

REQUERIMIENTOS FUNCIONALES:

1. Registro de inicio de sesión
 - a. Integración con el sistema de gestión académica de la universidad para validar usuarios.
 - b. Verificación de identidad por correo o SMS.
2. Lista de contactos dinámica:
 - a. Estudiantes: Añadir contactos mediante búsqueda por nombre o matrícula.
 - b. Profesores: Visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.
3. Chat individual
 - a. Interfaz sencilla para enviar y recibir mensajes.
 - b. Indicadores de "escribiendo", leído y entregado.
4. Chat grupal
 - a. Creación de grupos por estudiantes o profesores (según permisos).
 - b. Gestión de grupos por el creador (añadir/eliminar miembros).
5. Historial de mensajes
 - a. Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.
6. Notificaciones en tiempo real
 - a. Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.
7. Búsqueda de mensajes y contactos
 - a. Filtros avanzados para localizar conversaciones o usuarios específicos
8. Integración con calendario académico:
 - a. Recordatorios automático de clases o actividades asociadas a un curso

REQUERIMIENTOS NO FUNCIONALES

- Seguridad
 - Uso de encriptación para los mensajes(e.g., cifrado end-to-end)
- Rendimiento
 - Respuesta rápida en la carga de mensajes y sincronización en tiempo real
- Mantenibilidad
 - Código modular y documentado para facilitar actualizaciones
- Disponibilidad
 - Sistema operativo 24/7 con menos del 1% de tiempo de inactividad

PRIORIZACIÓN POR EL MÉTODO MOSCOW

MUST

- Registro de inicio de sesión
- Seguridad
- Disponibilidad
- Chat grupal

SHOULD

- Mantenibilidad
- Chat individual
- Lista de contactos dinámica:
- Historial de mensajes
- Rendimiento

COULD

- Búsqueda de mensajes y contactos
- integración con calendario académico

WON'T

- Notificaciones en tiempo real

ESTIMACIÓN POR LE MÉTODO Planning poker

Números fibonacci: 1,2,3,5,8,13,21,34,55

Javier Giraldo:

Tarea	Estimado	Justificación
Registro de inicio de sesión	3	Es una tarea básica en desarrollo de software actual. La complejidad radica en la validación y autenticación.
Seguridad	8	La seguridad End-to-end requiere de pruebas largas y detalladas para poder entregar un producto en condiciones óptimas.
Disponibilidad	13	La configuración para tener una inactividad tan baja implica la elaboración de estructuras complejas y robustas.
Chat grupal	8	El manejo de roles, usuarios es una tarea ciertamente simple. Preparar la estructura para el manejo de mensajes es la parte donde radica la dificultad.
Mantenibilidad	5	La documentación requiere de tiempo, organizar el código respectivamente para que sea legible, entendible y ordenado consume tiempo.
Chat individual	5	El problema sería la sincronización de los mensajes además de organizar otra estructura para el manejo de mensajes.

Lista de contactos dinámica	5	La base de datos debe estar construida de la forma más óptima para poder filtrar, agregar y buscar contactos de forma dinámica. Sin embargo la estructura de la base de datos no significa un reto mayor.
Historial de mensajes	8	La construcción de las estructuras adecuadas para el manejo de los chats grupales dan cierto apoyo sobre este punto. Sin embargo sigue siendo una tarea que requiere de un manejo eficiente de datos.
Búsqueda de mensajes y contactos	5	La complicación de esto radica en los puntos de la organización de las bases de datos respectivas que se suponen deben estar configurados de forma eficiente.
Integración del calendario académico	8	Vincular las actividades, notificaciones y mantenerlo sincronizado con el calendario académico de los usuarios puede ser complejo según la disponibilidad de acceso a este.
Notificaciones en tiempo real	3	La dificultad radica en la sincronización de eventos entre dispositivos y plataformas, sin embargo hay varias herramientas en las que se puede apoyar como desarrollador.

Álvaro Romero:

Tarea	Estimado	Justificación
Registro de inicio de sesión	2	Porque no es una tarea difícil con herramientas actuales como API para manejar la autenticación.
Lista de contactos dinámica	2	Porque es simplemente guardar algunos datos por usuario que se quiera añadir a la lista.
Chat individual	13	Debido a que se tienen que hacer conexiones asíncronas en usuarios puede resultar en varios edge cases.
Chat grupal	5	Porque con un chat individual hecho no debería ser muy difícil implementar uno grupal. Aunque sigue siendo complicado.
Historial de mensajes	3	Porque con los chats ya hechos debería estar casi que implementado, solo quedaría la búsqueda y puede dar algunos problemas.
Notificaciones en tiempo real	2	Porque al igual que antes con los chats ya hechos no debería ser difícil.
Búsqueda de mensajes y contactos	3	Ya debe haber herramientas para buscar los datos e incluso si no, no es particularmente difícil.
Integración con calendario académico	3	Lo más difícil puede ser hacer un buen front end pero las operaciones back end deben ser fáciles.

