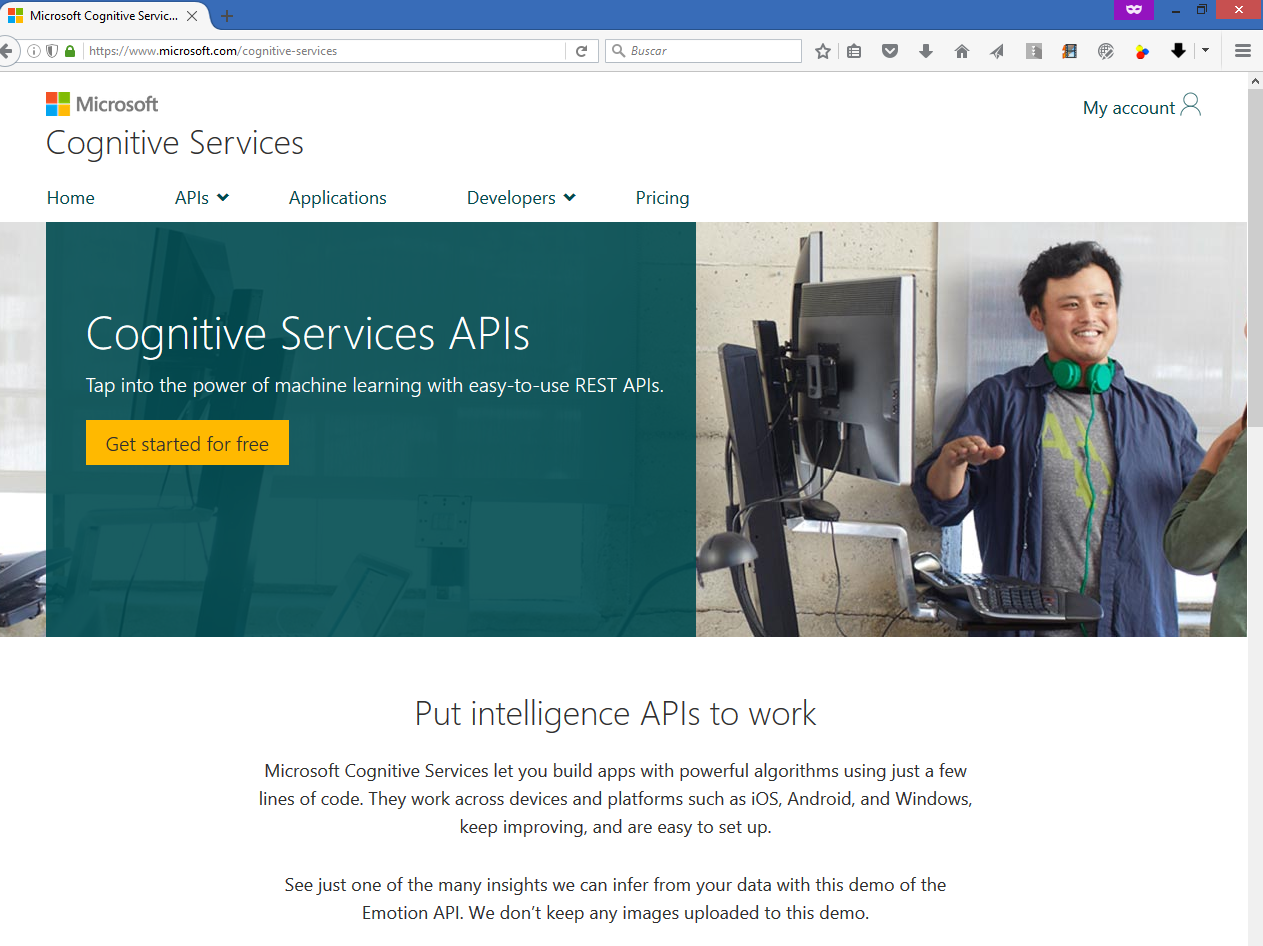
**Práctica 07 – Microsoft Cognitive Services parte 1**

El objetivo de esta práctica es crear una aplicación que nos permita analizar una imagen y obtener información de interés, por ejemplo, reconocer las emociones que expresa un rostro en una fotografía e identificar elementos que componen una imagen.

