

Tarea 03

Sebastian Andrade Cedano
Cristian Steven Motta Ojeda
Juan Esteban Santacruz Corredor

Ingeniería de software I

Universidad Nacional de Colombia

Bogotá D.C.
2024

Priorización

Funcionales:

- Registro e inicio de sesión. **(Must)**
- Lista de contactos dinámica. **(Should)**
- Chat individual. **(Must)**
- Chat grupal. **(Should)**
- Historial de mensajes. **(Must)**
- Notificaciones en tiempo real. **(Won't)**
- Búsqueda de mensajes y contactos. **(Must)**
- Integración con calendario académico. **(Won't)**

No funcionales:

- Seguridad. **(Must)**
- Rendimiento. **(Must)**
- Mantenibilidad. **(Should)**
- Disponibilidad. **(Should)**

Estimación personal **(Juan Esteban Santacruz Corredor)**

Funcionales:

- *Registro e inicio de sesión (8 días):* Con el uso de herramientas como JWT y Aragón 2, se podrá gestionar la autenticación y manejo de claves. Sin embargo, como no tengo experiencia con estas tecnologías, necesitaré algunos días para estudiarlas e implementarlas (aprox 3 días estudiando y 2 días implementando). Por otra parte, con React puedo desarrollar el frontend en poco tiempo, gracias al conocimiento que tengo en la tecnología (aprox 1 días). Cualquier contratiempo se contempla para los días 6 y 7.
- *Lista de contactos dinámica (3 días):* Con el uso de React para el frontend, una base de datos MySQL para el backend y una API que conecte ambos, se puede desarrollar esta tarea. Aunque tengo el conocimiento desarrollando este tipo de funcionalidades, hace tiempo que no las implemento, por lo que me tomará aproximadamente 3 días para completarlo.
- *Chat individual (5 días):* Con React y WebSockets se puede desarrollar un frontend y agregar funcionalidades para la comunicación en tiempo real con los usuarios. A pesar de que nunca he desarrollado este tipo de funcionalidad, me desenvuelvo bien con proyectos basados en Node.js, por lo que el estudio de WebSockets me tomaría aproximadamente 2

días al igual que la integración con el frontend. Cualquier contratiempo se contempla para el día 5.

- Chat grupal (**5 días**): Muy similar al punto anterior.
- *Historial de mensajes (3 días)*: Con buena planificación, esta tarea podría hacerse en menos tiempo. No es necesario aprender nuevas tecnologías y he desarrollado este tipo de funcionalidades anteriormente.
- *Notificaciones en tiempo real (3 días)*: Dado que esta funcionalidad tiene baja prioridad, ya habré ganado experiencia con WebSockets al desarrollar el chat individual. Con mi conocimiento en react se podrá implementar en un día el frontend.
- *Búsqueda de mensajes y contactos (3 días)*: Dependiendo la cantidad de datos con los que se esté trabajando, puedo implementar la búsqueda desde la misma api que tiene la conexión a la base de datos.
- *Integración con calendario académico (8 días)*: Es una funcionalidad que nunca he desarrollado, por lo que necesitare tiempo para investigar, encontrar las tecnologías adecuadas y aprenderlas.

No funcionales:

- *Seguridad (8 días)*: Se requiere conocimientos en tecnologías y buenas prácticas de ciberseguridad, como autenticación (JWT o OAuth), encriptación (bcrypt o Argon2) y validación. Son tecnologías y conceptos que desconozco por lo que necesitare tiempo para estudiarlos.
- *Rendimiento (5 días)*: afortunadamente react facilita este aspecto, sin embargo se necesita optimizar al máximo las respuestas con las apis y dependiendo la cantidad de datos/consultas será necesario implementar paginación, por lo que es necesario estudiarlo.
- *Mantenibilidad (3 días)*: Es importante manejar buenas prácticas durante todo el desarrollo, gracias a que me estoy tomando el tiempo de estudiar las tecnologías este punto no debería tomar mucho tiempo.
- *Disponibilidad (5 días)*: será necesario aprender a manejar servicios como AWS, azure o google cloud para garantizar disponibilidad de los servicios.

