



# Xamarin Diplomado

## Latinoamérica

# Módulo 2: Introducción al desarrollo para Android



Enrique Aguilar  
@enriqueaguilar  
Follow #XamarinDiplomado

Lección 2: Introducción a los Activities en Xamarin



**Xamarin**  
MOST VALUABLE  
PROFESSIONAL

**Xamarin** Diplomado  
Latinoamérica



# Agenda

## Lección 2

### Introducción a los Activities

#### Tema 1

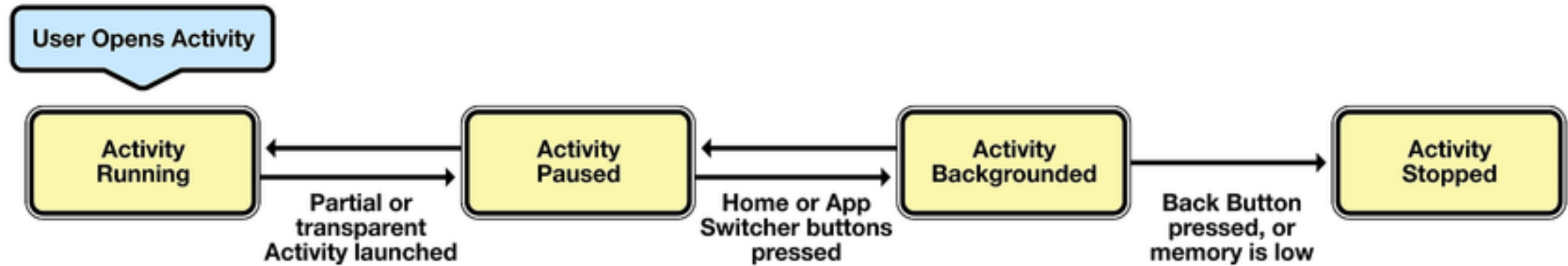
#### Ciclo de vida de los Activities

#### Tema 2

#### Activities y paso de variables entre Ventanas



# Ciclo de vida de los Activities



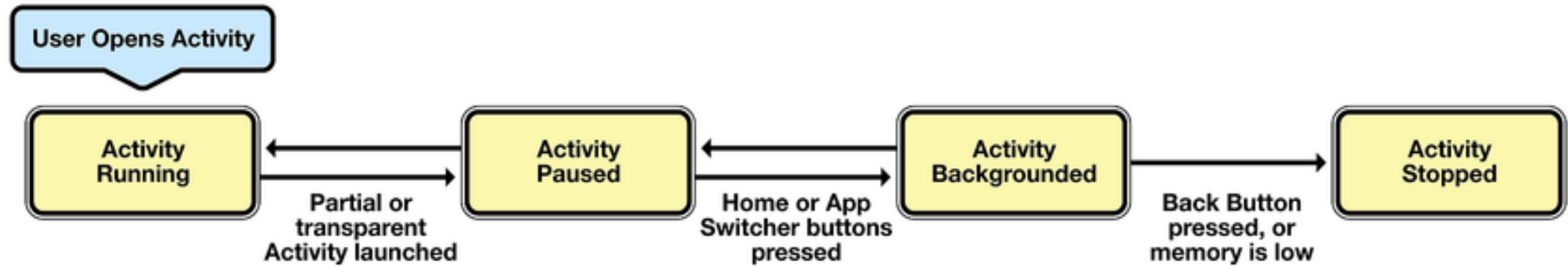
## Activo

Es cuando se encuentra corriendo en ese momento, es el más importante y solo puede ser eliminada la actividad por el sistema operativo si se desborda la memoria o cuando la interfaz gráfica no responde, por ejemplo.

## Pausado

Cuando el dispositivo pasa a reposo o una actividad permanece parcialmente oculta por una nueva instancia, Es una etapa donde permanece en ejecución y mantiene su estado. Es el segundo en prioridad.

# Ciclo de vida de los Activities



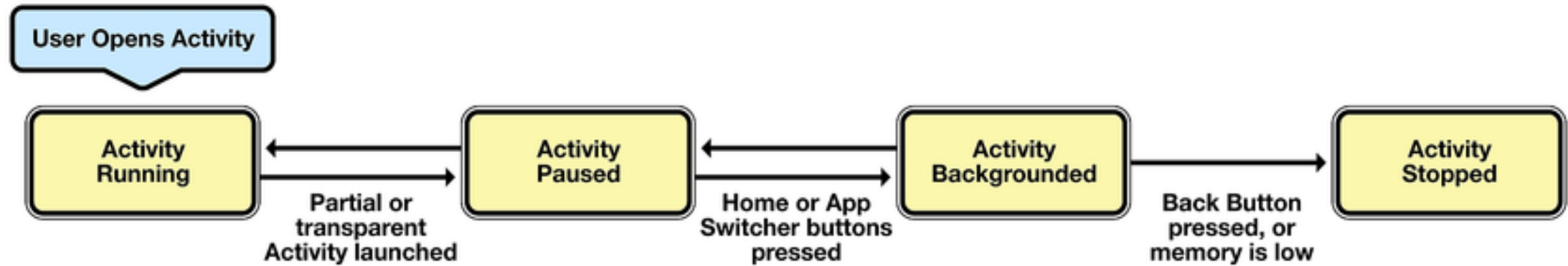
## Activo

Es cuando se encuentra en ejecución, es el más importante y solo puede ser eliminada por el sistema operativo si se desborda la memoria o cuando la interfaz gráfica no responde, por ejemplo.

## Pausado

Cuando el dispositivo pasa a reposo o una actividad permanece parcialmente oculta por una nueva instancia, Es una etapa donde permanece en ejecución y mantiene su estado. Es el segundo en prioridad.

