



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR DE  
INGENIEROS DE  
TELECOMUNICACIÓN

# Proyecto Final Aplicaciones Telemáticas

Desarrollo de una aplicación de mensajería entre  
estudiantes y docentes

**Autores:** Gorka Aguirreurreta Vidal, Jorge Marco Arráez

27 de junio de 2022

# Índice general

1. Introducción	3
2. Funcionamiento de la aplicación	4
3. Explicación algorítmica	6

# 1. Introducción

Esta aplicación de mensajería está basada, en cuanto al backend, en la API y el SDK de CometChat y en lo que al frontend respecta, en un UI Kit donde se proporcionan todos los layouts necesarios para la correcta visualización de esta aplicación.

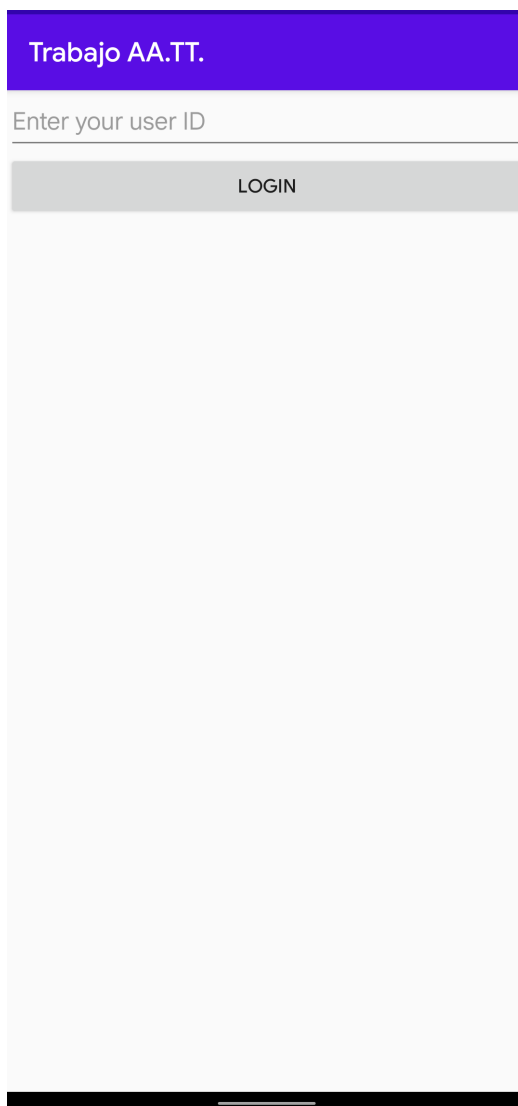
Hemos querido desarrollar una aplicación de mensajería en tiempo real, donde se almacenen todos los mensajes enviados en la nube y se muestran siempre por pantalla. En este caso lo hemos enfocado a una aplicación donde los alumnos y profesores tengan tanto un grupo en particular como un grupo en conjunto para preguntar posibles dudas de la asignatura de una manera fácil y sin exponer ninguna información que pudiese vulnerar la privacidad, como por ejemplo sí se haría en Whatsapp o Telegram.

Esta aplicación es fácilmente escalable a cualquier enfoque distinto que se quiera dar, ya que se pueden crear más usuarios, grupos y proteger con contraseña los inicios de sesión, además, se tiene la posibilidad de implementar llamadas de voz, videollamadas y envío de mensajes multimedia.

