PROFE ESTUVE INVESTIGANDO DEMASIADOS CODIGOS PARA TRATAR DE ENTENDER Y ESTE FUE EL MAS PARECIDO A LO QUE ME LOGRÉ ENTENDER.

Se conoce como **interfaz de usuario** al medio que permite a una persona comunicarse con una [**máquina**](https://definicion.de/maquina/). La interfaz, en este caso, está compuesta por los puntos de contacto entre un usuario y el equipo. Además del mencionado ejemplo del mouse, otra interfaz de este tipo es la pantalla del [**monitor**](https://definicion.de/monitor/) o el [**teclado**](https://definicion.de/teclado).

CLASS

Cada clase es un modelo que define un conjunto de variables -el estado, y métodos apropiados para operar con dichos datos -el comportamiento. Cada objeto creado a partir de la clase se denomina instancia de la clase. Las clases de objetos son un pilar fundamental de la programación orientada a objetos.

ACTIVITY

Podemos decir que todas las pantallas de una aplicación son una “activity”. Más adelante vamos a ver que existen algunas variaciones, pero por ahora digamos que todas lo son. Es decir, que si una aplicación tiene cinco pantallas, tiene 5 “Actividades” o *activities*.

Las activities están conformadas por dos partes: la parte lógica y la parte gráfica.

La parte lógica es una archivo .java que es la clase que se crea para poder manipular, interactuar y colocar el código de esa actividad.

La parte gráfica es un XML que tiene todos los elementos que estamos viendo de una pantalla declarados con etiquetas parecidas a las del HTML, es decir, que el diseño de una aplicación en Android se hace similar a una página web; XML es un primo de HTML.

Resumiendo, una actividad está conformada por la parte lógica (un archivo Java) y la parte gráfica (un archivo XML).

Adentrando más en el tema, ya sabemos que tenemos un archivo .java, esto quiere decir que tenemos una clase principal, al ser una actividad extiende de la clase Activity (por eso el nombre) que nos proporciona Android para crear actividades con sus métodos asignados.

Veamos una actividad básica:   
Este sería nuestro archivo "ManActivity" como el del ejercicio del "HolaMundo".

package com.example.holamundodesarrollowebcom;

import android.os.Bundle;

import android.app.Activity;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

Expliquemos por líneas:

package com.example.holamundodesarrollowebcom;

Esta línea únicamente indica el nombre del paquete en donde se encuentra nuestra clase.

import android.os.Bundle;

import android.app.Activity;

Los dos *imports*, son la forma de decir que necesitamos esos archivos para trabajar dentro de la clase, los que ya nos da Android para no tener que escribir las cosas desde cero.

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

En esta última sección de código lo que estamos haciendo es crear una clase que se llama "MainActivity" y la estamos extendiendo de acvitity, en español esto es el concepto de herencia de la famosa programación orientada a objetos, estamos diciendo que "mainactivity" es una clase que hereda las cosas de la clase Activity que ya tiene Android definida.

Todas las *activities* deben llevar por lo menos un método, el método "oncreate", que es en donde se crea la actividad o podemos decir que es donde se le da vida.

Del método "onCreate" lo más importante es la línea de código: SetContentView(R.Layout.acivity\_main).

Que es la que hace el trabajo de enlazar la parte lógica con la parte gráfica. El archivo XML que va a mostrarse cuando se mande a llamar la clase "MainActivity" es el archivo XML llamado "activity\_main".

Para cerrar la explicación: si yo creo una actividad nueva y la llamo "VentanaPrinicipal", debo hacer que herede de activity si quiero que funcione como actividad y para decirle que el archivo XML que va a mostrar sea el "ventanaprincipal.xml" o "pepito.xml". La línea que dice "SetContentView" debe llevar dentro algo parecido a esto: "setContentView"(R.layout.ventanaprincipal).