# Ciclo de vida de los Activities



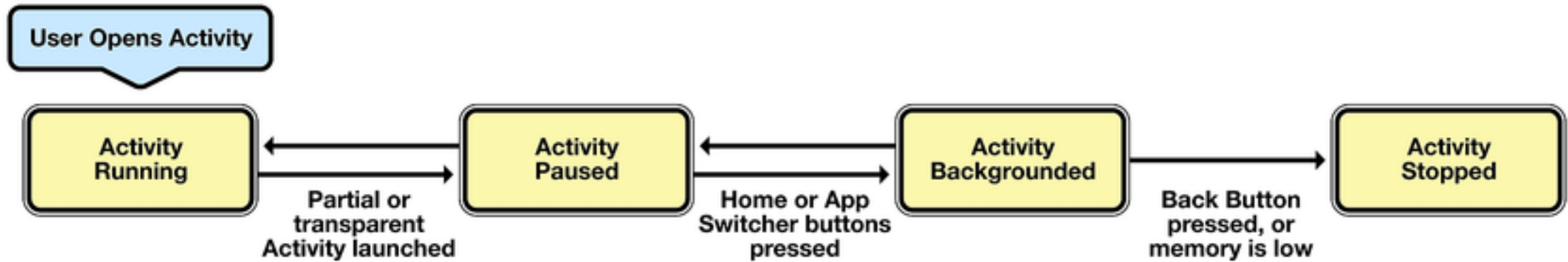
## Activo

Es cuando se encuentra corriendo en ese momento, es el más importante y solo puede ser eliminada la actividad por el sistema operativo si se desborda la memoria o cuando la interfaz gráfica no responde, por ejemplo.

## Pausado

Cuando el dispositivo pasa a reposo o una permanece parcialmente oculta por una nueva instancia. Es una etapa donde permanece en ejecución y mantiene su estado. Es el segundo en prioridad.

# Ciclo de vida de los Activities



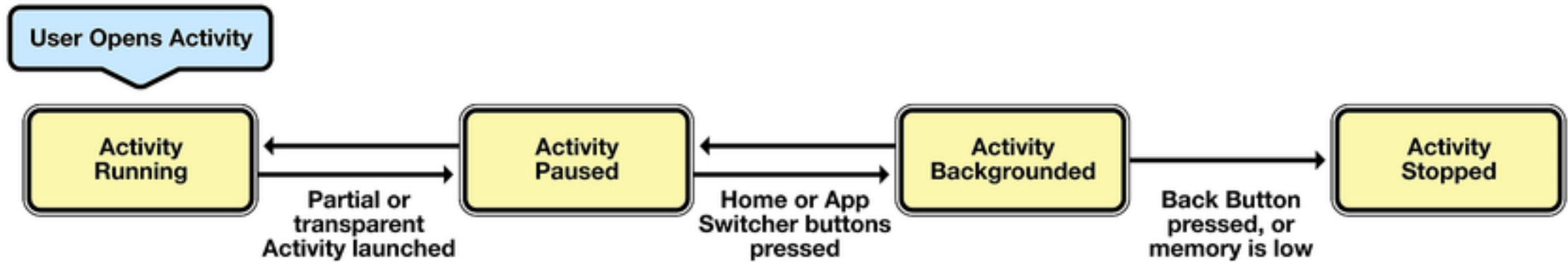
Actividad Reiniciada

Si el usuario navega de regreso a la actividad se pudiera reiniciar o restaurar a un estado anterior.

Detenido / Background

Es la actividad que es detenida o que pasa a trabajos de background. Si la actividad permanece detenida durante un tiempo considerable, el sistema puede eliminar la actividad por completo.

# Ciclo de vida de los Activities



## Actividad Reiniciada

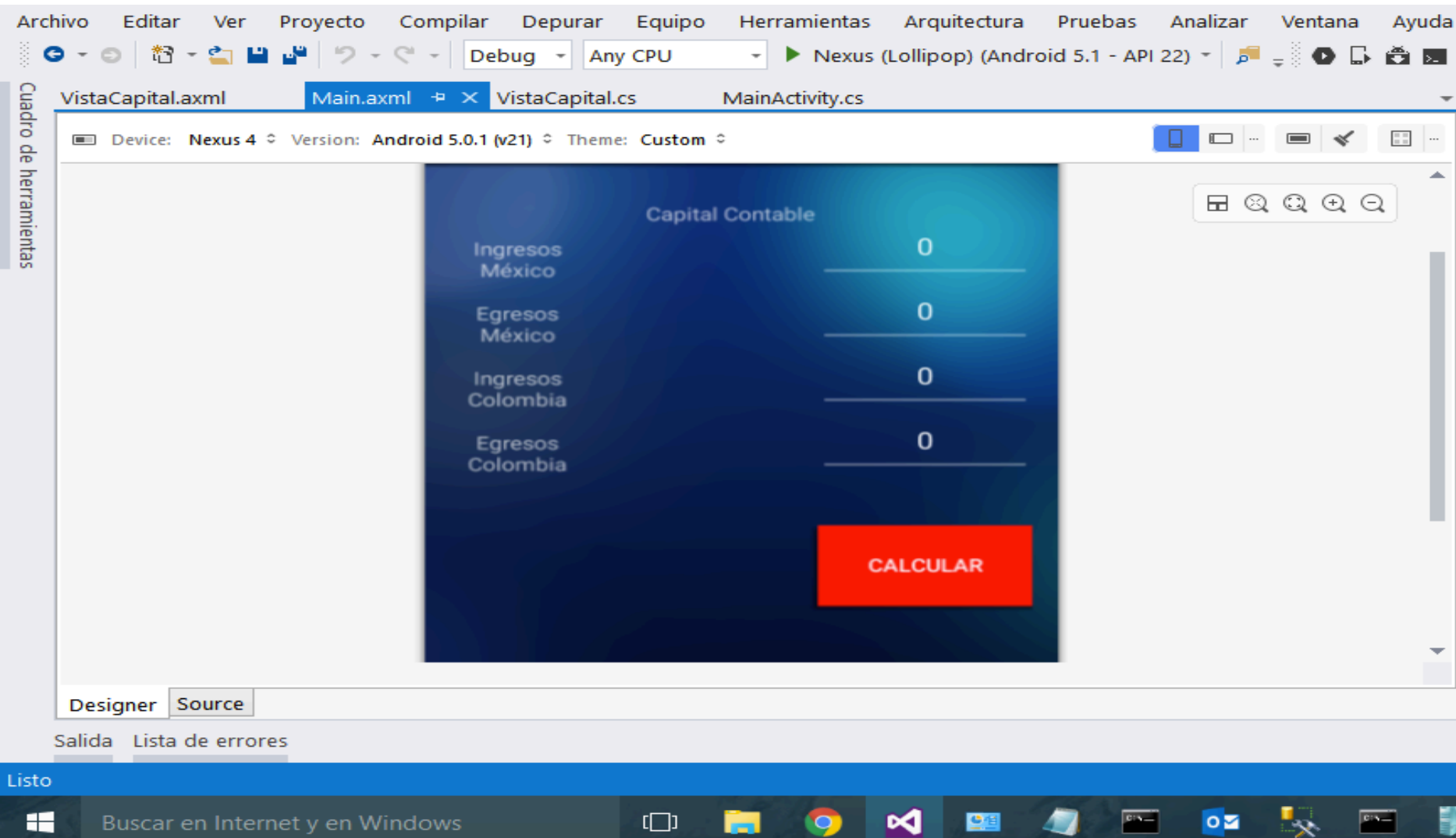
Si una actividad ha sido pausada o detenida. Si el usuario navega de regreso a la actividad se pudiera reiniciar o restaurar a un estado anterior.

