de la notificación
- **Métodos:**
 - +sendNotification(): void - Envía la notificación al usuario

4. Logging (Registro de Actividad)

- **Atributos:**
 - -log_id: String - Identificador del registro
 - -type: TipLog - Tipo de registro (enum: ERROR, ACCESS, AUDIT)
 - -message: String - Mensaje del registro
 - -date: Date - Fecha del registro
- **Métodos:**

- +saveLog(): void - Guarda el registro en el sistema

5. Reward (Recompensa)

- **Atributos:**
 - -reward_id: String - Identificador de la recompensa
 - -type: RewardType - Tipo de recompensa (enum: BADGE, POINT)
 - -points: Integer - Puntos asociados
- **Métodos:**
 - +assignReward(): void - Asigna la recompensa al usuario

6. Progress (Progreso)

- **Atributos:**
 - -progress_id: String - Identificador del progreso
 - -status: ProgressStatus - Estado del progreso (enum: COMPLETED, IN_PROGRESS)
 - -completion_date: Date - Fecha de finalización
 - -user_id: String - Usuario asociado
 - -lesson_id: String - Lección asociada
- **Métodos:**
 - +updateStatus(): void - Actualiza el estado del progreso

7. Authentication (Autenticación)

- **Atributos:**
 - -token: String - Token de autenticación
 - -expiration: Date - Fecha de expiración
- **Métodos:**
 - +expiration(): Date - Obtiene la fecha de expiración
 - +validateToken(): boolean - Valida si el token es válido

8. Lesson (Lección)

- **Atributos:**
 - -lesson_id: String - Identificador único de la lección
 - -title: String - Título de la lección
 - -description: String - Descripción del contenido
 - -category: String - Categoría de la lección
 - -difficulty_level: Integer - Nivel de dificultad
- **Métodos:**
 - +loadContent(): void - Carga el contenido de la lección

9. Simulation (Simulación)

- **Atributos:**
 - -simulation_id: String - Identificador de la simulación
 - -scenariio: String - Escenario de la simulación
 - -difficulty_level: Integer - Nivel de dificultad
- **Métodos:**
 - +startSimulation(): void - Inicia la simulación

10. Exam (Examen)

- **Atributos:**
 - -exam_id: String - Identificador del examen
 - -title: String - Título del examen
 - -description: String - Descripción del examen
 - -difficulty_level: Integer - Nivel de dificultad
 - -lesson_id: String - Lección asociada
- **Métodos:**
 - +generateExam(): void - Genera un nuevo examen
 - +grade(): Double - Califica el examen

11. Answer (Respuesta)

- **Atributos:**
 - -answer_id: String - Identificador de la respuesta
 - -question_id: String - Pregunta asociada
 - -user_id: String - Usuario que respondió
 - -selected_option: String - Opción seleccionada
 - -is_correct: Boolean - Indica si es correcta
- **Métodos:**
 - +validateAnswer(): boolean - Valida si la respuesta es correcta

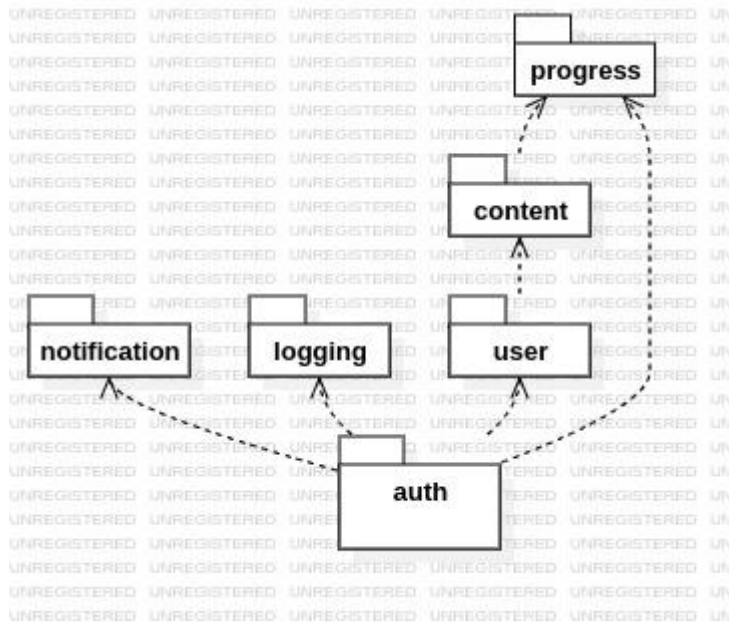
12. Question (Pregunta)

