

En este proyecto he realizado una aplicación de cuestionarios, en ella el usuario podrán incluir sus propios cuestionarios, lamentablemente no conseguí que leyera correctamente el archivo SQL para que pudiera realizar el propio cuestionario y tampoco pude implementar el ranking a tiempo

XML:

Login

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="#F7F9FB"
    tools:context=".Login">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ring"
        android:contentDescription="Logo" />

    <TextView
        android:id="@+id/tvLoginTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Bienvenido"
        android:textColor="#3E4A61"
        android:textSize="32sp"
        android:fontFamily="sans-serif-medium"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/imageView2" />

<TextView
    android:id="@+id/tvLoginSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp"
    android:text="Por favor, inicia sesión para
continuar"
    android:textColor="#6C7A92"
    android:textSize="16sp"
    android:fontFamily="sans-serif-light"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/tvLoginTitle" />

<TextView
    android:id="@+id/etqUser"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="Usuario"
    android:textColor="#3E4A61"
    android:textSize="18sp"
    android:fontFamily="sans-serif"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/tvLoginSubtitle"
/>

<EditText
    android:id="@+id/user"
    android:layout_width="0dp"
    android:layout_height="48dp"
```

```
        android:layout_marginHorizontal="24dp"
        android:layout_marginTop="8dp"
        android:background="@drawable/edittext_background"
        android:hint="Ingresa su usuario"
        android:padding="12dp"
        android:textColor="#3E4A61"
        android:textSize="16sp"
        android:fontFamily="sans-serif"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/etqUser"
    />
```

```
<TextView
    android:id="@+id/etqPassword"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Contraseña"
    android:textColor="#3E4A61"
    android:textSize="18sp"
    android:fontFamily="sans-serif"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/user" />
```

```
<EditText
    android:id="@+id/password"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:layout_marginHorizontal="24dp"
    android:layout_marginTop="8dp"
    android:background="@drawable/edittext_background"
    android:hint="Ingresa su contraseña"
    android:inputType="textPassword"
    android:padding="12dp"
    android:textColor="#3E4A61"
    android:textSize="16sp"
    android:fontFamily="sans-serif"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/etqPasword" />

<Button
    android:id="@+id/ingresar"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:layout_marginHorizontal="24dp"
    android:layout_marginTop="24dp"
    android:backgroundTint="#4CAF50"
    android:text="Ingresar"
    android:textColor="#FFFFFF"
    android:textSize="18sp"
    android:fontFamily="sans-serif-medium"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password"
/>

<Button
    android:id="@+id/delete"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:layout_marginHorizontal="24dp"
    android:layout_marginTop="16dp"
    android:backgroundTint="#F44336"
    android:text="Borrar Usuario"
    android:textColor="#FFFFFF"
    android:textSize="18sp"
    android:fontFamily="sans-serif-medium"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/ingresar"
/>

<Button
    android:id="@+id/signin"
```

```
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginHorizontal="24dp"
        android:layout_marginTop="16dp"
        android:backgroundTint="#2196F3"
        android:text="Registrarse"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/delete" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Inicio

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F4F4F9"
    tools:context=".inicio">

    <TextView
        android:id="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:fontFamily="sans-serif-medium"
        android:text="Inicio"
        android:textColor="#3E4A61"
        android:textSize="32sp"
        android:textStyle="bold"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btn_cuest"
    android:layout_width="240dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="72dp"
    android:backgroundTint="#4CAF50"
    android:fontFamily="sans-serif"
    android:text="Cuestionarios"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv_title"
/>

<Button
    android:id="@+id/btn_perfil"
    android:layout_width="240dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="116dp"
    android:backgroundTint="#2196F3"
    android:fontFamily="sans-serif"
    android:text="Perfil"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_cuest" />

<Button
```

```
        android:id="@+id/btn_ranking"
        android:layout_width="240dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="148dp"
        android:backgroundTint="#FF9800"
        android:fontFamily="sans-serif"
        android:text="Ranking"
        android:textAllCaps="false"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_perfil" />

<TextView
    android:id="@+id/tv_footer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:fontFamily="sans-serif-light"
    android:text="App Cuestionarios © 2024"
    android:textColor="#888888"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Perfil

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".perfil">

<TextView
    android:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Perfil de Usuario"
    android:textSize="24sp"
    android:layout_marginTop="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/tvUsername"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nombre de usuario: "
    android:textSize="18sp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintTop_toBottomOf="@id/tvTitle"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/tvPuntuacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Puntuación: "
    android:textSize="18sp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
```



```
app:layout_constraintTop_toBottomOf="@id/tvUsername"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Cuestionario

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="#F4F4F9"
    tools:context=".cuestionarios">

    <TextView
        android:id="@+id/tvTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:gravity="center"
        android:text="Cuestionarios Disponibles"
        android:textColor="#3E4A61"
        android:textSize="28sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ScrollView
        android:id="@+id/scrollViewCuestionarios"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_margin="16dp"
```

```
app:layout_constraintBottom_toTopOf="@+id/btn_añadir"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTitle">

    <LinearLayout
        android:id="@+id/linearLayoutCuestionarios"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="8dp"/>
</ScrollView>

<Button
    android:id="@+id/btn_añadir"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    android:backgroundTint="#2196F3"
    android:fontFamily="sans-serif"
    android:text="Añadir Cuestionario"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.027"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/btn_eliminar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="4dp"
    android:text="Eliminar Cuestionarios"

app:layout_constraintBaseline_toBaselineOf="@+id/btn_añadi
r"
```

```
app:layout_constraintStart_toEndOf="@+id/btn_añadir" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Preguntas

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:id="@+id/main">

    <!-- TextView para la pregunta -->
    <TextView
        android:id="@+id/textPregunta"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Pregunta"
        android:textSize="18sp"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:gravity="center" />

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:layout_constraintTop_toBottomOf="@id/textPregunta"
        android:layout_marginTop="16dp">
```

```
<RadioButton
    android:id="@+id/rbtnOpcion1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Opción 1" />

<RadioButton
    android:id="@+id/rbtnOpcion2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Opción 2" />

<RadioButton
    android:id="@+id/rbtnOpcion3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Opción 3" />

<RadioButton
    android:id="@+id/rbtnOpcion4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Opción 4" />
</RadioGroup>

