

PRACTICA 07.

OBJETIVO GENERAL.

En esta práctica, mediante una validación llevaremos al usuario a otra actividad dando la bienvenida al usuario que se logueo con credenciales correctas, estos datos serán pasados de la actividad inicial a la actividad de bienvenida, como se ilustra a continuación:



OBJETIVOS ESPECÍFICOS:

- Utilizar LinearLayout como contenedor principal de la aplicación.
- Crear objetos Actividad Origen:
 - Un TextView : Utilizado para desplegar mensaje
 - Dos EditText : Para introducir un nombre de usuario y password.
 - Un Button : Utilizado para verificar los datos capturados por el usuario.
- Crear un evento que verifique cada uno de los datos capturado según su dominio y si son correctos, que despliegue la segunda actividad dando la bienvenida.
- Crear objetos Actividad Destino:
 - Ocho TextView : Utilizado para desplegar mensajes y datos
 - Un Button : Utilizado para regresar a la activad de origen.

CREACION DE LA INTERFACE GRAFICA.

Usaremos el código siguiente para la interface gráfica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="20dp">

    <TextView
        android:id="@+id/txtMensaje"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="40dp"
        android:layout_marginTop="40dp"
        android:text="Login al Sistema"
        android:textAlignment="center"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/edtUsuario"
        android:hint="Usuario Id"
        android:textSize="30sp"
        android:layout_margin="10dp"
        android:padding="5dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/edtPassword"
        android:inputType="textPassword"
        android:hint="Password"
        android:textSize="30sp"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnEntrar"
        android:text="Entrar al Sistema"
        android:textSize="30sp"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

DESARROLLO DE LA LOGICA DE APLICACIÓN.

El código inicial de nuestra actividad principal se verá:

```
package com.example.delio.app01a;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Es aquí donde iremos agregando líneas de código para darle funcionalidad a nuestra aplicación.

DECLARACION DE VARIABLES

En primer lugar debemos de crear variables que serán usadas para relacionar la parte grafica con la lógica de la aplicación, en este caso, son todos los elementos que tendrán acción al funcionar la aplicación.

```
private EditText edtUsuario, edtPassword;
private Button btnEntrar;
```

Después de la declaración de variables, DEBEMOS de relacionar los elementos de la interface gráfica con nuestras variables de nuestra lógica de aplicación.

```
edtUsuario = (EditText) findViewById(R.id.edtUsuario);
edtPassword = (EditText) findViewById(R.id.edtPassword);
```

Vamos a crear la lógica para el botón al evento de tocar el botón:

```
btnEntrar = (Button) findViewById(R.id.btnEntrar);
btnEntrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        validarDatos();
    }
});
```

En este caso, dentro del evento click del botón, invocaremos un método que realizará la validación de los datos, en este caso es:

```
validarDatos();
```

A continuación esta la implementación, esta implementación prepara una nueva actividad mediante la clase Intent:

```
private void validarDatos(){
    String user = edtUsuario.getText().toString();
    String pass = edtPassword.getText().toString();

    if(TextUtils.isEmpty(user) || TextUtils.isEmpty(pass)){
        mensajeUsuario("Los campos son obligatorios");
    } else if (user.equals("delio") && pass.equals("123456")){
        String nombre = "Delio";
        String apellidos = "Coss Camilo";
        String telefono = "2292-163520";
        String correo = "delioc@itver.edu.mx";

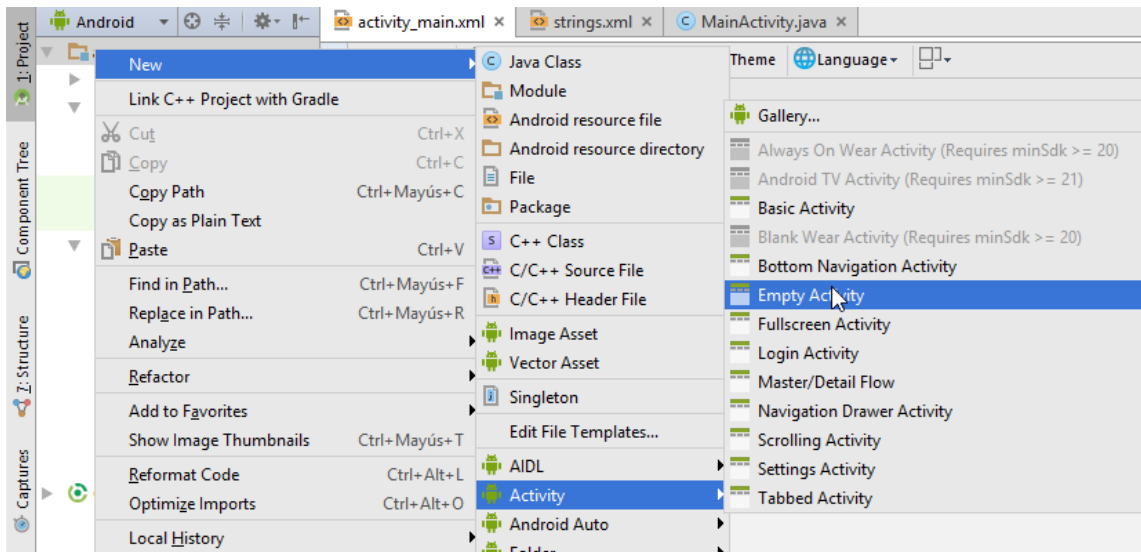
        Intent AboutUs = new Intent(this, AboutUsActivity.class);
        AboutUs.putExtra("user", user);
        AboutUs.putExtra("nombre", nombre);
        AboutUs.putExtra("apellidos", apellidos);
        AboutUs.putExtra("telefono", telefono);
        AboutUs.putExtra("correo", correo);
        startActivity(AboutUs);
    } else {
        mensajeUsuario("Credenciales no validas");
    }
}

