

# MANUAL DE TÉCNICO

## APLICACIÓN DE RECONOCIMIENTO FACIAL

### Manual Técnico

**Para:**

**LIGA DEPORTIVA  
PARROQUIAL DE  
LICAN**

Versión 1.0

**Preparado por:**

Elvis Bladimir Vargas Amaguaya [evargas.fis@unach.edu.ec](mailto:evargas.fis@unach.edu.ec)

Jessica Margoth HumalaYucta [jessicahumala@gmail.com](mailto:jessicahumala@gmail.com)

**Fecha:** Riobamba - Diciembre 2020

## Contenido

Contenido.....	2
<b>1. INTRODUCCIÓN .....</b>	<b>4</b>
<b>2. OBJETIVO DE ESTE MANUAL.....</b>	<b>4</b>
<b>3. DIRIGIDO .....</b>	<b>4</b>
<b>4. CONOCIMIENTOS PREVIOS .....</b>	<b>4</b>
<b>5. ESPECIFICACIONES TÉCNICAS.....</b>	<b>5</b>
5.1    HARDWARE.....	5
5.2    SOFTWARE .....	5
<b>6. DESCRIPCIÓN DE LA APLICACIÓN MOVIL FACEIDENT .....</b>	<b>5</b>
<b>7. DESCRIPCIÓN DEL COTENIDO DE LA APLICACIÓN MOVIL FACEIDENT .....</b>	<b>5</b>
7.1    ANDROIDSTUDIO 3.4.2.....	6
<b>8. FUNCIONES PRINCIPALES DEL LA APLICACION FACEIDENT .....</b>	<b>6</b>
8.1    CREACIÓN DE LA APLICACION EN ANDROIDSTUDIO .....	6
8.2    CONEXIÓN DE BASE DE DATOS.....	6
8.3    CREACION DEL PATRÓN DE LA APLICACION MOVIL .....	7
8.4    CREACION DE VIEWMODEL .....	7
8.5    CAPA ACCESO A DATOS .....	9
8.6    VISTAS DE LA APLICACIÓN FACEIDENT. ....	11
<b>9. DISTRIBUCIÓN DE ROLES DE USUARIO .....</b>	<b>12</b>

## TABLA DE ILUSTRACIONES

<b>Ilustración 1.</b> Plataforma de desarrollo.....	6
<b>Ilustración 2.</b> Creación del proyecto en Android .....	6
<b>Ilustración 3.</b> Conexión con la base de datos .....	6
<b>Ilustración 4.</b> Estructura del proyecto.....	7
<b>Ilustración 5.</b> ViewModels para la aplicación FACEIDENT .....	8
<b>Ilustración 6.</b> Ejemplo de ViewModel.....	8
<b>Ilustración 7.</b> Código acceso a datos .....	9
<b>Ilustración 8.</b> Ejemplo capa de la aplicación FACEIDENT .....	9
<b>Ilustración 9.</b> Vistas de la aplicación.....	11
<b>Ilustración 10.</b> Ejemplo de vista.....	11

## **1. INTRODUCCIÓN**

La aplicación móvil de reconocimiento facial FACEIDENT para la Liga Deportiva Parroquial de Licán (LDPL) está desarrollada de forma que permita que la información contenida pueda confiable, editada y/o modificada por el personal encargado de la institución. Para alcanzar estos propósitos se ha hecho uso de JAVA que es un lenguaje que se adecua a las nuevas necesidades de las aplicaciones móviles actuales, sin la necesidad de actualizaciones, a través de esta tecnología la administración y seguridad de la información se dará de forma centralizada y segura de datos, al mismo tiempo facilita la actualización eficiente de dicha información.

En cuanto a este manual se ha considerado incluir todos los aspectos técnicos necesarios para el manejo y control de la aplicación para los directivos de la LDPL.

## **2. OBJETIVO DE ESTE MANUAL**

El objetivo primordial de este Manual es ayudar y guiar al técnico a informarse y utilizar herramientas para que la aplicación FACEIDENT de la LDPL entre en producción (ejecución), para de esa manera poder hacer uso de la información deseada para poder despejar todas las dudas existentes y para poder comprender:

- Guía para gestión de herramientas para poner en funcionamiento la aplicación móvil de la LDPL.
- Conocer cómo utilizar la aplicación, mediante una descripción detallada e ilustrada de las opciones.
- Conocer el alcance de toda la información por medio de una explicación detallada e ilustrada de cada una de las páginas que lo conforman el manual técnico.

