

“Año de la recuperación y consolidación de la economía peruana”

Universidad Tecnológica de los Andes



Asignatura:
Programación móvil

Profesor:
Mg. POCCORI UMERES GODOFREDO

- Alumnos:
- Castro Muñoz, Vil Mijael
 - Clarck Gervassi Bravo Paucar
 - Jafet Antony Anaya Estrada
 - Paul Andre Huarac Yaranga
 - Daniel Huisa Ramos

ÍNDICE

1.	Introducción.....	3
1.1	Definición	4
1.2	Objetivos.....	5
1.3	Funcionalidades del Proyecto.....	6
2.	Calendario del Proyecto.....	8
3.	Infraestructura.....	10
3.1	Elementos de hardware	11
3.2	Elementos de software.....	11
4.	Lenguajes Incluidos	12
5.	Marco Teórico y Antecedentes.....	13
5.1	Requerimientos Funcionales.....	15
5.2	Requerimientos No Funcionales.....	17
5.3	Casos de Uso	19
5.4	Diagrama de Clases	23
5.5	Diagrama de Objetos.....	27
5.6	Diagrama de Secuencia	28
5.7	Diagrama de Componentes	29
5.8	Diseño Técnico	30
-	Arquitectura del Servicio.....	30
-	Capa de Presentación.....	31
-	Arquitectura de Base de Datos	32
-	Prototipo	33
	Bibliografía	34

1. Introducción

La importancia de las nuevas tecnologías en la optimización de la tutoría y la orientación escolar en la actualidad. Dentro del marco de la transformación digital concerniente a la esfera educativa contemporánea se plantea la necesidad de implementar las fuentes tecnológicas que permitan incrementar la interacción entre los agentes directamente involucrados en el proceso de enseñanza-aprendizaje. Por ende, dados los desafíos que el Colegio Inca Garcilaso de la Vega, como institución escolar, enfrenta a la hora de administración eficiente de la tutoría y comunicación debida entre los tutores y los estudiantes, requiere una solución adecuadamente e integrada.

El presente proyecto contempla el diseño y desarrollo de una aplicación móvil, concebida como una herramienta integral que permite no solo la programación de sesiones de tutoría, sino también la administración de materiales pedagógicos, el envío de notificaciones y la evaluación sistemática del desempeño de los tutores. . Tales funcionalidades se orientan a robustecer el andamiaje académico, optimizando la experiencia educativa mediante un entorno digital accesible y eficiente.

En el presente informe se describe con seriedad el análisis, diseño y planificación en torno al aplicativo, estableciendo con exactitud los objetivos y funcionalidades del aplicativo, además de los recursos tecnológicos necesarios para su implementación. Este documento busca ser un aporte al debate en torno a la introducción de las tecnologías móviles en el sistema educativo, al mismo tiempo que se constituye en una proposición concreta para solventar los problemas del Colegio Inca Garcilaso de la Vega.

1.1 Definición

El proyecto **Tutorías Móviles** surge como una respuesta a la creciente necesidad de integrar tecnologías digitales en los procesos educativos, particularmente en el contexto de la educación secundaria. Según diversos estudios, la implementación de herramientas tecnológicas no solo facilita la organización de actividades académicas, sino que también promueve una interacción más dinámica y efectiva entre estudiantes y docentes, favoreciendo el aprendizaje significativo en entornos digitales.

Esta aplicación móvil tiene como propósito principal fortalecer la comunicación entre tutores y estudiantes del Colegio Inca Garcilaso de la Vega, a través de funcionalidades que permitan gestionar tutorías de manera eficiente. En este sentido, se busca ofrecer una solución que abarque la programación de sesiones, el intercambio de materiales educativos, el envío de notificaciones oportunas y la evaluación del desempeño de los tutores, todo ello dentro de una plataforma accesible desde dispositivos móviles.

La relevancia de este proyecto radica en su capacidad para abordar problemáticas concretas, tales como la falta de sincronización en la agenda de tutorías, la limitada disponibilidad de recursos pedagógicos en formato digital y la ausencia de mecanismos estructurados para evaluar la calidad de las tutorías. Asimismo, la propuesta se alinea con los principios de innovación educativa, que destacan la importancia de utilizar tecnologías móviles como herramientas para personalizar y optimizar los procesos de aprendizaje (Laudon & Laudon, 2016).

De este modo, **Tutorías Móviles** no solo se presenta como un proyecto tecnológico, sino como una iniciativa que contribuye a la transformación educativa del Colegio Inca Garcilaso de la Vega, al promover la integración de metodologías digitales adaptadas a las demandas del siglo XXI.

1.2 Objetivos

Objetivo General.

