

# Laboratorio

---

## *Conectando a Azure Mobile Apps*

Versión: 1.0.0  
Diciembre de 2016



[Miguel Muñoz Serafín](#)  
@msmdotnet



## CONTENIDO

### INTRODUCCIÓN

#### EJERCICIO 1: CONECTANDO LA APLICACIÓN XAMARIN.FORMS A AZURE MOBILE APPS

Tarea 1. Crear y hospedar el servidor back-end para la aplicación Xamarin.Forms.

Tarea 2. Importar los datos de ejemplo.

Tarea 3. Agregar el paquete NuGet **Microsoft.Azure.Mobile.Client** a la solución.

Tarea 4. Modificar la clase **Cat**.

Tarea 5. Agregar el código para consumir los datos del back-end.

Tarea 6. Modificar la clase **Repository**.

### RESUMEN

# Introducción

---

Por supuesto que ser capaz de consumir datos desde un servicio RESTful es genial, pero ¿Qué hay acerca de consumir datos desde una aplicación Back-end? Aquí es donde **Azure Mobile Apps** entra en acción. En este laboratorio actualizaremos nuestra aplicación Xamarin.Forms para utilizar una aplicación back-end **Azure Mobile Apps**.

## Objetivos

Al finalizar este laboratorio, los participantes serán capaces de:

- Crear una aplicación back-end **Azure Mobile Apps**.
- Consumir los datos del back-end desde una aplicación Xamarin.Forms utilizando el paquete NuGet **Microsoft.Azure.Mobile.Client**.

## Requisitos

Para la realización de este laboratorio es necesario contar con lo siguiente:

- Un equipo de desarrollo con sistema operativo Windows 10 y Visual Studio 2015 Community, Professional o Enterprise con la plataforma Xamarin.
- Un equipo Mac con la plataforma Xamarin.
- Una conexión a Internet.
- Una suscripción Microsoft Azure.
- El archivo [Cats.csv](#) con datos de ejemplo.

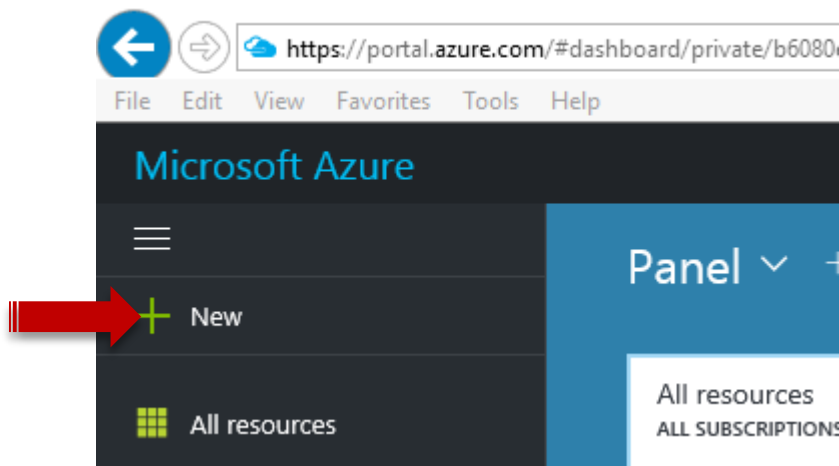
Tiempo estimado para completar este laboratorio: **60 minutos**.

# Ejercicio 1: Conectando la aplicación Xamarin.Forms a Azure Mobile Apps.

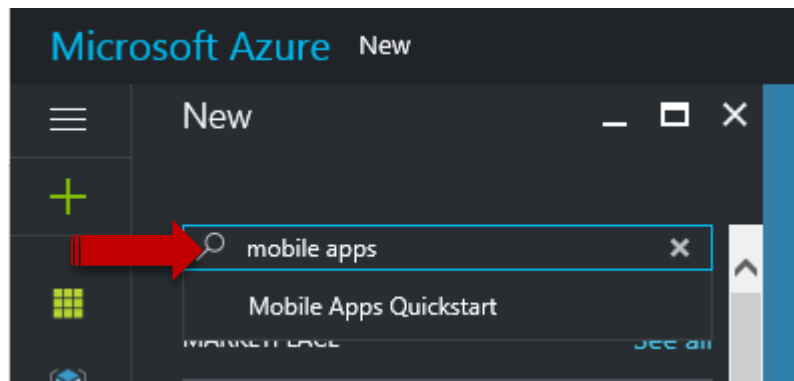
En este ejercicio crearás un servidor back-end utilizando el servicio **Azure Mobile Apps**. Posteriormente importarás datos de ejemplo y modificarás la aplicación Xamarin.Forms para consumir los datos expuestos por el servidor back-end creado.