## **3. DIRIGIDO**

Este manual está orientado a los técnicos u otros tipos de personal encargado de la LDPL. Solamente dichas personas están autorizadas a realizar modificaciones en el sistema.

Una vez finalizado el proyecto, los directivos de la LDPL está encargado de definir políticas, normas, etc. para la administración de la aplicación FACEIDENT.

También a través de este manual el personal podrá estar en la capacidad de supervisar el cumplimiento de políticas, normas, etc. que permitan el correcto funcionamiento de los Sistemas.

Definir conjuntamente con los departamentos pertinentes, los contenidos o cambios para la aplicación móvil para que también de igual forma puedan ser capacitados en herramientas necesarias para el mantenimiento y ejecución

## **4. CONOCIMIENTOS PREVIOS**

Los conocimientos mínimos que deben tener las personas que operarán la aplicación móvil y deberán utilizar este manual son:

- Conocimientos básicos acerca de Programas Utilitarios
- Conocimientos básicos en adobe acrobat
- Conocimiento básico de Internet

- Conocimiento básico de Windows
- Conocimientos básicos de un teléfono celular

## **5. ESPECIFICACIONES TÉCNICAS**

Para la implementación de la aplicación móvil para la preservación y validación de la información obtenida en la LDPL para lo cual requerimos lo siguiente:

### **5.1 HARDWARE**

#### **Cliente Requerido**

- Teléfono celular con una capacidad mínima de 2GB en Ram y un almacenamiento de 16GB o posterior.

#### **Servidor**

- Se recomienda que se utilicen los requerimientos expuestos anteriormente para la mejor funcionalidad del Sistema FACEIDENT.

### **5.2 SOFTWARE**

- El software requerido para correr la aplicación móvil FACEIDENT es Android.
- Acceso al internet durante la ejecución de la aplicación a través del teléfono celular.

## **6. DESCRIPCIÓN DE LA APLICACIÓN MOVIL FACEIDENT**

El Sistema constará de toda la información preservada que se maneja en la LDPL, tales como: identificación de jugadores por medio de reconocimiento facial en tiempo real, contendrá una interfaz atractiva para los usuarios de esta aplicación, la cual será aprobada y validada por los directivos de la institución mencionada.

## **7. DESCRIPCIÓN DEL COTENIDO DE LA APLICACIÓN MOVIL FACEIDENT**

Para la realización de esta aplicación móvil denominada FACEIDENT se basó fundamentalmente en las necesidades que tenía la LDPL, aquí pueden validar toda la información relacionada con su funcionamiento en general. Una de los requerimientos de la aplicación es que sea fácil de navegar y utilizar para el usuario a través de su interfaz, es por este motivo que esta aplicación tiene las siguientes características.

- ✓ Plataforma AndrodiStudio 3.4.2.
  - Imágenes
  - Módulos
  - Librerías

A continuación, se tiene toda la información fundamental de cada uno de estos aspectos.

## 7.1 ANDROIDSTUDIO 3.4.2

Es un IDE, una interfaz de desarrollo. En realidad, es una especie de escritorio de trabajo para un desarrollador. Allí se encuentra nuestro proyecto, las carpetas del mismo, los archivos que hay en él, y todo lo necesario para acabar creando la aplicación. Lo mejor de Android Studio es que ha sido creado por Google y fue presentado hace tan solo unos meses, por lo que no hablamos de una herramienta antigua y nada depurada, sino de un programa muy moderno que encima ha sido creado por los mismos que han creado el sistema operativo.

## 8. FUNCIONES PRINCIPALES DEL LA APLICACION FACEIDENT

Se requiere instalar del programa base para la ejecución de la aplicación móvil, entre ellos tenemos a AndroidStudio 3.4.2.