Facilitar el acceso a tutorías personalizadas y recursos educativos desde dispositivos móviles, optimizando la comunicación entre tutores y estudiantes, y contribuyendo al fortalecimiento del aprendizaje en el Colegio Inca Garcilaso de la Vega.

- Desarrollar un sistema de programación de tutorías: Implementar una funcionalidad que permita a tutores y estudiantes coordinar sesiones de manera ágil, visualizando horarios disponibles y estableciendo recordatorios automáticos.
- Implementar un módulo de gestión de materiales educativos: Diseñar una herramienta que permita el intercambio de recursos pedagógicos en formatos digitales, optimizando la disponibilidad y accesibilidad de contenidos.

- Desarrollar un sistema de notificaciones automatizadas: Integrar una funcionalidad que envíe alertas sobre tutorías programadas, nuevos materiales subidos y cambios en la agenda académica.
- Diseñar un sistema de evaluación de tutores: Incorporar un mecanismo que permita a los estudiantes calificar y brindar retroalimentación sobre las tutorías recibidas, con el objetivo de garantizar la calidad del servicio educativo.
- Fomentar la adopción de tecnologías móviles en el ámbito educativo: Promover el uso de dispositivos digitales como herramientas para mejorar la organización, comunicación y acceso a recursos en el entorno escolar.

1.3 Funcionalidades del Proyecto

El proyecto, como se ha descrito anteriormente, consiste en una aplicación móvil para facilitar la comunicación y organización entre tutores y estudiantes. La app estará diseñada para responder a las necesidades educativas del Colegio Inca Garcilaso de la Vega y contará con las siguientes funcionalidades:

- Registro y acceso: Los usuarios, ya sean estudiantes o tutores, podrán registrarse y acceder a la plataforma mediante su correo electrónico y una contraseña.

- Programación de tutorías: La aplicación permitirá a los estudiantes reservar sesiones con sus tutores, seleccionando horarios disponibles de forma sencilla y práctica.
- Gestión de materiales educativos: Los tutores podrán subir recursos pedagógicos en formatos digitales, los cuales estarán disponibles para que los estudiantes los descarguen y consulten.
- Notificaciones automáticas: Los usuarios recibirán alertas sobre tutorías próximas, nuevos materiales disponibles o cambios en la programación.
- Evaluación de tutorías: Los estudiantes podrán calificar las sesiones de tutoría y dejar comentarios sobre su experiencia, ayudando a mejorar el servicio.
- Panel de administración: Los tutores contarán con una herramienta para gestionar sus horarios, materiales subidos y las evaluaciones recibidas.
- Diseño intuitivo y responsivo: La aplicación estará optimizada para dispositivos móviles, ofreciendo una experiencia amigable y accesible.
- Historial de actividades: Los usuarios podrán consultar un registro de sus tutorías pasadas y materiales descargados.

- Comunicación eficiente: Se incluirá un sistema de mensajería interna para que tutores y estudiantes puedan coordinar detalles sobre las tutorías directamente desde la aplicación.

2. Calendario del Proyecto

El desarrollo de la aplicación móvil Tutorías Móviles se organizará en cinco fases clave, con un tiempo estimado para cada una, comenzando desde la semana pasada y ajustado para cumplir con el plazo de dos meses.

- Fase 1: Planificación y Análisis de Requisitos

Duración: 1 semana

Fechas: Semana 1 (del 1 al 7 de enero)

- Recolección de requisitos funcionales y no funcionales.
- Definición del alcance y objetivos del proyecto.
- Establecimiento del plan de trabajo y cronograma.
- Entregable: Documento con el plan de trabajo y requisitos del proyecto.

- Fase 2: Diseño y Arquitectura

Duración: 2 semanas

Fechas: Semana 2 y 3 (del 8 al 21 de enero)

- Diseño de la arquitectura técnica (base de datos, servidores, etc.).
 - Creación de prototipos de la interfaz de usuario.
 - Revisión y validación de los prototipos con los interesados.
 - Entregable: Documento de diseño técnico y prototipos de la interfaz.
- Fase 3: Desarrollo e Implementación del Código

Duración: 4 semanas

Fechas: Semana 4 a 7 (del 22 de enero al 18 de febrero)

 - Desarrollo de las funcionalidades principales de la aplicación: registro de usuarios, programación de tutorías, intercambio de materiales, etc.
 - Integración de la base de datos y sistemas de notificación.
 - Desarrollo de la funcionalidad de evaluación de tutorías y mensajería interna.
 - Entregable: Versión beta de la aplicación con las funcionalidades principales implementadas.
- Fase 4: Pruebas y Ajustes

Duración: 2 semanas

Fechas: Semana 8 y 9 (del 19 de febrero al 4 de marzo)

 - Pruebas unitarias, de integración y de usabilidad.
 - Identificación y corrección de errores.
 - Optimización del rendimiento de la aplicación.
 - Realización de pruebas de aceptación con un grupo piloto de usuarios.

