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();
```

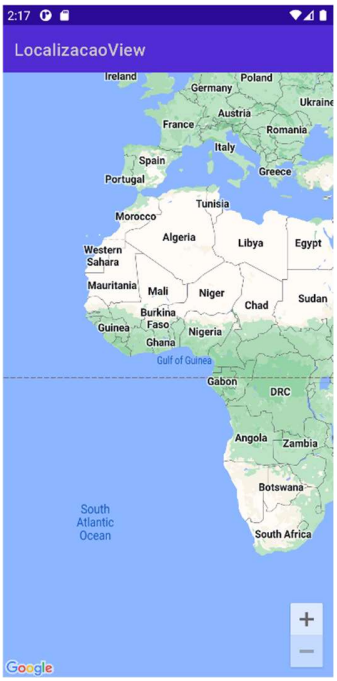
10. Clique com o direito na pasta Views/Usuarios e crie uma contente page extensão ".xaml" (.NET MAUI) chamada **LocalizacaoView.Xaml**. Faça a declaração do namespace para o uso de mapas e crie um controle do tipo Map conforme sinalizado:

```
<?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}">
        </maps:Map>
</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 BaseViewModel. 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;
```

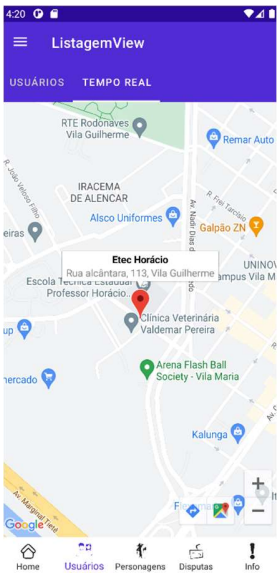
16. Abra a parte de código da ContentPage LocalizacaoView.xaml e defina a ligação com a classe ViewModel

```
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