- **Atributos:**
 - -question_id: String - Identificador de la pregunta
 - -text: String - Texto de la pregunta
 - -options: List<String> - Lista de opciones
 - -correct_answer: String - Respuesta correcta

Enumeraciones:

1. **UserType:** BEGINNER, ADVANCED, ADMIN
2. **RewardType:** BADGE, POINT
3. **TipLog:** ERROR, ACCESS, AUDIT
4. **ProgressStatus:** COMPLETED, IN_PROGRESS

- Diagrama de paquetes



- Descripción de paquetes

1. Paquete "auth"

Descripción: Contiene las clases relacionadas con la autenticación, registro y gestión de usuarios.

- **Clases incluidas:** Registration, Authentication, User
- **Responsabilidades:**
 - Gestionar el registro de nuevos usuarios
 - Manejar la autenticación y autorización
 - Administrar los perfiles de usuario y sus tipos
 - Controlar los procesos de inicio y cierre de sesión

2. Paquete "notification"

Descripción: Gestiona el sistema de notificaciones para mantener informados a los usuarios.

- **Clases incluidas:** Notification
- **Responsabilidades:**
 - Crear y enviar notificaciones a los usuarios
 - Gestionar los tipos de mensajes y su distribución

- Mantener un registro de las notificaciones enviadas

3. Paquete "logging"

Descripción: Proporciona mecanismos para registrar actividades, eventos y errores del sistema.

- **Clases incluidas:** Logging, TipLog (enumeración)
- **Responsabilidades:**
 - Registrar acciones de los usuarios
 - Documentar errores del sistema
 - Mantener un histórico de auditoría
 - Clasificar los diferentes tipos de registros

4. Paquete "user"

Descripción: Gestiona elementos relacionados con los usuarios más allá de la autenticación.

- **Clases incluidas:** Reward, RewardType (enumeración)
- **Responsabilidades:**
 - Administrar el sistema de recompensas
 - Asignar insignias y puntos a los usuarios
 - Gestionar los diferentes tipos de incentivos

5. Paquete "content"

Descripción: Contiene las clases relacionadas con el contenido educativo y evaluaciones.

- **Clases incluidas:** Lesson, Exam, Question, Answer, Simulation
- **Responsabilidades:**
 - Gestionar el contenido de las lecciones
 - Administrar los exámenes y sus preguntas
 - Controlar las simulaciones interactivas
 - Procesar las respuestas de los usuarios

6. Paquete "progress"

Descripción: Maneja el seguimiento del avance de los usuarios a través del contenido.

- **Clases incluidas:** Progress, ProgressStatus (enumeración)
- **Responsabilidades:**
 - Registrar el progreso de los usuarios en lecciones
 - Actualizar el estado de finalización
 - Generar estadísticas de avance
 - Mantener el historial de actividades completadas

Relaciones entre Paquetes:

1. **auth** → **notification**: El sistema de autenticación utiliza el paquete de notificaciones para informar a los usuarios sobre eventos relacionados con su cuenta.
2. **auth** → **logging**: Los procesos de autenticación generan registros de actividad para auditoría y seguridad.
3. **auth** → **user**: El sistema de autenticación interactúa con el paquete de usuario para gestionar recompensas basadas en la actividad.
4. **auth** → **progress**: Los usuarios autenticados tienen su progreso rastreado a través del sistema.
5. **content** → **progress**: El contenido educativo se relaciona con el progreso para actualizar el avance del usuario.
6. **user** → **content**: Los usuarios acceden y utilizan el contenido educativo del sistema.

Persistencia de Datos

- Diagrama de clases persistentes
 - Usuario:
 - id_usuario
 - nombre
 - correo
 - contraseña
 - tipo_usuario
 - Lección
 - id_leccion
 - titulo
 - descripcion
 - categoría
 - nivel_dificultad
 - Progreso
 - id_progreso

- id_usuario
- id_leccion
- estado
- fecha_completado
- Examen
 - id_examen
 - titulo
 - descripcion
 - nivel_dificultad
- Pregunta
 - id_pregunta
 - id_examen
 - texto_pregunta
 - tipo_pregunta
- Opcion
 - id_opcion
 - id_pregunta
 - texto_opcion
 - es_correcta
- Historial_Notificaciones
 - id_historial
 - id_usuario
 - mensaje
 - fecha_envio