Ilustración 1. Plataforma de desarrollo

### 8.1 CREACIÓN DE LA APLICACION EN ANDROIDSTUDIO

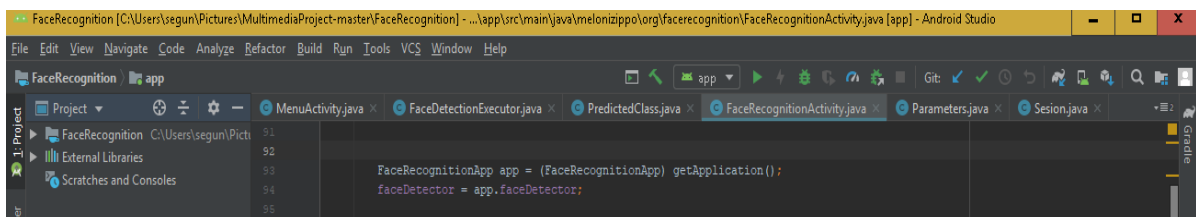


Ilustración 2. Creación del proyecto en Android

### 8.2 CONEXIÓN DE BASE DE DATOS.

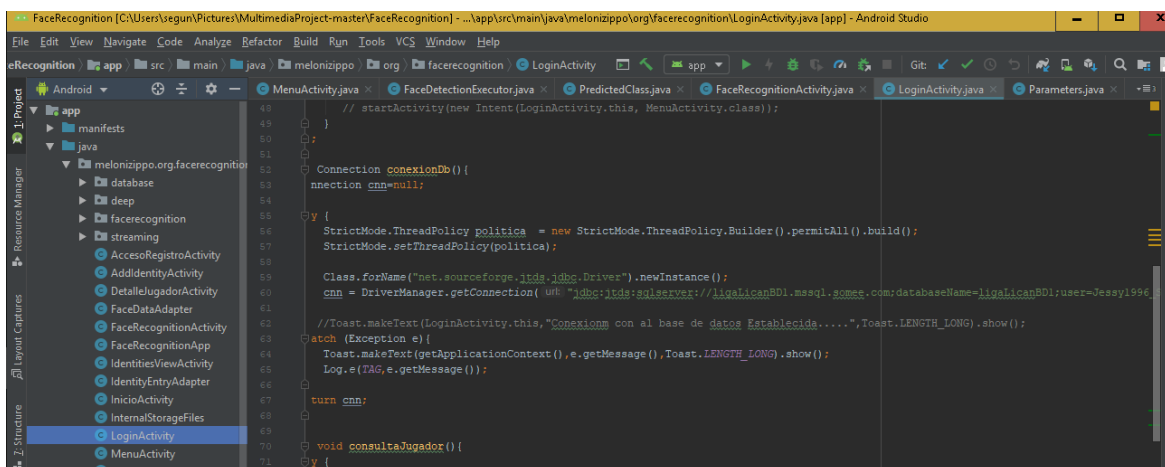


Ilustración 3. Conexión con la base de datos

### 8.3 CREACION DEL PATRÓN DE LA APLICACION MOVIL

Los componentes de la arquitectura de Android son una colección de bibliotecas que te ayudan a diseñar apps sólidas que puedan someterse a prueba y admitan mantenimiento. Comienza con clases para administrar el ciclo de vida de los componentes de la IU y manejar la persistencia de los datos y ViewModel almacena datos relacionados con la IU que no se destruyen en las rotaciones de apps.

