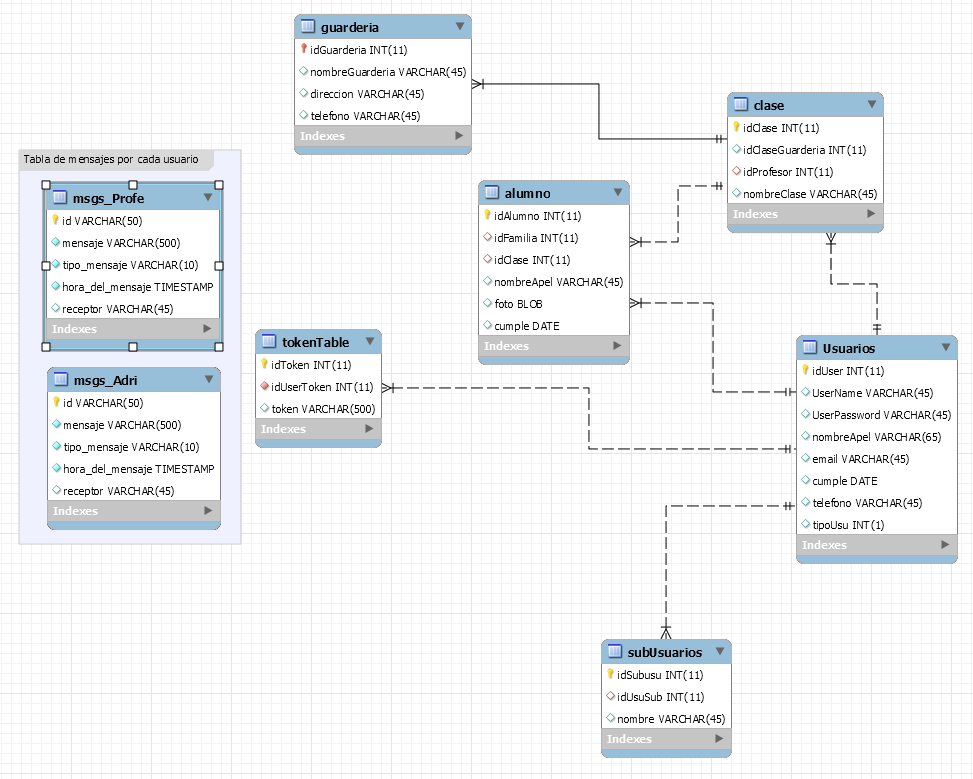
**Base de datos**

Inspirado en [Kinder360](https://kinder360.com/).

Esquema de la base de datos.

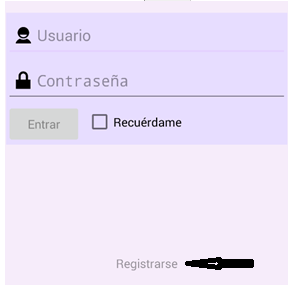


El proyecto emplea la librería Volley para comunicar el servidor con la aplicación, creando tráfico de datos mediante archivos Json, los cuales son generados por scripts.php desplegados en el servidor, y extraídos con la librería Volley medinate JSONRequests.

**Funcionamiento.**

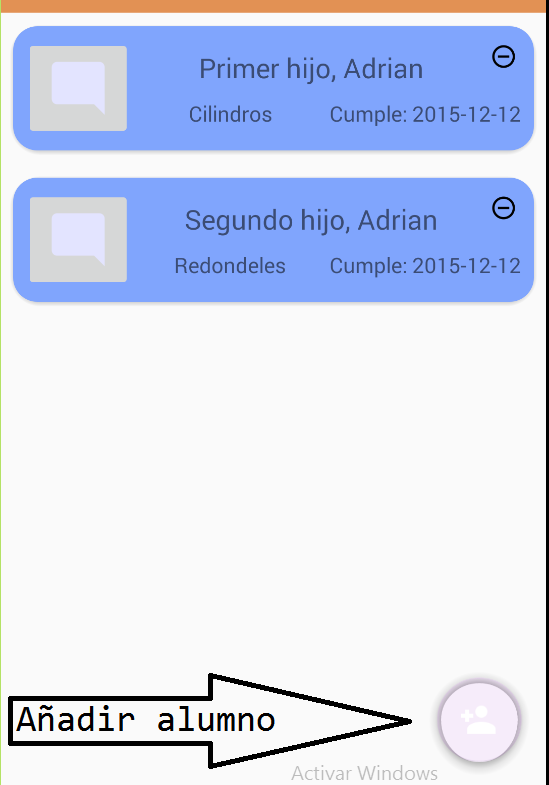
El inicio de sesión consta de un Shared Preferences, cada vez que se inicie sesión comprobará si tiene **g**uardada la contraseña, la cual existirá en caso de que el usuario haya logeado previamente pulsando el Checkbox “Recuérdame”. En la pantalla del login, también se comprobará que el Token de Firebase (se explicará más adelante) esté correctamente actualizado, comparándolo con el último registrado en el SharedPreferences.

En caso de no tener cuenta, se puede crear una pulsando el botón “Registrarse” .

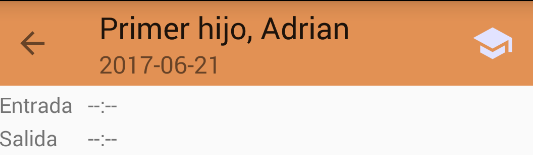


Dependiendo de qué tipo de usuario registremos, (Familia u Educador), iremos a una pantalla u otra desde el Login.

