



Neste artigo vou mostrar como podemos obter a geolocalização com base nas informações do dispositivo e exibir a posição no Mapa em aplicações Xamarin Forms usando o [Visual Studio com Xamarin e a linguagem C#](#).

A geolocalização é um processo pelo qual se estabelece o local de onde o usuário está efetuando determinada operação, como por exemplo: atender uma chamada de celular, acessar um site, tirar uma fotografia, utilizar o GPS, entre outros.

**Curso de Xamarin Forms Vídeo Aulas**

Desenvolva para Android, iOS e Windows Phone

A geolocalização tira proveito dos serviços de localização específicos do dispositivo para ajudar a fornecer um local para um determinado usuário em qualquer momento.

Há muitos usos comuns a geolocalização em aplicativos para dispositivos móveis, como exibir dados meteorológicos para a localização de um usuário, orientar um usuário para um destino ou ajudar a encontrar um restaurante nas proximidades. Mesmo se você não estiver construindo um aplicativo de meteorologia ou de mapas, existem outras aplicações úteis de serviços de localização, incluindo recursos específicos de localização ou pré-preenchimento de formulários de entrada de dados (*cidade, estado, CEP*) para ajudar a melhorar as taxas de conversão.

Neste artigo vamos obter a geolocalização de um usuário a partir dos dados do celular, exibindo a **latitude e longitude** a seguir vamos exibir a sua posição no Google Maps.

Para realizar essas tarefas vamos usar duas APIs prontas que nos oferecem os recursos necessários, São elas :

- **Geolocator Plugin** - Identificado no Nuget como : [Xam.Plugin.Geolocator](#) - Acessa a geolocalização do dispositivo obtendo latitude, longitude, etc.

Instalação via Package Manger Console : **Install-Package Xam.Plugin.Geolocator -Version 3.0.4**

- **Maps Plugin for Xamarin** - Identificado no Nuget como : [Xam.Plugin.ExternalMaps](#) - Abre mapas externos para navegar até uma geolocalização ou endereço específico.

Instalação via Package Manger Console : **Install-Package Xam.Plugin.ExternalMaps**

No exemplo usado neste artigo eu vou me basear na documentação das APIs que podem ser obtidas nos links a seguir :

- **Geolocator Plugin** - <https://components.xamarin.com/gettingstarted/geolocatorplugin>
- **Maps Plugin for Xamarin** - <https://components.xamarin.com/gettingstarted/externalmapsplugin>

A utilização é bem simples. Qualquer dúvida, ou para mais detalhes consulte a documentação.

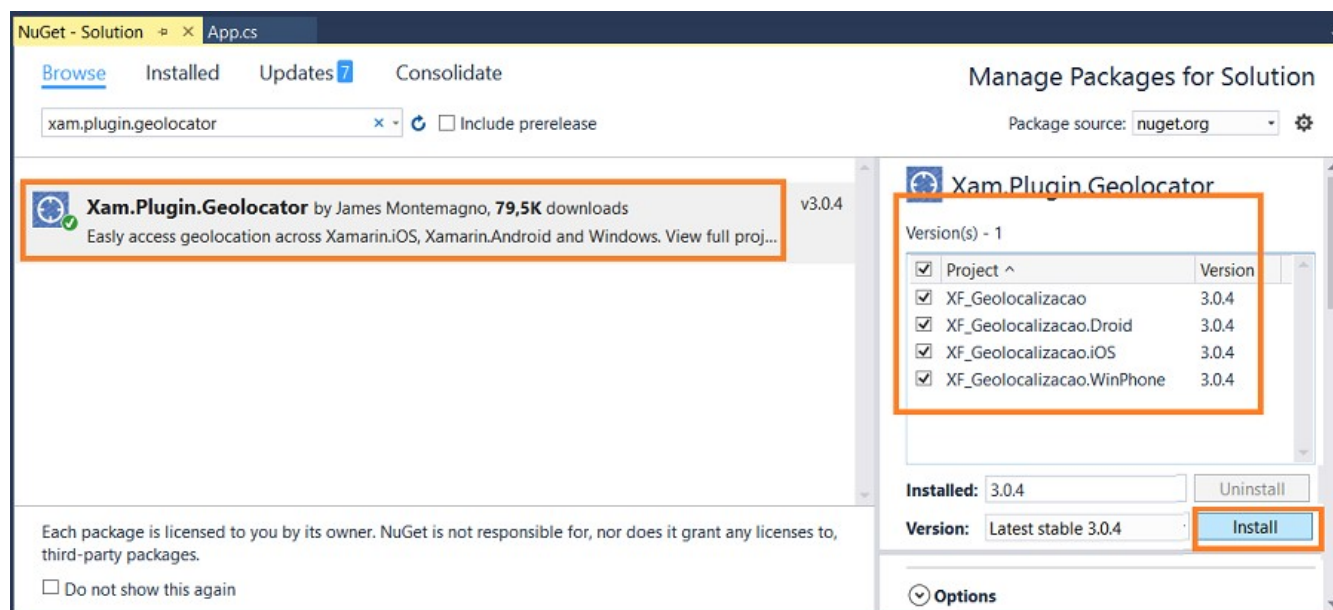
Um detalhe importante é que você deve instalar ambos os plugins em TODOS os projetos da sua solução Xamarin Forms.