## Tarea 1. Crear y hospedar el servidor back-end para la aplicación Xamarin.Forms.

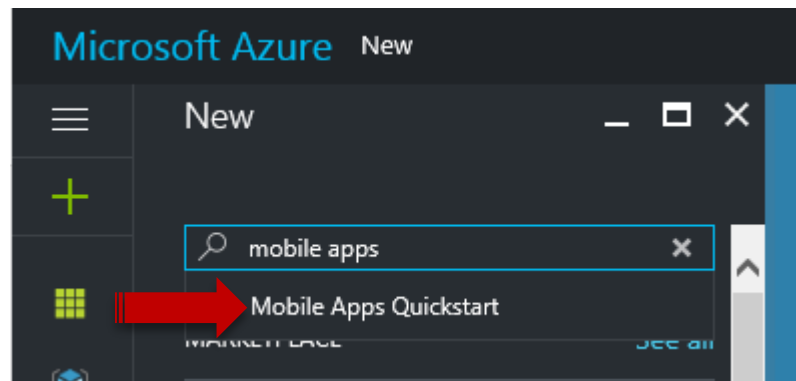
1. Accede al portal de administración de **Microsoft Azure** en la siguiente dirección: <http://portal.azure.com>.
2. Haz clic en la opción **+ New**.



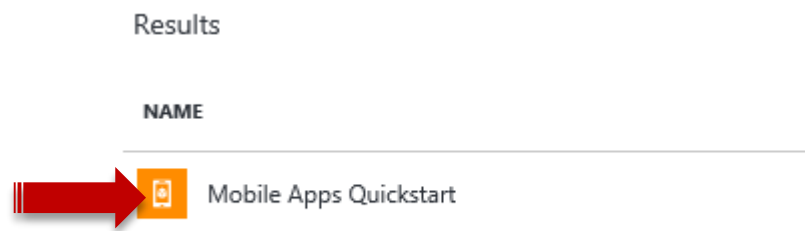
3. En el cuadro de búsqueda escribe **mobile apps**.



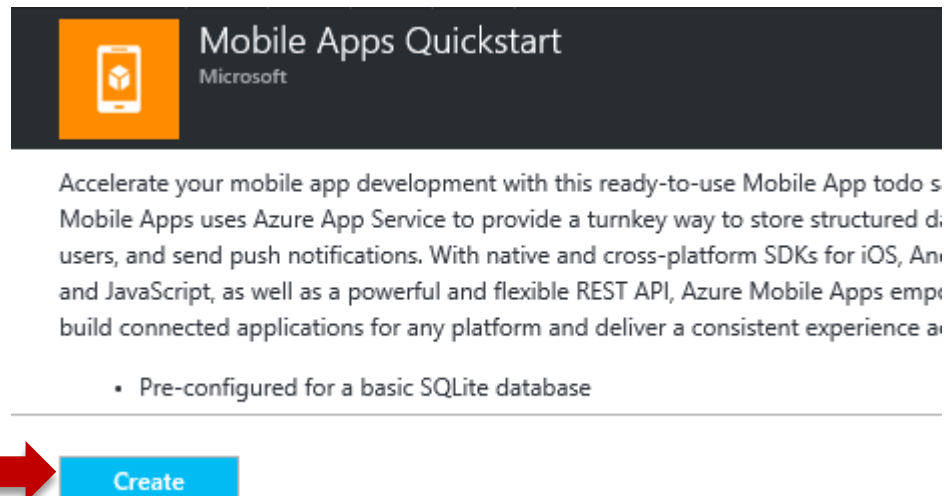
4. En la lista de resultado, haz clic en **Mobile Apps Quickstart**.



