Integrantes:

- Juan Daniel Ramirez Mojica
- Daniel Santiago Cocinero Jimenez
- Natalia Carolina Bautista Sanchez
- Baruj Vladimir Ramirez Escalante

Tarea 03 - Análisis de Requerimientos

Problema

- El objetivo del cliente es crear una red que permita facilitar la comunicación entre su comunidad académica.
- Dentro de la app se esperan dos roles, estudiantes y profesores, cada rol tiene formas de comunicación distintas. El presupuesto se tomará como infinito. (no se harán métodos de estimación por costo y riesgo).

Roles

01. Estudiantes: Los estudiantes pueden buscar y agregar a otros estudiantes como contactos, enviando mensajes individuales o creando grupos entre ellos, sin restricción. También pueden interactuar con profesores, pero solo si existe una relación académica previa o están inscritos en un curso actual con ellos. En caso de no tener una relación académica el estudiante deberá enviar una solicitud al profesor para poder comunicarse con él.
02. Profesores: Profesores pueden contactar solo a estudiantes que estén o hayan estado inscritos en sus cursos. Tendrán un grupo de chat con los estudiantes asociados a las materias que estén dictando. Los profesores pueden agregar a otros profesores como contactos sin restricciones. Se notificarán a los estudiantes de las actividades académicas asociadas al curso. Encuestas a los estudiantes.

Contexto

Se presenta un nuevo cliente, el cual busca desarrollar una aplicación para mejorar la comunicación entre su comunidad académica. La institución desea que esta red conecte a estudiantes y profesores, permitiendo interacciones efectivas dentro de un marco que respete las relaciones académicas establecidas.

El cliente no tiene restricciones presupuestarias, lo que permite enfocarse en la implementación de funcionalidades robustas y una experiencia de usuario óptima. La aplicación deberá contemplar dos roles principales: estudiantes y profesores, cada uno con diferentes permisos y formas de interacción.

Los estudiantes podrán comunicarse libremente entre ellos y con los profesores, siempre que exista una relación académica previa o se formalice mediante una solicitud. Por su parte, los profesores tendrán acceso a estudiantes asociados a sus cursos y podrán gestionar grupos de comunicación con ellos, además de poder interactuar sin restricciones con otros profesores.

El sistema deberá integrarse con la plataforma académica de la universidad para validar usuarios, mantener un registro confiable de las relaciones académicas y automatizar notificaciones de actividades académicas. La implementación de medidas de seguridad, como la encriptación de mensajes, será fundamental para proteger la privacidad de la información, mientras que el rendimiento y la disponibilidad garantizarán una experiencia fluida para los usuarios.

Requerimientos

01. Requerimientos funcionales

a. Registro e inicio de sesión:

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

b. Lista de contactos dinámica:

i. Estudiantes: Añadir contactos mediante búsqueda por nombre o matrícula.

c. Chat individual:

i. Interfaz sencilla para enviar y recibir mensajes. Indicadores de "escribiendo", leído y entregado.

d. Chat grupal:

i. Creación de grupos por estudiantes o profesores (según permisos).

ii. Gestión de grupos por el creador (añadir/eliminar miembros).

e. Historial de mensajes:

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

f. Notificaciones en tiempo real:

i. Alertas por notificaciones.

g. Búsqueda de mensajes y contactos:

i. Filtros avanzados para localizar conversaciones o usuarios específicos.

h. Integración con calendario académico:

i. Recordatorios automáticos de clases o actividades asociadas a un curso.

02. Requerimientos NO funcionales

a. Seguridad:

i. Uso de encriptación para mensajes (e.g., cifrado end-to-end).

b. Rendimiento:

i. Respuesta rápida en la carga de mensajes y sincronización en tiempo real.

c. Mantenibilidad (Dado a que no es medible se va a ver cómo testing):

i. Código modular y documentado para facilitar actualizaciones.

d. Disponibilidad:

i. Sistema operativo 24/7 con menos del 1% de tiempo de inactividad.