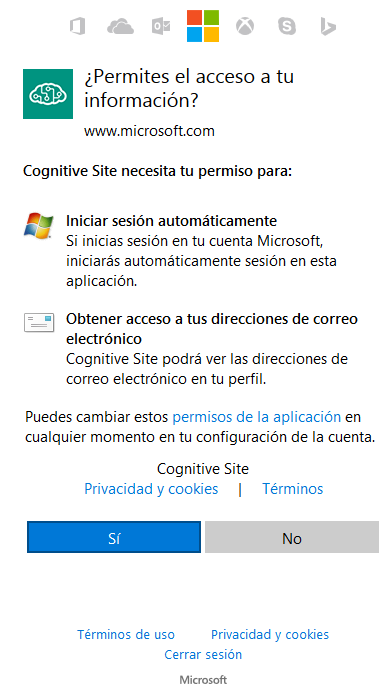
Para esta práctica necesitas una cuenta de Microsoft Cognitive Services. Existen planes gratuitos que puedes utilizar para probar el servicio.

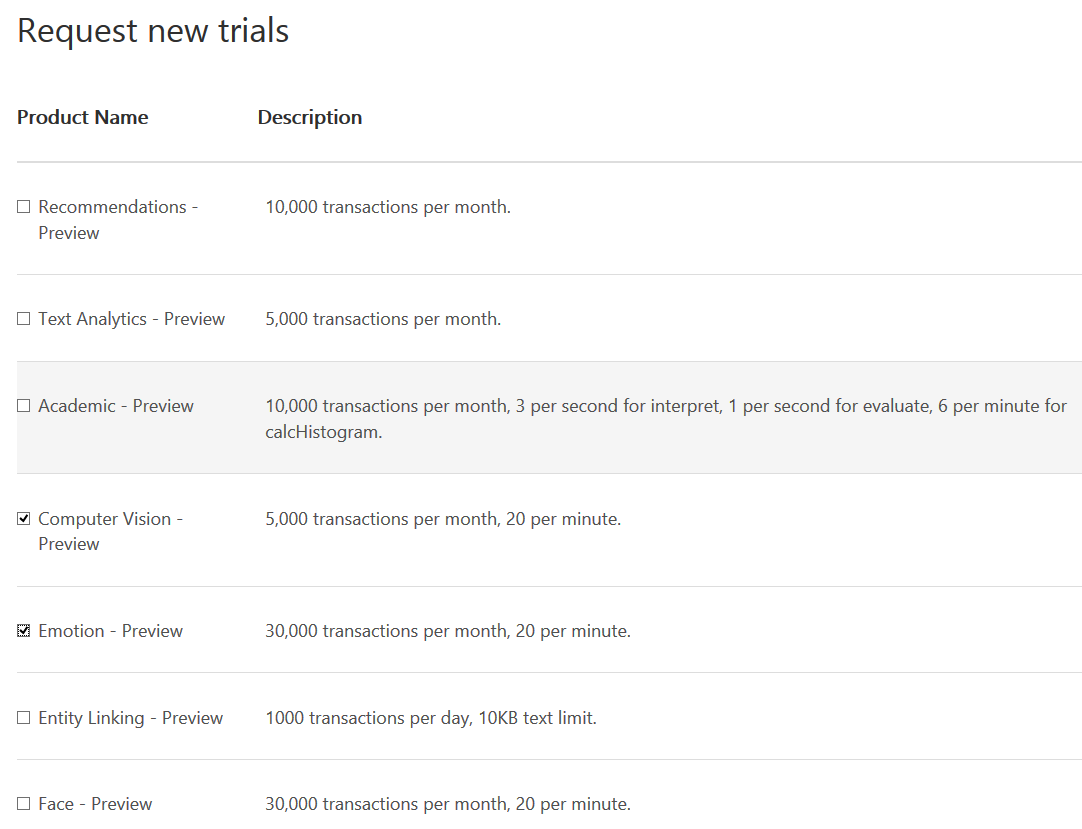
**Paso 1:** Ingresa al portal de Microsoft Cognitive Services (<http://microsoft.com/cognitive>).