<Button
    android:id="@+id/btnSiguiente"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Siguiente"

app:layout_constraintTop_toBottomOf="@id/radioGroup"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="32dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Resultado

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="#F7F9FB"
    tools:context=".resultado">

    <!-- Logo o icono de resultado -->
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ring"
        android:contentDescription="Logo de resultado" />

    <!-- Título "Resultado" -->
    <TextView
        android:id="@+id/tvResultadoTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Resultado"
        android:textColor="#3E4A61"
        android:textSize="32sp"
        android:fontFamily="sans-serif-medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/imageView2" />
```

```
<!-- Subtítulo debajo del resultado -->
<TextView
    android:id="@+id/tvResultadoSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp"
    android:text="Puntaje final"
    android:textColor="#6C7A92"
    android:textSize="16sp"
    android:fontFamily="sans-serif-light"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tvResultadoTitle"
/>

<!-- Resultados, ejemplo "0/5" -->
<TextView
    android:id="@+id/resultado"
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:background="#FFFFFF"
    android:text="0/10"
    android:textAlignment="center"
    android:textSize="60sp"
    android:textColor="#3E4A61"
    android:layout_gravity="center"
    android:padding="20dp"
    app:layout_constraintBottom_toTopOf="@+id/fin"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_centerHorizontal="true"
    android:elevation="4dp"
    android:radius="20dp" />

<!-- Botón para terminar -->
<Button
    android:id="@+id/fin"
```

```
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginHorizontal="24dp"
        android:layout_marginTop="24dp"
        android:backgroundTint="#4CAF50"
        android:text="Terminar"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:fontFamily="sans-serif-medium"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/resultado"
    />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Kotlin:**Login**

```
package com.example.cuestionariosteven2damb

import android.content.ContentValues
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class Login : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)
    }
}
```

```
        val botonIngresar =  
findViewById<Button>(R.id.ingresar)  
        val botonBorrar = findViewById<Button>(R.id.delete)  
        val botonSing = findViewById<Button>(R.id.signin)  
        val userInsert = findViewById<EditText>(R.id.user)  
        val passInsert =  
findViewById<EditText>(R.id.password)  
        val admin =  
AdminSQLiteOpenHelper(this, "administracion", null, 1)  
        val bd = admin.writableDatabase  
        val registro = ContentValues()  
        //Primer usuario y contraseña  
        registro.put("user", "Paco")  
        registro.put("password", "1234")  
        bd.insert("sesion", null, registro)  
        bd.close()  
  
        // Botón Ingresar a MainActivity  
        botonIngresar.setOnClickListener {  
            val usuario = userInsert.text.toString()  
            val pasword = passInsert.text.toString()  
            if (usuario.isNotEmpty() &&  
pasword.isNotEmpty()) {  
                val bd = admin.writableDatabase  
                val fila = bd.rawQuery("select user,  
password from sesion where user = ? and password = ?",  
arrayOf(usuario, pasword))  
                if (fila.moveToFirst()) {  
                    val intent = Intent(this,  
inicio::class.java)  
                    startActivity(intent)  
                } else {  
                    Toast.makeText(this, "Usuario o  
contraseña incorrectos", Toast.LENGTH_SHORT).show()  
                }  
                fila.close()  
                bd.close()  
            } else {
```



```
        Toast.makeText(this, "Usuario y Contraseña obligatorios", Toast.LENGTH_SHORT).show()
    }
}
// Botón Borrar usuario y contraseña
botonBorrar.setOnClickListener {
    val usuario = userInsert.text.toString()
    val password = passInsert.text.toString()

    if (usuario.isNotEmpty() && password.isNotEmpty()) {
        val bd = admin.writableDatabase

        // Elimina el usuario que coincida con el nombre y la contraseña
        val deletedRows = bd.delete("sesion", "user = ? AND password = ?", arrayOf(usuario, password))

        bd.close()

        if (deletedRows > 0) {
            Toast.makeText(this, "Usuario borrado exitosamente", Toast.LENGTH_SHORT).show()
            userInsert.text.clear()
            passInsert.text.clear()
        } else {
            Toast.makeText(this, "Usuario no encontrado", Toast.LENGTH_SHORT).show()
        }
    } else {
        Toast.makeText(this, "Usuario y contraseña son obligatorios para borrar", Toast.LENGTH_SHORT).show()
    }
}

botonSing.setOnClickListener {
    val usuario = userInsert.text.toString()
    val password = passInsert.text.toString()
}
```

```
        if (usuario.isNotEmpty() &&
password.isNotEmpty()) {
            val bd = admin.writableDatabase

            val registro = ContentValues().apply {
                put("user", usuario)
                put("password", password)
            }

            // Insertar el nuevo usuario en la base de
datos
            val resultado = bd.insert("sesion", null,
registro)

            bd.close()

            if (resultado != -1L) {
                // Si la inserción fue exitosa,
resultado devuelve el ID de la fila insertada
                Toast.makeText(this, "Usuario
registrado exitosamente", Toast.LENGTH_SHORT).show()
            } else {
                // Si la inserción falló, resultado
devuelve -1
                Toast.makeText(this, "Error al
registrar usuario", Toast.LENGTH_SHORT).show()
            }
        } else {
            Toast.makeText(this, "Usuario y contraseña
son obligatorios para registrar",
Toast.LENGTH_SHORT).show()
        }
    }
}
```

Inicio:

```
package com.example.cuestionariosteven2damb
```

```
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class inicio : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_inicio)

        // Configurar los botones

        findViewById<Button>(R.id.btn_cuest).setOnClickListener {
            val intent = Intent(this,
cuestionarios::class.java)
            startActivity(intent)
        }

        findViewById<Button>(R.id.btn_perfil).setOnClickListener {
            val intent = Intent(this, perfil::class.java)
            startActivity(intent)
        }
    }
}
```

