

Desarrollo de App

Detalle Técnico

Cómputo móvil

Alumnos

- Ortega Mendoza Rubén
- Pazaran Estrada Erick Iván
- Torres Gonzalez Billy Martin

Contenido

Levantamiento de requerimientos de la app	2
Reglas de Negocio:	2
Requerimientos Funcionales:	3
Requerimientos No Funcionales:	3
Justificación de funcionalidad	4
Funcionalidad:	4
Viabilidad:	4
Inviabilidad:	4
Determinar alcance y MVP de la app.	5
Alcance:	5
MVP:	5
Separación Modelo Vista Controlador	6
Modelo	6
Vista	6
Controlador	7
Tipo de permisos	7
Servicios para funcionalidades	8
Backend	9
API	9
Tecnología y Autenticación	9
Flujo de Datos	10
Del Usuario a la Aplicación	10
Del Servicio Backend al Usuario	10
Persistencia de Datos	10
Seguridad de los Datos	11
Pantallas	12

Levantamiento de requerimientos de la app

Reglas de Negocio:

- Registro de Usuarios:
Los usuarios deben registrarse con información básica (nombre, edad, correo electrónico). Los menores de edad deben contar con el consentimiento de sus padres o tutores.
- Privacidad y Seguridad:
Los datos personales deben ser encriptados y protegidos.
La plataforma debe cumplir con las regulaciones de privacidad de datos (ej. GDPR, CCPA).

- Pago de Servicios:
Los usuarios podrán acceder a ciertas funcionalidades gratuitas, pero se requerirá un pago para acceder a clases avanzadas o sesiones con tutores personalizados.

Requerimientos Funcionales:

- Perfil de Usuario:
Creación de perfiles personalizados que incluyan preferencias de aprendizaje, nivel de habilidad y progreso.
- Generación de Tutores Personalizados:
Sistema de inteligencia artificial que evalúe el nivel de conocimiento y diseñe un plan de estudios personalizado para cada usuario.
Recomendaciones de ejercicios, problemas y materiales de estudio adaptados a cada usuario.
- Clases en Vivo:
Calendario de clases en vivo con horarios predefinidos y variedad de temas.
Plataforma de chat en tiempo real para interactuar con los tutores y otros usuarios durante las clases.
- Sistema de Evaluación:
Mecanismos de evaluación para realizar seguimiento del progreso de los usuarios.
Retroalimentación instantánea sobre el desempeño en las clases y ejercicios.
- Comunidad y Foros:
Espacios para discusiones y preguntas en una comunidad de usuarios interesados en matemáticas.

Requerimientos No Funcionales:

- Rendimiento:
La aplicación debe ser fluida y responder rápidamente, incluso en condiciones de alta demanda de usuarios.
Tiempo de respuesta del sistema no debe exceder los límites aceptables.
- Usabilidad y Diseño:
Interfaz intuitiva y amigable para usuarios de todas las edades.
Soporte para diferentes dispositivos móviles y sistemas operativos (iOS, Android).
- Seguridad:
Protección contra intrusiones, ataques de seguridad y filtraciones de datos.
Actualizaciones periódicas para corregir vulnerabilidades de seguridad.
- Compatibilidad y Escalabilidad:
Integración con diferentes navegadores y dispositivos móviles.
Capacidad de expansión para admitir un crecimiento futuro en el número de usuarios.
- Disponibilidad:
Garantizar un tiempo de actividad alto y un mínimo tiempo de inactividad para mantenimiento programado.

Justificación de funcionalidad

Funcionalidad:

La implementación de un tutor personalizado a través de inteligencia artificial (IA) dentro de la aplicación Studentz tiene sus pros y contras.

Viabilidad:

Accesibilidad inmediata: La generación de un tutor personalizado bajo demanda proporciona acceso instantáneo a la asistencia educativa, eliminando la espera y las limitaciones de tiempo asociadas con los tutores humanos.

Amplitud de temas y personalización: La IA puede abarcar una amplia gama de temas y adaptarse a las necesidades específicas de cada usuario, ofreciendo una experiencia educativa altamente personalizada.

Reducción de costos: En comparación con un tutor humano, hacer uso de IA puede resultar más económico, ya que elimina los gastos continuos asociados con la contratación de profesionales humanos.