Instalando as APIs em todos os projetos da solução

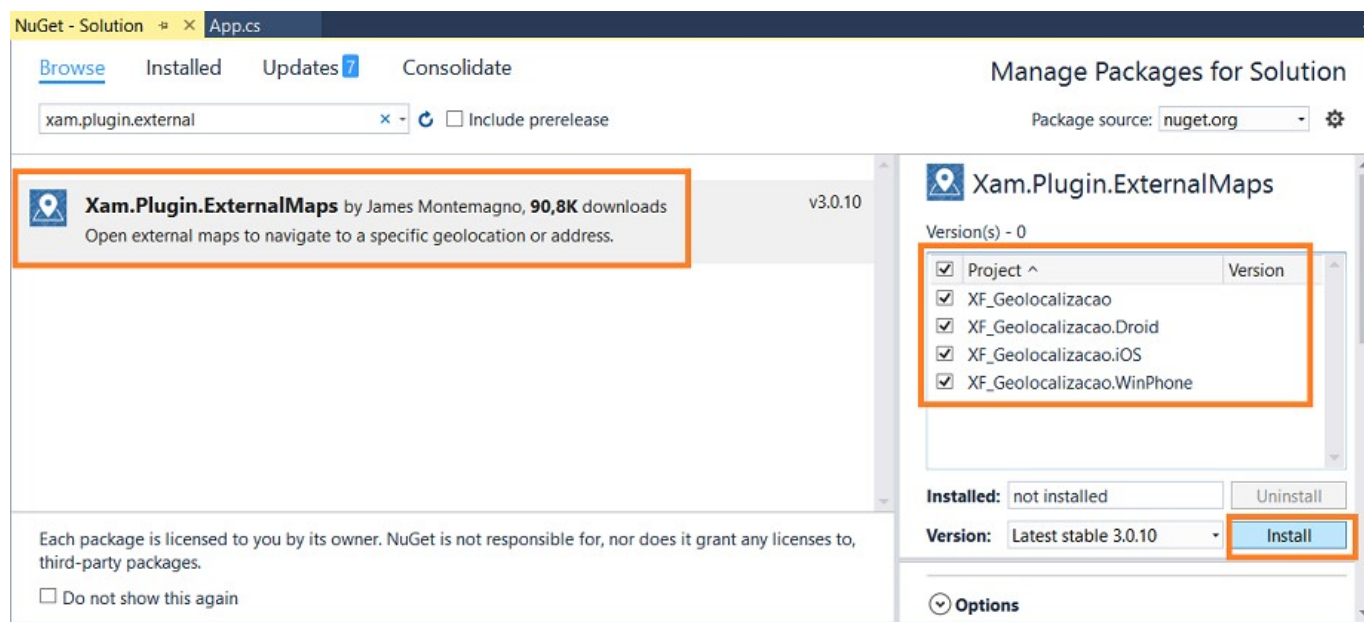
No menu **Tools**, clique em **Nuget Package Manager** e a seguir em **Manage Nuget Packages for Solution**;

Digite : [xam.plugin.geolocator](#) para localizar o pacote e a seguir marque o pacote e instale-o em todos os

projetos:



Repita o procedimento acima e agora digite : [xam.plugin.external](#) para localizar o pacote e a seguir marque o pacote e instale-o em todos os projetos:



Pronto ! agora já temos todos os recursos necessários nos projetos.

