MensajerO

Trabajo práctico

Objetivos	● Afianzar los conocimientos adquiridos durante la cursada. ● Poner en práctica la coordinación de tareas dentro de un grupo de trabajo. ● Realizar un aplicativo de complejidad media con niveles aceptables de calidad y usabilidad.
Instancias de Entrega	**Checkpoint 1:** clase 8 (dd/mm/aaaa). **Checkpoint 2:** clase 13 (dd/mm/aaaa). **Entrega:** clase 16 (dd/mm/aaaa).
Criterios de Evaluación	● Empleo de buenas prácticas de programación ● Empleo de test unitarios ● Coordinación de trabajo grupal. ● Planificación y distribución de tareas para cumplir con los plazos de entrega pautados. ● Cumplimiento de todos los requerimientos técnicos y funcionales. ● Facilidad de instalación y ejecución del sistema final. ● Calidad de la documentación técnica y manuales entregados. ● Buena presentación del trabajo práctico y cumplimiento de las normas de entrega establecidas por la cátedra

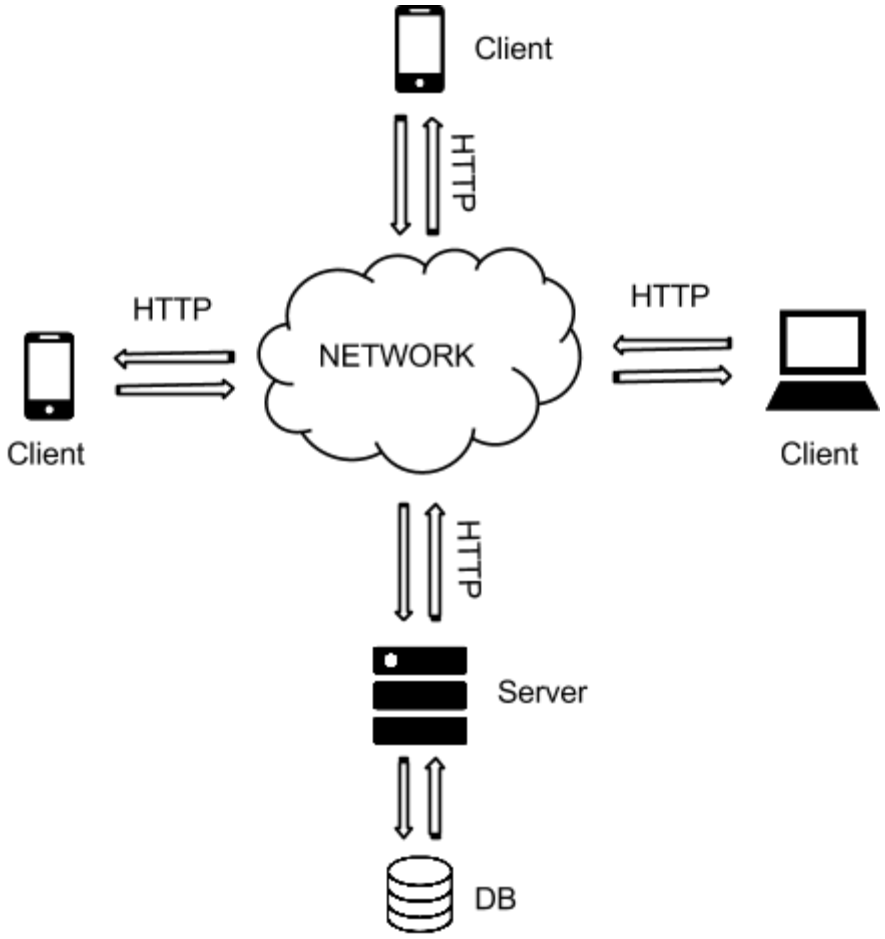
Índice

Introducción

Debido al auge de las aplicaciones para mensajerías y su integración con la web y dispositivos móviles, la empresa AppMaker© nos ha encargado el desarrollo de una aplicación de mensajería.
Esta empresa busca posicionarse en el mercado de aplicaciones de mensajería buscando conectar esta aplicación con sus otras aplicaciones que poseen gran cantidad de usuarios.