Método MoSCoW - Respuestas propias

1. **Must Have:**
   - **a.** Registro e inicio de sesión.
   - **b.** Lista de contactos dinámica.
   - **c.** Chat grupal.
   - **d.** Historial de mensajes.
   - **e.** Disponibilidad.
2. **Should Have:**
   - **a.** Chat individual.
   - **b.** Mantenibilidad.
   - **c.** Seguridad.
   - **d.** Rendimiento.
3. **Could have:**
   - **a.** Notificaciones en tiempo real.
   - **b.** Búsqueda de mensajes y contactos.

4. **Won't have (futuras versiones):**
    a. Integración con calendario académico.

Método MoSCoW - Respuestas de las diapositivas

5. **Must Have:**
    a. Registro e inicio de sesión.
    b. Búsqueda de mensajes y contactos.
    c. Chat individual.
    d. Historial de mensajes.
    e. Seguridad.
6. **Should Have:**
    a. Chat grupal.
    b. Rendimiento.
    c. Disponibilidad
    d. Lista de contactos dinámica.
7. **Could have:**
    a. Mantenibilidad.
8. **Won't have (futuras versiones):**
    a. Notificaciones en tiempo real.
    b. Integración con calendario académico.

Estimación:

Para el proceso de estimación se utilizará la metodología "Planning Poker" haciendo uso de la herramienta https://planningpokeronline.com/.

Se tendrá en cuenta que el proyecto se desarrollará utilizando Python sin frameworks, lo que implica un enfoque manual en la creación de funcionalidades. Además, se empleará una base de datos no relacional, MongoDB, para gestionar la información de manera eficiente y flexible (De acuerdo a lo mencionado en clase).

Individual

01. Registro e inicio de sesión.

- **a.** Natalia Bautista: **5 días.**
    - i. Un login no es una de las partes que requieren tanto trabajo, es las primeras cosas que se hacen y no es tan complejo.
- **b.** Santiago Cocinero: **5 días.**
    - i. Debido a que no creo que tome más de 5 días hacer un login, no es algo muy complejo y hay bastante información sobre eso en internet.
- **c.** Vladimir Ramírez: **13 días.**
    - i. Esta es la base del proyecto, el cual requiere la creación de múltiples subsistemas para funcionar de manera correcta, incluyendo el buscar e implementar una base de datos, verificación de identidad, conexión con sms, creación del usuario dentro de la aplicación, etc.
- **d.** Juan Daniel Ramírez: **5 días.**
    - i. Desde mi experiencia, no veo tan complicado implementar este inicio de sesión, especialmente si se utilizan herramientas como Firebase; sin embargo, hace algún tiempo que no uso estas herramientas, por lo que podría tomarnos un poco más de tiempo.

02. Búsqueda de mensajes y contactos.

- **a.** Natalia Bautista: **8 días.**
    - i. Considero que una búsqueda se no es tan complejo como para tardar más de una semana haciéndolo, se deben implementar consultas a la base de datos, implementar una barra de búsqueda
- **b.** Santiago Cocinero: **8 días.**
    - i. Creo que es un poco más complejo que el login debido a que son varias consultas sobre una DB mongo algo que no creo que sea sencillo de implementar.
- **c.** Vladimir Ramírez: **5 días.**
    - i. Los mensajes se buscan dentro del mismo perfil del usuario, por lo que no debería ser muy extenso, Con la base de datos ya creada anteriormente, solo toca ver como realizar las búsquedas.
- **d.** Juan Daniel Ramírez: **13 días.**

i. Desde mi punto de vista, al implementar el buscar de cursos debemos asegurarnos de establecer de manera correcta la base de datos y yo no he trabajado con vasos de datos no relacionales. Por lo que necesitaría un tiempo para estudiarlo y lograr implementarlo de manera correcta, ya que esta es fundamental dado que es la implementación de la base de datos.

