

12 - Lanzar un segundo "Activity" y pasar parámetros

[Listado completo de tutoriales](#)



CURREN
LIQUIDACIÓN TOTAL
-90%
EN TODOS LOS RELOJES

→ Ver Fotos

Hemos visto en el concepto anterior que un programa puede tener más de una ventana representando cada ventana con una clase que hereda de AppCompatActivity.

Una situación muy común es que la primer ventana necesite enviar datos a la segunda para que a partir de estos proceda a efectuar una acción.

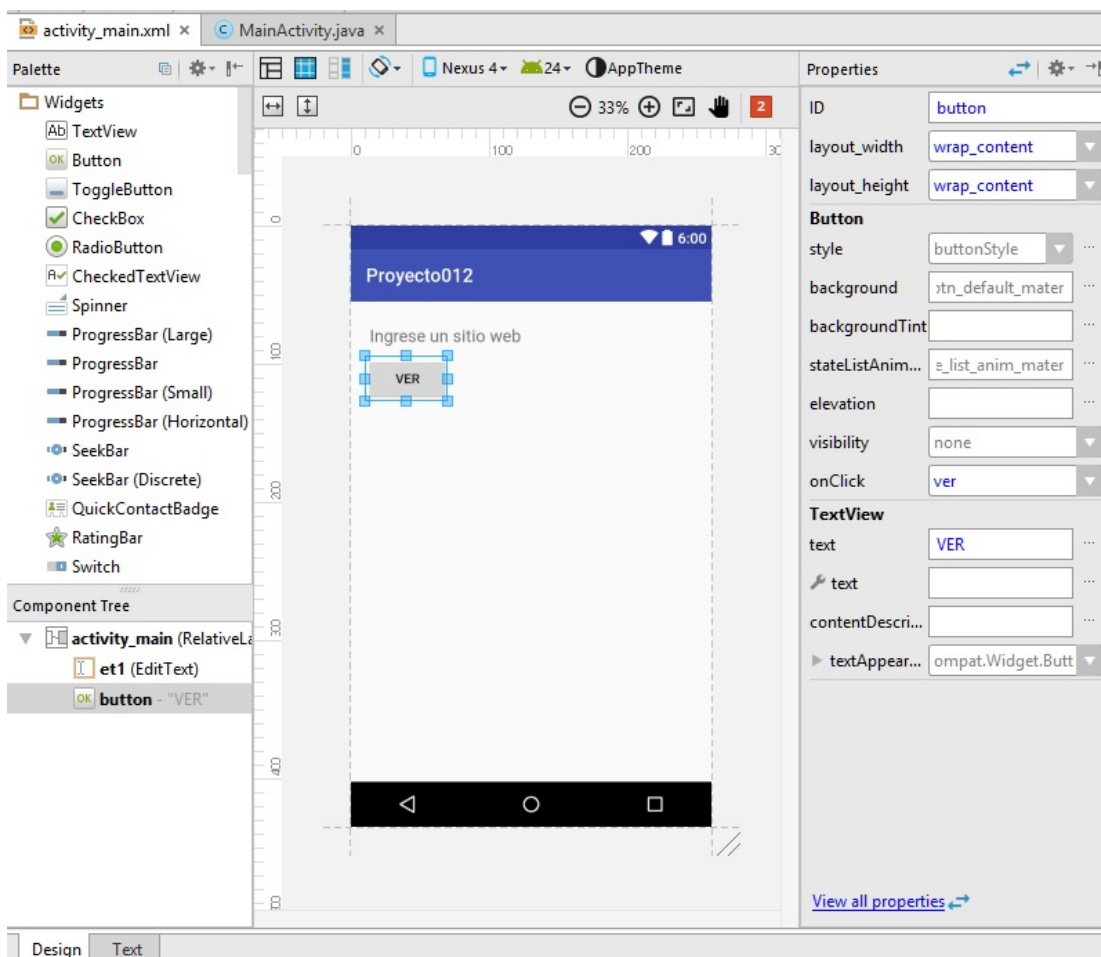
Problema:

Confeccionar un programa que solicite el ingreso de una dirección de un sitio web y seguidamente abrir una segunda ventana que muestre dicha página.

