Windows

Aplicaciones Universales de Windows

¿Qué dispositivo estoy corriendo?

Víctor Moreno

Microsoft MVP
@vmorenoz

Microsoft

¿Qué voy a aprender?

En esta demostración, aprenderás como identificar la plataforma que esta corriendo una aplicación universal de Windows 10.

Objetivo

Analizar un proyecto en Visual Studio 2015 que nos permita identificar mediante la API:

Windows.System.Profile.AnalyticsInfo.VersionInfo.DeviceFamily

Bajo que dispositivo esta corriendo nuestra aplicación universal de Windows 10. Las plataformas a distinguir serán:

- Windows 10 Mobile.
- Windows 10 IoT.
- Windows 10 XBOX.
- Windows 10 Desktop.
- Windows 10 Hololens.

Requerimientos

- Visual Studio 2015.
- Windows 10.

Demostración…

Proyecto

Este proyecto se encuentra disponible en GitHub para su descarga:

https://github.com/vemoreno/WhatDeviceRunning

Proyecto

Archivo MainPage.xaml | MainPage.cs

Estos archivos mantienen la interfaz y el código principal del proyecto que identificará que dispositivo se está corriendo.

Archivo DeviceMobile.xaml | DeviceMobile.cs

Estos archivos mantienen la interfaz y el código para identificar algunas operaciones que se pueden realizar con la familia Mobile (botones de retroceso, cámara, etc).

MainPage.xaml

Basta con presionar el botón de “What device am I running?” para que el código haga su trabajo.

MainPage.cs

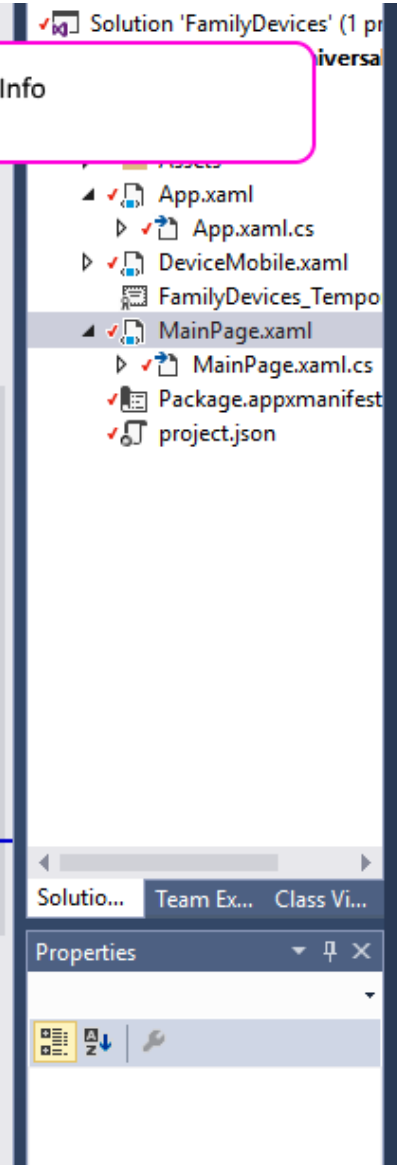
El código de identificación se encuentra en el evento click de este botón.