03. Chat individual.

a. Natalia Bautista: **13 días.**

i. Se necesita configurar los servidores y bases de datos, se debe implementar autenticación de usuarios, establecer un sistema de envío en tiempo real, crear la interfaz, implementar las notificaciones.

b. Santiago Cocinero: **5 días.**

i. No se mucho de websockets así que siento que la complejidad de la implementación viene a la investigación de eso.

c. Vladimir Ramírez: **8 días.**

i. Entre la implementación de la interfaz y la comunicación en tiempo real de los dos usuarios me parece un tiempo prudente.

d. Juan Daniel Ramírez: **5 días.**

i. Considero que 5 días son suficientes para implementarlo, ya que la base de datos ya está configurada y permite gestionar los mensajes. Sin embargo, estoy dispuesto a extender el tiempo a 8 días, ya que esta funcionalidad será la base para el chat grupal. Además, creo que hay muchas cosas que se pueden encontrar en internet las cuales pueden facilitar el trabajo.

04. Historial de mensajes.

a. Natalia Bautista: **5 días.**

i. Se debe diseñar como se mostrará el historial de mensajes y definir cómo se mostrarán, si por fecha, usuario o conversación por ejemplo

b. Santiago Cocinero: **5 días.**

i. Es conectar el sistema de mensajes con la base de datos, creo que la mayor parte del tiempo se va diseñando cómo se va a almacenar eso en la base de datos.

c. Vladimir Ramírez: **5 días.**

i. Sobre todo es la implementación de el guardado y recuperación de los mensajes, y tal vez complementar la interfaz anterior de chats individuales.

d. Juan Daniel Ramírez: **5 días.**

i. Considero que ya con el chat privado como con la base de datos, el historial no debe ser complicado; sin embargo, toca asegurarse de que el historial quede montado de la manera correcta.

05. Seguridad.

a. Natalia Bautista: **8 días.**

i. Se deben analizar los riesgos y posibles vulnerabilidades, además de asegurar que solo los usuarios puedan leer los mensajes sin que terceros puedan interceptarlos.

b. Santiago Cocinero: **8 días.**

i. Por mi experiencia con la ciberseguridad se desestima mucho el trabajo de protección de la información y siendo esta una aplicación de chat siento que la forma de proteger la información debe ser robusta cosa que lleva un poco más de tiempo.

c. Vladimir Ramírez: **8 días.**

i. Desconozco la forma exacta de implementar la seguridad, por lo que hay que dedicarle un tiempo a investigar para implementarlo de la forma correcta.

d. Juan Daniel Ramírez: **13 días.**

i. Considero que 13 días es un poco corto y me gustaría que fueran 15; dado que no conozco de seguridad ni he tomado clases relacionadas a eso,lo único que conozco es RSA.

06. Chat grupal.

a. Natalia Bautista: **3 días.**

i. Se debe implementar la lógica para crear grupos, agregar o eliminar usuarios.

b. Santiago Cocinero: **2 días.**

i. Creo que toma poco tiempo debido a que es modificar un poco el sistema de chat individual.

c. Vladimir Ramírez: **5 días.**

i. Basándose en el chat individual y lo implementado en los usuarios, no debería demorar mucho tiempo.

d. Juan Daniel Ramírez: **2 días.**

i. Dado que ya se tendría tanto el chat grupal como el historial y la base de datos; no veo muy complicado esta implementación.

07. Rendimiento.

a. Natalia Bautista: **3 días.**

i. Se deben revisar las consultas a la base de datos para que sean eficientes, reducir el tamaño de los archivos, actualizar solo las partes cambiantes.

b. Santiago Cocinero: **3 días.**

i. Creo que los días dedicados solo al rendimiento son pocos porque en general el enfoque del rendimiento se debe llevar desde el inicio, con el uso de base de datos no relacional y frameworks ligeros para optimizar.

c. Vladimir Ramírez: **5 días.**

i. Si se implementan tecnologías que soporten el uso que se le va a dar, lo único demorado es el realizar pruebas para verificar el rendimiento.

d. Juan Daniel Ramírez: **3 días.**

i. La verdad no conozco mucho del tema; pero en internet deben haber varios consejos o tecnologías que se puedan usar las cuales faciliten este requerimiento.

