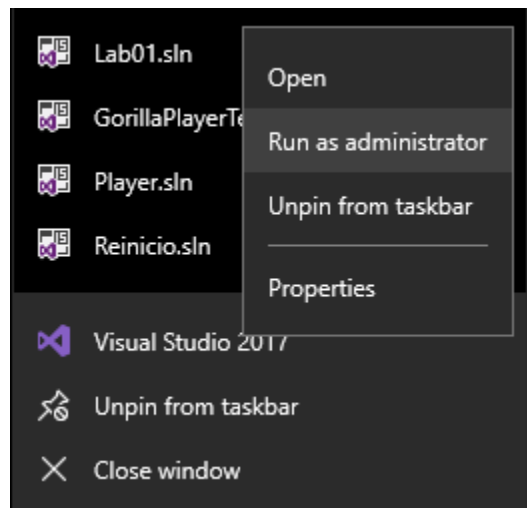


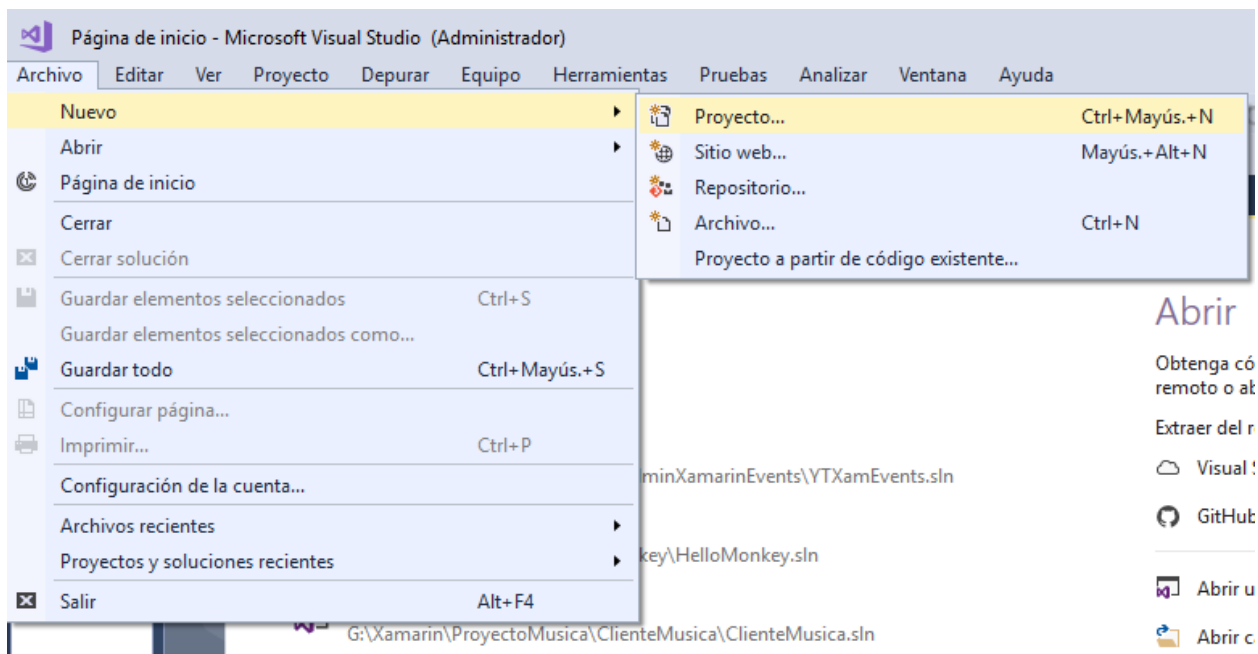
## **Práctica: Introducción a Xamarin – Autor: Luis Beltrán**

En esta sesión vamos a crear una sencilla aplicación móvil multiplataforma con Xamarin que se puede ejecutar en Android, iOS y Windows 10. La idea es demostrar el uso de varios Views (controles) y acceder a un servicio web que devuelva información, la cual será mostrada a continuación.

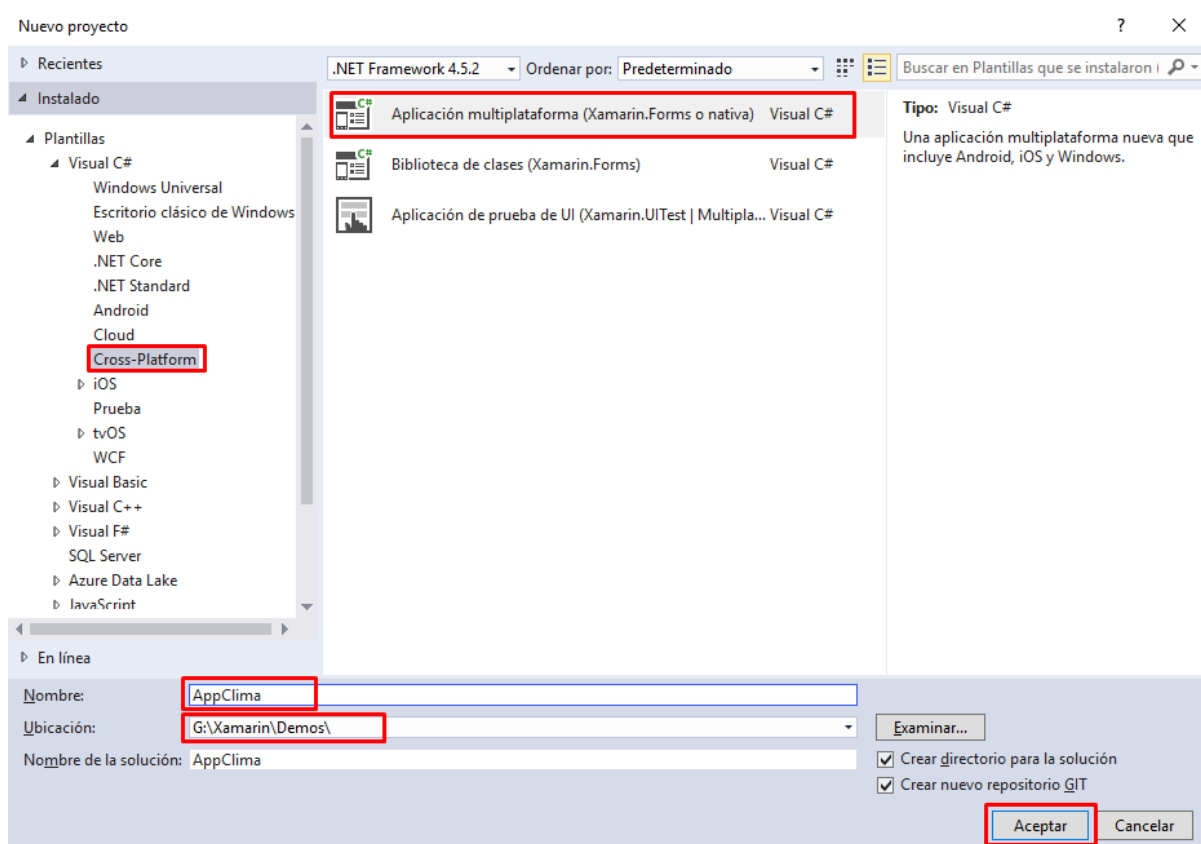
**Paso 1.** Abre Visual Studio (se recomienda hacerlo bajo el contexto de Administrador).



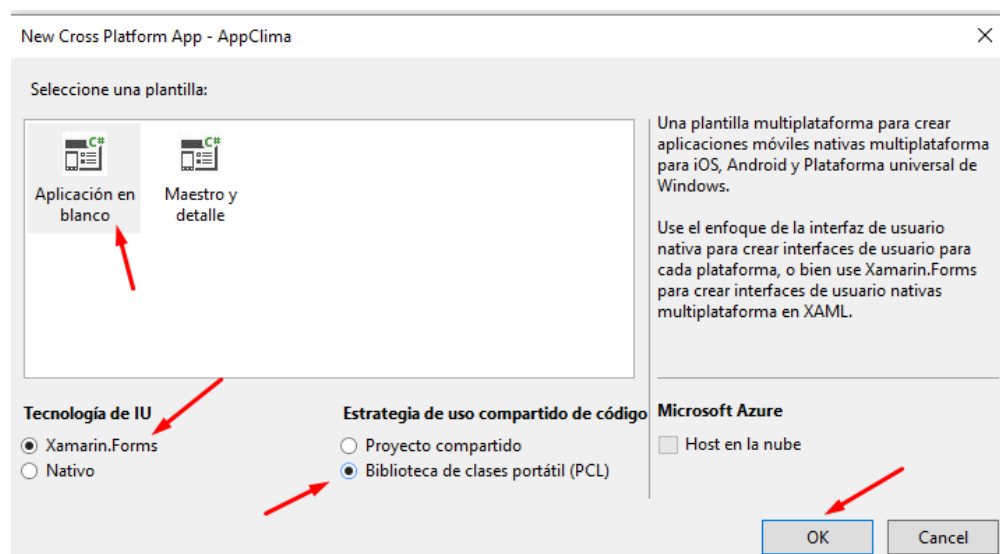
**Paso 2.** Da clic en **Nuevo → Proyecto**



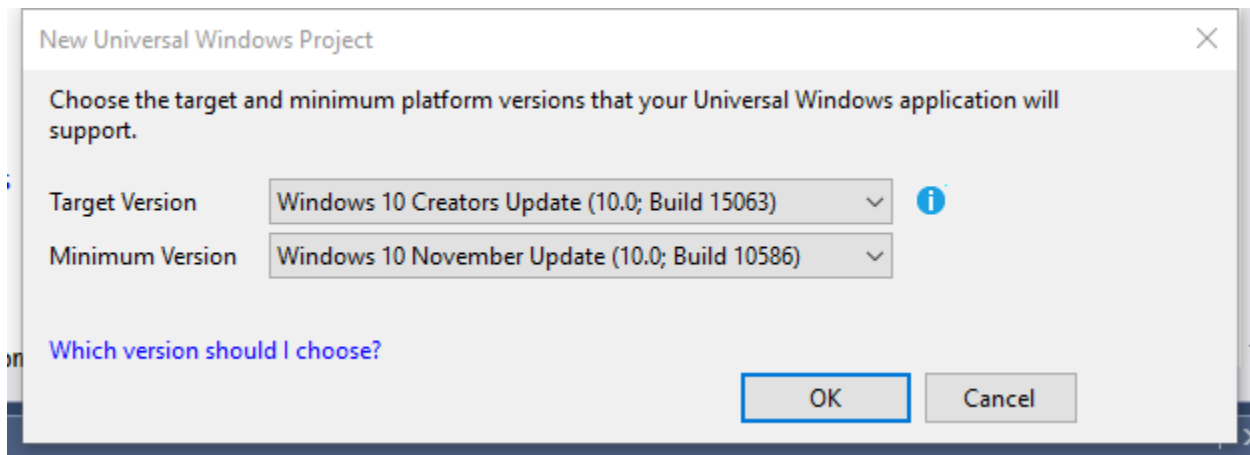
**Paso 3.** De la categoría **Cross-Platform** selecciona **Aplicación multiplataforma (Xamarin.Forms o nativa)** y coloca el nombre de proyecto **AppClima**. Además, la ruta del proyecto debe ser una ubicación corta para evitar problemas de ruta larga (un error muy común en Xamarin).