MainPage.cs

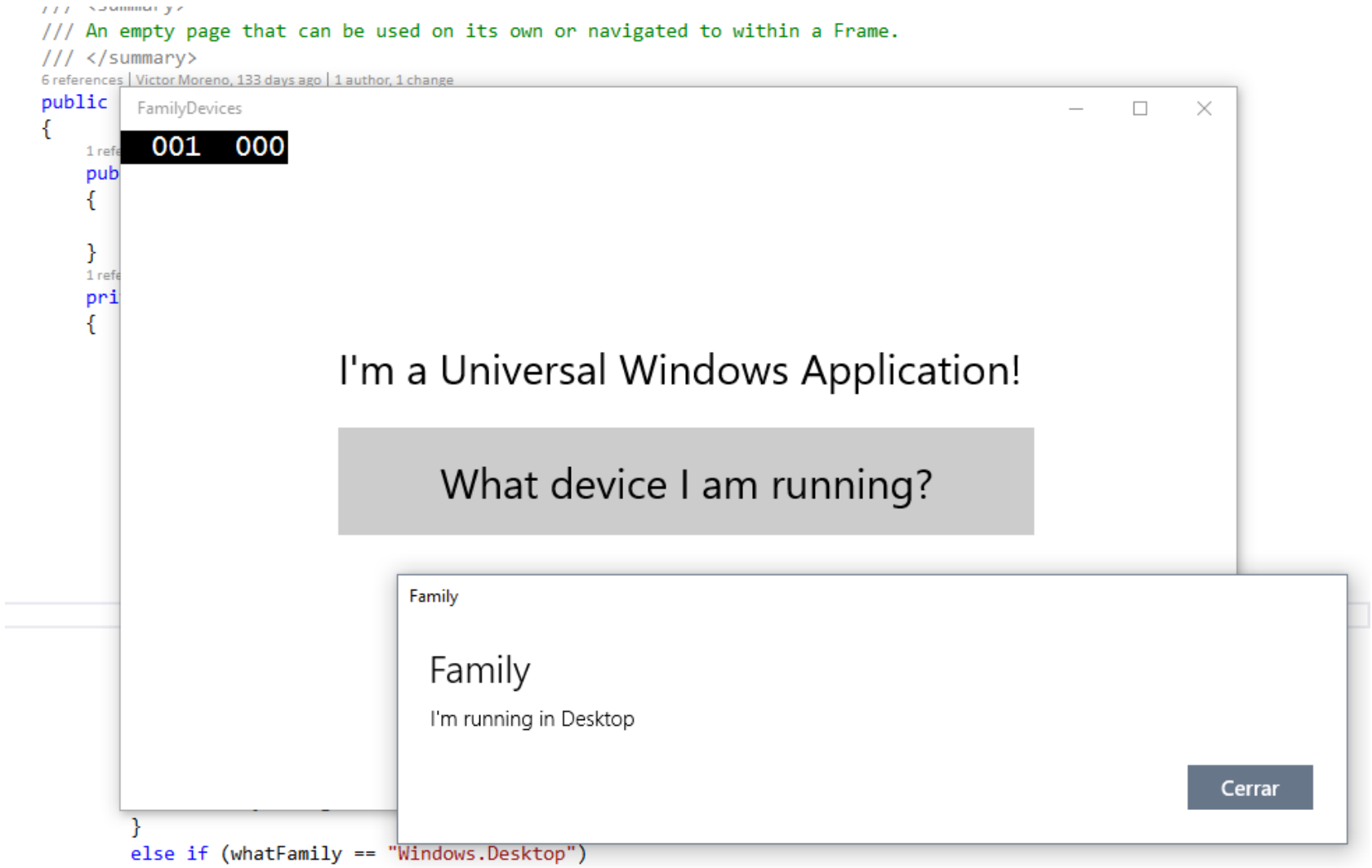
```
private async void btnWhatDevice_Click(object sender, RoutedEventArgs e)
{
    string whatFamily = Windows.System.Profile.AnalyticsInfo.VersionInfo.DeviceFamily;

    if (whatFamily == "Windows.Mobile")
    {
        MessageDialog myMessage = new MessageDialog("I'm running in Mobile Device", "Family");
        await myMessage.ShowAsync();

        this.Frame.Navigate(typeof(DeviceFamily));
    }
    else if (whatFamily == "Windows.IoT")
    {
        MessageDialog myMessage = new MessageDialog("I'm running in an electronic device (IoT)", "Family");
        await myMessage.ShowAsync();
    }
    else if (whatFamily == "Windows.Xbox")
    {
        MessageDialog myMessage = new MessageDialog("I'm running in Xbox console", "Family");
        await myMessage.ShowAsync();
    }
    else if (whatFamily == "Windows.Desktop")
    {
        MessageDialog myMessage = new MessageDialog("I'm running in Desktop", "Family");
        await myMessage.ShowAsync();
    }
    else if (whatFamily == "Windows.Hololens")
    {
        MessageDialog myMessage = new MessageDialog("I'm running in Hololens glasses", "Family");
        await myMessage.ShowAsync();
    }
}
```