Inviabilidad:

Costos de desarrollo y mantenimiento: Implementar un sistema de IA para generar tutores personalizados puede ser costoso tanto en la fase de desarrollo como en la de mantenimiento. Se requeriría la inversión en tecnología avanzada de IA y en recursos humanos especializados.

Requisitos de ancho de banda y capacidad de procesamiento: Las video tutorías y la interacción en tiempo real con un avatar virtual necesitarán un ancho de banda considerable y una capacidad de procesamiento robusta. Esto podría generar gastos adicionales significativos para mantener servidores potentes y una infraestructura estable.

Limitaciones técnicas y calidad: La capacidad de la IA para proporcionar una enseñanza de alta calidad y personalización puede verse afectada por las limitaciones actuales de la tecnología. La experiencia del usuario podría no ser tan fluida, especialmente en comparación con la interacción con un tutor humano.

Solicitudes simultáneas y escalabilidad: Si múltiples usuarios solicitan un tutor personalizado al mismo tiempo, el sistema debe estar diseñado para manejar estas solicitudes concurrentes sin comprometer la calidad del servicio. Esto puede requerir una escalabilidad compleja y costosa.

En resumen, la implementación de un tutor personalizado a través de inteligencia artificial en la aplicación Studentz ofrece beneficios notables en términos de acceso inmediato, amplitud de temas y reducción de costos. Sin embargo, las dificultades asociadas con los costos de desarrollo y mantenimiento, los requisitos técnicos de ancho de banda y capacidad de procesamiento, así como las limitaciones de calidad podrían hacer que esta funcionalidad sea inviable o al menos difícil de ejecutar sin una inversión significativa y un análisis detallado de los recursos requeridos.

Determinar alcance y MVP de la app.

Alcance:

Registro y Perfiles de Usuario: Crear perfiles de usuario donde se puedan almacenar preferencias, historial de clases, progreso, etc.

Clases en Vivo y Chat: Ofrecer clases en vivo de matemáticas con un chat integrado para la interacción entre usuarios y profesores.

Biblioteca de Recursos: Proporcionar acceso a una variedad de recursos educativos, como videos, documentos, ejercicios prácticos, etc.

Sistema de Calificación y Retroalimentación: Permitir a los usuarios calificar las clases y proporcionar comentarios para mejorar la calidad de la enseñanza.

Suscripciones: Implementar un sistema de suscripción mensual y anual para acceder a las clases y recursos premium.

Soporte al Cliente: Ofrecer soporte técnico y asistencia a los usuarios para resolver problemas y responder preguntas.

MVP:

Dentro del MVP (Producto Mínimo Viable), se enfocaría en lo esencial para validar la viabilidad y utilidad de la aplicación Studentz, lanzando una versión básica con las siguientes características:

- Registro y Perfiles: Funcionalidad básica para que los usuarios puedan registrarse, iniciar sesión y editar sus perfiles.
- Clases en Vivo y Chat: Implementación de clases en vivo con un chat sencillo para la interacción entre usuarios y un instructor. No se incluiría la generación de tutores personalizados en esta etapa inicial.
- Biblioteca Limitada: Una selección inicial de recursos educativos disponibles para los usuarios.
- Sistema de Suscripción: Implementación básica del sistema de suscripción para acceder al contenido premium.
- Soporte Básico: Ofrecer un canal de soporte limitado para preguntas frecuentes y problemas técnicos simples.

Separación Modelo Vista Controlador

Modelo

Lógica de Negocio: Todas las operaciones y lógica relacionada con el negocio y los datos se encuentran en el modelo. Por ejemplo, la lógica para generar tutores personalizados, el manejo de datos de usuarios y suscripciones, el acceso a la base de datos, etc.

Representación de Datos: El modelo se encarga de representar los datos y su estructura. Esto puede incluir clases, estructuras, o cualquier otra representación de los datos necesarios para la aplicación.

Vista

Interfaz de Usuario (UI): Todas las partes visuales y elementos con los que los usuarios interactúan se encuentran en la vista. Por ejemplo, la interfaz gráfica para ver las clases disponibles, la opción de suscripción, botones para interactuar con la aplicación, etc.

Presentación de Datos: La vista se encarga de mostrar la información al usuario de una manera comprensible y atractiva. Esto puede incluir la presentación de resultados de búsqueda, estadísticas de progreso del usuario, mensajes de confirmación, entre otros.

