

PRIMERA APROXIMACIÓN DEL PROYECTO DE LA ASIGNATURA TMDAD.

Nota a la que aspirar: Aprobado.

Requisitos del sistema:

- Implementa adecuadamente las historias 1, 2, 3 y 5:

- 1) Como usuario Quiero poder enviar mensajes a otros usuarios Para tener conversaciones puntuales.

- 2) Como usuaria Quiero poder compartir ficheros de mi equipo con otros usuarios Para complementar las conversaciones con información adicional.

- 3) Como usuario Quiero poder crear salas de chat permanentes Para tener un registro de las conversaciones pasadas.

- 5) Como superusuario Quiero poder enviar mensajes a todos los usuarios suscritos Para mostrar anuncios de interés y publicidad de nuevas características del sistema.

- Soporta correctamente a varios usuarios simultáneos.

- Se despliega y ejecuta en al menos tres máquinas virtuales de las cuales al menos una debe estar en un proveedor cloud,

La documentación:

Justifica y documenta adecuadamente las principales decisiones arquitecturales tomadas, con especial énfasis en las que tienen que ver con la distribución del sistema, con al menos dos vistas arquitecturales: una de componentes y conectores, y otra de distribución en estilo de despliegue.

Stack tecnológico empleado:

Según la sugerencia (que no recomendación) de los profesores:

- Lenguaje: Java 8
- Construcción: Gradle
- Framework: Spring Boot
- Base de datos: MySQL
- Control de versiones: https://github.com/Danny2411/TMDAD_2019/

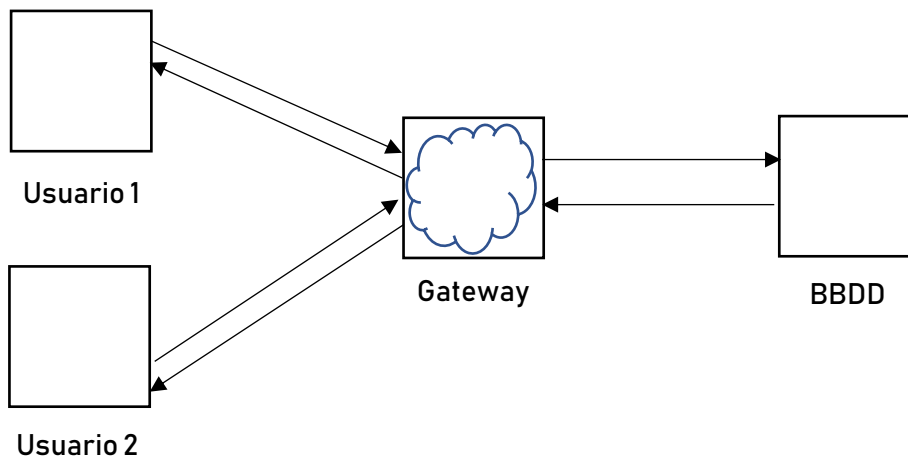
Descripción de las máquinas:

El esquema del que se parte es uno conformado por tres máquinas:

- Máquina cliente:
 - o Java
 - o Interfaz de usuario (¿Quizá Angular?)
 - o Máquina local.

- Máquina servidor (Gateway):
 - o Java
 - o Spring Boot
 - o Máquina en Cloud.
- BBDD (almacén de la información):
 - o MySQL
 - o Máquina local.

Conexión entre las máquinas:



La funcionalidad básica es como se ha mostrado: Los usuarios no se comunican directamente. Sus peticiones se envían a un servidor central desplegado en nube, que almacenará la información en una BBDD de la cual se recuperarán los mensajes y las conversaciones cuando proceda.

- Los clientes envían peticiones HTTP al servidor.
- El servidor envía mensajes HTTP a los clientes.
- El servidor envía queries a la BBDD mediante librerías predefinidas.
- La BBDD envía los resultados de las queries al servidor mediante librerías.

Datos intercambiados:

Mensajes: Hay dos tipos de mensajes, de texto y ficheros. Como sólo se dispone de una BBDD y es relacional, se van a definir todos los contenidos como secuencias de bytes. Bastará con indicar qué tipo de contenido tiene el mensaje.

Usuarios: Hay dos tipos de usuarios, en función de los privilegios. Un indicador en la base de datos será suficiente para saber qué acciones puede realizar.

Por tanto, el mensaje debería incluir:

- Usuario que envía
- Usuario que recibe

- Contenido del mensaje
- Tipo del mensaje

Base de datos:

Se quiere guardar información de los mensajes y de los usuarios.

Un mensaje podría necesitar los siguientes campos:

- Usuario Origen: Quién lo envía.
- Usuario Destino: Quién lo recibe.
 - o Comodín para mensajes de broadcast.
- Contenido del mensaje.
- Tipo del mensaje.
- Timestamp del mensaje.

Un usuario podría necesitar los siguientes campos:

- Nombre único de Usuario: Como identificador
- Tipo de usuario

Funcionamiento del sistema:

- Usuario se conecta a la aplicación como cliente.
- Usuario pide al servidor todos los mensajes destinados a él.
- Servidor realiza query de mensajes cuyo origen o destino sea usuario.
- BBDD busca la información.
- Servidor recibe la información.
- Servidor agrupa mensajes por sala (origen o destino idéntico).
- Cliente recibe mensajes agrupados.
- Interfaz muestra cada grupo en una sala.
- Repetir cada X tiempo por si aparecen mensajes simultáneos.

Posible interfaz de usuario:

IRENE	Buenos días ¡Hola! ¿Comemos hoy sushi? Vale, a la 1:30 en Plaza España.
LAURA	
DIEGO	
MAMÁ	
ADMINISTRADOR	