

**Tarea #03**  
**Requerimientos de usuario**

**Presentado por Grupo 6:**

Ana María González Hernández - [anagonzalezhe@unal.edu.co](mailto:anagonzalezhe@unal.edu.co)  
Daniel Felipe Soracipa - [dsoracipa@unal.edu.co](mailto:dsoracipa@unal.edu.co)  
Juan José Medina Guerrero - [jmedinagu@unal.edu.co](mailto:jmedinagu@unal.edu.co)  
Samuel Josué Vargas Castro - [samvargasca@unal.edu.co](mailto:samvargasca@unal.edu.co)

**Profesor:**  
Oscar Eduardo Alvarez Rodriguez  
[oalvarezr@unal.edu.co](mailto:oalvarezr@unal.edu.co)

Noviembre 28 de 2024



**Universidad Nacional de Colombia**  
**Facultad de Ingeniería**  
**Departamento de Ingeniería de Sistemas e Industrial**  
**2024**

## Análisis de requerimientos

### Priorización por Moscow

(M): Must have

(S): Should have

(C): Could have

(W): Won't Have

- **Requerimientos funcionales:**

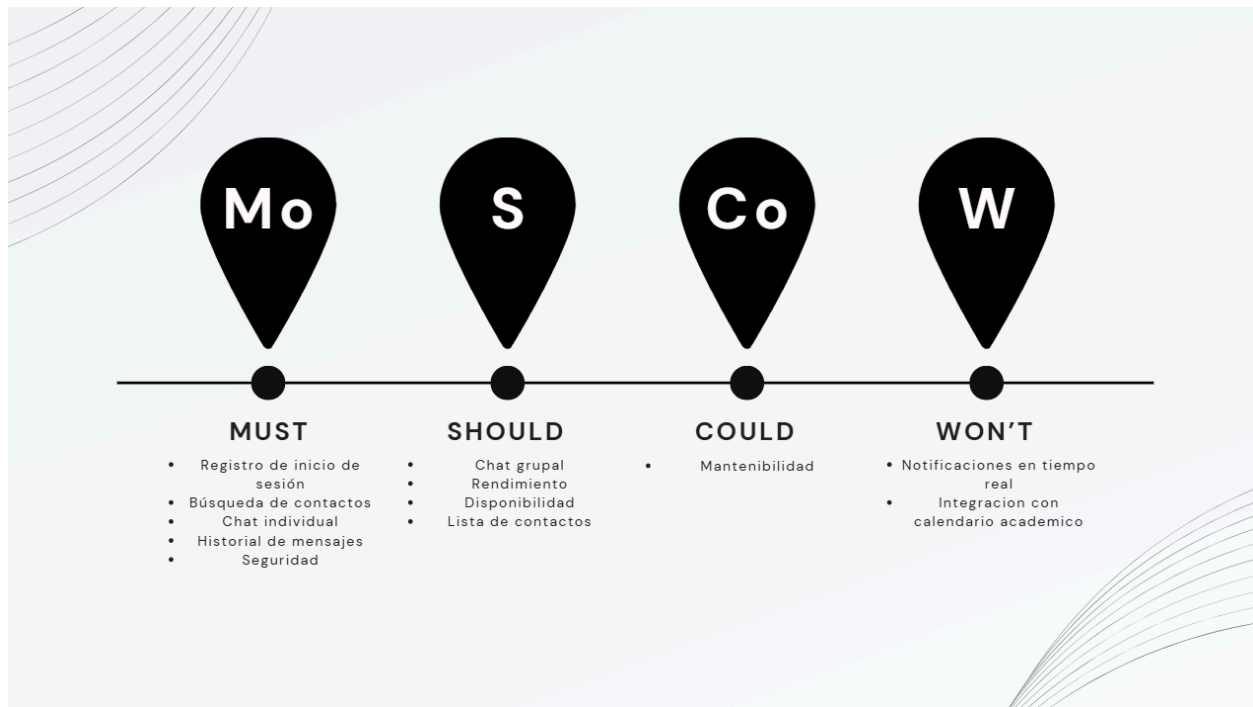
- Registro e inicio de sesión
  - (M)
- Lista de contactos dinámica
  - (M)
- Chat individual
  - (M)
- Chat grupal
  - (M)
- Historial de mensajes
  - (M)
- Notificaciones en tiempo real
  - (S)
- Búsqueda de mensajes y contactos
  - (C)
- Integración con calendario académico
  - (C)

- **Requerimientos no funcionales:**

- Seguridad (M)
- Rendimiento (S)
- Mantenibilidad (S)
- Disponibilidad (M)

Según el grupo:

## Estimación



Con base en la priorización realizada por el grupo expositor, se realiza ahora la estimación de tiempos mediante la estrategia planning poker, obteniendo el tiempo promedio entre los estudiantes del grupo.

