

**Universidad Nacional de Colombia**  
**Ingeniería de Software I**  
**2024-2**  
**Tarea #3**

Pablo Andres Niño Barreto - [pninob@unal.edu.co](mailto:pninob@unal.edu.co)

Jeisson Duvan Bareño Ruiz - [jbareno@unal.edu.co](mailto:jbareno@unal.edu.co)

Víctor Camilo Cañón Castellanos - [vcanonc@unal.edu.co](mailto:vcanonc@unal.edu.co)

Jacobo Alzate Corredor - [jalzateco@unal.edu.co](mailto:jalzateco@unal.edu.co)

**Matriz de priorización MosCoW**

<b>Must Have</b> <i>Esenciales para la viabilidad del producto</i>	<b>Should Have</b> <i>Importantes, pero con menor prioridad</i>	<b>Could Have</b> <i>Deseables, pero no esenciales</i>	<b>Won't Have</b> <i>Se pueden eliminar</i>
Integración con el sistema de gestión académica de la universidad para validar usuarios.	Interfaz sencilla para enviar y recibir mensajes.	Indicadores de "escribiendo", leído y entregado.	Recordatorios automáticos de clases o actividades asociadas a un curso.
Verificación de identidad por correo o SMS	Rendimiento	Filtros avanzados para localizar conversaciones o usuarios específicos.	Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.
Los estudiantes podrán añadir contactos mediante búsqueda por nombre o matrícula.	Disponibilidad	Mantenibilidad	
Los profesores podrán visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.			
Creación de grupos por estudiantes o profesores (según permisos).			
Gestión de grupos por el creador (añadir/eliminar miembros).			

Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.			
Seguridad			

### Estimación individual (usando Poker Planning)

Se utilizará la siguiente baraja para la estimación del esfuerzo necesario para implementar dichos requerimientos:

Unset

(0, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?, ∞, ☕)

### Víctor Camilo Cañón Castellanos

- Integración con el sistema de gestión académica de la universidad para validar usuarios.  
RTA/ 5, considero que la universidad ya debe contar con algún sistema de autenticación y/o ya cuenta incluso con credenciales para cada usuario.
- Verificación de identidad por correo o SMS.  
RTA/ 3, existen muchas librerías para el envío automatizado de emails, por lo que considero que puede ser de implementación sencilla.
- Los estudiantes podrán añadir contactos mediante búsqueda por nombre o matrícula.  
RTA/ 13, considero que la funcionalidad de búsqueda debe ser bien implementada para que tenga un buen rendimiento teniendo en cuenta que se va a contar con una base de datos no relacional.
- Los profesores podrán visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.  
RTA/ 20, dado que no conozco cómo crear y gestionar bases de datos no relacionales, creo que sería prudente primero aprender a hacer buenos modelos en estás para que este no sea un lastre para el rendimiento del sistema.
- Interfaz sencilla para enviar y recibir mensajes.  
RTA/ 5, ya existen muchas “plantillas” y frameworks que pueden hacer este trabajo un poco más sencillo y atractivo estéticamente.
- Indicadores de "escribiendo", leído y entregado.  
RTA/ 5, al igual que el anterior, ya existen muchas formas para implementar chats con buena estética y “accesibilidad” para el usuario.
- Creación de grupos por estudiantes o profesores (según permisos).  
Rta/ 13, al igual que con el requerimiento #4 creo que me tomaría un poco de tiempo

para implementarlo. Aunque está vez menos suponiendo que ya habría diseñado el modelo que tendrá el sistema.

