



Aula 10 - Geolocalização

1. Abra classe **Models/Usuarios.cs** e confirme a existência das propriedades referente a geolocalização

```
public double? Latitude { get; set; }  
2 references  
public double? Longitude { get; set; }
```

2. Abra a classe **Request.cs** da pasta Services e adicione os métodos genéricos que vão consumir a API

- Método Put

```
public async Task<int> PutAsync<TResult>(string uri, TResult data, string token)  
{  
    HttpClient httpClient = new HttpClient();  
    httpClient.DefaultRequestHeaders.Authorization  
        = new AuthenticationHeaderValue("Bearer", token);  
  
    var content = new StringContent(JsonConvert.SerializeObject(data));  
    content.Headers.ContentType = new MediaTypeHeaderValue("application/json");  
    HttpResponseMessage response = await httpClient.PutAsync(uri, content);  
  
    string serialized = await response.Content.ReadAsStringAsync();  
  
    if (response.StatusCode == System.Net.HttpStatusCode.OK)  
        return int.Parse(serialized);  
    else  
        return 0;  
}
```

- Método Get

```
public async Task<TResult> GetAsync<TResult>(string uri, string token)  
{  
    HttpClient httpClient = new HttpClient();  
    httpClient.DefaultRequestHeaders.Authorization  
        = new AuthenticationHeaderValue("Bearer", token);  
  
    HttpResponseMessage response = await httpClient.GetAsync(uri);  
    string serialized = await response.Content.ReadAsStringAsync();  
    TResult result = await Task.Run(() =>  
        JsonConvert.DeserializeObject<TResult>(serialized));  
    return result;  
}
```



- Método Delete

```
public async Task<int> DeleteAsync(string uri, string token)
{
    HttpClient httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Bearer", token);
    HttpResponseMessage response = await httpClient.DeleteAsync(uri);
    string serialized = await response.Content.ReadAsStringAsync();
    if (response.StatusCode == System.Net.HttpStatusCode.OK)
        return int.Parse(serialized);
    else
        return 0;
}
```

3. Abra a classe **UsuarioService** e crie um construtor que receba um token como parâmetro conforme a sinalização. Ao final deste procedimento a classe terá dois construtores.

```
public UsuarioService()
{
    _request = new Request();
}

private string _token;
0 references
public UsuarioService(string token)
{
    _request = new Request();
    _token = token;
}
```

4. Abra a classe **UsuarioService** e adicione os métodos abaixo que vão atualizar a localização de um usuário, listar todos os usuários e buscar um usuário por id.

```
public async Task<int> PutAtualizarLocalizacaoAsync(Usuario u)
{
    string urlComplementar = "/AtualizarLocalizacao";
    var result = await _request.PutAsync(apiUrlBase + urlComplementar, u, _token);
    return result;
}

//using System.Collections.ObjectModel
public async Task<ObservableCollection<Usuario>> GetUsuariosAsync()
{
    string urlComplementar = string.Format("{0}", "/GetAll");
    ObservableCollection<Models.Usuario> listaUsuarios = await
_token);
    _request.GetAsync<ObservableCollection<Models.Usuario>>(apiUrlBase + urlComplementar,
return listaUsuarios;
}
```



5. Abra a classe **UsuarioViewModel** e antes do método **AutenticarUsuario**, adicione dois atributos para controlar a operação de coleta de geolocalização

```
private CancellationTokenSource _cancellationTokenSource;  
private bool _isCheckingLocation;
```

1 reference

```
public async Task AutenticarUsuario()//Método para autenticar um usuário  
{
```

6. Adicione a codificação para coletar a geolocalização do usuário e salvar a informação na API. A codificação deverá ficar antes da mensagem de boas-vindas. Será necessário deixar a localização do dispositivo ativada para os testes.

```
//Início da coleta de Geolocalização atual para Atualização na API  
_isCheckingLocation = true;  
_cancellationTokenSource = new CancellationTokenSource();  
GeolocationRequest request =  
    new GeolocationRequest(GeolocationAccuracy.Medium, TimeSpan.FromSeconds(10));  
  
Location location = await Geolocation  
    .Default.GetLocationAsync(request, _cancellationTokenSource.Token);  
  
Usuario uLoc = new Usuario();  
uLoc.Id = uAutenticado.Id;  
uLoc.Latitude = location.Latitude;  
uLoc.Longitude = location.Longitude;  
  
UsuarioService uServiceLoc = new UsuarioService(uAutenticado.Token);  
await uServiceLoc.PutAtualizarLocalizacaoAsync(uLoc);  
//Fim da coleta de Geolocalização atual para Atualização na API
```

```
await Application.Current.MainPage  
    .DisplayAlert("Informação", mensagem, "Ok");
```

```
Application.Current.MainPage = new AppShell();
```



-
7. Abra a classe **LocalizacaoViewModel.cs** criando o atributo do tipo **UsuarioService** que consumirá a API e construtor que vai ler o token das Preferences passando para a classe de serviço.

```
public class LocalizacaoViewModel : BaseViewModel
{
    private UsuarioService uService;
    1 reference
    public LocalizacaoViewModel()
    {
        string token = Preferences.Get("UsuarioToken", string.Empty);
        uService = new UsuarioService(token);
    }
}
```



