

Laboratório

Conectando ao Azure Mobile Apps

Versão: 1.0.0
Dezembro de 2016



[Miguel Muñoz Serafín](#)
@msmdotnet



CONTEÚDO

INTRODUÇÃO

EXERCÍCIO 1: CONECTANDO A APLICAÇÃO XAMARIN.FORMS AO AZURE MOBILE APPS

Tarefa 1. Criar e hospedar o servidor back-end para a aplicação Xamarin.Forms.

Tarefa 2. Importar os dados de exemplo.

Tarefa 3. Adicionar o pacote NuGet **Microsoft.Azure.Mobile.Client** a solução.

Tarefa 4. Modificar a classe **Cat**.

Tarefa 5. Adicionar o código para consumir os dados do back-end.

Tarefa 6. Modificar a classe **Repository**.

RESUMO

Introdução

Claro que ser capaz de consumir dados de um serviço RESTful é ótimo, mas e quanto ao consumo de dados de um aplicativo Back-end? Este é o lugar onde **Azure Mobile Apps** entra em ação. Neste laboratório atualizaremos nossa aplicação Xamarin.Forms para utilizar uma aplicação back-end **Azure Mobile Apps**.

Objetivos

Ao finalizar este laboratório, os participantes serão capazes de:

- Criar uma aplicação back-end **Azure Mobile Apps**.
- Consumir os dados do back-end de uma aplicação Xamarin.Forms utilizando o pacote NuGet **Microsoft.Azure.Mobile.Client**.

Requisitos

Para a realização deste laboratório é necessário contar com o seguinte:

- Um equipamento de desenvolvimento com o Windows 10 e Visual Studio 2015 Community, Professional ou Enterprise com a plataforma Xamarin.
- Um equipamento Mac com a plataforma Xamarin.
- Uma conexão de Internet.
- Uma assinatura Microsoft Azure.
- O arquivo [Cats.csv](#) com dados de exemplo.

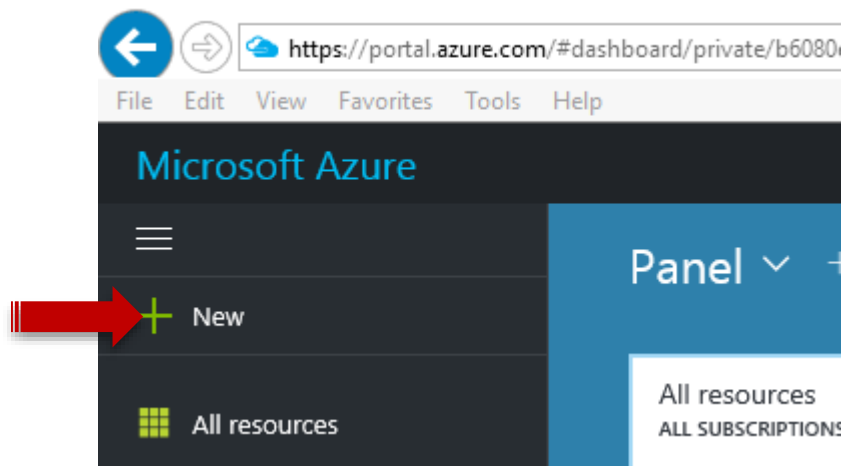
Tempo estimado para completar este laboratório: **60 minutos**.

Exercício 1: Conectando a aplicação Xamarin.Forms ao Azure Mobile Apps.

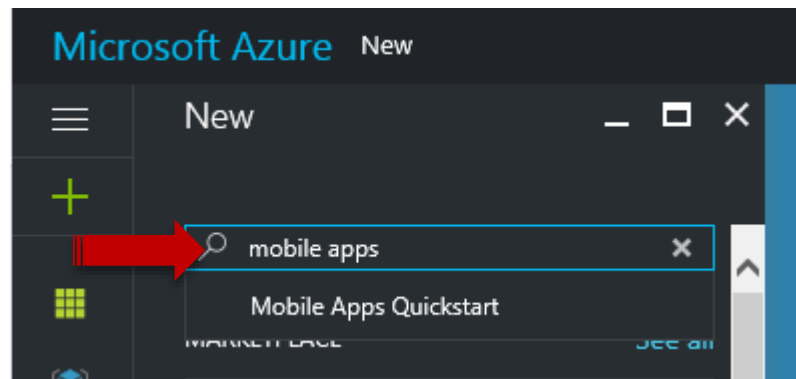
Neste exercício criaremos um servidor back-end utilizando o serviço **Azure Mobile Apps**. Em seguida, importará os dados de exemplo e modificará o aplicativo Xamarin.Forms para consumir os dados expostos pelo servidor back-end criado.

