



# Desarrollo de Software para Móviles DSM104

## Parcial Práctico 02

### Porcentaje de realización de ejercicios

Ejercicio 1 (Menú) -> 100%

Ejercicio 2 (Login) -> 100%

Ejercicio 3 (Splash) -> 100%

Ejercicio 4 (CRUD) -> 100%

### Presentado por:

Bonilla Avilés, David Alejandro – BA181927

Cruz Gonzalez, José Roberto -CG181933

Garay Alvarado, Bryan Walberto – GA181935

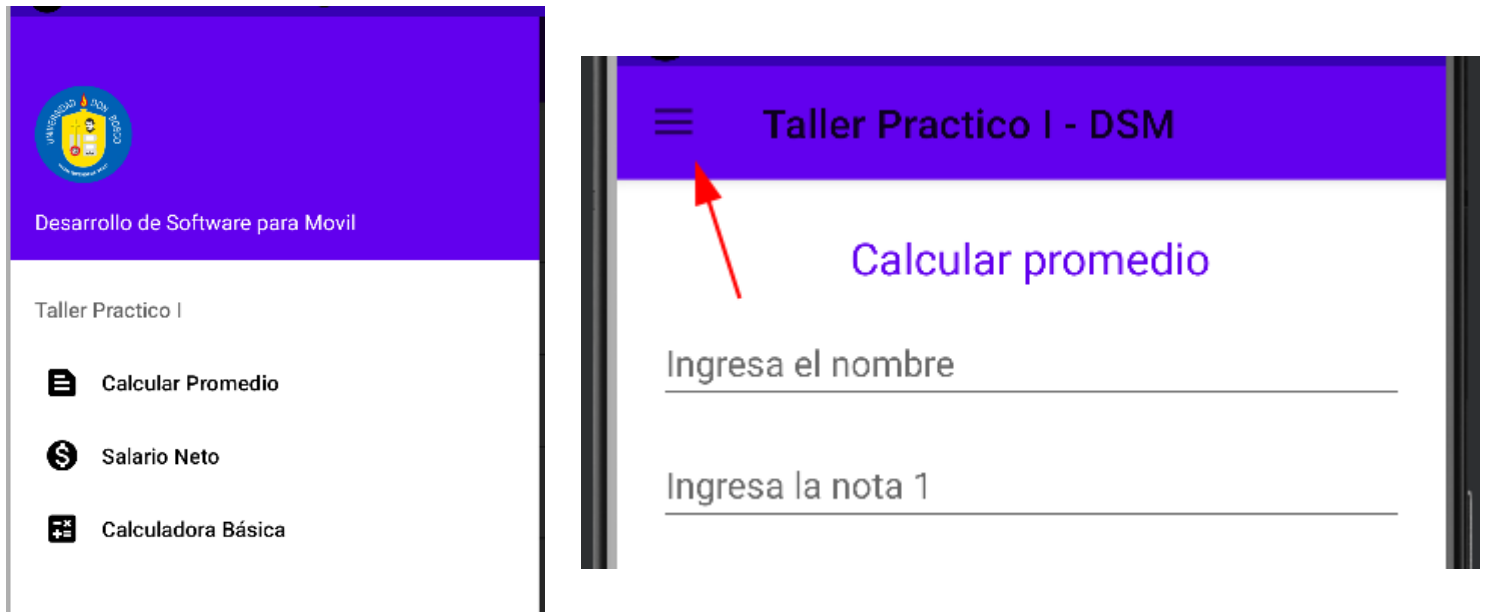
Repositorio:

<https://github.com/tito027/Taller Practico I DSM>

**UNIVERSIDAD DON BOSCO**

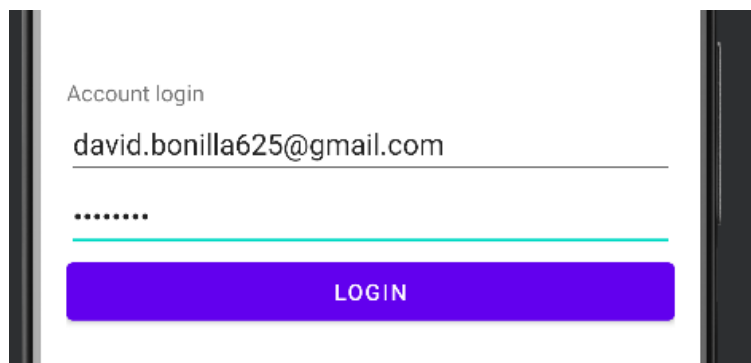
## Ejercicio 1: Menú

Se ha implementado el menú mostrado en clases con las distintas opciones dentro de la aplicación:



## Ejercicio 2: Login

Dentro de este apartado se muestra el login realizado con firebase (usuario/contraseña), se opto por esta alternativa debido a la facilidad en la implementación del backend.



```

David Bonilla
private fun loginUserAccount() {
    progressBar?.setVisibility(View.VISIBLE)
    val email: String
    val password: String
    email = emailTV?.getText().toString()
    password = passwordTV?.getText().toString()
    if (TextUtils.isEmpty(email)) {
        Toast.makeText(applicationContext, text: "Please enter email...", Toast.LENGTH_LONG)
            .show()
        return
    }
    if (TextUtils.isEmpty(password)) {
        Toast.makeText(applicationContext, text: "Please enter password!", Toast.LENGTH_LONG)
            .show()
        return
    }
    mAuth?.signInWithEmailAndPassword(email, password)
        ?.addOnCompleteListener { task ->
            if (task.isSuccessful) {
                Toast.makeText(
                    applicationContext,
                    text: "Login successful!",
                    Toast.LENGTH_LONG
                ).show()
                progressBar?.setVisibility(View.GONE)
                val intent = Intent( packageContext: this@LoginActivity, MainActivity::class.java)
                startActivity(intent)
            } else {
                Toast.makeText(
                    getApplicationContext(),
                    text: "Login failed! Please try again later",
                    Toast.LENGTH_LONG
                ).show()
                progressBar?.setVisibility(View.GONE)
            }
        }
    }
}

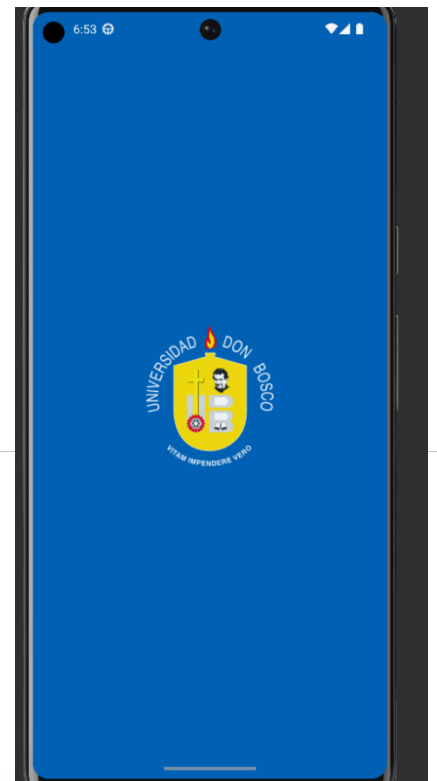
```

### Ejercicio 3: Splash

Dentro de este se solicita una pantalla de carga, splash que dure 20 segundos antes de iniciar la aplicación, que cuente con el nombre de la aplicación, logo de la universidad y nombres de los personas integrantes del equipo.

#### Splash por temas

Dicha pantalla puede ser realizada de varias maneras, en las últimas versiones de Android studio se puede utilizar un tema alternativo , definirlo por defecto y cambiarlo en el onCreate de la actividad main. Esto tiene limitantes ya que solo puede ser aplicada la personalización en la imagen de carga y en el fondo de la pantalla. Se aplico esta forma al inicio de carga de la aplicación.



```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/blue_vdb" />
</layer-list>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Taller Practico I - DSM"
        android:networkSecurityConfig="@xml/network_security_config"
        android:supportRtl="true"
        android:theme="@style/Theme.TallerPracticoIDSM"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
            android:name=".LoginActivity"
            android:exported="false" />
        <activity
            android:name=".RegisterActivity"
            android:exported="true"></activity>
        <activity
            android:name=".MainActivity"
            android:exported="true" />
        <activity
            android:name=".SplashActivity"
            android:exported="true"
            android:theme="@style/SplashTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

manifest > application > activity

```

unknown +3
override fun onCreate(savedInstanceState: Bundle?) {
    //Thread.sleep(20000)
    //Cargar el tema por defecto
    setTheme(R.style.Theme_TallerPracticoIDSM)
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    //setContentView(R.layout.activity_register)

    val toolbar: Toolbar = findViewById(R.id.toolbar_main)
    setSupportActionBar(toolbar)
}

```

## Splash por actividades

Otra forma de realizar un splash para representar la carga de la aplicación es mediante una actividad, se necesita realizar una actividad transitoria que se puede personalizar como se muestra el diseño del lado derecho. Después de definir dicha actividad como la main es necesario redirigir después del tiempo de carga a la actividad requerida.