8. Ainda na classe **LocalizacaoViewModel**, crie um método que busque os usuários na API e apresente no mapa

```
public async void ExibirUsuariosNoMapa()
{
    try
    {
        //using AppRpgEtec.Models
        ObservableCollection<Usuario> ocUsuarios = await uService.GetUsuariosAsync();
        List<Usuario> listaUsuarios = new List<Usuario>(ocUsuarios);
        Map map = new Map();

        foreach (Usuario u in listaUsuarios)
        {
            if (u.Latitude != null && u.Longitude != null)
            {
                double latitude = (double)u.Latitude;
                double logitude = (double)u.Longitude;
                Location location = new Location(latitude, logitude);

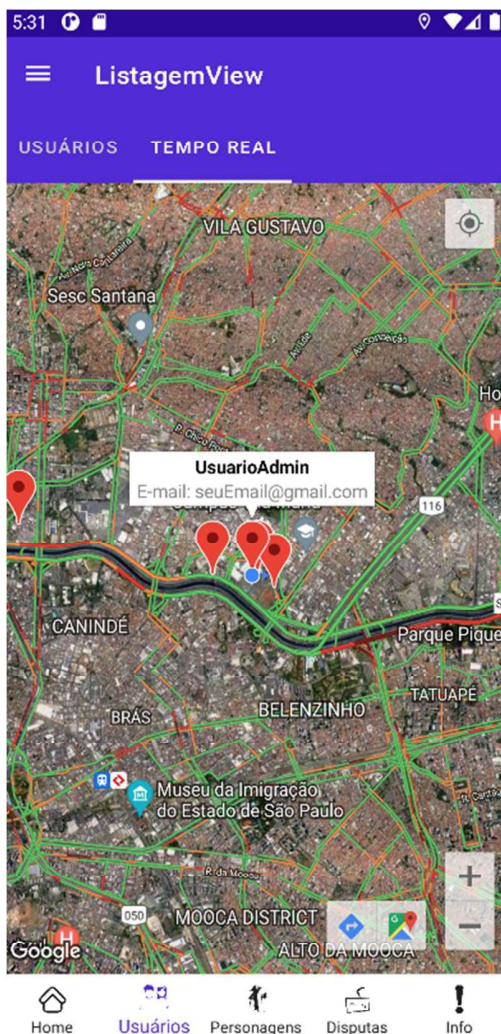
                Pin pinAtual = new Pin()
                {
                    Type = PinType.Place,
                    Label = u.Username,
                    Address = $"E-mail: {u.Email}",
                    Location = location
                };
                map.Pins.Add(pinAtual);
            }
        }
        MeuMapa = map;
    }
    catch (Exception ex)
    {
        await Application.Current.MainPage.DisplayAlert("Erro", ex.Message, "OK");
    }
}
```




9. Faça a chamada do método no construtor da view **LocalizacaoView.xaml.cs** no lugar do método que inicializava o mapa.

```
public LocalizacaoView()  
{  
    InitializeComponent();  
  
    viewModel = new LocalizacaoViewModel();  
    BindingContext = viewModel;  
    //viewModel.InicializarMapa();  
    viewModel.ExibirUsuariosNoMapa();  
}
```

- Execute o aplicativo para realizar os testes





Referências:

Microsoft Learning

Mapas:

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/controls/map>

Geolocalização:

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/device/geolocation?tabs=android>

Geocoding - Transcrição de endereço em pin ou de pin em endereço:

<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/device/geocoding?tabs=android>

Referências – Xamarin (Alguns componentes não funcionarão no MAUI)

https://www.youtube.com/watch?v=T9g_fkm7mhA

Plugin Permission:

<https://social.msdn.microsoft.com/Forums/en-US/8294f783-4182-4e3e-bff8-c80610331c81/need-a-fix-on-current-location-access-on-xamarinforms?forum=xamarinforms>

Google Maps:

<https://bertuzzi.medium.com/o-x-do-xamarin-forms-mapas-mas-com-google-maps-d-e9b57071b4ec>

<https://julianocustodio.com.br/2017/08/09/gps-xamarin-forms/>

Mapas com MVVM:

https://medium.com/@pedro_jesus/usando-mapas-em-xamarin-forms-com-mvvm-7f4cab769725

<https://stackoverflow.com/questions/28098020/bind-to-xamarin-forms-maps-map-from-viewmodel>

Geolocator:

<https://stackoverflow.com/questions/36630509/how-to-get-current-location-or-move-to-current-location-in-xamarin-forms-map>

<http://rsamorim.azurewebsites.net/2018/03/01/xamarin-forms-capturando-mudanca-de-localizacao/>

Aplicação com exemplos do Plugin de Mapas:

<https://github.com/amay077/Xamarin.Forms.GoogleMaps/tree/master/XFGoogleMapSample>

Aplicação clone do Uber:

<https://www.xamboy.com/2019/07/03/introducing-xuber-uber-clone-using-xamarin-forms/>