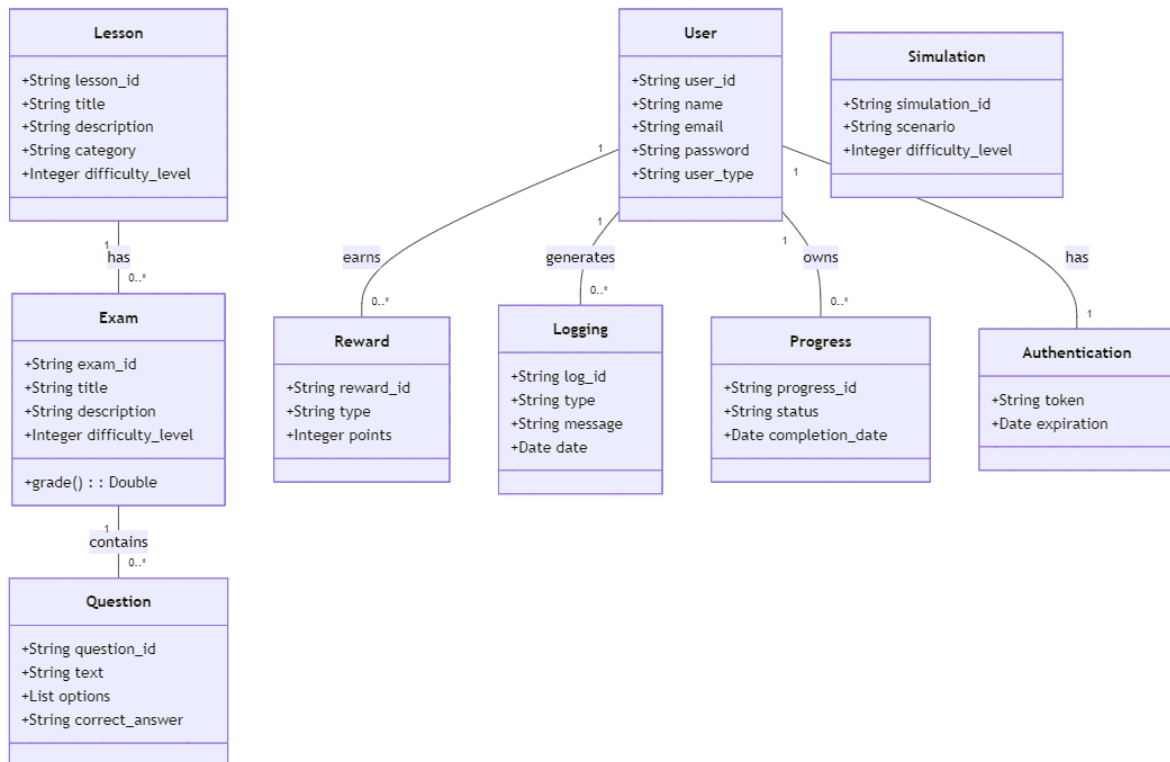


Figura 1. Diagrama de clases persistentes

- Tecnologías de almacenamiento consideradas
 - PostgreSQL: PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto que se destaca por su robustez y capacidad para manejar datos complejos.
 - Ventajas: El motor SQL facilita el procesamiento eficiente de grandes volúmenes de datos y consultas complejas, además garantiza la consistencia y fiabilidad de las operaciones mediante el cumplimiento de las propiedades de Atomicidad, Consistencia, Aislamiento y Durabilidad; permitiendo la creación de funciones y tipos de datos personalizados.
 - Desventajas: Su amplia gama de características puede requerir una curva de aprendizaje más pronunciada, además, en entornos con recursos limitados, su uso intensivo de memoria y CPU puede suponer un problema.

- MySQL: Es un sistema de gestión de bases de datos relacional de código abierto, conocido por su simplicidad y eficiencia en operaciones de lectura.
 - Ventajas: Su configuración y administración son intuitivas, facilitando su integración, además, su uso optimizado para operaciones de lectura rápidas es adecuado para aplicaciones con alta demanda de consultas simples.
 - Desventajas: Carece de algunas características avanzadas presentes en otros SGBD, como PostgreSQL. Además, su soporte para tipos de datos personalizados es limitado.
- MariaDB: MariaDB es una bifurcación de MySQL, creada por el equipo original de desarrollo, esta busca ofrecer mejoras en rendimiento y características adicionales.
 - Ventajas: Permite una transición sencilla desde MySQL, manteniendo la mayoría de las funcionalidades y comandos, ofreciendo mejoras de eficiencia y velocidad.
 - Desventajas: Al no ser respaldado por Oracle, puede carecer de ciertas garantías y soporte empresarial.
- Firebase: Es una plataforma de desarrollo de aplicaciones respaldada por Google, que ofrece una base de datos NoSQL basada en documentos y en tiempo real. Está diseñada para aplicaciones web y móviles, proporcionando sincronización en tiempo real y escalabilidad en la nube.
 - Ventajas: Permite que el progreso de los usuarios y las notificaciones se actualicen instantáneamente en todos los dispositivos. Fácil integración con apps móviles