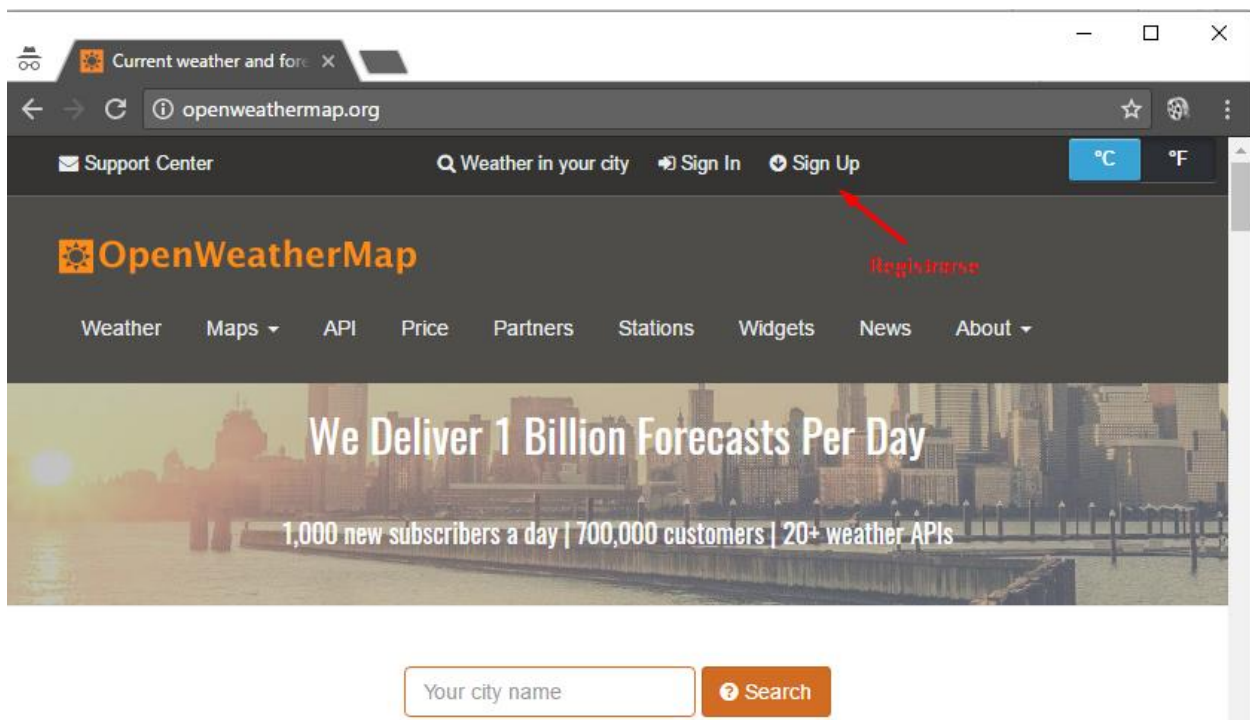
**Paso 4.** Selecciona la plantilla **Aplicación en blanco**, la tecnología de IU **Xamarin.Forms** y la estrategia de uso compartido de código **Biblioteca de clases portátil (PCL)**. Da clic en **OK**.



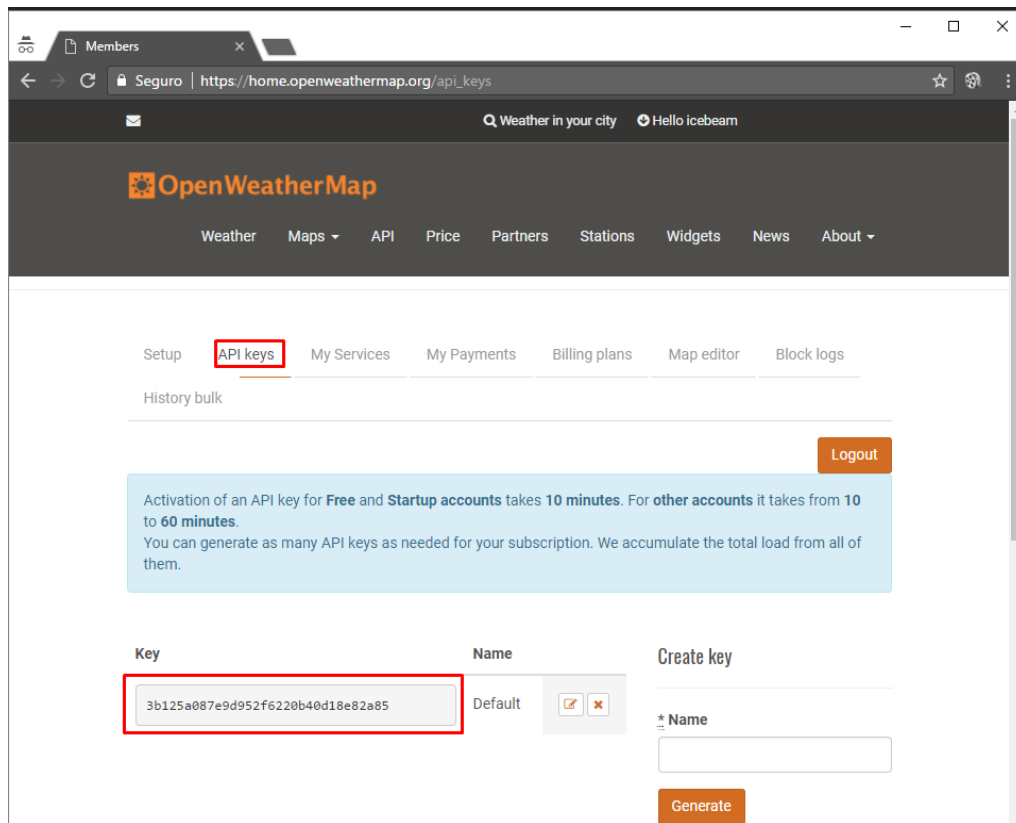
**Paso 5.** Si tienes instalado el SDK de Windows 10, aparecerá la ventana de selección del Target y Minimum Version. Selecciónalas a conveniencia, según la versión que tengas instalada.



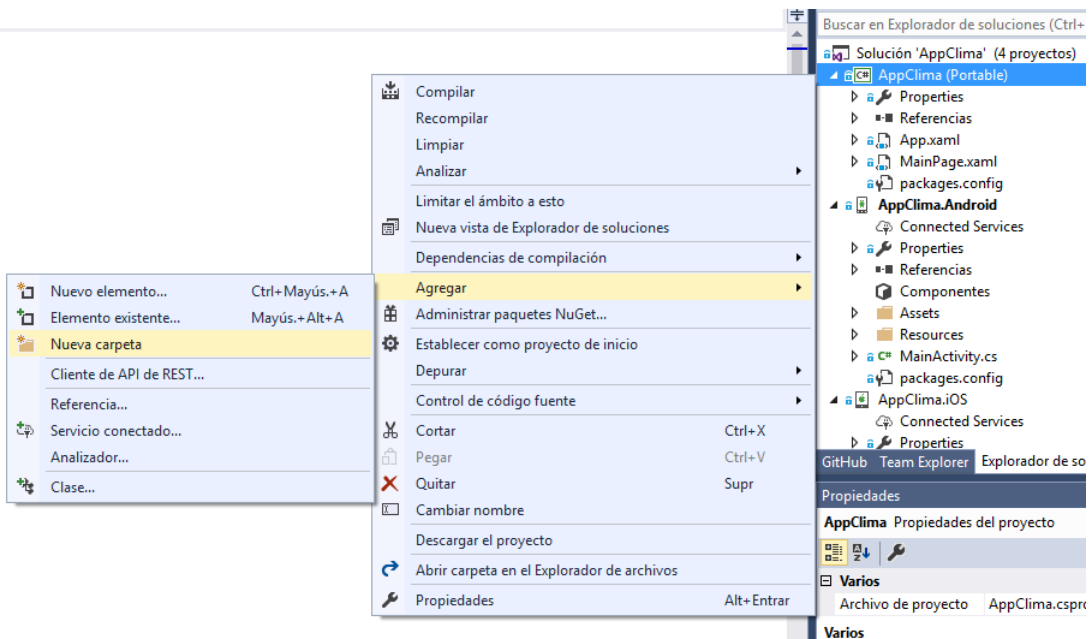
**Paso 6.** Mientras se genera la solución, regístrate en el sitio **openweathermap.org** y da clic en **Sign Up**.



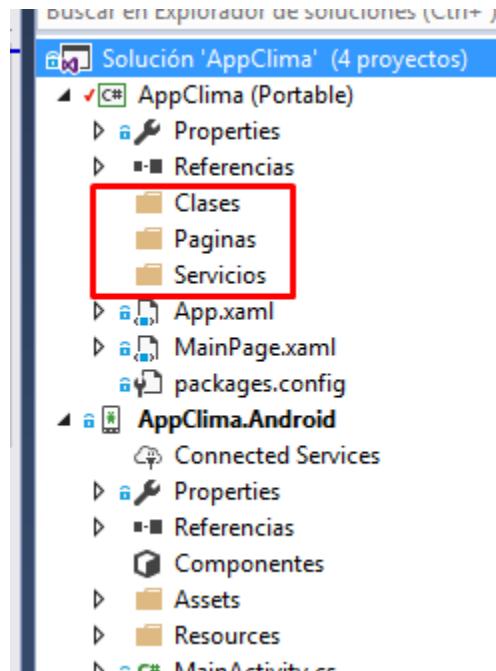
**Paso 7.** Una vez registrado, da clic en **API Keys** y copia el valor de la llave que aparece.