API AnalyticsInfo

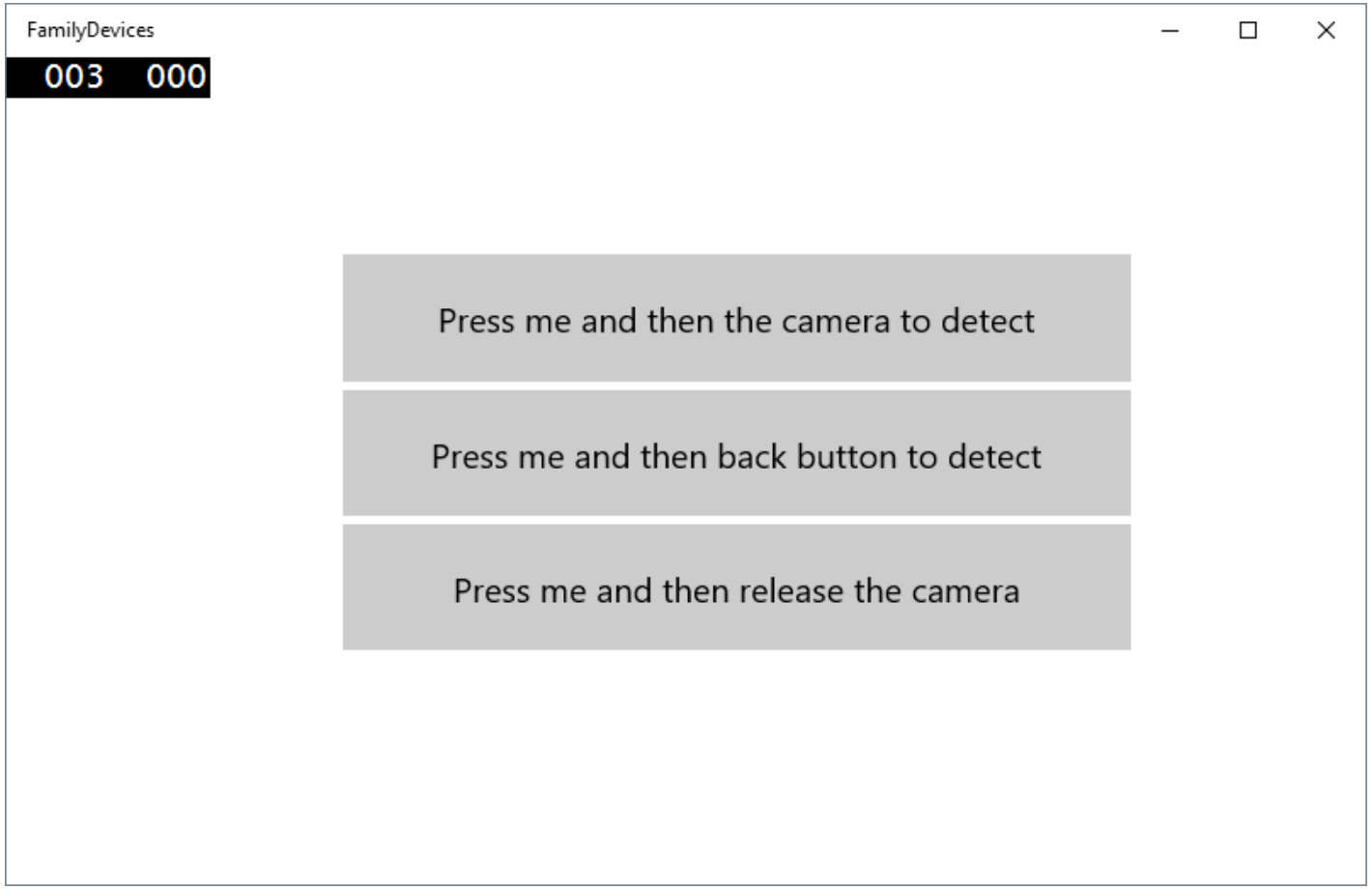
Valores de API AnalyticsInfo



Ejecutando


/// An empty page that can be used on its own or navigated to within a Frame.
/// </summary>
6 references | Victor Moreno, 133 days ago | 1 author, 1 change
public
{
pub
{
}
pri
{
FamilyDevices
001 000
I'm a Universal Windows Application!
What device I am running?
Family
Family
I'm running in Desktop
Cerrar
}
else if (whatFamily == "Windows.Desktop")

