




Mapas

1. Acesse o endereço <https://console.developers.google.com> e faça autenticação com uma conta google. Clique para concordar com os termos de serviços, caso seja solicitado.
2. No menu superior esquerdo, clique em “Selecione um projeto” e na janela que se abrirá, clique em “Novo Projeto”

Novo projeto

 Você tem 19 projects restantes na sua cota. Solicite um aumento ou exclua projetos. [Saiba mais](#)


[MANAGE QUOTAS](#)

Nome do projeto *

MauíMaps

ID do projeto: mauimaps-387418. Não é possível alterá-lo depois. [EDITAR](#)

Local *

 Sem organização

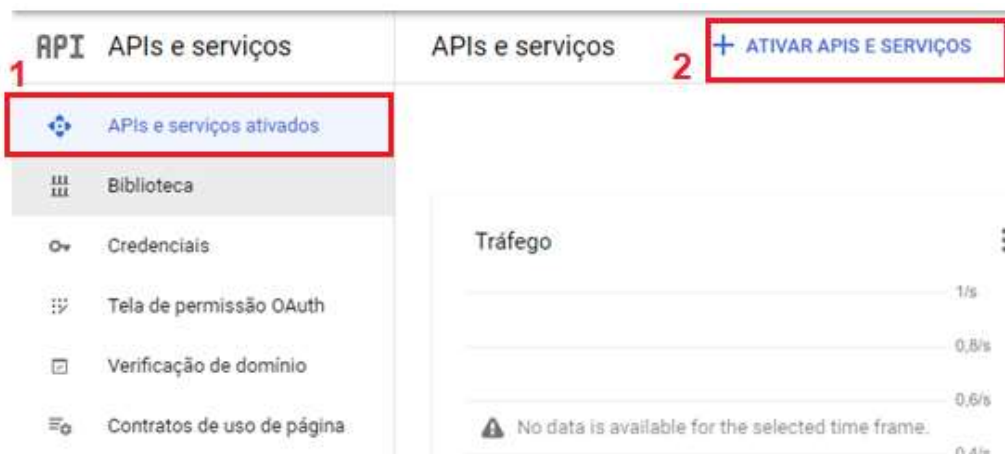
[PROCURAR](#)

Pasta ou organização pai

CRIAR

 CANCELAR

3. Clique em “APIs e serviços” → “API e serviços ativados” e depois em “Ativar APIs e serviços”





4. Em mapas, selecione “Maps SDK for Android” e depois clique na opção “Ativar”



Maps for your native Android app.

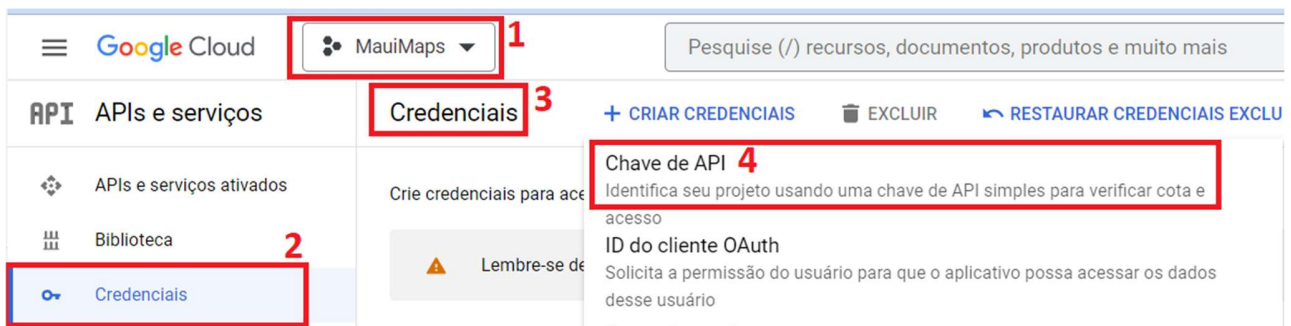
ATIVAR

VISÃO GERAL

DOCUMENTAÇÃO

SUPORTE

5. Volte para a Home do Google Cloud Plataform e com seu projeto selecionado, clique em Credenciais → Criar credenciais → Chave de API. Copie esta chave para um lugar seguro, pois usaremos ela na Solution do Visual Studio.



- Existem maneiras de deixar sua chave mais segura, para que ela não seja usada por robôs ou não seja clonada, para isso, deve ser feita uma vinculação da chave a uma única solution do Visual Studio. Isso será abordado mais à frente.



6. Abra o arquivo **AndroidManifest.xml** (pasta Platforms/Android/Resources) para adicionar as permissões : **ACCESS_NETWORK_STATE**, **ACCESS_COARSE_LOCATION**, **ACCESS_FINE_LOCATION**, **ACCESS_mock_LOCATION**, conforme abaixo

Required permissions

<input type="checkbox"/>	ACCESS_BLOBS_ACROSS_USERS
<input type="checkbox"/>	ACCESS_CHECKIN_PROPERTIES
<input checked="" type="checkbox"/>	ACCESS_COARSE_LOCATION
<input checked="" type="checkbox"/>	ACCESS_FINE_LOCATION
<input type="checkbox"/>	ACCESS_LOCATION_EXTRA_COMMANDS
<input type="checkbox"/>	ACCESS_MEDIA_LOCATION
<input checked="" type="checkbox"/>	ACCESS_mock_LOCATION
<input checked="" type="checkbox"/>	ACCESS_NETWORK_STATE

7. Clique com o direito neste mesmo arquivo e escolha → Open with → XML (Text) Editor. Iremos incluir as informações da chave da API. portanto abra um espaço na tag application e codifique conforme abaixo.

```
<application android:allowBackup="true"
    android:icon="@mipmap/has_logo_transparente"
    android:roundIcon="@mipmap/has_logo_transparente_round"
    android:supportsRtl="true">
```

```
<meta-data android:name="com.google.android.geo.API_KEY"
    android:value="CHAVE_DA_SUA_API_GOOGLE_MAPS">
</meta-data>
```

```
</application>
```

8. Clique com o botão direito na Solution e em “Manage Nuget Packages for Solution”. Na aba Browser pesquise por Microsoft.Maui.Controls.Maps, selecione o projeto e depois clique em Install.