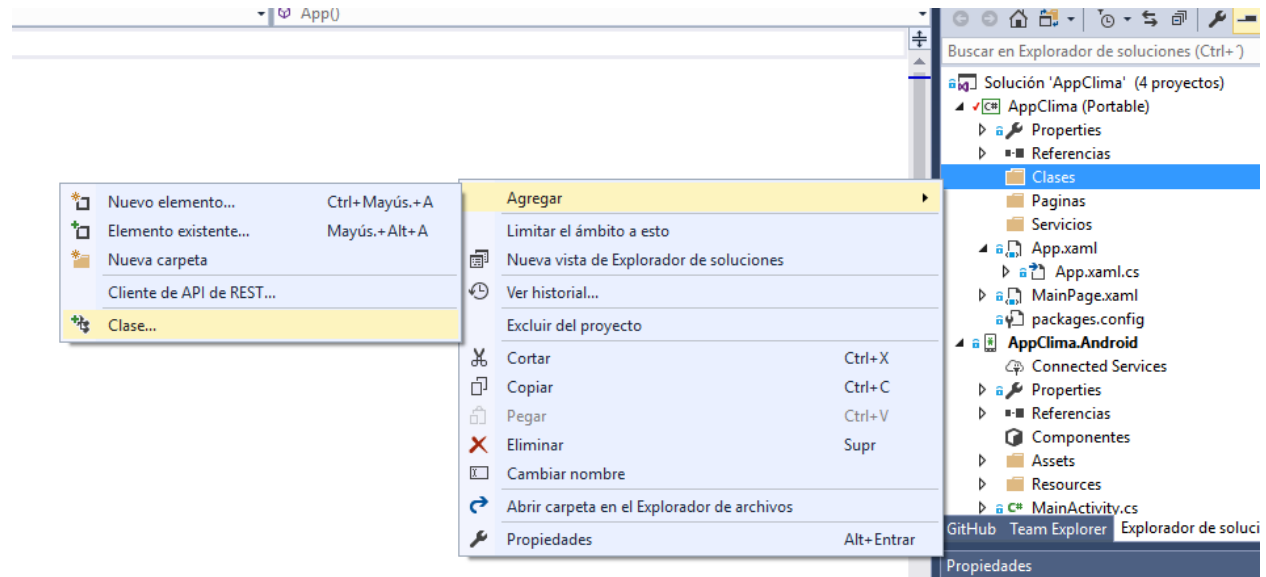
**Paso 8.** Regresando a Visual Studio, da clic derecho en el **proyecto portable** y selecciona **Agregar → Nueva carpeta**.



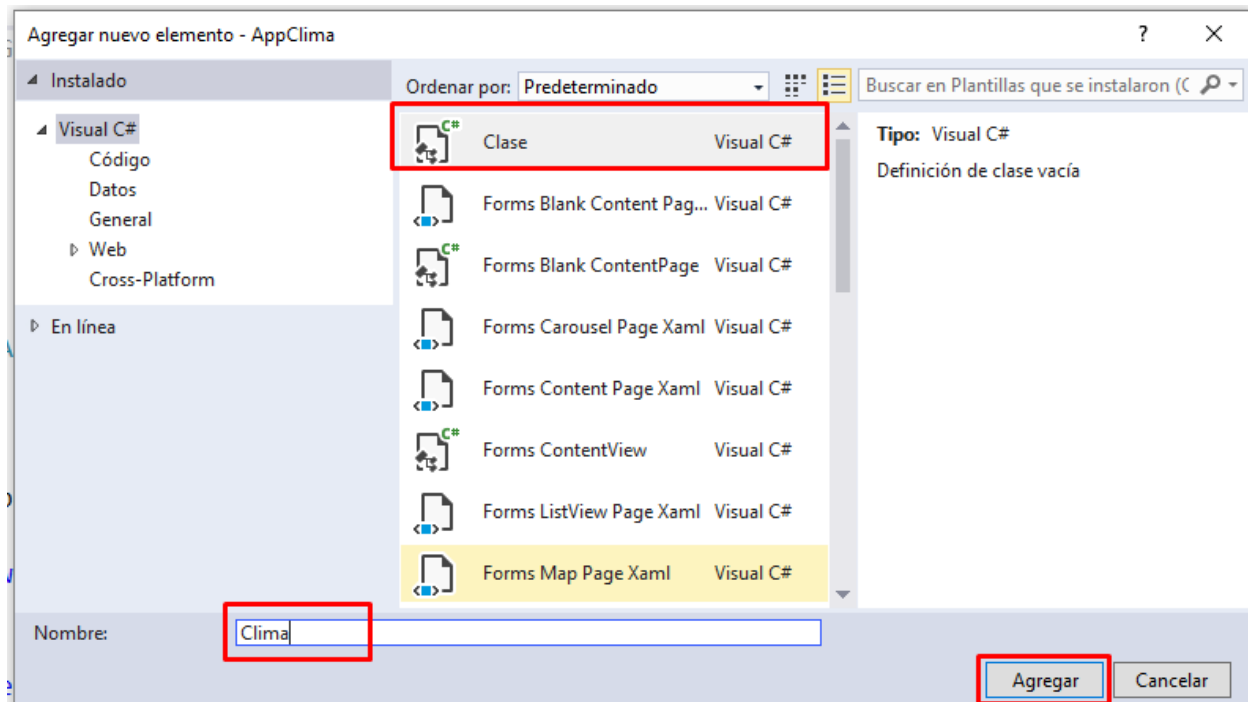
**Paso 9.** El nombre de la carpeta será **Clases**. Repite el proceso y crea 2 carpetas más: **Paginas** y **Servicios**.



**Paso 10.** Sobre la carpeta **Clases** da clic derecho y selecciona **Agregar → Clase**.



**Paso 11.** El nombre de esta clase es **Clima**.

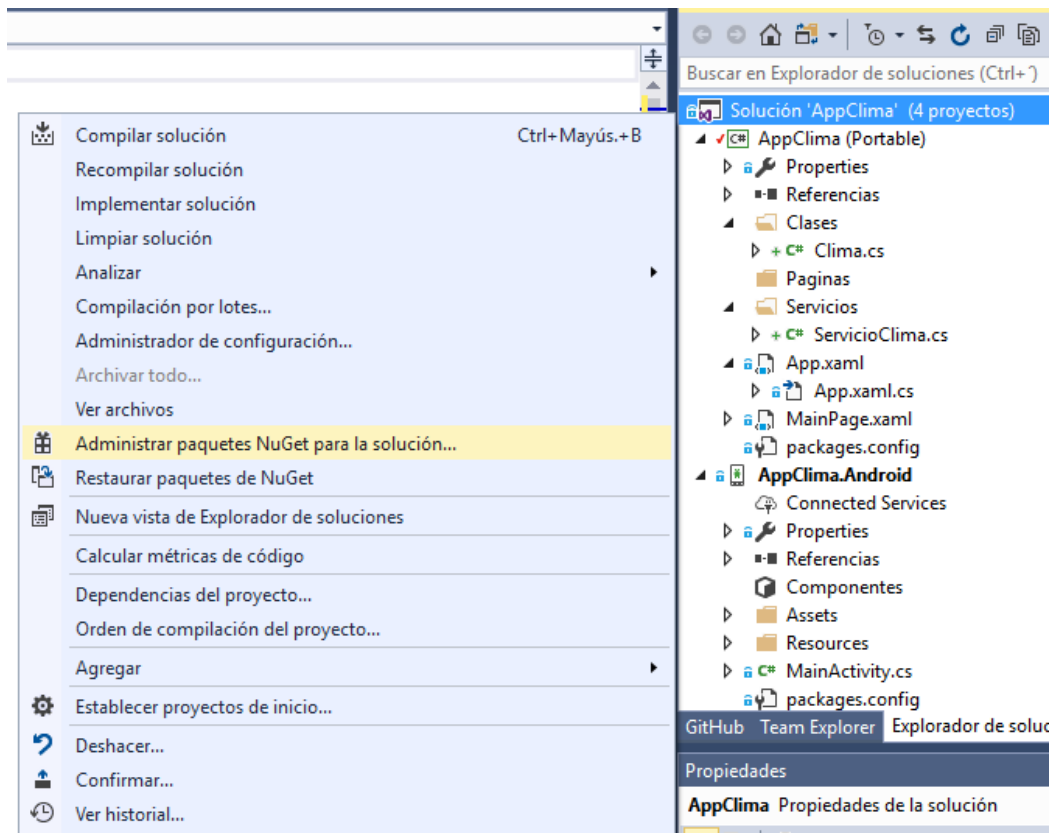


**Paso 12.** A continuación se presenta el código de esta clase. Como puedes observar, es un modelo que permitirá obtener datos de un servicio web. En el constructor se inicializa cada propiedad en cadena vacía.

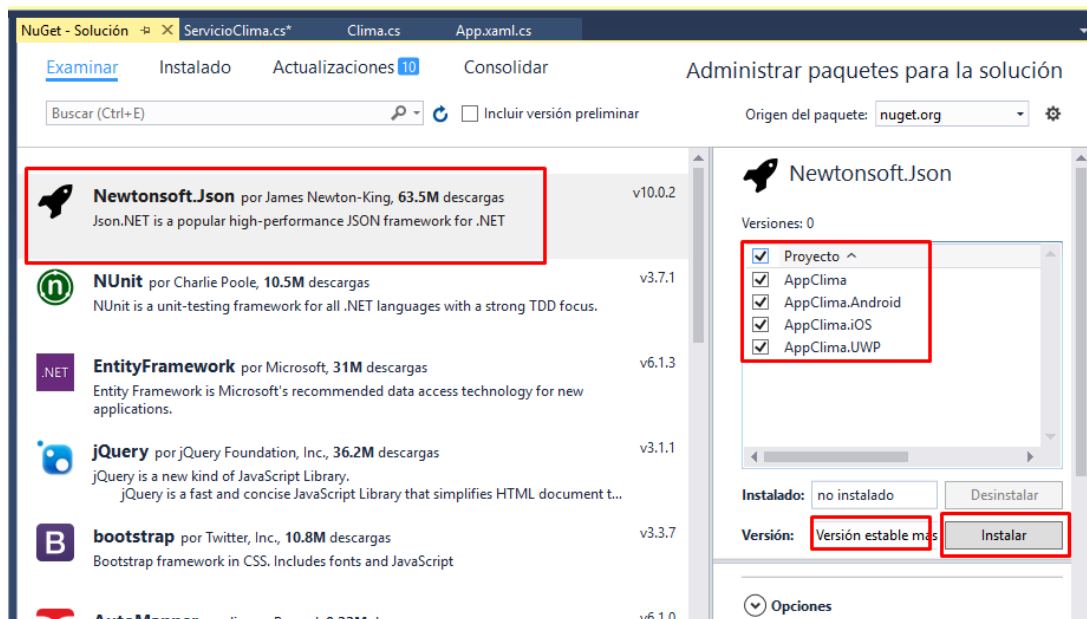
```
namespace AppClima.Clases
{
    public class Clima
    {
        public string Titulo { get; set; }
        public string Temperatura { get; set; }
        public string Viento { get; set; }
        public string Humedad { get; set; }
        public string Visibilidad { get; set; }
        public string Amanecer { get; set; }
        public string Ocaso { get; set; }

        public Clima()
        {
            this.Titulo = "";
            this.Temperatura = "";
            this.Viento = "";
            this.Humedad = "";
            this.Visibilidad = "";
            this.Amanecer = "";
            this.Ocaso = "";
        }
    }
}
```