Tarefa 1. Criar e hospedar o servidor back-end para a aplicação Xamarin.Forms.

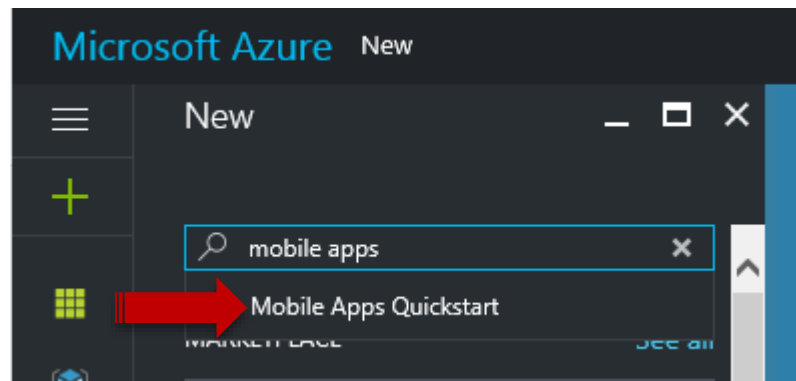
1. Acesse o portal de gerenciamento **Microsoft Azure** no seguinte endereço:
<http://portal.azure.com>.
2. Clique na opção **+ New**.



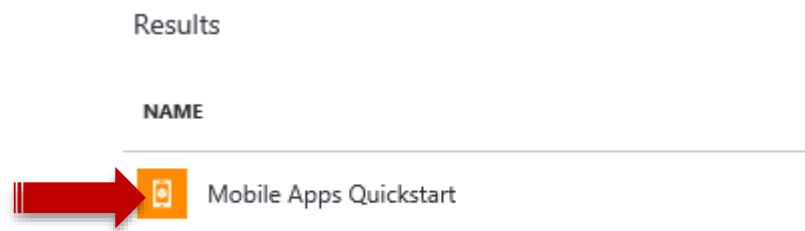
3. No quadro de busca escreva **mobile apps**.



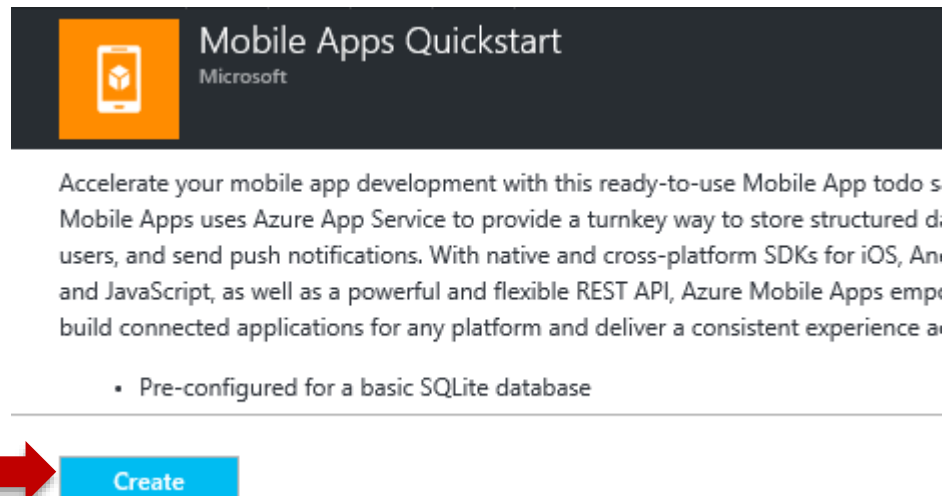
4. Na lista de resultado, clique em **Mobile Apps Quickstart**.