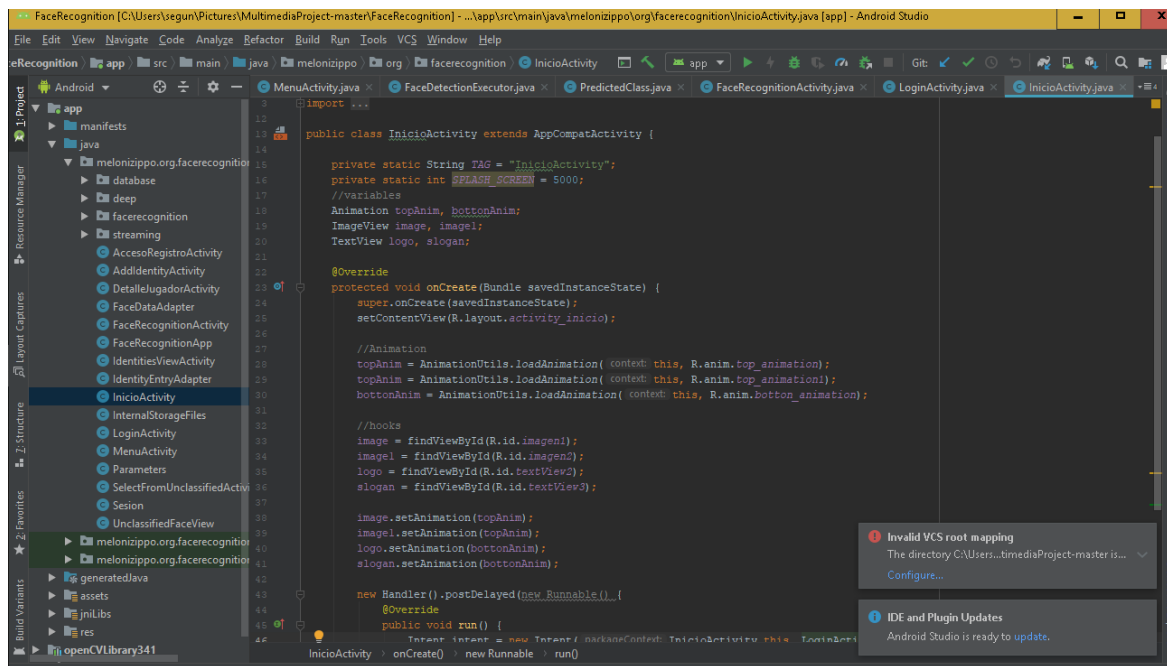


Ilustración 4. Estructura del proyecto

### 8.4 CREACION DE VIEWMODEL

Se diseñó la clase ViewModel a fin de almacenar y administrar datos relacionados con la IU de manera optimizada para los ciclos de vida. La clase ViewModel permite que los datos sobrevivan a cambios de configuración, como las rotaciones de pantallas.

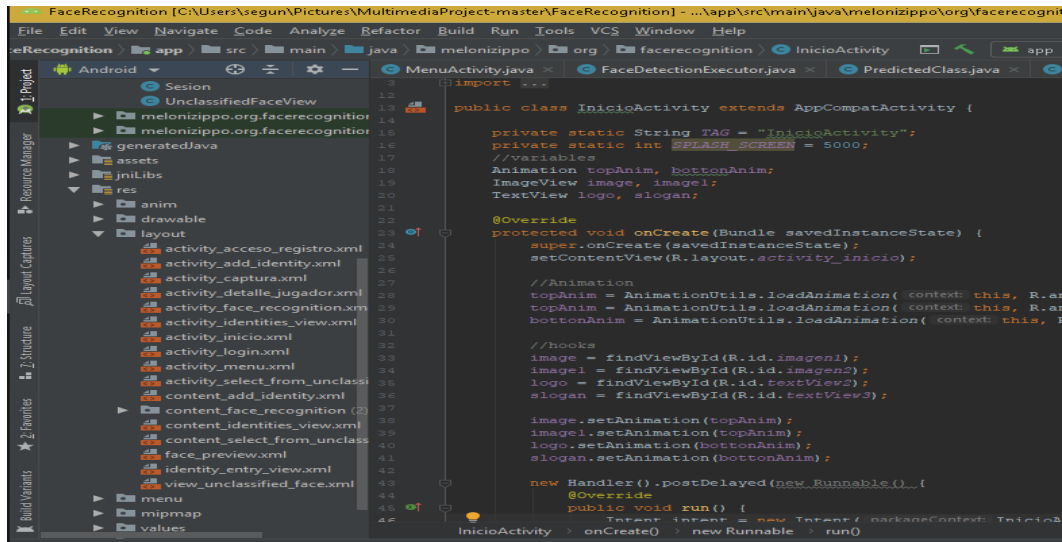


Ilustración 5. ViewModels para la aplicación FACEIDENT

### EJEMPLO DE ViewModels.

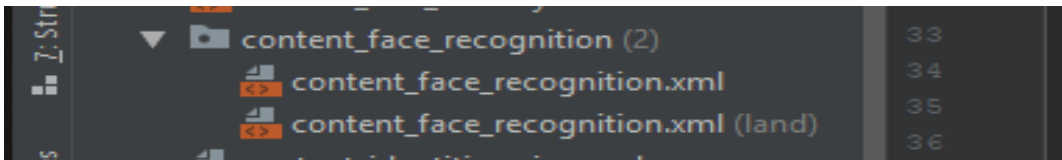


Ilustración 6. Ejemplo de ViewModel

### CÓDIGO DE CREACION DEL VIEWMODEL:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context=".FaceRecognitionActivity"
tools:showIn="@layout/activity_face_recognition">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_constraintLeft_toLeftOf="parent"
    tools:layout_editor_absoluteX="0dp"
    app:layout_constraintBottom_toBottomOf="parent">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
        <TextView
            android:id="@+id/classificationTextView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textAlignment="center"
            android:text="" />
    </ScrollView>
```