## Detenido / Background

Es la actividad que es detenida o que pasa a trabajos de background. Si la actividad permanece detenida durante un tiempo considerable, el sistema puede eliminar la actividad por completo.



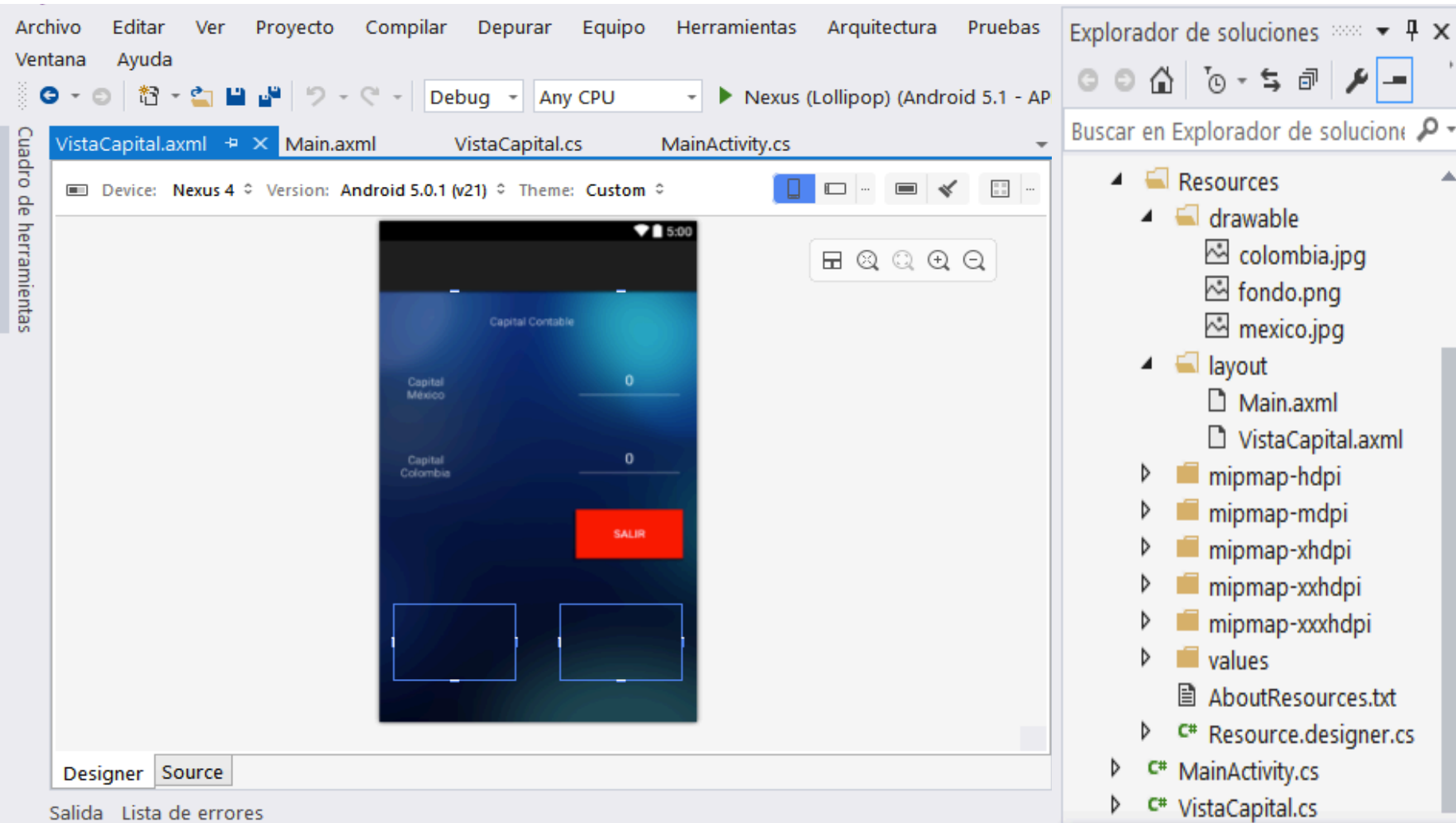
# Activities y paso de variables entre Ventanas



Creamos una interfaz gráfica que contenga 4 cajas de texto con sus respectivas etiquetas y un botón de Calcular.