5. Na guia **Results**, clique em **Mobile Apps Quickstart**.



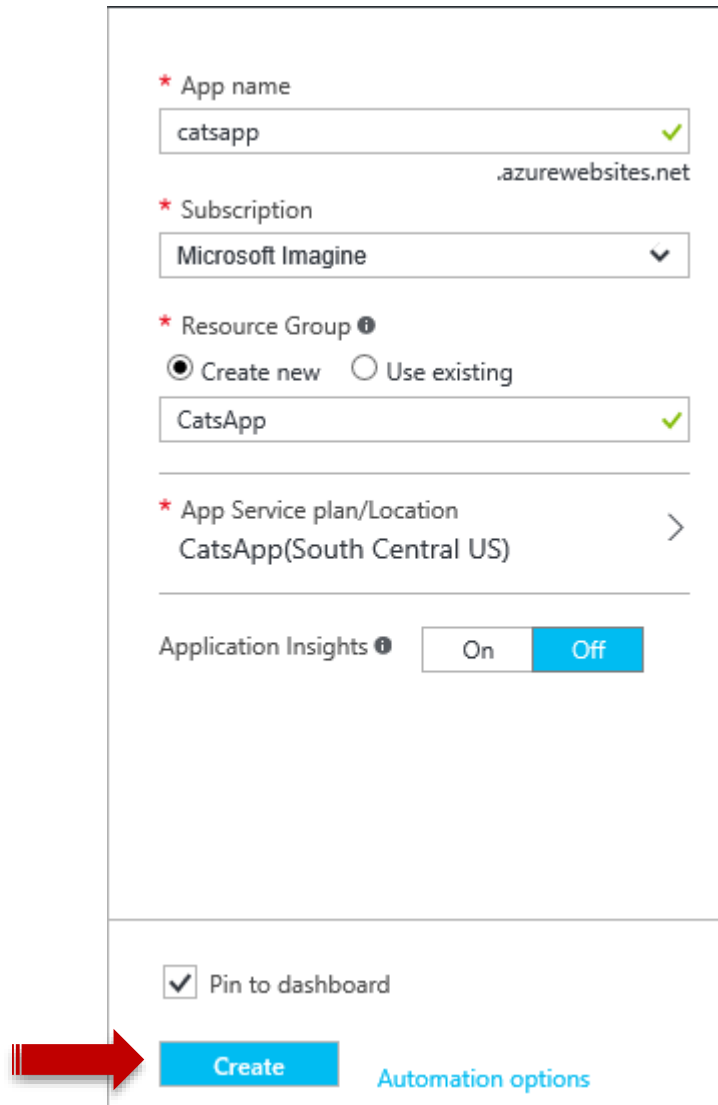
6. Na guia **Mobile Apps Quickstart** clique em **Create**.



7. Na nova guia **Mobile Apps Quickstart** forneça as seguintes informações:
- **App name.** Este é um nome exclusivo para o aplicativo que você precisa quando você configurar o back-end da aplicação Xamarin.Forms. Você precisa selecionar um nome exclusivo global.
 - **Subscription.** Escolha uma assinatura Microsoft Azure.

- **Resource Group.** Selecione **Create new** e nomeie **CatsApp**. Um grupo de recursos é um grupo de serviços relacionados que podem ser facilmente removidos mais tarde.
- **App Service plan/Location.** Clique neste campo e selecione **Create New**, forneça um nome único, por exemplo, **CatsApp**, selecione um local (geralmente deve selecionar o local mais próximo aos clientes do aplicativo), selecione o plano **F1 Free** e clique em **OK**.

