




Aula 11 – Programação de Dispositivos Móveis

Professor: Me. Wellington Tuler Moraes
Curso: ADS – IFSP – Campus Cubatão

Adicionando Mapas ao Xamarin.Forms

- O Xamarin.Forms.Maps utiliza APIs nativas de mapas para cada plataforma.
- Isto provê rápida e familiar experiência ao usuário.
- Reflete em alguns passos de configuração para cada API.
- Uma vez configurado, o controle de Mapa trabalha como qualquer outro elemento de código comum no Xamarin.Forms.

Passos para inicialização dos Mapas

- Inicialização dos Mapas: Usando Map, código de inicialização adicional é necessário.
- Configuração de Plataforma: Cada plataforma requer alguma configuração para trabalhar com mapas.
- Mapas em C#: Mostra mapas e pinos em C#.
- Mapas em XAML: Mostra mapas usando XAML.

Exemplo do uso de Mapas



Inicialização dos Mapas

- Os mapas para uma aplicação Xamarin.Forms são pacotes separados (**Xamarin.Forms.Maps**) que devem ser adicionados usando o NuGet package em cada solução.
- No Android este pacote também tem uma dependência do GooglePlayServices (outro NuGet) que é baixado automaticamente quando adicionado o Xamarin.Forms.Maps.

Inicialização do Maps

- Após instalado o pacote NuGet, é necessário chamar o método de inicialização, que na plataforma Android se dá da seguinte forma:
 - `Xamarin.FormsMaps.Init(this, bundle);`
- Deve-se colocar o código dentro da classe `MainActivity.cs` no método `OnCreate`.
- Após adicionar o pacote no método de inicialização de cada aplicação a API poderá ser usada no projeto PCL ou Shared.

Configuração da Plataforma – Android

- Para usar a API 2 do Google Maps no Android, devemos gerar uma API Key e adicionar em nosso projeto Android.
- Após seguir os passos para obter a API, coloque no arquivo Properties/AndroidManifest.xml:

```
<meta-data android:name="com.google.android.maps.v2.API_KEY"
    android:value="AbCdEfGhIjKlMnOpQrStUvWxValueGoesHere" />
```

Configuração da Plataforma – Android

- Sem uma API Key válida os controles de mapa apresentarão uma caixa cinza no Android.
- Lembre-se de gerar outra chave usando o arquivo Keystore que será usado para publicar a versão de alguma aplicação adicionada ao Google Play Store.
- A chave que você gerou para desenvolvimento e debugging não trabalhará em um app baixado do Play Store.

Configuração da Plataforma – Android

- Você também precisará ativar as permissões necessárias com um clique direto de botão no projeto Android e selecionando:
- Propriedades => Manifesto Android => Permissões Necessárias:

- AccessCoarseLocation
- AccessFineLocation
- AccessLocationExtraCommands
- AccessMockLocation
- AccessNetworkState
- AccessWifiState
- Internet

some of these are shown in the screenshot below:

required permissions	
<input checked="" type="checkbox"/> AccessCoarseLocation	
<input checked="" type="checkbox"/> AccessFineLocation	
<input checked="" type="checkbox"/> AccessLocationExtraCommands	

Usando Maps

```
public class MapPage : ContentPage {
    public MapPage() {
        var map = new Map(
            MapSpan.FromCenterAndRadius(
                new Position(37,-122), Distance.FromMiles(0.3)))
        {
            IsShowingUser = true,
            HeightRequest = 100,
            WidthRequest = 960,
            VerticalOptions = LayoutOptions.FillAndExpand
        };
        var stack = new StackLayout { Spacing = 0 };
        stack.Children.Add(map);
        Content = stack;
    }
}
```

Tipo de Mapa

- O conteúdo do mapa pode ser alterado através da propriedade MapType, para mostrar ruas, imagem de satélite ou a combinação de ambos.
 - map.MapType == MapType.Street;
- Valores válidos são:
 - Hybrid
 - Satellite
 - Street (o padrão).

