

## **Actividad #4 Ingesoft**

**Presentado por:**

**Laura Sofia Vargas Rodriguez - [lavargasro@unal.edu.co](mailto:lavargasro@unal.edu.co)**

**Jeronimo Bermudez Hernandez - [jebermudez@unal.edu.co](mailto:jebermudez@unal.edu.co)**

**Nicolas Quezada Mora - [nquezada@unal.edu.co](mailto:nquezada@unal.edu.co)**

**Sharick Yelixa Torres Monroy - [shtorres@unal.edu.co](mailto:shtorres@unal.edu.co)**

**Profesor:**

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

**Jueves 28 de Noviembre**



**Universidad Nacional de Colombia**

**Facultad de Ingeniería**

**Departamento de Ingeniería de Sistemas e Industrial**

**2024**

## **Requerimientos funcionales mediante MOSCOW**

- 1). Registro e inicio de sesión - Must
- 2). Listado de contactos dinámica -Could
- 3). Chat individual - Must
- 4). Chat grupal - Must
- 5). Historial de mensajes - Could
- 6). Notificaciones en tiempo real - Should
- 7). Búsqueda de mensajes y contactos - Must
- 8). Integración de calendario académico - Won't

## **Requerimientos no funcionales**

- 2.1). Seguridad - Must
- 2.2). Rendimiento - Should
- 2.3). Mantenibilidad - Should
- 2.4). Disponibilidad - Must

## **Comparativa con el ejercicio:**

Mo

1, 7, 3, 5, 2.1

S

4, 2.2, 2.4, 2

Co

2.3

W

6, 8

## **Estimaciones y fibonacci (Nicolas)**

### **1). Registro e inicio de sesión: 5**

- Crear las pantallas para que los usuarios puedan registrarse, iniciar sesión y recuperar sus contraseñas. Es algo básico pero lleva su tiempo.

### **2). Listado de contactos dinámica: 5**

- Mostrar los contactos y que se actualicen automáticamente según lo que haga el usuario. No es complicado, pero requiere cuidado.

### **3). Chat individual: 8**

- Un chat para que dos personas hablen en tiempo real. Requiere bastante trabajo porque debe ser rápido y funcional.

### **4). Chat grupal: 13**

- Similar al chat individual, pero con más lógica para manejar varios usuarios al mismo tiempo. Esto lo hace más complejo.

### **5). Historial de mensajes: 8**

- Guardar y mostrar los mensajes anteriores en el chat, con opción de ver los más antiguos. Suena simple, pero lleva tiempo organizarlo.

### **6). Notificaciones en tiempo real: 8**

- Que el sistema avise al usuario cuando haya nuevos mensajes o cambios importantes. Esto requiere sincronización constante.

### **7). Búsqueda de mensajes y contactos: 5**

- Permitir que los usuarios encuentren rápidamente mensajes o contactos. No es difícil, pero debe ser eficiente.

### **8). Integración de calendario académico: 13**

- Incluir un calendario que permita organizar eventos o sincronizar con uno externo, como Google Calendar. Es útil pero bastante trabajo.
-

## **Requerimientos no funcionales**

### **2.1). Seguridad: 8**

- Asegurar que todo esté protegido como contraseñas, datos personales y chats. Es importante y requiere cuidado.

### **2.2). Rendimiento: 8**

- Hacer que el sistema sea rápido y no se quede corto, incluso con muchos usuarios. Hay que optimizar varios detalles.

### **2.3). Mantenibilidad: 5**

- Diseñar todo de manera organizada para que sea fácil de corregir o mejorar en el futuro. Requiere algo de planificación.

### **2.4). Disponibilidad: 8**

- Asegurarse de que la app funcione todo el tiempo sin caídas. Esto implica prevenir errores y mantener el sistema estable.

## **Estimaciones y fibonacci (Sofia)**

### **Requerimientos funcionales**

#### **1). Registro e inicio de sesión: 5**

- Ya que generalmente se necesita una base de datos puede llegar a ser un poco demorado, es mejor estimar un tiempo extra por si se presenta algún imprevisto

#### **2). Listado de contactos dinámica: 5**

- No es complicado pero puede llegar a ser demorado de crear dependiendo de cómo se maneje

#### **3). Chat individual: 13**

- Considero que crear un chat funcional en tiempo real que sea eficiente y se acomode a las necesidades de los usuarios requiere tiempo

4). Chat grupal: 13

- Al manejar más usuarios al mismo tiempo puede llegar a complicarse un poco más que el individual, sin embargo teniendo el individual ya hecho se tiene una base sobre la cual trabajar y se puede ahorrar tiempo

5). Historial de mensajes:8

- Debido a que se debe configurar de manera específica la base de datos para traer los mensajes más antiguos y tenerlos en cuenta puede llegar a ser demorado.