DeviceFamily.xaml



DeviceFamily.cs

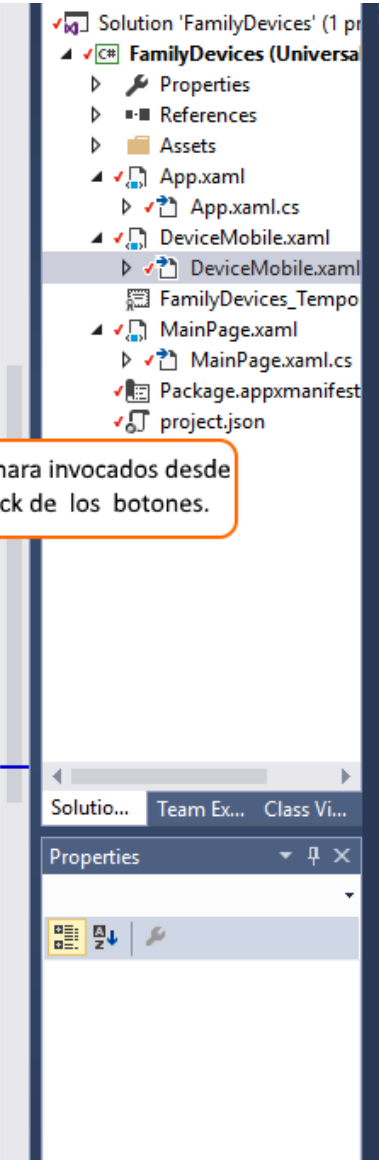
Basta con presionar cada botón de la ventana para que el código haga su trabajo.

Este código es referente a una acción en especial de la familia Windows 10 Mobile.

DeviceFamily.cs

```
private void btnSeeCamera_Click(object sender, RoutedEventArgs e)
{
    Windows.Phone.UI.Input.HardwareButtons.CameraPressed += HardwareButtons_CameraPressed;
}

1 reference | Victor Moreno, 133 days ago | 1 author, 1 change
private async void HardwareButtons_CameraPressed(object sender, Windows.Phone.UI.Input.CameraEventArgs e)
{
    MessageDialog myMessage = new MessageDialog("You pressed the camera!");
    await myMessage.ShowAsync();
}

1 reference | Victor Moreno, 133 days ago | 1 author, 1 change
private void button1_Click(object sender, RoutedEventArgs e)
{
    Windows.Phone.UI.Input.HardwareButtons.BackPressed += HardwareButtons_BackPressed;
}

1 reference | Victor Moreno, 133 days ago | 1 author, 1 change
private async void HardwareButtons_BackPressed(object sender, Windows.Phone.UI.Input.BackPressedEventArgs e)
{
    MessageDialog myMessage = new MessageDialog("You pressed back button!");
    await myMessage.ShowAsync();
}

1 reference | Victor Moreno, 133 days ago | 1 author, 1 change
private void btnReleaseCamera_Click(object sender, RoutedEventArgs e)
{
    Windows.Phone.UI.Input.HardwareButtons.CameraReleased += HardwareButtons_CameraReleased;
}

1 reference | Victor Moreno, 133 days ago | 1 author, 1 change
private async void HardwareButtons_CameraReleased(object sender, Windows.Phone.UI.Input.CameraEventArgs e)
{
    MessageDialog myMessage = new MessageDialog("Camera released!");
    await myMessage.ShowAsync();
}
```

Eventos de cámara invocados desde cada evento click de los botones.



Windows

Víctor Moreno
@vmorenoz

http://blogs.itpro.es/eduardocloud

Microsoft