Para resolver este problema utilizaremos el control visual WebView que nos permite mostrar el contenido de un sitio web.

Crearemos un nuevo proyecto llamado proyecto012.

1 - Nuestro primer Activity tendrá la siguiente interfaz visual (ver controles):



Tenemos un control de tipo EditText (ID = et1, hint = "Ingrese un sitio web") y otro de tipo Button (inicializar la propiedad onClick con el nombre de método llamado "ver").

El código fuente de esta Activity es:

```
package com.tutorialesprogramacionya.proyecto012;

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

public class MainActivity extends AppCompatActivity {
    private EditText et1;

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

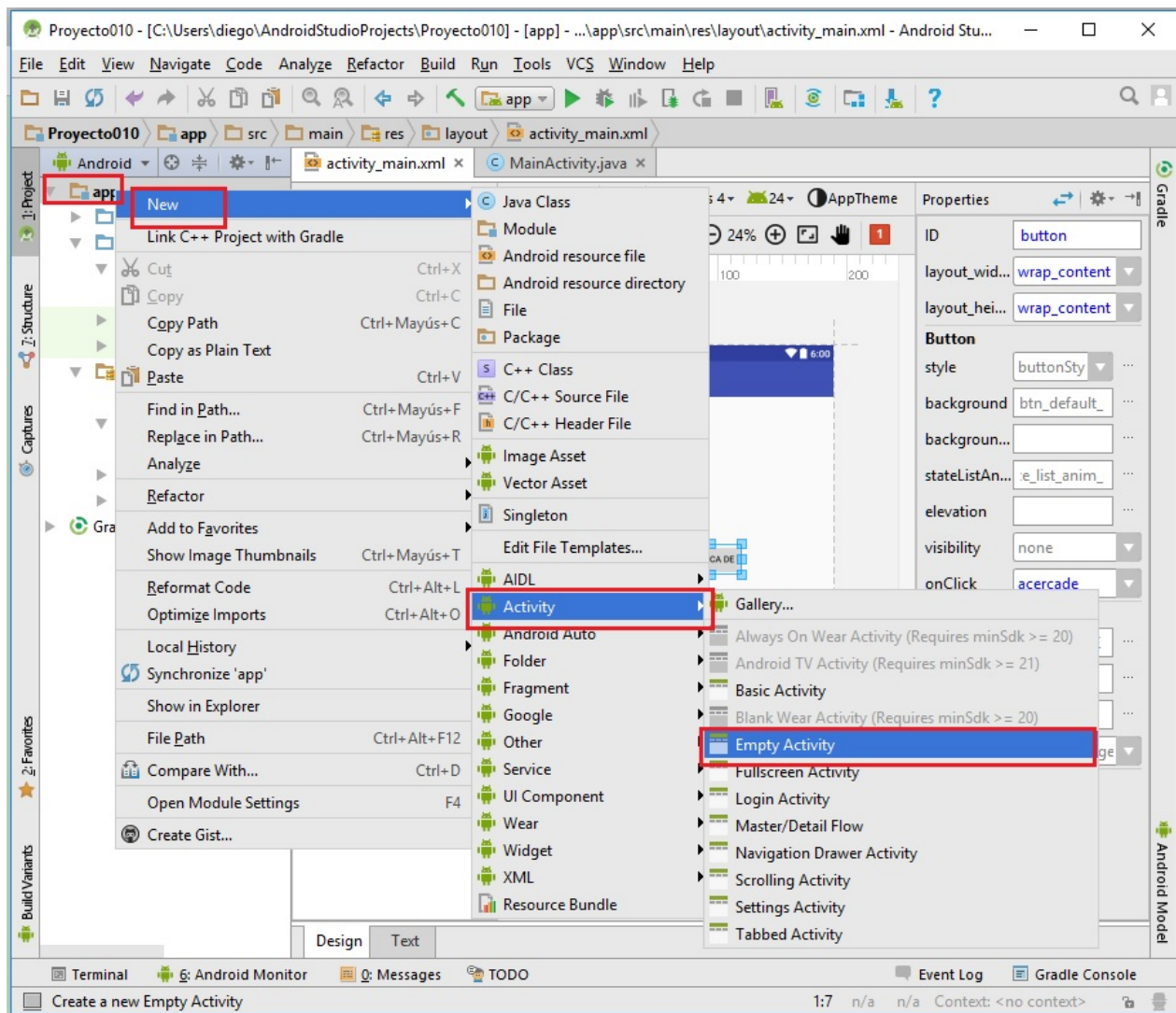
        et1=(EditText)findViewById(R.id.et1);
    }

    public void ver (View v) {
        Intent i=new Intent(this,Actividad2.class);
        i.putExtra("direccion", et1.getText().toString());
        startActivity(i);
    }
}
```