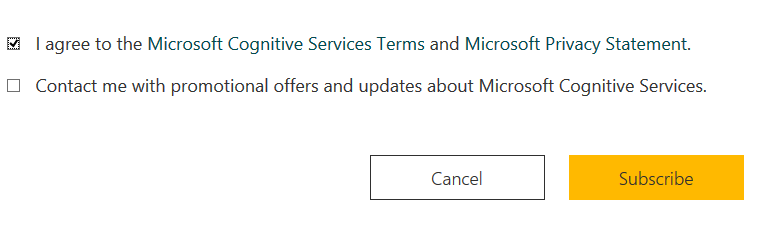


**Paso 2.** Regístrate dando clic en My account. Acepta la solicitud de permisos.

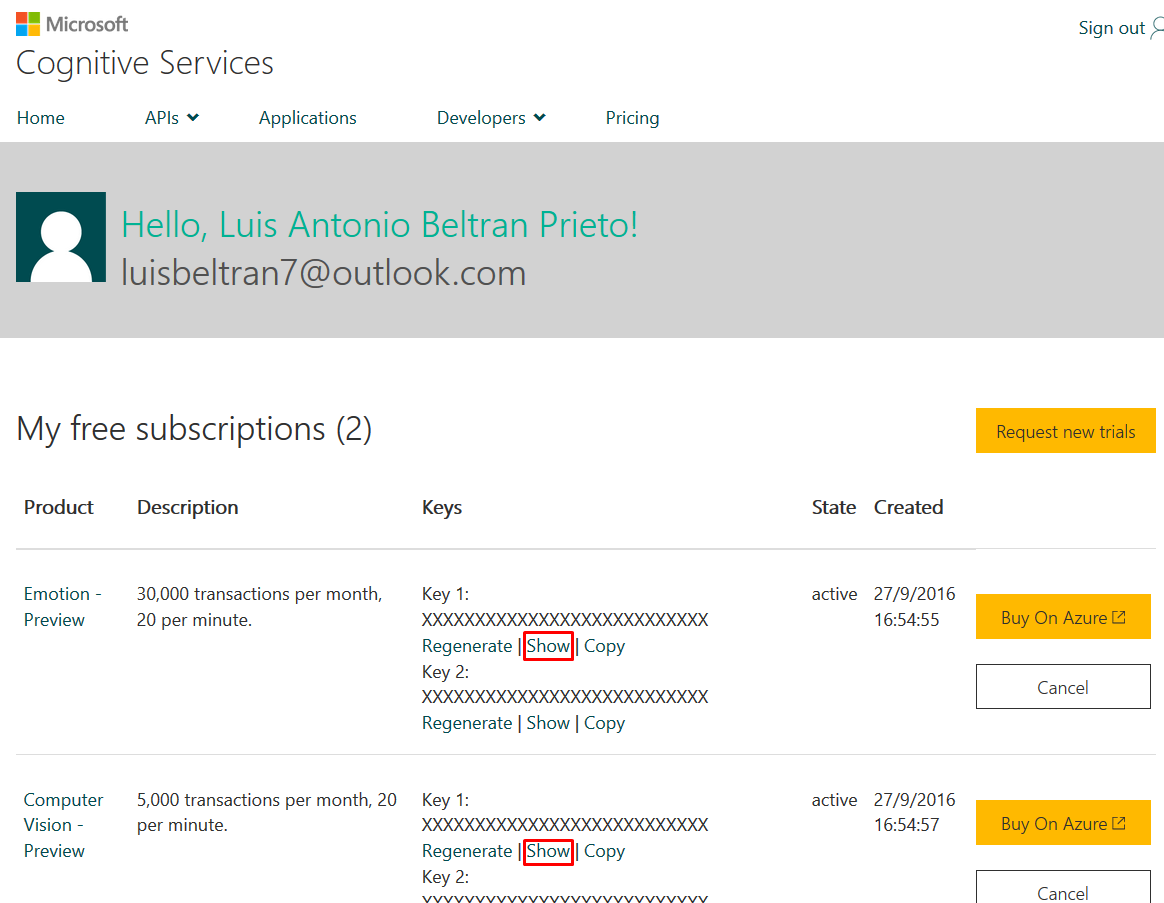


**Paso 3.** Da clic en el botón **Request new trials** para poderte suscribir a los servicios que desees (todos ofrecen planes gratuitos). Para esta práctica utilizaremos los servicios de **Emotions** y **ComputerVision**, por lo que al menos deberás seleccionar estos 2. Siempre puedes volver a tu cuenta y dar clic en el botón Request new trials para seleccionar más servicios que no hayas elegido inicialmente.