private void mensajeUsuario(String mensaje){
    Toast.makeText(getApplicationContext(),mensaje,
    Toast.LENGTH_LONG).show();
}
```

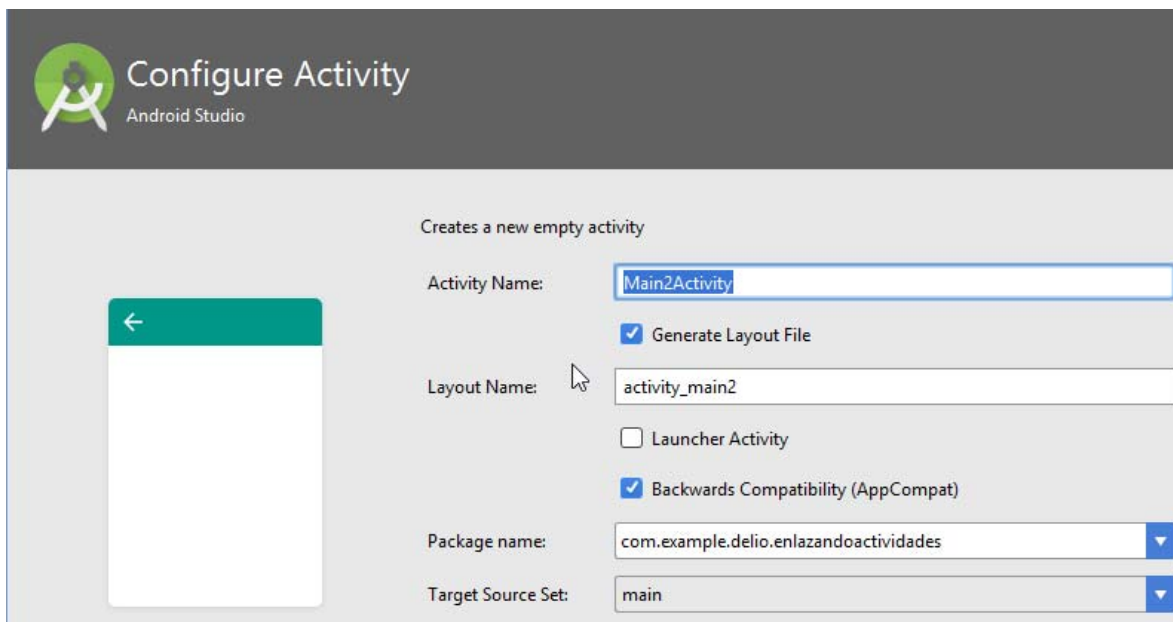
Cuando demos click en el botón, se ejecutara una acción que mostrara una segunda actividad en la cual mostrara datos generales del usuario de la aplicación siempre y cuando los datos estén correctos, esto simula un acceso a una base de datos y carga de los datos del usuario en cuestión.

Vamos a crear la segunda actividad realizando lo siguiente:

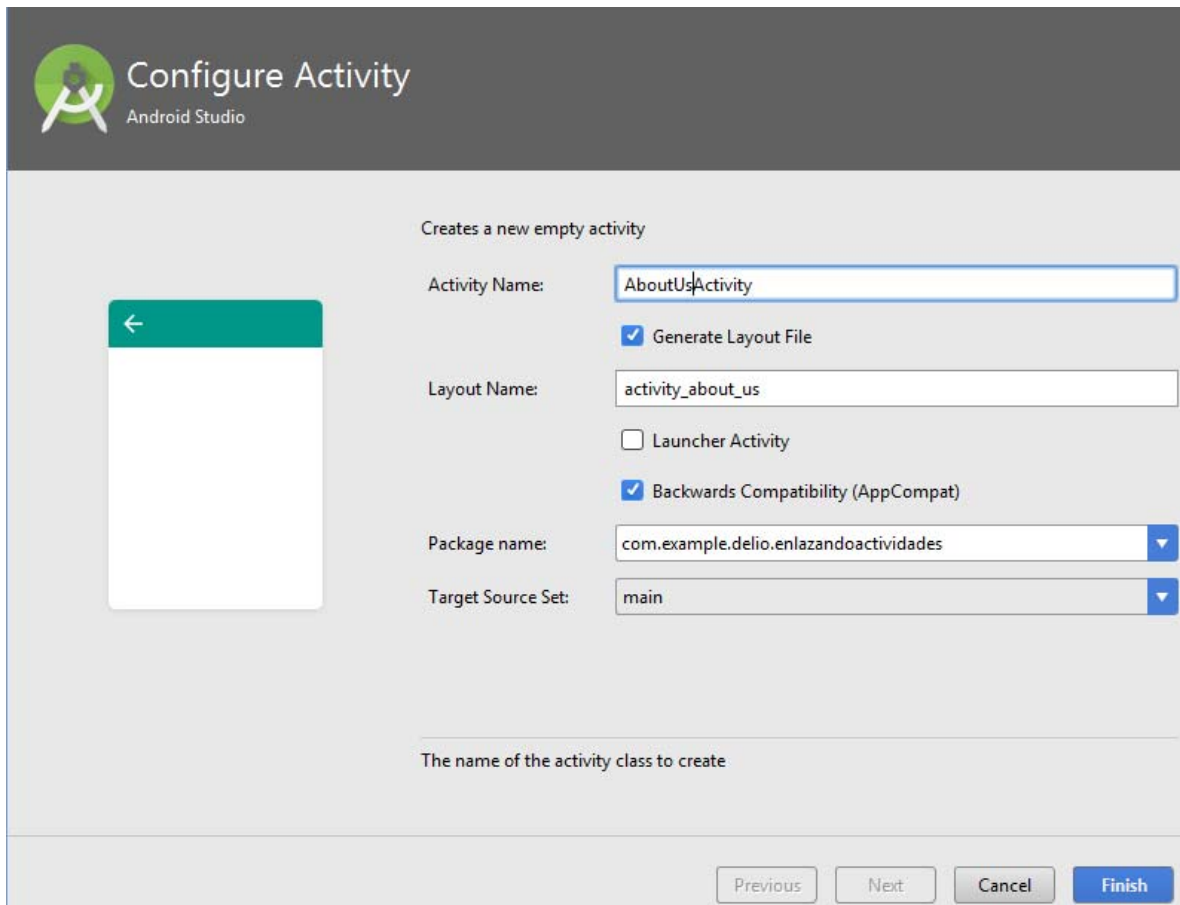
Daremos un click derecho en la opción app, seleccionamos New, Activity y finalmente Empty Activity, como se indica en la figura.



Desplegará la siguiente ventana de configuración:



Cambiaremos el nombre de la actividad Main2Activity por AboutUsActivity



