

Trabajo Fin de Máster

Chat application for Android with multiple languages

Le Danny Yang

Tutor: Pere-Pau Vázquez Alcocer

Abril 2020

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



- 1 Motivación y objetivos
- 2 Tecnologías utilizadas
- 3 Diseño de la arquitectura
- 4 Modelo de traducción gratis vs de pago
- 5 Implementación
- 6 Conclusión

Motivación

La principal motivación de este proyecto es permitir a usuarios de cualquier procedencia la capacidad de comunicarse entre ellos.

Objetivo

Implementación de una aplicación de mensajería.

Requisitos mínimos

- 1 Pantalla de registro
- 2 Perfil de usuario
- 3 Conversación privada
- 4 Envío de emoticonos
- 5 Envío de archivos multimedia
- 6 Envío de localización

Requisitos del proyecto

- 1 Diseño de la arquitectura.
- 2 Diseño de una interfaz de usuario usable.
- 3 Implementación de la traducción de mensajes en tiempo real.

Tecnologías utilizadas

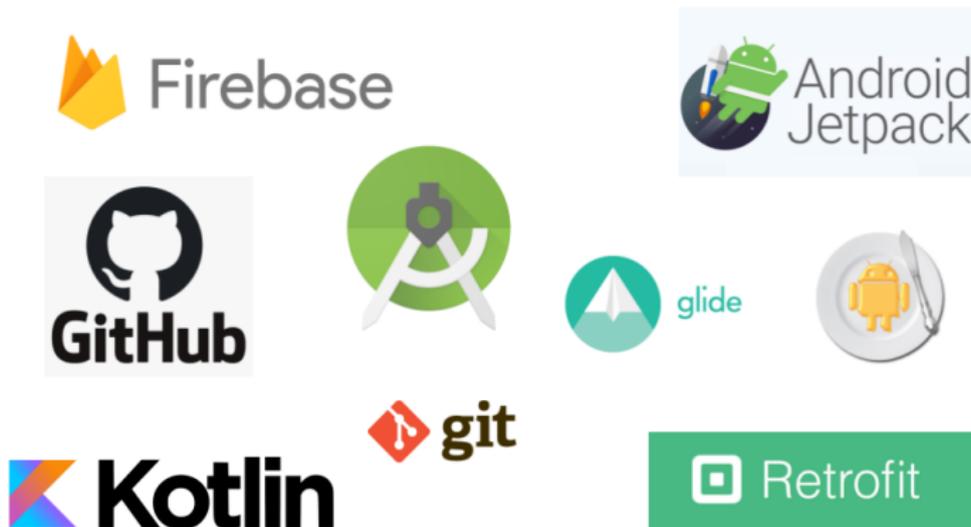


Figura: Tecnologías usadas en el desarrollo del proyecto

Diseño de la arquitectura: MVVM

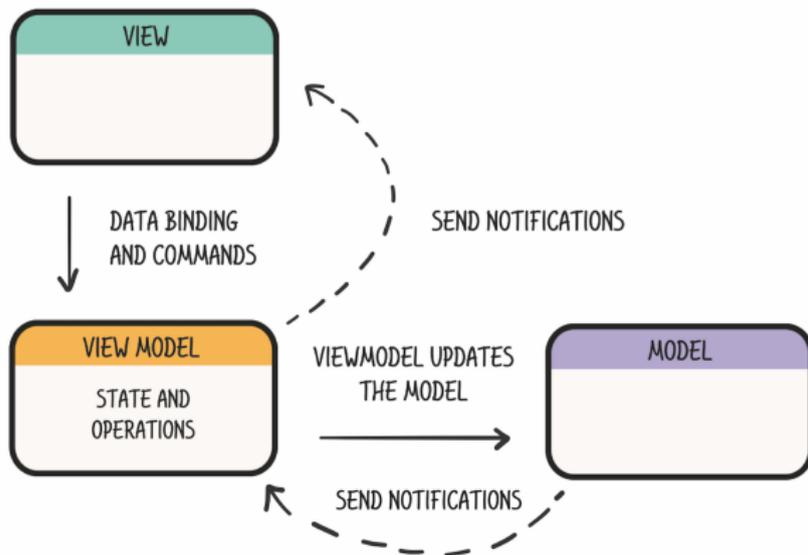


Figura: Esquema de MVVM

Componentes en la arquitectura MVVM

- 1 Model: encargado de procesar la lógica de negocio.
- 2 View: es la interfaz donde se visualiza el estado actual de la aplicación.
- 3 ViewModel: encargado de vincular la información obtenida a partir del model y actualizar la view.

