



AppTempoAgora – Consumindo uma API REST

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS



Prof. Me. Tiago A. Silva



VERSÃO 2020
www.tiago.blog.br

PLANO DE AULA

OBJETIVO: BUSCAR UMA LOCALIZAÇÃO NO MAPA COM XAMARIN

- **Nesta aula vamos aprender:**

- ✓ ***Consumir uma API REST***
- ✓ ***Dividir nosso projeto em Camadas***
- ✓ ***Criar uma classe de domínio***
- ✓ ***Entender o conceito de Tasks***

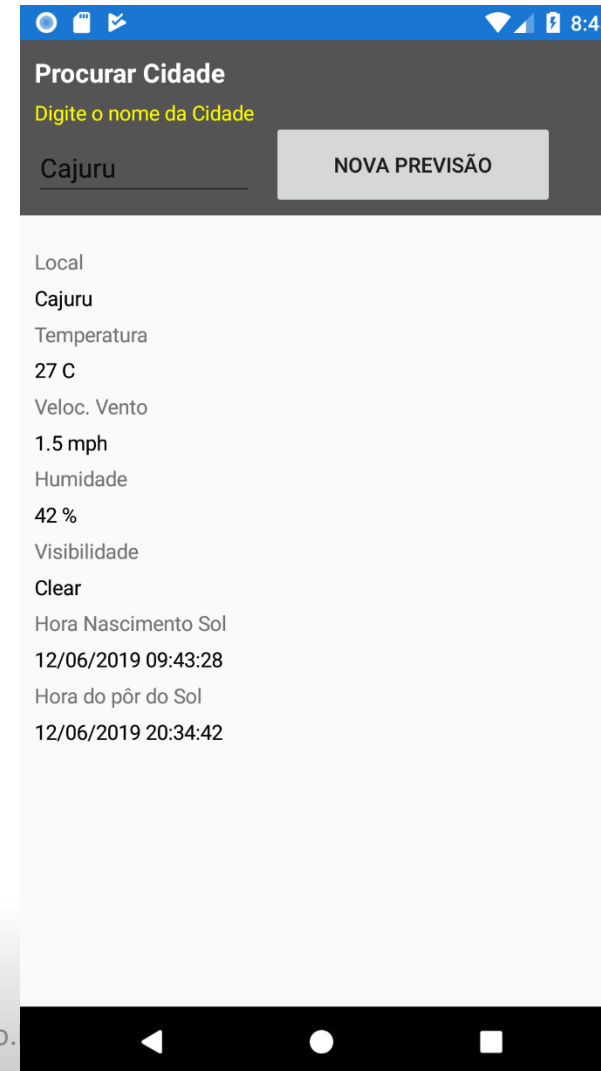
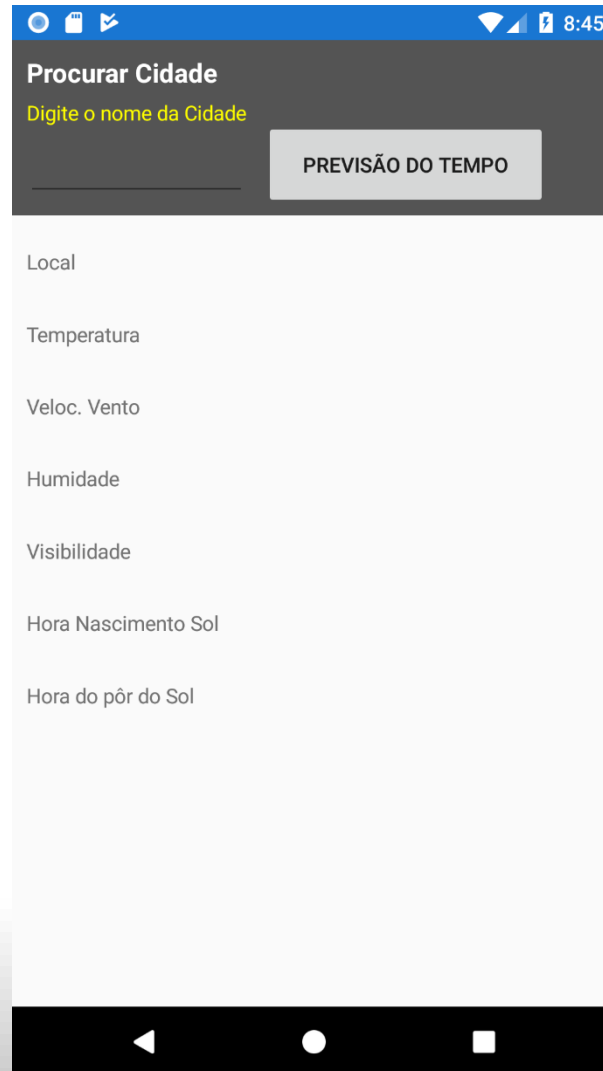


QUAL É O OBJETIVO DO NOSSO APP?