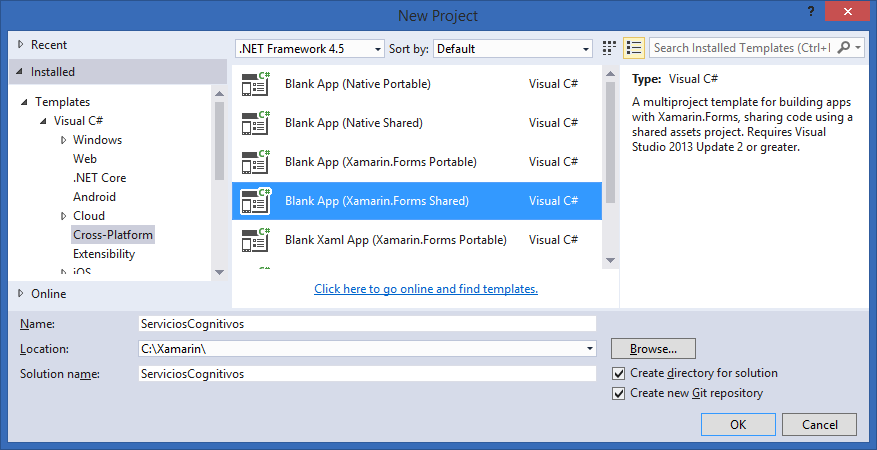




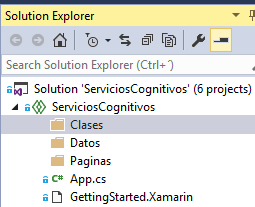
**Paso 4.** Obtén las llaves de los servicios de Emotions y ComputerVision dando clic en el botón **Show** en cada servicio. Copia estas llaves y pégalas en el bloc de notas, las utilizaremos más adelante.



**Paso 5.** Inicia Visual Studio 2015 y crea un nuevo proyecto de tipo **Xamarin.Forms Shared** llamado **ServiciosCognitivos**:



**Paso 6.** Agrega 3 carpetas en el proyecto **ServiciosCognitivos** con los nombres **Datos**, **Clases** y **Paginas** (da clic derecho en el nombre del proyecto y selecciona **Agregar** 🡪 **Nueva carpeta**):



**Paso 7.** Este paso lo tienes que repetir por cada proyecto que quieras probar: Vamos a agregar 3 paquetes **Nuget**. Para ello, da clic derecho en el nombre del proyecto (Android, iOS, UWP, Windows, Windows Phone) y selecciona la opción **Manage Nuget packages**. Los nombres de los paquetes a agregar son:

* Xam.Plugin.Media
* Microsoft.Bcl.Build
* Microsoft.ProjectOxford.Emotion
* Microsoft.ProjectOxford.Vision

**Paso 8.** En este paso realizaremos tareas de configuración de los plugins instalados en el paso 7 para el proyecto de **Android** (si no estás probando el proyecto de **Android**, continúa al siguiente paso).

1. Abre MainActivity.cs
2. Agrega el espacio de nombres **Plugin.Permissions**:

using Plugin.Permissions;

1. Agrega el siguiente código antes del método OnCreate

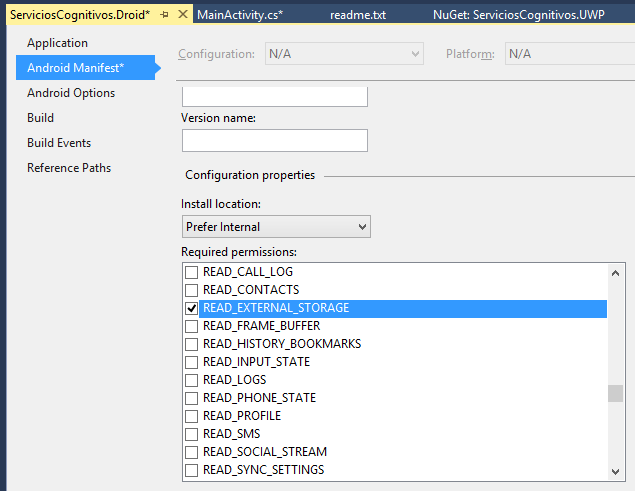
public override void OnRequestPermissionsResult(int requestCode, string[] permissions, Permission[] grantResults)

{

PermissionsImplementation.Current.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

1. Da doble clic sobre **Properties** del proyecto de Android.
2. Selecciona **Android Manifest**.
3. Marca los permisos **READ\_EXTERNAL\_STORAGE** y **WRITE\_EXTERNAL\_STORAGE**.