- Entregable: Informe de pruebas y versión final de la aplicación lista para la presentación.
- Fase 5: Preparación para la Presentación Final
 - Duración: 1 semana
 - Fechas: Semana 10 (del 5 al 11 de marzo)
 - Elaboración de la presentación en PowerPoint (PPT).
 - Preparación del material de apoyo para la presentación final (documentación, video demostrativo, etc.).
 - Ensayo de la presentación. Entregable: Presentación final del proyecto, lista para su entrega y exposición.

3. Infraestructura

Para el proyecto se usaran distintos recursos, una parte con infraestructura propia y otra usando el framework de desarrollo móvil flutter y en un entorno de desarrollo que es el visual.

3.1 Elementos de hardware

En este sentido a lo largo del proyecto estamos usando nuestras maquinas (laptops). A continuacion lo nombramos.

Componente	Características	Tareas
Ordenador portátil hp pavilion	Procesador: Intel(R) Core(TM) i3- 2.13GHz RAM instalada: 4.0 GB (3.8 GB utilizable) Windows 10 Pro	Diseño de la aplicación. Redacción de la documentación creación del informe y presentación
Asus Rog strix g513	Procesador: AMD RYZEN 7 6800H RAM instalada:16.0 GB DDR5 4600Hz (15.6GB utilizable) Windows 11 pro	Diseño de la aplicación. Terminal para las pruebas usando el Android studio
Acer nitro rtx3650	Procesador: 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz 2.69 GHz RAM instalada: 12.0 GB (11.8 GB utilizable) Windows 10 Pro	Diseño de la aplicación. Terminal para las pruebas usando el Android studio

3.2 Elementos de software

En este caso se usarán distintas aplicaciones que se muestran en la siguiente tabla.

Herramienta	Tareas
Visual Studio Code.	Entorno eficiente y ágil para desarrollar tu aplicación móvil, permitiéndote concentrarte en la programación sin distracciones.
Android Studio	Entorno de desarrollo integrado oficial para la plataforma Android
Microsoft office 2016	Programa de ofimática para desarrollar el informe
Google chorme	Navegador web para búsqueda de información.
Google classrrom	Repositorio donde están las guías facilitadas por el docente.
WinRAR	Compresor de archivos de varios formatos.
Lucidchart	Herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML

4. Lenguajes Incluidos

El desarrollo de la aplicación móvil se lleva a cabo utilizando Dart, un lenguaje de programación moderno y eficiente, diseñado específicamente para crear aplicaciones móviles de alto rendimiento. Dart es un lenguaje orientado a objetos, fácil de aprender y optimizado para el desarrollo de aplicaciones en Flutter, el framework utilizado en este proyecto. Este lenguaje será la base del aplicativo, permitiendo la creación de interfaces dinámicas y funcionales, así como la gestión de la lógica de negocio detrás de la aplicación.

Flutter, el framework de desarrollo, proporciona un conjunto robusto de herramientas y librerías propias que facilitan la creación de interfaces gráficas, la interacción con servicios backend, y la implementación de características avanzadas como animaciones y transiciones. La integración de Dart con Flutter garantiza un alto rendimiento en dispositivos móviles, utilizando el motor gráfico Skia para renderizar interfaces rápidas y fluidas.

Además de Dart y Flutter, en el proyecto se utilizarán tecnologías adicionales que permitirán una mejor gestión de datos y la comunicación con el servidor, como JSON para el intercambio de datos entre el cliente y el servidor. Esta estructura facilitará la implementación de funcionalidades como la carga de datos en tiempo real y la interacción con APIs externas.

En resumen, el uso de Dart y Flutter en este proyecto permite aprovechar las ventajas de un desarrollo ágil y de alto rendimiento para plataformas móviles, manteniendo la flexibilidad y escalabilidad necesarias para cubrir las funcionalidades requeridas por la aplicación.

5. Marco Teórico y Antecedentes

Definición de conceptos clave:

Los sistemas de información son herramientas digitales que permiten recopilar, procesar y distribuir datos, facilitando la gestión y organización de procesos. En el contexto del

desarrollo móvil, estos sistemas se integran en aplicaciones diseñadas para ofrecer funcionalidades específicas en dispositivos portátiles como smartphones. Por otro lado, las bases de datos actúan como repositorios estructurados que almacenan y organizan información crítica para su recuperación eficiente.

