REPÚBLICA BOLIVARIANA DE VENEZUELA

UNIVERSIDAD NACIONAL EXPERIMENTAL DE LOS LLANOS CENTRALES

“RÓMULO GALLEGOS”

ÁREA DE INGENIERÍA DE SISTEMAS

UNIDAD CURRICULAR: PROYECTO II

Herramienta digital de Gestión Educativa: Una alternativa de optimización del proceso de enseñanza-aprendizaje AIS

Autor: José Pérez
C.I.: V-30.039.812
Tutor Académico: Betania Belisario
Tutor Metodológico: Mely Orta

San Juan de los Morros, Noviembre de 2024

Resumen

El presente proyecto busca desarrollar una aplicación móvil orientada a la gestión educativa en el ámbito de Ingeniería en Sistemas. Este sistema pretende automatizar procesos clave, mejorar la comunicación y la sincronización de datos entre usuarios en momentos clave del proceso educativo y centralizar el acceso a recursos académicos, todo a través de una plataforma digital adaptada a las necesidades de estudiantes y profesores. La metodología de desarrollo empleada es Scrum, una metodología ágil que permite un desarrollo iterativo y eficiente, enfocado en la adaptabilidad y la retroalimentación continua. La aplicación está diseñada para mejorar la experiencia educativa en términos de gestión, interacción y accesibilidad.

Palabras clave: gestión educativa, aplicación móvil, aprendizaje digital, metodología ágil, Scrum, Ingeniería en Sistemas.

Índice

4. Diagnóstico Situacional

4.1 Descripción del Contexto de la Situación Problemática Planteada

La transformación digital ha impulsado avances significativos en la educación, proporcionando herramientas tecnológicas que mejoran el acceso y la gestión de recursos académicos. Sin embargo, en países como Venezuela, muchas instituciones enfrentan dificultades para implementar plataformas tecnológicas integradas debido a limitaciones económicas y técnicas. En América Latina, estas barreras dificultan la modernización de procesos educativos esenciales, afectando la calidad y la eficiencia de los sistemas educativos. En la Universidad Nacional Experimental de los Llanos Centrales "Rómulo Gallegos", los procesos relacionados con la evaluación, el acceso a materiales educativos y la comunicación entre estudiantes y profesores carecen de un sistema unificado, lo que genera ineficiencias y desorganización. Este proyecto propone una solución innovadora mediante el desarrollo de una aplicación móvil que centraliza y optimiza estas funciones, adaptándose a las necesidades específicas de la comunidad universitaria.

4.2 Justificación del Problema

La carencia de una herramienta de gestión educativa eficaz afecta la calidad del aprendizaje y la comunicación entre estudiantes y profesores. Al centralizar la gestión educativa en una plataforma digital, que permitirá el acceso continuo a evaluaciones, recursos educativos. Esto no solo aliviará la carga administrativa de los profesores, sino que también ofrecerá a los estudiantes una forma de evaluación continua, promoviendo un aprendizaje personalizado y mejorando su rendimiento académico.

4.3 Objetivos del Proyecto

Objetivo General: Implementar una aplicación móvil de gestión educativa que centralice y optimice procesos de enseñanza y aprendizaje en el área de Ingeniería en Sistemas.

- **Objetivos Específicos:**
    - Diagnosticar la situación actual de los procesos educativos en el área de Ingeniería en Sistemas para identificar necesidades y limitaciones.
    - Determinar los requerimientos técnicos y funcionales necesarios para implementar una aplicación móvil.
    - Diseñar el sistema de gestión educativa que centralice el acceso a evaluaciones, calificaciones y una biblioteca digital, optimizando la interacción educativa digital.
    - Evaluar el funcionamiento de la aplicación móvil, asegurando que cumpla con los estándares de accesibilidad, eficiencia y satisfacción de usuarios (docentes y estudiantes) en el monitoreo de progreso académico y gestión de recursos.

4.4 Procesos que van a Automatizar dentro del área de ingeniería informática

Los procesos automatizados por la aplicación incluyen la gestión de evaluaciones, la comunicación entre docentes y estudiantes, y la organización y acceso a materiales educativos. Estos procesos anteriormente se realizaban manualmente o a través de sistemas dispersos, lo que generaba ineficiencias y pérdida de tiempo.

5. Determinación, Instalación y Configuración de las Herramientas de Desarrollo

5.1 Plataforma de Desarrollo

Para la implementación de esta aplicación, se ha seleccionado Kotlin como el lenguaje de programación principal para el desarrollo del frontend en Android, y Nest.js para el backend. PostgreSQL es la base de datos elegida, por su capacidad para manejar grandes volúmenes de datos y su flexibilidad en la gestión de relaciones complejas. Esta combinación de herramientas permite una estructura sólida y escalable, necesaria para soportar el crecimiento de la aplicación.

5.2 Arquitectura del Sistema de Información

La arquitectura del sistema sigue un modelo de tres capas:

- **Frontend**: Desarrollado en Jetpack Compose con Kotlin, proporcionando una interfaz amigable y adaptable para los usuarios.
- **Backend**: Implementado con Nest.js, que actúa como intermediario entre el frontend y la base de datos, gestionando la lógica de negocio.
- **Base de Datos**: PostgreSQL, alojada en los servidores de la UNERG, asegurando que los datos permanezcan dentro de la infraestructura de la universidad.

5.3 Selección del Entorno del Sistema de Información

El sistema será desplegado en los servidores de la Universidad Nacional Experimental de los Llanos Centrales "Rómulo Gallegos" (UNERG), lo que garantiza que la aplicación cumpla con los requisitos de seguridad y accesibilidad de la institución.

5.4 Metodología para el Desarrollo del Sistema de Información

Se ha seleccionado la metodología Scrum, una metodología ágil que facilita el desarrollo iterativo y adaptativo del sistema. Esto permite realizar sprints cortos, en los cuales se recogen y analizan comentarios de los usuarios para hacer mejoras continuas. Esta metodología se adapta al contexto de desarrollo del proyecto, asegurando que las funcionalidades puedan adaptarse a las necesidades de estudiantes y profesores.