Ajustes a serem feitos em cada plataforma

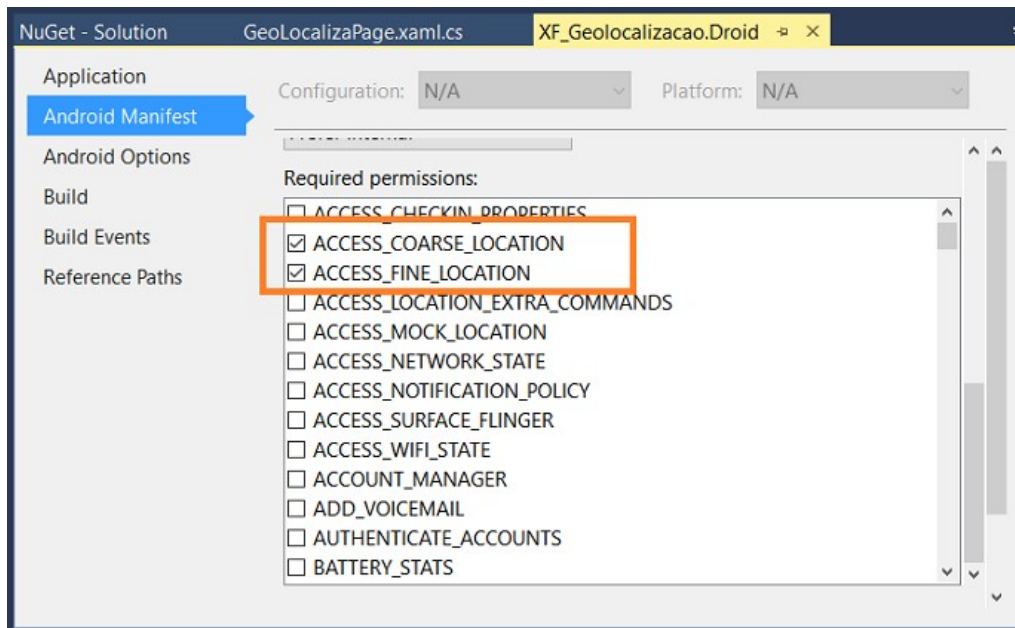
No momento em que esse artigo foi escrito os ajustes necessários em cada plataforma eram os seguintes :

Android

Para permitir que um aplicativo Android acesse os serviços de localização, precisamos ativar duas permissões do Android:

- 1 - [ACCESS_COARSE_LOCATION](#)
- 2 - [ACCESS_FINE_LOCATION](#)

Uma forma de fazer isso é abrir a janela de propriedades para o projeto Android e a seguir clicar em **Android Manifest** e marcar as opções conforme mostra a figura a seguir:



iOS

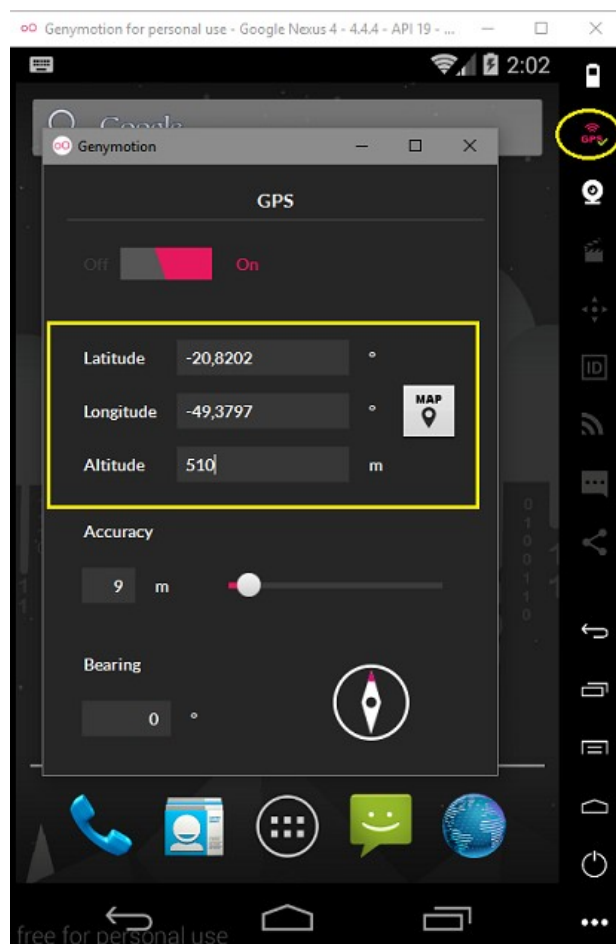
Voce vai precisar adicionar a chave **NSLocationWhenInUsageDescription** ou **NSLocationAlwaysUsageDescription** no arquivo **Info.plist**, juntamente com uma nova entrada de string para a chave que descreve exatamente o que você estará fazendo com a localização do usuário.