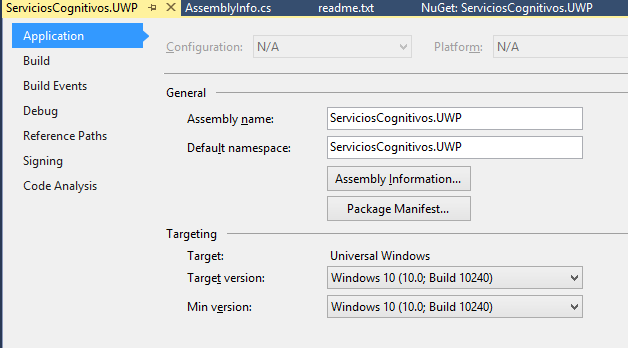
1. Guarda y cierra el archivo
2. Abre el archivo **AssemblyInfo.cs** (expande Properties en el Explorador de Soluciones)
3. Agrega las siguientes líneas de código al final del archivo:

[assembly: UsesFeature("android.hardware.camera", Required = false)]

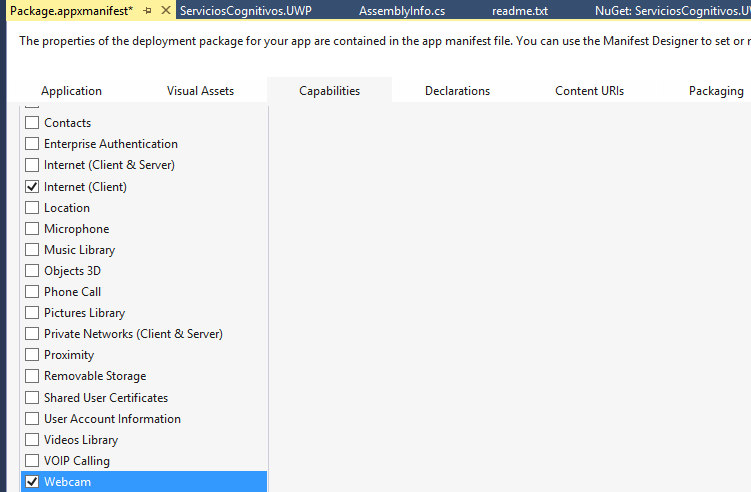
[assembly: UsesFeature("android.hardware.camera.autofocus", Required = false)]

**Paso 9.** En este paso realizaremos tareas de configuración de los plugins instalados en el paso 7 para el proyecto de **UWP** (si no estás probando el proyecto de **UWP**, continúa al siguiente paso).

1. Da doble clic sobre **Properties** del proyecto de UWP.
2. Da clic en el botón **Package Manifest**.



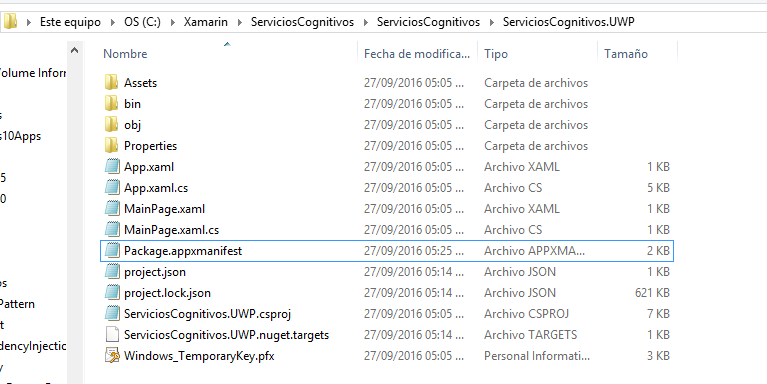
1. En Capabilities del proyecto, selecciona Webcam



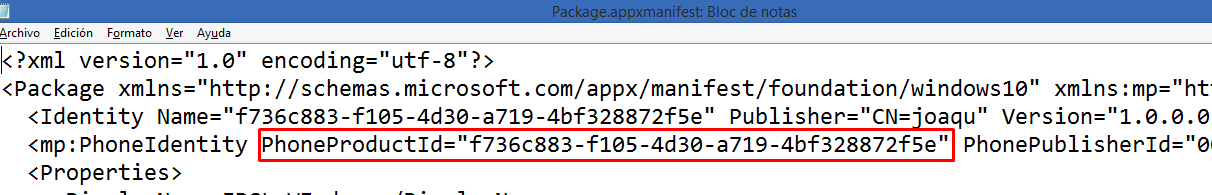
1. Guarda y cierra el archivo.

Los pasos que a continuación se presentan se muestran en caso de que la compilación más adelante falle. Son pasos que en mi caso fueron necesarios para ejecutar la app correctamente. Desconozco el motivo por el que la compilación no funcionó de inmediato.