6. Desarrollo de la Aplicación

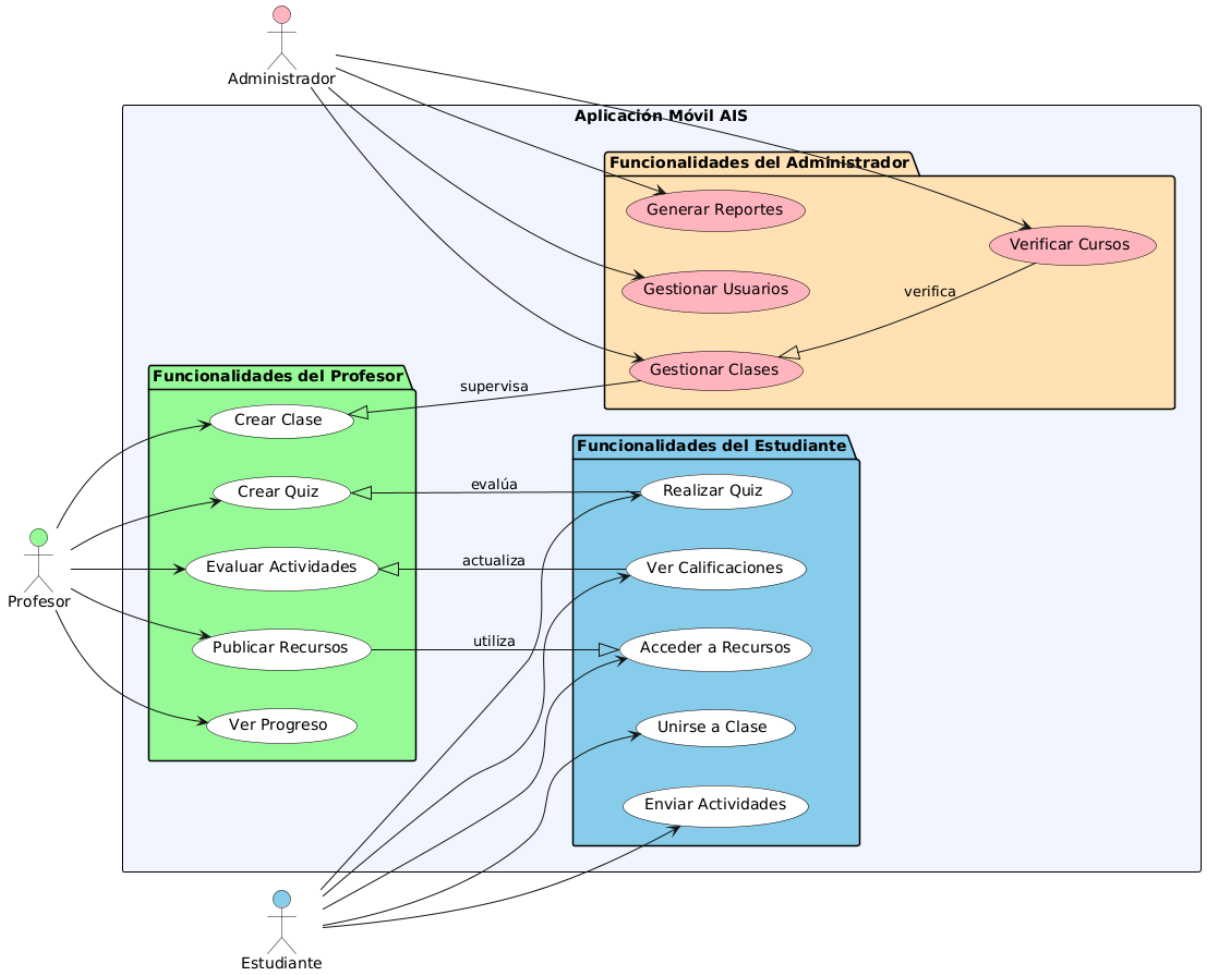
6.1 Fase de Planificación

- **6.1.1 Descripción:** En esta fase se definen los requerimientos funcionales y no funcionales del sistema, alineando el desarrollo con las necesidades específicas de los usuarios finales.
- **6.1.2 Requerimientos Funcionales del Proyecto:**
    - **Servidor**:
        - Servidor físico el backend y la base de datos.
        - Conexión estable a internet para garantizar la sincronización y acceso a los datos.
    - **Dispositivos para Usuarios:**
        - Teléfonos inteligentes con sistema operativo Android 8.0 (Oreo) o superior para que los estudiantes y profesores puedan interactuar con la aplicación móvil.
        - Computadoras o laptops para administradores y profesores, utilizadas en la gestión de usuarios, clases y evaluaciones.

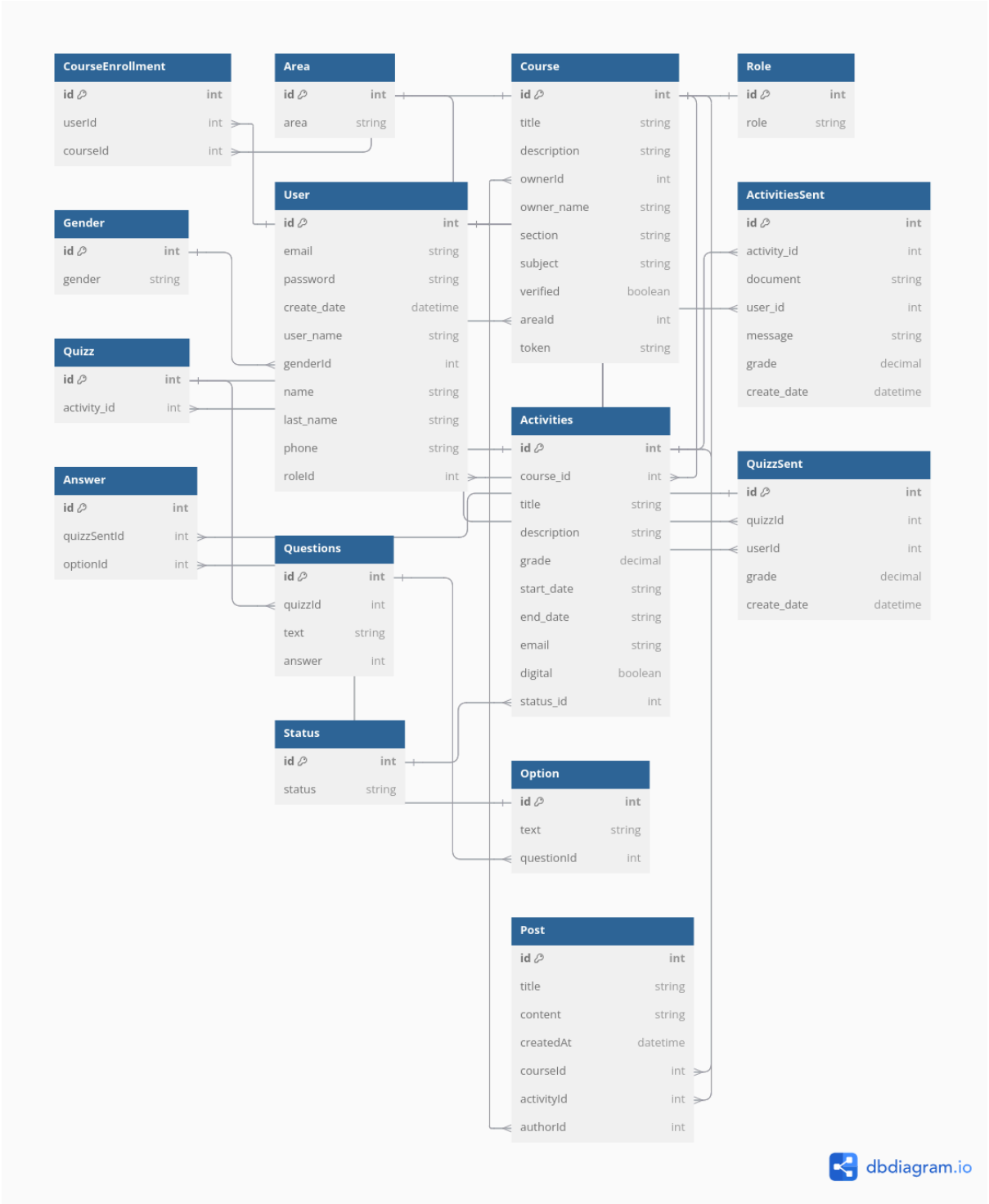
- **6.1.3 Requerimientos No Funcionales del Sistema de Información:**
    - **Acceso a Internet**:
        - Conexión a internet para estudiantes, profesores y administradores, que permita sincronizar datos y realizar acciones oportunamente dentro del sistema.
    - **Compatibilidad**:
        - Dispositivos móviles y navegadores actualizados para garantizar un funcionamiento óptimo.
    - **Facilidad de Uso**:
        - Una interfaz intuitiva que no requiera experiencia técnica avanzada para navegar y utilizar el sistema.
- **6.1.4 Restricciones:**
    - Dependencia de conexión a Internet para el uso completo de la aplicación.
    - Limitación en dispositivos que no cumplen con los requerimientos mínimos de Android para el funcionamiento óptimo de la aplicación.