Como podemos ver la diferencia con el concepto anterior es que llamamos al método putExtra de la clase Intent. Tiene dos parámetros de tipo String, en el primero indicamos el nombre del dato y en el segundo el valor del dato:

```
public void ver(View v) {
    Intent i=new Intent(this,Actividad2.class);
    i.putExtra("direccion", et1.getText().toString());
    startActivity(i);
}
```

La segunda interfaz visual (recordemos que debemos presionar el botón derecho en la ventana Project (sobre app) y seleccionar la opción New -> Activity -> Empty Activity):



En el diálogo que aparece especificamos el nombre de nuestra segunda Activity y lo llamaremos Actividad2:

New Android Activity

Configure Activity

Android Studio

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

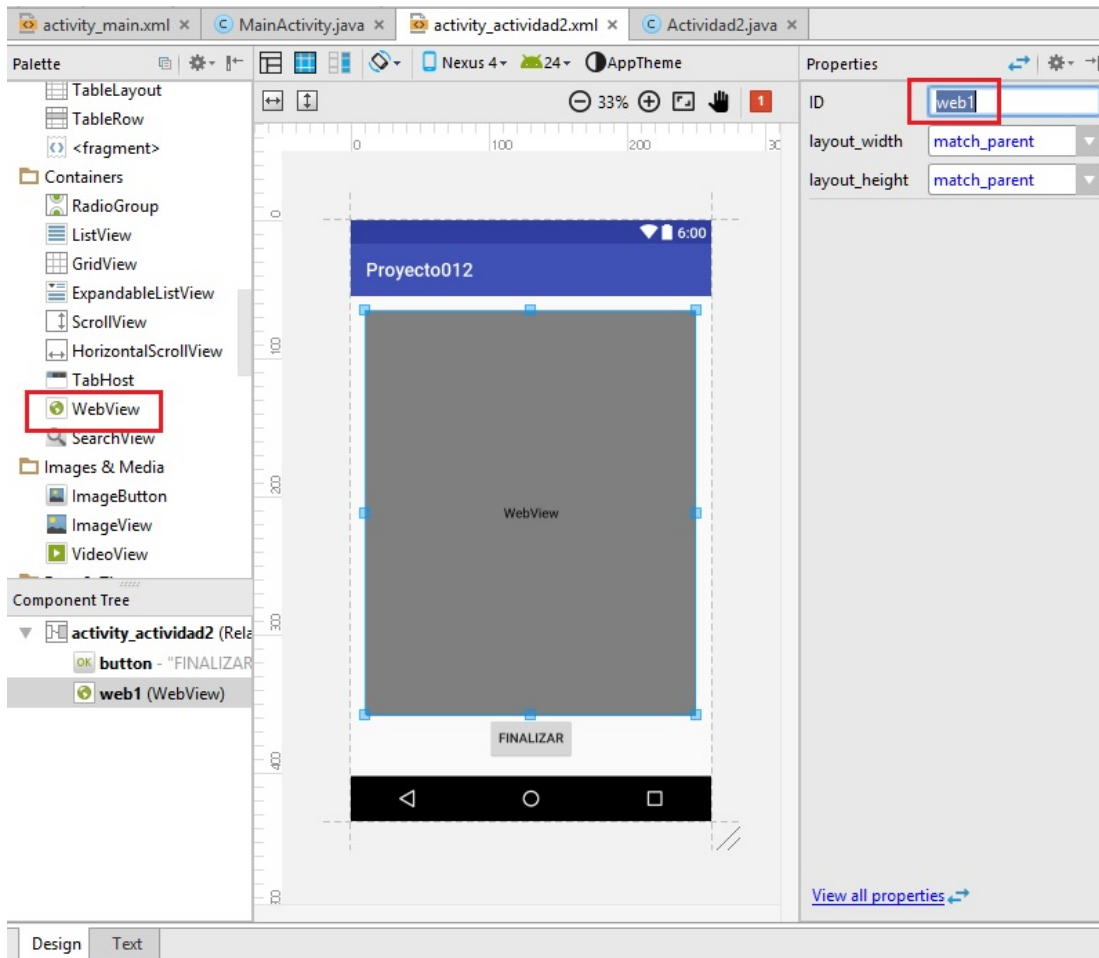
☒ Backwards Compatibility (AppCompat)