Podemos re-utilizar el ejercicio del Módulo 2 – Lección 1.

# Activities y paso de variables entre Ventanas

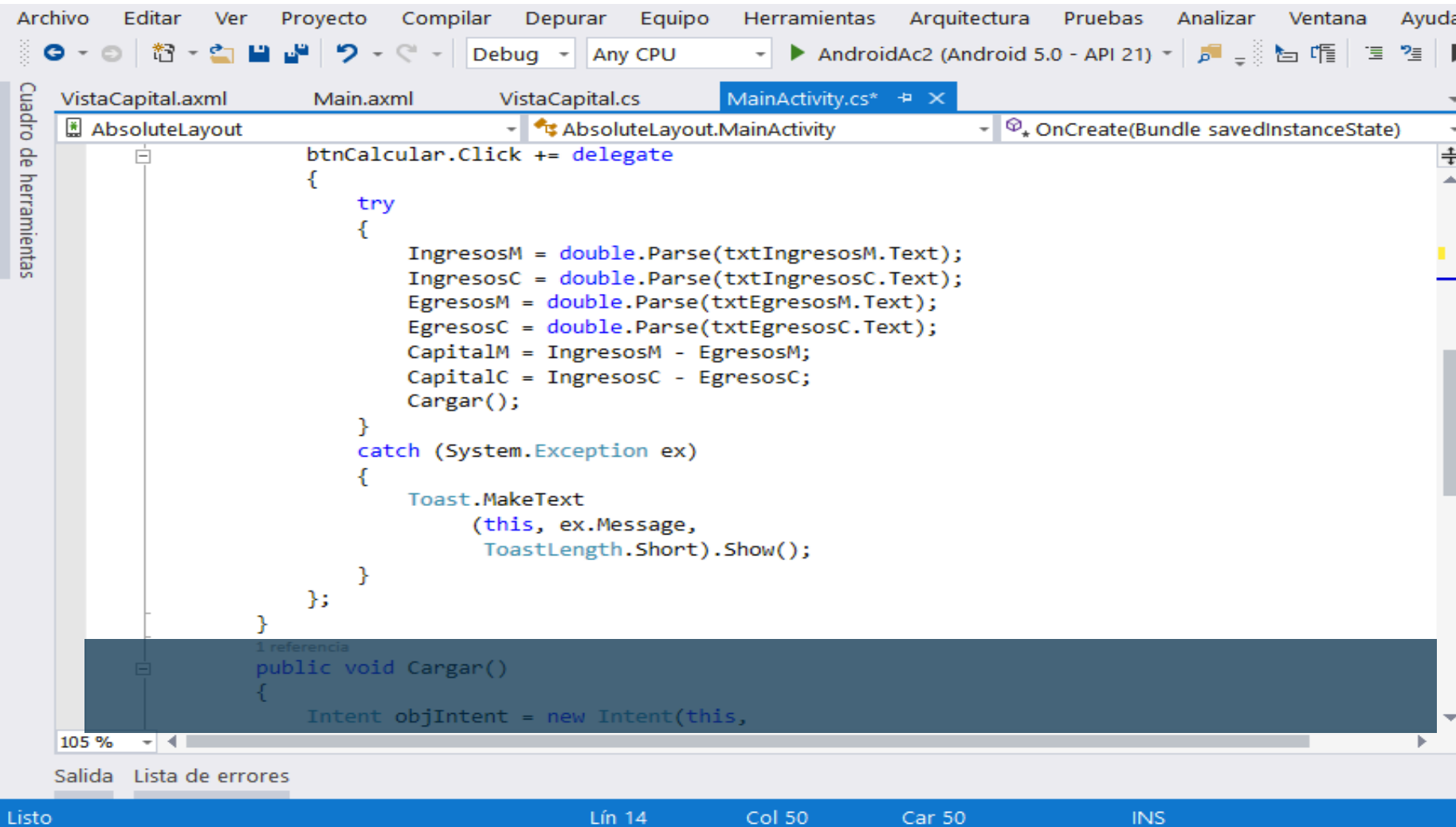


En el explorador de soluciones agregamos al proyecto un nuevo elemento de tipo Android Layout y un nuevo Activity.

Podemos reutilizar el código axml de la pantalla principal, copiar y pegar y asignamos la ubicación de los elementos copiados, y les agregamos el *id*.

De la misma forma agregamos a la carpeta drawable dos imágenes, en este caso referentes a México y Colombia.

# Activities y paso de variables entre Ventanas



```
Archivo  Editar  Ver  Proyecto  Compilar  Depurar  Equipo  Herramientas  Arquitectura  Pruebas  Analizar  Ventana  Ayuda
Debug  Any CPU  AndroidAc2 (Android 5.0 - API 21)
VistaCapital.xml  Main.xml  VistaCapital.cs  MainActivity.cs*
AbsoluteLayout  AbsoluteLayout.MainActivity  onCreate(Bundle savedInstanceState)
btnCalcular.Click += delegate
{
    try
    {
        IngresosM = double.Parse(txtIngresosM.Text);
        IngresosC = double.Parse(txtIngresosC.Text);
        EgresosM = double.Parse(txtEgresosM.Text);
        EgresosC = double.Parse(txtEgresosC.Text);
        CapitalM = IngresosM - EgresosM;
        CapitalC = IngresosC - EgresosC;
        Cargar();
    }
    catch (System.Exception ex)
    {
        Toast.MakeText
            (this, ex.Message,
             ToastLength.Short).Show();
    }
};
}
1 referencia
public void Cargar()
{
    Intent objIntent = new Intent(this,
```