6.2 Fase de Diseño

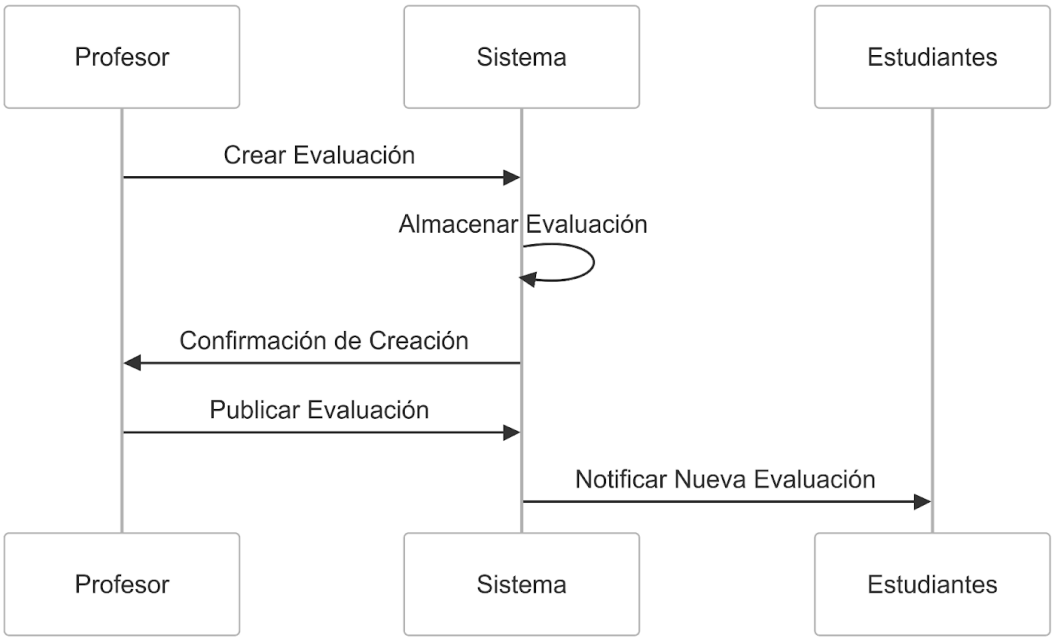
- **6.2.1 Diagrama de caso de uso:**



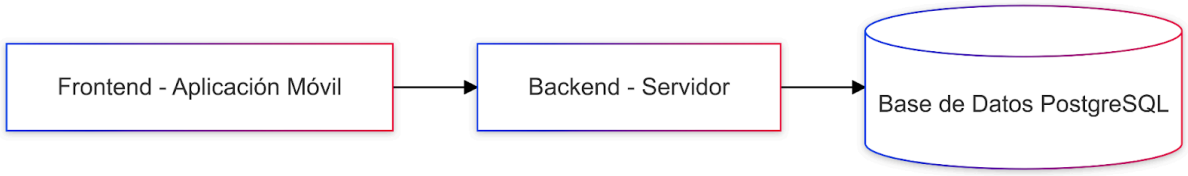
- **6.2.2 Diagrama de Entidad Relación:**


CourseEnrollment
id int
userId int
courseId int
Area
id int
area string
Course
id int
title string
description string
ownerId int
owner_name string
section string
subject string
verified boolean
areaId int
token string
Role
id int
role string
User
id int
email string
password string
create_date datetime
user_name string
genderId int
name string
last_name string
phone string
roleId int
Gender
id int
gender string
ActivitiesSent
id int
activity_id int
document string
user_id int
message string
grade decimal
create_date datetime
Quizz
id int
activity_id int
Activities
id int
course_id int
title string
description string
grade decimal
start_date string
end_date string
email string
digital boolean
status_id int
QuizzSent
id int
quizzId int
userId int
grade decimal
create_date datetime
Answer
id int
quizzSentId int
optionId int
Questions
id int
quizzId int
text string
answer int
Status
id int
status string
Option
id int
text string
questionId int
Post
id int
title string
content string
createdAt datetime
courseId int
activityId int
authorId int
dbdiagram.io

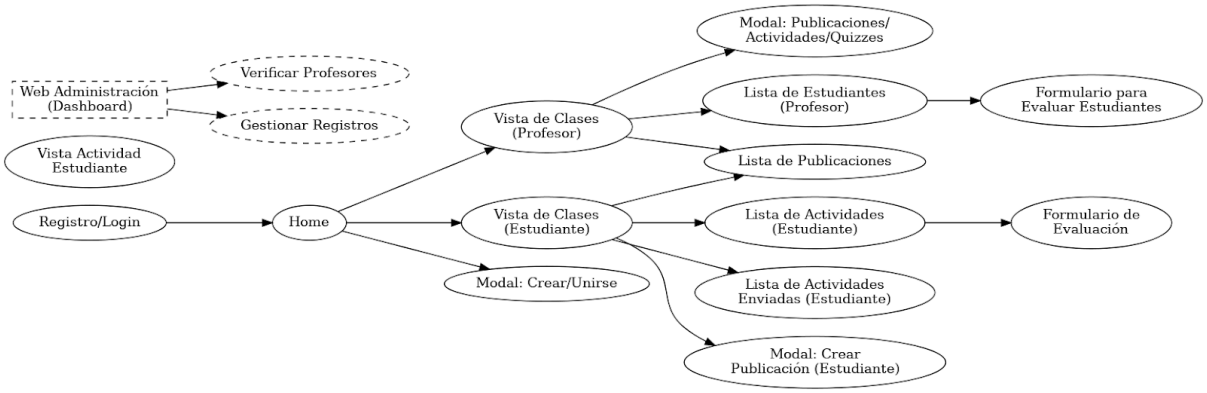
- **6.2.3 Diagrama de Secuencia:**



- **6.2.4 Diagrama Arquitectónico:**



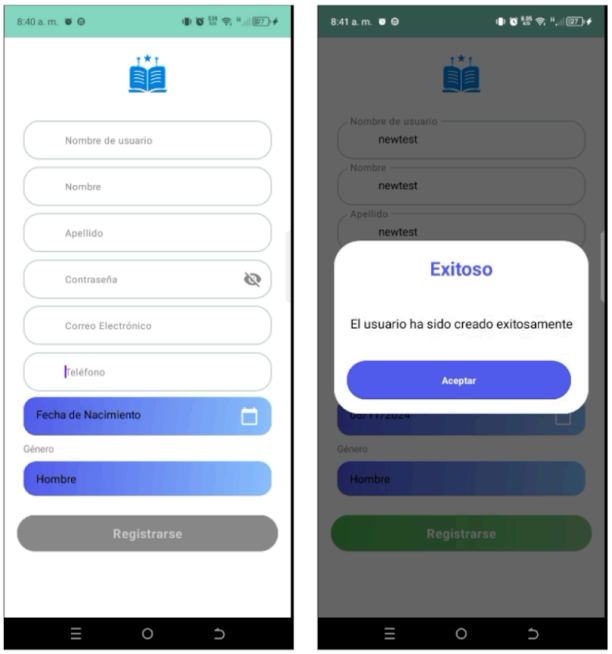
- **6.2.5 Diagrama de Navegación:**



- **6.2.6 Diseño Bocetos del Sistema:**

  1. Vista de Registro

- **Propósito**: Permitir que nuevos usuarios se registren en la aplicación.
- **Flujo**:
    1. El usuario completa los campos y pulsa "Registrar".
    2. Tras el registro exitoso, el usuario es redirigido automáticamente al login.