- Desventajas: Si los usuarios acceden a muchos datos al mismo tiempo, los costos pueden subir rápidamente. No permite consultas SQL avanzadas.
- MongoDB: Es una base de datos NoSQL orientada a documentos que almacena datos en formato JSON/BSON. Es altamente escalable y flexible, ideal para aplicaciones con estructuras de datos dinámicas.
 - Ventajas: Ideal para manejar usuarios con distintos niveles de aprendizaje, módulos personalizados. Beneficioso si se generan muchas preguntas, exámenes y registros de progreso de usuarios.
 - Desventajas: Si necesitas asegurar que una lección y su progreso se guarden juntos de forma 100% consistente. Los datos duplicados en documentos JSON pueden hacer que el sistema use más espacio de almacenamiento.
- Apache Cassandra: Es una base de datos distribuida NoSQL diseñada para manejar grandes volúmenes de datos de forma escalable y con alta disponibilidad. Utiliza un modelo basado en columnas en lugar de documentos o tablas tradicionales.
 - Ventajas: Si EduVial almacena grandes cantidades de datos de usuarios, pruebas y recompensas, Cassandra puede manejarlo eficientemente. No hay riesgo de caídas en el servicio.
 - Desventajas: No permite relaciones complejas como SQL, lo que puede dificultar el análisis de datos educativos detallados. No es tan fácil de administrar como Firebase o MongoDB.

Dado que la aplicación maneja datos complejos, se seleccionó PostgreSQL como la tecnología de almacenamiento principal, esto con el motivo de que los datos sean beneficiados por su modelo objeto-relacional; garantizando las propiedades ACID de la base de datos. Además, la capacidad de PostgreSQL para escalar horizontal y verticalmente, junto con su extensibilidad y soporte para una amplia variedad de tipos de datos, lo convierte en una opción robusta para aplicaciones que requieren adaptabilidad y crecimiento.