105 %

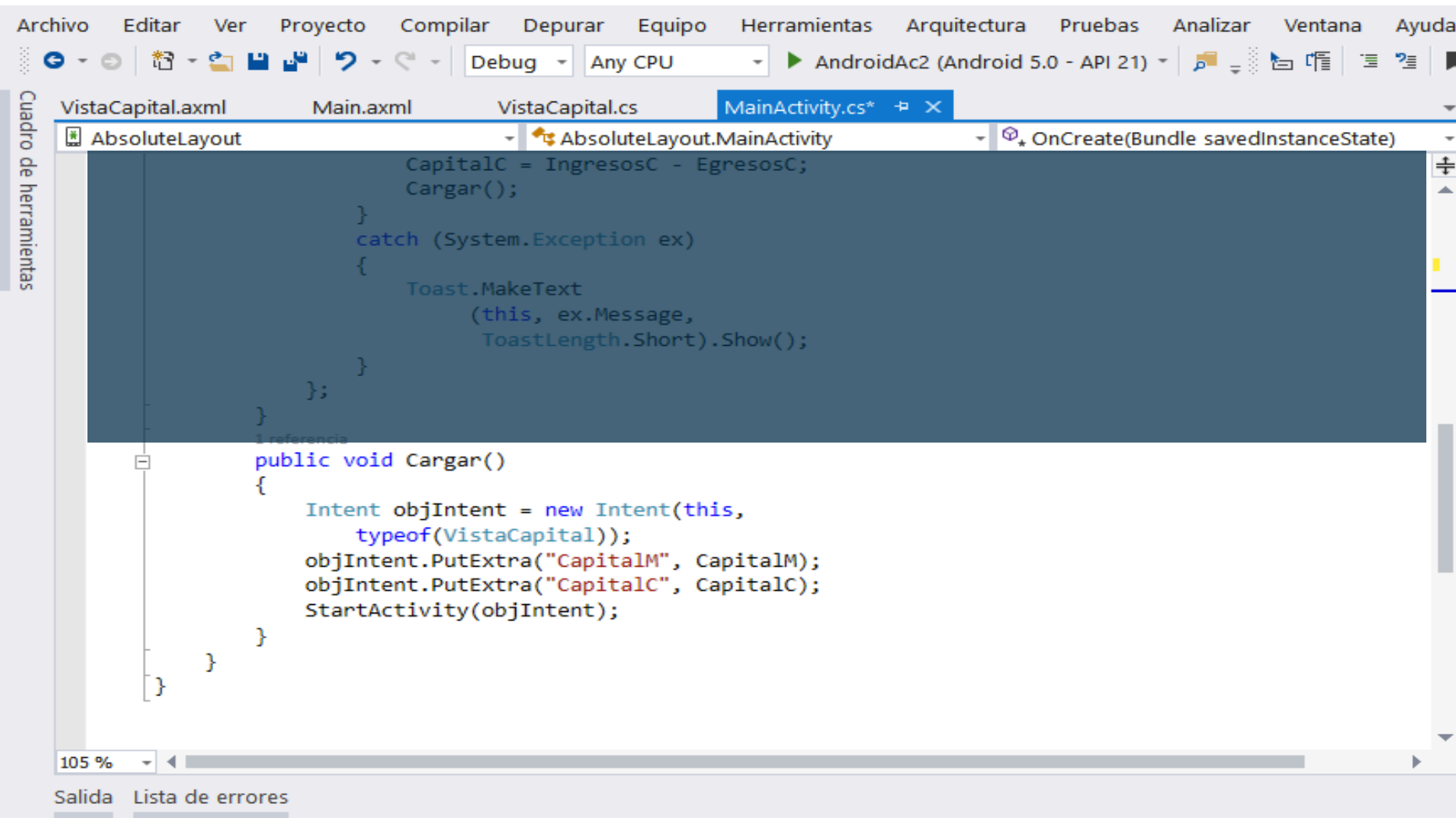
Salida Lista de errores

Listo Lín 14 Col 50 Car 50 INS

Al igual que el ejercicio anterior se realizan los enlaces con los elementos de la interfaz gráfica debajo del *SetContentView*.

Recibimos los datos en las variables y hacemos la operación.