8. Marque a caixa **Pin to dashboard** e clique em **Create** para criar a aplicação.



* App name
catsapp ✓
.azurewebsites.net


* Subscription
Microsoft Imagine ▼

* Resource Group ⓘ
☒ Create new ☐ Use existing
CatsApp ✓

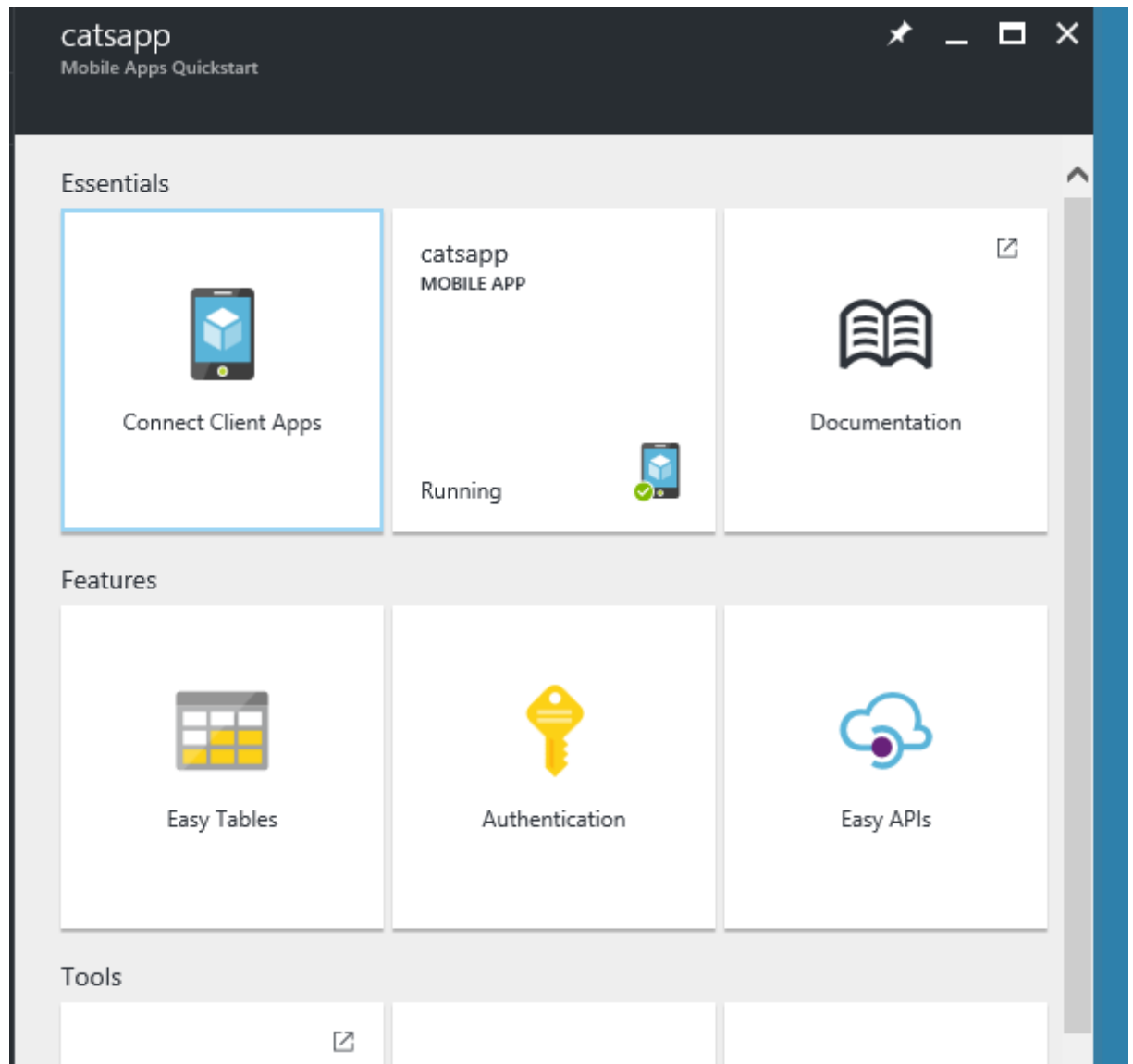
* App Service plan/Location
CatsApp(South Central US) >