- Diagrama entidad relación

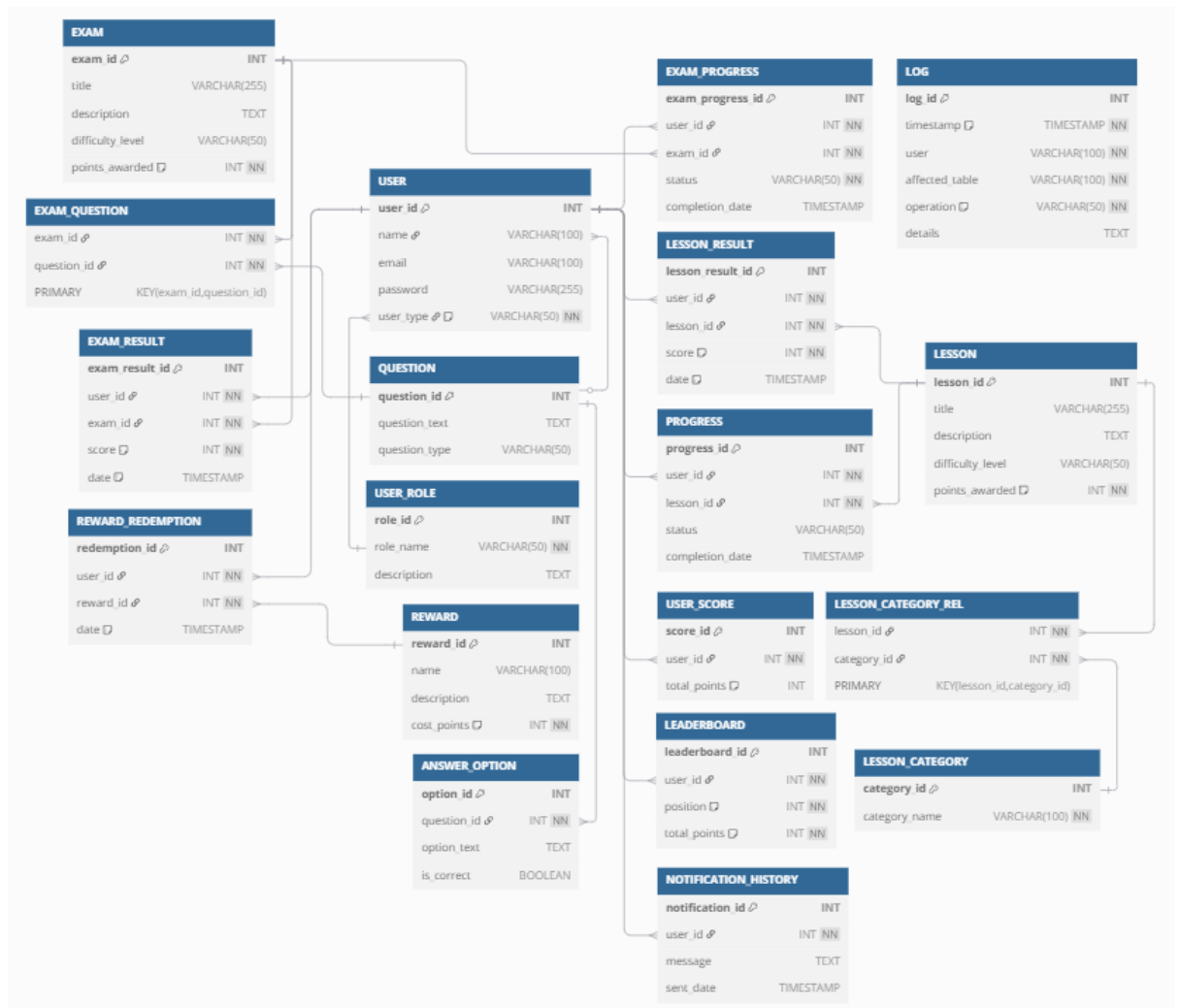


Figura 2. Diagrama entidad-relación

Tecnología a utilizar

Front: Flutter

Descripción de la tecnología :

Flutter es un framework de desarrollo de interfaces de usuario creado por Google que permite construir aplicaciones nativas para Android, iOS, Web y escritorio desde una única base de código. Se basa en el lenguaje **Dart** y utiliza su propio motor de renderizado para lograr un alto rendimiento y personalización de la UI.

- Lenguaje principal: Dart.
- Paradigma: Basado en widgets reutilizables con un enfoque declarativo.
- Compatibilidad: Soporte nativo para integración con hardware como cámara, GPS y sensores mediante plugins. Compatible con Firebase, SQLite y APIs REST.
- Ventajas y Desventajas:
 - Desarrollo rápido gracias a Hot Reload.
 - UI altamente personalizable y fluida, ideal para animaciones.
 - Soporte multiplataforma (Android, iOS, Web y Desktop).
 - Rendimiento cercano al nativo gracias a su motor gráfico propio (Skia).
 - Gran comunidad y documentación oficial detallada.

Desventajas:

- El lenguaje Dart es menos común
- Las apps creadas con Flutter suelen ser más pesadas
- Dependencia del motor gráfico de flutter
- Por qué la seleccionamos

Se eligió Flutter porque permite desarrollar una aplicación con un solo código para Android e iOS, asegurando un alto rendimiento y una interfaz atractiva. Además, su sistema de widgets altamente personalizable lo hace ideal para implementar gráficos, notificaciones y experiencias interactivas de manera eficiente.

Back:Node.js con Express

Descripción de la tecnología :

Es un entorno de ejecución para JavaScript en el servidor, basado en el motor V8 de Chrome. Express.js es un framework ligero para construir APIs de manera eficiente.

- Lenguaje principal: JavaScript.
- Paradigma: Asíncrono y basado en eventos.
- Compatibilidad: Funciona bien con bases de datos NoSQL (MongoDB, Firebase) y SQL (PostgreSQL, MySQL).

- Ventajas y Desventajas:

Alto rendimiento con peticiones concurrentes.
Gran ecosistema de paquetes con npm.
Escalabilidad con microservicios y WebSockets.
Fácil integración con bases de datos modernas.

Desventajas:

No es ideal para aplicaciones con alto uso de CPU.
Puede requerir más configuraciones de seguridad.

- Por qué la seleccionamos

Node.js con Express se seleccionó para el backend porque permite manejar peticiones de manera eficiente, facilitando la creación de una API REST para gestionar usuarios, preguntas y estadísticas de aprendizaje.

REPO:

https://github.com/andresm220/Ing_Software