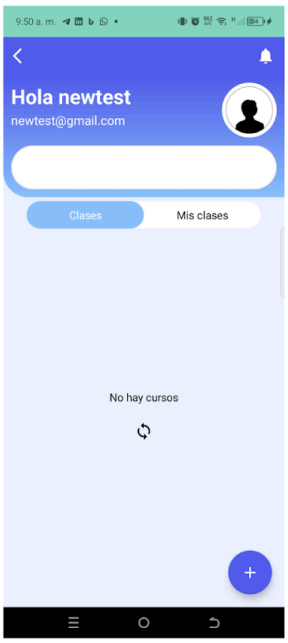


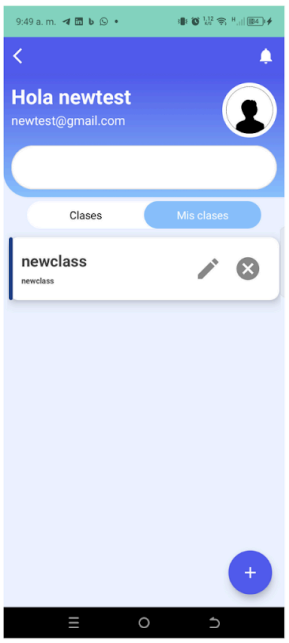
2. Vista de Login

- **Propósito**: Permitir que los usuarios registrados inicien sesión.
- **Flujo**:
    1. El usuario ingresa las credenciales.
    2. Si son inválidas, se muestra un mensaje de error.

3. Home

- **Propósito**: Punto central para el usuario tras iniciar sesión, desde donde puede acceder a las clases creadas o unidas y realizar acciones principales.
- **Elementos Principales**:
    - **Lista de Clases Creadas**:
        - Muestra las clases que el usuario ha creado.
        - Cada clase incluye información como el nombre, código y opciones para gestionar la clase.
    - **Lista de Clases Unidas**:
        - Muestra las clases a las que el usuario se ha unido como estudiante.
        - Cada clase incluye el nombre, profesor asignado y el progreso.
    - **Botón para Abrir el Modal de Opciones**:
        - Al pulsarlo, abre un modal con las opciones:
            - **Crear Clase**: Lleva al formulario de creación de clase.
            - **Unirse a Clase**: Solicita un código de clase para unirse.
- **Conexiones**:
    - Acceso directo a la Vista de Clases, según el rol del usuario.

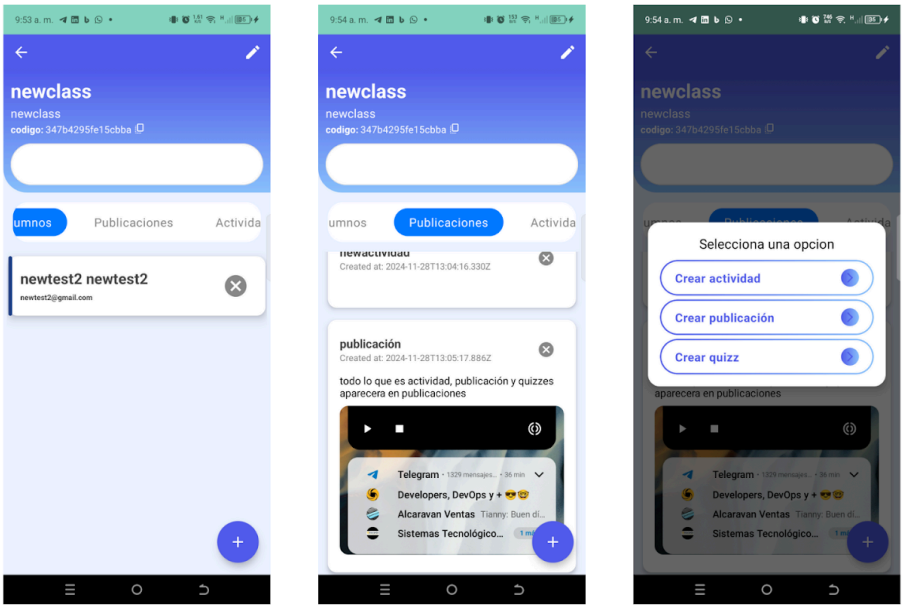




4. Vista de Clases (Profesor)

- **Propósito**: Espacio para que los profesores gestionen las clases que han creado.
- **Elementos Principales**:
    - **Lista de Estudiantes**:
        - Muestra los estudiantes inscritos en la clase.
        - Cada estudiante tiene un botón para acceder al formulario de evaluación.
    - **Lista de Publicaciones**:
        - Muestra todas las publicaciones de la clase, ordenadas por fecha.
        - Los profesores pueden eliminar o editar publicaciones existentes.
    - **Lista de Actividades**:

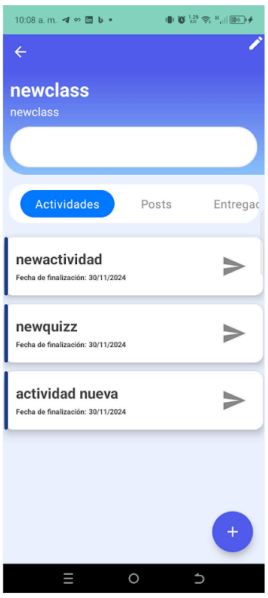
        - Incluye las actividades creadas por el profesor, como tareas o quizzes.
        - Muestra el estado de cada actividad (en curso, evaluada, vencida).
    - **Botón para Abrir el Modal de Gestión**:
        - Abre un modal con las siguientes opciones:
            - **Crear Actividad**: Lleva al formulario para definir una nueva tarea o proyecto.
            - **Crear Publicación**: Permite añadir recursos educativos en el feed de la clase.
            - **Crear Quiz**: Genera un examen de selección simple.



5. Vista de Clases (Estudiante)

- **Propósito**: Espacio para que los estudiantes interactúen con las actividades y recursos de la clase.
- **Elementos Principales**:
    - **Lista de Actividades Generales**:
        - Muestra las tareas asignadas y sus fechas límite.
        - Cada actividad tiene un botón que lleva al formulario para subir una evaluación.
    - **Lista de Publicaciones**:
        - Muestra las publicaciones del profesor, como guías y recursos educativos.
        - Los estudiantes pueden marcar publicaciones como vistas.
    - **Lista de Actividades Enviadas**:
        - Muestra las actividades que el estudiante ya envió.
        - Indica si fueron evaluadas y permite consultar las calificaciones.
    - **Botón para Abrir el Modal de Creación**:
        - Abre un modal que permite:

- **Crear Publicación**: Opción para que el estudiante comparta recursos relevantes con el grupo (si se permite en la configuración de la clase).