Un mazo típico contiene tarjetas con sucesivos términos de la sucesión de Fibonacci incluyendo un cero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 (días)

### Convenciones:

- (AG) Ana González
- (DS) Daniel Soracipa
- (JM) Juan Medina
- (SV) Samuel Vargas

### ● Must

- Registro e inicio de sesión

#### ■ (AG): 3

El requerimiento es sencillo pero puede ser ligeramente más complejo dependiendo de la integración con sistemas externos o validaciones adicionales, por lo que, creo que 3 días sería suficiente.

#### ■ (DS): 2

El registro e inicio de sesión se puede armar rápido usando servicios como Firebase o AWS, que ya tienen sistemas listos para autenticar usuarios. Solo sería cuestión de integrarlo con las configuraciones iniciales del proyecto

- (JM): 3  
El requerimiento de inicio de sesión debido a las plataformas actuales es posible realizarlo en este corto tiempo integrando servicios como Firebase , Google Cloud, AWS, entre otros donde facilitan la creación de servicios de login.
- (SV): 3  
El requerimiento por sí solo es posible realizarlo en un corto tiempo utilizando servicios especializados para la autenticación tales como Firebase. Sin embargo, debido a que es el primer paso en el proyecto, también conlleva la realización de todas las configuraciones iniciales de las tecnologías utilizadas en el mismo.
- Búsqueda de contactos
  - (AG): 2  
El requerimiento probablemente esté basado en una búsqueda directa en la base de datos y requiera de un diseño de interfaz mínimo.
  - (DS): 2  
La búsqueda de contactos es sencilla: basta con consultar la base de datos filtrando según los roles de usuario y diseñar una interfaz básica que permita buscar por nombre o correo
  - (JM): 2  
Consiste en implementar funcionalidades para agregar, eliminar y actualizar contactos, junto con la sincronización de estos datos en el frontend y backend.
  - (SV): 3  
El requerimiento incluye la creación de la base de datos y la diferenciación de usuarios según roles de profesor o estudiante. Sumado a ello, es necesario establecer la información que se necesita para cada usuario, como el nombre o correo electrónico.
- Chat individual
  - (AG): 5  
La gestión de los mensajes en ambos lados (cliente y servidor) y la comunicación en tiempo real implicarían más tiempo de desarrollo.
  - (DS): 5  
Para el chat individual se necesita manejar bien los mensajes en tiempo real y asegurarse de que la base de datos tenga la arquitectura correcta para almacenar conversaciones entre usuarios.
  - (JM): 3  
plica establecer comunicación en tiempo real (usando WebSockets o alternativas), almacenar mensajes en la base de datos y diseñar una interfaz básica para el chat.
  - (SV): 2  
El chat individual planteado entre distintos usuarios debe poderse comunicar mediante el acceso a la base de datos, sin embargo, se debe gestionar la arquitectura que ocuparán dentro de la base de datos.

- Historial de mensajes

- (AG): 5

Aunque el requerimiento esté muy vinculado con lo desarrollado en el “Chat individual”, requiere de un almacenamiento eficiente de los mensajes en la base de datos y su respectiva recuperación para mostrarlos en la interfaz. Además dependiendo de los volúmenes de mensajes, se podría requerir un poco más de tiempo en la implementación de mejores técnicas para optimizar el rendimiento.

- (DS): 5

El historial de mensajes depende del chat individual. Solo se necesita implementar cómo mostrar los mensajes guardados, cargando los más recientes primero para que la experiencia sea fluida

- (JM): 3

Requiere consultas eficientes a la base de datos para recuperar mensajes con funcionalidades como paginación o carga progresiva.

- (SV): 2

El historial de mensajes depende directamente de lo desarrollado en el requerimiento anterior, específicamente del almacenamiento de los mensajes enviados entre usuarios. Esto solamente requeriría entonces el manejo por el lado del cliente de estos datos y su carga adecuada en el chat (cargar inicialmente los más recientes para que el usuario acceda a lo más importante en el menor tiempo posible).