Modelo de traducción gratis vs de pago

¿Por qué?

El desarrollador debe pagar una cuota por mensaje traducido.

Precios mensuales

API Translation: edición básica

Función	Puede optar al uso gratuito	Precio
Detección de idioma	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de NMT)	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de PBMT)	✓	20 USD por millón de caracteres*

Figura: Cuota de Google Cloud Platform

Cómo realizamos ciertas funcionalidades

- 1 Traducción de mensajes.
- 2 Envío de mensajes.

Proceso de traducción local

- 1 Usuario U1 (Español) envía mensaje M en español.
- 2 El mensaje M se traduce al inglés y se guarda en Firebase.
- 3 Usuario U2 (Chino) recibe mensaje M y comprueba si el mensaje original está en su idioma.
- 4 Si lo está, no traduce y recibe el contenido original.
- 5 Si no lo está, obtiene el contenido en inglés y traduce del inglés al chino.

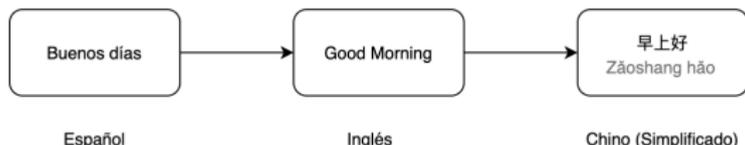


Figura: Esquema de traducción con un lenguaje intermedio (Inglés)

Proceso de envío de mensajes

- 1 Usuario U1 envía mensaje M.
- 2 Usuario U2 está escuchando la conversación por si hay mensajes nuevos.
- 3 Se añade M a la conversación.
- 4 Se notifica del mensaje M a U2, actualiza la conversación y la variable `isReceived` a `True`.
- 5 Se borra mensaje M de Firebase.
- 6 U1 y U2 tiene el mensaje M en sus dispositivos.

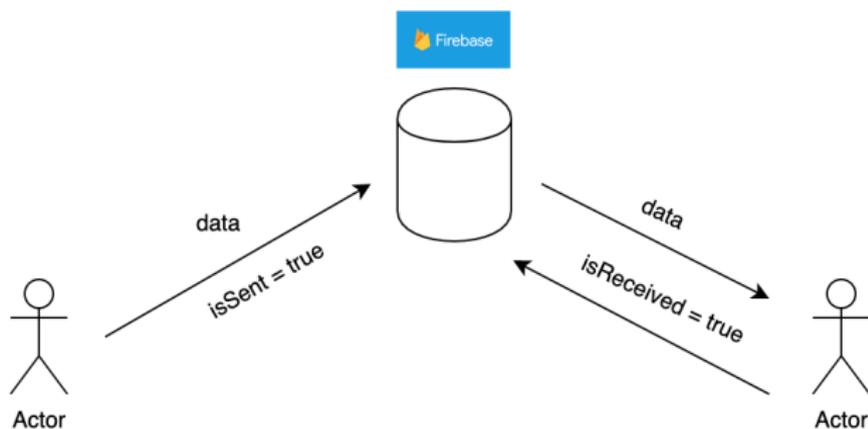
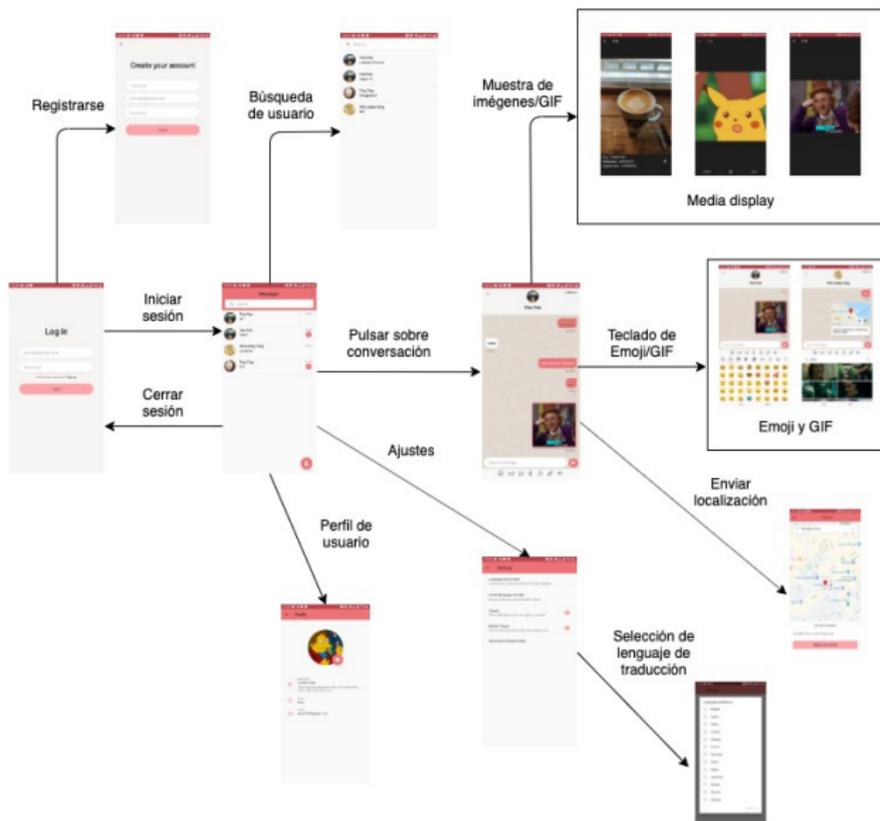