Seguridad	13	Para asegurar que todo el sistema sea seguro y que no se escapen fácilmente vulnerabilidades.
Rendimiento	8	Porque para este punto probablemente se deben refactorizar partes del código y esto puede tomar tiempo.
Mantenibilidad	5	Esta podría ya estar hecha si se hace bien el código. Pero añadir documentación para el proyecto puede llegar a tomar mucho tiempo si no.
Disponibilidad	21	Porque puede tomar cambios no solo al software sino a los servidores usados y esto puede tomar mucho tiempo.

Manuel Navas:

Tarea	Estimado	Justificación
Registro de inicio de sesión	2	mediante el uso de librerías para encriptar contraseñas es una tarea bastante básica y común que no llevaría mucho tiempo
Lista de contactos dinámica	2	
Chat individual	13	Requiere implementar comunicación en tiempo real y garantizar la persistencia de mensajes, lo que implica un esfuerzo moderado pero manejable.
Chat grupal	5	Aunque similar al chat individual, se requiere manejar múltiples usuarios simultáneamente en un canal, lo que añade algo de complejidad.
Historial de mensajes	8	Implica integrar un sistema de almacenamiento y recuperación eficiente de mensajes pasados, lo que lleva tiempo pero es factible con buenas prácticas.
Notificaciones en tiempo real	3	Con el uso de servicios como Firebase Cloud Messaging, esta funcionalidad puede integrarse rápidamente.
Búsqueda de mensajes y contactos	3	Requiere la implementación de un sistema de indexación o filtrado, pero con herramientas actuales puede lograrse en poco tiempo
Integración con calendario académico	13	Requiere integrar APIs de calendarios externos y sincronizar eventos con la aplicación, lo que demanda un tiempo significativo de desarrollo y pruebas.
Seguridad	13	Implementar medidas de seguridad como cifrado, autenticación robusta y pruebas contra vulnerabilidades demanda una cantidad importante de tiempo para garantizar la calidad.
Rendimiento	5	Optimizar la aplicación para que sea eficiente en diferentes escenarios y dispositivos requiere un esfuerzo moderado.

Mantenibilidad	5	Asegurar que el código sea limpio, modular y documentado demanda tiempo para planificación y ejecución durante el desarrollo.
Disponibilidad	5	Implementar soluciones como balanceo de carga y sistemas redundantes garantiza la disponibilidad del sistema, lo cual requiere un esfuerzo moderado.

Tomás Vallejo:

Tarea	Estimado	Justificación
Registro de inicio de sesión	2	No es una tarea muy compleja, implica diseñar una interfaz, implementar validaciones básicas y proteger las credenciales
Seguridad	9	Implementar cifrado end-to-end requiere conocimiento especializado y pruebas rigurosas para garantizar que los datos sean inaccesibles incluso si el servidor es comprometido.
Disponibilidad	5	Asegurar alta disponibilidad requiere configurar servidores redundantes, equilibradores de carga y un sistema de monitoreo para garantizar tiempo de actividad, entre otros
Chat grupal	6	Construir la funcionalidad de grupos incluye creación, gestión de permisos, sincronización de mensajes entre miembros y notificaciones específicas
Mantenibilidad	5	Crear un código modular y bien documentado es un esfuerzo constante durante el desarrollo, que incluye revisiones de código y documentación técnica.
Chat individual	13	Crear un sistema de chat básico requiere diseñar la interfaz, y realizar una implementación para manejar los mensajes y establecer actualizaciones en tiempo real
Lista de contactos dinámica	3	Implica diseñar dos flujos distintos (estudiantes y profesores), realizar búsquedas optimizadas en la base de datos y establecer reglas para la actualización automática
Historial de mensajes	3	Almacenar y recuperar mensajes requiere diseñar una base de datos eficiente
Búsqueda de mensajes y contactos	3	Crear filtros avanzados para buscar información implica optimización de consultas y desarrollo de una interfaz intuitiva para el usuario final.
Integración del calendario académico	6	Implica conectar datos del sistema de gestión académica, generar recordatorios automáticos y diseñar notificaciones para el usuario. La sincronización de datos en tiempo real añade dificultad.

Notificaciones en tiempo real	4	Involucra la implementación de notificaciones push , manejo de eventos en el backend y creación de configuraciones para silenciar chats específicos.
-------------------------------	---	--

CONSENSO

Tarea	Estimado
Registro de inicio de sesión	3
Seguridad	8
Disponibilidad	13
Chat grupal	8
Mantenibilidad	5
Chat individual	5
Lista de contactos dinámica	5
Historial de mensajes	8
Búsqueda de mensajes y contactos	5
Integración del calendario académico	8
Notificaciones en tiempo real	3