Perfil

```
package com.example.cuestionariosteven2damb

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import android.database.sqlite.SQLiteDatabase

class perfil : AppCompatActivity() {
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_perfil)

    // Obtener la referencia de los TextViews donde
    mostraremos los datos
    val tvUsername: TextView =
    findViewById(R.id.tvUsername)
    val tvPuntuacion: TextView =
    findViewById(R.id.tvPuntuacion)

    // Obtener el código de sesión
    val codigoSesion = obtenerCodigoSesion()

    // Si hay código de sesión válido, recuperar los
    datos del usuario
    if (codigoSesion != null) {
        // Instancia de la base de datos
        val adminBD = AdminSQLiteOpenHelper(this,
        "cuestionariosDB", null, 1)
        val db = adminBD.readableDatabase

        // Consultar el usuario con el código de sesión
        val cursor = db.rawQuery("SELECT user,
        puntuacion FROM sesion WHERE codigo = ?",
        arrayOf(codigoSesion.toString()))

        if (cursor.moveToFirst()) {
            // Obtener el nombre de usuario y la
            puntuación
            val nombreUsuario =
            cursor.getString(cursor.getColumnIndexOrThrow("user"))
            val puntuacion =
            cursor.getInt(cursor.getColumnIndexOrThrow("puntuacion"))

            // Mostrar la información en los TextViews
            tvUsername.text = "Nombre de usuario:
            $nombreUsuario"
```

```
        tvPuntuacion.text = "Puntuación: $puntuacion"
    } else {
        // En caso de que no se encuentren datos,
        mostrar un mensaje
        tvUsername.text = "Nombre de usuario: No
        encontrado"
        tvPuntuacion.text = "Puntuación: 0"
    }

    cursor.close()
    db.close()
} else {
    // Si no hay sesión activa, mostrar un mensaje
    de error
    tvUsername.text = "No hay sesión activa."
    tvPuntuacion.text = "Puntuación: N/A"
}
}

// Este método simula obtener el código de sesión del
usuario registrado.
private fun obtenerCodigoSesion(): Int? {
    return 1 // Cambia este valor por el código del
usuario actual registrado.
}
}
```

Cuestionarios

```
package com.example.cuestionariosteven2damb

import android.app.Activity
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.*
```

```
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext

class cuestionarios : AppCompatActivity() {

    private lateinit var layoutCuest: LinearLayout
    private lateinit var btnAnyadir: Button
    private lateinit var btnEliminarDatos: Button
    private lateinit var btnMostrarPreguntas: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_cuestionarios)

        layoutCuest =
findViewById(R.id.linearLayoutCuestionarios)
        btnAnyadir = findViewById(R.id.btn_añadir)
        btnEliminarDatos = findViewById(R.id.btn_eliminar)

        cargarCuestionarios() // Cargar los cuestionarios
desde la base de datos

        btnAnyadir.setOnClickListener { cargarArchivoSQL()
} // Llamar a la función para cargar archivo SQL

        btnEliminarDatos.setOnClickListener {
eliminarDatosTabla("cuestionario") } // Eliminar datos de
la tabla "cuestionario"

        // Botón para mostrar todas las preguntas
        btnMostrarPreguntas = Button(this).apply {
            text = "Mostrar todas las preguntas"
            setOnClickListener { mostrarPreguntas() } //
Mostrar preguntas al hacer clic
        }
    }
}
```

```
        layoutCuest.addView(btnMostrarPreguntas) // Añadir
el botón de preguntas a la vista
    }

    private fun eliminarDatosTabla(nombreTabla: String) {
        val adminBD = AdminSQLiteOpenHelper(this,
            "cuestionariosDB", null, 1)
        val db = adminBD.writableDatabase

        try {
            val filasAfectadas = db.delete(nombreTabla,
                null, null) // Eliminar todos los registros de la tabla
            db.close()

            // Mostrar mensaje de confirmación
            if (filasAfectadas > 0) {
                Toast.makeText(this, "Datos eliminados de
la tabla $nombreTabla.", Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(this, "La tabla $nombreTabla
ya está vacía.", Toast.LENGTH_SHORT).show()
            }

            cargarCuestionarios() // Recargar los
cuestionarios tras eliminar datos

        } catch (e: Exception) {
            // Mostrar error si falla la eliminación
            Toast.makeText(this, "Error al eliminar datos:
${e.message}", Toast.LENGTH_LONG).show()
            Log.e("EliminarDatos", "Error eliminando datos
de la tabla $nombreTabla", e)
        }
    }

    private fun cargarCuestionarios() {
        val adminBD = AdminSQLiteOpenHelper(this,
            "cuestionariosDB", null, 1)
        val db = adminBD.readableDatabase
```

```
        val cursor = db.rawQuery("SELECT id_cuestionario,
nombre FROM cuestionario", null)

        if (cursor.moveToFirst()) {
            layoutCuest.removeAllViews() // Limpiar la
vista antes de cargar los nuevos cuestionarios
            do {
                val idCuestionario =
cursor.getInt(cursor.getColumnIndexOrThrow("id_cuestionari
o"))

                val nombreCuestionario =
cursor.getString(cursor.getColumnIndexOrThrow("nombre"))

                val boton = Button(this).apply {
                    text = nombreCuestionario
                    maxLines = 1
                    ellipsize =
android.text.TextUtils.TruncateAt.END
                    setOnClickListener {
                        abrirCuestionario(idCuestionario,
nombreCuestionario) // Abrir el cuestionario al hacer clic
                    }
                }
                layoutCuest.addView(boton) // Añadir el
botón a la vista
            } while (cursor.moveToNext())
        } else {
            Toast.makeText(this, "No hay cuestionarios
disponibles.", Toast.LENGTH_SHORT).show() // Si no hay
cuestionarios
        }

        cursor.close()
        db.close()
    }

    private fun cargarArchivoSQL() {
```



```
Toast.makeText(this@cuestionarios, "Archivo vacío o no  
válido.", Toast.LENGTH_LONG).show()  
    }  
    return@launch  
}  
  
    val adminBD =  
AdminSQLiteOpenHelper(this@cuestionarios,  
"cuestionariosDB", null, 1)  
    val db = adminBD.writableDatabase  
  
    val sqlStatements =  
sqlContent.split(";") // Dividir por sentencias SQL  
    sqlStatements.forEach { statement  
->  
        val cleanStatement =  
statement.trim()  
        if  
(cleanStatement.isNotEmpty()) {  
            try {  
db.execSQL(cleanStatement) // Ejecutar cada sentencia SQL  
            } catch (sqlEx: Exception)  
{  
                Log.e("SQLExecution",  
"Error al ejecutar sentencia: $cleanStatement", sqlEx)  
            }  
        }  
    }  
  
    withContext(Dispatchers.Main) {  
        cargarCuestionarios() //  
Recargar los cuestionarios después de cargar el archivo  
    }  
} catch (e: Exception) {  
    Log.e("FileProcessing", "Error al  
procesar el archivo SQL", e)  
    withContext(Dispatchers.Main) {
```

```
Toast.makeText(this@cuestionarios, "Error al cargar el
archivo: ${e.message}", Toast.LENGTH_LONG).show()
    }
    }
    }
}

private fun mostrarPreguntas() {
    val adminBD = AdminSQLiteOpenHelper(this,
"cuestionariosDB", null, 1)
    val db = adminBD.readableDatabase

    val cursor = db.rawQuery("SELECT * FROM preguntas",
null)

    val datos = StringBuilder()
    if (cursor.moveToFirst()) {
        do {
            try {
                val idPregunta =
cursor.getInt(cursor.getColumnIndexOrThrow("id_pregunta"))
                val textoPregunta =
cursor.getString(cursor.getColumnIndexOrThrow("pregunta"))
                datos.append("ID: $idPregunta -
Pregunta: $textoPregunta\n") // Añadir pregunta a los
datos
            } catch (e: Exception) {
                Log.e("CursorError", "Error al obtener
los datos del cursor", e)
            }
        } while (cursor.moveToNext())
    } else {
        datos.append("No hay preguntas disponibles.")
    }
    // Si no hay preguntas
}
```

```
        cursor.close()
        db.close()

        val scrollView = ScrollView(this)
        val textView = TextView(this).apply {
            text = datos.toString()
            textSize = 16f
            setPadding(16, 16, 16, 16)
        }
        scrollView.addView(textView)

        AlertDialog.Builder(this)
            .setTitle("Lista de Preguntas")
            .setView(scrollView)
            .setPositiveButton("Cerrar") { dialog, _ ->
dialog.dismiss() } // Botón para cerrar
            .create()
            .show()
    }

    private fun abrirCuestionario(idCuestionario: Int,
nombre: String) {
        AlertDialog.Builder(this)
            .setTitle("Seleccionar Cuestionario")
            .setMessage("¿Quieres abrir el cuestionario '$nombre'?")
            .setPositiveButton("Sí") { _, _ ->
                val intent = Intent(this,
preguntas::class.java).apply {
                    putExtra("id_cuestionario",
idCuestionario) // Pasar ID y nombre del cuestionario
                    putExtra("nombre", nombre)
                }
                startActivity(intent) // Iniciar la
actividad de preguntas
            }
            .setNegativeButton("No", null)
            .show()
    }
}
```

```
}
```