- Seguridad

- (AG): 8

El requerimiento abarca múltiples aspectos como encriptación de mensajes, protección contra ataques o manejo de tokens. Dependiendo del nivel de seguridad requerido, también se podrían mirar medidas adicionales como conexiones HTTPS o una revisión de la arquitectura o código, por lo que, considero que ocho días sería un tiempo prudente.

- (DS):5

La seguridad requiere encriptar los mensajes tanto en la base de datos como en su transmisión. Podemos usar herramientas ya existentes para proteger las comunicaciones y reforzar las conexiones.

- (JM): 5

Considera la implementación de validaciones, cifrado de datos sensibles y protección contra ataques como XSS o CSRF.

- (SV): 5

El requerimiento implica que las comunicaciones entre usuarios no sean inseguras. Por ello, es necesario la implementación de encriptación de mensajes en la base de datos y en su transmisión, para lo cual se aprovecharán servicios en línea sobre esta línea protectiva.

- **Should**

- Chat grupal

- (AG): 5

Con base en el “Chat individual” el tiempo de desarrollo se puede reducir

un poco, sin embargo, como el chat grupal debe gestionar múltiples usuarios en la conversación, ciertas funciones adicionales como la asignación de roles o el trabajo en la sincronización adecuada requerirán algo de tiempo.

- (DS): 5

El chat grupal usa como base el individual, pero hay que gestionar varios usuarios a la vez, manejar roles como admin o moderador, y ajustar cómo se almacena y sincroniza la información en la base de datos.

- (JM): 3

Es una extensión del chat individual que agrega complejidad al gestionar grupos, roles de usuarios (admin, miembros), y sincronización de mensajes.

- (SV): 5

En este requerimiento se aprovecha como base lo desarrollado en el procedimiento de chat grupal. Sin embargo, es necesario diseñar la arquitectura que tendrá su almacenamiento en la base de datos y el método de acceso de los usuarios a su información. Además, es posible que se presente la necesidad de insertar nuevos roles según el estatus en el grupo, como moderador o administrador.

- Rendimiento

- (AG): 13

Mejorar el rendimiento implica analizar toda la arquitectura de la aplicación, desde las consultas que se realizan a la base de datos hasta el tiempo de respuesta del servidor o el rendimiento de la interfaz. Identificar los puntos en los que se requiere un mejor rendimiento, además de las estrategias de optimización, su respectiva implementación y pruebas extensivas requieren de más tiempo a comparación de los anteriores requerimientos.

- (DS): 8

Optimizar el rendimiento implica revisar todo: desde consultas en la base de datos hasta tiempos de respuesta. Se deben ajustar las estrategias para que los mensajes lleguen rápido y sin fallos, especialmente en tiempo real.

- (JM): 8

Se enfoca en optimizar consultas, manejar concurrencia y configurar caché para mejorar la velocidad.

- (SV): 8

En el desarrollo de este requerimiento se busca la sincronización en tiempo real de los mensajes y una respuesta rápida en la comunicación. Por ello, es necesario replantear alternativas a las utilizadas inicialmente para mejorar la velocidad de transferencia de mensajes. Sin embargo, considerando un diseño adecuado en los requerimientos anteriores, esta transición debe ser correctamente manejada sin ningún tipo de retraso.

- Disponibilidad

- (AG): 8

La disponibilidad es un trabajo constante que dependiendo de la

plataforma y cantidad de usuarios requiere de diferentes herramientas y procedimientos para ser elaborado.

- (DS):8

Para garantizar disponibilidad constante, toca buscar alternativas más robustas para servidores y bases de datos, evaluando su costo y escalabilidad para mantener el servicio siempre activo.

- (JM): 5

Cubre la configuración de un entorno robusto en producción, con despliegue en la nube, balanceo de carga y sistemas de respaldo.

- (SV): 3

En este requerimiento se busca un servicio constante el 99% del tiempo, por ello es necesario buscar alternativas a los servicios de comunicación y de base de datos utilizados. Esto conlleva analizar distintas opciones desde el punto de vista desarrollador y financiero.

- Lista de contactos

- (AG):3

Diseñar e implementar una lista de contactos es relativamente sencillo, especialmente si se basa en una base de datos existente.

- (DS): 3

La lista de contactos se construye sobre la base de datos, permitiendo agregar usuarios manualmente o con sincronización automática según las relaciones académicas, como alumnos y profesores de un mismo grupo.

- (JM): 5

Consiste en implementar funcionalidades para agregar, eliminar y actualizar contactos, junto con la sincronización de estos datos en el frontend y backend.