Configure Activity
Android Studio

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

☒ Backwards Compatibility (AppCompat)

Package name:

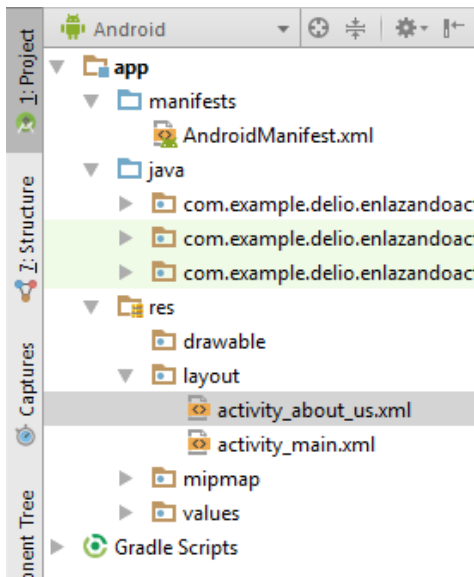
Target Source Set:

The name of the activity class to create

Finalmente damos click en finish.

NOTA: Trate de usar la notación CAMEL al momento de nombrar las actividades.

Vea en la ventana Project el árbol de componentes, deberá de tener ya dos actividades, la principal y la última que creamos:



Ahora que hemos creado la parte gráfica, debemos que crear la lógica que le dará funcionamiento a la aplicación.

En primer lugar debemos de crear la lógica para la actividad principal para que sea capaz de invocar a la segunda actividad, esto lo hacemos mediante la clase Intent, como se ilustra:

```
43 | | | Intent AboutUs = new Intent(this, AboutUsActivity.class);
```

Tenga en cuenta que el constructor recibe dos parámetros, la actividad de origen y la actividad destino.

Posteriormente, para pasar datos usaremos el método putExtra para pasar parámetros (si es que hay datos de la actividad que se usaran en la actividad destino).

```
44 | | | AboutUs.putExtra("user", user);
45 | | | AboutUs.putExtra("nombre", nombre);
46 | | | AboutUs.putExtra("apellidos", apellidos);
47 | | | AboutUs.putExtra("telefono", telefono);
48 | | | AboutUs.putExtra("correo", correo);
```

Y finalmente invocamos a la actividad destino

```
49 | | | startActivity(AboutUs);
```

Esto nos llevará a la actividad destino, con esto terminamos la lógica de la actividad de origen.

CREACION DE LA INTERFACE GRAFICA.

Usaremos el código siguiente para la interface gráfica.

```

<?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:orientation="vertical"
    android:layout_width="match_parent"
    android:padding="16dp"
    android:layout_height="match_parent"
    tools:context="com.example.delio.app07.MainActivity">

    <TextView
        android:id="@+id/txtBienvenida"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_marginTop="20dp"
        android:text="Bienvenido al Sistema"
        android:textAlignment="center"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/txtNombre"
        android:textSize="24sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/txtApellidos"
        android:textSize="24sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/txtDatosCredencial"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_marginTop="20dp"
        android:text="Su Credencial Id es"
        android:textAlignment="center"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/txtUser"
        android:textSize="24sp"
        android:layout_marginTop="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/txtDatosContacto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"

```



```

        android:layout_marginTop="20dp"
        android:text="Sus Datos de Contacto"
        android:textAlignment="center"
        android:textSize="30sp" />

<TextView
    android:id="@+id/txtCorreo"
    android:textSize="24sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<TextView
    android:id="@+id/txtTelefono"
    android:textSize="24sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/btnRegresar"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="16dp"
    android:text="Regresar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>

```

DESARROLLO DE LA LOGICA DE APLICACIÓN.

El código inicial de nuestra actividad principal se verá:

```

private TextView txtUser, txtNombre, txtApellidos, txtCorreo,
txtTelefono;
private Button btnRegresar;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_about_us);

    txtUser      = (TextView) findViewById(R.id.txtUser);
    txtNombre    = (TextView) findViewById(R.id.txtNombre);
    txtApellidos = (TextView) findViewById(R.id.txtApellidos);
    txtCorreo    = (TextView) findViewById(R.id.txtCorreo);
    txtTelefono  = (TextView) findViewById(R.id.txtTelefono);
    btnRegresar  = (Button)   findViewById(R.id.btnRegresar);

    Bundle bundle = getIntent().getExtras();

    txtUser.setText(bundle.getString("user").toString());
    txtNombre.setText(bundle.getString("nombre").toString());
    txtApellidos.setText(bundle.getString("apellidos").toString());
    txtCorreo.setText(bundle.getString("correo").toString());
    txtTelefono.setText(bundle.getString("telefono").toString());

    btnRegresar.setOnClickListener(new View.OnClickListener() {

```

Programación de Dispositivos Móviles.

Delio Coss Camilo.

```

        @Override
        public void onClick(View view) {
            finish();
        }
    });
}

```

Analicemos el código principal de la actividad destino, que es la de obtener los datos que vienen como parámetros de la actividad de origen.

En primer lugar usamos la clase Bundle como se indica:

```

25      Bundle bundle = getIntent().getExtras();

```

En la instancia generada en este caso bundle, es un objeto que contiene un arreglo asociativo con la llave y su valor de cada dato enviado por la actividad de origen. Mediante el siguiente código, podemos acceder a cada uno de los valores:

```

27      txtUser.setText(bundle.getString("user").toString());
28      txtNombre.setText(bundle.getString("nombre").toString());
29      txtApellidos.setText(bundle.getString("apellidos").toString());
30      txtCorreo.setText(bundle.getString("correo").toString());
31      txtTelefono.setText(bundle.getString("telefono").toString());

```

Finalmente programamos el evento click del botón y terminamos la actividad y regresamos a la actividad de origen.

```

33      btnRegresar.setOnClickListener(new View.OnClickListener() {
34          @Override
35          public void onClick(View view) {
36              finish();
37          }
38      });

```

Con esto tenemos una interacción entre dos actividades con paso de datos como parámetros.