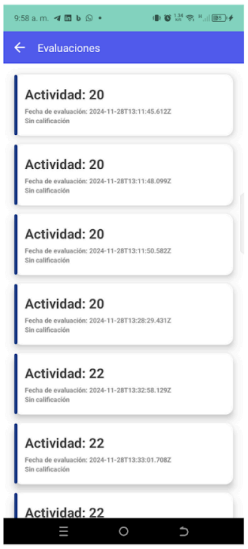


6. Formulario de Evaluación

- **Propósito**: Espacio central para que estudiantes y profesores gestionen las actividades evaluativas.
- **Funciones Principales**:
  - **Estudiantes**:
    - Subir tareas en formato digital (archivos, texto, multimedia).
    - Adjuntar notas o comentarios adicionales.
    - Ver un resumen de las actividades enviadas.
  - **Profesores**:
    - Consultar las actividades enviadas por cada estudiante.
    - Evaluar con comentarios y asignar calificaciones.
    - Marcar actividades como aprobadas o pendientes de revisión.

ESTUDIANTE

PROFESOR



6.3 Fase de Codificación (Programación)

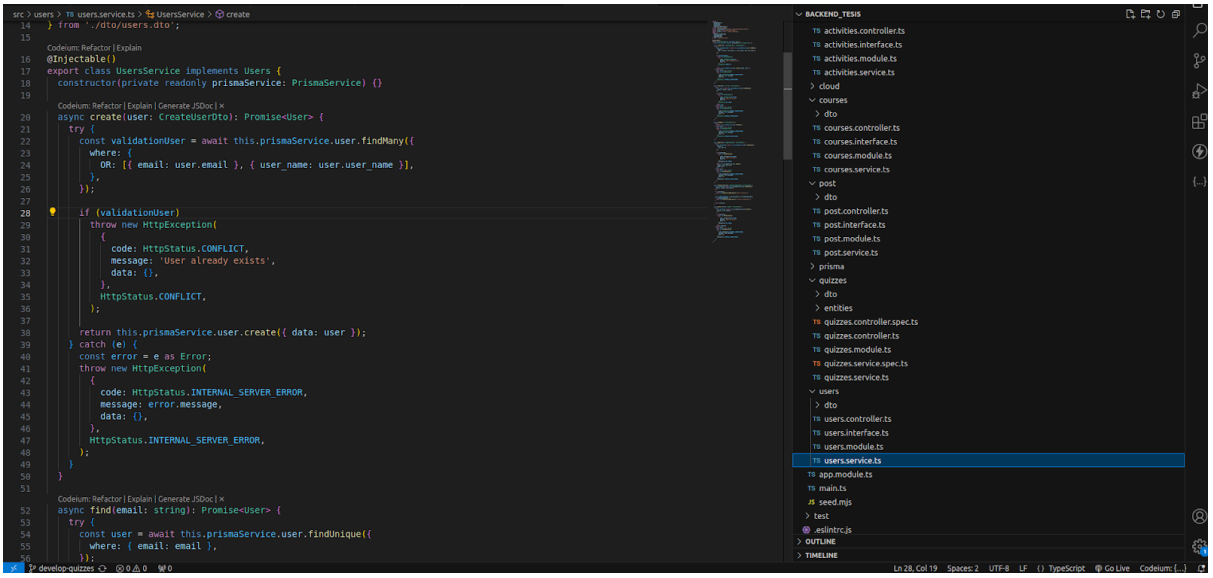
6.3.1 Requerimientos de Desarrollo

Para garantizar la correcta implementación del sistema, se han identificado los siguientes requerimientos de desarrollo:

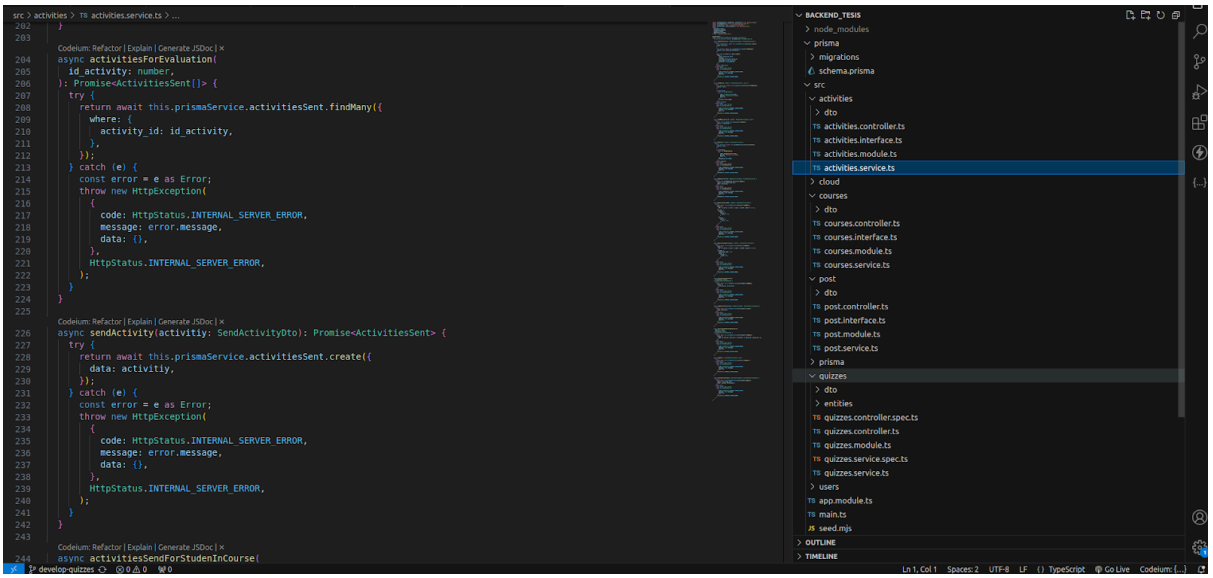
- **Ambiente de Desarrollo:**
  - **Frontend Móvil:**
    - Android Studio con Kotlin: Para el desarrollo de la interfaz móvil con Jetpack Compose.
  - **Backend:**
    - Nest.js: Framework para el desarrollo del backend en TypeScript.
    - PostgreSQL: Base de datos relacional para almacenar información del sistema.
  - **Web Administrativa:**
    - Next.js: Para la interfaz web destinada a los administradores.
- **Configuración de Dependencias:**
  - **Instalación y configuración de librerías para Nest.js:**
    - Prisma: ORM para la gestión de la base de datos.
    - Class-validator: Validación de datos en el backend.
  - **Configuración de dependencias para Kotlin:**
    - Retrofit para la comunicación con el backend.
    - Room para almacenamiento local en el dispositivo.
  - **Integración de herramientas:**
    - Configurar ruta estatica para el almacenamiento de archivos.
    - Docker para despliegue y pruebas locales.