- **Vamos consultar o clima de uma cidade utilizando um serviço gratuito de previsão de tempo.**
- **A ideia é passarmos o nome de uma localização e o serviço da OpenWeather irá devolver, via JSON, as informações como Temperatura, Visibilidade, Humidade do Ar, etc.**
- **Vamos apresentar essas informações na tela para o usuário**

COMO VAI FICAR O RESULTADO FINAL?

COMO VAI FICAR DEPOIS DE PRONTO



PARTE I: OBTER UMA CHAVE DE API NO OPEN WEATHER

OBTENDO UMA CHAVE DE API NO OPEN WEATHER

- Acesse: https://home.openweathermap.org/users/sign_in
- Faça seu cadastro: nome de usuário e senha
- Veja os slides a seguir.


Sign In To Your Account

☐ Remember me


Not registered? [Create an Account.](#)

Lost your password? [Click here to recover.](#)




 Support Center

Weather in your cityHello tiagotas

WeatherMaps ▾APIPricePartnersStationsWidgetsNewsAbout ▾

New ProductsSetupAPI keysServicesPaymentsBilling plansBlock logsHistory bulkLogout



API for Agricultural on agromonitoring.com

Try our simple and fast APIs to satellite imagery, weather data and other products such as:

- Satellite imagery archive (True & False color, NDVI & EVI indices)
- Weather (current data, forecast and history)
- Accumulated temperature and precipitation
- Soil temperature and moisture

All information and API documentation is on agromonitoring.com. Read more in our [Blog](#).

[How to start](#)

