**Integrantes del equipo:**

Alan Cichelo.

Juan Nicolás Nieva

Miguel Angel de Robles (Encargado de la comunicación)

**Usuarios de SLACK:**

Alan Cichelo

Juan Nicolás

Miguel Angel de Robles

**Usuario de TRELLO:**

Juan Nicolás Nieva (juannicolasnieva)

Alan Cichelo(alancichelo)

Miguel (miguel90790895)

**Link de Trello:** [**https://trello.com/b/MlcYzkPO/red-social**](https://trello.com/b/MlcYzkPO/red-social)

**Arquitectura:**

**Requerimientos no funcionales**

**1-**

-La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.

-El sistema debe controlar el acceso a la “Red Social” validando los datos de los usuarios

-El sistema debe poseer interfaces gráficas bien formadas pero debe ser simple y amigable para el usuario

2-

-El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.

-El sistema debe estar disponible las 24 hs.

-El sistema debe tener respaldo. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.

-El sistema debe contar con un módulo de ayuda en línea.

-El sistema debe estar documentado tanto técnicamente como funcionalmente

**3-**

-El sistema debe ser capaz de procesar N procesos por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al Software Testing de servicios web.

-Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos. El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

-Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

-El sistema debe estar preparado para una futura escalabilidad

-El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

**4-**

-Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.

-El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.

-Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.

-Todas las comunicaciones internas de chat deben ser encriptadas

**5-**

-El sistema debe contar con manuales de usuario estructurados adecuadamente.

-El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 6 horas.

-La tasa de errores cometidos por el usuario deberá ser menor del 3-5% de las transacciones totales ejecutadas en el sistema.

**Requerimientos funcionales  
1-**-El sistema debe permitir agregar un contacto como “Amigo”

-Invitar “Amigos” a un grupo

-Crear un grupo de estudio-Mostrar grupo de estudio-Editar un grupo de estudio

~~-~~Crear un perfil de usuario-Mostrar perfil de usuario-Editar perfil de usuario-Borrar perfil de usuario.

-El sistema deberá permitir que un usuario se identifique

**2-**

-El sistema deberá permitir a un usuario que inicia sesión por primera vez, cargar información sobre materias aprobadas

-El sistema deberá mostrar grupos de interés relacionados las materias del usuario

-El sistema deberá mostrar grupos de estudio creados por el usuario

-Eliminar un grupo de estudio

-El sistema debe permitir enviar mensajes entre usuarios

**3-**

-Editar perfil de usuario-Mostrar perfil de usuario-Eliminar perfil de usuario-Añadir foto de perfil de usuario

-El sistema deberá permitir a un usuario postularse como tutor

-Perfil tutor aceptar solicitud-Perfil tutor eliminar solicitud

-El perfil tutor deberá poder administrar las solicitudes de x materia por parte de los usuarios

-Debe permitir que los usuarios administradores(tutor o creador del grupo) o quienes tengan asignados los permisos correspondientes, puedan crear nuevos niveles dentro de la estructura de clasificación.

-El sistema deberá permitir que un usuario le envié una solicitud (Mensaje) a un tutor, con el objetivo de ser aceptado por este último.

-Eliminar mensajes entre usuarios

-Permitir consultar la información de cada contacto

**Servicios –**

**HMTL5**

El lenguaje más importante y conocido para la creación de páginas web, dándoles una estructura y un contenido. Simple de usar para desarrollar la pagina web y, mas dado, que tenemos experiencia consigo.

**CSS3**

Al igual que HTML no es considerado un lenguaje de programación sino un lenguaje de estilo, versión mejorada de css y simple de usar

**JavaScript**

JavaScript es un lenguaje de programación orientado a objetos, la mejor opción para darle dinamismo a la pagina.

**React**

Liberia de Js para desarrollar de una manera mas ordenada y con menos código de js , y centrada a la manipulación del Dom, la mejor opción para darle dinamismo a la pagina.

**MongoDB**

Base de datos NoSQL, almacena los datos en una estructura

llave-valor. Lo utilizaremos por su mayor velocidad comparando con las relacionales para las Requests.

**Express**

Módulo de NodeJS de alto nivel de desarrollo web que ofrece un conjunto de características para las aplicaciones web. Muy simple para crear la API

**Sequelize**

Framework ORM que da soporte a Node.js para bases de datos NoSQL es muy completo, que utilizamos en un proyecto anterior y tenemos experiencia.

**SoapUI**

Herramienta de testing muy completa

**Postman**

Herramienta de testing para API-REST la cual tenemos experiencia de uso.

**Figma**

Herramienta mas usada en la actualidad y mas completo de las herramientas de UX gratuitas

**Dudas**

Cuando el usuario inicie sesión a Red Social, el historial académico lo tomara desde una base de datos de la UNAHUR; ¿O cómo se validará que la información que cargue sea la correcta?  
¿Como se validara cuando un usuario quiera ser “Tutor”, ¿que realmente haya aprobado dichas materias?; ¿Se tomará en cuenta un determinado promedio como filtro? ¿Qué server vamos a alquilar?

En caso de contratar un servidor, el precio no baja de los $1000. Podríamos tener un dedicado y probar que todo funcione bien y en caso de que la facultad quiera ponerlo en producción que asuma los gastos.

¿Cuántos usuarios estimas que entren a dicha Red?Máximo de 10000 a 15000 mil usuarios, pero empezaremos con 100.¿Como vamos a repartir el trabajo? ¿Quién se va a encargar del front- end y el back-end? - Definir ¿Separan front-end y back-end? Si. ¿El back-end lo van a empaquetar todo junto?No entendí a que hace referencia. En caso que se separe front-end y back-end, como le hace los requisitos al servidor?Mediante una API.