



UNIVERSIDAD AUTONOMA DE NUEVO LEON



Facultad de Ingeniería Mecánica y Eléctrica

Ingeniería de Dispositivos Móviles

“Manual Técnico Ideal Weight”

ING. Carlos Bernardo Garza Treviño

Nombres

Adolfo Heriberto Baez Otero #1643878

Víctor Manuel Martínez Tavernier #1719384

Jesús Eduardo Espinoza Walle #1400758

San Nicolás de los Garza N.L. a 7 de noviembre de 2017

Introducción:

La finalidad de este manual técnico es dar al lector algo de información acerca de cómo se ha desarrollado esta interfaz, durante la lectura de éste se verán herramientas y algunas partes de código, escritas por el programador; no pretende ser un curso de aprendizaje de ningún tipo, ni de las herramientas aquí mencionadas. Para un mayor detalle sobre las herramientas que se utilizaron, se recomienda que se busquen sus respectivos manuales.

Objetivo:

Dar una guía al lector del desarrollo que se llevó a cabo, a diferencia del manual del usuario, este no está enfocado a la usabilidad de la interfaz, solo al entendimiento general de dicha aplicación.

Requerimientos

Para poder crear en su mayoría la aplicación se requirió el software Android Studio, dicho programa se utilizó para desarrollar tanto la interfaz como la lógica del proyecto, se requirió de un conocimiento razonable en Java, diseño en XML y PHP para darle un estilo uniforme y legible a la interfaz de la aplicación y que así el usuario no se confundiera entre fondos mal diseñados, botones demasiado pequeños o demasiado grandes, contenido demasiado implícito que proporciona confusión y en ocasiones hace que el usuario no pueda utilizar bien la aplicación.

Contenido

Desarrollo

Para el desarrollo de esta interfaz se utilizó Android Studio (Figura 1), el software se puede obtener en el siguiente recurso

["https://developer.android.com/studio/index.html"](https://developer.android.com/studio/index.html).



Figura 1: Android Studio

La aplicación fue programada para la versión de Android 4.4 (KitKat) para que tuviera un amplio rango de disponibilidad en los dispositivos móviles con sistema operativo Android.

A screenshot of the AndroidManifest.xml file in Android Studio. The file is open in the "Text" editor, showing the XML code for the application manifest. The code includes the application tag with attributes like android:allowBackup, android:icon, android:label, and android:theme. It also lists several activities, including PrimeraPantalla, BackButton, SplashScreen, registro, NivelesActivity, Calculadora, ListContenido, ContenidoAlimentos, ActionPantalla, AlimentosDesayunoN1, AlimentosAlmuerzoN1, AlimentosComidaN1, AlimentosCenaN1, Comidas, and Cenas. The intent-filter for the main activity is also shown.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.idealweight2actualizado" >
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".PrimeraPantalla" />
        <activity android:name=".BackButton" />
        <activity android:name=".SplashActivity" />
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <activity>
            <activity android:name=".registro" />
            <activity android:name=".NivelesActivity" />
            <activity android:name=".Calculadora" />
            <activity android:name=".ListContenido" />
            <activity android:name=".ContenidoAlimentos" />
            <activity android:name=".ActionPantalla" />
            <activity android:name=".AlimentosDesayunoN1" />
            <activity android:name=".AlimentosAlmuerzoN1" />
            <activity android:name=".AlimentosComidaN1" />
            <activity android:name=".AlimentosCenaN1" />
            <activity android:name=".Comidas" />
            <activity android:name=".Cenas" />
        </activity>
    </application>
</manifest>
```

Este código es uno de los más importantes, ya que en este archivo llamado android manifest se declaran todas las actividades que se utilizaron para desarrollar las pantallas de la aplicación.

Otra parte muy importante del código es la parte del login.

```
19 public class PrimeraPantalla extends AppCompatActivity {
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_primera_pantalla);
25
26         //Login datos
27         final EditText etUsername = (EditText) findViewById(R.id.Username);
28         final EditText etPassword = (EditText) findViewById(R.id.Password);
29
30         //Referencia a los elementos
31         final Button bLogin = (Button) findViewById(R.id.bLogin);
32         final TextView tRegistrar = (TextView) findViewById(R.id.tRegistrar);
33
34         //Agregar su listener (Escuchador)
35         tRegistrar.setOnClickListener((v) -> {
36             Intent registerIntent = new Intent( packageContext: PrimeraPantalla.this, registro.class);
37             PrimeraPantalla.this.startActivity(registerIntent);
38             //finish();
39         });
40
41         bLogin.setOnClickListener((v) -> {
42
43             final String username = etUsername.getText().toString();
44             final String password = etPassword.getText().toString();
45
46             Response.Listener<String> responseListener = new Response.Listener<String>() {
47                 @Override
48                 public void onResponse(String response) {
49                     try {
```

```
                    JSONObject jsonResponse = new JSONObject(response);
                    boolean success = jsonResponse.getBoolean( name: "success");
                    if (success){
                        String name = jsonResponse.getString( name: "name");
                        Intent intent = new Intent( packageContext: PrimeraPantalla.this, ActionPantalla.class);
                        intent.putExtra( name: "name", name);
                        PrimeraPantalla.this.startActivity(intent);
                    }
                    else{
                        AlertDialog.Builder builder = new AlertDialog.Builder( context: PrimeraPantalla.this);
                        builder.setMessage( "Login Failed")
                                .setNegativeButton( text: "Retry", listener: null)
                                .create()
                                .show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });

        LoginRequest loginRequest = new LoginRequest(username, password, responseListener );
        RequestQueue queue = Volley.newRequestQueue( context: PrimeraPantalla.this);
        queue.add(loginRequest);
    }
}
```

En esta parte del Código se encuentra toda la lógica desarrollada para la pantalla de inicio de sesión en la aplicación

```
package com.example.adolfo.idealweight;

import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;
```

```
1 package com.example.adolfo.idealweight;
2
3 import android.app.AlertDialog;
4 import android.content.DialogInterface;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.Menu;
8 import android.content.Intent;
9 import android.view.MenuItem;
10 import android.widget.EditText;
11 import android.widget.Toast;
12
13 import android.os.Build;
14 import android.support.annotation.RequiresApi;
15 import android.text.format.Time;
16 import android.util.Log;
17 import android.view.View;
18 import android.widget.Button;
19 import android.widget.ImageButton;
20 import android.widget.TextView;
21
22 import java.util.Locale;
```

Estos son algunos de los recursos que se utilizaron, dichas librerías se usaron para poder utilizar complementos y funciones dentro del entorno de las pantallas de la aplicación.

Y una de las partes más complejas de desarrollar para la aplicación es la de pantalla de inicio una vez que se realiza el login.

```

public class ActionPantalla extends AppCompatActivity {
    //Crear las variables de los objetos que vamos a utilizar
    private ImageButton ib;

    @RequiresApi(api = Build.VERSION_CODES.N)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.action_pantalla);

        final TextView etwMsg = (TextView) findViewById(R.id.wMsg);

        Intent intent = getIntent();
        String name = intent.getStringExtra("name");
        String message = name + " Bienvenido";
        etwMsg.setText(message);

        ib = (ImageButton) findViewById(R.id.imagenBtn);

        //Pase al calendario
        TextView yLabel = (TextView) findViewById(R.id.yLabel);
        TextView mLabel = (TextView) findViewById(R.id.mLabel);
        TextView dLabel = (TextView) findViewById(R.id.dLabel);

        Time today = new Time(Time.getCurrentTimezone());
        today.setNow();

        int dia = today.monthDay;
        int mes = today.month;
        int afo = today.year;
        mes = mes+1;

        dLabel.setText(" " + dia);
        mLabel.setText("Mes: " + mes);
        yLabel.setText(" " + afo);

        //Agregar su listener(Escuchador)
        ib.setOnClickListener(new View.OnClickListener() {
            Intent pantalla1 = new Intent( packageContext: ActionPantalla.this, NivelesActivity.class);
            startActivity(pantalla1);
            //Finish();
        });

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.commonmenu, menu);
            return super.onCreateOptionsMenu(menu);
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            int id = item.getItemId();

            if (id==R.id.actCalculadora)
            {
                // TODO: Add action handling code here
            }
        }
    }
}

```

```

else if (id==R.id.actCerrar)
{
    //Toast.makeText(this, "Está cerrando sesión", Toast.LENGTH_SHORT).show();
    //startActivity(new Intent(this, Calculadora.class));
    final AlertDialog.Builder builder= new AlertDialog.Builder( context: ActionPantalla.this);
    builder.setMessage("¿Deseas Cerrar Sesión?");
    // builder.setTitle("Ideal Weight");
    builder.setCancelable(true);
    builder.setPositiveButton( text: "Si", (dialog, i) -> { finish(); });
    builder.setNegativeButton( text: "No", (dialog, i) -> {
        dialog.cancel();
    });

    AlertDialog dialog = builder.create();
    dialog.show();
}

else if (id==R.id.actSalir)
{
    // Toast.makeText(this, "Está saliendo de la aplicación", Toast.LENGTH_SHORT).show();
    //startActivity(new Intent(this, Calculadora.class));
    final AlertDialog.Builder builder= new AlertDialog.Builder( context: ActionPantalla.this);
    builder.setMessage("¿Desea salir de la Aplicación?");
    // builder.setTitle("Ideal Weight");
    builder.setCancelable(true);
    builder.setPositiveButton( text: "Si", (dialog, i) -> { finish(); });
    builder.setNegativeButton( text: "No", (dialog, i) -> {
        dialog.cancel();
    });
}
}

```

En esta parte del Código se encuentra toda la lógica desarrollada para la pantalla principal una vez que se haya iniciado sesión en la aplicación

Para realizar la base de datos se utilizó un servidor gratuito llamado almacenado en la siguiente página “<https://mex.000webhost.com/>” llamada 000webhost.



Una de las tablas más importantes es la del usuario, en donde se almacenaban diversos datos del usuario registrado en la aplicación.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	name	varchar(16)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index More
3	username	varchar(16)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index More
4	age	tinyint(4)			No	None			Change Drop Primary Unique Index More
5	password	varchar(16)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index More
6	height	int(11)			No	None			Change Drop Primary Unique Index More
7	weight	int(11)			No	None			Change Drop Primary Unique Index More

En esta tabla se almacenaban datos importantes del usuario que se registra en la aplicación como lo son un nombre, nombre de usuario, una contraseña, edad, peso y estatura.