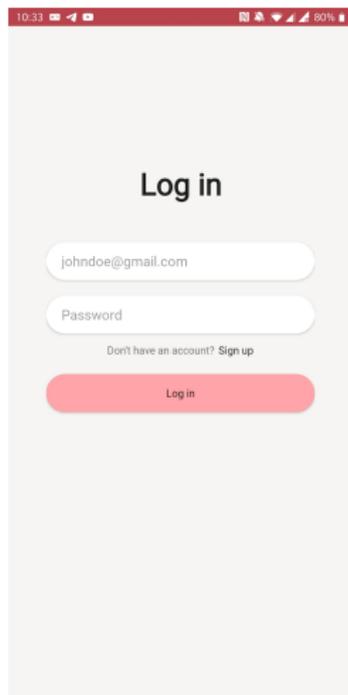
Figura: Diagrama de envío de mensajes

La interfaz de usuario en una aplicación Android, es todo lo que el usuario puede ver e interactuar.

- 1 TextView: elemento que muestra texto al usuario.
- 2 EditText: elemento que permite al usuario añadir texto.
- 3 Button: elemento que permite realizar una acción al pulsarlo.
- 4 ImageView: elemento que permite visualizar imágenes al usuario.
- 5 RecyclerView: elemento de Android que permite mostrar al usuario una lista con componentes.
- 6 Toolbar: se trata de una barra superior en la interfaz.
- 7 SearchView: elemento de búsqueda.
- 8 MapView: elemento de visualización de mapas.

Diagrama de flujos del proyecto





Acciones

- Iniciar sesión.
- Ir a la pantalla de registro.

Requerimientos

- Tener una cuenta ya creada.

10:33 80%

←

Create your account

John Doe

johndoe@gmail.com

Password

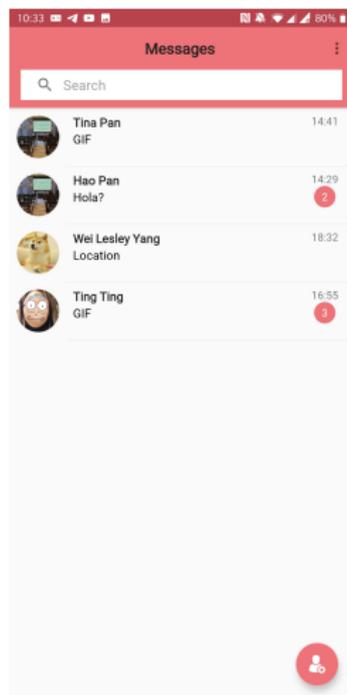
Join us

Acciones

- Crear una cuenta nueva.

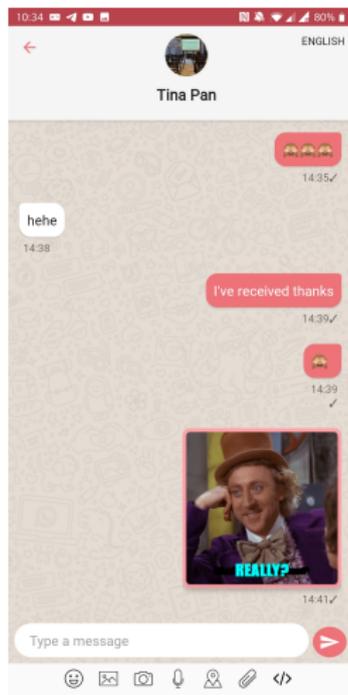
Requerimientos

- Dominio del correo debe ser @gmail.com.
- Usuario no debe existir.
- Longitud de la contraseña mayor de 8 caracteres.