Se va a desarrollar una aplicación móvil

Aquí se cuenta con los REST y los Métodos que se van a ocupar o bueno el que recibe datos POST o GET manejan Json, reciben o envían datos por Json y se separan como quiera

Se hace un registro y se dará lo que necesita el RESt lo regresa y se pinta hará una consulta para consumir los rest.

**<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="**[**http://schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**" android:orientation="vertical"  
 android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:padding="20dip">   
   
 <EditText android:id="@+id/txtEmail" android:layout\_height="wrap\_content"  
 android:layout\_width="fill\_parent" android:hint="Email" android:inputType="textEmailAddress" />  
   
 <EditText android:id="@+id/txtPass" android:layout\_height="wrap\_content"   
 android:layout\_width="fill\_parent" android:hint="Password" android:inputType="textPassword"  
 android:layout\_marginTop="15dip" android:layout\_marginBottom="30dip" />  
   
   
 <Button android:id="@+id/btnLogin" android:layout\_height="wrap\_content"   
 android:layout\_width="wrap\_content" android:text="enviar"  
 android:layout\_gravity="center\_horizontal" android:paddingLeft="15dip" android:paddingRight="15dip" />  
   
 <TextView android:id="@+id/link\_to\_register"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dip"  
 android:layout\_marginBottom="40dip"  
 android:text="registrate!"  
 android:gravity="center"  
 android:textSize="20dip"  
 android:textColor="#0b84aa"/>  
   
 <TextView android:id="@+id/login\_error"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dip"  
 android:layout\_marginBottom="40dip"  
 android:gravity="center"  
 android:textSize="20dip"  
 android:textColor="#0b84aa"/>  
</LinearLayout>**

La siguiente pantalla será la de registro bastante parecida a la anterior. Aquí a mayores pedimos un nombre de usuario para poder utilizar en la aplicación en lugar del email.

**<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="**[**http://schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**"  
 android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:padding="20dip" android:orientation="vertical">  
   
 <EditText android:id="@+id/txtUserName" android:layout\_height="wrap\_content" android:layout\_width="fill\_parent"   
 android:hint="UserName" android:inputType="text" />  
   
 <EditText android:id="@+id/txtEmail" android:layout\_marginTop="15dip" android:layout\_height="wrap\_content"  
 android:layout\_width="fill\_parent" android:hint="Email" android:inputType="textEmailAddress" />  
   
 <EditText android:id="@+id/txtPass" android:layout\_height="wrap\_content"   
 android:layout\_width="fill\_parent" android:hint="Password" android:inputType="textPassword"  
 android:layout\_marginTop="15dip" android:layout\_marginBottom="30dip" />  
   
   
 <Button android:id="@+id/btnRegister" android:layout\_height="wrap\_content"   
 android:layout\_width="wrap\_content" android:text="enviar"  
 android:layout\_gravity="center\_horizontal" android:paddingLeft="15dip" android:paddingRight="15dip" />  
   
 <TextView android:id="@+id/link\_to\_login"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dip"  
 android:layout\_marginBottom="40dip"  
 android:text="Ya tienes cuenta! Accede!"  
 android:gravity="center"  
 android:textSize="20dip"  
 android:textColor="#0b84aa"/>  
   
 <TextView android:id="@+id/register\_error"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dip"  
 android:layout\_marginBottom="40dip"  
   
 android:gravity="center"  
 android:textSize="20dip"  
 android:textColor="#0b84aa"/>  
   
   
</LinearLayout>**

Ahora crearemos las dos activities para los layouts que acabamos de crear, no os olvideis de declararlos también en el archivo AndroidManifest.xml. Veamos primero que hace la Activity de Login.