Controlador

Gestión de Eventos: El controlador maneja las interacciones del usuario y los eventos que ocurren en la aplicación. Por ejemplo, al hacer clic en un botón para suscribirse, el controlador captura ese evento y realiza las acciones correspondientes.

Enrutamiento y Lógica de Flujo: El controlador decide cómo interactuarán los datos del modelo con la vista. Gestiona las solicitudes del usuario, decide qué datos se deben mostrar y cómo deben presentarse en la vista. La vista se encarga de la presentación visual, el modelo maneja los datos y la lógica de negocio, mientras que el controlador gestiona las interacciones y el flujo de la aplicación.

Tipo de permisos

Política de Privacidad: Es esencial incluir una política de privacidad clara y comprensible para los usuarios. Esta política debe explicar cómo se recopilan, utilizan, almacenan y protegen los datos de los usuarios.

Permisos de la Aplicación: Se solicitan solo los permisos necesarios para el funcionamiento de la aplicación. Acceso a la cámara, micrófono, datos del usuario, datos de pago.

Cumplimiento Legal y Normativo: Asegúrate de cumplir con todas las leyes y regulaciones locales e internacionales aplicables, como la Ley de Protección de Datos (GDPR en la Unión Europea), COPPA (Children's Online Privacy Protection Act), entre otras, según corresponda a tu aplicación y público objetivo.

Calidad y Seguridad de la Aplicación: La aplicación debe ser estable, segura y funcionar según lo anunciado. Las tiendas de aplicaciones revisan la calidad de las aplicaciones antes de su publicación para garantizar una experiencia satisfactoria para los usuarios.

Pago y Monetización: Si tu aplicación incluye compras integradas, suscripciones u otros modelos de monetización, asegúrate de seguir las políticas y pautas establecidas por la

tienda de aplicaciones con respecto a la divulgación de precios, manejo de pagos y reembolsos.

Servicios para funcionalidades

Clases en Vivo y Chat

WebRTC (Web Real-Time Communication): Para la transmisión de video en tiempo real y la comunicación entre usuarios.

Firebase Realtime Database: Para implementar el chat en tiempo real entre usuarios.

Justificación: WebRTC es una tecnología ampliamente utilizada para comunicaciones en tiempo real y streaming de video en aplicaciones web y móviles. Ofrece funcionalidades para la transmisión de video en vivo de alta calidad. Para el chat en tiempo real, Firebase Realtime Database permite una comunicación instantánea entre usuarios de manera eficiente y escalable.

Sistema de Suscripciones

Stripe: Para procesamiento de pagos y gestión de suscripciones.

Justificación: Stripe ofrece SDK y API robustas que permiten integrar fácilmente sistemas de pago dentro de aplicaciones móviles. Proporciona una experiencia de usuario segura y confiable al mismo tiempo que ofrecen opciones de suscripción y procesamiento de pagos.

Generación de Tutor Personalizado (Bajo Demanda)

ChatGPT, Google Cloud AI Platform o Amazon Lex: Para la implementación de la inteligencia artificial de la generación de tutorías personalizadas.

Unity: Para la creación de avatares virtuales y la interfaz de usuario en 3D.

Justificación: Google Cloud AI Platform y Amazon Lex ofrecen capacidades de aprendizaje automático y procesamiento de lenguaje natural para la creación de asistentes virtuales inteligentes que puedan actuar como tutores personalizados. Para la representación visual de estos tutores, el uso de Unity puede ser beneficioso debido a su capacidad para desarrollar avatares.

Gestión de Datos y Usuarios

Firebase: Para la gestión de la base de datos, autenticación de usuarios y almacenamiento de archivos.

Justificación: Firebase ofrece soluciones integrales para la gestión de usuarios, bases de datos en tiempo real, autenticación segura y almacenamiento de archivos. Permiten escalar la aplicación, asegurar los datos y proporcionar una experiencia de usuario confiable.

Backend

Contaría con las siguientes características:

Registro de usuarios con autenticación segura (preferiblemente mediante tokens JWT).

Almacenamiento de información de perfil de usuario, preferencias y datos relevantes.

Almacenamiento y gestión de información sobre clases disponibles, temas, materiales educativos, etc.

Posibilidad de programar y modificar clases, asignar avatar de tutor, y mantener actualizada la información.

Implementación de un sistema de notificaciones para recordatorios, actualizaciones de contenido, adición de temas, etc.