1. Abre el Explorador de Archivos y localiza el archivo Package.appxmanifest del proyecto de UWP. Ábrelo.



1. Casi al inicio del contenido, identifica la propiedad PhoneProductId.



1. Modifica cualquier elemento de la cadena hexadecimal por un valor válido, por ejemplo cambia el último carácter “e” por un “7”.
2. Guarda y cierra el archivo

**Paso 10.** Comienza la programación. En la carpeta **Clases**, agrega una clase llamada **ServicioImagenes** dando clic derecho sobre la carpeta **Clases** y seleccionando Agregar 🡪 Nuevo elemento. El código completo de la clase es:

using Plugin.Media;

using Plugin.Media.Abstractions;

using System.Threading.Tasks;

namespace ServiciosCognitivos.Clases

{

public class ServicioImagenes

{

public static async Task<MediaFile> TomarFoto(bool usarCamara)

{

await CrossMedia.Current.Initialize();

if (usarCamara)

{

if (!CrossMedia.Current.IsCameraAvailable || !CrossMedia.Current.IsTakePhotoSupported)

{

return null;

}

}

var file = usarCamara

? await CrossMedia.Current.TakePhotoAsync(new StoreCameraMediaOptions

{

Directory = "Cognitive",

Name = "test.jpg"

})

: await CrossMedia.Current.PickPhotoAsync();

return file;

}

}

}

Básicamente se expone un método que decide si la imagen se tomará de la cámara o se seleccionará de la librería de Imágenes del dispositivo. La imagen obtenida se almacena temporalmente y se retorna la referencia.

**Paso 11.** Crea una segunda clase en esta misma carpeta. El nombre es **ServicioEmociones** y el código se muestra a continuación: **(NOTA: Debes reemplazar LLAVE-EMOCIONES) por la llave del servicio Emotions del paso 4).**

using Microsoft.ProjectOxford.Emotion;

using System.IO;

using System.Linq;

using System.Threading.Tasks;

using System.Collections.Generic;

namespace ServiciosCognitivos.Clases

{

public class ServicioEmociones

{

static string key = "LLAVE-EMOCIONES";

public static async Task<Dictionary<string, float>> ObtenerEmociones(Stream stream)

{

EmotionServiceClient cliente = new EmotionServiceClient(key);

var emociones = await cliente.RecognizeAsync(stream);

if (emociones == null || emociones.Count() == 0)

return null;

return emociones[0].Scores.ToRankedList().ToDictionary(x => x.Key, x => x.Value);

}

}

}

El método anterior hace una consulta al servicio Emotions, le pasa un Stream de la imagen que se desea analizar por el servicio, y se devuelve precisamente el resultado del análisis en forma de diccionario. Cada elemento es una emoción y su puntuación (score).

**Paso 12.** Finalmente, crea una clase llamada ServicioComputerVision en la misma carpeta y coloca el código que se muestra a continuación: **(NOTA: Debes reemplazar LLAVE-COMPUTERVISION) por la llave del servicio Emotions del paso 4)**

using System.IO;

using System.Linq;

using Microsoft.ProjectOxford.Vision;

using System.Threading.Tasks;

using Microsoft.ProjectOxford.Vision.Contract;

namespace ServiciosCognitivos.Clases

{

public class ServicioComputerVision

{

static string key = "LLAVE-COMPUTERVISION";

public static async Task<AnalysisResult> ObtenerDescripcionImagen(Stream imageStream)

{

VisionServiceClient cliente = new VisionServiceClient(key);

VisualFeature[] features =

{

VisualFeature.Tags,

VisualFeature.Categories,

VisualFeature.Description,

VisualFeature.Adult,

VisualFeature.ImageType,

VisualFeature.Color,

VisualFeature.Faces

};

return await cliente.AnalyzeImageAsync(imageStream, features.ToList(), null);

}

}

}

El método anterior hace una consulta al servicio ComputerVision, le pasa un Stream de la imagen que se desea analizar por el servicio, y se devuelve precisamente el resultado del análisis en un objeto AnalysisResult, que contiene gran cantidad de información que se mostrará en una página.