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key**Name****Create key**

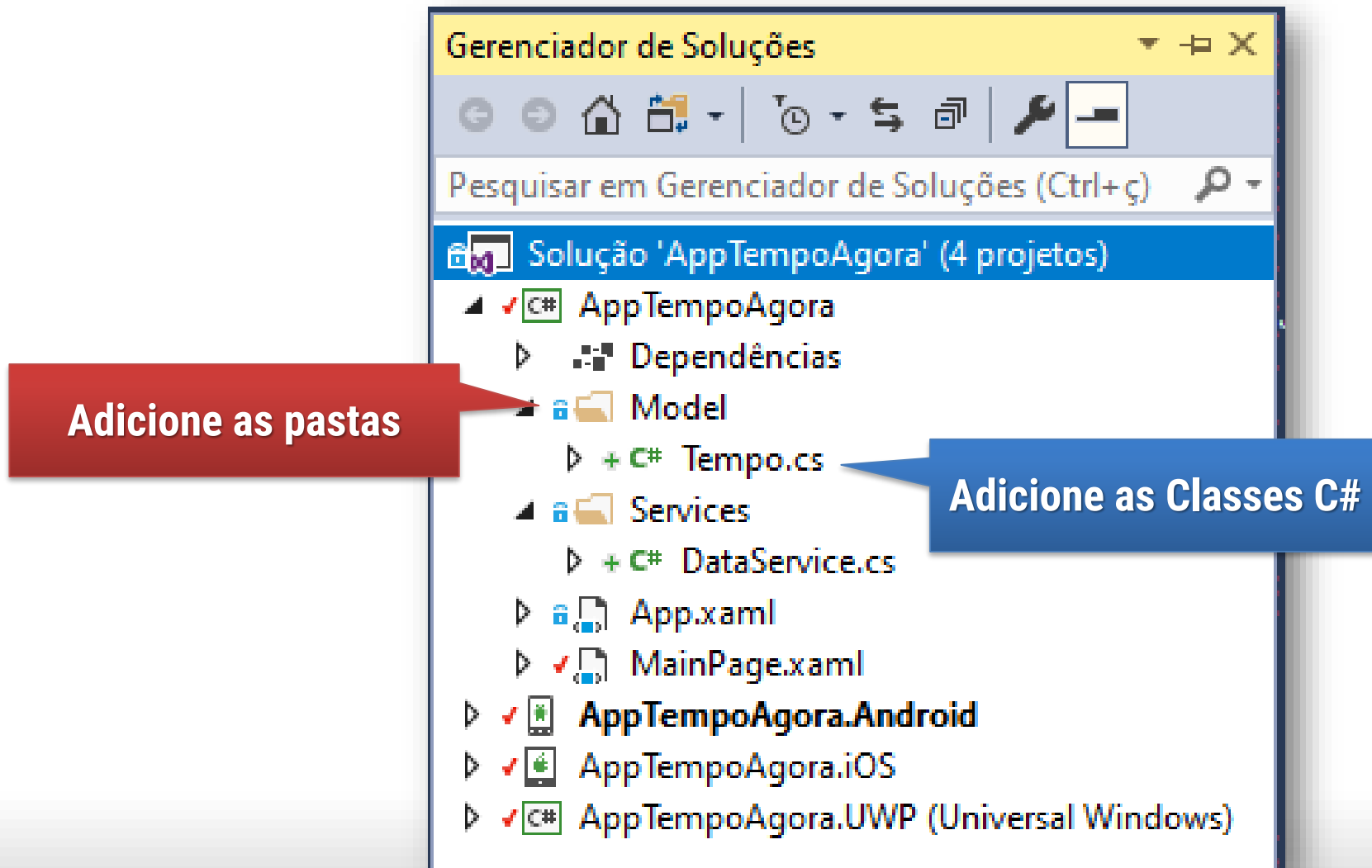
Default