```

        <melonizippo.org.facerecognition.streaming.AutoFitSurfaceView
            android:id="@+id/camera_preview"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>
</android.support.constraint.ConstraintLayout>

```

## 8.5 CAPA ACCESO A DATOS

Contiene la lógica de la aplicación., son las clases y métodos que se comunican directamente con la base de datos, todas las funciones que accederán a las tablas y harán los correspondientes SELECT, UPDATE, INSERT, DELETE.

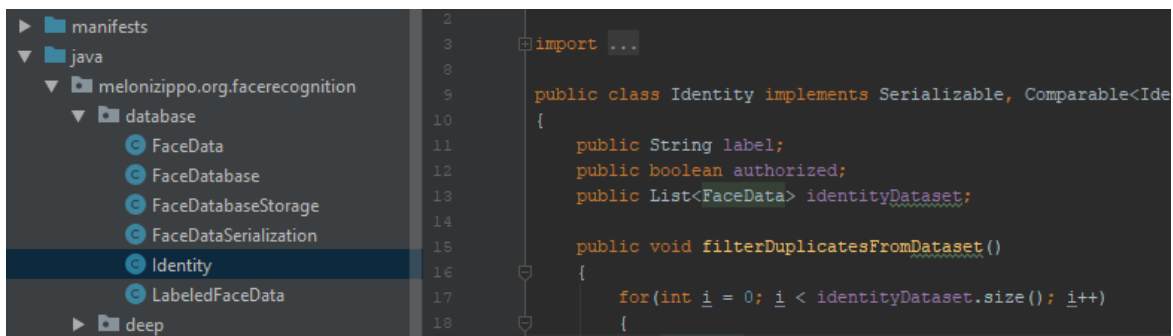


Ilustración 7. Código acceso a datos

## EJEMPLO CAPA DE LA APLICACIÓN FACEIDENT.

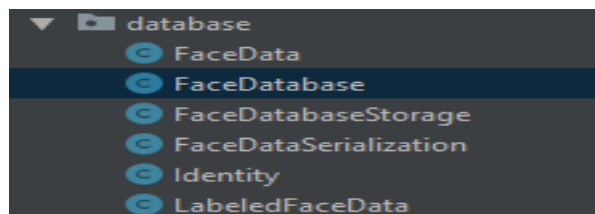


Ilustración 8. Ejemplo capa de la aplicación FACEIDENT

## CÓDIGO DE CREACION DE LA CAPA DE ACCESO A DATOS:

```

public class FaceDatabaseStorage
{
    public static final String IMAGE_DATABASE_NAME =
"image_database.dat";
    private static final String TAG = "FaceDatabaseSerializer";

    private static File internalStorage;

    public static void setFileStorage(File storage)
    {
        internalStorage = storage;
    }

    private static FaceDatabase faceDatabase;
    public static FaceDatabase getFaceDatabase()
    {
        if(faceDatabase == null)
            loadFromInternalStorage();
        return faceDatabase;
    }
}

```

```

public static void setFaceDatabase(FaceDatabase newFaceDatabase)
{
    faceDatabase = newFaceDatabase;
}

public static void loadFromInternalStorage()
{
    try(
        FileInputStream fis =
FaceRecognitionApp.getAppContext().openFileInput(IMAGE_DATABASE_NAME);
        ObjectInputStream ois = new ObjectInputStream(fis))
    {
        FaceDatabase db = (FaceDatabase) ois.readObject();
        Log.i(TAG, "Database loaded");
        faceDatabase = db;
    }
    catch (Exception ex)
    {
        Log.i(TAG, "Database loading failed");
        faceDatabase = new FaceDatabase();
    }
}