Manage Packages for Solution

Browse Installed Updates Consolidate

Microsoft.Maui.Controls.Maps x Include prerelease Package source: nuget.org

Microsoft.Maui.Controls.Maps by 8.0.21
Maps and mapping support for .NET Multi-platform App UI (.NET MAUI) apps.

Microsoft.Maui.Maps by Microsoft, 8.0.21
Maps and mapping support for .NET Multi-platform App UI (.NET MAUI) apps. This pac...

FmgLib.MauiMarkup.Maps by FmgYa, 8.0.14
C# Code Markup Library for .NET MAUI (Microsoft.Maui.Controls.Maps)

Microsoft.Maui.Controls.Maps nuget.org

Versions - 0

<input checked="" type="checkbox"/>	Project
<input checked="" type="checkbox"/>	AppRpgEtec

Installed: not installed Uninstall

Version: Latest stable 8.0.21 Install

Package source mapping is off. Configure



9. Abra o arquivo MauiProgram.cs para configurar o uso de mapas.

```
public static MauiApp CreateMauiApp()
{
    var builder = MauiApp.CreateBuilder();
    builder
        .UseMauiApp<App>()
        .ConfigureFonts(fonts =>
        {
            fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
            fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
        })
        .UseMauiMaps();
}
```

10. Clique com o direito na pasta Views/Usuarios e crie uma view (.NET MAUI) chamada **LocalizacaoView.Xaml**. Faça a declaração do namespace para o uso de mapas e crie um controle do tipo Map conforme sinalizado:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppRpgEtec.Views.Usuarios.LocalizacaoView"
    xmlns:maps="clr-namespace:Microsoft.Maui.Controls.Maps;assembly=Microsoft.Maui.Controls.Maps"
    Title="LocalizacaoView">
    <maps:Map x:Name="mapa" ItemsSource="{Binding MeuMapa}" />
</ContentPage>
```

11. Na view AppShell.xaml, realize a edição para que o menu de Usuários vire uma tab

```
<ShellContent Title="Usuários" Icon="menuusuarios.svg"
    ContentTemplate="{DataTemplate viewsUsuarios:LocalizacaoView}" />
```



- Execute o aplicativo para realizar os testes.



12. Na pasta ViewModels/Usuarios, crie uma classe chamada **LocalizacaoViewModel.cs** herdando de BaseViewModels. Realize o using a seguir:

```
using Map = Microsoft.Maui.Controls.Maps.Map;
```

13. Faça a programações iniciais declarando um atributo e propriedade do tipo Map

```
private Map meuMapa;  
9 references  
public Map MeuMapa  
{  
    get => meuMapa;  
    set  
    {  
        if (value != null)  
        {  
            meuMapa = value;  
            OnPropertyChanged();  
        }  
    }  
}
```




14. Ainda na classe **LocalizacaoViewModel.cs**, crie um método para ativar a chamar a localização atual

```
public async void InicializarMapa()
{
    try
    {
        //Próxima etapa aqui
    }
    catch (Exception ex)
    {
        await Application.Current.MainPage
            .DisplayAlert("Erro", ex.Message, "OK");
    }
}
```

15. Faça a programação abaixo dentro do bloco try

```
//Coordenadas geográficas da escola
Location location = new Location(-23.5200241d, -46.596498d);
Pin pinEtec = new Pin()
{
    Type = PinType.Place,
    Label = "Etec Horácio",
    Address = "Rua alcântara, 113, Vila Guilherme",
    Location = location
};

Map map = new Map();
MapSpan mapSpan = MapSpan
    .FromCenterAndRadius(location, Distance.FromKilometers(5));
map.Pins.Add(pinEtec);
map.MoveToRegion(mapSpan);

MeuMapa = map;
```



16. Abra a parte de código da ContentPage LocalizacaoView.xaml e defina a ligação com a classe ViewModel

```
public partial class LocalizacaoView : ContentPage
{
    LocalizacaoViewModel viewModel;
    1 reference
    public LocalizacaoView()
    {
        InitializeComponent();

        viewModel = new LocalizacaoViewModel();
        viewModel.InicializarMapa();

        BindingContext = viewModel;
    }
}
```

17. Abra o layout da view e altere o controle de mapa deixando como segue

```
<maps:Map x:Name="mapa" ItemsSource="{Binding MeuMapa}">
    <maps:Map.ItemTemplate>
        <DataTemplate>
            <maps:Pin Location="{Binding Location}"
                Address="{Binding Address}"
                Label="{Binding Label}" />
        </DataTemplate>
    </maps:Map.ItemTemplate>
</maps:Map>
```

- Execute para testar a visualização do Pin no mapa.





18. Acrescente as propriedades sinalizadas abaixo e execute para observar as mudanças.

```
<maps:Map x:Name="mapa" ItemsSource="{Binding MeuMapa}"  
    MapType="Hybrid" IsShowingUser="true"  
    IsZoomEnabled="True" IsTrafficEnabled="True">  
    <maps:Map.ItemTemplate>  
        <DataTemplate>
```

Abaixo está o resultado esperado