5. En la ficha **Results**, haz clic en **Mobile Apps Quickstart**.



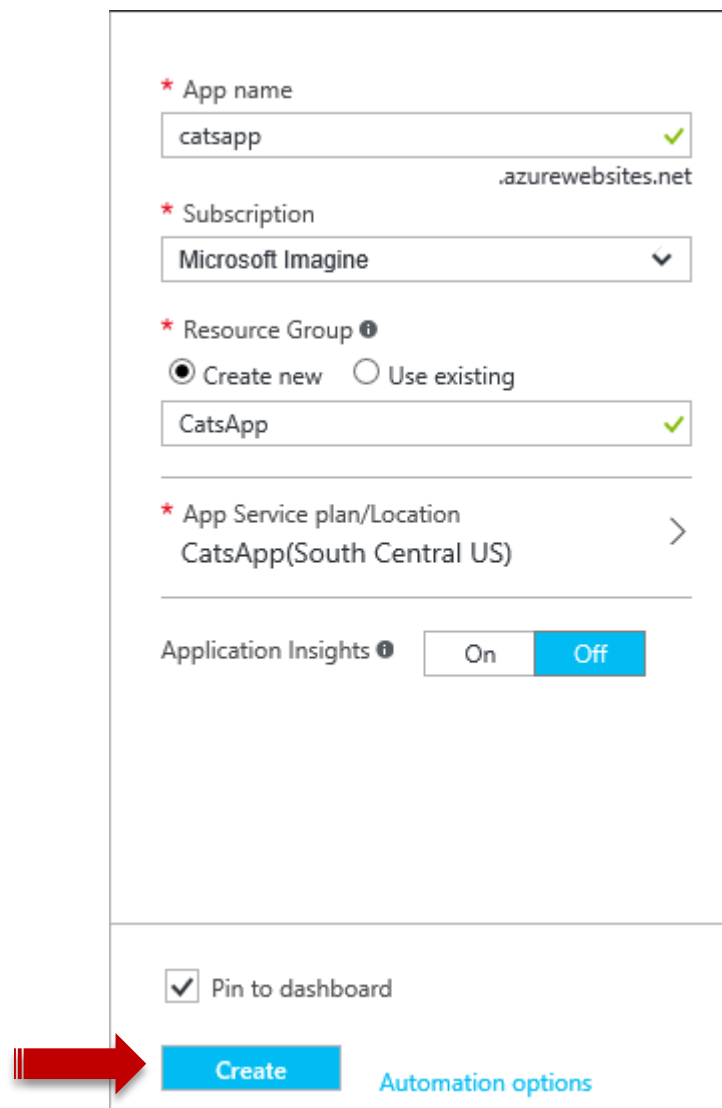
6. En la ficha **Mobile Apps Quickstart** haz clic en **Create**.



7. En la nueva ficha **Mobile Apps Quickstart** proporciona la siguiente información:
- **App name.** Este es un nombre único para la aplicación que necesitarás cuando configures el back-end de la aplicación Xamarin.Forms. Necesitarás seleccionar un nombre único global.
  - **Subscription.** Selecciona una suscripción Microsoft Azure.

- **Resource Group.** Selecciona **Create new** y nómbralo **CatsApp**. Un grupo de recursos es un grupo de servicios relacionados que posteriormente puede ser eliminado fácilmente.
- **App Service plan/Location.** Haz clic en ese campo y selecciona **Create New**, proporciona un nombre único, por ejemplo, **CatsApp**, selecciona una ubicación (típicamente deberías seleccionar la ubicación más cercana a los clientes de la aplicación), selecciona el plan **F1 Free** y haz clic en **OK**.