Windows Phone

Você vai precisar habilitar a permissão : **ID_CAP_LOCATION**

Ajustes no Android e no GenyMotion

No nosso exemplo eu vou fazer a emulação do projeto **Android** usando o **Genymotion**. Dessa forma, como eu não vou usar um dispositivo físico, vou definir os dados iniciais do GPS do GenyMotion, informando uma latitude, longitude e altitude conforme mostra a figura abaixo:



Se você for usar outro emulador procure a opção de configuração para o GPS e defina os dados iniciais. Se for usar um dispositivo físico isso não é necessário.

Realizando essas tarefas poderemos criar a aplicação e usar os recursos das APIs para obter a geolocalização.

Vamos lá...

Recursos usados:

- [Visual Studio Community 2015](#) ou **Xamarin Studio**
- [Xamarin](#)

Nota: Baixe e use a versão **Community 2015** do VS ela é grátis e é equivalente a versão **Professional**.

Criando o projeto no Visual Studio 2015 Community

Abra o [Visual Studio Community 2015](#) e clique em **New Project**;

Selecione Visual C#, o template **Cross Platform** e a seguir **Blank App (Xamarin.Forms Portable)**;

Nota : A opção *Portable* ([Portable Class Library - PCL](#)) - Inclui todo o código comum em uma biblioteca de vínculo dinâmico (DLL) que pode então ser referenciada a partir de outros projetos;

Informe o nome **XF_Geolocalizacao** e clique no botão OK;

Ao clicar no botão OK, será criada uma solução contendo 4 projetos. (*Dependendo do seu ambiente pode haver alguma variação nos projetos.*)

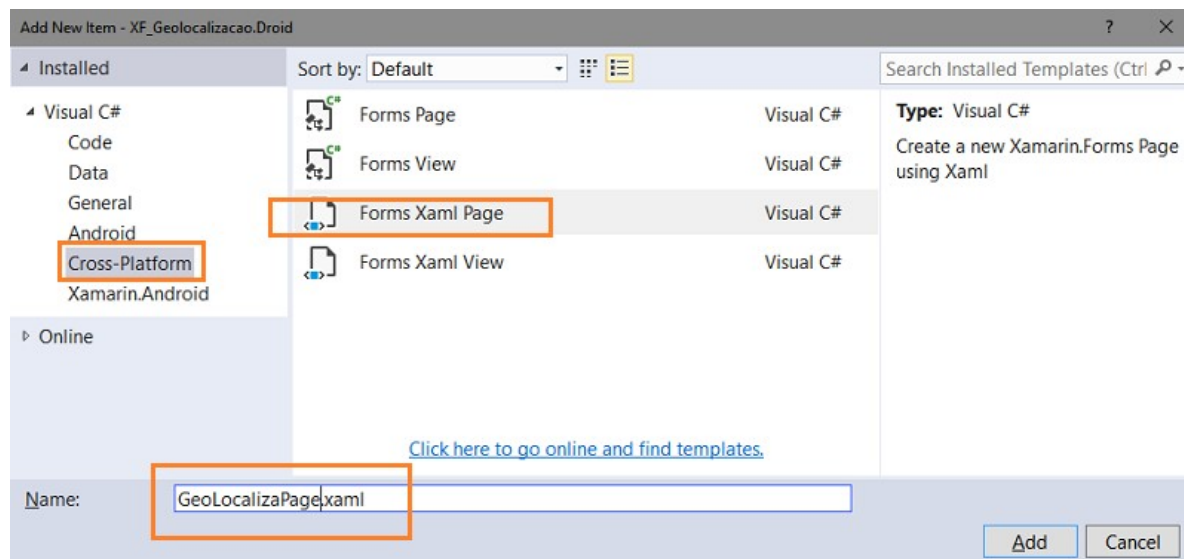
O projeto comum possui a classe **App.cs** que irá conter o código compartilhado e que vamos usar neste

artigo.

Ao final teremos as referências incluídas em todos os projetos da nossa solução.

Selecione o projeto e a seguir clique no menu **Project** e em **Add New Item**;

Selecione **Cross Platform** e o template **Forms Xaml Page** e informe o nome **GeoLocalizaPage.xaml**:



Esta será a página principal da nossa aplicação.