*** Name**[Generate](#)

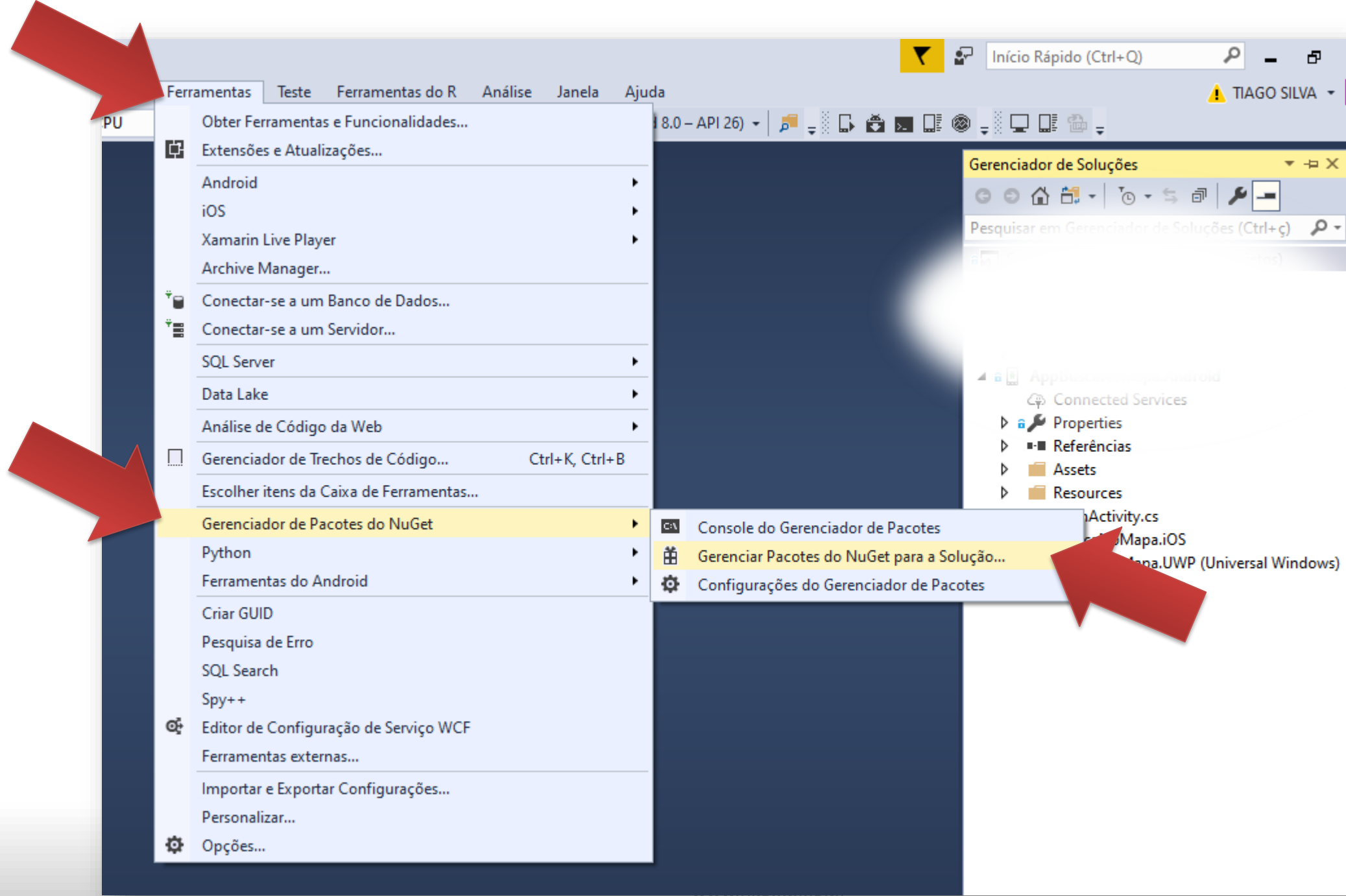
Aqui irá aparecer sua API Key depois que você se cadastrou. **Guarde-a**, vamos usá-la mais a frente!