Preguntas

```
package com.example.cuestionariosteven2damb

import android.content.Intent
import android.database.sqlite.SQLiteDatabase
import android.os.Bundle
import android.util.Log
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.app.AlertDialog

class preguntas : AppCompatActivity() {

    private lateinit var textPregunta: TextView
    private lateinit var radioGroup: RadioGroup
    private lateinit var btnSiguiente: Button
    private lateinit var adminBD: AdminSQLiteOpenHelper
    private lateinit var db: SQLiteDatabase

    private var preguntaActual = 0
    private var idCuestionario = 0
    private var dificultad = ""
    private var preguntas = mutableListOf<Map<String,
String>>()
    private var puntos = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_preguntas)

        // Obtener el id del cuestionario y dificultad
        seleccionada
        idCuestionario =
intent.getIntExtra("id_cuestionario", 0)
        dificultad = intent.getStringExtra("dificultad") ?:
""
```

```
// Inicializar las vistas
textPregunta = findViewById(R.id.textPregunta)
radioGroup = findViewById(R.id.radioGroup)
btnSiguiente = findViewById(R.id.btnSiguiente)

// Cargar las preguntas del cuestionario
cargarPreguntas()

// Configurar el botón "Siguiente"
btnSiguiente.setOnClickListener {
    verificarRespuesta()
    if (preguntaActual < preguntas.size) {
        cargarPregunta()
    } else {
        mostrarResultado()
    }
}

}

// Función para cargar las preguntas desde la base de
datos según el cuestionario y dificultad
private fun cargarPreguntas() {
    adminBD = AdminSQLiteOpenHelper(this,
"cuestionariosDB", null, 1)
    db = adminBD.readableDatabase

    val cursor = db.rawQuery(
        "SELECT pregunta, opcion1, opcion2, opcion3,
opcion4, correcta FROM preguntas WHERE id_cuestionario = ?
AND dificultad = ?",
        arrayOf(idCuestionario.toString(), dificultad)
    )

    // Verificamos si la consulta devuelve algún
resultado
    if (cursor.moveToFirst()) {
        do {
            // Verificamos que los índices de las
columnas sean válidos
```

```
        val preguntaIndex =
cursor.getColumnIndex("pregunta")
        val opcion1Index =
cursor.getColumnIndex("opcion1")
        val opcion2Index =
cursor.getColumnIndex("opcion2")
        val opcion3Index =
cursor.getColumnIndex("opcion3")
        val opcion4Index =
cursor.getColumnIndex("opcion4")
        val correctaIndex =
cursor.getColumnIndex("correcta")

        if (preguntaIndex == -1 || opcion1Index ==
-1 || opcion2Index == -1 || opcion3Index == -1 ||
opcion4Index == -1 || correctaIndex == -1) {
            Log.e("BaseDeDatos", "Algunas columnas
no existen en el cursor")
            return // Terminamos la ejecución si
alguna columna no existe
        }

        // Si todos los índices son válidos,
agregamos la pregunta a la lista
        val pregunta = mapOf(
            "pregunta" to
cursor.getString(preguntaIndex),
            "opcion1" to
cursor.getString(opcion1Index),
            "opcion2" to
cursor.getString(opcion2Index),
            "opcion3" to
cursor.getString(opcion3Index),
            "opcion4" to
cursor.getString(opcion4Index),
            "correcta" to
cursor.getString(correctaIndex)
        )
        preguntas.add(pregunta)
```

```
        } while (cursor.moveToNext())

        Log.d("BaseDeDatos", "Preguntas cargadas
correctamente: ${preguntas.size} preguntas encontradas")
    } else {
        Log.e("BaseDeDatos", "No se encontraron
preguntas para este cuestionario y dificultad")
    }

    cursor.close()
    db.close()

    // Verificamos si se han cargado preguntas
    if (preguntas.isEmpty()) {
        Log.e("BaseDeDatos", "La lista de preguntas
está vacía. No se pueden cargar preguntas.")
        Toast.makeText(this, "No se encontraron
preguntas para este cuestionario.",
Toast.LENGTH_SHORT).show()
        return
    }

    // Aleatorizamos el orden de las preguntas
    preguntas.shuffle()

    // Cargar la primera pregunta
    cargarPregunta()
}

// Función para cargar una pregunta aleatoria
private fun cargarPregunta() {
    if (preguntaActual < preguntas.size) {
        val pregunta = preguntas[preguntaActual]
        textPregunta.text = pregunta["pregunta"]
        radioButton.removeAllViews() // Esto elimina
todas las opciones anteriores
```



```
// Crear los RadioButtons para las opciones
val opciones = listOf(
    pregunta["opcion1"],
    pregunta["opcion2"],
    pregunta["opcion3"],
    pregunta["opcion4"]
).shuffled() // Aleatorizamos el orden de las
opciones

for (i in opciones.indices) {
    val radioButton = RadioButton(this).apply {
        text = opciones[i]
        id = i // Asignamos un id único a cada
opción
    }
    radioGroup.addView(radioButton)
}
} else {
    mostrarResultado()
}
}

// Función para verificar la respuesta seleccionada
private fun verificarRespuesta() {
    val pregunta = preguntas[preguntaActual]
    val seleccionada = radioGroup.checkedRadioButtonId
    if (seleccionada != -1) {
        val respuestaSeleccionada =
findViewById<RadioButton>(seleccionada).text.toString()
        if (respuestaSeleccionada ==
pregunta["correcta"]) {
            puntos++
        }
        preguntaActual++
    } else {
        Toast.makeText(this, "Selecciona una opción",
Toast.LENGTH_SHORT).show()
    }
}
```

```
}

// Función para mostrar el resultado final
private fun mostrarResultado() {
    AlertDialog.Builder(this)
        .setTitle("Cuestionario finalizado")
        .setPositiveButton("Aceptar") { _, _ ->
            // Guardar la puntuación final
            Resultados.nota = puntos.toDouble()

            // Redirigir a la actividad de resultados
            val intent = Intent(this,
resultado::class.java)
            startActivity(intent)
            finish()
        }
        .show()
}
}
```