# Activities y paso de variables entre Ventanas

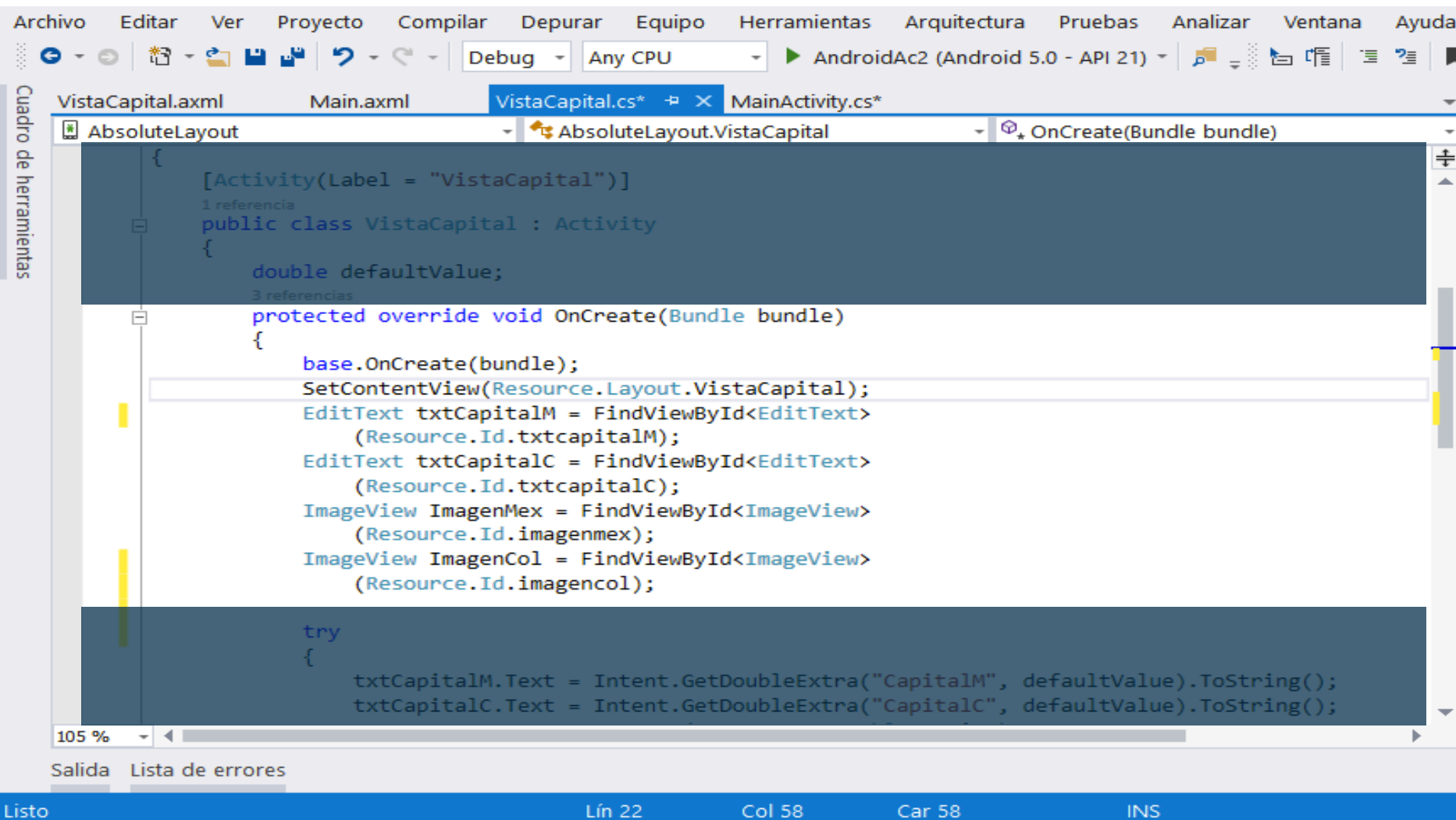


```
CapitalC = IngresosC - EgresosC;
Cargar();
}
catch (System.Exception ex)
{
    Toast.MakeText
        (this, ex.Message,
         ToastLength.Short).Show();
}
};
}
}

public void Cargar()
{
    Intent objIntent = new Intent(this,
        typeof(VistaCapital));
    objIntent.PutExtra("CapitalM", CapitalM);
    objIntent.PutExtra("CapitalC", CapitalC);
    StartActivity(objIntent);
}
}
```

Agregamos un método llamado `Cargar`, en el cual se genera una variable de tipo *Intent*, y en la cual adjuntamos con la instrucción *PutExtra* el nombre clave y la variable cuyo contenido se enviará al siguiente Activity.

# Activities y paso de variables entre Ventanas



```
Archivo  Editar  Ver  Proyecto  Compilar  Depurar  Equipo  Herramientas  Arquitectura  Pruebas  Analizar  Ventana  Ayuda
Debug  Any CPU  AndroidAc2 (Android 5.0 - API 21)
VistaCapital.xml  Main.xml  VistaCapital.cs*  MainActivity.cs*
AbsoluteLayout  AbsoluteLayout.VistaCapital  * OnCreate(Bundle bundle)
[Activity(Label = "VistaCapital")]
1 referencia
public class VistaCapital : Activity
{
    double defaultValue;
    3 referencias
    protected override void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);
        SetContentView(Resource.Layout.VistaCapital);
        EditText txtCapitalM = FindViewById<EditText>
            (Resource.Id.txtcapitalM);
        EditText txtCapitalC = FindViewById<EditText>
            (Resource.Id.txtcapitalC);
        ImageView ImagenMex = FindViewById<ImageView>
            (Resource.Id.imagenmex);
        ImageView ImagenCol = FindViewById<ImageView>
            (Resource.Id.imagencol);

        try
        {
            txtCapitalM.Text = Intent.GetDoubleExtra("CapitalM", defaultValue).ToString();
            txtCapitalC.Text = Intent.GetDoubleExtra("CapitalC", defaultValue).ToString();
        }
    }
}
```