PARTE II: ESTRUTURA DO PROJETO

Crie um projeto conforme a estrutura abaixo:



PARTE III: ADICIONANDO PACOTES NUGET



NuGet - Soluções

Tempo.cs MainPage.xaml MainPage.xaml.cs

Procurar Instalado Atualizações 7 Consolidar

Newtonsoft.Json x ↻ ☐ Incluir pré-lançamento

Origem do pacote: nuget.org ⚙

Newtonsoft.Json por James Newton-King, 232M downloads v12.0.2
Json.NET is a popular high-performance JSON framework for .NET

Amplified.ValueObjects.Newtonsoft.Json por Nicklas Jensen, 1,56K downloads
Provides JSON serialization using Newtonsoft.Json for Amplified.ValueObjects.

Remote.Linq.Newtonsoft.Json por Christof Senn, 14,4K downloads
Provides Json.NET serialization settings for Remote.Linq types.

aqua-core-newtonsoft-json por Christof Senn, 12,7K downloads
Provides Json.NET serialization settings for Aqua types.

CommonSerializer.Newtonsoft.Json por Brannon King, 8,94K downloads
CommonSerializer.Newtonsoft Class Library

Newtonsoft.Json ✓

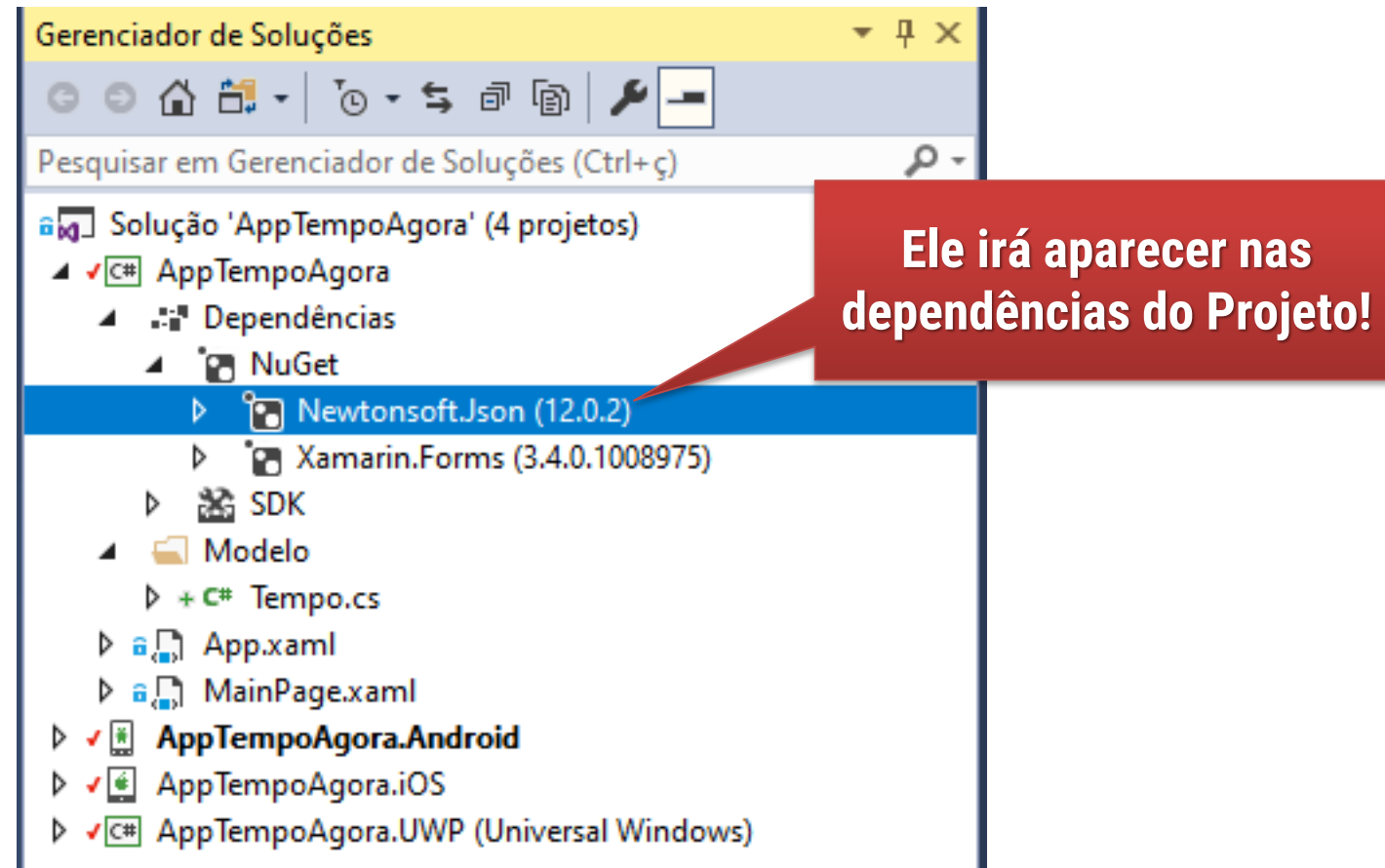
Versões - 0

<input checked="" type="checkbox"/>	Projeto	Versão
<input checked="" type="checkbox"/>	AppTempoAgora	
<input checked="" type="checkbox"/>	AppTempoAgora\AppTempoAgora.Android\AppTempoAgora	
<input checked="" type="checkbox"/>	AppTempoAgora\AppTempoAgora.iOS\AppTempoAgora	
<input checked="" type="checkbox"/>	AppTempoAgora\AppTempoAgora.UWP\AppTempoAgora	

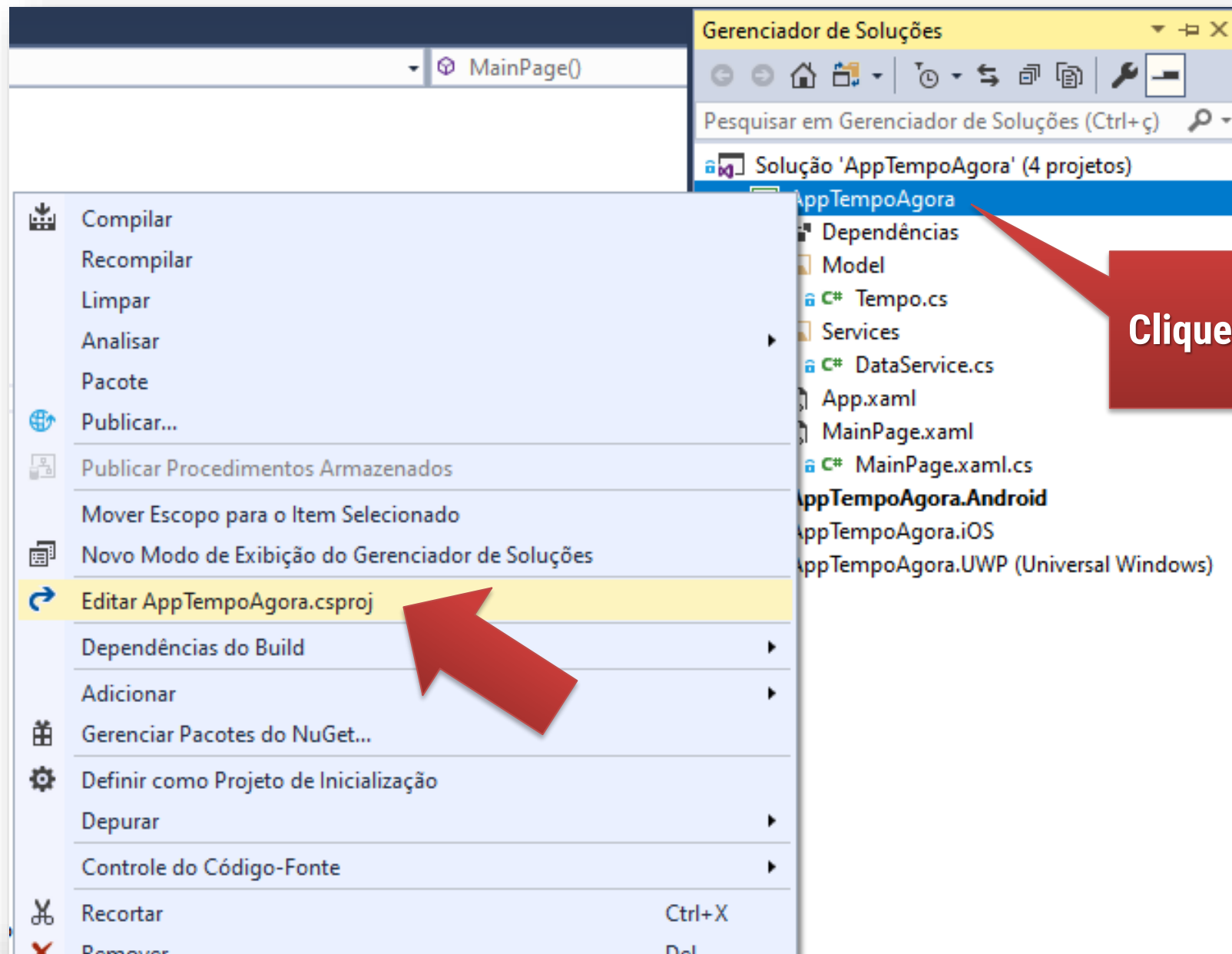
Instalado: não instalado Desinstalar

Versão: Estável mais recente 12.0.2 Instalar

Como eu sei que um pacote Nuget foi adicionado?



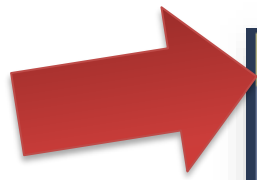
REFERENCIANDO 0 C# 4.5



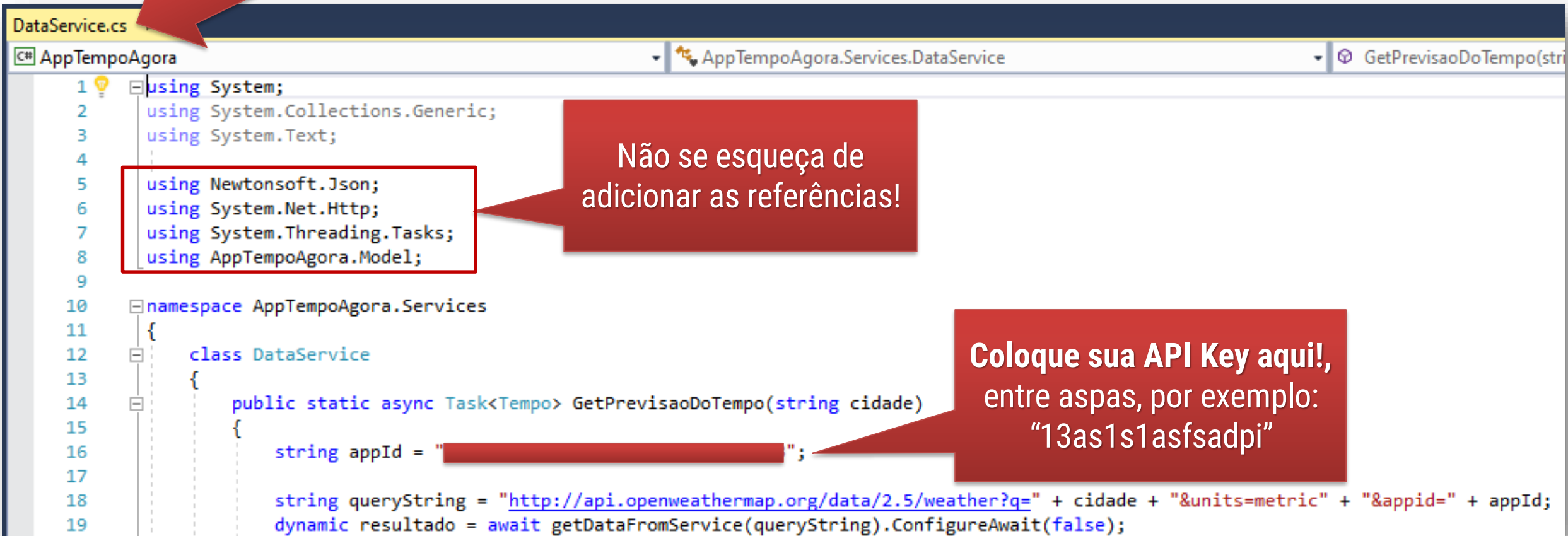
```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netstandard2.0</TargetFramework>
5   </PropertyGroup>
6
7   <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|AnyCPU'">
8     <DebugType>pdbonly</DebugType>
9     <DebugSymbols>>true</DebugSymbols>
10  </PropertyGroup>
11
12  <ItemGroup>
13    <PackageReference Include="Newtonsoft.Json" Version="12.0.2" />
14    <PackageReference Include="Xamarin.Forms" Version="3.4.0.1008975" />
15    <PackageReference Include="Microsoft.CSharp" Version="4.5.0" />
16  </ItemGroup>
17 </Project>
```

Adicione esta linha

PARTE IV: EDITANDO OS ARQUIVOS



```
Tempo.cs  X
C# AppTempoAgora AppTempoAgora.Model.Tempo
1  using System;
2      using System.Collections.Generic;
3      using System.Text;
4
5  namespace AppTempoAgora.Model
6  {
7      class Tempo
8      {
9          public string Title { get; set; }
10         public string Temperature { get; set; }
11         public string Wind { get; set; }
12         public string Humidity { get; set; }
13         public string Visibility { get; set; }
14         public string Sunrise { get; set; }
15         public string Sunset { get; set; }
16
17
18         public Tempo()
19         {
20             // Como as Labels se vinculam a estes valores vamos defini-los
21             // como uma string vazia no construtor
22             this.Title = " ";
23             this.Temperature = " ";
24             this.Wind = " ";
25             this.Humidity = " ";
26             this.Visibility = " ";
27             this.Sunrise = " ";
28             this.Sunset = " ";
29         }
30     }
31 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 using Newtonsoft.Json;
6 using System.Net.Http;
7 using System.Threading.Tasks;
8 using AppTempoAgora.Model;
9
10 namespace AppTempoAgora.Services
11 {
12     class DataService
13     {
14         public static async Task<Tempo> GetPrevisaoDoTempo(string cidade)
15         {
16             string appId = " ";
17
18             string queryString = "http://api.openweathermap.org/data/2.5/weather?q=" + cidade + "&units=metric" + "&appid=" + appId;
19             dynamic resultado = await getDataFromService(queryString).ConfigureAwait(false);
20         }
21     }
22 }
```

```
20
21 if (resultado["weather"] != null)
22 {
23     Tempo previsao = new Tempo();
24     previsao.Title = (string)resultado["name"];
25     previsao.Temperature = (string)resultado["main"]["temp"] + " C";
26     previsao.Wind = (string)resultado["wind"]["speed"] + " mph";
27     previsao.Humidity = (string)resultado["main"]["humidity"] + " %";
28     previsao.Visibility = (string)resultado["weather"][0]["main"];
29     DateTime time = new DateTime(1970, 1, 1, 0, 0, 0, 0);
30     DateTime sunrise = time.AddSeconds((double)resultado["sys"]["sunrise"]);
31     DateTime sunset = time.AddSeconds((double)resultado["sys"]["sunset"]);
32     previsao.Sunrise = String.Format("{0:d/MM/yyyy HH:mm:ss}", sunrise);
33     previsao.Sunset = String.Format("{0:d/MM/yyyy HH:mm:ss}", sunset);
34     return previsao;
35 }
36 else
37 {
38     return null;
39 }
40 }
```

```

43 public static async Task<dynamic> getDataFromService(string queryString)
44 {
45     HttpClient client = new HttpClient();
46     var response = await client.GetAsync(queryString);
47     dynamic data = null;
48     if (response != null)
49     {
50         string json = response.Content.ReadAsStringAsync().Result;
51         data = JsonConvert.DeserializeObject(json);
52     }
53     return data;
54 }
55
56
57 public static async Task<dynamic> getDataFromServiceByCity(string city)
58 {
59     string appId = " ";
60
61     string url = string.Format("http://api.openweathermap.org/data/2.5/forecast/daily?q={0}&units=metric&cnt=1&APPID={1}", city.Trim(), appId);
62     HttpClient client = new HttpClient();
63     var response = await client.GetAsync(url);
64     dynamic data = null;
65     if (response != null)
66     {
67         string json = response.Content.ReadAsStringAsync().Result;
68         data = JsonConvert.DeserializeObject(json);
69     }
70     return data;
71 }
72 }
73