- (SV): 8

En este requerimiento se plantea que los estudiantes puedan agregar chats con otros usuarios, además de que los profesores tengan de manera automática chats directos con sus estudiantes actuales y anteriores. Con base en ello, es necesario almacenar las listas de contactos en la base de datos y mantener una comunicación con la información académica de los grupos de las materias.

- **Could**

- Mantenibilidad

- (AM): 8

En este caso, siendo un proyecto relativamente pequeño requerirá de una refactorización en las etapas finales del desarrollo para poder asegurar una buena calidad en el código y que sea lo suficientemente robusto para perdurar en el tiempo.

- (DS):--

La mantenibilidad del código es algo que se asegura a lo largo del desarrollo, no es algo que se pueda definir en días específicos, ya que depende de cómo se mantenga limpio y organizado el código mientras avanzamos en el proyecto

- (JM): 8  
Documentación básica, estructuración del proyecto, y configuración de pruebas unitarias y de integración.
  - (SV): Paralelo  
Debido a que la mantenibilidad del código es consecuencia directa del trabajo del desarrollador a lo largo del proyecto, es necesario mantener de manera paralela este requerimiento, el cual no es posible estimar en días adecuadamente.
- **Won't**
  - Notificaciones en tiempo real
    - (AM): 3  
Las notificaciones de tiempo real requieren establecer un sistema que permita enviar eventos desde el servidor al cliente de manera eficiente pero no son una tarea realmente muy difícil de completar.
    - (DS): 3  
Para que las notificaciones lleguen al instante, hay que montar un sistema que esté atento a cualquier movimiento en los chats y mande un aviso rápido a los usuarios. No es muy complicado, pero sí hay que ajustar algunos detalles en la app para que se puedan silenciar chats o personalizar las alertas.
    - (JM): 3  
Involucra la implementación de servicios como WebSockets o Firebase Cloud Messaging, tanto en el backend como en el frontend.
    - (SV): 5  
Para este requerimiento es necesario realizar escuchas directas a cada uno de los chats para que en caso de que se realice una nueva adición a la base de datos de mensajes, se informe a los dispositivos de todos los usuarios pertenecientes al chat. Además, es necesario modificar la interfaz de usuario y las configuraciones locales de la aplicación para que sea posible silenciar chats específicos.
  - Integración con calendario académico
    - (AM): 5  
Este requerimiento probablemente implica conectar la aplicación con una API de calendario (como Google Calendar) para sincronizar eventos académicos. Aunque es un desarrollo relativamente directo, depende de la complejidad de la API y de los datos que se desean integrar.
    - (DS): 2  
Lo del calendario académico es sencillo: solo toca jalar los datos de las clases y eventos desde la base de datos y mostrarlos en la app. La parte extra sería incluir un calendario en la interfaz, pero no debería ser algo que nos tome demasiado tiempo
    - (JM): 3  
Requiere conectar el sistema con APIs externas como Google Calendar o Microsoft Calendar, manejar autenticación (OAuth2) y sincronizar eventos.



■ (SV): 13

Es necesario agregar una funcionalidad completamente nueva de calendario para que sea posible acceder a la información de horarios y materias dentro de la aplicación. Además, es necesario acceder a la base de datos académica para obtener la información de los usuarios y sus respectivas clases. Por último, es necesario agregar la posibilidad de que los profesores añadan información de actividades de la materia

Con base en las estimaciones individuales, se realizó un promedio grupal para decidir entonces la longitud final estimada para cada requerimiento y el tiempo total del proyecto.

Estimación de requerimientos					
Requerimiento	AG	DS	JM	SV	Promedio
Registro e inicio de sesión	3	2	3	3	2,75
Búsqueda de contactos	2	2	2	3	2,25
Chat individual	5	5	3	2	3,75
Historial de mensajes	5	5	3	2	3,75
Seguridad	8	5	5	5	5,75
Chat grupal	5	5	3	5	4,50
Rendimiento	13	8	8	8	9,25
Disponibilidad	8	8	5	3	6,00
Lista de contactos	3	3	5	8	4,75
Mantenibilidad	8	--	8	Paralelo	Paralelo
Notificaciones en tiempo real	3	3	3	5	3,50
Integración con calendario académico	5	2	3	13	5,75
<b>Total Estimación</b>	68	48	51	57	52,00

De esta manera, el equipo realizó un cálculo para la duración del proyecto de aproximadamente 52 días, incluyendo todos los requerimientos.