La aplicación constará de dos componentes:

- Un servidor, el cual será el responsable del procesamiento, almacenamiento y el delivery de los mensajes
- Un cliente, el cual será responsable de visualizar los mensajes de las conversaciones que realiza el usuario activo



Aplicaciones Requeridas

Servidor

Se trata de una aplicación linux por consola destinada a mantenerse en ejecución por períodos prolongados de tiempo. Esta aplicación debe brindar una interfaz para la comunicación de los diferentes clientes que se conecten a la misma. Para la interfaz de comunicación se deberá brindar una API REST [1], que definirá la forma de las solicitudes y respuestas de los diferentes servicios que brindará el servidor.

Servicio de autenticación

El servidor dispondrá de un servicio para la autenticación de los clientes. Este servicio consta de una solicitud de autenticación, que viene junto con las credenciales del usuario. La respuesta a la solicitud es un token (identificador) de la sesión del usuario.

Servicio de registración de usuarios

El servidor debe disponer de un servicio que permita la registración de nuevos usuarios en el sistema.

Servicio para delivery de conversaciones

El servidor brindará un servicio para el delivery de conversaciones. Cada conversación está compuesta por un header, que contiene el emisor, el receptor del mensaje y el token de autenticación, y un cuerpo, que es el contenido del mensaje en sí. El servidor debe tener un registro de las conexiones activas en el servidor para saber hacia donde dirigir el mensaje en curso.

Servicio de delivery de difusión

El servidor brindará un servicio (que puede ser el mismo que el delivery de conversaciones) que permitirá enviar un mensaje a todos los usuarios que se encuentren conectados en el momento de envío del mensaje.

Almacenamiento de conversaciones

El servidor debe persistir las conversaciones para que las mismas puedan existir en una ejecución posterior del servidor.

Servicio de consulta de usuarios disponibles

El servidor brindará un servicio para la consulta de los usuarios disponibles en el sistema. Se entiende por usuario disponible un usuario autenticado y que ha ingresado al sistema.

Checkin de usuarios

El servidor a través del checkin del usuario deberá registrar el acceso del mismo a un lugar específico ya cargado previamente en el sistema. En los datos de la registración de acceso al lugar se indicará la posición del usuario (latitud, longitud).

La administración de estos lugares la realizará el administrador del sistema, con lo que no será necesario desarrollar una administración sobre esa información. Dentro de los datos asociados a un lugar se encontrará su ubicación geográfica (latitud,longitud).

Servicio de administración de perfil de usuario

El servidor proveerá un servicio en donde el usuario podrá actualizar los datos asociados a su perfil, y los demás usuarios podrán consultar su estado. Estos datos consisten en:

- Nombre
- Foto de perfil
- Estado de conexión {conectado, desconectado}.
- Último checkin (optativo).

Log

El servidor debe constar con un sistema de log en donde se registren los eventos que se generen durante la ejecución del mismo. El sistema de log debe permitir configurar el nível de los eventos que desean registrar. Estos níveles son:

Nivel	Condiciones
Error	Condición de falla catastrófica, el sistema no puede funcionar. (criterio de las 2 a.m.) Condición que haga que la aplicación no pueda ejecutar una funcionalidad. Ejemplo: No es posible conectarse con la base de datos
Warn	Cualquier condición anómala que afecte el funcionamiento del sistema, pero no impida la funcionalidad básica Ejemplos: Uso de APIs deprecadas Mal uso de APIs
Info	Cualquier acción correspondiente a un caso de uso iniciada por el usuario o el sistema. Información que permita trazar el historial de las entidades. Ejemplos: Conexión a la base de datos exitosa Conexión de nuevo cliente
Debug	Información de contexto que permita resolver un problema técnico. Debe ser útil incluso sin el código fuente Ejemplo: Datos de login para la DB

Cliente

El sistema cliente posee una interfaz gráfica que permitirá visualizar los mensajes del usuario que se encuentra utilizando la aplicación.

Autentificación

Para que el cliente pueda enviar y recibir conversaciones el mismo debe autentificarse con el servidor. Para esto el cliente debe permitir al usuario ingresar usuario y contraseña. Se deberá diseñar una pantalla de ingreso a la aplicación.

Registración

En caso de que la persona que desee utilizar la aplicación no tenga un usuario registrado la aplicación debe permitir poder registrar un usuario nuevo.