```

**Coloque sua API Key aqui!,
entre aspas, por exemplo:
"13as1s1asfsadpi"**


```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4             xmlns:local="clr-namespace:AppTempoAgora"
5             x:Class="AppTempoAgora.MainPage">
6
7     <StackLayout>
8         <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand" BackgroundColor="#545454">
9             <StackLayout Padding="10,10,10,10" HorizontalOptions="Start">
10                 <Label Text="Procurar Cidade" TextColor="White" FontAttributes="Bold" FontSize="Medium" />
11                 <Label x:Name="cidadeLabel" Text="Digite o nome da Cidade" TextColor="Yellow" FontSize="Small"/>
12                 <Entry x:Name="cidadeEntry"/>
13             </StackLayout>
14             <StackLayout Padding="0,0,0,10" VerticalOptions="End">
15                 <Button x:Name="btnPrevisao" Text="Previsão do Tempo" WidthRequest="185"
16                         BorderWidth="1" Clicked="btnPrevisao_Clicked"/>
17             </StackLayout>
18         </StackLayout>
19         <StackLayout Padding="10,10,10,10" HorizontalOptions="Start">
20             <Image x:Name="imgTempo" />
21             <Label Text="Local" />
22             <Label Text="{Binding Title}" TextColor="Black" />
23             <Label Text="Temperatura" />
24             <Label x:Name="tempLabel" TextColor="Black" Text="{Binding Temperature}" />
25             <Label Text="Veloc. Vento" />
26             <Label x:Name="windLabel" Text="{Binding Wind}" TextColor="Black" />
27             <Label Text="Humidade" />
28             <Label x:Name="humidityLabel" Text="{Binding Humidity}" TextColor="Black"/>
29             <Label Text="Visibilidade" />
30             <Label x:Name="visibilitylabel" Text="{Binding Visibility}" TextColor="Black"/>
31             <Label Text="Hora Nascimento Sol" />
32             <Label x:Name="sunriseLabel" Text="{Binding Sunrise}" TextColor="Black" />

```

```
33         <Label Text="Hora do pôr do Sol" />
34         <Label x:Name="sunsetLabel" Text="{Binding Sunset}" TextColor="Black"/>
35     </StackLayout>
36 </StackLayout>
37
38 </ContentPage>
39
```

MainPage.xaml.cs

C# AppTempoAgora

AppTempoAgora.Ma

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Xamarin.Forms;
7
8 using AppTempoAgora.Model;
9 using AppTempoAgora.Services;
10
11 namespace AppTempoAgora
12 {
13     public partial class MainPage : ContentPage
14     {
15         public MainPage()
16         {
17             InitializeComponent();
18             this.Title = "Previsão do Tempo";
19             //define o databinding para os objetos padrão iniciais
20             this.BindingContext = new Tempo();
21         }
22     }
23 }
```

Não se esqueça de
adicionar as referências!

```
22
23 private async void btnPrevisao_Clicked(object sender, EventArgs e)
24 {
25     try
26     {
27         if (!String.IsNullOrEmpty(cidadeEntry.Text))
28         {
29             Tempo previsaoDoTempo = await DataService.GetPrevisaoDoTempo(cidadeEntry.Text);
30             this.BindingContext = previsaoDoTempo;
31             btnPrevisao.Text = "Nova Previsão";
32         }
33     }
34     catch (Exception ex)
35     {
36         await DisplayAlert("Erro ", ex.Message, "OK");
37     }
38 }
39
40
41
```

OBRIGADO!

- Encontre este **material on-line** em:
 - www.tiago.blog.br/
 - SIGA
- Em caso de **dúvidas**, entre em contato:
 - **Tiago:** tiago.silva197@fatec.sp.gov.br