6). Notificaciones en tiempo real:8

- Integrar y probar la funcionalidad de notificaciones en las plataformas objetivo requiere un tiempo significativo, sobretodo para garantizar la experiencia del usuario, sin embargo usar servicios como Firebase puede agilizar el proceso

7). Búsqueda de mensajes y contactos:5

- Implementar maneras de buscar mensajes y contactos puede tardar algo de tiempo, sin embargo se reduce de manera significativa teniendo en cuenta que ya tenemos los datos necesarios almacenados

8). Integración de calendario académico:8

- Es algo demorado debido a que se deben vincular los eventos a los usuarios o grupos específicos.

**Requerimientos no funcionales**

2.1). Seguridad:8

- Debido a la importancia de la protección de datos me parece importante dedicarle tiempo

2.2). Rendimiento:13

- Se pueden usar herramientas para analizar y optimizar, además de reducir la carga en la base de datos, sin embargo es un proceso que necesita un tiempo considerable

2.3). Mantenibilidad:8

- Un código limpio y documentación detallada podrían mejorar el proceso, sin embargo para cubrir todos los aspectos se requiere tiempo

2.4). Disponibilidad: 5

- Utilizar frameworks y librerías de código abierto bien documentadas y mantenidas puede acelerar el desarrollo, sin embargo la mayoría de funciones necesarias para asegurar el funcionamiento de la aplicación requieren de una revisión considerable

## **Estimaciones y fibonacci (Jerónimo)**

### **Requerimientos funcionales**

#### **1). Registro e inicio de sesión: 3**

Con el uso de tecnologías como firebase no debería ser muy complicado. Sin estas tecnologías sería un poco más complejo debido a la naturaleza de la base de datos con la que trabajamos.

#### **2). Listado de contactos dinámica: 5**

No debería ser muy complicado, la implementación de la lista de contactos debe seguir un CRUD simple.

#### **3). Chat individual: 21**

No sé mucho pero el profe dijo que era difícil entonces le creo.

#### **4). Chat grupal: 5**

Teniendo el chat individual implementar el grupal no debería ser tan complicado.

#### **5). Historial de mensajes: 8**

Requiere del diseño de el guardado de los datos en la base de datos, un registro que lo envíe y una función de consultar, aparte de la interfaz.

#### **6). Notificaciones en tiempo real: 5**

Es necesario conectar al servidor y escuchar los eventos para mandar la notificación al cliente

#### **7). Búsqueda de mensajes y contactos: 3**

Como ya existe un registro de los mensajes y de los contactos, es más simple implementar este tipo de búsquedas.

#### **8). Integración de calendario académico: 5**

Es necesario guardar y editar los datos del calendario en la base de datos, sobre eso hay que implementar una interfaz

### **Requerimientos no funcionales**

#### **2.1). Seguridad: 3**

No sabía pero resulta que no es muy complicado hacer un cifrado end to end.

#### **2.2). Rendimiento: 21**

Voy a implementar todo con ciclos.

#### **2.3). Mantenibilidad: 13**

Verificar el código y documentarlo correctamente.

#### **2.4). Disponibilidad: 3**

Asegurar la disponibilidad del servidor.

## **Estimaciones y fibonacci (Sharick)**

### **Requerimientos funcionales**

1). Registro e inicio de sesión: 5

El desarrollo incluye validación de datos, integración con base de datos, el diseño de un front end adecuado, en este caso, depende de las herramientas y recursos que se utilicen.

2). Listado de contactos dinámica: 5

Al ser dinámico, requiere de una implementación que permita el filtrado y búsquedas en tiempo real.

3). Chat individual: 8

Sus funcionalidades son un tanto más complicadas de desarrollar, debe permitir el envío y recepción de mensajes en tiempo real, previniendo errores y asegurando una interfaz adecuada.

4). Chat grupal: 13

Mayor complejidad en cuanto a que se requiere el relacionamiento de varios usuarios de manera simultánea, sincronizando el envío y recepción de mensajes, permisos y, con esto, la administración de grupos.

5). Historial de mensajes: 3

Se emplean consultas para recuperación de datos y su respectivo almacenamiento en la base de datos.