Checkin

Desde el cliente el usuario podrá confirmar su presencia (checkin) en un lugar. Para obtener la posición del cliente se deberá poder utilizar la ubicación del dispositivo [2]. Este checking debe ser enviado al servidor para que surta efecto.

Interfaz gráfica

La visualización de las conversaciones de un usuario es la parte central del cliente. En esta interfaz se visualizarán las conversaciones activas del usuario con los demás usuarios del sistema.

Entre las características de la aplicación se encuentran:

- Visualización de conversaciones
- Lista de usuarios conectados
- Envío de conversaciones
- Visualización de estado de usuario
- Configuración de perfil

Visualización de conversaciones

La aplicación debe permitir visualizar las conversaciones activas que tenga el usuario dentro de la misma. Cuando el usuario desactive una conversación la misma no será listado de entre las conversaciones activas.

Lista de usuarios conectados

La aplicación debe ser capaz de listar los usuarios conectados al sistema de mensajería. Este listado además debe permitir consultar la información básica del usuario: nombre, hora de última actividad, estado (ej. online, do not disturb).

Envío de conversaciones

Dentro de la aplicación el usuario podrá continuar una conversación que tenía activa, enviando un nuevo mensaje al mismo destinatario que tenía la conversación, o generar una nueva conversación.

Visualización de estado de usuario

El usuario conectado podrá ver el estado de un usuario particular. Se entiende por estado de usuario al conjunto de información formado por:

- Estado de conexión (conectado/desconectado)
- Último checkin (con la hora en la que fue realizado).
- Foto de perfil del usuario

Configuración de perfil
Desde la aplicación se podrá configurar el perfil del usuario. Dentro de esta configuración se podrá seleccionar una foto para el perfil y se permitirá cambiar el estado de la conexión, esto permitirá que el usuario pueda aparecer como desconectado, y de esta forma los demás usuarios no puedan verlo conectado.

Log
El cliente debe constar con un sistema de log en donde registro los eventos que se generen durante la ejecución del mismo. El criterio de los niveles de log a utilizar es el mismo definido para la aplicación servidor.

Instalación de aplicación
La instalación de la aplicación cliente debe ser un proceso simple y que ayude a un usuario inexperto a poder realizar la instalación de la misma sin mayores inconvenientes.

Documentación
Es condición necesaria para la aprobación del trabajo práctico la entrega de una documentación formal de carácter profesional.

Pruebas
El desarrollo de la aplicación se deberá adaptar a los estándares de calidad utilizados por AppMaker©. Dentro de estos estándares se encuentran:

- Pruebas unitarias [3]
- Métricas (ej. code coverage) [4]
- Respetar estándar para estilo de codificación

Restricciones
- El servidor debe ser desarrollado en C/C++. El servidor debe soportar múltiples conexiones en simultáneo.
- Para la compilación de la aplicación servidor se deberá utilizar CMake [5]
- La aplicación debe desarrollarse en Java utilizando el SDK de Android [6]
- Para la documentación técnica del trabajo práctico se utilizará Sphinx [7]
- Para la documentación del código fuente se deberá utilizar documentación estilo Javadoc [8] (ej. Doxygen [9]).
- Para el almacenamiento de la información del servidor (ej. conversaciones, cuentas de usuarios) se deberá utilizar una base de dato de tipo clave-valor de tipo NoSQL[10]. Se recomienda utilizar rocksdb. [11]
- Se deberá utilizar un Git como sistema de versionamiento para el código fuente [12]

Referencias
- [0] https://www.android.com/
- [1] http://en.wikipedia.org/wiki/Representational_state_transfer

- [2] http://developer.android.com/reference/android/location/LocationManager.html
- [3] http://en.wikipedia.org/wiki/Unit_testing
- [4] http://en.wikipedia.org/wiki/Code_coverage
- [5] http://www.cmake.org/
- [6] http://developer.android.com/sdk/index.html
- [7] http://sphinx-doc.org/
- [8] http://es.wikipedia.org/wiki/Javadoc
- [9] http://www.stack.nl/~dimitri/doxygen/
- [10] http://en.wikipedia.org/wiki/NoSQL
- [11] http://rocksdb.org/
- [12] http://es.wikipedia.org/wiki/Git