```

import ...
class SplashActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash)
        Handler().postDelayed({
            val intent = Intent(packageContext, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }, delayMillis: 20000) // espera 2 segundos antes de pasar a MainActivity
    }
}

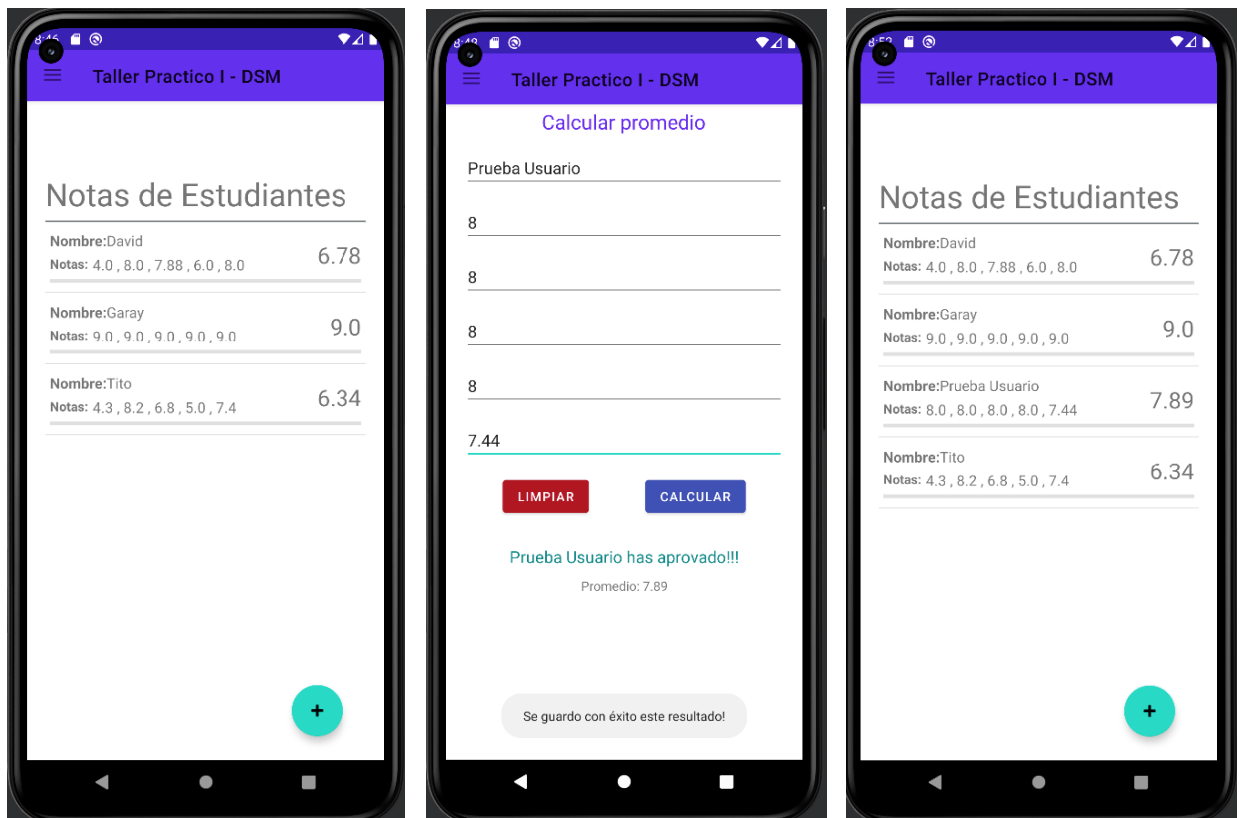
```

## Ejercicio 4

Dentro de este se solicitan dos aplicaciones que hagan uso de un CRUD en firebase, por lo tanto se presentan las siguientes soluciones:

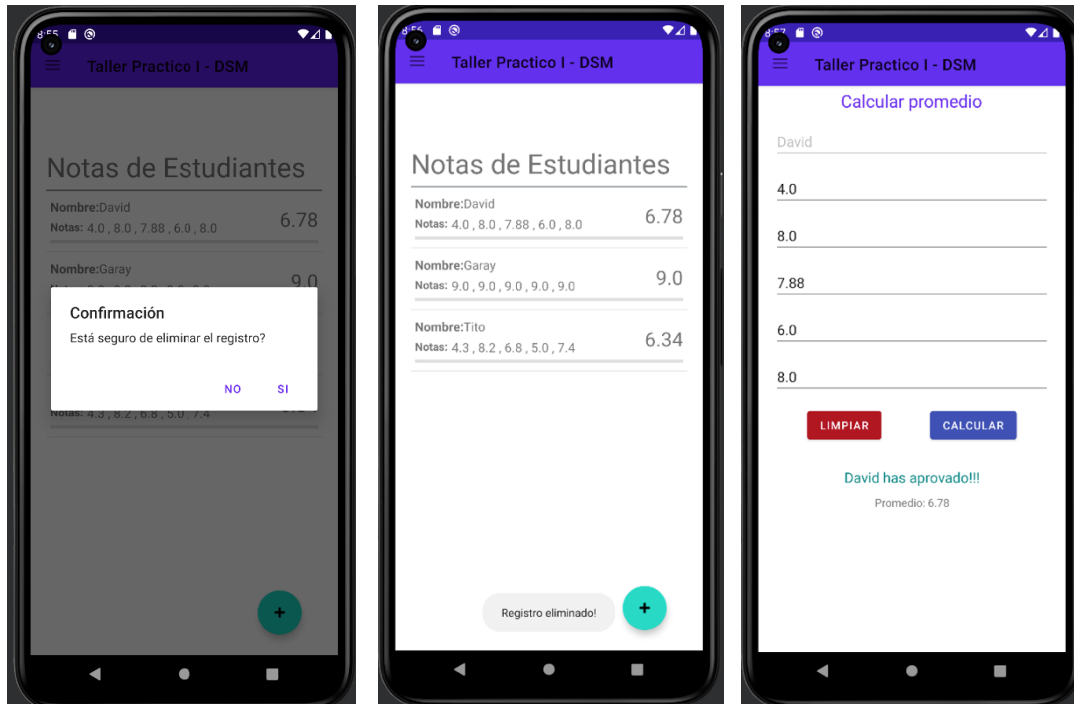
### Promedio de un estudiante CRUD Firebase

La aplicación es en base al ejercicio del desafío anterior solicitando las notas al estudiante además de su nombre para calcular el promedio de este, por ello se ha agregado un listado donde se puede ver todos los registros ingresados a la base desde esta aplicación, en este listado se puede apreciar el nombre del estudiante, las notas obtenidas y el promedio resultante:



También se presentan en las pantallas anteriores el ingreso de un nuevo registro a la base de datos, se pueden apreciar los mensajes de retroalimentación y el nuevo dato en el listado.

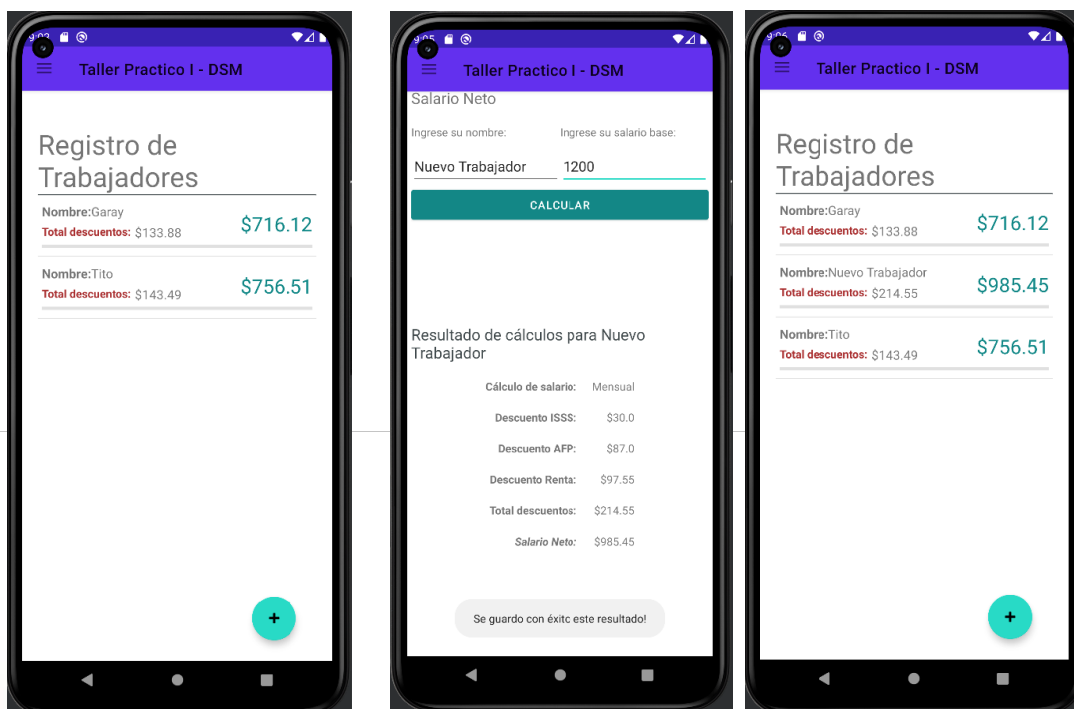
En el listado cada usuario puede editar el registro haciendo un gesto de “click” sobre el elemento que contiene el dato, en caso de mantener este gesto durante un periodo mayor de tiempo se procederá a abrir una ventana para eliminar el registro de la base de datos:



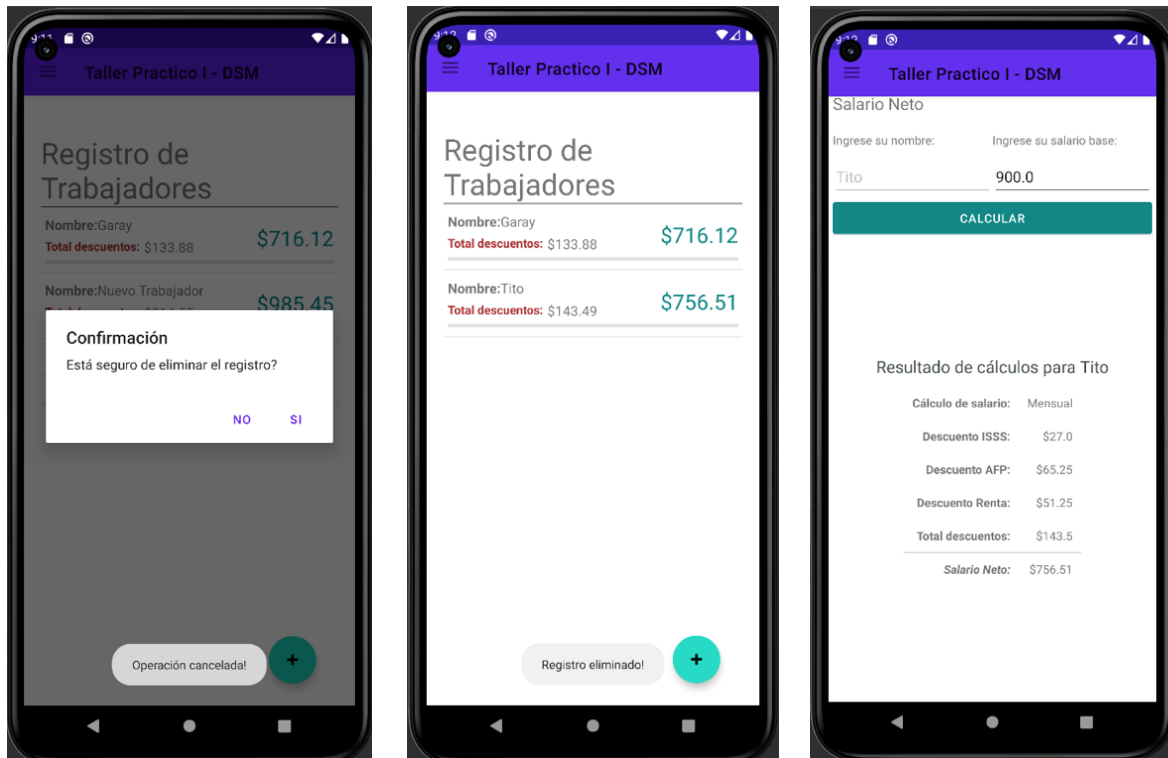
Se puede observar que al momento de querer editar el registro el input que posee el nombre del estudiante se bloquea ya que se está editando el registro asociado a él.

## Salario de trabajadores CRUD Firebase

La aplicación es en base al ejercicio del desafío anterior solicitando el nombre y salario base del trabajador para calcular los descuentos sujetos a las leyes de El Salvador:



Se puede visualizar en las pantallas anteriores cómo al dar click en en el botón flotante de agregar se abre el formulario para ingresar y guardar el nuevo registro y se da un mensaje de retroalimentación al usuario, de igual manera al mantener un evento de click prolongado se procederá a eliminar el dato o con un solo click a modificar los datos del empleado de la siguiente manera:



Se puede apreciar en las pantallas anteriores las operaciones de eliminar y modificar y al momento de modificar el input para el nombre igualmente se encuentra inhabilitado para no cambiar el nombre del trabajador. A continuación se muestra un registro editado.