6). Notificaciones en tiempo real: 8

Se requiere el uso de algunas herramientas o tecnologías a parte que garanticen la sincronización de los eventos con el usuario.

7). Búsqueda de mensajes y contactos: 5

Requiere de una implementación que permita el filtrado y búsquedas rápidas, manejando grandes volúmenes de datos.

8). Integración de calendario académico: 8

Pese a que se puede vincular con algunas tecnologías ya creadas, se debe considerar la sincronización de información o eventos y el diseño de una interfaz personalizada.

**Requerimientos no funcionales**

2.1). Seguridad: 8

Se deben tener en cuenta muchos aspectos, tales como: autenticación, cifrado de datos, manejo apropiado de contraseñas, etc, esto requiere de bastante cuidado, sin embargo existen implementaciones que facilitan este proceso.

2.2). Rendimiento: 13

Se tienen en cuenta la optimización de procesos, consultas, uso eficiente de recursos y almacenamiento de datos, etc

2.3). Mantenibilidad: 5

Se debe procurar llevar un desarrollo bien documentado y cuidadoso a lo largo de todo el proceso para evitar que se alargue este periodo.

2.4). Disponibilidad: 8

Se debe realizar un análisis que permita prevenir errores varios, evitando caídas y garantizando la estabilidad del sistema y una ejecución adecuada de cada funcionalidad.

## **Estimaciones y fibonacci (Consenso)**

### **Requerimientos funcionales**

#### **1). Registro e inicio de sesión: 5**

- Pese a que ciertas tecnologías agilizarán el desarrollo, seguimos considerando que el hecho de tener que integrar una base de datos sumado a la autenticación que requiere cada usuario suele ser un poco más demorado

#### **2). Listado de contactos dinámica: 5**

- Todos tenemos la misma idea del tiempo que llevaría realizar este requerimiento, no debería ser muy demorado sin embargo se mantiene un tiempo para imprevistos

#### **3). Chat individual: 13**

- Pese a que en un inicio no suene tan difícil, consideramos que sus funcionalidades son un tanto más complicadas de desarrollar, debe permitir el envío y recepción de mensajes en tiempo real, además de adaptarse a todas las necesidades de los usuarios, lo cual requiere tiempo

#### **4). Chat grupal: 13**

- Mayor complejidad, ya que requiere implementar diferentes usuarios interactuando al mismo tiempo, lo cual puede llegar a complicarse un poco más que el chat individual.

#### **5). Historial de mensajes: 8**

- Pese a que suena simple requiere del diseño de el guardado de los datos en la base de datos, un registro que lo envíe, además de consultas para recuperación de datos.

#### **6). Notificaciones en tiempo real:8**

- Se requiere el uso de algunas herramientas o tecnologías a parte que garanticen la sincronización de los eventos con el usuario, para garantizar la experiencia del mismo, sin embargo usar servicios como Firebase puede agilizar el proceso

#### **7). Búsqueda de mensajes y contactos:5**



- Requiere de una implementación que permita el filtrado y búsquedas rápidas, manejando grandes volúmenes de datos, sin embargo se reduce de manera significativa el tiempo que se le debe dedicar teniendo en cuenta que ya tenemos los datos necesarios almacenados

#### 8). Integración de calendario académico:8

- Es necesario guardar y editar los datos del calendario en la base de datos lo cual es demorado, pero se puede vincular con algunas tecnologías ya creadas

### **Requerimientos no funcionales**

#### 2.1). Seguridad:8

- Se deben tener en cuenta muchos aspectos, tales como: autenticación, cifrado de datos, manejo apropiado de contraseñas,etc, lo cual requiere tiempo, sin embargo hay algunos cifrados que facilitan el proceso

#### 2.2). Rendimiento:13

- Se pueden usar herramientas para analizar y optimizar, además de reducir la carga en la base de datos, sin embargo es un proceso que necesita un tiempo considerable

#### 2.3). Mantenibilidad:5

- Diseñar todo de manera organizada para que sea fácil de corregir, y un código limpio y documentación detallada podrían mejorar el proceso, sin embargo requiere tiempo

#### 2.4). Disponibilidad:8

- Asegurar que la aplicación funcione sin interrupciones es crucial, esto implica prevenir caídas mediante el uso de frameworks, redundancia de servidores, y un monitoreo constante, aunque algunas herramientas ayudan a reducir la complejidad, se debe dedicar bastante tiempo a pruebas y ajustes para garantizar la estabilidad y confiabilidad del sistema.