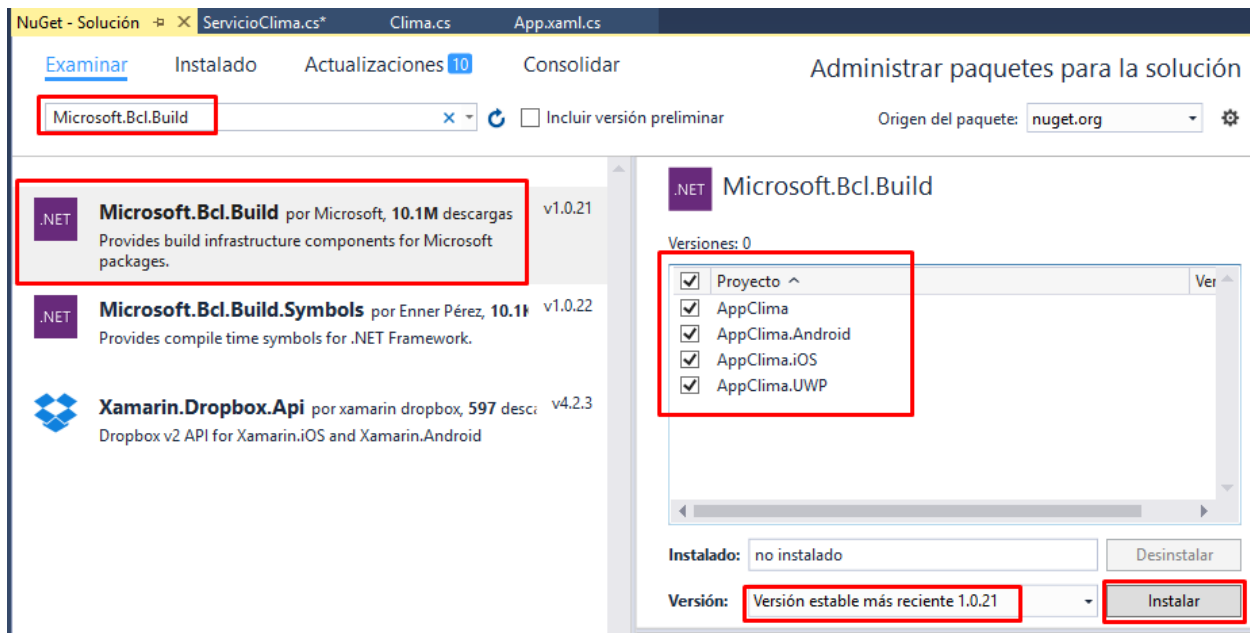
**Paso 13.** Da clic derecho en la solución y selecciona la opción **Administrar paquetes NuGet** para la solución...



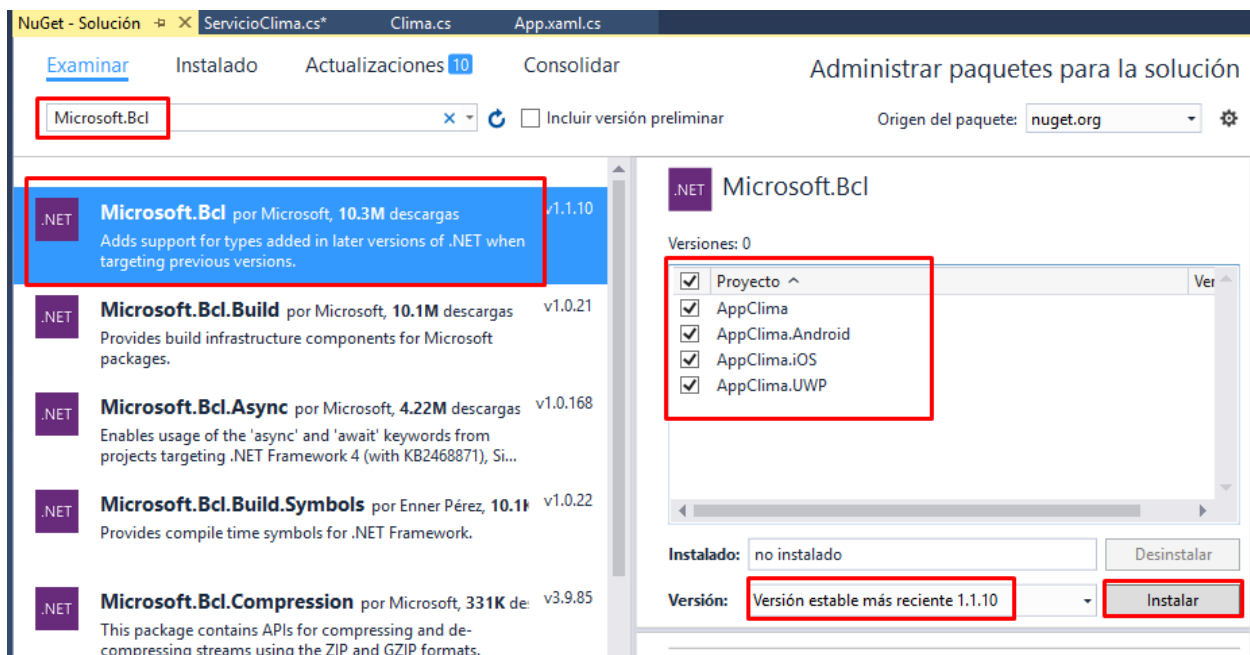
**Paso 14.** Busca el paquete **Newtonsoft.Json**, selecciona **todos los proyectos de tu solución**, elige una **versión estable** (no preview) y da clic en **Instalar**.



**Paso 15.** A continuación instala el paquete **Microsoft.Bcl.Build** nuevamente seleccionando todos los proyectos de tu solución y la última versión estable.

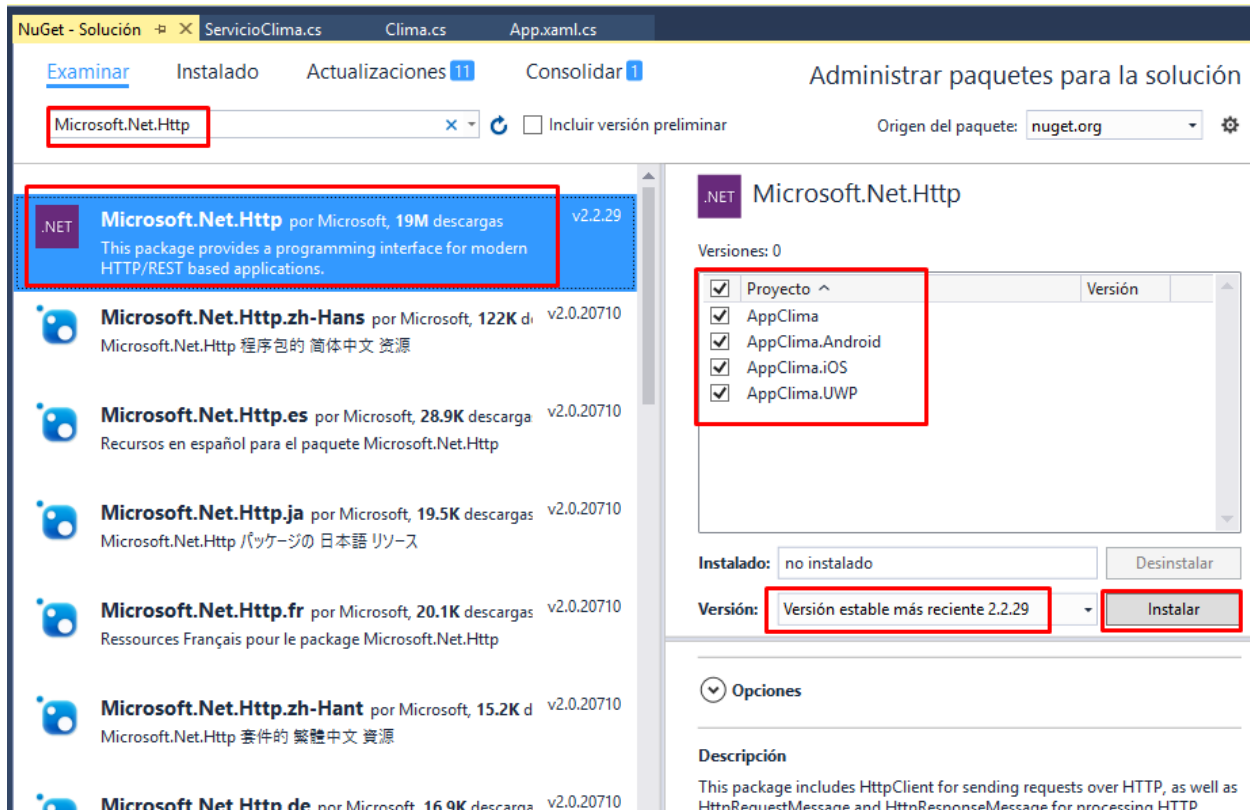


**Paso 16.** El siguiente paquete a instalar es **Microsoft.Bcl** en todos los proyectos de tu solución y la versión estable más reciente.