Acciones

- Búsqueda de conversaciones.
- Ir a ChatActivity.
- Ir a UserSearcherActivity.
- Ir a UserProfileActivity.
- Ir a SettingsActivity.
- Cerrar sesión.



Acciones

- Envío de mensajes.
- Envío de emoticonos.
- Envío de GIF.
- Envío de localización.
- Envío de imágenes.

Requerimientos

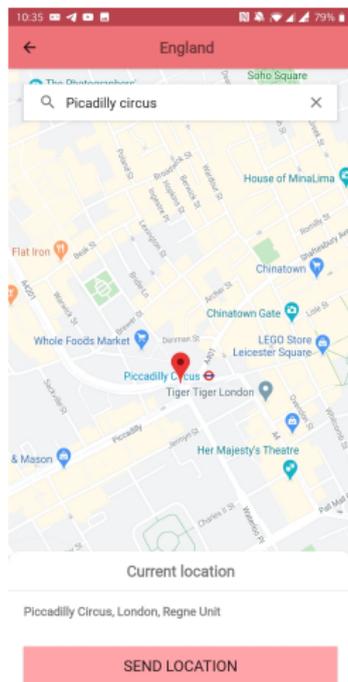
- Acceso a Internet.



Acciones

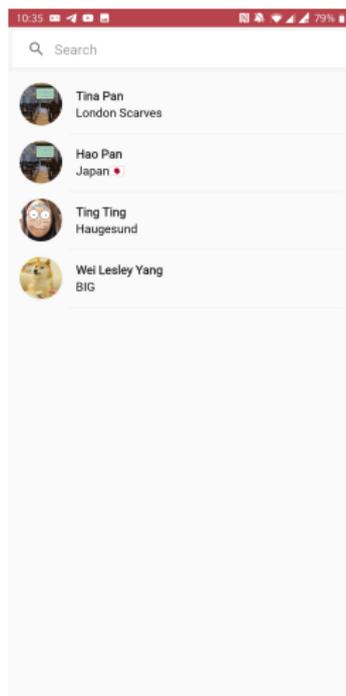
- Visualización de la imagen.
- Descarga de la imagen.

LocationSenderActivity



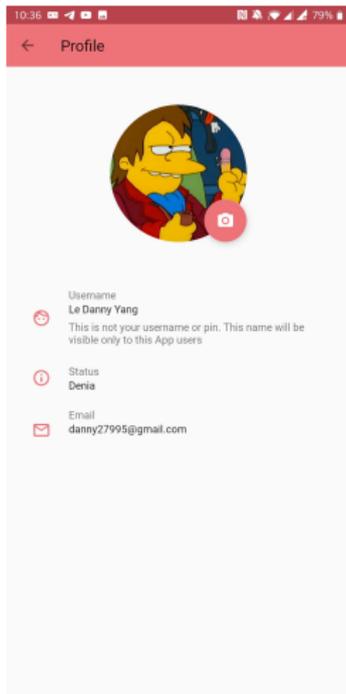
Acciones

- Búsqueda de una dirección.
- Relocalización al pinchar sobre el mapa.
- Envío de localización.



Acciones

- Búsqueda de usuarios.
- Creación de nueva conversación en caso de no existir.

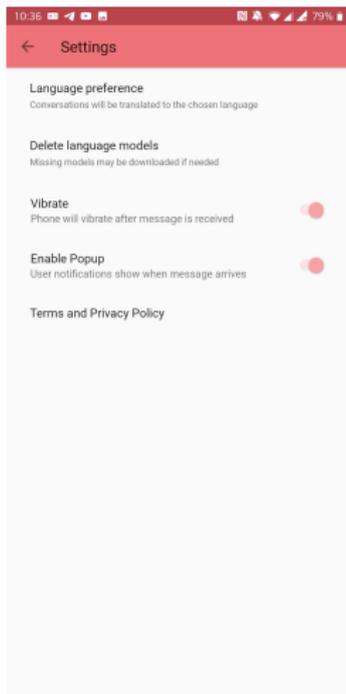


Acciones

- Actualización de la foto de perfil.
- Actualización del nombre de usuario.
- Actualización del estado.

Restricciones

- No se puede actualizar el correo debido a que tiene que ser único.

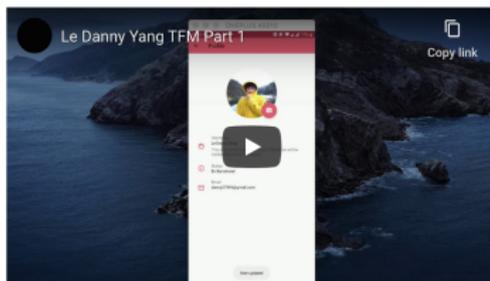


Acciones

- Descarga de modelos de traducción.
- Eliminación de modelos de traducción.

Acciones futuras

- Desactivación de la vibración.
- Desactivación del pop-up al recibir un mensaje.



En este vídeo muestra de manera breve el funcionamiento de la aplicación. Se puede observar la lista de conversaciones que posee el usuario, se muestra también el perfil de usuario así como la acción de actualizar la imagen de perfil. Posteriormente se descarga el modelo de traducción, en este caso, el castellano.

Figura: Página web creada para el TFM

Link: <https://ledannyang.github.io/tfm>

- 1 Envío de audio.
- 2 Envío de documentos.
- 3 Envío de canciones.
- 4 Envío de fragmentos de código.
- 5 Grupos de mensajería.
- 6 Llamadas y videollamadas.
- 7 Localización en directo (Live Location).
- 8 Encriptación de mensajes.

Objetivos cumplidos

- 1 Traducción en tiempo real.
- 2 Creación de una pantalla de inicio de sesión.
- 3 Creación de una pantalla donde modificar datos personales.
- 4 Implementación de funcionalidad básica (envío de emoticonos, imágenes, localización).

- 1 POJO: Plain Old Java Object, representa la clase más trivial.
- 2 Activity: en Android, representa una pantalla.
- 3 GCP: Google Cloud Platform.

Ventajas

- 1 Soporta varios conceptos de la programación moderna.
- 2 Más conciso y expresivo.
- 3 Null Safety.

Desventajas

- 1 Existe menos documentación en Kotlin.
- 2 Comunidad más reducida.

Ejemplo de clase POJO en Java

```
1 public class Person {
2     private String name;
3     private int age;
4
5     public Person(String name, int age) {
6         this.name = name;
7         this.age = age;
8     }
9
10    public String getName() {
11        return name;
12    }
13
14    public void setName(String name) {
15        this.name = name;
16    }
17
18    public int getAge() {
19        return age;
20    }
21
22    public void setAge(int age) {
23        this.age = age;
24    }
25 }
```

Figura: Clase POJO en Java

Ejemplo de clase POJO en Kotlin

```
1 data class Person(var name: String, var age: Int)
```

Figura: Clase POJO en Kotlin

Traducción de mensajes

Proceso de traducción en Cloud

- 1 Usuario U1 (Español) envía mensaje M en su lengua.
- 2 Usuario U2 (Chino) recibe el mensaje M y realiza una llamada a GCP con el mensaje M de U1 y traduce del español al chino.
- 3 Muestra mensaje M en su dispositivo.



Figura: Esquema de traducción sin lenguaje intermedio

Proceso de actualización del perfil de usuario

- 1 Usuario U1 observa la base de datos de Firebase.
- 2 Cuando otro usuario actualice su perfil de usuario, se actualiza Firebase.
- 3 Usuario U1 actualiza los usuarios en su dispositivo.

Notificación de actualización del perfil de usuario

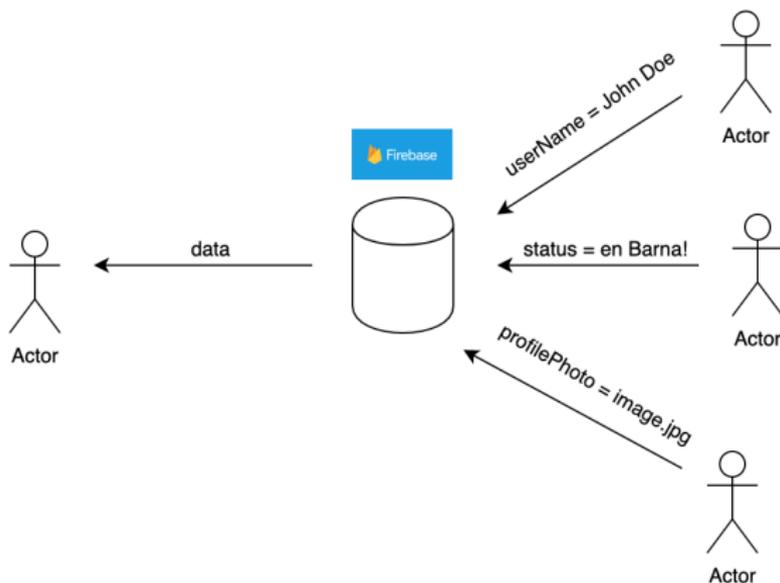


Figura: Diagrama de actualización de perfil de usuario