Map Region e MapSpan

- Como mostrado no código anterior, a instância de MapSpan mostra a visão inicial (ponto central e level de zoom) de um mapa quando carregado.
- O método MoveToRegion na classe Map pode ser utilizada para mudar a posição do zoom do mapa. Existem duas formas de criar uma instância de MapSpan:
 - MapSpan.FromCenterAndRadius() => método estático que cria uma posição específica.
 - New MapSpan() => usa a posição em graus da latitude e longitude.

- Para mudar o nível do zoom no mapa sem alterar a localização, devemos criar um novo MapSpan usando a localização corrente da propriedade: VisibleRegion.Center.



Map Pin

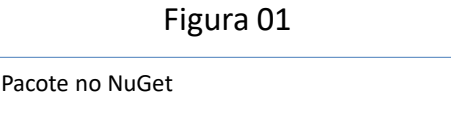
- Localizações podem ser marcadas com objetos do tipo Pin:

```
var position = new Position(37,-122); // Latitude, Longitude
var pin = new Pin {
    Type = PinType.Place,
    Position = position,
    Label = "custom pin",
    Address = "custom detail info"
};
map.Pins.Add(pin);
```

- Localizações podem ser marcadas com objetos do tipo Pin:

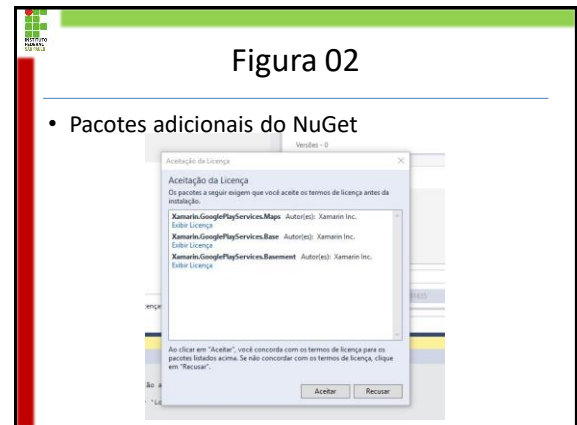
Figura 01

- Pacote no NuGet

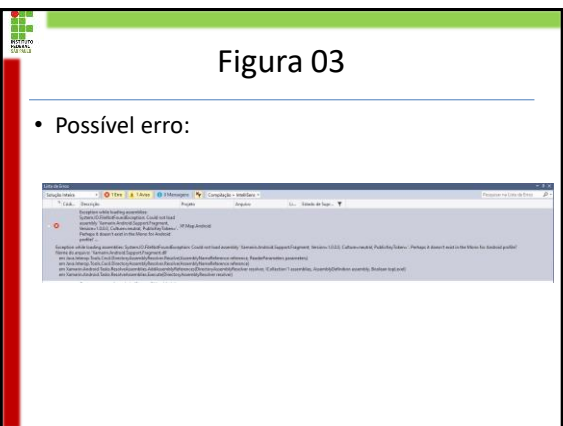


The screenshot displays the Visual Studio NuGet Package Manager window. On the left, the 'Installed' tab shows a list of packages. The right pane shows the details for the selected package, 'Xamarin.Forms.Maps'. The details include the version (1.0.0.0), license (MIT), and a link to the source code. The package is listed as 'Xamarin.Forms.Maps (v1.0.0.0)' and is described as 'Maps control and services for Xamarin.Forms'.

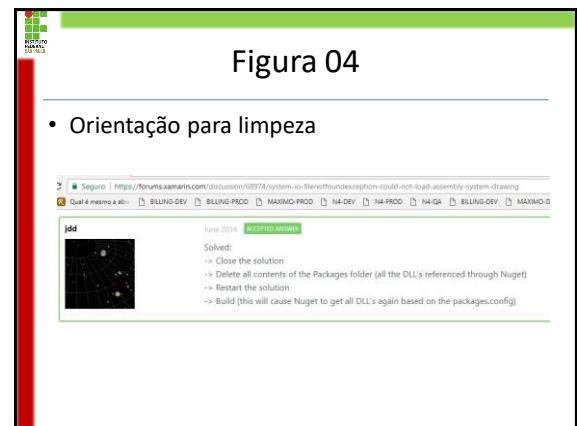
- Pacote no NuGet



- Pacotes adicionais do NuGet



- Possível erro:



- Orientação para limpeza