Teorías y tecnologías relacionadas:

Este proyecto se fundamenta en la utilización de tecnologías modernas como el framework Flutter y el lenguaje Dart, que facilitan el desarrollo ágil y multiplataforma. Flutter permite crear interfaces gráficas responsivas y eficientes, mientras que Dart ofrece un entorno robusto y escalable para manejar la lógica del negocio. Adicionalmente, se emplean herramientas como Android Studio para pruebas y Luci chart para diagramas conceptuales.

Antecedentes:

El diseño de soluciones móviles en la educación ha mostrado beneficios significativos en proyectos previos. Por ejemplo, aplicaciones educativas desarrolladas con tecnologías similares han optimizado procesos de enseñanza y aprendizaje, mejorando la comunicación entre docentes y estudiantes. Estas experiencias destacan la relevancia de integrar tecnologías móviles en entornos académicos para afrontar desafíos de coordinación, accesibilidad y evaluación del desempeño.

5.1 Requerimientos Funcionales

Gestión de Usuarios.

- Registro y autenticación: Los estudiantes y tutores podrán registrarse en la aplicación con su nombre, usuario (correo o código) y contraseña.
- Roles definidos: Los usuarios se dividirán en dos roles: Estudiante y Tutor, con permisos y funcionalidades diferentes para cada uno.
- Autenticación segura: La aplicación debe permitir un sistema de autenticación mediante nombre de usuario y contraseña, y garantizar la seguridad de las credenciales de acceso.
- Programación de Sesiones.
- Agendar sesiones: Los tutores podrán crear, modificar y cancelar sesiones de tutoría, definiendo fechas, horas y duración.
- Visualización de horarios disponibles: Los estudiantes podrán ver los horarios disponibles de los tutores y agendar sesiones en función de estos horarios.
- Recordatorios automáticos: La aplicación enviará notificaciones automáticas a los estudiantes y tutores para recordarles las sesiones programadas.

Gestión de Materiales:

- Subir y descargar archivos: Los tutores podrán subir materiales educativos como documentos PDF, imágenes y videos. Los estudiantes podrán descargarlos para su estudio.
- Categorías de materiales: Los materiales se organizarán por tipo (ej. PDFs, videos, imágenes) y por sesión de tutoría para facilitar su búsqueda.
- Visualización de materiales compartidos: Los estudiantes podrán visualizar los materiales compartidos por los tutores y tener acceso a ellos en cualquier momento.

Evaluación de Tutores:

- Sistema de calificación: Los estudiantes podrán calificar a los tutores al final de cada sesión, evaluando aspectos como claridad, ayuda brindada, puntualidad, entre otros.
- Comentarios: Además de las calificaciones, los estudiantes podrán dejar comentarios con sugerencias o apreciaciones sobre las tutorías.
- Historial de calificaciones: Los tutores podrán consultar su historial de calificaciones y comentarios recibidos.

Notificaciones:

- Alertas para sesiones programadas: La aplicación enviará recordatorios automáticos a los usuarios (tutores y estudiantes) sobre las sesiones programadas con antelación.
- Notificaciones de nuevos materiales: Los usuarios recibirán notificaciones cuando un tutor suba nuevos materiales o recursos educativos.

- **Enlace directo al chat:** Los estudiantes y tutores podrán acceder fácilmente al chat desde la pantalla principal o desde la sección de tutorías.

5.2 Requerimientos No Funcionales

Compatibilidad:

- La aplicación será compatible con dispositivos móviles con sistemas operativos Android e iOS , asegurando que los usuarios de ambos sistemas puedan acceder a todas las funcionalidades sin problemas.

Usabilidad:

- La interfaz de la aplicación será intuitiva y fácil de usar , permitiendo que tanto estudiantes como tutores de cualquier edad puedan interactuar con la aplicación sin dificultad.
- Se implementarán guías visuales y tutoriales breves para ayudar a los nuevos usuarios a familiarizarse con la plataforma.

Rendimiento:

- La aplicación debe responder rápidamente a las acciones de los usuarios, garantizando tiempos de carga mínimos y una experiencia fluida incluso en dispositivos de gama baja.
- La sincronización de datos entre usuarios (tutores y estudiantes) debe ser eficiente y en tiempo real.

Seguridad:

- Protección de datos personales: Todos los datos personales de los usuarios (nombre, correo, código, materiales, etc.) estarán protegidos mediante medidas de seguridad robustas.
- Encriptación de comunicaciones: Las comunicaciones entre tutores y estudiantes (en el chat, por ejemplo) deben ser encriptadas para garantizar la privacidad de la información intercambiada.
- Autenticación segura: El sistema de autenticación debe contar con medidas de seguridad como contraseñas encriptadas y opciones de recuperación de contraseñas seguras.

Escalabilidad:

- La aplicación debe ser capaz de manejar un número creciente de usuarios sin que el rendimiento se vea afectado, permitiendo la adición de nuevos usuarios y sesiones sin comprometer la estabilidad de la plataforma.

Accesibilidad:

- La aplicación debe ser accesible para personas con discapacidad visual o auditiva , implementando opciones como el modo de alto contraste , lectores de pantalla y subtítulos para videos.

Mantenimiento:

- El sistema debe ser fácil de mantener y actualizar, permitiendo la implementación de nuevas funcionalidades o correcciones de errores sin afectar la experiencia del usuario.

5.3 Casos de Uso

Identificador.	CU-001
Nombre	Registro de usuario
Prioridad	Normal/Alta.
Descripción	Permite a los usuarios registrarse proporcionando datos personales.
Actores	Sistema, Estudiante / Tutor
Precondición	Es necesario el usuario y código

Iniciada por	A petición del usuario
Post condición	El sistema guarda los datos

Identificador.	CU-002
Nombre	Inicio de Sesión
Prioridad	Alta.
Descripción	Los usuarios acceden a la aplicación autenticándose con su usuario y contraseña.
Actores	Estudiante / Tutor
Precondición	Es usuario debe estar registrado previamente
Iniciada por	A petición del usuario
Post condición	El sistema permite el acceso o muestra un mensaje de error

Identificador.	CU-004
Nombre	Agendar una sesión de tutoría
Prioridad	Alta.
Descripción	Los estudiantes seleccionan y reservan sesiones disponibles en el horario del tutor.
Actores	Estudiante
Precondición	El Estudiante debe estar autenticado.
Iniciada por	Estudiante
Post condición	La sesión queda reservada y se notifica al tutor y al estudiante

Identificador.	CU-005
Nombre	Subir materiales educativos
Prioridad	Normal.

Descripción	Los tutores pueden subir documentos, videos o imágenes asociados a las sesiones de tutoría.
Actores	Tutor
Precondición	El tutor debe estar autenticado.
Iniciada por	Tutor
Post condición	El sistema guarda los materiales y los hace disponibles para los estudiantes

Identificador.	CU-006
Nombre	Descargar materiales educativos
Prioridad	Normal.
Descripción	Los estudiantes descargan los materiales educativos proporcionados por el tutor.
Actores	Estudiante
Precondición	El Estudiante debe estar autenticado.
Iniciada por	Estudiante
Post condición	Los materiales son descargados al dispositivo del estudiante

Identificador.	CU-007
Nombre	Evaluar al Tutor
Prioridad	Normal.
Descripción	Los estudiantes califican y comentan sobre la calidad de los tutores recibidas.
Actores	Estudiante
Precondición	El Estudiante debe haber participado en una sesión.
Iniciada por	Estudiante
Post condición	La evaluación queda registrada en el sistema.

Identificador.	CU-008
Nombre	Enviar mensajes (Chat)
Prioridad	Alta.
Descripción	Los usuarios pueden comunicarse mediante un chat en tiempo real.
Actores	Estudiante/Tutor
Precondición	Ambos deben estar autenticado.
Iniciada por	Usuarios
Post condición	El mensaje es enviado y recibido en tiempo real.

Identificador.	CU-009
Nombre	Recibir Notificaciones
Prioridad	Normal.
Descripción	Los usuarios reciben alertas sobre eventos importantes como sesiones programadas nuevos materiales.
Actores	Estudiante/Tutor
Precondición	Ambos deben estar autenticado.
Iniciada por	Usuarios
Post condición	Las notificaciones se mostraran en el dispositivo del usuario.

Identificador.	CU-010
Nombre	Cerrar Sesión
Prioridad	Normal.

Descripción	Los usuarios finalizan su sesión en la aplicación para proteger sus datos personales.
Actores	Estudiante/Tutor
Precondición	Ambos deben estar autenticado.
Iniciada por	Usuarios
Post condición	El sistema finaliza la sesión y redirige a la pantalla de inicio.

5.4 Diagrama de Clases

Clases Principales del Diagrama

Clase: Usuario

Atributos:

idUsuario: int

nombre: String

correo: String

codigo: String

rol: String (Tutor o Estudiante)

Métodos:

registrarUsuario(): void

iniciarSesion(): boolean

cerrarSesion(): void

Clase: Tutor (Hereda de Usuario)

Atributos:

especialidad: String

horariosDisponibles: List<String>

Métodos:

programarSesion(fecha: Date, hora: String): void

gestionarMateriales(material: Material): void

Clase: Estudiante (Hereda de Usuario)

Atributos:

nivelAcademico: String

Métodos:

reservarSesion(idSesion: int): void

calificarTutor(idTutor: int, puntuacion: int, comentario: String):

void

descargarMaterial(idMaterial: int): Material

Clase: Sesion

Atributos:

idSesion: int

fecha: Date

hora: String

estado: String (Pendiente, Confirmada, Cancelada)

tutor: Tutor

estudiante: Estudiante

Métodos:

crearSesion(tutor: Tutor, fecha: Date, hora: String): void

cancelarSesion(): void

confirmarSesion(): void

Clase: Material

Atributos:

idMaterial: int

titulo: String

descripcion: String

tipo: String (PDF, Video, Imagen)

archivo: File

tutor: Tutor

Métodos:

subirMaterial(): void

consultarMaterial(idMaterial: int): Material

Clase: Chat

Atributos:

idChat: int

participantes: List<Usuario>

mensajes: List<Mensaje>

Métodos:

enviarMensaje(remitente: Usuario, contenido: String): void

recibirMensaje(): List<Mensaje>

Clase: Mensaje

Atributos:

idMensaje: int

contenido: String

fechaEnvio: Date

remitente: Usuario

Métodos:

crearMensaje(contenido: String, remitente: Usuario): void

Clase: Notificacion

Atributos:

idNotificacion: int

tipo: String (Sesion, Material, Recordatorio)

mensaje: String

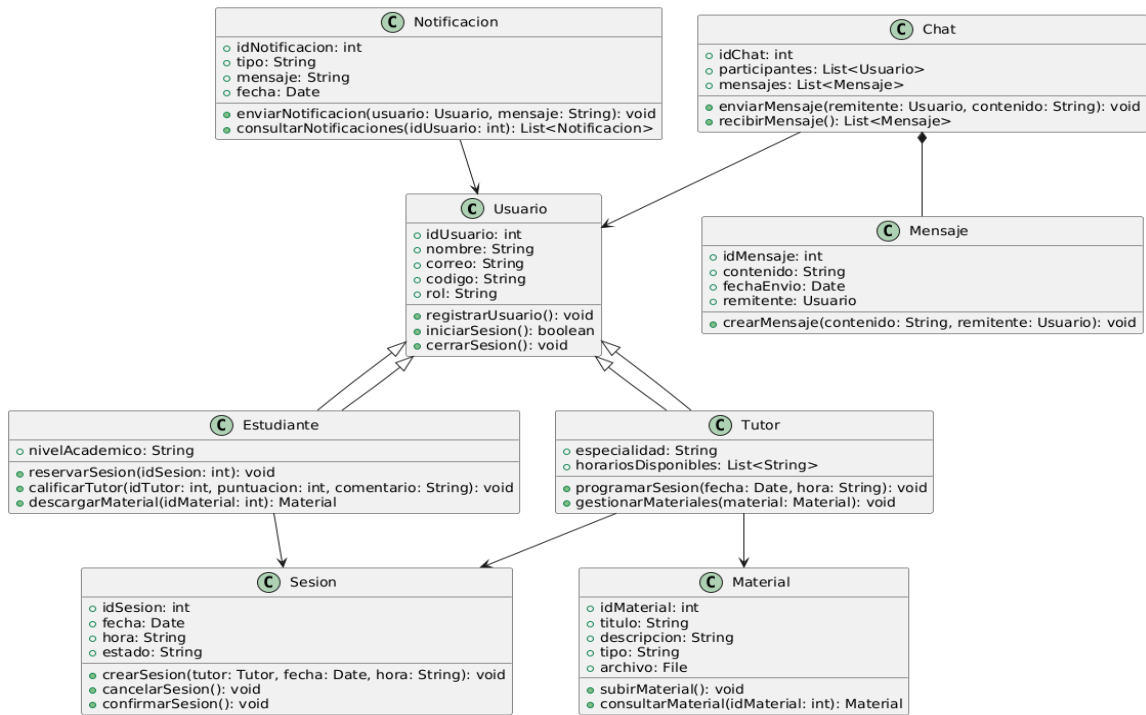
fecha: Date

usuario: Usuario

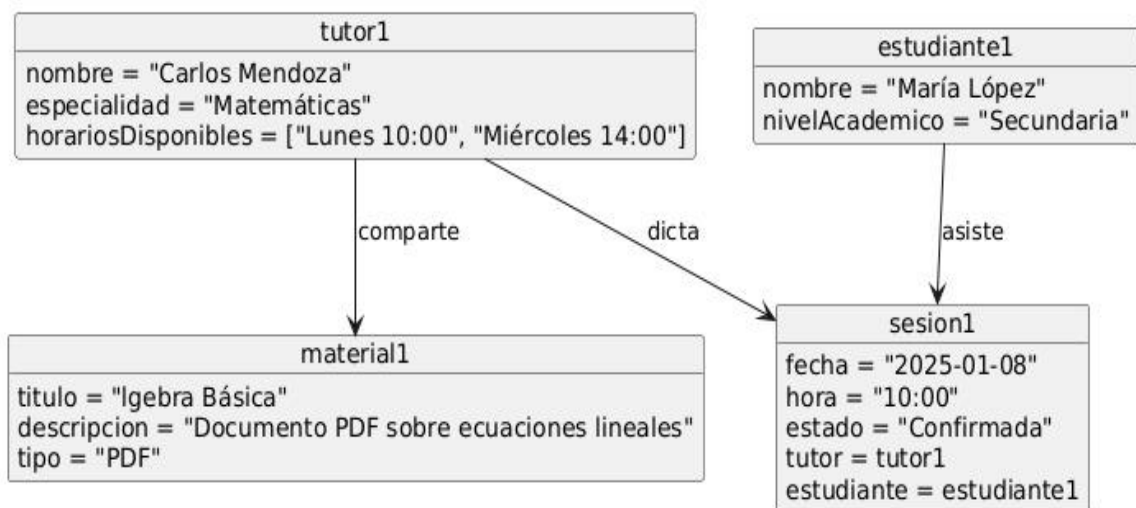
Métodos:

enviarNotificacion(usuario: Usuario, mensaje: String): void

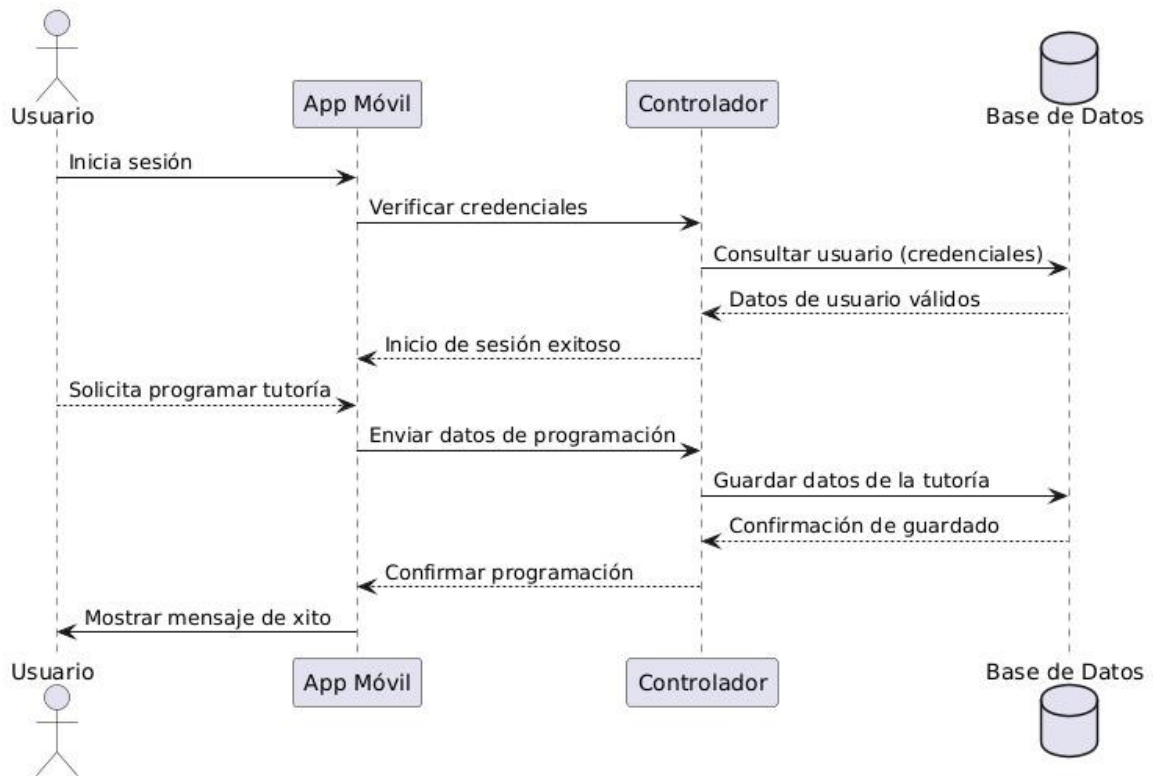
consultarNotificaciones(idUsuario: int): List<Notificacion>



5.5 Diagrama de Objetos



5.6 Diagrama de Secuencia



5.7 Diagrama de Componentes



5.8 Diseño Técnico

El diseño técnico del sistema sigue un modelo cliente-servidor, donde el dispositivo móvil actúa como cliente y los servicios alojados en la nube son el servidor. Este modelo permite una comunicación eficiente entre los diferentes componentes del sistema y facilita el mantenimiento y la escalabilidad.