Resultado

```
package com.example.cuestionariosteven2damb

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class resultado : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_resultado)

        val resultado =
findViewById<TextView>(R.id.resultado)
        val finButton = findViewById<Button>(R.id.fin)
```

```
val puntos = intent.getIntExtra("puntos", 0)
resultado.text = "Tu nota es: $puntos/10"

finButton.setOnClickListener {
    // Regresar al inicio
    val intent = Intent(this, inicio::class.java)
    intent.flags = Intent.FLAG_ACTIVITY_CLEAR_TOP
    // or Intent.FLAG_ACTIVITY_NEW_TASK
    startActivity(intent)
    finish() // Finalizar la actividad actual
}
}
```

Base de datos

```
package com.example.cuestionariosteven2damb

import android.content.Context
import android.database.sqlite.SQLiteOpenHelper
import android.database.sqlite.SQLiteDatabase
import
android.database.sqlite.SQLiteDatabase.CursorFactory

class AdminSQLiteOpenHelper(context: Context, name:
String, factory: CursorFactory?,
                           version: Int) :
SQLiteOpenHelper(context, name, factory, version) {

    override fun onCreate(db: SQLiteDatabase) {
        // Creación de la tabla 'sesion'
        db.execSQL("create table sesion(codigo int primary
key, user text, password text, puntuacion int)")

        // Creación de la tabla 'preguntas'
        db.execSQL("create table preguntas(id int primary
key, pregunta text, opcion1 text, opcion2 text, opcion3
```

```
text, opcion4 text, correcta text, dificultad text,
id_cuestionario int)")

    // Creación de la tabla 'cuestionario'
    db.execSQL("create table
cuestionario(id_cuestionario int primary key, nombre
text)")

    // Creación de la tabla 'ranking'
    db.execSQL("create table ranking(id_ranking int
primary key, id_user int, id_cuestionario int, puntuacion
int, fecha date)")
}

    override fun onUpgrade(db: SQLiteDatabase, oldVersion:
Int, newVersion: Int) {
    }
}
```

Clase adicional para calcular resultado

```
package com.example.cuestionariosteven2damb

import androidx.appcompat.app.AppCompatActivity

object Resultados : AppCompatActivity() {
    var nota: Double = 0.0
}
```

Capturas Interfaz:

Login



Bienvenido

Por favor, inicia sesión para continuar

Usuario

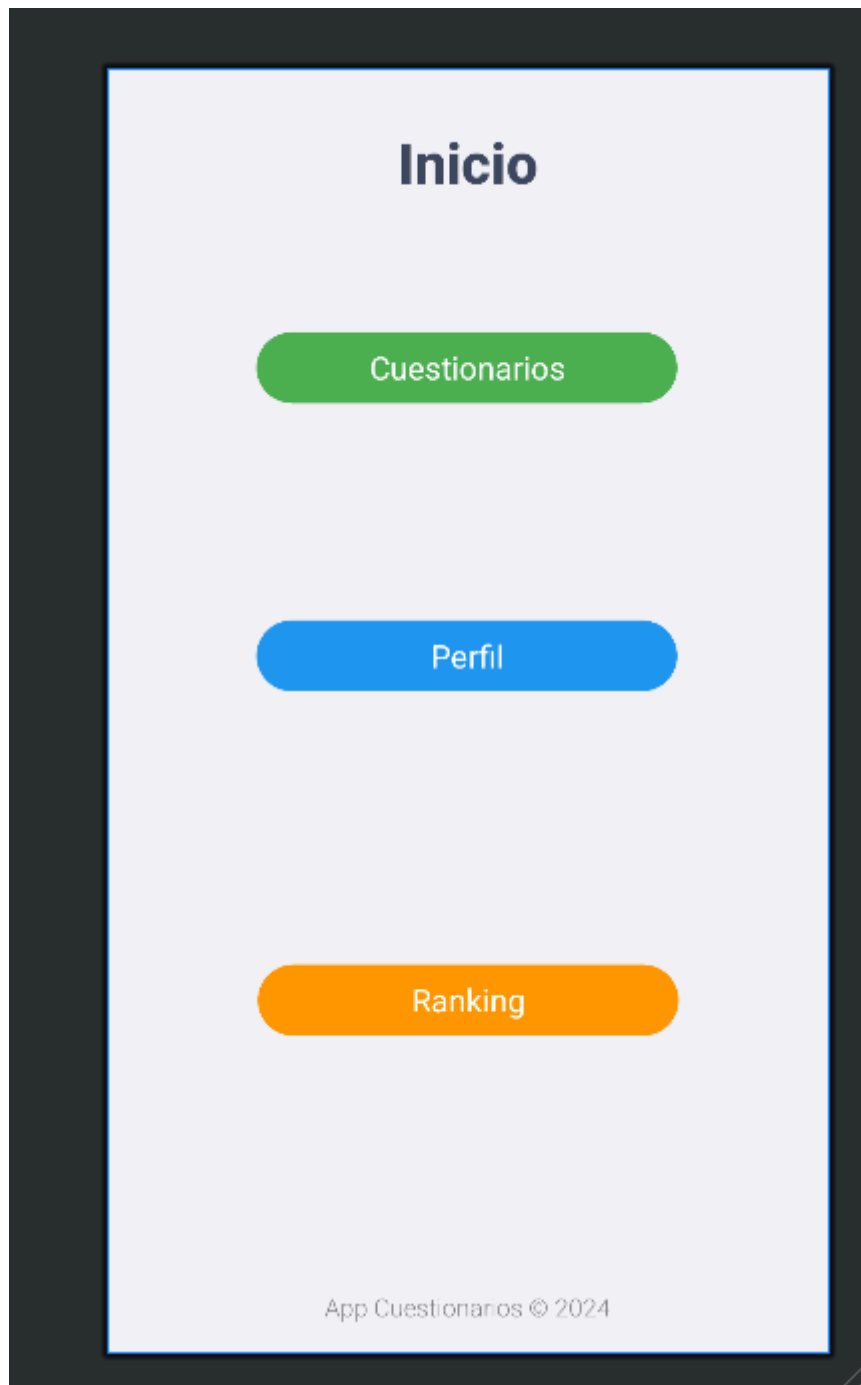
Contraseña

Ingresar

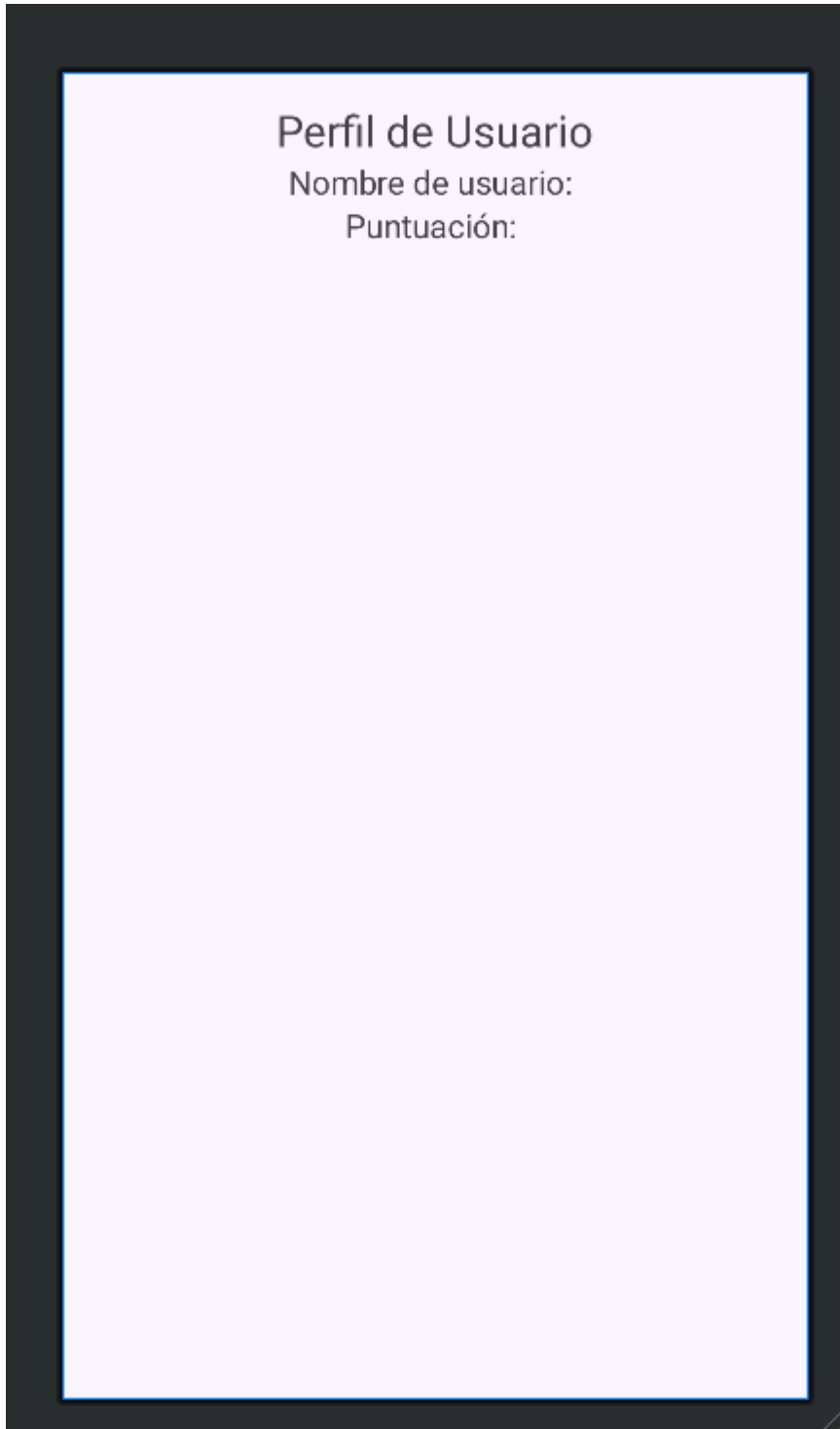
Borrar Usuario

Registrarse

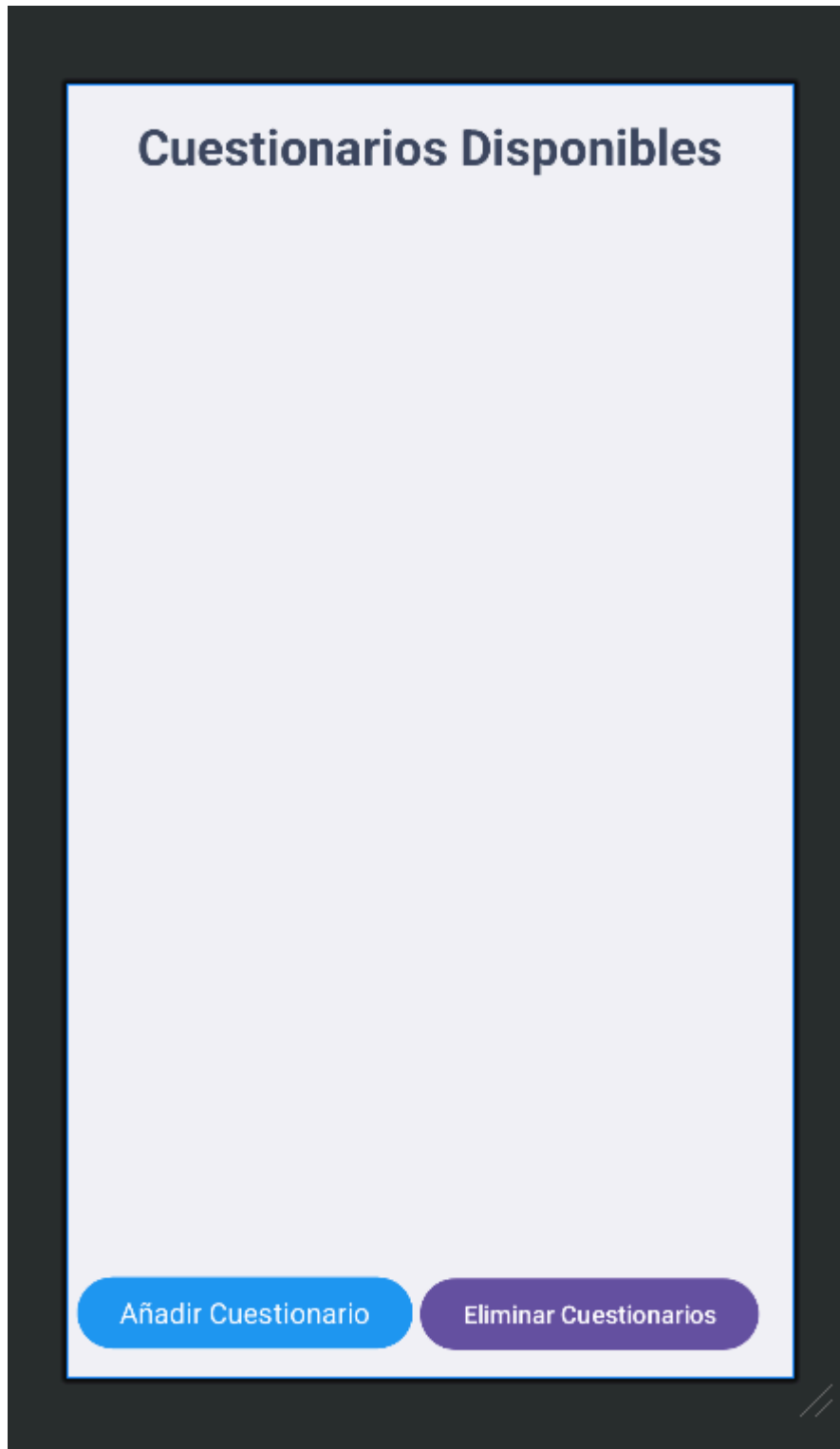
Inicio



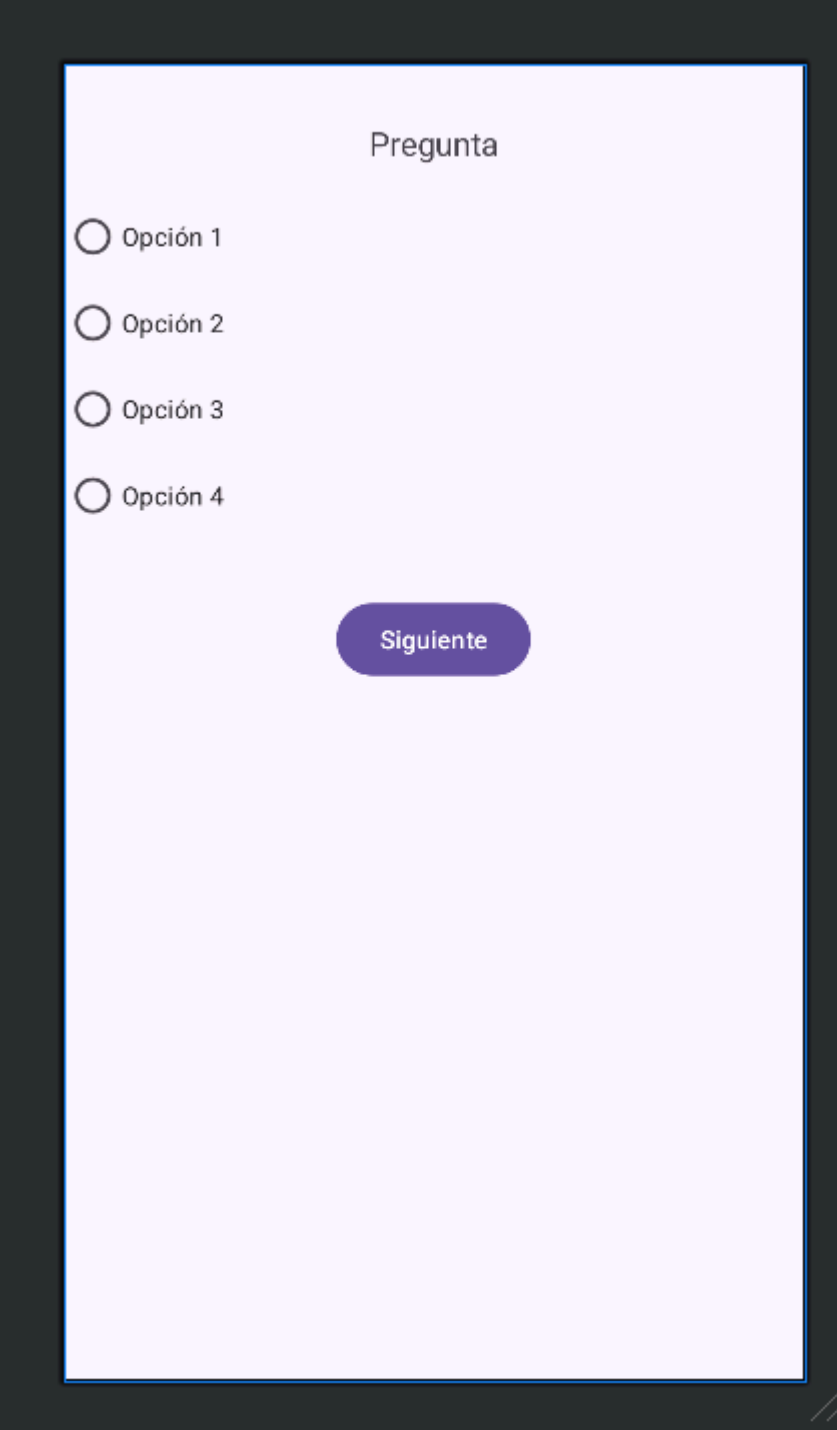
Perfil



The image shows a screenshot of an Android application interface. It features a dark grey header bar at the top. Below the header is a light purple rectangular area containing the text 'Perfil de Usuario'. Underneath