8. Selecciona la casilla **Pin to dashboard** y haz clic en **Create** para crear la aplicación.



\* App name  
catsapp ✓  
.azurewebsites.net

\* Subscription  
Microsoft Imagine ▼

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

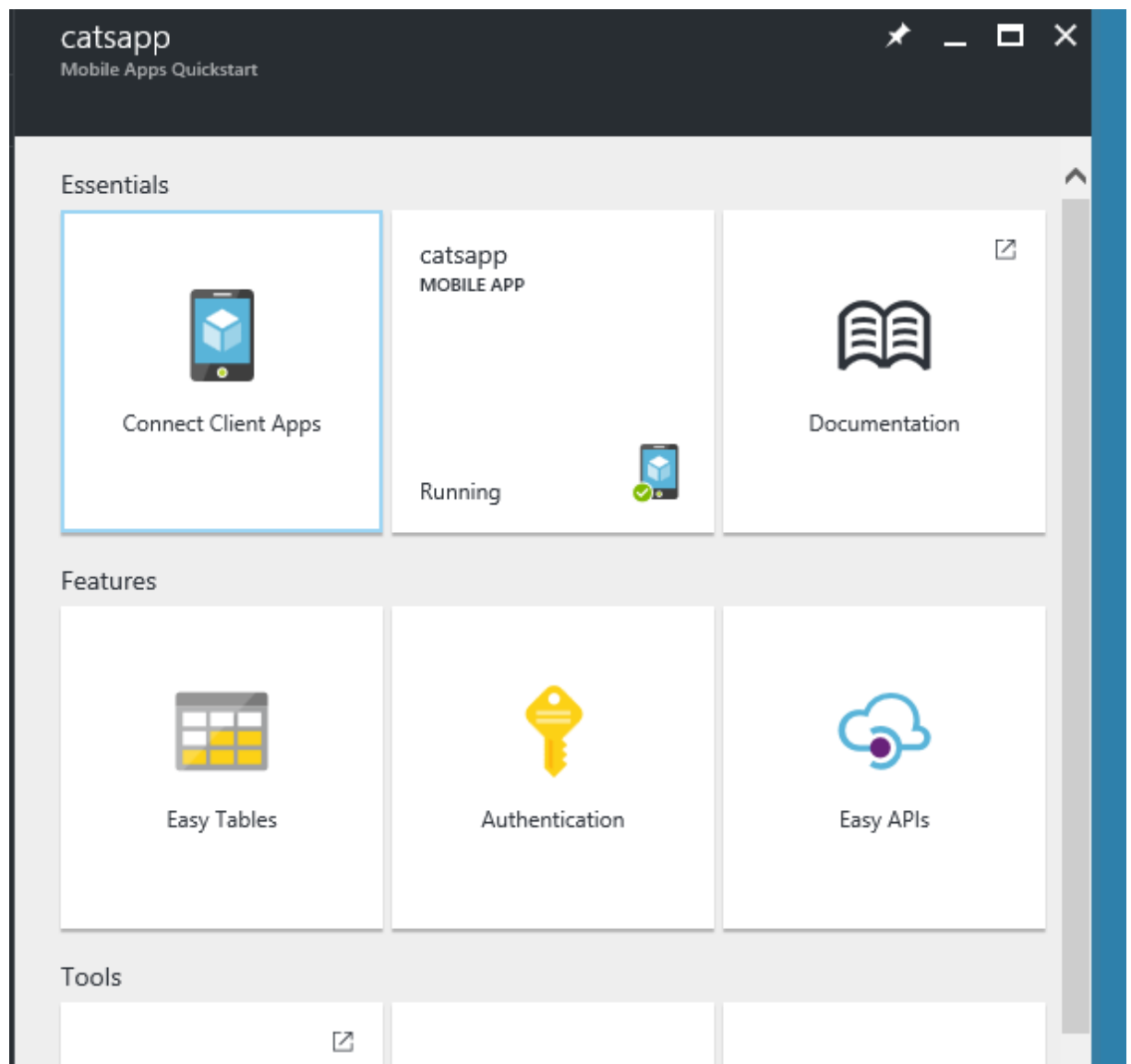
\* App Service plan/Location  
CatsApp(South Central US) >