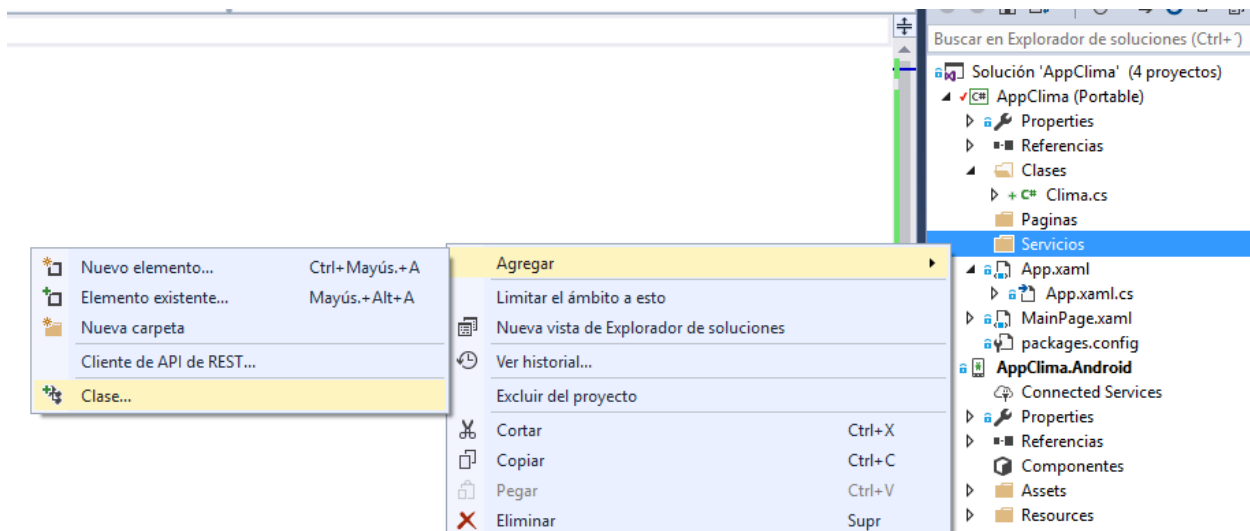




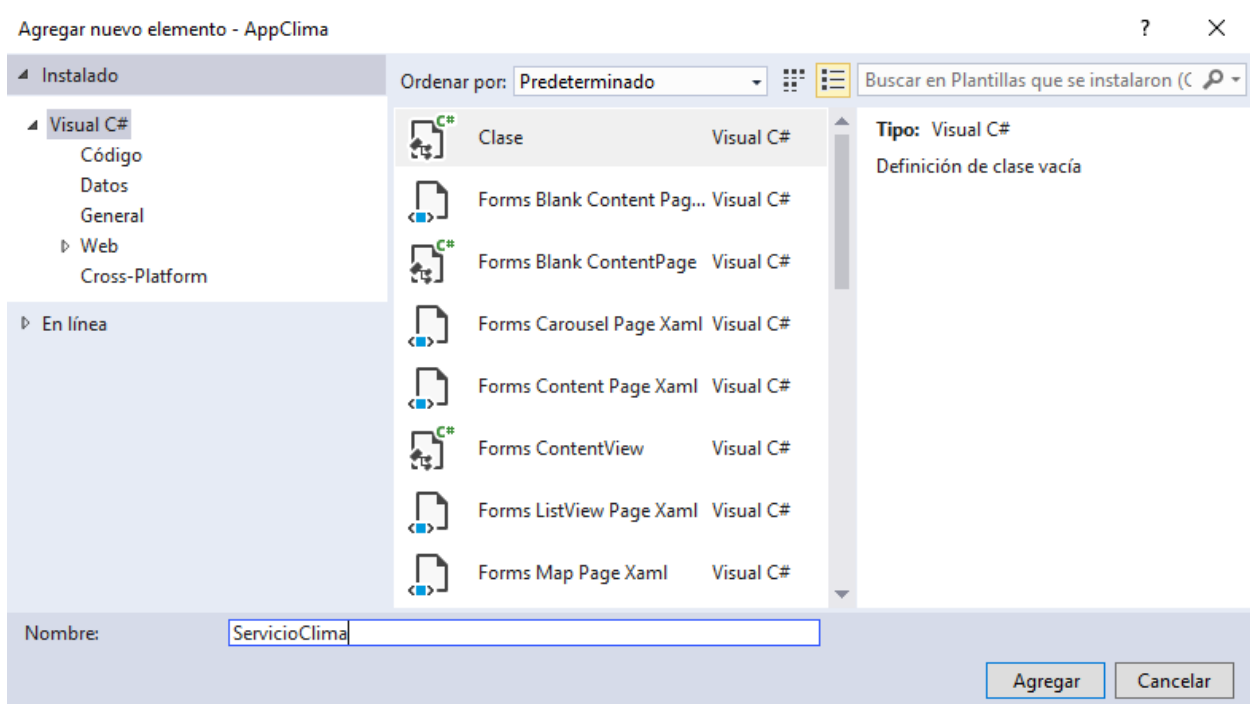
**Paso 17.** Finalmente, busca el paquete **Microsoft.Net.Http** e instálalo en todos los proyectos de tu solución, eligiendo la versión más reciente.



**Paso 18.** Ahora, en la carpeta **Servicios** agrega una nueva clase.



**Paso 19.** El nombre de esta clase es **ServicioClima**.



**Paso 20.** Y el código se muestra a continuación:

**NOTA:** Lo ideal es que utilices tu llave en la línea static string Key.

```
using System.Threading.Tasks;
using Newtonsoft.Json;
using System.Net.Http;
using Newtonsoft.Json.Linq;
using AppClima.Clases;
using System;

namespace AppClima.Servicios
{
    public static class ServicioClima
    {
        static string Key = "3b125a087e9d952f6220b40d18e82a85";

        public static async Task<Clima> ConsultarClima(string ciudad)
        {
            var conexion =
                $"http://api.openweathermap.org/data/2.5/weather?q={ciudad}&appid={Key}";

            using (var cliente = new HttpClient())
            {
                var petition = await cliente.GetAsync(conexion);

                if (petition != null)
                {
                    var json = petition.Content.ReadAsStringAsync().Result;
                }
            }
        }
    }
}
```

```

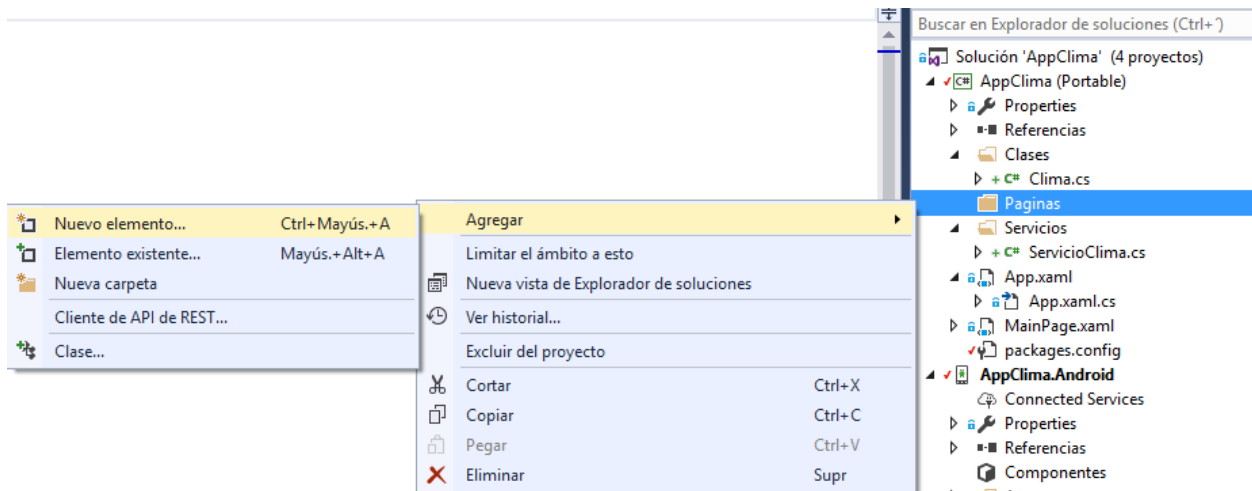
var datos = (JContainer)JsonConvert.DeserializeObject(json);

if (datos["weather"] != null)
{
    var clima = new Clima();
    clima.Titulo = (string)datos["name"];
    clima.Temperatura = ((float)datos["main"]["temp"] -
273.15).ToString("N2") + " °C";
    clima.Viento = (string)datos["wind"]["speed"] + " mph";
    clima.Humedad = (string)datos["main"]["humidity"] + " %";
    clima.Visibilidad = (string)datos["weather"][0]["main"];

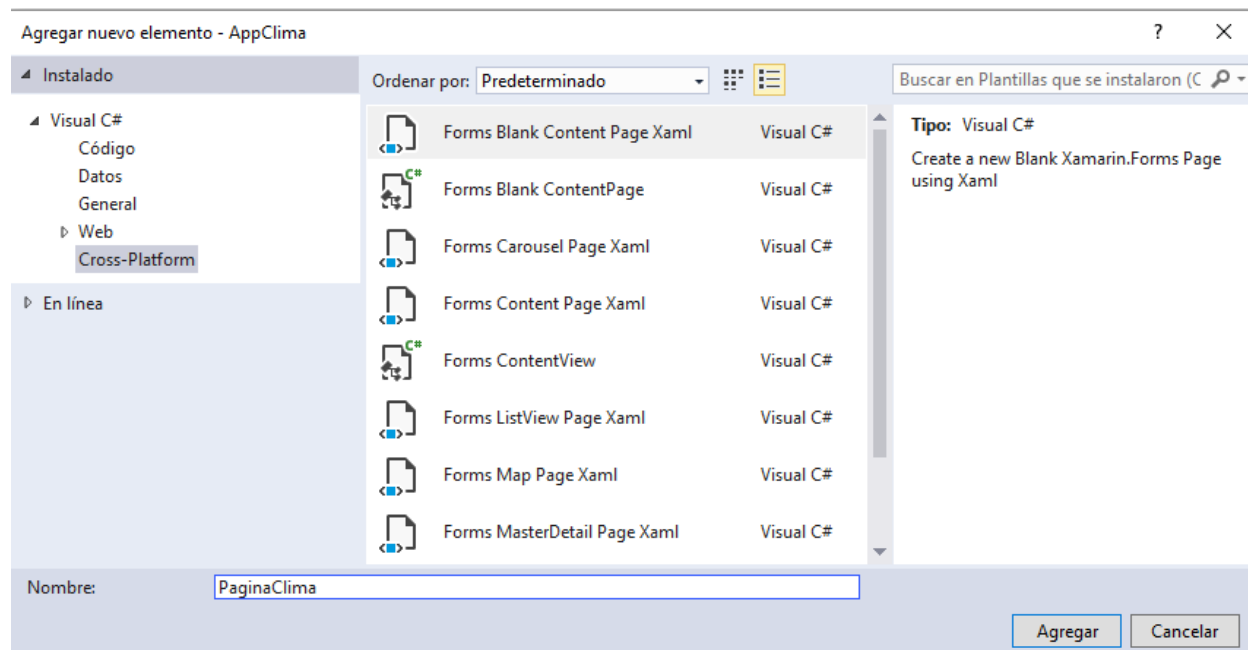
    var fechaBase = new DateTime(1970, 1, 1, 0, 0, 0, 0);
    var amanecer =
fechaBase.AddSeconds((double)datos["sys"]["sunrise"]);
    var ocaso = fechaBase.AddSeconds((double)datos["sys"]["sunset"]);
    clima.Amanecer = amanecer.ToString() + " UTC";
    clima.Ocaso = ocaso.ToString() + " UTC";
    return clima;
}
}
}
return default(Clima);
}
}
}
}
}

```

**Paso 21.** Ahora en la carpeta **Paginas** da clic derecho y selecciona **Agregar → Nuevo elemento**.



**Paso 22.** Selecciona **Forms Blank Content Page Xaml** de la categoría **Cross-Platform**. El nombre de este nuevo elemento es **PaginaClima**.