Implementación de funciones en tiempo real para el chat entre usuarios durante las clases en vivo.

Integración con servicios de pago para gestionar suscripciones y procesar transacciones de pagos.

Gestión de roles y permisos para usuarios, garantizando que accedan solo a las funcionalidades autorizadas.

Recopilación y análisis de datos para evaluar el progreso del usuario, interacciones en la aplicación, entre otros.

API

A continuación, se presentan ejemplos hipotéticos de algunos endpoints que podrían estar presentes en la API:

/auth/signup: Endpoint para registro de nuevos usuarios.

/auth/login: Endpoint para inicio de sesión y obtención de token JWT.

/classes: Endpoint para obtener información sobre las clases disponibles.

/classes/{class_id}: Endpoint para acceder a detalles específicos de una clase.

/user/profile: Endpoint para acceder y modificar el perfil de usuario.

/subscriptions: Endpoint para gestionar suscripciones y pagos.

Tecnología y Autenticación

Se desarrollaría en un Servidor Virtual Privado de inicio con un sistema operativo Red Hat Linux.

Lenguaje de Programación y Framework: Se podría utilizar Node.js con Express.js para la construcción del backend debido a su eficiencia, escalabilidad y la amplia gama de librerías disponibles.

Base de Datos: Una base de datos en Firebase podría ser una opción viable para almacenar información relacionada con usuarios, clases y contenido educativo.

Autenticación: Se puede implementar un sistema de autenticación basado en tokens JWT para asegurar las peticiones a la API y proteger los endpoints sensibles.

Flujo de Datos

Del Usuario a la Aplicación

El usuario interactúa con la interfaz de la aplicación para acceder a clases, configurar su perfil, ver recursos, etc.

La aplicación móvil envía solicitudes al servicio backend a través de API restful para obtener información, enviar datos o realizar acciones específicas.

Del Servicio Backend al Usuario

El servicio backend procesa las solicitudes del usuario, accede a la base de datos para obtener información requerida y genera respuestas adecuadas.

La aplicación móvil recibe los datos del backend y los presenta al usuario a través de la interfaz de usuario.

Persistencia de Datos

Los datos de los usuarios, información de perfil, historial de clases, preferencias, etc., se almacenan en una base de datos en el servicio backend.

La información de las clases, recursos educativos y datos relevantes para la funcionalidad de la aplicación también se almacena en la base de datos del backend.

Se puede implementar un mecanismo de almacenamiento en caché en la aplicación móvil para acceder a datos previamente obtenidos y almacenados localmente en caso de falta de conexión.

La aplicación podría ofrecer funcionalidades limitadas en modo sin conexión, como acceder a recursos descargados previamente, ver clases guardadas localmente, etc.