**Paso 13.** En la carpeta **Paginas**, agrega un **Forms Xaml Page** llamado **PaginaAnalisisEmociones**. A continuación se muestra el código XAML, seguido del código C# de la página:

**XAML**

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="ServiciosCognitivos.Paginas.PaginaAnalisisEmociones">

<StackLayout Padding="0">

<Image x:Name="imgFoto" WidthRequest="150" HeightRequest="150" Aspect="AspectFit"/>

<Button x:Name="btnCamara" x:Id="btnCamara" Text="Usar cámara" Clicked="btnFoto\_Clicked"/>

<Button x:Name="btnGaleria" x:Id="btnGaleria" Text="Seleccionar de la galería" Clicked="btnFoto\_Clicked"/>

<Button x:Name="btnAnalizarEmociones" Text="Analizar emociones" Clicked="btnAnalizarEmociones\_Clicked"/>

<Label x:Name="lblResultado" Text="---" />

<StackLayout x:Name="panelResultados"/>

</StackLayout>

</ContentPage>

**C#**

using System;

using System.Collections.Generic;

using Xamarin.Forms;

using System.IO;

using ServiciosCognitivos.Clases;

namespace ServiciosCognitivos.Paginas

{

public partial class PaginaAnalisisEmociones : ContentPage

{

static Stream streamCopy;

public PaginaAnalisisEmociones()

{

InitializeComponent();

}

async void btnFoto\_Clicked(object sender, EventArgs e)

{

var usarCamara = ((Button)sender).Text.Contains("cámara");

var file = await ServicioImagenes.TomarFoto(usarCamara);

panelResultados.Children.Clear();

lblResultado.Text = "---";

imgFoto.Source = ImageSource.FromStream(() => {

var stream = file.GetStream();

streamCopy = new MemoryStream();

stream.CopyTo(streamCopy);

stream.Seek(0, SeekOrigin.Begin);

file.Dispose();

return stream;

});

}

async void btnAnalizarEmociones\_Clicked(object sender, EventArgs e)

{

if (streamCopy != null)

{

streamCopy.Seek(0, SeekOrigin.Begin);

var emociones = await ServicioEmociones.ObtenerEmociones(streamCopy);

if (emociones != null)

{

lblResultado.Text = "---Análisis de Emociones---";

DibujarResultados(emociones);

}

else lblResultado.Text = "---No se detectó una cara---";

}

else lblResultado.Text = "---No has seleccionado una imagen---";

}

void DibujarResultados(Dictionary<string, float> emociones)

{

panelResultados.Children.Clear();

foreach (var emocion in emociones)

{

Label lblEmocion = new Label() {

Text = emocion.Key,

TextColor = Color.Blue,

WidthRequest = 90

};

BoxView box = new BoxView(){

Color = Color.Lime,

WidthRequest = 150 \* emocion.Value,

HeightRequest = 30,

HorizontalOptions = LayoutOptions.StartAndExpand

};

Label lblPorcentaje = new Label(){

Text = emocion.Value.ToString("P4"),

TextColor = Color.Maroon

};

StackLayout panel = new StackLayout(){

Orientation = StackOrientation.Horizontal

};

panel.Children.Add(lblEmocion);

panel.Children.Add(box);

panel.Children.Add(lblPorcentaje);

panelResultados.Children.Add(panel);

}

}

}

}

**Paso 14. A**grega otro **Forms Xaml Page** de nombre **PaginaAnalisisComputerVision** en la misma carpeta **Paginas**. A continuación se muestra el código XAML, seguido del código C# de la página:

**XAML**

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="ServiciosCognitivos.Paginas.PaginaAnalisisComputerVision">

<StackLayout VerticalOptions="FillAndExpand">

<Image x:Name="imgFoto" WidthRequest="150" HeightRequest="150" Aspect="AspectFit"/>

<Button x:Name="btnCamara" x:Id="btnCamara" Text="Usar cámara" Clicked="btnFoto\_Clicked"/>

<Button x:Name="btnGaleria" x:Id="btnGaleria" Text="Seleccionar de la galería" Clicked="btnFoto\_Clicked"/>

<Button x:Name="btnAnalizarImagen" Text="Analizar imagen" Clicked="btnAnalizarImagen\_Clicked"/>

<Label x:Name="lblAdulto" Text="---" />

<Label x:Name="lblRacista" Text="---" />

<Label x:Name="lblCategorias" Text="---" />

<Label x:Name="lblColor" Text="---" />

<Label x:Name="lblTags" Text="---" />

<Label x:Name="lblCaptions" Text="---" />

<Label x:Name="lblCaras" Text="---" />

<Label x:Name="lblTags2" Text="---" />

</StackLayout>

</ContentPage>

**C#**

using System;

using Xamarin.Forms;

using System.IO;

using ServiciosCognitivos.Clases;

using System.Linq;

namespace ServiciosCognitivos.Paginas

{

public partial class PaginaAnalisisComputerVision : ContentPage

{

static Stream streamCopy;

async void btnFoto\_Clicked(object sender, EventArgs e)

