



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - Sede Bogotá
Facultad de Ingeniería
Departamento de ingeniería química e ingeniería de sistemas
Curso: Ingeniería de Software

Estudiantes: Marlon David Pabon, Luis Gabriel Peraza, [Nicolas Gomez Lozano](#),
[Juan Camilo Rosero Santisteban](#)

Practica Analisis De Requerimientos Metodología MoSCoW

Previo a la elaboración de este archivo, se llevó a cabo una discusión detallada sobre los tiempos estimados y el nivel de importancia asignado a cada requerimiento. Este análisis se realizó considerando las diferentes perspectivas del equipo y las necesidades del proyecto. A continuación, se presenta un resumen que refleja un planteamiento aproximado de los requerimientos, integrando las opiniones y prioridades identificadas durante dicha discusión.

REQUERIMIENTOS FUNCIONALES

Must Have

1. Registro de inicio de sesión

Justificación: El registro de inicio de sesión es fundamental para garantizar la seguridad del sistema y asegurar que solo los usuarios autorizados accedan. Este requerimiento es crucial para cualquier aplicación que maneje datos sensibles o interactúe con múltiples usuarios, por lo que su implementación es indispensable desde el principio.

2. Lista de contactos dinámica

Justificación: La capacidad de tener una lista de contactos actualizada en tiempo real es esencial para el funcionamiento del sistema. Sin esta funcionalidad, la experiencia del usuario sería significativamente limitada, ya que los usuarios no podrían interactuar de forma eficiente. Esta característica es crítica para garantizar la interactividad de la aplicación.

3. Chat grupal

Justificación: El chat grupal es una de las funcionalidades clave de cualquier aplicación de mensajería. Permitir que varios usuarios se comuniquen simultáneamente es esencial para la usabilidad del sistema. Sin un chat grupal funcional, el sistema no cumpliría con uno de sus principales objetivos, por lo que debe ser implementado en la primera fase.

Should Have

4. Chat individual

Justificación: Aunque el chat individual es importante para la interacción entre usuarios, no es tan crítico como el chat grupal. El sistema aún podría funcionar sin esta funcionalidad, pero su ausencia afectaría la experiencia general. Debido a su importancia, se implementará después de los requerimientos "Must Have", pero debe estar disponible en una etapa temprana.

5. Notificaciones en tiempo real

Justificación: Las notificaciones en tiempo real mejoran la experiencia del usuario al mantenerlo informado de eventos importantes, como nuevos mensajes. Aunque son útiles y mejoran la interacción, no son esenciales para el funcionamiento básico del sistema, por lo que se consideran de menor prioridad respecto a otras funcionalidades más centrales.

Could Have

6. Historial de mensajes

Justificación: El historial de mensajes es una característica que mejora la experiencia del usuario, permitiéndole acceder a conversaciones anteriores. Sin embargo, no es una funcionalidad esencial en las primeras etapas del proyecto. Aunque es deseable, puede posponerse para una fase posterior sin afectar la operatividad básica del sistema.

Would Have

7. Búsqueda de mensajes y contactos

Justificación: La búsqueda de mensajes y contactos, aunque útil, no es una prioridad para el lanzamiento inicial del sistema. Si bien mejoraría la usabilidad, su implementación no es crítica para la funcionalidad básica del sistema. Este requerimiento puede ser considerado en fases posteriores del proyecto.

8. Integración con calendario académico

Justificación: La integración con el calendario académico proporciona valor agregado al sistema, pero no es un requisito esencial en la fase inicial. Si bien permite a los usuarios gestionar mejor sus actividades, su ausencia no interfiere con las funcionalidades principales de la aplicación.

REQUERIMIENTOS NO FUNCIONALES

Must Have

1. Seguridad

Justificación: La seguridad es una prioridad absoluta para cualquier sistema que maneje información sensible. La protección de datos y el control de acceso son fundamentales para garantizar la confianza de los usuarios y evitar posibles vulnerabilidades. Sin medidas de seguridad adecuadas, el sistema no podría operar de manera segura ni cumplir con los estándares requeridos.

Should Have

2. Disponibilidad

Justificación: La alta disponibilidad es esencial para que los usuarios puedan acceder al sistema en cualquier momento. Aunque no es necesario que el sistema esté disponible al 100% sin interrupciones, debe garantizarse un

tiempo de inactividad mínimo y estar preparado para manejar fallos sin afectar la experiencia del usuario.

Could Have

3. Rendimiento

Justificación: Un rendimiento adecuado es importante para asegurar que el sistema responda rápidamente y gestione eficientemente a un número alto de usuarios. Sin embargo, en fases iniciales, un rendimiento promedio será suficiente, siempre que se realicen pruebas y optimizaciones a medida que el sistema se desarrolle.

Would Have

4. Mantenibilidad

Justificación: Si bien la mantenibilidad es importante a largo plazo, no es esencial en las fases iniciales del proyecto. Se garantiza que el código sea modular y escalable, pero la atención plena a la mantenibilidad puede posponerse hasta que el sistema esté más consolidado y funcionando correctamente.

* Redacción mejorada con IA