Estimación personal (Cristian Steven Motta Ojeda)

Funcionales:

- *Registro e inicio de sesión (3 días)*: A pesar de que el requerimiento es relativamente sencillo teniendo en cuenta las tecnologías y librerías que hay en el mercado, este podría tornarse más complicado si se agrega autenticación con terceros.
- *Lista de contactos dinámica (3 días)*: Se debe realizar un ajuste a la base de datos para manejar las consultas y mantener la información de manera adecuada. Además, se necesita una interfaz adecuada para que el manejo por el usuario sea cómodo.
- *Chat individual (5 días)*: Este requerimiento incluye establecer comunicación en tiempo real mediante WebSockets u otro protocolo similar, además de diseñar una interfaz sencilla pero funcional. El tiempo también contempla el manejo de almacenamiento de mensajes en la base de datos y garantizar que la experiencia del usuario sea fluida.
- *Chat grupal (5 días)*: El chat grupal agrega complejidad porque, además del envío y recepción de mensajes en tiempo real, implica gestionar la creación de grupos, roles de usuarios, etc. También se necesita optimizar la entrega simultánea de mensajes a varios usuarios.
- *Historial de mensajes (5 días)*: La funcionalidad requiere consultas eficientes a la base de datos para recuperar mensajes antiguos. Es importante optimizar la carga inicial y añadir paginación para evitar problemas de rendimiento en chats con historiales extensos.
- *Notificaciones en tiempo real (3 días)*: Esto puede lograrse mediante servicios como Firebase Cloud Messaging o herramientas similares, asegurando que las notificaciones sean rápidas y confiables.
- *Búsqueda de mensajes y contactos (3 días)*: Este requerimiento necesita optimizar consultas para permitir búsquedas rápidas en la base de datos, utilizando índices en campos clave. La interfaz debe facilitar la interacción del usuario, como mediante un cuadro de texto que muestre resultados filtrados dinámicamente.
- *Integración con calendario académico. (5 días)*: Esta funcionalidad consiste en conectar el sistema con un calendario o crear uno para gestionar eventos académicos. También implica sincronizar los datos en ambas direcciones, lo cual puede ser un reto si se manejan varias fuentes de información.

No funcionales:

- *Seguridad (8 días):* Se requiere implementar cifrado de datos, validaciones y mecanismos de autenticación seguros. Además, es importante proteger el sistema contra ataques comunes como XSS, SQL Injection y ataques de fuerza bruta, lo que puede llevar tiempo adicional para pruebas y ajustes.
- *Rendimiento (13 días):* Se deben optimizar consultas a la base de datos, implementar caché y realizar pruebas de carga para identificar cuellos de botella. También se consideran ajustes en el manejo de conexiones en tiempo real para asegurar que el sistema funcione bien cuando haya un alto flujo de usuarios.
- *Mantenibilidad (8 días):* Esta tarea incluye escribir documentación detallada, implementar pruebas unitarias y estructurar el proyecto para facilitar futuras modificaciones. También es necesario organizar el código de manera que sea comprensible y reutilizable.
- *Disponibilidad (8 días):* Garantizar la disponibilidad requiere configurar redundancia básica, como balanceo de carga y copias de seguridad automáticas. También se incluyen configuraciones para monitorizar el sistema en tiempo real y minimizar el tiempo de inactividad mediante alertas rápidas y acciones correctivas.

Estimación personal (Sebastian Andrade Cedano)

Funcionales:

- *Registro e inicio de sesión (1 día):* Dado que ya tengo experiencia desarrollando sistemas de inicio de sesión, puedo implementar este requerimiento en un día. Considero que puedo manejar tanto el registro como la autenticación básica sin complicaciones mayores.
- *Lista de contactos dinámica (2 días):* La lista de contactos debe actualizarse en tiempo real para reflejar cambios como nuevos registros o estados de conexión. Aunque tengo experiencia en bases de datos, la sincronización en tiempo real requerirá investigar e implementar tecnologías como WebSocket o polling eficiente.
- *Chat individual (5 días):* Implementar un chat en tiempo real con tecnologías como WebSocket requiere lógica tanto en frontend como en backend. Aunque entiendo los conceptos básicos, no tengo experiencia directa con estas tecnologías, así que probablemente necesite tiempo adicional para aprender y probar.