Application Insights ⓘ

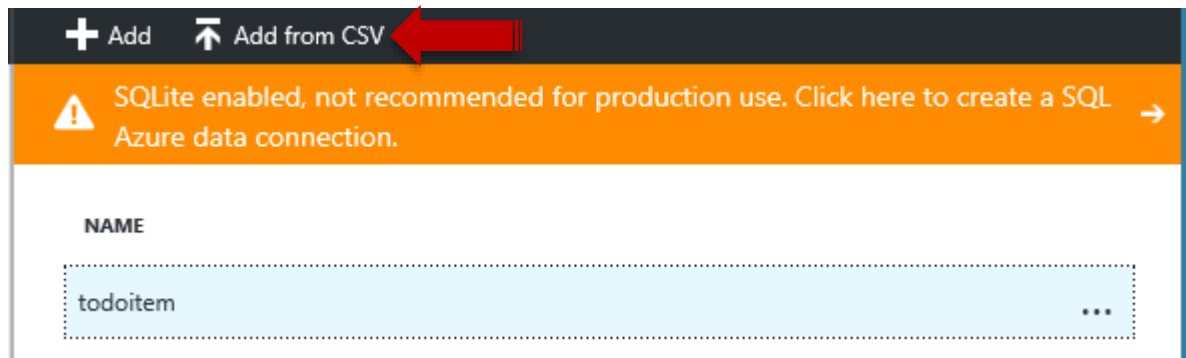
☒ Pin to dashboard

 [Automation options](#)

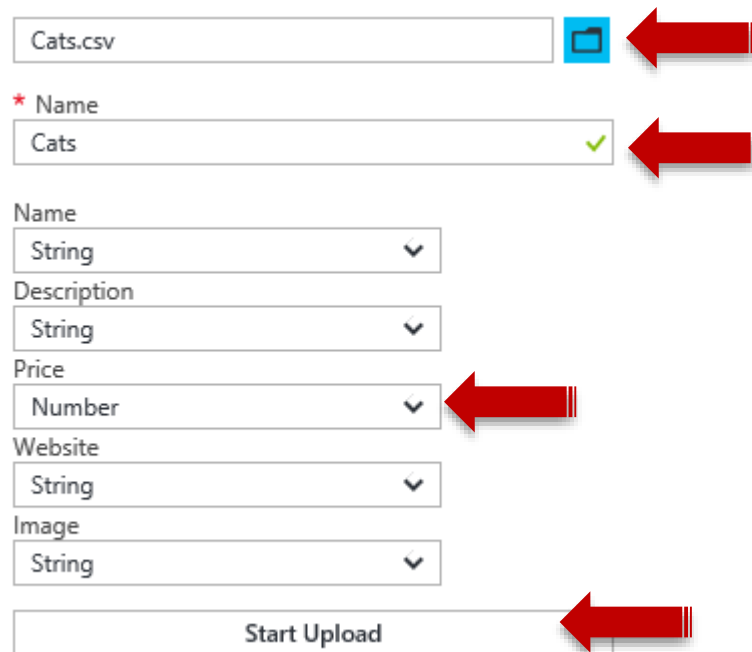
O processo de criação da aplicação levará de 3 a 5 minutos. Quando o processo de criação estiver completo, você verá uma tela semelhante a seguinte.

**Tarefa 2. Importar os dados de exemplo.**

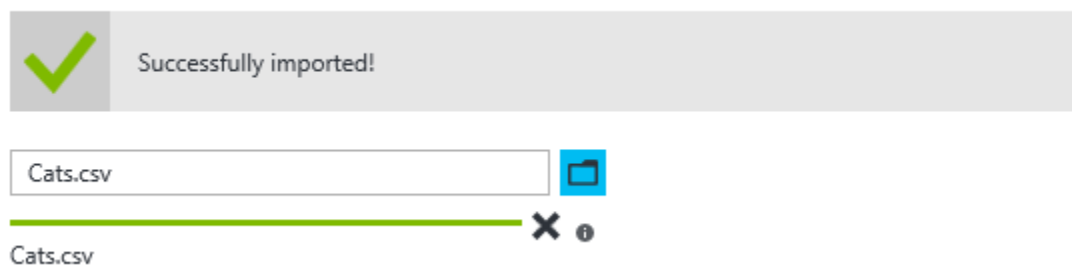
1. Na seção **Features**, clique em **Easy Tables**. Você verá a tabela **TodoItem**. Vamos criar uma nova tabela e subir dados de exemplo.
2. Selecione a opção **Add from CSV**.



