



**Tarea 3**  
**Universidad Nacional de Colombia**  
**Ingeniería Software**



Tarea 3

Cristian Felipe Moreno Gómez

Diego Alejandro Rojas Reina

Edinson Sanchez Fuentes

Maicol Sebastián Olarte Ramírez

Ingeniería de Software I

Ingeniería de sistemas y computación

Facultad de Ingeniería

Universidad Nacional de Colombia

Bogotá, Colombia

2024



**Tarea 3**  
**Universidad Nacional de Colombia**  
**Ingeniería Software**



— **Prioridad planteada por el grupo 5.**

- **Requerimientos funcionales**

Registro e Inicio de Sesión (*must have*)

Lista de Contactos Dinámica (*must have*)

Chat Grupal (*must have*)

Chat Individual (*should have*)

Historial de Mensajes (*should have*)

Notificaciones en Tiempo Real (*should have*)

Búsqueda de Mensajes y Contactos (*could have*)

Integración con Calendario Académico (*could have*)

**Requerimientos no funcionales.**

Seguridad (*must have*)

Disponibilidad (*must have*)

Rendimiento (*should have*)

Mantenibilidad (*should have*)

— **Prioridad de requerimientos, corrección en clase.**

**FUNCIONALES**

REGISTRO E INICIO DE SESIÓN (*must have*)

HISTORIAL DE MENSAJES (Individual y grupal) (*must have*)

CHAT GRUPAL (*must have*)

BÚSQUEDA DE MENSAJES Y CONTACTOS (*must have*)

CHAT INDIVIDUAL (*should have*)

LISTA DE CONTACTOS DINÁMICA (*should have*)

NOTIFICACIONES EN TIEMPO REAL (*won't have*)

INTEGRACIÓN CON CALENDARIO ACADÉMICO (*won't have*)

**NO FUNCIONALES**

SEGURIDAD (*must have*)

DISPONIBILIDAD (*should have*)

RENDIMIENTO (*should have*)

MANTENIBILIDAD (*could have*)



— **Resumen de la Estimación Total en Días (Escala Fibonacci):**

**Must Have:**

- *Registro e inicio de sesión: 5 días*

Teniendo en cuenta que el inicio y registro de sesión involucra un diseño de base de datos, un pequeño frontend y backend, considerando que, las pautas y conocimientos sobre el tema, este tiempo nos parece razonable.

- *Historial de mensajes: 5 días*

Este ítem al ser fundamental para que sobreviva una red social, lo consideramos de gran cuidado, además, se debe tener en cuenta que se necesita un historial de mensajes grupal e individual, así que, es esencial hacer una buena consulta a la base de datos y testear que todo esté en correcto funcionamiento.

- *Chat grupal: 8 días*

Este ítem puede generar una dificultad considerable, debido a que es vital garantizar la interacción entre todos los miembros del respectivo grupo proporcionando una interfaz sencilla, fácil de entender, agradable

- *Búsqueda de mensajes y contactos: 3 días*

Esta tarea puede alcanzarse en un tiempo relativamente corto utilizando las consultas necesarias a la base de datos y conectando con el servidor y a su vez con un backend sencillo.

- *Seguridad: 8 días*

Al tener muy poca experiencia en este tema nos parece prudente colocar este tiempo para cumplir con requerimiento, considerando otro aspecto, este ítem es muy importante para la aplicación ya que de él depende primero el funcionamiento, segundo la seguridad tanto de los usuarios de aplicación, tanto como de los desarrolladores en cuestiones legales.

**Total Must Have = 5 + 5 + 8 + 3 + 8 = 29 días**

**Should Have:**

- *Chat individual: 5 días*

Hace uso de una interfaz que resulta sencilla por lo que la complejidad de realización no es muy alta, donde realmente requiere mayormente una inversión de tiempo es en la sincronización para visualizar el estado de los mensajes además de verificar los permisos ya que estos están asociados.

- *Lista de contactos dinámica: 5 días*



### Tarea 3

## Universidad Nacional de Colombia

### Ingeniería Software



Este módulo requiere de sincronizar a los usuarios por lo que nuevamente ocupa la base de datos para hacer llamadas de esta y tener actualizados los contactos por lo tanto su desarrollo no implica una complejidad demasiado alta.

- *Disponibilidad: 8 días*

Para esto se deben aplicar buenas técnicas de programación que incluye buen soporte haciendo mantenimientos en horas con mínimo tráfico para afectar en los menos posible la experiencia de los usuarios.

- *Rendimiento: 5 días*

Haciendo uso de técnicas eficientes que incluyen test y revisión de código previamente hecho para determinar y reducir los tiempos de carga, lo cual es importante pero no fundamental en una primera instancia.

**Total Should Have = 5 + 5 + 8 + 5 = 23 días**

**Could Have:**

- *Mantenibilidad: 5 días*

Es fundamental e implica documentar el código en pequeños bloques a medida que se construye en el sistema para que su revisión y posterior refactorización sea mucho más sencilla de manera que sea tanto entendible como legible.

**Total Could Have = 5 días**

**Won't Have:**

- *Notificaciones en tiempo real: 0 días*

No se considera esencial para el funcionamiento del sistema y se puede incorporar de forma que sea a uso opcional del usuario.

- *Integración con calendario académico: 0 días*

Esta parte si bien es un plus y algo muy interesante de implementar, no es esencial para el aplicativo, es decir, este se va a seguir manteniendo de la misma manera si está el ítem o no.

**Total, Won't Have = 0 días**



**Tarea 3**  
**Universidad Nacional de Colombia**  
**Ingeniería Software**



— **Estimación Total:**

**29 días (Must Have) + 23 días (Should Have) + 5 días (Could Have) = 57 días**

*Como conclusión esta estimación nos puede dar una idea sobre lo que se debe hacer y en qué tiempo, teniendo en cuenta su dificultad, claro está que depende de varios factores como por ejemplo el uso de x/y tecnología, la experiencia de los miembros del equipo, el tiempo disponible de cada uno de los integrantes, entre otras consideraciones.*