Seguridad de los Datos

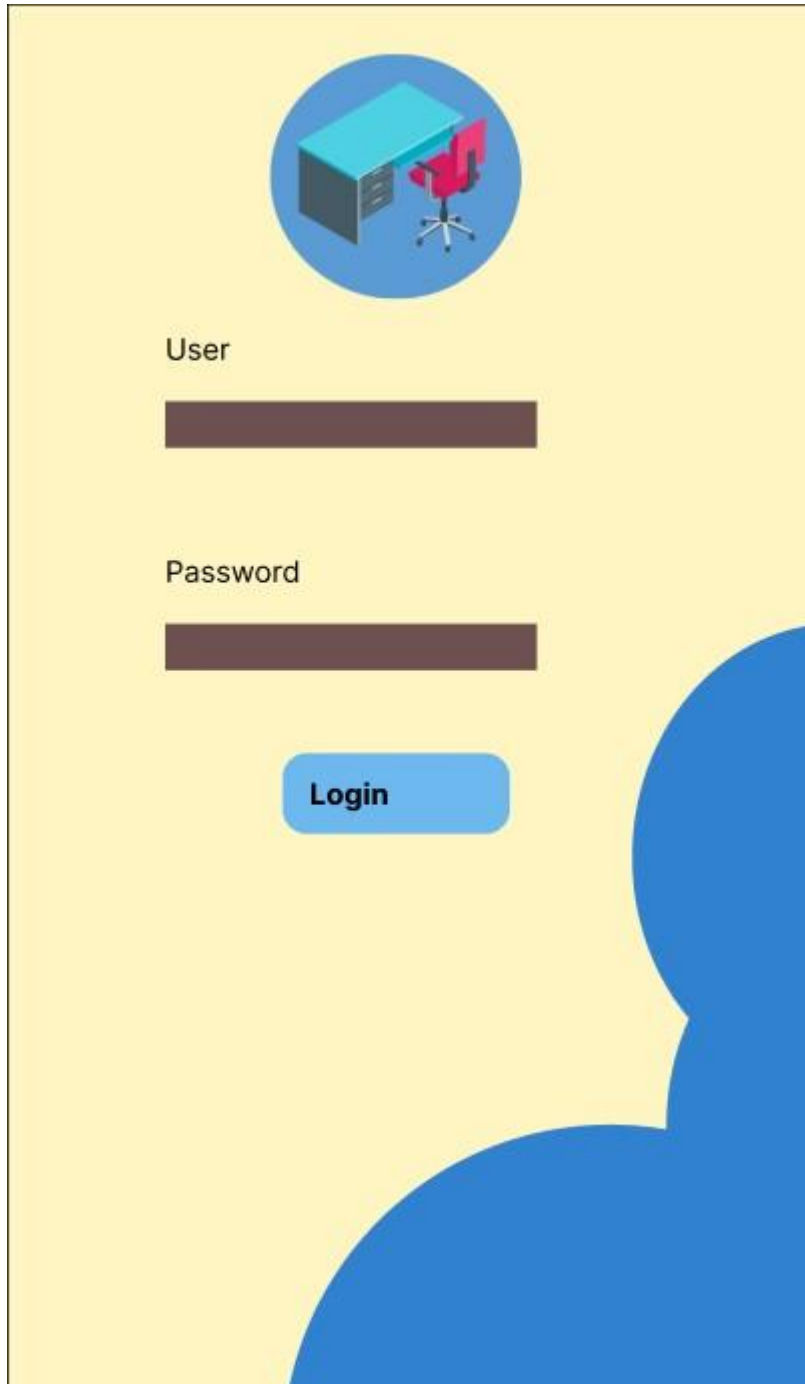
Se debe implementar una capa de seguridad en la comunicación entre la aplicación y el servicio backend utilizando protocolos seguros como HTTPS para proteger la transferencia de datos.

La autenticación de usuarios se realizará mediante tokens JWT para asegurar que solo usuarios autorizados accedan a la información.

Se aplicarán técnicas de encriptación para proteger los datos sensibles almacenados en la base de datos del servicio backend.


Pantallas

- Pantalla de Inicio de Sesión: Permite a los usuarios iniciar sesión en sus cuentas existentes.

The illustration shows a login interface on a yellow background. At the top center is a circular icon containing a blue desk and a red office chair. Below this icon, the word "User" is positioned above a dark brown rectangular input field. Further down, the word "Password" is positioned above another dark brown rectangular input field. Below the password field is a light blue rounded rectangular button with the word "Login" in black text. In the bottom right corner, there is a large, abstract blue shape that resembles a stylized person's head and shoulder.

- Pantalla de Registro: Facilita a los nuevos usuarios la creación de una cuenta en la aplicación.

A mockup of a registration screen with a yellow background and a blue abstract shape at the bottom right. At the top center is a circular icon containing a blue desk and a red chair. Below the icon is the title "Register" in bold black text. Further down are three labels: "Username", "E-mail", and "Password", each followed by a dark brown rectangular input field.



Register

Username

E-mail

Password

- Pantalla de Perfil de Usuario: Muestra la información personal del usuario y permite editarla si es necesario.



Edita tu perfil

Username

E-mail

País

Tipo de suscripción



- Pantalla de Inicio: Presenta opciones principales y acceso rápido a las funcionalidades clave de la aplicación.



- Pantalla de Clases Disponibles: Muestra una lista de las clases en vivo y grabadas disponibles para los usuarios.



- Pantalla de Biblioteca de Recursos: Proporciona acceso a una variedad de recursos educativos como videos, documentos, ejercicios, etc.



- Pantalla de Configuración de Notificaciones: Permite a los usuarios personalizar las notificaciones de la aplicación.



- Pantalla de Suscripciones: Muestra las opciones de suscripción disponibles y permite a los usuarios gestionar su membresía.