3. Selecione o arquivo **Cats.csv**. Isto irá adicionar o nome da nova tabela e a lista de campos.
4. Verifique se o nome da tabela é **Cats**.
5. Altere o tipo de dados do campo **Price** de tipo **Number**.
6. Selecione **Start Upload** para importar os dados.



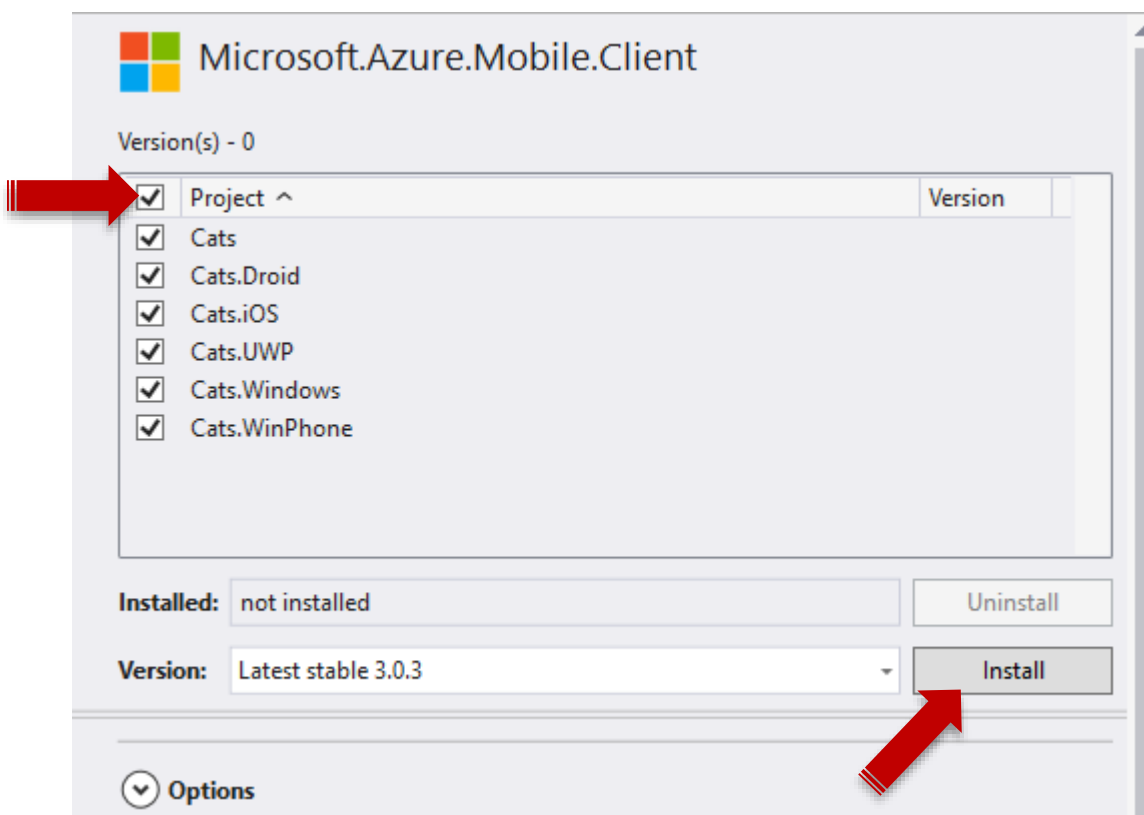
No final do processo será mostrada uma mensagem similar a seguinte.



Tarefa 3. Adicione o pacote NuGet Microsoft.Azure.Mobile.Client à solução.

Nesta tarefa você adicionará à solução o pacote NuGet **Microsoft.Azure.Mobile.Client** que contém as classes necessárias para poder consumir os dados da aplicação Back-end.