6.3.2 Desarrollo de los Módulos del Sistema

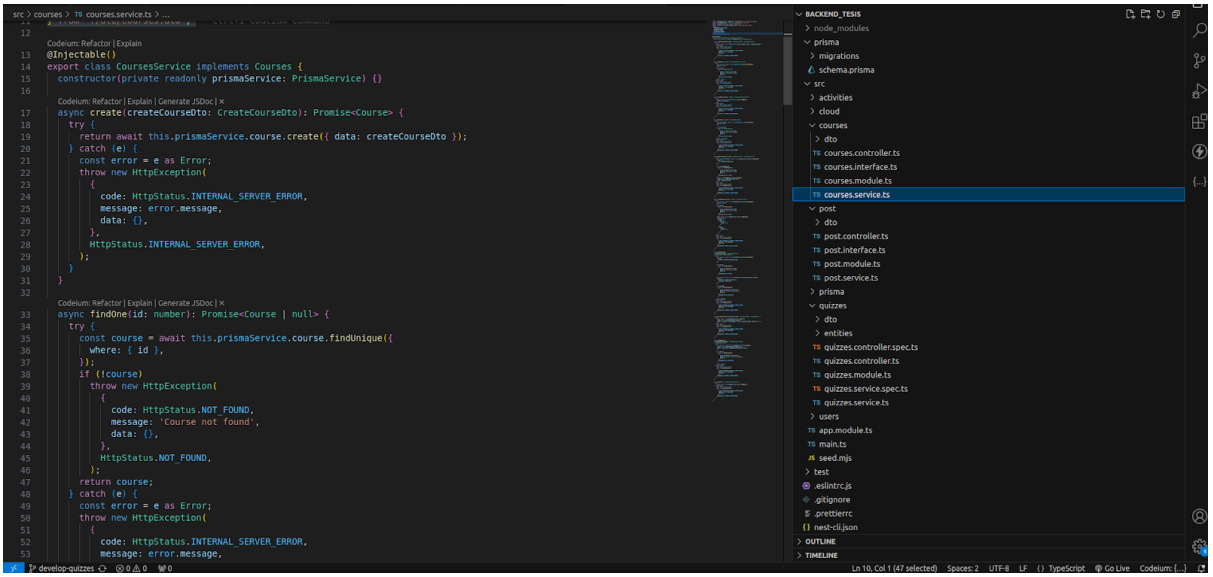
1. **Módulo de Administración:**
    - **Propósito:** Permitir al administrador gestionar usuarios, roles y clases.
    - **Funcionalidades:**
        - **Asignar roles a los profesores (verificar su vínculo con la universidad).**
        - **Gestión de usuarios:**
            - Crear, editar, bloquear o eliminar usuarios.
        - **Gestión de clases:**
            - Revisar las clases creadas por los profesores.
            - Eliminar clases que incumplan las normas.
    - **Interacción:**
        - Acceso desde un dashboard web.
        - Uso de formularios y tablas para la gestión.



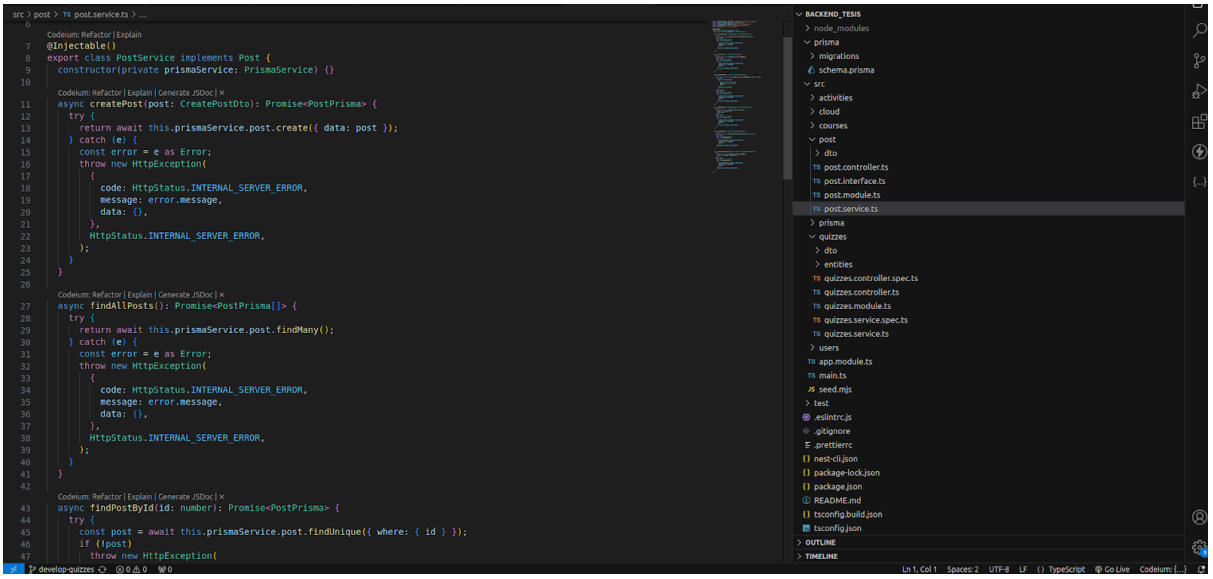
2. **Módulo de Gestión de Evaluaciones:**
    - **Propósito:** Facilitar la creación y gestión de evaluaciones y calificaciones.
    - **Funcionalidades:**
        - Crear evaluaciones, incluyendo tareas y quizzes.
        - Asignar calificaciones a las actividades enviadas por los estudiantes.
        - Generar reportes de calificaciones y progreso por clase o estudiante.
    - **Interacción:**
        - Los profesores acceden desde la Vista de Clases.
        - Los estudiantes reciben notificaciones de evaluaciones asignadas.



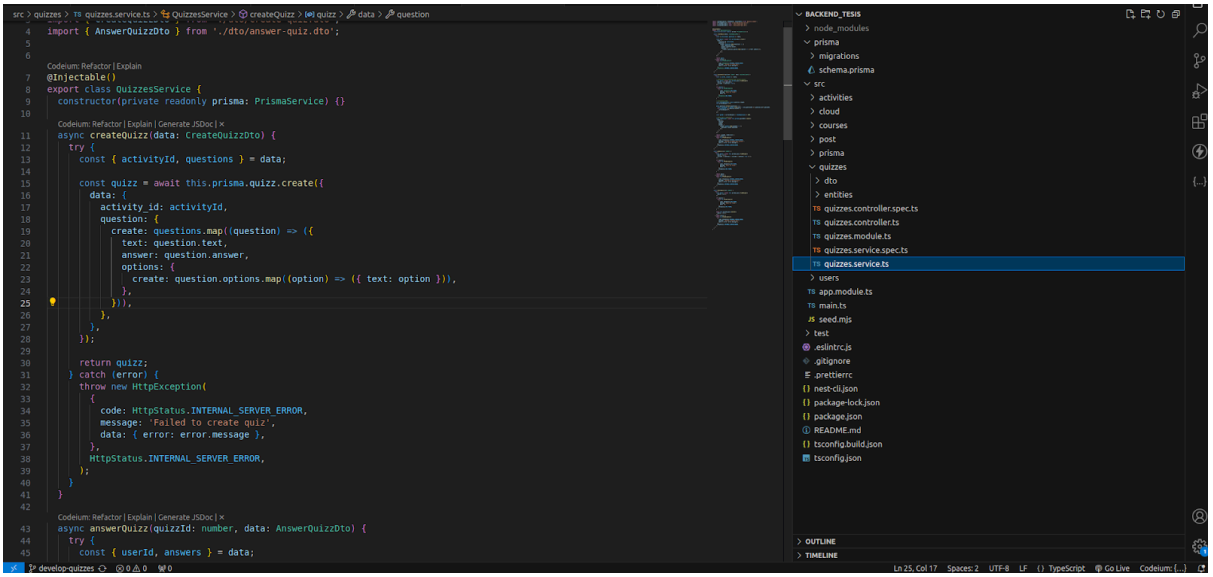
3. **Módulo de Clases:**
    - **Propósito:** Centralizar la creación, gestión y acceso a las clases.
    - **Funcionalidades:**
        - Crear clases con detalles como nombre, descripción y código único.
        - Unirse a clases mediante un código compartido.
        - **Gestión de los inscritos:**
            - Visualización de estudiantes inscritos.
            - Expulsión de estudiantes por parte del profesor.
    - **Interacción:**
        - Profesores y estudiantes acceden a través del Home.