- Arquitectura del Servicio

La arquitectura del servicio sigue el modelo cliente-servidor, donde:

- Los **dispositivos móviles** actúan como clientes que realizan solicitudes al servidor para obtener datos o ejecutar funcionalidades.
- El **servidor centralizado** maneja la lógica de negocio, el almacenamiento de datos y la gestión de usuarios.

El diseño utiliza una **arquitectura modular** basada en el patrón **MVC (Modelo-Vista-Controlador)**:

- **Modelo:** Responsable de la gestión de datos y la comunicación con la base de datos.
- **Vista:** Presenta la información al usuario final de manera amigable y comprensible.
- **Controlador:** Maneja la interacción del usuario y actúa como puente entre la vista y el modelo.

Este enfoque asegura que las operaciones del sistema sean escalables, modulares y fáciles de mantener.

- Capa de Presentación

La capa de presentación está diseñada para ofrecer una experiencia de usuario intuitiva y accesible. Utiliza el framework **Flutter**, que permite desarrollar interfaces gráficas dinámicas y responsivas, compatibles con dispositivos Android e iOS.

Características principales:

- **Interfaz gráfica intuitiva:** La navegación y las opciones están organizadas de forma que incluso usuarios sin experiencia tecnológica puedan utilizar la aplicación con facilidad.
- **Diseño responsivo:** Las vistas se adaptan automáticamente al tamaño y orientación del dispositivo.
- **Elementos interactivos:** Incluye botones, formularios y gráficos que mejoran la interacción con el usuario.

Además, la capa de presentación asegura una comunicación fluida con la lógica de negocio (Controlador) mediante APIs.

- Arquitectura de Base de Datos

El sistema utiliza una base de datos centralizada para almacenar información clave. Sus principales características son:

- **Modelo relacional:** Diseñado para garantizar la integridad y consistencia de los datos.
- **Tablas principales:**
 - **Usuarios:** Contiene atributos como idUsuario, nombre, correo, y rol (tutor o estudiante).
 - **Sesiones:** Registra tutorías programadas con atributos como idSesion, fecha, hora, tutor y estudiante.
 - **Materiales:** Almacena recursos pedagógicos subidos por los tutores.
 - **Mensajes:** Guarda el historial de comunicaciones entre usuarios.
 - **Notificaciones:** Registra alertas enviadas a los usuarios.

Además, el sistema utiliza **JSON** para el intercambio de datos entre la base de datos y los dispositivos, optimizando la comunicación y asegurando tiempos de respuesta rápidos.

- Prototipo

El prototipo del sistema fue desarrollado utilizando **Flutter** como framework y **Dart** como lenguaje de programación. Sus principales características son:

- **Programación de tutorías:** Los estudiantes pueden reservar sesiones según la disponibilidad de los tutores.
- **Gestión de materiales:** Los tutores pueden subir y organizar materiales educativos, mientras que los estudiantes pueden descargarlos.
- **Sistema de notificaciones:** Envía recordatorios automáticos sobre tutorías programadas o nuevos materiales disponibles.
- **Interfaz interactiva:** Prototipo funcional que simula la navegación entre las secciones principales, como registro de usuario, reserva de sesiones, y comunicación mediante chat.

El prototipo fue diseñado para pruebas iniciales con usuarios, con el objetivo de validar la usabilidad y realizar ajustes antes de la implementación final.

Bibliografía

Lucidchart (2023)

Cómo crear documentos de diseño de software. Proporciona herramientas de diagramación para estructurar modelos y sistemas visuales.

<https://www.lucidchart.com/blog/es/como-crear-documentos-de-diseno-de-software>

Open Access UOC (2023)

Documento académico que describe arquitecturas técnicas en aplicaciones móviles y proyectos finales.

<https://openaccess.uoc.edu/bitstream/10609/66085/3/mromeropoTFG0617memoria.pdf>

Dart.dev (2023)

Sitio oficial del lenguaje de programación Dart, enfocado en aplicaciones de alto rendimiento.

<https://dart.dev/>

Flutter.dev (2023)

Plataforma oficial para el desarrollo de aplicaciones móviles utilizando Flutter.

<https://flutter.dev/>

PlantUML (2023)

Herramienta para crear diagramas UML y visualizaciones de sistemas de forma automatizada.

<https://plantuml.com/es/>

UML.org (2023)

Sitio oficial de UML (Unified Modeling Language), que detalla estándares de modelado visual para sistemas.

<https://www.uml.org/>