1. Selecione a opção **Manage NuGet Packages for Solution...** do menu contextual da solução.
2. Selecione o arquivo **Browse**.
3. Na caixa de pesquisa escreva **Microsoft.Azure.Mobile.Client**.
4. Na lista de resultados selecione **Microsoft.Azure.Mobile.Client**.
5. No painel direito, selecione a caixa **Project** para indicar que o pacote deve ser instalado em todos os projetos e clique em **Install**.



6. Aceite as mudanças e acordo de licenciamento.
7. Se solicitado, reinicie o Visual Studio.

Tarefa 4. Modifique a classe Cat.

É necessário fazer algumas alterações ao modelo **Cat** para poder usar durante o processo de recuperação de dados do serviço back-end.

1. Abra o arquivo **Cat.cs** localizado na pasta **Models** do projeto PCL.

2. Adicione o seguinte código no início do arquivo para importar o espaço de nomes **Microsoft.WindowsAzure.MobileServices**.

```
using Microsoft.WindowsAzure.MobileServices;
```

3. Adicione o atributo **DataTable** para a classe **Cat** para indicar o nome da tabela que será utilizada para obter os dados.

```
[DataTable("Cats")]  
public class Cat
```

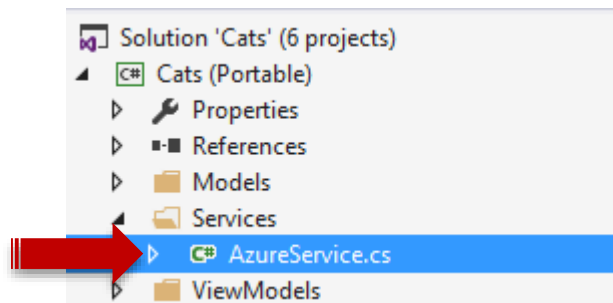
4. Adicione o seguinte código para a classe **Cat** para definir a propriedade que permitirá levar o controle de concorrência de cada registro da tabela.

```
[Version]  
public string AzureVersion { get; set; }
```

Tarefa 5. Adicione código para consumir os dados do back-end.

Nesta tarefa adicionará para a classe **AzureService** com o código que permitirá obter os dados de uma tabela do back-end **Azure Mobile Apps**.

1. Criar um novo diretório chamado **Services** na raiz do projeto PCL.
2. Dentro do diretório **Services**, adiciona uma nova classe chamada **AzureService**.



3. No início do **AzureService.cs** adiciona o seguinte código para importar o espaço de nomes **Microsoft.WindowsAzure.MobileServices**.

```
using Microsoft.WindowsAzure.MobileServices;
```

4. Modifique a definição da classe para que aceite um parâmetro de tipo e que seja pública

```
public class AzureService<T>  
{  
  
}
```

5. Adicione para a classe **AzureService** o seguinte código para definir uma variável que armazene uma instância de **IMobileServiceClient**.

```
IMobileServiceClient Client;
```

6. Adicione para a classe **AzureService** o seguinte código para definir uma variável que armazene uma instância de **IMobileServiceTable** representando uma mesa de back-end.

```
IMobileServiceTable<T> Table;
```

7. Adicione o seguinte código para definir o construtor da classe **AzureService**. O código inicia o cliente do back-end e a instância **IMobileServiceTable** para poder realizar operações com o quadro do back-end. Certifique-se de atribuir a URL de seu **Azure Mobile Apps** na variável **MyAppServiceURL**.

```
public AzureService()  
{  
    string MyAppServiceURL = "http://tuappname.azurewebsites.net";  
    Client = new MobileServiceClient(MyAppServiceURL);  
    Table = Client.GetTable<T>();  
}
```

8. Adicione à classe **AzureService** o seguinte código que permitirá obter os dados de uma tabela como uma coleção **IEnumerable**.

```
public Task<IEnumerable<T>> GetTable()  
{  
    return Table.ToEnumerableAsync();  
}
```

Tarefa 6. Modificar a classe Repository.