**Paso 23.** Una página de Xamarin.Forms consta de dos elementos: vista de diseño y vista de código. En la vista de diseño se agregan los controles que el usuario de la aplicación observará. El código es el siguiente:

```
<?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="AppClima.Paginas.PaginaClima">
    <ScrollView>
        <StackLayout BackgroundColor="#545454">
            <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
                <StackLayout Padding="10,10,10,10" HorizontalOptions="Start">
                    <Label Text="Clima de una ciudad" TextColor="White"
FontAttributes="Bold" FontSize="Medium" />
                    <Label Text="Ciudad" TextColor="#a8a8a8" FontSize="Small"/>
                    <Entry x:Name="txtCiudad" TextColor="White" BackgroundColor="Black" />
                </StackLayout>
                <StackLayout Padding="0,0,0,10" VerticalOptions="End">
                    <Button x:Name="btnBuscar" Text="Buscar clima" WidthRequest="185"
BorderWidth="1" Clicked="btnBuscar_Clicked"/>
                </StackLayout>
            </StackLayout>
            <StackLayout Padding="10,10,10,10" HorizontalOptions="Start">
                <Label Text="Ubicación" TextColor="#a8a8a8" FontSize="Small"/>
                <Label Text="{Binding Titulo}" TextColor="White" FontSize="Medium"/>
                <Label Text="Temperatura" TextColor="#a8a8a8" FontSize="Small"/>
                <Label Text="{Binding Temperatura}" TextColor="White" FontSize="Medium"/>
                <Label Text="Viento" TextColor="#a8a8a8" FontSize="Small"/>
                <Label Text="{Binding Viento}" TextColor="White" FontSize="Medium"/>
                <Label Text="Humedad" TextColor="#a8a8a8" FontSize="Small"/>
                <Label Text="{Binding Humedad}" TextColor="White" FontSize="Medium"/>
                <Label Text="Visibilidad" TextColor="#a8a8a8" FontSize="Small"/>
            </StackLayout>
        </ScrollView>
    </ContentPage>
```

```

        <Label Text="{Binding Visibilidad}" TextColor="White" FontSize="Medium"/>
        <Label Text="Amanecer" TextColor="#a8a8a8" FontSize="Small"/>
        <Label Text="{Binding Amanecer}" TextColor="White" FontSize="Medium"/>
        <Label Text="Ocaso" TextColor="#a8a8a8" FontSize="Small"/>
        <Label Text="{Binding Ocaso}" TextColor="White" FontSize="Medium"/>
    </StackLayout>
</StackLayout>
</ScrollView>
</ContentPage>

```

**Paso 24.** En la vista de código escribe lo siguiente:

```

using AppClima.Clases;
using AppClima.Servicios;
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace AppClima.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaClima : ContentPage
    {
        public PaginaClima()
        {
            InitializeComponent();

            this.BindingContext = new Clima();
        }

        private async void btnBuscar_Clicked(object sender, EventArgs e)
        {
            if (!String.IsNullOrEmpty(txtCiudad.Text))
            {
                var clima = await ServicioClima.ConsultarClima(txtCiudad.Text);

                if (clima != null)
                {
                    this.BindingContext = clima;
                    btnBuscar.Text = "Buscar de nuevo";
                }
            }
        }
    }
}

```

**Paso 25.** A continuación, localiza el constructor de la clase **App.xaml.cs** y modifica la instrucción de **MainPage** por la siguiente, que consiste en mostrar la página que acabamos de crear y codificar.

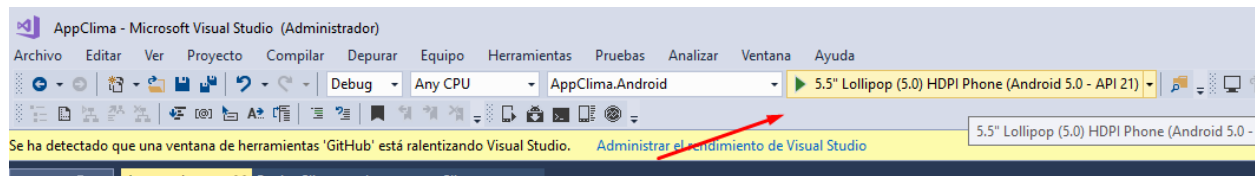
```

public App()
{
    InitializeComponent();

    MainPage = new AppClima.Paginas.PaginaClima();
}

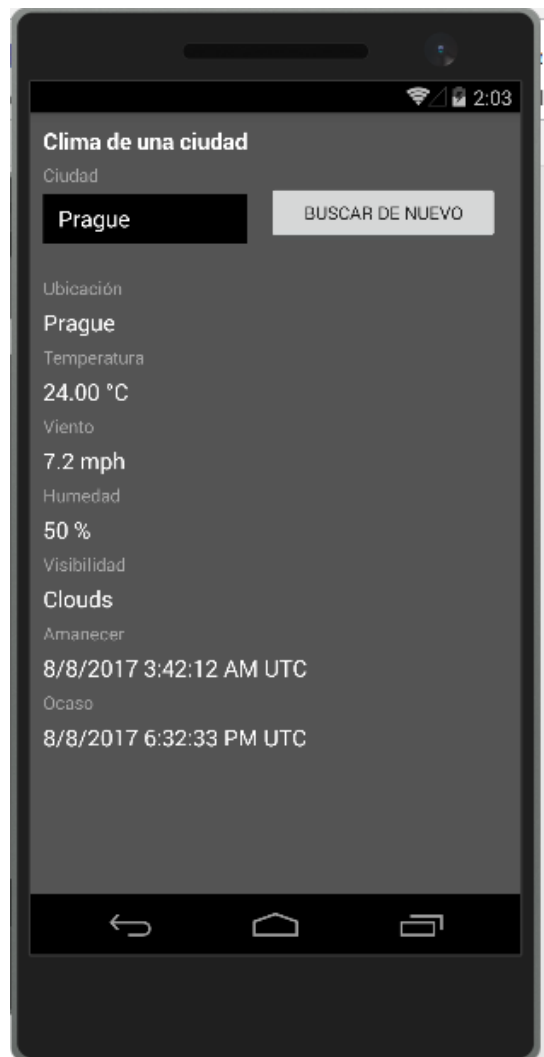
```

**Paso 26. ¡Listo!** ¡Ya podemos probar el funcionamiento de la aplicación! Comenzaremos con **Android**, para lo cual debes seleccionar tu emulador o dispositivo de la lista y dar clic para que se realice la compilación y ejecución de la app.

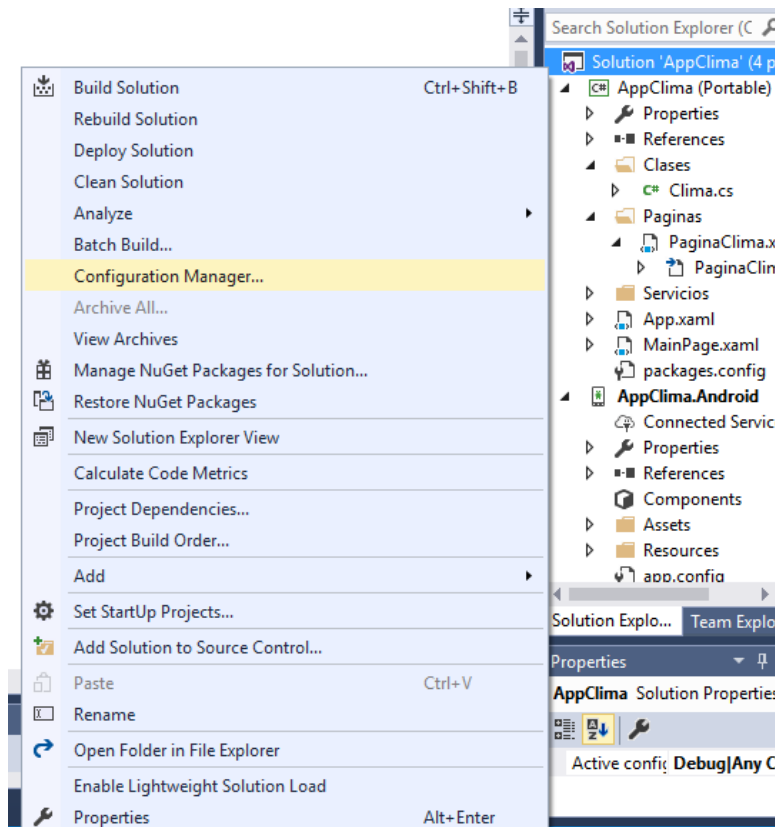


**Paso 27.** Se debe mostrar la aplicación en funcionamiento; en primer lugar, aparecerá **el diseño** de nuestra interfaz. Si escribes **el nombre de una ciudad** y das clic en **Buscar clima**, se deberá mostrar información del clima actual de la ciudad.

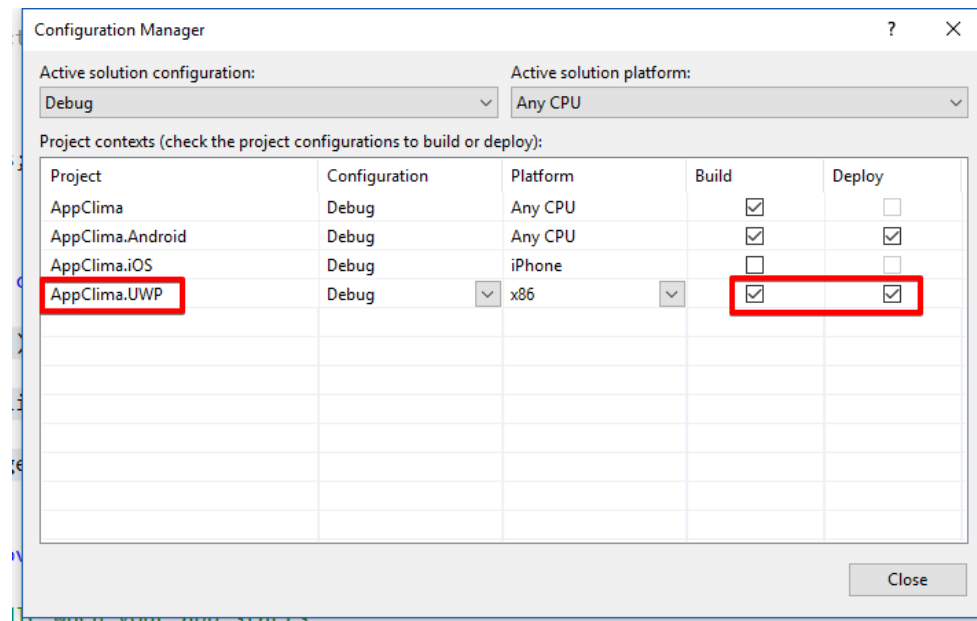




**Paso 28.** Si deseas probar tu aplicación en Windows 10, da clic derecho en la solución y selecciona **Administrador de configuración**.