Package name:

Target Source Set:

The name of the activity class to create

Disponemos un objeto de la clase WebView que se encuentra en la pestaña "Containers" y un Button:



Codificamos la funcionalidad de la segunda actividad

```
package com.tutorialesprogramacionya.proyecto012;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;

public class Actividad2 extends AppCompatActivity {
    WebView web1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_actividad2);

        web1=(WebView) findViewById(R.id.web1);

        Bundle bundle = getIntent().getExtras();
        String dato=bundle.getString("direccion");
        web1.loadUrl("http://" + dato);
    }

    public void finalizar(View v) {
        finish();
    }
}
```

```
}  
}
```

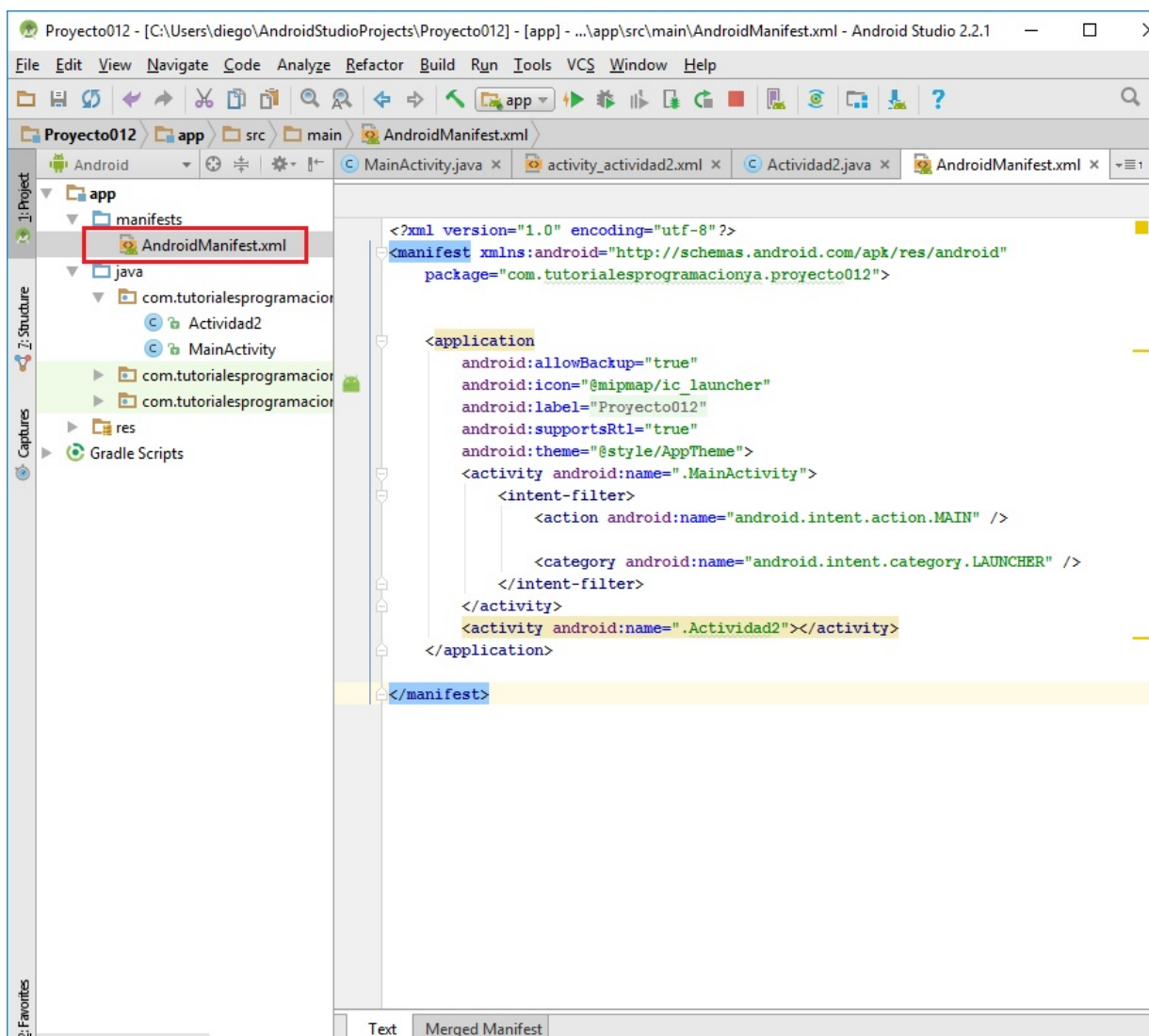
En esta clase definimos una variable de tipo Bundle y la inicializamos llamando al método `getExtras()` de la clase `Intent` (esto lo hacemos para recuperar el o los parámetros que envió la otra actividad (`Activity`)):

```
Bundle bundle = getIntent().getExtras();  
String dato=bundle.getString("direccion");  
web1.loadUrl("http://" + dato);
```