105 %

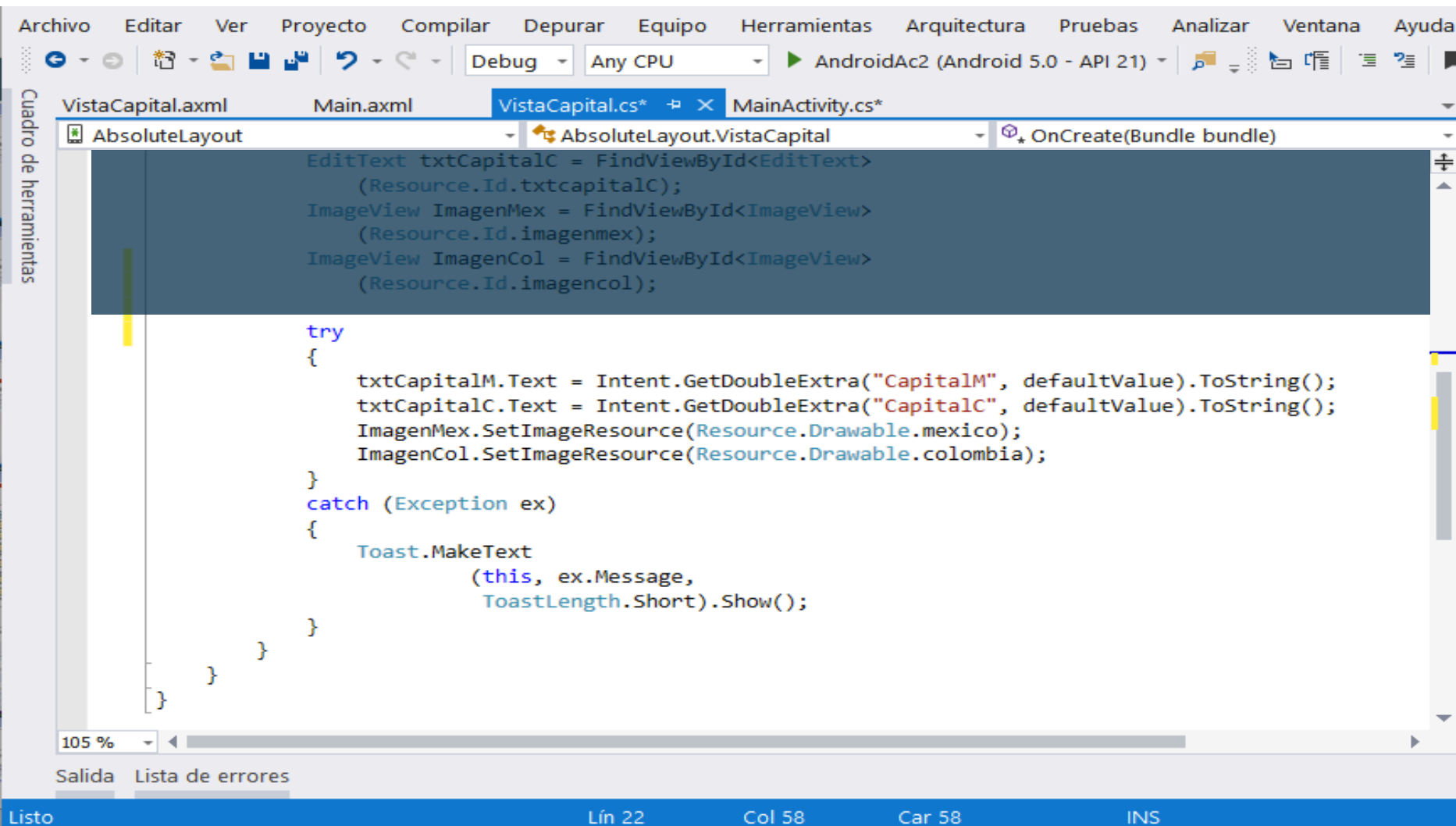
Salida Lista de errores

Listo Lín 22 Col 58 Car 58 INS

En el nuevo Activity hacemos el enlace de los elementos de la nueva Interfaz Gráfica.

En el *SetContentView* se deberá de indicar que el Layout que se toma es el de VistaCapital.

# Activities y paso de variables entre Ventanas



```
Archivo  Editar  Ver  Proyecto  Compilar  Depurar  Equipo  Herramientas  Arquitectura  Pruebas  Analizar  Ventana  Ayuda
Debug  Any CPU  AndroidAc2 (Android 5.0 - API 21)
VistaCapital.xml  Main.xml  VistaCapital.cs*  MainActivity.cs*
AbsoluteLayout  AbsoluteLayout.VistaCapital  onCreate(Bundle bundle)
EditText txtCapitalC = FindViewById<EditText>
    (Resource.Id.txtcapitalC);
ImageView ImagenMex = FindViewById<ImageView>
    (Resource.Id.imagemex);
ImageView ImagenCol = FindViewById<ImageView>
    (Resource.Id.imagencol);

try
{
    txtCapitalM.Text = Intent.GetDoubleExtra("CapitalM", defaultValue).ToString();
    txtCapitalC.Text = Intent.GetDoubleExtra("CapitalC", defaultValue).ToString();
    ImagenMex.SetImageResource(Resource.Drawable.mexico);
    ImagenCol.SetImageResource(Resource.Drawable.colombia);
}
catch (Exception ex)
{
    Toast.MakeText
        (this, ex.Message,
        ToastLength.Short).Show();
}
}
```

Salida Lista de errores

Listo Lín 22 Col 58 Car 58 INS

En las cajas de texto se recibe de la instrucción *Intent.GetDoubleExtra* el contenido de la variable a un tipo de datos similar al que recibe y la clave de identificación del dato que procede de el Activity anterior.

De la misma forma se asignan las imágenes en los objetos *ImageView* para su visualización.



# Tema 2: Activities y paso de variables entre Ventanas

The image shows a development environment with Visual Studio on the left and the Xamarin Android Player on the right. The Visual Studio window displays the 'VistaCapital.xml' file, which contains a table titled 'Capital Contable' with four rows: 'Ingresos México', 'Egresos México', 'Ingresos Colombia', and 'Egresos Colombia'. The values in the table are all 0. The Xamarin Android Player window shows the app running on a Nexus (Lollipop) device. The app's interface is a 'Convertidor' screen with a blue background. It has four input fields for the same categories as the table in Visual Studio: 'Ingresos México' (5000), 'Egresos México' (2000), 'Ingresos Colombia' (4000), and 'Egresos Colombia' (1000). A red 'Calcular' button is at the bottom. The status bar at the top of the player shows the time as 11:09. The bottom of the Visual Studio window shows the command prompt output, indicating a successful compilation and deployment to the Nexus (Lollipop) device.

Visual Studio - Microsoft Visual Studio

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas Analizar

VistaCapital.xml Main.xml VistaCapital.cs MainActivity.cs

Device: Nexus 4 Version: Android 5.0.1 (v21) Theme: Custom

Capital Contable

Ingresos México	0
Egresos México	0
Ingresos Colombia	0
Egresos Colombia	0

Salida

Mostrar salida de: Compilación

```
2>C:\Users\Enrique\AppData\Local\Android\android-sdk\build-tools\23.0.1\zipalign.exe 4 "C:\Users\Enrique\AppData\Local\Android\android-sdk\platform-tools\adb.exe -s 10.71.34.101:5555 dev
2>C:\Users\Enrique\AppData\Local\Android\android-sdk\platform-tools\adb.exe -s 10.71.34.101:5555 she
2>C:\Users\Enrique\AppData\Local\Android\android-sdk\platform-tools\adb.exe -s 10.71.34.101:5555 she
2>C:\Users\Enrique\AppData\Local\Android\android-sdk\platform-tools\adb.exe -s 10.71.34.101:5555 she
2>C:\Users\Enrique\AppData\Local\Android\android-sdk\platform-tools\adb.exe -s 10.71.34.101:5555 she
2>Compilación correcta.
2>Deploy successfully on Nexus (Lollipop).
===== Compilar: 1 correctos, 0 incorrectos, 0 actualizados, 0 omitidos =====
===== Implementar: 1 correctos, 0 incorrectos, 0 omitidos =====
```

Salida Lista de errores

Operación Implementar finalizada correctamente

Xamarin Android Player - [Nexus (Lollipop)]

Convertidor

Capital Contable

Ingresos México	5000
Egresos México	2000
Ingresos Colombia	4000
Egresos Colombia	1000

Calcular

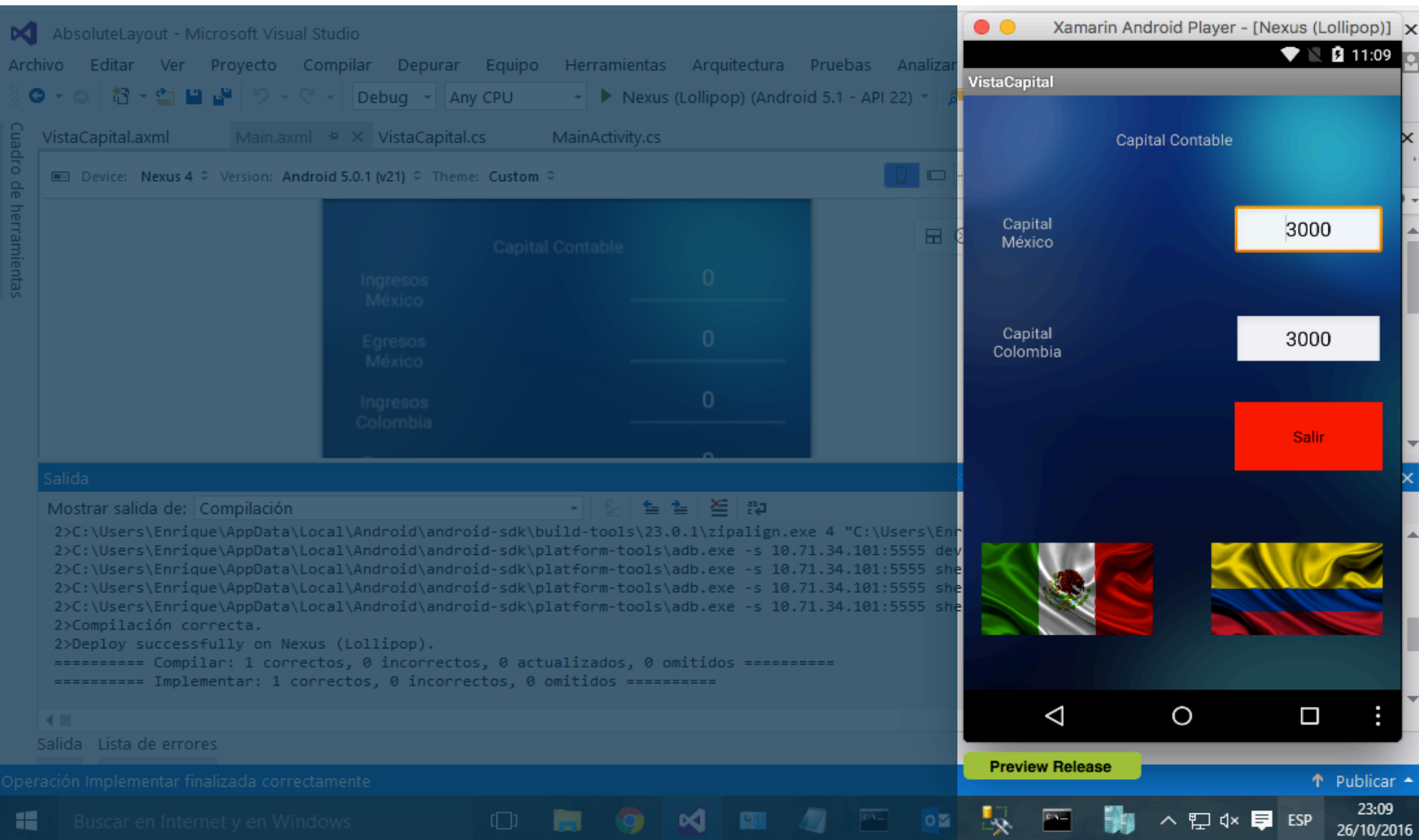
Preview Release

Publicar

23:09 26/10/2016

Al ejecutar llenamos los datos de la interfaz y procedemos a presionar el botón de Calcular.

# Tema 2: Activities y paso de variables entre Ventanas



Se envían los datos al segundo Activity, quién muestra la información en las cajas de texto asignadas, además de cargar las imágenes.

Para el botón de salir se puede utilizar la instrucción *Android.OS.Process.KillProcess(Android.OS.Process.MyPid());* para terminar la aplicación.



DEMO



# Gracias por su atención

## Módulo 2: Introducción al desarrollo para Android

Lección 2  
Introducción a los Activities



**Xamarin**  
MOST VALUABLE  
PROFESSIONAL

Enrique Aguilar  
@enriqueaguilar  
Follow  
#XamarinDiplomado