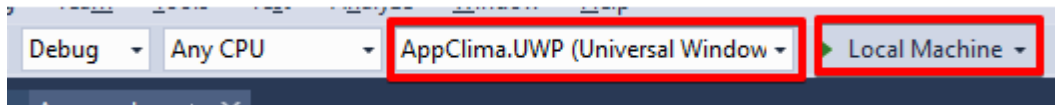


**Paso 29.** En el proyecto **UWP** marca las opciones **Build** y **Deploy**. Da clic en Cerrar.

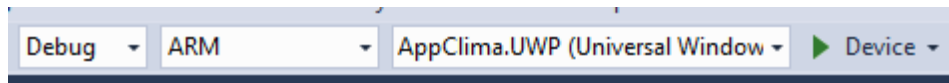




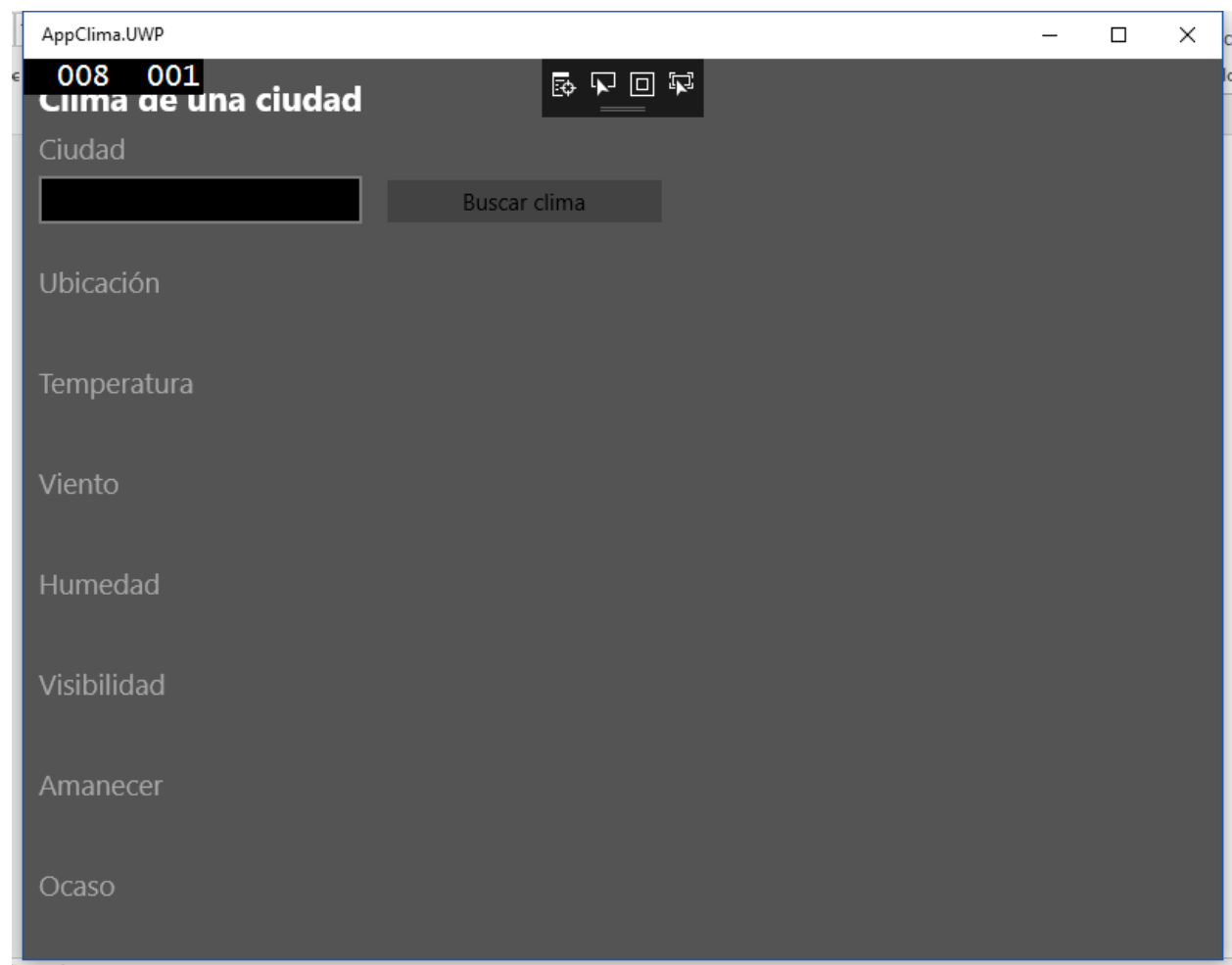
**Paso 30.** Selecciona **AppClima.UWP** de la lista de proyectos y elige **Local Machine** (para probar en la computadora), **un emulador** de la lista o **Device** (en caso de contar con un teléfono o tablet con Windows 10 instalado).

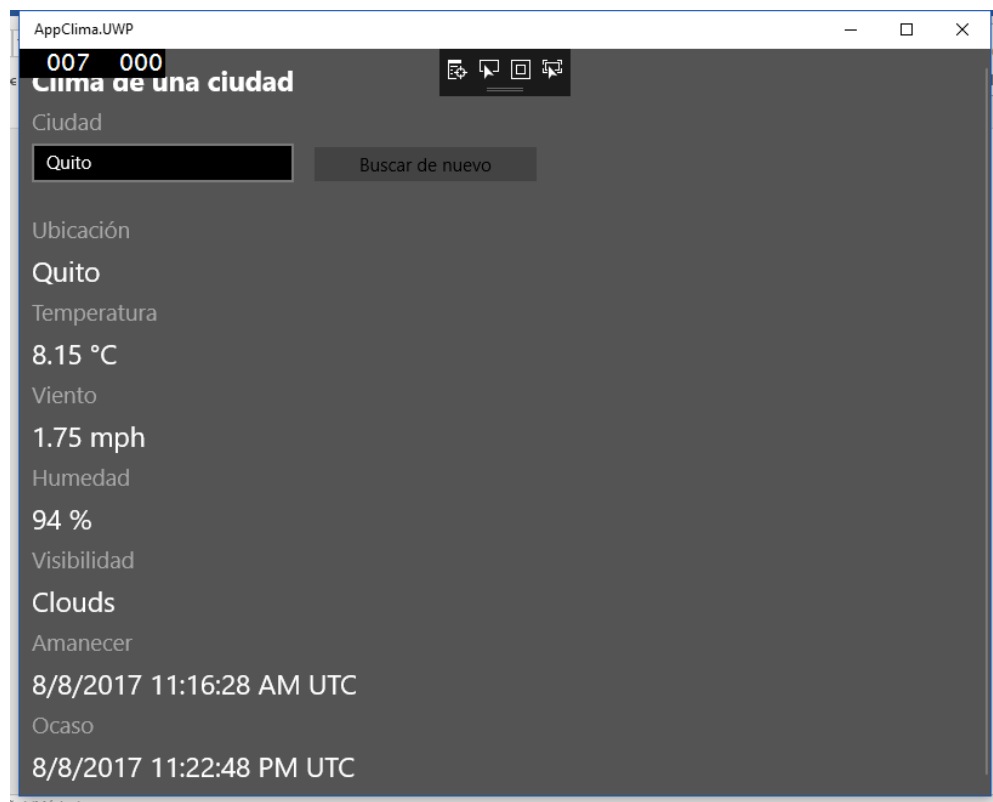
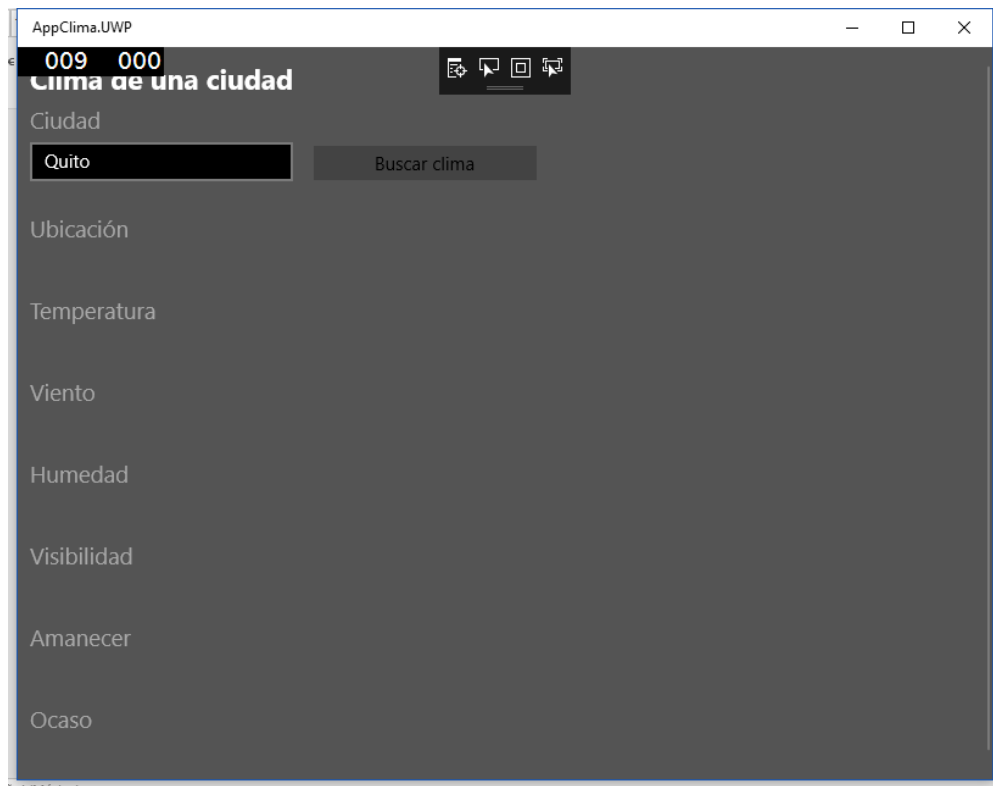


En el caso de un dispositivo con Windows 10, es probable que debas cambiar Any CPU por la arquitectura correspondiente, por ejemplo ARM:



**Paso 31.** A continuación se presentan las pruebas realizadas en una computadora con Windows 10 instalado.



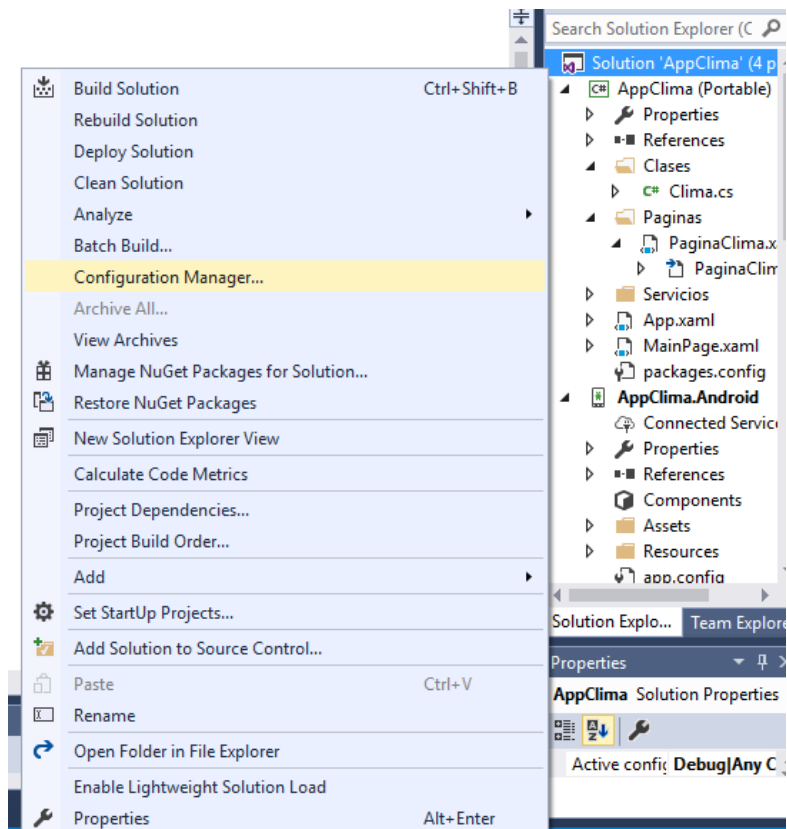


**Paso 32.** En el caso de un teléfono con Windows 10 instalado, éste es el resultado.

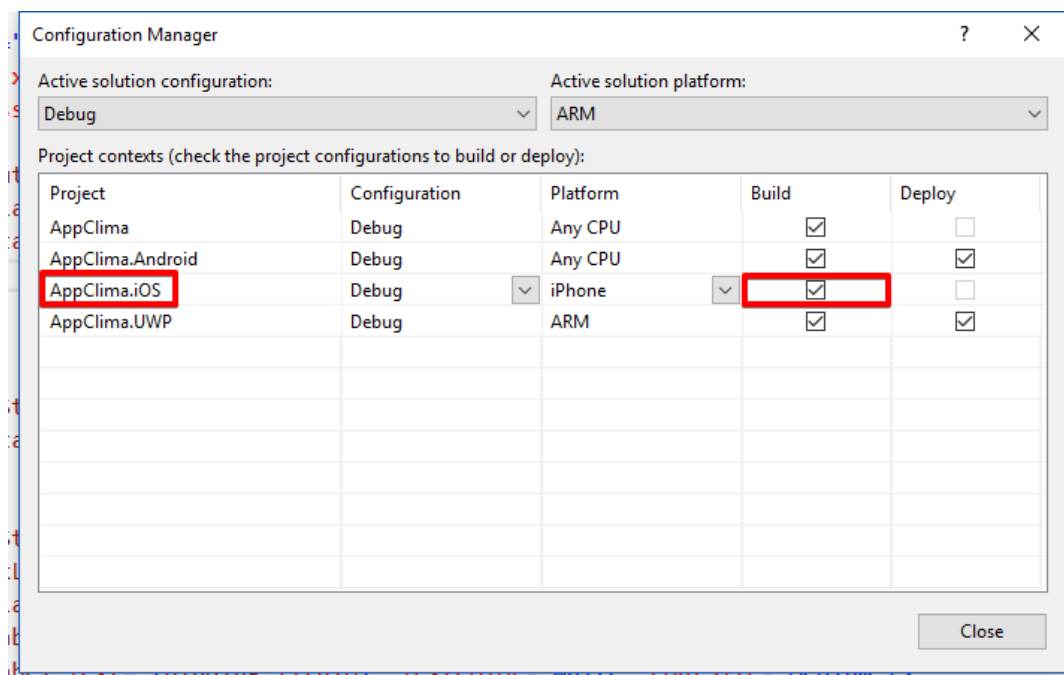




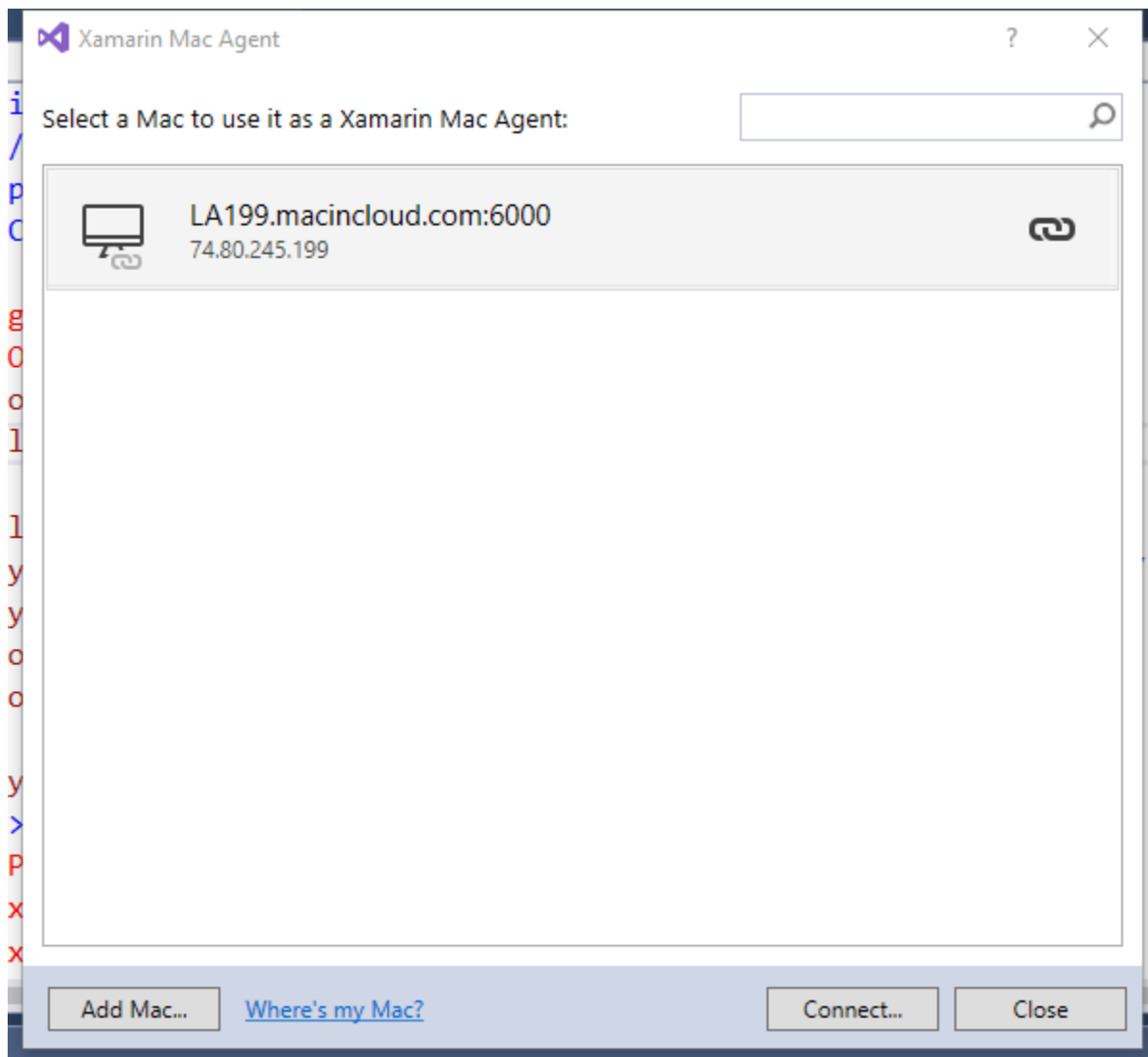
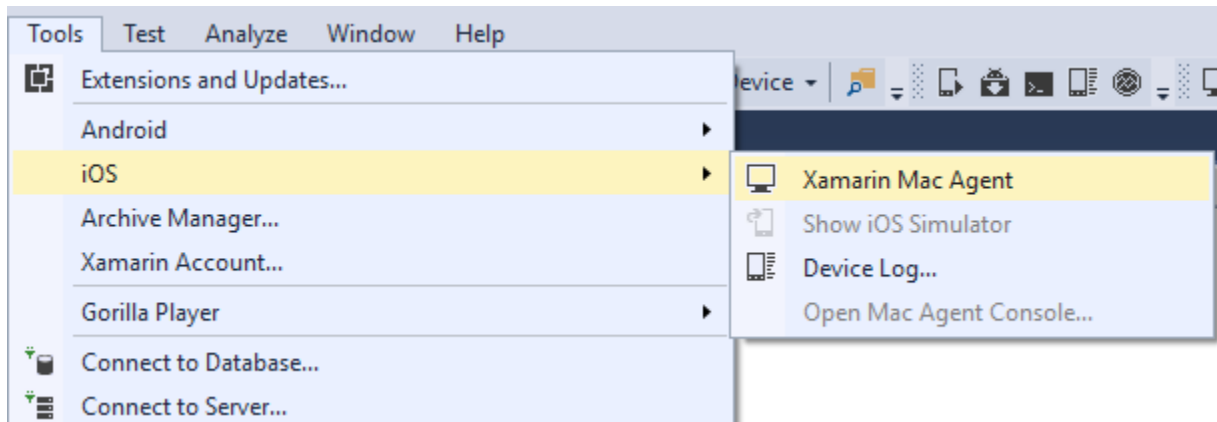
**Paso 33.** Finalmente, la prueba en **iOS**. Primero, hay que abrir **Configuration Manager** nuevamente



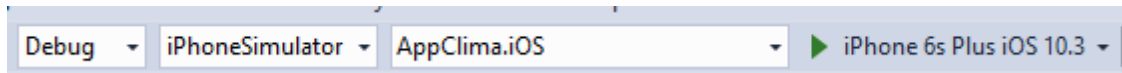
**Paso 34.** Marca **Build** en el proyecto **AppClima.iOS**.



**Paso 35.** Conéctate a tu Mac de manera remota utilizando el **Xamarin Mac Agent** desde **Herramientas** → **iOS**:



**Paso 36.** Elige **AppClima.iOS** de la lista, selecciona **iPhoneSimulator** y elige un **dispositivo** o **emulador** de la lista.



**Paso 37.** En tu Mac remota, observa el resultado de la ejecución de la app:

