

## Guía proyecto ForoChat

### Implementar FCM

#### - Crear clase que extiende de **FirebaseMessagingService**

- Sobreescribir el método `onMessageReceived` para mostrar las notificaciones.
- La información a mostrar vendrá por `remoteMessage.getData()` -> `HashMap` con título, body, fecha, etc.
- Mostrar una notificación con esta info

#### - Añadir la clase anterior a **AndroidManifest**

- Con el intent-filter `com.google.firebase.MESSAGING_EVENT`

#### - Obtener el token de notificaciones

- Se puede obtener con el método **`FirebaseMessaging.getInstance().getToken().addOnCompleteListener...`**
- En `LoginActivity`, si el login es correcto, se obtiene el token y se almacena como tú quieras (`Preference`, `BBDD`, Pasarlo por `Intent` de una actividad a otra).
- El token se almacenará en la base de datos, en el documento del foro donde escribamos, para que el resto de usuario pueda mandarnos notificaciones con nuevos mensajes.

### Pantalla Registro

Usar **`FirebaseAuth`**.para el registro y el envío del email de confirmación

### Pantalla Login

Usar **`FirebaseAuth`**.para que el usuario pueda entrar en la App.

- Comprueba que esté registrado y que haya aceptado el correo de confirmación.
- Si todo lo anterior se cumple, recuerda obtener el token de notificaciones

### Listado de foros

Usar **`Firestore`**.

Habrà una colección donde cada documento será un foro.

- El foro puede tener los campos con la información que tú quieras, pero tendrá que tener un nombre para identificarlo.
- El foro, como documento de `Firestore`, tendrá una colección para los mensajes.
- Cada foro puede tener una colección de tokens, para así saber a qué usuarios mandar notificaciones con un mensaje nuevo.

### Pantalla de un foro

Usar **`Firestore`** para mostrar la colección de mensajes de un foro.

- La colección se ha de obtener ordenada por fecha
- El listado de mensajes puede ser de arriba a abajo o al contrario, pero hay que distinguir de alguna forma tus mensajes del resto de usuarios.

### Clase Message

Puede tener los campos que creas necesarios

- Se recomienda un campo de tipo **com.google.firebase.Timestamp**. Es compatible con **Firestore**, y contendrá la fecha y hora en la que se creó el mensaje. Así será fácil ordenar los mensajes por Fecha.
- El Timestamp se crea `Timestamp currentTimeandTime = new Timestamp(new Date());`
- Se adjunta este método con el que puedes formatear esta fecha para mostrarla luego en la App. Recuerda que outputPattern puedes personalizarlo para mostrar la fecha en el formato que quieras:

```
public String parseDateToddMMyyyy(Date time) {
    String outputPattern = "EEE, MMM d '-' HH:mm";
    SimpleDateFormat outputFormat = new SimpleDateFormat(outputPattern);
    outputFormat.setTimeZone(TimeZone.getTimeZone("Europe/Madrid"));
    String str = outputFormat.format(time);
    return str;
}
```

## Envío de notificaciones

Se usa **Retrofit** para el envío de notificaciones.

- Se adjunta como ha de ser el método HTTP para enviar la notificación.
- Tendrás que obtener el SERVER\_TOKEN de tu proyecto en la consola de Firebase-
- La notificación a enviar tiene este formato:

```
    "registration_ids" :
["duG_pAj2T6CW6KwCTEtJUu:APA91bGtd_UJTMN9coOiBc0wYtxaEpvOAx5NqTkQfOqimPHJlQcJyQhUSQ19honl
X5H8ZOALMaN8fCTSBub3-Fpu48aKn2E0uomD4fx9MXP4myC4pXp3vxrJ_MLq7rwawaQ1MJ6-Whm"],
    "data" : {
        "body" : "Notification Body",
        "title": "Notification Title"
    }
}
```

- Donde registraton\_ids es un ArrayList<String> con los token a los que se quiere mandar la notificación
- El objeto data puede tener los campos que quieras, es la información que luego se obtiene en la clase que hereda de FirebaseMessagingService para mostrar la notificación.