this, there are two labels: 'Nombre de usuario:' and 'Puntuación:'. Each label is followed by a white rectangular input field. The entire interface is set against a dark background.

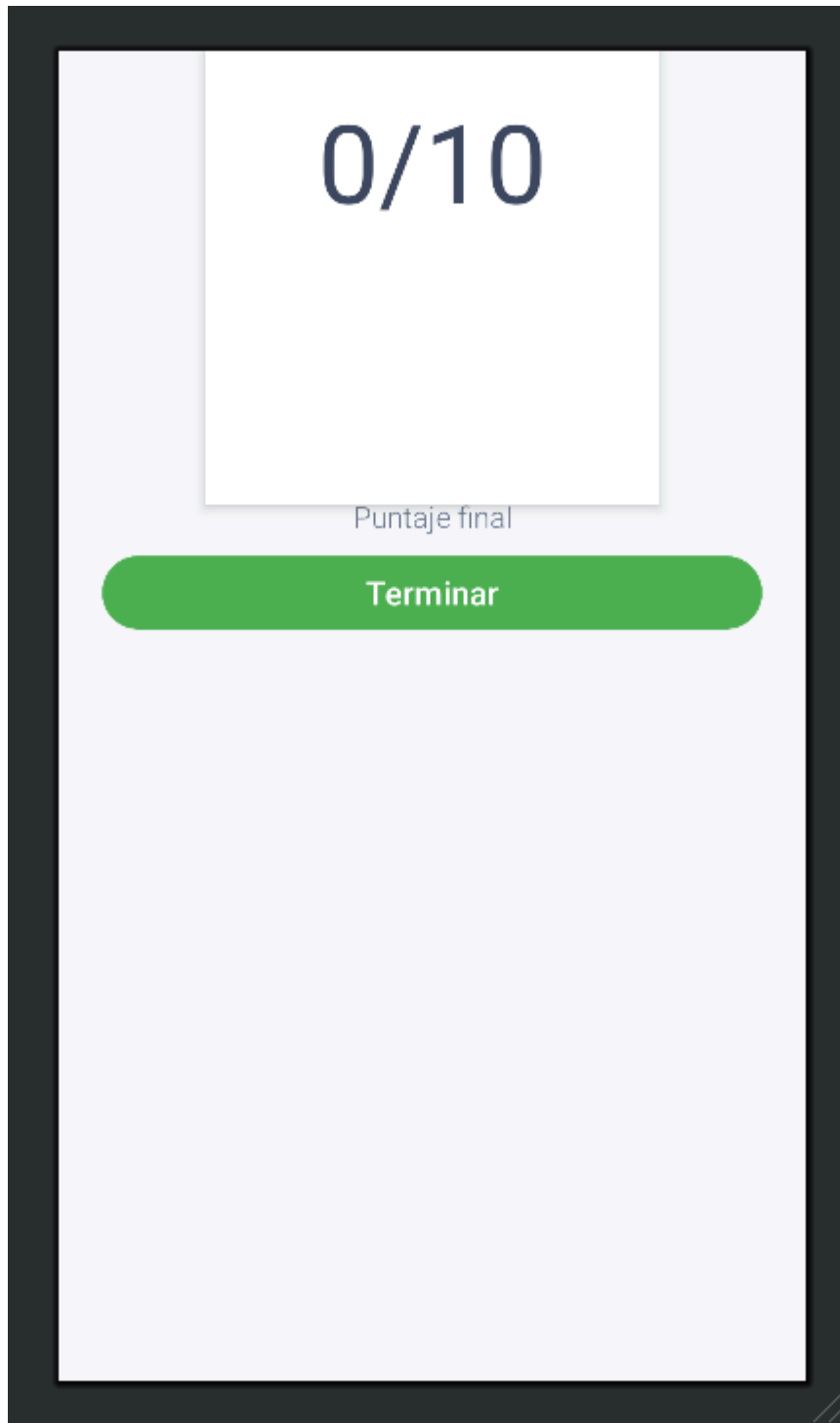


Preguntas



The image shows a screenshot of an Android application interface. It features a light purple rectangular area centered on a dark grey background. At the top of the purple area, the word "Pregunta" is written in a dark grey font. Below this, there are four radio button options, each consisting of a small circle followed by the text "Opción 1", "Opción 2", "Opción 3", and "Opción 4" respectively. At the bottom center of the purple area, there is a rounded rectangular button with a dark purple background and the word "Siguiente" in white text. The entire interface is framed by a dark grey border.

Resultado



[2ºDAM-B] PGL

CIFP César Manrique

Nombre: Ángel Steven Guerra Marrero

Proyecto Android

Enlace al video explicativo:

<https://drive.google.com/file/d/1jxOb3-YINbw0R92dw5XmgUUoTDLb4QV9/view?usp=sharing>