**public class Login extends Activity {  
  
 private TextView lblGotoRegister;  
 private Button btnLogin;  
 private EditText inputEmail;  
 private EditText inputPassword;  
 private TextView loginErrorMsg;  
   
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.login);  
   
 inputEmail = (EditText) findViewById(R.id.txtEmail);  
 inputPassword = (EditText) findViewById(R.id.txtPass);  
 btnLogin = (Button) findViewById(R.id.btnLogin);  
 loginErrorMsg = (TextView) findViewById(R.id.login\_error);  
   
 btnLogin.setOnClickListener(new View.OnClickListener() {  
   
 public void onClick(View view) {  
 String email = inputEmail.getText().toString();  
 String password = inputPassword.getText().toString();  
   
 Usuario usuario = new Usuario();  
   
 usuario.setOnLoginUsuario(new OnLoginUsuario() {  
 @Override  
 public void onLoginWrong(String msg) {loginErrorMsg.setText(msg);}  
 @Override  
 public void onLoginCorrect(JSONObject json, String msg) {  
 loginErrorMsg.setText("");  
 Intent itemintent = new Intent(Login.this, ActivityPrincipal.class);  
 Login.this.startActivity(itemintent);  
 }  
 });   
 usuario.login(Login.this, email, password);  
 }  
 });  
   
 lblGotoRegister = (TextView) findViewById(R.id.link\_to\_register);  
 lblGotoRegister.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {Intent itemintent = new Intent(Login.this, Register.class);  
 Login.this.startActivity(itemintent);}  
 });  
 }  
   
}**

Como veis el código es muy sencillo, declaramos e inicializamos los elementos. El botón de login se encarga de inicializar una clase Usuario que veremos más adelante, por lo pronto vemos que tiene un método login al que le pasamos el contexto de la activity, un email y una contraseña que se encargará de enviar al servidor para validar.

En el listener onLoginUsuario podremos saber si el login es correcto o erróneo o si se ha producido algún tipo de error. En el caso de ser un login correcto enviamos al usuario a la activity de la pantalla principal.

Por último tenemos otro botón que envía al usuario a la pantalla de registro por si este no se ha creado un usuario todavía. Vamos a ver como sería el código de esta pantalla.

**public class Register extends Activity {  
  
 private TextView lblGotoLogin;  
 private Button btnRegister;  
 private EditText inputFullName;  
 private EditText inputEmail;  
 private EditText inputPassword;  
 private TextView registerErrorMsg;   
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.register);   
  
 inputFullName = (EditText) findViewById(R.id.txtUserName);  
 inputEmail = (EditText) findViewById(R.id.txtEmail);  
 inputPassword = (EditText) findViewById(R.id.txtPass);  
 btnRegister = (Button) findViewById(R.id.btnRegister);  
  
 registerErrorMsg = (TextView) findViewById(R.id.register\_error);  
  
 btnRegister.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View view) {  
 String name = inputFullName.getText().toString();  
 String email = inputEmail.getText().toString();  
 String password = inputPassword.getText().toString();  
 Usuario usuario = new Usuario();  
   
 usuario.setOnRegisterUsuario(new OnRegisterUsuario() {   
 @Override  
 public void onRegisterFinish(JSONObject json, String msg) {  
 registerErrorMsg.setText("");  
 Intent itemintent = new Intent(Login.this, ActivityPrincipal.class);  
 Register.this.startActivity(itemintent);>  
}   
 @Override  
 public void onRegisterFail(String msg) {registerErrorMsg.setText(msg);}   
 @Override  
 public void onRegisterException(Exception e, String msg) {registerErrorMsg.setText(msg);}  
 });   
 usuario.register(Register.this, name, email, password);  
 }  
 });  
   
 lblGotoLogin = (TextView) findViewById(R.id.link\_to\_login);  
 lblGotoLogin.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {Intent itemintent = new Intent(Register.this, Login.class);  
 Register.this.startActivity(itemintent);}  
 });  
   
 }  
   
}**

Como en la pantalla de Login, volvemos a inicializar todo, y seguimos utilizando una clase Usuario. En este caso hacemos uso de un método llamado register al que pasamos el contexto de la activity, un nombre, email y contraseña. Como antes en un listener llamado onRegisterUsuario nos dirá si todo ha ido bien o si ha fallado algo. En el caso de ir todo bien la clase Usuario se encarga de hacer el login y acceder a la pantalla principal de la aplicación.