8. Gestión de grupos por el creador (añadir/eliminar miembros).  
RTA/ 5, una vez realizada la base de datos, la creación y eliminación de datos será sencilla de programar.
9. Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.  
RTA/ 13 o 20, creo que en este caso, como se guardan tantos mensajes en un chat sería necesario hacer testeos para verificar que se tiene un tiempo de respuesta aceptable.
10. Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.  
RTA/ 5, creo que sería de fácil implementación ya que solo es necesario un buen manejo de las notificaciones tomando como base las preferencias del usuario que se guardan en la BD.
11. Filtros avanzados para localizar conversaciones o usuarios específicos.  
RTA/ 20, La implementación de filtros avanzados requiere de un buen uso de los métodos de optimización que tenga la base de datos y el cómo el back-end interactúa con esta.
12. Recordatorios automáticos de clases o actividades asociadas a un curso.  
RTA/ 13, los recordatorios estarían relacionados con un calendario lo que haría un poco más grande el esfuerzo para realizar la implementación.
13. Seguridad.  
RTA/ 5, realmente hoy en día existen muchas librerías para la encriptación de contraseñas, dependerá un poco del nivel de seguridad, pero como no es necesario reinventar la rueda creo que puede ser sencillo.
14. Rendimiento.  
RTA/ 3, si se tuvo en cuenta este apartado en todo el desarrollo del resto de requerimientos... Considero que si se trabajó teniendo el enfoque en tener un buen rendimiento.
15. Mantenibilidad.  
RTA/ 8, creo que la mantenibilidad puede ser un poco dispendioso de llevar a cabo en un software, pero eso asegura que el sistema no dependa de un solo programador o empresa, lo que permite agregar más funcionalidades si se requiere de forma fácil, limpia y segura con la integridad de la aplicación. Le pongo 8 porque considero que es algo con lo que ya se debería ser consciente y procurando realizar buenas prácticas en el código y documentación asegurando la mejor calidad para que otros programadores puedan trabajar en el futuro sobre el proyecto.
16. Disponibilidad.  
RTA/ 13, para mantener el sistema funcionando se requiere de formas de monitorearlo constantemente, y eso sin tener en cuenta el cómo afectan otros requerimientos a este, como lo podría ser el rendimiento y la seguridad.

### **Jeisson Duvan Bareño Ruiz**

1. Integración con el sistema de gestión académica de la universidad para validar usuarios.  
RTA/ 8 Debería ser una tarea que no maneje mucho tiempo ya que la universidad deberá tener su propio software y conectar este con nuestro sistema no debería ser muy complejo
2. Verificación de identidad por correo o SMS.  
RTA/ 5 Esta es una tarea sencilla, el mandar correos automatizados es tener una plantilla y generar algunos códigos, en el caso de los SMS debería ser una tarea similar
3. Los estudiantes podrán añadir contactos mediante búsqueda por nombre o matrícula.  
RTA/ 8 El añadir contactos no es una tarea muy difícil, es solo buscar en la base de datos y relacionar a las personas que se necesiten y la implementación de la búsqueda no debería ser tan demorada
4. Los profesores podrán visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.  
RTA/ 13 Requiere de una buena estructura organizada para relacionar a los profesores con sus clases actuales y anteriores, además de agregar a otros profesores que no es una ardua tarea
5. Interfaz sencilla para enviar y recibir mensajes.  
RTA/ 8 La interfaz es sencillo, el enviar y recibir mensajes es la parte compleja de esta implementación esta requiere de usar sincronización de mensajes entre usuarios
6. Indicadores de "escribiendo", leído y entregado.  
RTA/ 5 Esta tarea suena muy sencilla pero lo difícil es hacer que esa sea en tiempo real y que no aparezca el escribiendo cuando ya se enviaron los mensajes
7. Creación de grupos por estudiantes o profesores (según permisos).  
RTA/ 5 No debería ser una tarea que exija mucho tiempo, esta implementación es sencilla
8. Gestión de grupos por el creador (añadir/eliminar miembros).  
RTA/ 3 El añadir y eliminar miembros de los grupos creados debería ser una tarea bastante sencilla, requiere sincronización entre el frontend y el backend
9. Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.  
RTA/ 8 La búsqueda de palabras claves puede ser un poco difícil pero si se implementan tipos de búsquedas eficientes se puede hacer una tarea sencilla
10. Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.  
RTA/ 5 Esta implementación no debería ser un desafío ya que estas características no son complicadas, si se necesita una asociación con las alertas y hay que saber manejar esa parte

11. Filtros avanzados para localizar conversaciones o usuarios específicos.  
RTA/ 8 Para estos filtros se deben hacer múltiples pruebas ya que es común que si algo pequeño falla no se encuentre lo que se está buscando, si se hace organizadamente no habría problemas
12. Recordatorios automáticos de clases o actividades asociadas a un curso.  
RTA/ 5 Esto requiere de sincronización y recordatorios, estas tareas depende de cómo se vayan a implementar pueden ser más sencillas o más complicadas pero en general no serían problema
13. Seguridad.  
RTA/ 13 Depende del nivel de seguridad que se quiera incluir, pero si se usa muchos protocolos de seguridad o cifrado de extremo a extremo en los mensajes, o servicios extra la tarea se puede poner compleja
14. Rendimiento.  
RTA/ 13 Con las implementaciones habría que tener el código muy bien estructurado ya que entre más funcionalidades es más difícil que el rendimiento no se vea afectado
15. Mantenibilidad.  
RTA/ 8 Hay que estar revisando ciertas cosas para que esté sistemas de gestión se mantenga perfecto, pero no son pocas cosas, tal vez surjan errores ya que es muy común
16. Disponibilidad.  
RTA/ 8 Que el sistemas esté disponible para usarse por mucho tiempo y todo el tiempo, también depende de que tal está el rendimiento de este, si el sistemas es ligero es más probable que esté disponible siempre que se quiera usar

### **Pablo Andres Niño Barreto**

1. Integración con el sistema de gestión académica de la universidad para validar usuarios.  
RTA/ 13, la integración con un sistema externo suele implicar desafíos como compatibilidad de APIs, seguridad, y validación de datos. Además, podría requerir personalización según el sistema académico de la institución..
2. Verificación de identidad por correo o SMS.  
RTA/ 8, aunque existen servicios de terceros que facilitan esta funcionalidad, implementarlo correctamente con seguridad y manejo de errores puede requerir cierto esfuerzo.
3. Los estudiantes podrán añadir contactos mediante búsqueda por nombre o matrícula  
RTA/ 5, el reto principal sería implementar una búsqueda eficiente en la base de datos. Si el sistema ya está diseñado para manejar usuarios, esta tarea sería relativamente sencilla
4. Los profesores podrán visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.  
RTA/ 8, esta funcionalidad requiere lógica compleja para poder clasificar a los estudiantes y con eso los profesores pueda visualizarlos, así el cómo poder hacer que se agreguen manualmente los mismos

5. Interfaz sencilla para enviar y recibir mensajes.  
RTA/ 8, aunque el diseño de la interfaz sea básico, el tener la opción de envío y recepción de mensajes aumenta la complejidad.
6. Indicadores de "escribiendo", leído y entregado.  
RTA/ 13, la implementación de esta función requiere sincronización y actualizaciones en tiempo real, lo cual tiene un nivel de complejidad alto.
7. Creación de grupos por estudiantes o profesores (según permisos).  
RTA/ 5, para tener un manejo de roles se necesita la implementación de roles, teniendo esto ya implementado, el trabajo adicional no sería exagerado.
8. Gestión de grupos por el creador (añadir/eliminar miembros).  
RTA/ 5, la gestión de los grupos tampoco implica una dificultad alta, requiere tener un manejo de permisos por medio del rol del líder.
9. Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.  
RTA/ 13, almacenar los mensajes no es una tarea compleja, pero implementar búsqueda de mensajes y que sea eficiente, requiere de un desafío técnico mayor.
10. Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.  
RTA/ 8, implementar el manejo de notificaciones en los chats, representa un desafío técnico/lógico mayor.
11. Filtros avanzados para localizar conversaciones o usuarios específicos.  
RTA/ 13, los filtros avanzados exigen una eficiencia y optimización de bases de datos, de forma que los resultados sean rápidos y precisos.
12. Recordatorios automáticos de clases o actividades asociadas a un curso.  
RTA/ 8, El enviar recordatorios de reuniones/clases programadas es posible gracias a herramientas modernas, aunque requiere de la integración con calendarios.
13. Seguridad.  
RTA/ 20, implementar un sistema eficiente en encriptación, autenticación, prevención de vulnerabilidades, lo cual lo vuelve una tarea demasiado compleja.
14. Rendimiento.  
RTA/ 13, optimizar el rendimiento implica ajustar la arquitectura del sistema, de forma que soporte el uso de múltiples usuarios al mismo tiempo.
15. Mantenibilidad.  
RTA/ 13, diseñar un sistema al cual resulte fácil de hacer mantenimientos, requiere de buenas prácticas desde el momento del desarrollo, llega a tomar tiempo y puede resultar en una tarea compleja.
16. Disponibilidad.  
RTA/ 13, garantizar que se encuentre disponible en todo momentos, requiere de un diseño complejo, y de mantenerlo monitoreado constantemente, lo cual implica una tarea demasiado compleja.

### **Jacobo Alzate Corredor**

1. Integración con el sistema de gestión académica de la universidad para validar usuarios.

- RTA/ 20 Este proceso puede ser demorado ya que depende de que sistemas use la Universidad para almacenar a los usuarios, se puede usar una API
2. Verificación de identidad por correo o SMS.  
RTA/ 8 Hay que implementar los típicos códigos de verificación, lo cuál creería que no es tan difícil
  3. Los estudiantes podrán añadir contactos mediante búsqueda por nombre o matrícula.  
RTA/ 8 Para esto se necesitaría una buena base de datos, con esto el trabajo sería fácil de implementar
  4. Los profesores podrán visualizar automáticamente a estudiantes actuales y anteriores, además de agregar profesores manualmente.  
RTA/ 13 Se requiere una base de datos más amplia, con el historial de cada grupo de alumnos de cada clase de cada profesor lo cuál es más complejo
  5. Interfaz sencilla para enviar y recibir mensajes.  
RTA/ 8 Aunque sea “sencilla” esta necesita que los mensajes se vean en tiempo real y esta implementación no es tan fácil de realizar
  6. Indicadores de "escribiendo", leído y entregado.  
RTA/ 8 De nuevo esto requiere de actualizaciones en tiempo real, y esto puede ser complejo
  7. Creación de grupos por estudiantes o profesores (según permisos).  
RTA/ 5 Esta tarea no debería ser compleja ya que la creación de conjuntos no es la veo tan difícil
  8. Gestión de grupos por el creador (añadir/eliminar miembros).  
RTA/ 5 Añadir o eliminar miembros tampoco debería ser una tarea muy compleja, requiere de permisos de usuarios para el gestor de cada grupos
  9. Almacenamiento de mensajes para consultas futuras, con opción de buscar por palabras clave.  
RTA/13 El problema está mayormente en optimizar la búsqueda de palabras claves, creo que el almacenamiento de los mensajes no sería mayor problema
  10. Alertas para mensajes nuevos con opción de silenciar chats o grupos específicos.  
RTA/ 8 Estas alertas requieren sincronización con notificaciones, depende de en donde se quiera implementar si es en un teléfono o en un computador puede ser más fácil o más difícil
  11. Filtros avanzados para localizar conversaciones o usuarios específicos.  
RTA/ 8 Depende de la cantidad de filtros que se quieran, por ejemplo si es para archivos hay que categorizarlos y se puede volver más complejo, pero no es una tarea que vea excesivamente difícil.
  12. Recordatorios automáticos de clases o actividades asociadas a un curso.  
RTA/ 13 Esto también requiere la sincronización con notificaciones, a parte de necesitar del historial del curso para programar las clases o actividades por fuera de este horario
  13. Seguridad.  
RTA/ 20 Requiere la autenticación de usuarios que habíamos dicho, a parte necesitamos protecciones adicionales para las bases de datos y cifrado de mensajes, etc.

14. Rendimiento.

RTA/ 8 Puede que después de implementar todas estas cosas el sistema siga funcionando bien pero hay que optimizar para tener el mejor rendimiento posible

15. Mantenibilidad.

RTA/ 13 Requiere de estar constantemente haciendo pruebas de todas las implementaciones que se hagan y eso puede resultar en una tarea compleja

16. Disponibilidad.

RTA/ 8 Las tareas se pueden extender debido a problemas inesperados

## REFERENCIAS

- IA abierta. (2024). *ChatGPTv2* (versión del 6 de noviembre de 2023) [Modelo de lenguaje grande]. <https://chat.openai.com/chat>