08. Disponibilidad.

a. Natalia Bautista: **5 días.**

i. Hay que asegurarse de que se maneje el tráfico entre varios servidores, implementar backups para evitar la pérdida de información, configurar los sistemas para que escalen según la demanda.

b. Santiago Cocinero: **2 días.**

i. Con la opción tan amplia que brinda la nube y con un desarrollo óptimo sin crashes seguidos en la aplicación, los días de solo implementación de disponibilidad no creo que sean demasiados.

c. Vladimir Ramírez: **5 días.**

i. Nuevamente, si se implementan buenas tecnologías, no debería ser demorado.

d. Juan Daniel Ramírez: **2 días.**

i. Dado que no hay límite de dinero a utilizar, se puede pagar un buen servicio y con una infraestructura adecuada.

09. Lista de contactos dinámica.

a. Natalia Bautista: **3 días.**

i. Desarrollar consultas que actualicen la lista automáticamente según los criterios definidos

b. Santiago Cocinero: **2 días.**

i. Creo que es solo implementar algunas buenas consultas en la base de datos.

c. Vladimir Ramírez: **5 días.**

i. Al tener la base de datos, esto solo implica hacer las búsquedas en esta y restringir dependiendo del rol.

d. Juan Daniel Ramírez: **2 días.**

i. Por lo que entiendo, sería solo aplicar unos filtros a las bases de datos que ya están montadas, no debería tomar mucho tiempo.

10. Mantenibilidad.

a. Natalia Bautista: **5 días.**

i. Estructurar el código de manera modular, asegurarse de que el código sea legible con nombres y funciones entendibles para facilitar su mantenimiento.

b. Santiago Cocinero: **8 días.**

i. Creo que sería bueno dedicarle una semana al testeo de la aplicación.

c. Vladimir Ramírez: **13 días.**

i. Este proceso se debería realizar durante todo el desarrollo, un día y algo más dependiendo de la función debería bastar.

d. Juan Daniel Ramírez: **2 días.**

i. Dado a que se va a ver cómo testing y la idea sería asegurarse que a medidas que se implementan los requerimientos nos aseguremos que queden bien implementados, el testing no debería demorar mucho.

11. Notificaciones en tiempo real.

a. Natalia Bautista: **5 días.**

i. Desarrollar la lógica para manejar el envío de notificaciones, configurar el sistema de colas para manejar las notificaciones y evitar sobrecargas en el servidor, se debe permitir además que el usuario personalice las mismas.

b. Santiago Cocinero: **1 día.**

i. No creo que tome demasiado mandar una notificación al usuario.

c. Vladimir Ramírez: **5 días.**

i. El sistema de silenciar chats específicos puede demorarse un poco más para integrarse con otros sistemas anteriores.

d. Juan Daniel Ramírez: **2 días.**

i. Similar a los requerimientos anteriores, ya con el resto de requerimientos implementados, las notificaciones deben ser relativamente sencillas de implementar.

12. Integración con calendario académico.

a. Natalia Bautista: **8 días.**

i. Se debe permitir sincronizar los eventos claves con una interfaz para visualizarlos.

b. Santiago Cocinero: **13 días.**

i. Creo que la implementación conlleva a una creación de un rol diferente con una vista y permisos que conllevan un tiempo considerable.

c. Vladimir Ramírez: 5 **días.**

i. Basándose en el sistema de notificaciones anterior, solo toca ir agendando las notificaciones según se crean actividades en un curso.

d. Juan Daniel Ramírez: **5 días.**

i. Lo único que veo complicado es la integración con el sistema de la universidad; no solo nunca lo he hecho sino que desconozco si nos

deben dar una vista especializada a nosotros; además se deberá obtener información como las clases y sus horarios correspondientes.

Consenso Grupal

Tras haber analizado los resultados obtenidos en el poker planning y haber comunicado las razones por las cuales habíamos dado x votación, este fue el consenso al que se llegó.

01. Registro e inicio de sesión: **7 días.**

02. Búsqueda de mensajes y contactos: **8.5 días.**

03. Chat individual: **7.5 días.**

04. Historial de mensajes: **5 días.**

05. Disponibilidad: **9 días.**

06. Chat grupal: **3 días.**

07. Rendimiento: **3.5 días.**

08. Disponibilidad: **3.5 días.**

09. Lista de contactos dinámica: **3 días.**

10. Mantenibilidad (testing): **7 días.**

11. Notificaciones en tiempo real: **3 días.**

12. Integración con calendario académico: **7 días.**