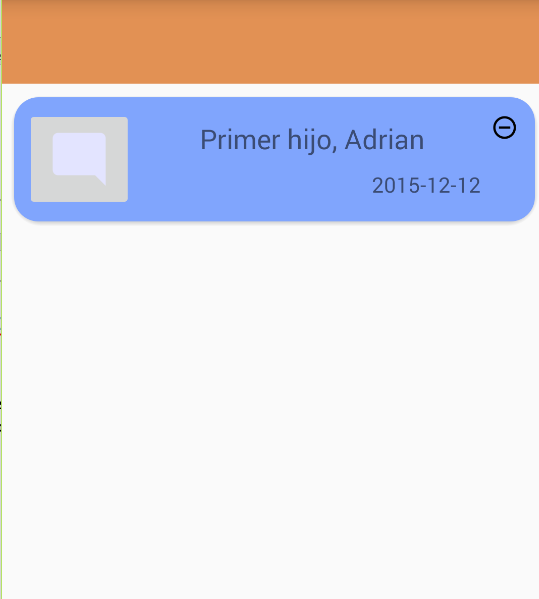
En caso de ser Familiar, pasaremos a una pantalla en la que veremos a todos los alumnos que tenemos registrados en la aplicación, así como la clase en la que se encuentran registrados.



En ella podremos pulsar sobre la fotografía de cualquiera de nuestros hijos registrados, y nos llevará a una tabla de notificaciones del alumno, con un botón (sombrero a la derecha del toolbar) para contactar con el maestro al cargo de él.



Si logeamos como profesorado, obtendremos una lista de las clases de las que estamos encargados, al pulsar sobre una de ellas, tendremos la lista de los alumnos de esta.



Desde esta pantalla podremos contactar con el familiar que registró al alumno, haciendo click sobre su foto.

El sistema de chat crea una tabla de mensajes por cada usuario, la cual contendrá qué tipo de mensaje es (Emisor/receptor). Cada vez que se envíe un mensaje, además de guardarlo en ambas tablas (emisor y receptor), llamará a un script PHP que enviará una notificación vía Firebase al usuario mediante el token del receptor (que está almacenado en la tabla tokenTable.

Los token son la clave única que emplea Firebase para comunicar dispositivos, estos se implementan en nuestra aplicación declarando las clases pertinentes y sus services en el AndroidManifest.xml. y se lanza mediante un phpscript que contendrá otra clave única – Google API KEY -

**Firebase**

Clase FirebaseID – Controlará el token del dispositivo (Si se reinstala la aplicación, el token cambia).

**public class** FireBaseId **extends** FirebaseInstanceIdService {  
 @Override  
 **public void** onTokenRefresh() {  
 **super**.onTokenRefresh();  
 }  
}

Service:

<**service  
 android:name=".Services.FireBaseId"  
 android:enabled="true"  
 android:exported="true"**>  
 <**intent-filter**>  
 <**action android:name="com.google.firebase.INSTANCE\_ID\_EVENT"** />  
 </**intent-filter**>  
</**service**>

FirebaseMensaje - Con esta clase manejaremos las notificaciones que llegan de Firebase (lanzadas desde el script en el servidor).

**public class** FireBaseMensaje **extends** FirebaseMessagingService {  
 @Override  
 **public void** onMessageReceived(RemoteMessage remoteMessage) {  
  
 **super**.onMessageReceived(remoteMessage);  
 String mensaje = remoteMessage.getData().get(**"mensaje"**);  
 String hora = remoteMessage.getData().get(**"hora"**);  
 String cabecera = remoteMessage.getData().get(**"hora"**);  
  
 Message(mensaje, hora);  
 ShowNotification(cabecera, mensaje, hora);  
 }  
 **private void** Message(String mensaje, String hora){  
  
 Intent i = **new** Intent(MessagesActivity.***MESSAGE***);  
 i.putExtra(**"mensaje\_key"**, mensaje);  
 i.putExtra(**"hora\_key"**, hora);  
 LocalBroadcastManager.*getInstance*(getApplicationContext()).sendBroadcast(i);  
 }  
  
  
 *//Para que funcione la notificación dentro de la aplicación* **private void** ShowNotification(String cabecera, String mensaje, String hora){  
 Intent i = **new** Intent(**this**, MessagesActivity.**class**);  
  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(**this**, 0, i, PendingIntent.***FLAG\_ONE\_SHOT***);  
  
 Uri uri = RingtoneManager.*getDefaultUri*(RingtoneManager.***TYPE\_NOTIFICATION***);  
  
  
 Builder builder = **new** Builder(**this**);  
 builder.setAutoCancel(**true**);  
 builder.setContentTitle(cabecera);  
 builder.setContentText(mensaje + hora.substring(11,13));  
 builder.setSound(uri);  
 builder.setSmallIcon(R.drawable.***ic\_launcher***);  
 builder.setContentIntent(pendingIntent);  
 NotificationManager notificationManager = (NotificationManager) getSystemService(Context.***NOTIFICATION\_SERVICE***);  
  
 Random random = **new** Random();  
  
 *//Para que no sobreescriba la antigua notificación le meto un random* notificationManager.notify(random.nextInt(),builder.build());  
 }  
}

Service:

<**service  
 android:name=".Services.FireBaseMensaje"  
 android:enabled="true"  
 android:exported="true"**>  
 <**intent-filter**>  
 <**action android:name="com.google.firebase.MESSAGING\_EVENT"** />  
 </**intent-filter**>  
</**service**>