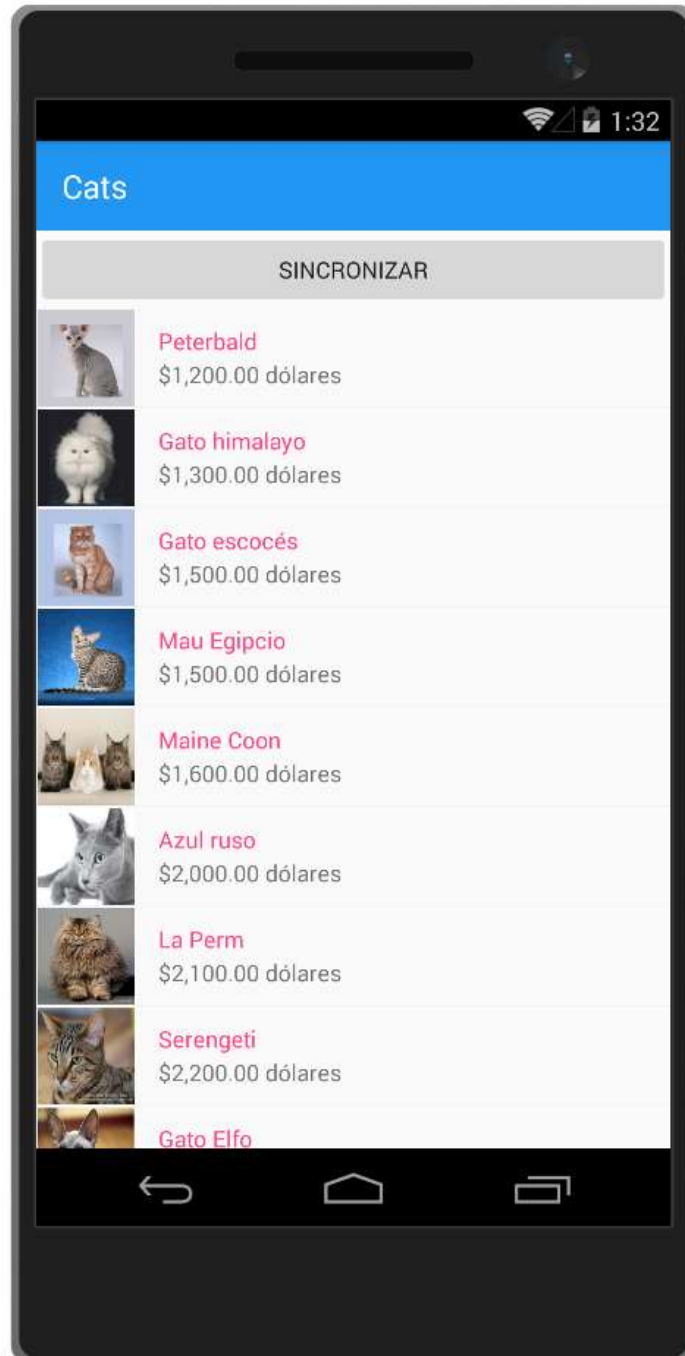
Nesta tarefa, você irá modificar a classe **Repository** para que possa obter os dados de back-end **Azure Mobile Apps** no lugar do serviço RESTful.

1. Abra o arquivo **Repository.cs**.
2. Substitua o código do método **GetCats** pelo seguinte.

```
public async Task<List<Cat>> GetCats()  
{  
    var Service = new Services.AzureService<Cat>();  
    var Items = await Service.GetTable();  
    return Items.ToList();  
}
```

3. Execute a aplicação no emulador Android.

4. Toque no botão **Sincronizar**. Uma tela similar a seguinte será mostrada. Agora os dados estão sendo consumidos a partir do back-end **Azure Mobile Apps**.



5. Pare o aplicativo e retornar ao Visual Studio.
6. Teste o aplicativo em outras plataformas.

Resumo

Neste laboratório você modificou a aplicação Xamarin.Forms para consumir os dados de uma aplicação back-end hospedada em um **Azure Mobile Apps** de Microsoft Azure.

Quando tiver finalizado este laboratório, publique a seguinte mensagem no Twitter e Facebook:

Acabei o #Lab05 da #MaratonaXamarin e conheço a forma de consumo de um serviço back-end de Azure Mobile Apps a partir de aplicações Xamarin.Forms!