Então vamos abrir o arquivo **App.cs** e alterar o código conforme abaixo:

```
using Xamarin.Forms;

namespace XF_Bindable_Spinner
{
    public class App : Application
    {
        public App()
        {
            MainPage = new NavigationPage(new GeoLocalizaPage());
        }

        protected override void OnStart()
        {
            // Handle when your app starts
        }

        protected override void OnSleep()
        {
            // Handle when your app sleeps
        }

        protected override void OnResume()
        {
            // Handle when your app resumes
        }
    }
}
```

Aqui definimos a página **MainPage** como a página que vai iniciar a nossa aplicação.

Definindo o código da GeoLocalizaPage

Abra o arquivo **MainPage.xaml** e inclua as seguintes views usando um layout **StackLayout**:

- 1 Label - x:Name=**lblGeolocalizacao**
- 2 Buttons - x:Name = **btnGeolocalizacao** e **btnMostrarPosicaoNoMapa**

Vamos definir em cada Button um evento **Clicked** onde iremos definir o código da aplicação:

```
<?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="XF_Geolocalizacao.GeoLocalizaPage"
    Title="Geolocalização">

    <StackLayout VerticalOptions="Center" Spacing="5">

        <Label x:Name="lblGeolocalizacao" Text=""
            TextColor="Blue" HorizontalOptions="Center"/>

        <Button x:Name="btnGeolocalizacao" Text="Obter Geolocalização : Latitude/Longitude"
            TextColor="Black"
            Clicked="btnGeolocalizacao_Clicked" />

        <Button x:Name="btnMostrarPosicaoNoMapa" Text="Exibir posição no Mapa"
            TextColor="Black"
            Clicked="btnMostrarPosicaoNoMapa_Clicked"/>
    </StackLayout>

</ContentPage>
```

Agora abra o arquivo **MainPage.xaml.cs** e defina o código para os eventos de cada um dos Buttons:

```
using Plugin.ExternalMaps;
using Plugin.Geolocator;
using System;

using Xamarin.Forms;

namespace XF_Geolocalizacao
{
    public partial class GeoLocalizaPage : ContentPage
    {
        public GeoLocalizaPage()
        {
            InitializeComponent();
        }

        double latitude = 0;
        double longitude = 0;

        private async void btnGeolocalizacao_Clicked(object sender, EventArgs e)
        {
            lblGeolocalizacao.Text = "Obtendo a geolocalização....\n";

            try
            {
                var locator = CrossGeolocator.Current;
                locator.DesiredAccuracy = 50;

                var position = await locator.GetPositionAsync(timeoutMilliseconds: 10000);

                lblGeolocalizacao.Text += "Status: " + position.Timestamp + "\n";
                lblGeolocalizacao.Text += "Latitude: " + position.Latitude + "\n";
            }
        }
    }
}
```

```
        lblGeolocalizacao.Text += "Longitude: " + position.Longitude;

        latitude = position.Latitude;
        longitude = position.Longitude;
    }
    catch (Exception ex)
    {
        await DisplayAlert("Erro : ", ex.Message, "OK");
    }
}

private void btnMostrarPosicaoNoMapa_Clicked(object sender, EventArgs e)
{
    try
    {
        CrossExternalMaps.Current.NavigateTo("Macoratti", latitude, longitude);
    }
    catch (Exception ex)
    {
        DisplayAlert("Erro : ", ex.Message, "OK");
    }
}
}
```

Neste código cabe destacar o seguinte :