Application Insights ⓘ

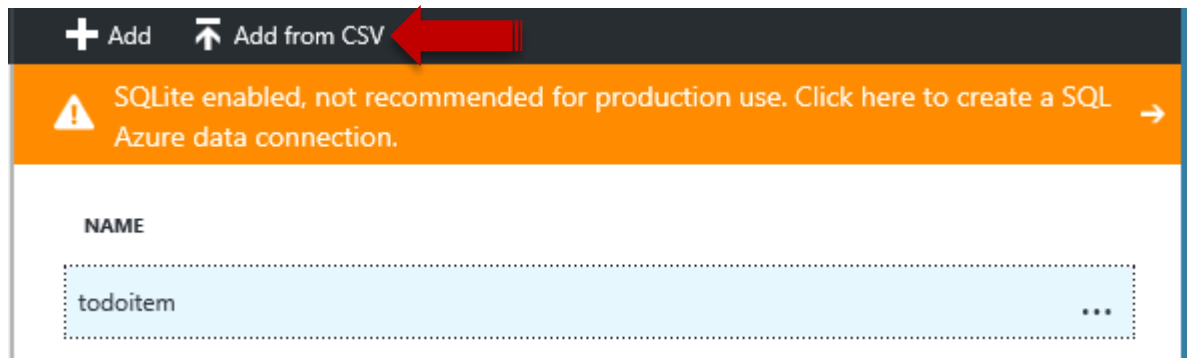
☒ Pin to dashboard

[Automation options](#)

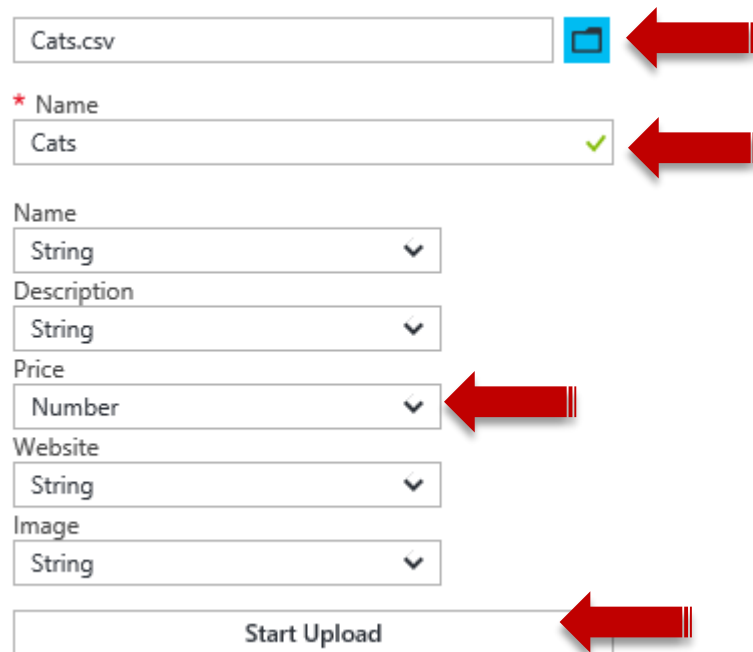
El proceso de creación de la aplicación tardará entre 3 y 5 minutos. Cuando el proceso de creación finalice podrás observar una pantalla similar a la siguiente.

**Tarea 2. Importar los datos de ejemplo.**

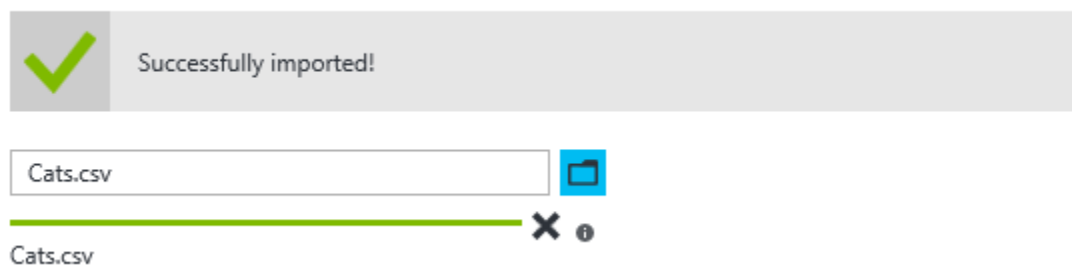
1. En la sección **Features**, haz clic en **Easy Tables**. Podrás ver la tabla creada **TodoItem**. Nosotros crearemos una nueva tabla y subiremos datos de ejemplo.
2. Selecciona la opción **Add from CSV**.



3. Selecciona el archivo **Cats.csv**. Esto agregará el nombre de la nueva tabla y la lista de campos.
4. Verifica que el nombre de la tabla sea **Cats**.
5. Cambia el tipo de datos del campo **Price** al tipo **Number**.
6. Selecciona **Start Upload** para importar los datos.



Al finalizar el proceso se mostrará un mensaje similar al siguiente.