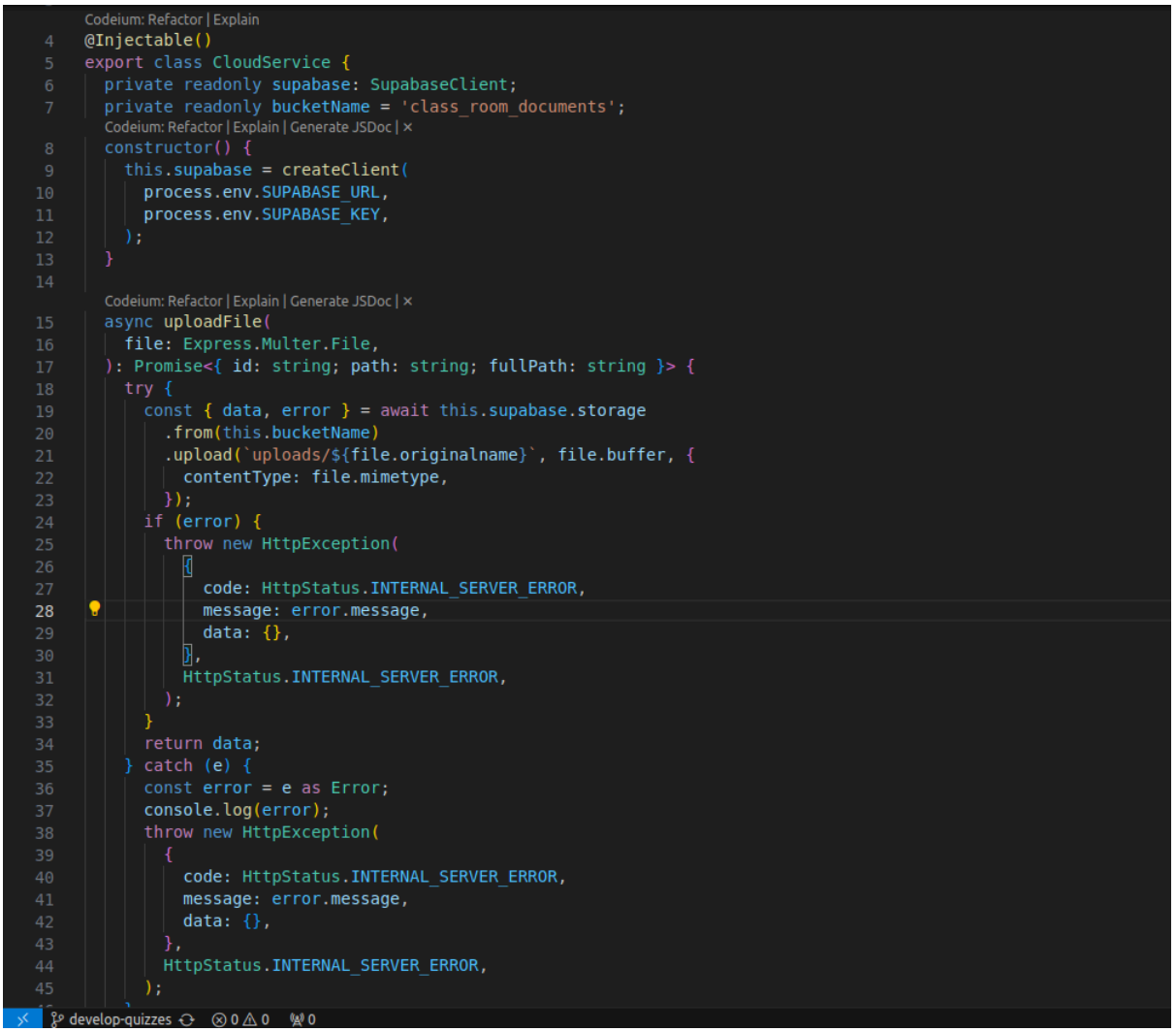
4. **Módulo de Publicaciones:**
    - **Propósito:** Permitir la publicación y acceso a recursos educativos dentro de las clases.
    - **Funcionalidades:**
        - Crear publicaciones con texto, imágenes o enlaces a recursos externos.
        - Editar y eliminar publicaciones.
        - Los estudiantes pueden marcar publicaciones como vistas.
    - Interacción:
        - Publicaciones visibles en la lista de publicaciones dentro de cada clase.



5. **Módulo de Exámenes de Selección Simple:**
    - **Propósito:** Crear quizzes y exámenes tipo test para los estudiantes.
    - Funcionalidades:
        - Configurar preguntas con opciones múltiples y una respuesta correcta.
        - Establecer fechas límite para completar los exámenes.
        - Calificar automáticamente las respuestas y registrar las puntuaciones.
    - **Interacción:**
        - Los estudiantes acceden al examen desde la Vista de Clases y ven los resultados tras completarlo.



6. **Módulo de Gestión de Archivos:**
    - **Propósito:** Facilitar el manejo de archivos subidos por los usuarios.
    - **Funcionalidades:**
        - Subida de archivos por estudiantes (tareas, proyectos).
        - Descarga de archivos por profesores para revisión.



```
Codeium: Refactor | Explain
@Injectable()
export class CloudService {
  private readonly supabase: SupabaseClient;
  private readonly bucketName = 'class_room_documents';
  Codeium: Refactor | Explain | Generate JSDoc | ×
  constructor() {
    this.supabase = createClient(
      process.env.SUPABASE_URL,
      process.env.SUPABASE_KEY,
    );
  }

  Codeium: Refactor | Explain | Generate JSDoc | ×
  async uploadFile(
    file: Express.Multer.File,
  ): Promise<{ id: string; path: string; fullPath: string }> {
    try {
      const { data, error } = await this.supabase.storage
        .from(this.bucketName)
        .upload(`uploads/${file.originalname}`, file.buffer, {
          contentType: file.mimetype,
        });
      if (error) {
        throw new HttpException(
          {
            code: HttpStatus.INTERNAL_SERVER_ERROR,
            message: error.message,
            data: {},
          },
          HttpStatus.INTERNAL_SERVER_ERROR,
        );
      }
      return data;
    } catch (e) {
      const error = e as Error;
      console.log(error);
      throw new HttpException(
        {
          code: HttpStatus.INTERNAL_SERVER_ERROR,
          message: error.message,
          data: {},
        },
        HttpStatus.INTERNAL_SERVER_ERROR,
      );
```

develop-quizzes ⊗ 0 ⚠ 0 0

7. Fase de Pruebas

- **7.1 Elaboración y Ejecución del Plan de Pruebas:** Definir el plan de pruebas incluyendo pruebas unitarias, de integración y de aceptación para verificar la correcta funcionalidad de las evaluaciones, las asignaciones y los exámenes de selección simple.
- **7.2 Análisis de Resultados:** Evaluar los resultados de las pruebas para garantizar que el sistema cumpla con los requerimientos definidos.