1- No evento Click do botão de comando - **Obter Geolocalização : Latitude/Longitude** - usamos o método **GetPositionAsync()** para obter a latitude e longitude a partir do dispositivo:

```
var locator = CrossGeolocator.Current;
locator.DesiredAccuracy = 50;

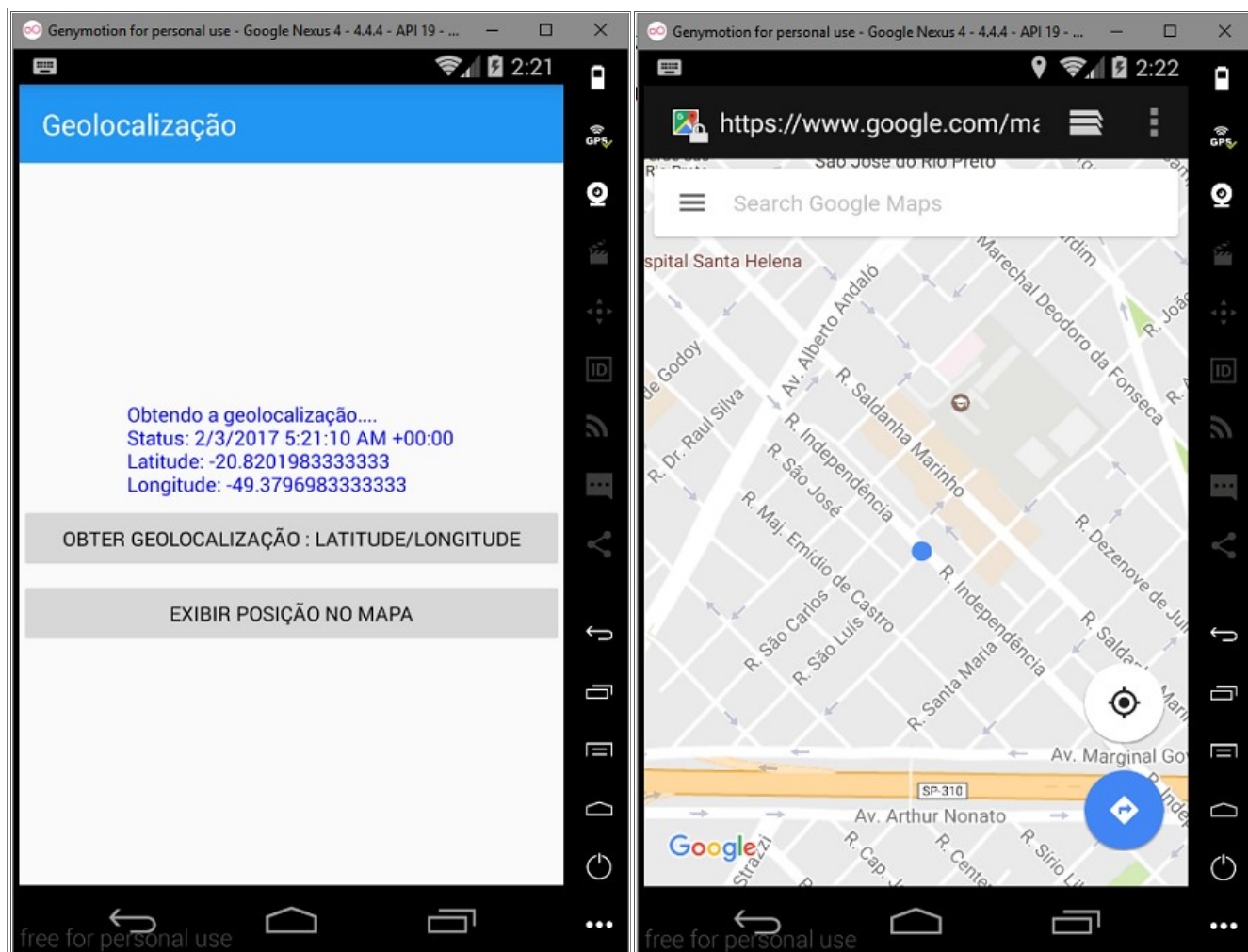
var position = await locator.GetPositionAsync(timeoutMilliseconds: 10000);
```

2- No evento Click do botão de comando - **Exibir posição no Mapa** - usamos o método **NavigateTo()** para exibir a posição no mapa:

```
CrossExternalMaps.Current.NavigateTo("Macoratti", latitude, longitude);
```

Executando o projeto e clicando no botão para obter a geolocalização iremos obter os valores conforme mostra a primeira figura.

A seguir clicando no botão para exibir o mapa veremos a posição da **latitude e longitude** exibida no mapa como mostra a segunda figura:



Pegue o código das páginas aqui : 📦 [XF Geolocalizacao.zip](#) (sem as referências)

"Em quem (Jesus) temos a redenção pelo seu sangue, a remissão das ofensas, segundo as riquezas da sua graça," Efésios 1:7

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Vídeo Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) **NEW**

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#) **NEW**

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) **NEW**
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin - Apresentando Xamarin.Forms - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net](#)
- <https://developer.xamarin.com/api/type/Android.Widget.ListView/>
- <https://developer.xamarin.com/api/property/Android.Widget.ListView.Adapter/>
- <https://developer.xamarin.com/guides/xamarin-forms/user-interface/animation/>

[José Carlos Macoratti](#)