{

var usarCamara = ((Button)sender).Text.Contains("cámara");

var file = await ServicioImagenes.TomarFoto(usarCamara);

imgFoto.Source = ImageSource.FromStream(() =>

{

var stream = file.GetStream();

streamCopy = new MemoryStream();

stream.CopyTo(streamCopy);

stream.Seek(0, SeekOrigin.Begin);

file.Dispose();

return stream;

});

}

public PaginaAnalisisComputerVision ()

{

InitializeComponent ();

}

async void btnAnalizarImagen\_Clicked(object sender, EventArgs e)

{

if (streamCopy != null)

{

streamCopy.Seek(0, SeekOrigin.Begin);

var vision = await ServicioComputerVision.ObtenerDescripcionImagen(streamCopy);

var adulto = vision.Adult;

lblAdulto.Text = String.Format("Contenido Adulto: {0} ({1})", adulto.IsAdultContent, adulto.AdultScore.ToString("P4"));

lblRacista.Text = String.Format("Contenido Racista: {0} ({1})", adulto.IsRacyContent, adulto.RacyScore.ToString("P4"));

var categorias = vision.Categories;

lblCategorias.Text = "Categorias: ";

categorias.ToList().ForEach(cat => lblCategorias.Text =

lblCategorias.Text + String.Format("{0} ({1}), ", cat.Name, cat.Score.ToString("P4")));

var color = vision.Color;

lblColor.Text = String.Format("Accent Color: {0}\nColor dominante:\nFondo: {1}\tFrente: {2}\n¿Es Blanco y Negro? {3}\nColores dominantes: ",

color.AccentColor, color.DominantColorBackground,

color.DominantColorForeground, color.IsBWImg);

color.DominantColors.ToList().ForEach(x => lblColor.Text = lblColor.Text + x + ", ");

var descripcion = vision.Description;

lblTags.Text = "Tags: ";

lblCaptions.Text = "Captions: ";

vision.Description.Tags.ToList().ForEach(tag => lblTags.Text = lblTags.Text + tag + ", ");

vision.Description.Captions.ToList().ForEach(cap => lblCaptions.Text = lblCaptions.Text + String.Format("{0} ({1}), ", cap.Text, cap.Confidence.ToString("P4")));

var caras = vision.Faces;

lblCaras.Text = "Caras: ";

caras.ToList().ForEach(cara => lblCaras.Text = lblCaras.Text + String.Format("{0} ({1}), ", cara.Gender, cara.Age));

var tags = vision.Tags;

lblTags2.Text = "Tags 2: ";

tags.ToList().ForEach(tag => lblTags2.Text = lblTags2.Text + String.Format("{0} - {1} ({2}), ", tag.Name, tag.Hint, tag.Confidence.ToString("P4")));

}

}

}

}

**Paso 15.** Agrega un tercer **Forms Xaml Page** llamado **PaginaNavegacion**, que permitirá acceder a las páginas creadas en los pasos anteriores. El código XAML y C# se presenta a continuación:

**XAML**

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="ServiciosCognitivos.Paginas.PaginaNavegacion">

<StackLayout>

<Button x:Name="btnEmociones" Text="Analizar emociones" Clicked="btnEmociones\_Clicked"/>

<Button x:Name="btnVision" Text="Analizar imagen" Clicked="btnVision\_Clicked"/>

</StackLayout>

</ContentPage>

**C#**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace ServiciosCognitivos.Paginas

{

public partial class PaginaNavegacion : ContentPage

{

public PaginaNavegacion ()

{

InitializeComponent ();

}

void btnEmociones\_Clicked(object sender, EventArgs e)

{

Navigation.PushAsync(new PaginaAnalisisEmociones());

}

void btnVision\_Clicked(object sender, EventArgs e)

{

Navigation.PushAsync(new PaginaAnalisisComputerVision());

}

}

}

**Paso 16.** Modifica App.cs para que la página de inicio se PaginaNavegacion

public App ()

{

// The root page of your application

MainPage = new NavigationPage(new Paginas.PaginaNavegacion());

}

**Paso 17.** Compila y ejecuta la aplicación, verificando su correcto funcionamiento en ambas plataformas.

**¡Felicidades! Has creado una aplicación móvil multiplataforma que analiza imágenes, detectando emociones y obteniendo información sobre su contenido. Todo esto es posible gracias a la inteligencia artificial contenida en los Cognitive Services.**