Conclusiones

El desarrollo de la aplicación móvil de gestión educativa "AIS" aborda de manera efectiva los desafíos asociados con la fragmentación y desorganización en los procesos educativos del área de Ingeniería en Sistemas. Este proyecto proporciona una solución tecnológica que centraliza y automatiza actividades clave, mejorando la accesibilidad a los recursos académicos y optimizando la gestión educativa.

Los principales logros del proyecto incluyen:

1. **Centralización de procesos académicos:** La aplicación unifica en un solo sistema funciones esenciales como la gestión de evaluaciones, calificaciones y acceso a recursos educativos, reduciendo la dependencia de herramientas dispersas o métodos manuales.
2. **Diseño y arquitectura tecnológica robusta:** Utilizando Kotlin para el desarrollo del frontend, Nest.js para la lógica del backend y PostgreSQL como base de datos, se asegura un sistema escalable, eficiente y adaptable a futuros requerimientos.
3. **Enfoque ágil en el desarrollo:** La metodología Scrum permitió un proceso iterativo que incorporó retroalimentación continua, asegurando que la solución estuviera alineada con las necesidades específicas de estudiantes y profesores.
4. **Impacto en la gestión educativa:** La herramienta simplifica tareas administrativas para los docentes, al mismo tiempo que mejora el acceso a información relevante para los estudiantes, fomentando una experiencia educativa más organizada y efectiva.

En conclusión, el proyecto "AIS" representa un avance importante en la gestión educativa universitaria, proporcionando una plataforma eficiente, accesible y diseñada específicamente para el contexto local. Además, establece una base sólida para futuras mejoras e implementación en otras áreas académicas.

Recomendaciones

1. **Implementación Progresiva**
   Se recomienda realizar la implementación de la aplicación en fases, comenzando con un programa piloto en una facultad o área específica. Esto permitirá identificar problemas o necesidades adicionales antes de un despliegue a mayor escala, minimizando el impacto en los usuarios.
2. **Mitigación de dependencia de Internet:**
   Implementar un modo offline que permita a los estudiantes descargar materiales y enviar evaluaciones sin conexión, sincronizando automáticamente al restablecerla.
3. **Capacitación de Usuarios**
   Es fundamental organizar talleres y sesiones de capacitación dirigidos a estudiantes, profesores y administradores. Esto garantizará que los usuarios comprendan las funcionalidades de la aplicación y se sientan cómodos utilizándola, maximizando así su adopción.
4. **Mantenimiento y Actualización**
   Se debe establecer un plan de mantenimiento continuo que contemple la revisión periódica de los servidores, la actualización de las librerías utilizadas y la corrección de errores identificados. Además, se sugiere incorporar nuevas funcionalidades basadas en la retroalimentación de los usuarios.
5. **Escalabilidad del Sistema**
   Es importante diseñar la aplicación teniendo en cuenta la posibilidad de su expansión hacia otras áreas académicas o universidades. Esto podría incluir mejoras en la infraestructura y la adición de módulos específicos para otras disciplinas.
6. **Integración con Otros Sistemas**
   Se recomienda evaluar la integración de la aplicación con plataformas educativas existentes, como sistemas de gestión del aprendizaje (LMS) y bibliotecas digitales. Esta interoperabilidad podría aumentar el valor de la herramienta y optimizar los procesos educativos.
7. **Optimización de Infraestructura**
   La Universidad debe considerar invertir en una infraestructura de servidores más robusta que permita soportar aplicaciones con funcionalidades en tiempo real, como el sistema de chats. Una infraestructura más adecuada garantizará una comunicación fluida y una mejor experiencia de usuario, eliminando la fragmentación actual en la comunicación educativa.
8. **Evaluación de Impacto**
   Realizar encuestas y análisis periódicos para evaluar el impacto de la herramienta en la calidad educativa y la satisfacción de los usuarios. Esto permitirá ajustar la estrategia de implementación y mejorar continuamente la aplicación.

Referencias Bibliográficas

1. **Agile Alliance**
   "Guía sobre metodologías ágiles en la gestión de proyectos," 2021. [En línea]. Disponible en: https://www.agilealliance.org.
   [Accedido: 15-oct-2024].
2. **Android Open Source Project**
   "Android Jetpack Compose: Documentación oficial," 2023. [En línea]. Disponible en: https://developer.android.com/jetpack/compose.
   [Accedido: 20-nov-2024].
3. **Node.js Foundation**
   "Documentación oficial de Node.js," 2023. [En línea]. Disponible en: https://nodejs.org/es/docs/.
   [Accedido: 10-nov-2024].
4. **Prisma**
   "Prisma: Modern database access for Node.js & TypeScript," 2023. [En línea]. Disponible en: https://www.prisma.io/docs/.
   [Accedido: 5-nov-2024].
5. **Docker Inc.**
   "Documentación oficial de Docker: Desarrollo y despliegue de aplicaciones," 2023. [En línea]. Disponible en: https://docs.docker.com.
   [Accedido: 12-dic-2024].
6. **Kotlin Academy** (Canal de YouTube)
   "Jetpack Compose for Beginners – Complete Tutorial," 2023. Disponible en: https://www.youtube.com/watch?v=_5qLSzoR_-w.
   [Accedido: 25-nov-2024].
7. **Scrum Guide Official** (Canal de YouTube)
   "Understanding Scrum in 15 Minutes," 2021. Disponible en: https://www.youtube.com/watch?v=9TycLR0TqFA.
   [Accedido: 30-oct-2024].
8. **PostgreSQL Academy** (Canal de YouTube)
   "Scaling PostgreSQL for Beginners," 2022. Disponible en: https://www.youtube.com/watch?v=aEK88ZM2Uho.
   [Accedido: 15-nov-2024].
9. **NestJS Documentation** (Canal de YouTube)
   "Building APIs with NestJS – Step by Step," 2023. Disponible en: https://www.youtube.com/watch?v=TxGQZ-UGKMo.
   [Accedido: 18-dic-2024].
10. **Docker Official** (Canal de YouTube)
    "Docker Essentials for Developers," 2023. Disponible en: https://www.youtube.com/watch?v=zJ6WbK9zFpI.
    [Accedido: 22-nov-2024].
11. **Venezuelan Education Initiative** (Canal de YouTube)
    "Digital Transformation in Venezuelan Education," 2024. Disponible en: https://www.youtube.com/watch?v=xyz-transformacion.
    [Accedido: 30-oct-2024].
12. **Agile Coach** (Canal de YouTube)
    "Scrum in Real Projects," 2022. Disponible en:

https://www.youtube.com/watch?v=real-scrum-101.
[Accedido: 15-oct-2024].
13. **Code with Andrea** (Canal de YouTube)
"Kotlin Best Practices for Modern Mobile Apps," 2023. Disponible en:
https://www.youtube.com/watch?v=kotlin-best-practices.
[Accedido: 12-nov-2024].