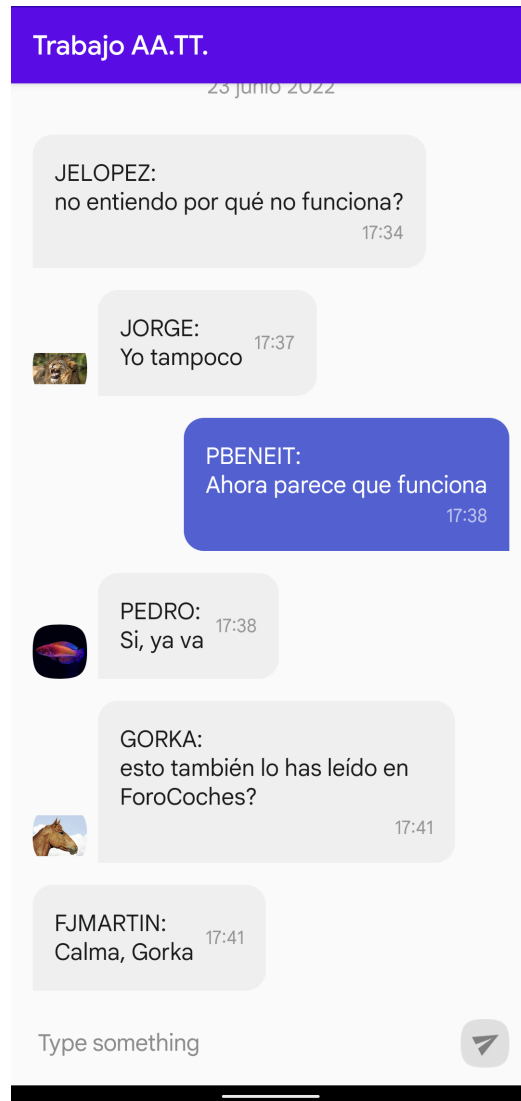
## 2. Funcionamiento de la aplicación

Vamos a ayudarnos de capturas de pantalla desde el dispositivo móvil para la explicación del funcionamiento de la aplicación.

En primer lugar, observamos la pantalla inicial cuando se lanza la aplicación, donde se pide que se introduzca un usuario. En este caso tenemos tres usuarios con el rol de alumnos(jorge, gorka y pedro) y otros tres con el rol de profesores(pbeneit, jelopez y fjmartin).



Los alumnos tienen un grupo con el resto de alumnos y otro junto con los profesores.  
Si iniciamos sesión como profesor tendríamos un grupo con el resto de profesores y el grupo junto con los alumnos.  
Cuando entramos en cualquiera de los grupos, vemos la siguiente interfaz:



Ahora solo hace falta mandar un mensaje para poder comunicarse con todos los integrantes del grupo.

### 3. Explicación algorítmica

La aplicación se sirve de diferentes rutinas auxiliares para articular la funcionalidad de mensajería.

En primer lugar, hemos creado un adaptador para poder establecer una comunicación entre el grupo elegido y la pantalla de selección de grupo. A continuación, en la carpeta `models`, hacemos un `override` de métodos ya definidos para poder operar con objetos de tipo mensaje y usuario con mayor facilidad.

Para terminar con los archivos auxiliares, nos topamos con `Constants.java` en la carpeta `utils`. En este archivo se especifican valores necesarios para poder usar los servicios de Comet Chat, como la clave de la API y el ID de la aplicación.

Si le echamos un vistazo a los archivos que cargan con la mayor parte del peso de la aplicación, visualizamos 3: `ChatActivity.java`, `GroupListActivity.java` y `MainActivity.java`.

- `MainActivity.java`: En `onCreate()` llamamos a `initCometChat()` para poder usar la infraestructura backend que define Comer Chat para sus clientes. También creamos usuarios y grupos, aunque esta tarea solo es necesaria hacerla una vez, por lo que lo hemos comentado para dejar constancia del proceso completo. Finalmente, llamamos a `redirectToGroupListScreen` cuando el usuario se registra correctamente.
- `redirectToGroupListScreen.java`: Aquí creamos un nuevo layout y hacemos un `get` para poder mostrar los grupos a los que pertenece el usuario con el que nos hemos logeado. Si el `fetch` de los grupos se completa sin errores, actualizamos la interfaz de usuario.
- `ChatActivity.java`: En esta clase están definidos los métodos relacionados con el intercambio de mensajes. Se lanza con el método `start` desde la clase previamente explicada, pasando como parámetros el contexto, `groupId` y `groupName`. En `onCreate()`, hacemos un `fetch` de los mensajes que se han mandado previamente, llamamos a `initViews()`, añadimos un listener para poder recibir los mensajes, y llamamos a `sendMessage()` para mandar mensajes.

En `sendMessage()` creamos un objeto de tipo `TextMessage` con el mensaje, el ID del receptor y el tipo de receptor (grupo o usuario, en nuestro caso son todo grupos). Cuando se envíe el mensaje, se usa el adaptador para añadirlo a una lista de mensajes y poder tenerlos todos listos para cuando otro usuario haga login y tenga que cargar todos los mensajes.

# Bibliografia

ChatKit: <https://github.com/stfalcon-studio/ChatKit>

CometChat: <https://www.cometchat.com/docs/android-chat-sdk/overview>