El método `loadUrl` de la clase `WebView` permite visualizar el contenido de un sitio web.

Importante

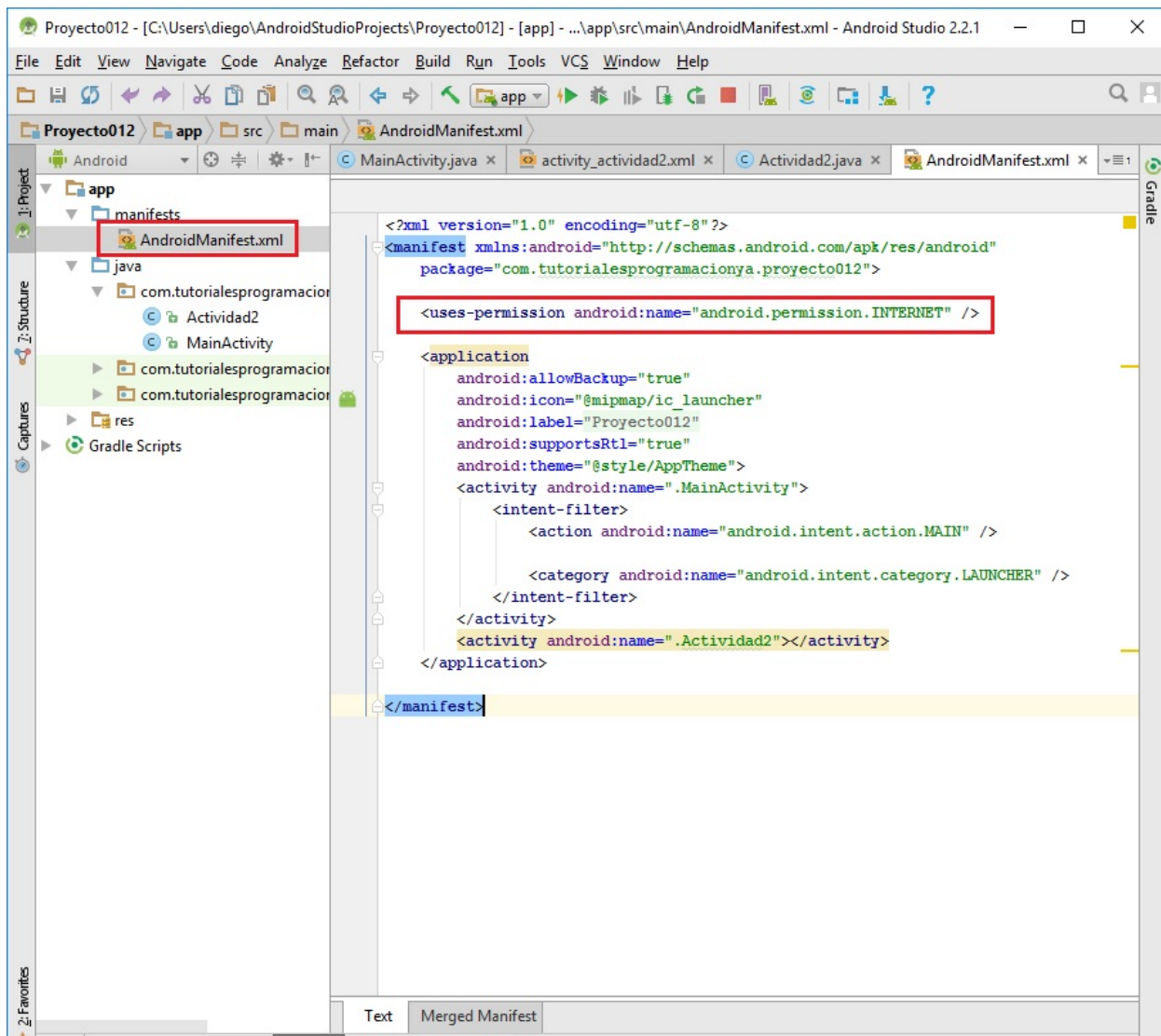
Como nuestra aplicación debe acceder a internet debemos hacer una configuración en el archivo "AndroidManifest.xml", podemos ubicar este archivo:



Agregamos el permiso tipeando lo siguiente en este archivo:

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

Luego el archivo `AndroidManifest.xml` queda con el permiso agregado:

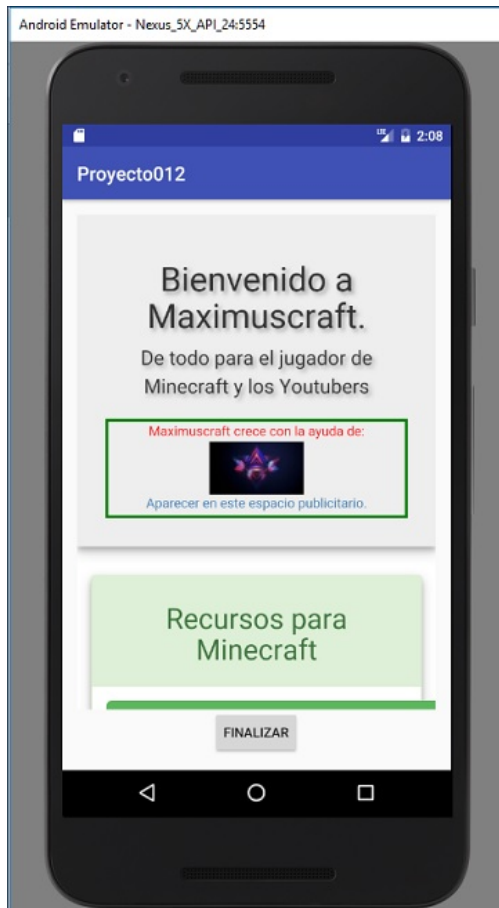


Ahora nuestro programa debería estar funcionando completamente.

La primer ventana debería ser algo similar a esto:



La segunda ventana debería ser algo similar a esto otro:



Este proyecto lo puede descargar en un zip desde este enlace: [proyecto012.zip](#)



Sistema de Contabilidad

Seguro y muy fácil de usar.
Facturas y NCF válidas ante la DGII.

[Retornar](#)