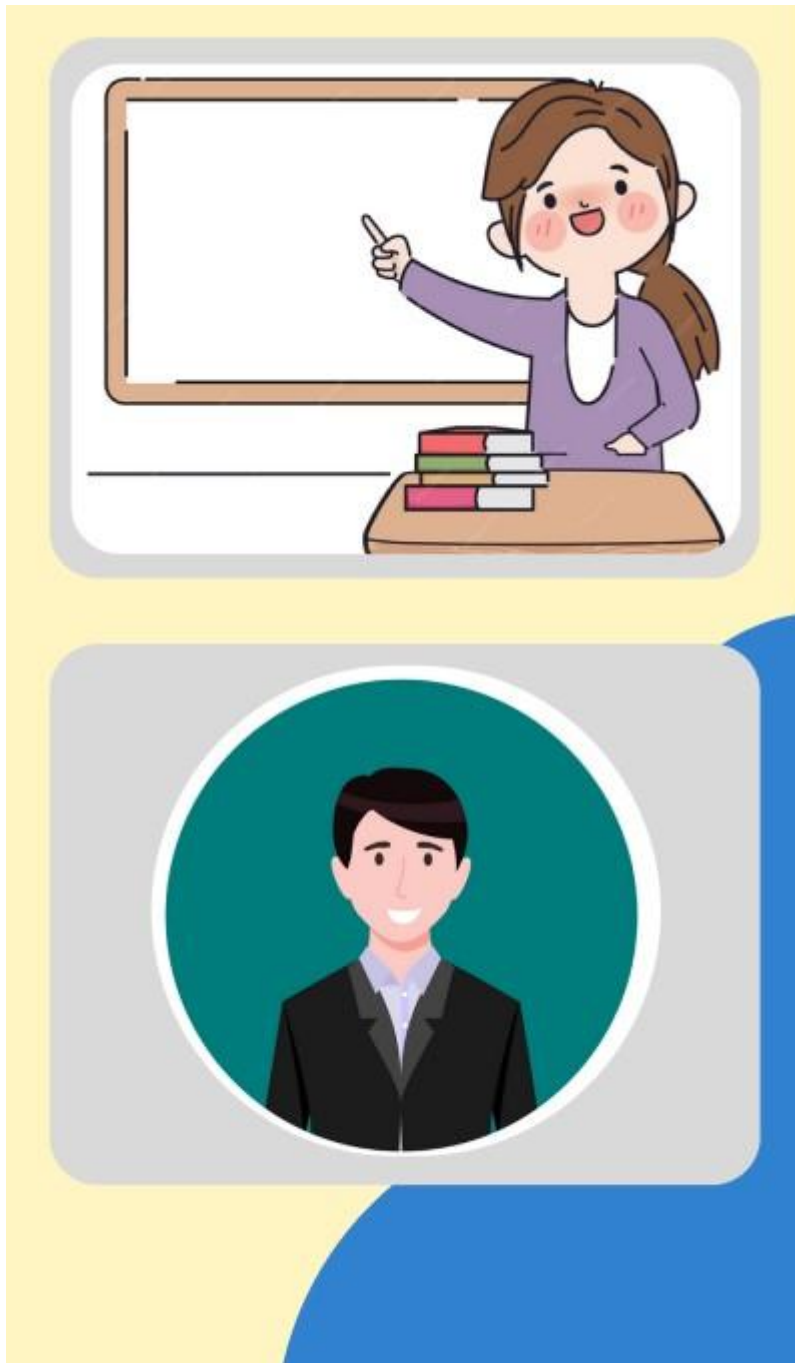
- Pantalla de FAQ/Ayuda: Proporciona respuestas a preguntas frecuentes y ayuda con problemas comunes.



- Pantalla de Selección de Clases: Muestra un listado de los temas a elegir para una clase.



- Pantalla de Interacción con el Tutor Virtual: Proporciona acceso a la tutoría personalizada a través de la inteligencia artificial.



- Pantalla de Clase Virtual y Mensajes/Chat con Otros Usuarios: Permite a los usuarios comunicarse entre sí para discutir temas relacionados con las clases.



- Pantalla de Compartir Clases: Permite a los usuarios compartir información sobre clases interesantes en redes sociales u otros medios.



- Pantalla de Feedback y Valoración de Clases: Permite a los usuarios proporcionar retroalimentación y calificaciones sobre las clases que han tomado.

¡Evalúa tu clase!

Tu opinión

Evaluación

★ ★ ★ ★ ☆

Mandar