public static void storeToInternalStorage()
{
    try(
        FileOutputStream fos =
FaceRecognitionApp.getAppContext().openFileOutput(IMAGE_DATABASE_NAME,
Context.MODE_PRIVATE);
        ObjectOutputStream oos = new ObjectOutputStream(fos)
    )
    {
        FileChannel fileChannel = fos.getChannel();
        FileLock fileLock = fileChannel.lock();

        oos.writeObject(faceDatabase);
        Log.d(TAG, "Database stored");

        fileLock.release();
    }
    catch (Exception ex)
    {
        Log.d(TAG, "Database store failed");
    }
}

public static void clear()
{
    faceDatabase.clear();
    Log.i(TAG, "Database reset");
    storeToInternalStorage();
}
}

```

## 8.6 VISTAS DE LA APLICACIÓN FACEIDENT.

Programaremos lo relativo a la interfaz gráfica, muestra la información al usuario e interactúa con él. Las vistas requerirán los datos se generarán las salidas, tal como nuestra aplicación requiera.

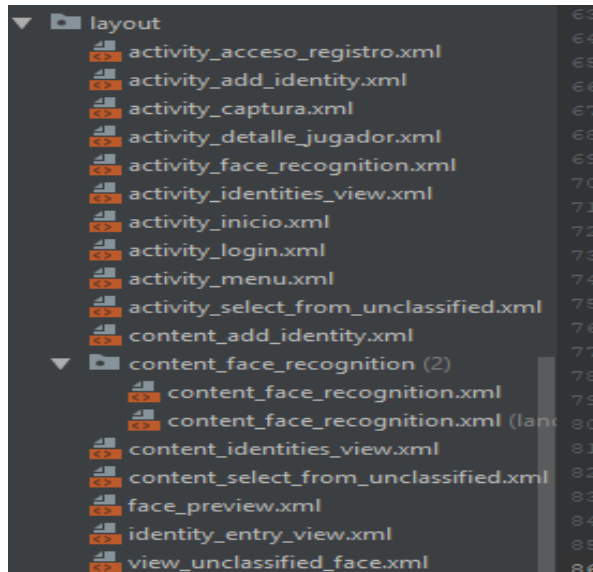


Ilustración 9. Vistas de la aplicación

## EJEMPLO DE VISTA EN LA APLICACIÓN.

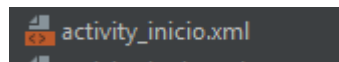


Ilustración 10. Ejemplo de vista

## CÓDIGO DE INICIO DE LA APLICACIÓN:

```
1.<?xml version="1.0" encoding="utf-8"?>
  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo3"
    android:orientation="vertical"
    tools:context=".InicioActivity">

    <ImageView
      android:id="@+id/imagen1"
      android:layout_width="match_parent"
      android:layout_height="251dp"
      android:layout_margin="30dp"
      android:src="@drawable/imagen11"
      tools:layout_editor_absoluteX="30dp"
      tools:layout_editor_absoluteY="41dp" />

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
```

```

</LinearLayout>

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:gravity="center"
    android:text="FaceIdent"
    android:textSize="24sp"

    android:typeface="serif"
    app:fontFamily="More Fonts..."
    tools:layout_editor_absoluteX="30dp"
    tools:layout_editor_absoluteY="571dp" />

<TextView
    android:id="@+id/textView3"

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Universidad Nacional de Chimborazo"
    tools:layout_editor_absoluteX="50dp"
    tools:layout_editor_absoluteY="616dp" />

<ImageView
    android:id="@+id/imagen2"
    android:layout_width="match_parent"
    android:layout_height="234dp"
    android:src="@drawable/img3"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="313dp" />

</LinearLayout>

```

## 9. DISTRIBUCIÓN DE ROLES DE USUARIO

Para la programación de este aplicativo se definieron 2 tipos de usuarios o roles donde cada uno interactúa con la aplicación de diferentes modos que son: administrador, usuario.

- **Administrador:** Este usuario realiza tareas de preservar los datos del sistema, registrar un nuevo usuario, editar información del usuario, asignar acciones y roles al usuario, visualizar a todos los usuarios registrados, acceder al sistema, administrar los archivos del sistema
- **Usuario:** Este usuario realiza tareas de acceder al sistema, verificar la identidad del jugador a través de la aplicación móvil FACEIDENT.