- *Chat grupal (8 días):* Un chat grupal es más complejo que el individual, ya que requiere lógica adicional para manejar múltiples usuarios en una misma sala. Al no haber desarrollado esta funcionalidad antes, necesitareé investigar y realizar pruebas adicionales.
- *Historial de mensajes (1 día):* Guardar mensajes en la base de datos y permitir su recuperación no debería ser demasiado complejo para mí, pero podría necesitar explorar buenas prácticas para organizar los datos, como el uso de índices o estructuras eficientes.
- *Notificaciones en tiempo real (3 días):* No tengo experiencia previa con sistemas de notificaciones en tiempo real. Implementar esta funcionalidad requerirá integrar servicios como Firebase Cloud Messaging o desarrollar un sistema propio con WebSocket, lo cual podría tomar tiempo adicional debido a la necesidad de aprendizaje y pruebas.
- *Búsqueda de mensajes y contactos (2 días):* La funcionalidad requiere diseñar un sistema de búsqueda eficiente en la base de datos. Aunque tengo conocimientos de bases de datos, es posible que necesite investigar cómo optimizar consultas para evitar problemas de rendimiento.
- *Integración con calendario académico. (2 días):* La integración probablemente implica mostrar eventos del calendario y sincronizar información en la aplicación. Si se utiliza una API externa, como Google Calendar, necesitareé tiempo para aprender su uso y realizar las integraciones necesarias en el backend y frontend.

No funcionales:

- *Seguridad (2 días):* Sé manejar aspectos básicos como cifrar contraseñas o usar HTTPS, pero implementar medidas más avanzadas como protección contra ataques XSS o CSRF requerirá investigar y realizar pruebas exhaustivas.
- *Rendimiento (8 días):* Optimizar el rendimiento implica ajustar consultas, minimizar el tiempo de respuesta y posiblemente implementar caching. Esto requiere análisis y pruebas, lo cual podría tomar más tiempo debido a mi limitada experiencia en optimización avanzada.
- *Mantenibilidad (1 día):* Configurar herramientas para facilitar el mantenimiento, como la creación de logs o la documentación del código, es algo que puedo manejar fácilmente en un día, ya que es un proceso principalmente organizativo.
- *Disponibilidad (5 días):* Garantizar la alta disponibilidad implica implementar redundancia, monitoreo y manejar caídas del sistema. Esto está fuera de mi experiencia actual, por lo que necesitareé tiempo significativo para aprender e implementar soluciones básicas.

Estimación grupal

Vamos a realizar una tabla donde recopilamos las estimaciones de tiempo de cada persona para cada funcionalidad y aspecto no funcional del proyecto. Luego calcularemos el promedio de estas estimaciones para tener una visión general y llegar a un acuerdo sobre el tiempo que dedicaremos a cada tarea. Esto nos permitirá ajustar los tiempos de manera equilibrada.

Requerimiento	Esteban	Cristian	Sebastian	Promedio
Registro e inicio de sesión	8	3	1	4
Lista de contactos dinámica	3	3	2	3
Chat individual	5	5	2	4
Chat grupal	5	5	3	4.3
Historial de mensajes	3	5	1	3
Notificaciones en tiempo real	3	3	3	3
Búsqueda de mensajes y contactos	3	3	1	2.3
Integración con calendario académico	8	5	2	5
Seguridad	8	8	2	6
Rendimiento	5	13	8	8.6
Mantenibilidad	3	8	1	4
Disponibilidad	5	8	5	6
Estimación total	59	69	31	53.2

De este modo, concluimos que el tiempo estimado de desarrollo teniendo en cuenta a todos los miembros del equipo es de aproximadamente **53** días. El propósito de calcular el promedio es identificar las fortalezas y debilidades de cada individuo, de manera que aquellos que tengan más facilidades en ciertos aspectos puedan apoyar a quienes presenten carencias, fomentando así el trabajo en equipo y el desarrollo colaborativo del proyecto.