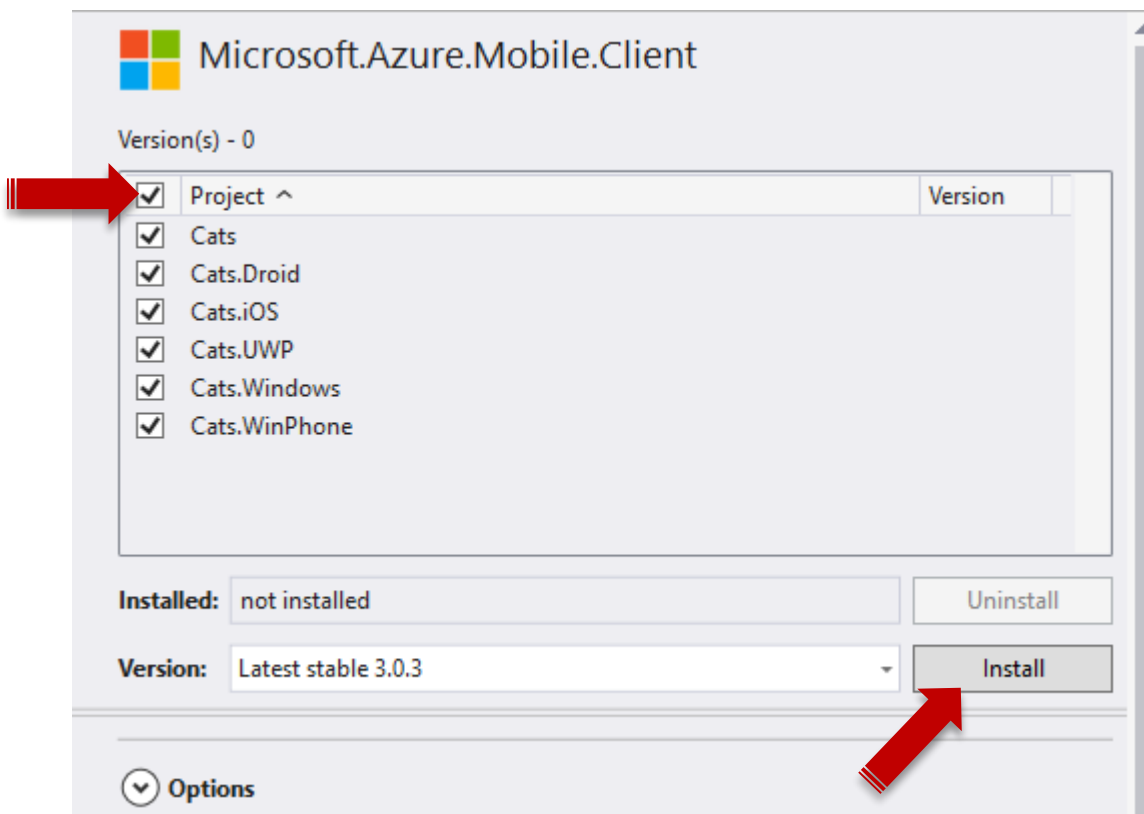




### Tarea 3. Agregar el paquete NuGet Microsoft.Azure.Mobile.Client a la solución.

En esta tarea agregarás a la solución el paquete NuGet **Microsoft.Azure.Mobile.Client** que contiene las clases necesarias para poder consumir los datos de la aplicación Back-end.

1. Selecciona la opción **Manage NuGet Packages for Solution...** del menú contextual de la solución.
2. Selecciona la ficha **Browse**.
3. En el cuadro de búsqueda escribe **Microsoft.Azure.Mobile.Client**.
4. En la lista de resultados selecciona **Microsoft.Azure.Mobile.Client**.
5. En el panel de la derecha selecciona la casilla **Project** para indicar que el paquete debe instalarse en todos los proyectos y haz clic en **Install**.



6. Acepta los cambios y el acuerdo de licencias.
7. En caso de que te sea solicitado, reinicia Visual Studio.

### Tarea 4. Modificar la clase Cat.

Es necesario hacer algunos cambios al modelo **Cat** para poder utilizarla durante el proceso de recuperación de datos desde el servicio back-end.

1. Abre el archivo **Cat.cs** ubicado en la carpeta **Models** del proyecto PCL.
2. Agrega el siguiente código al inicio del archivo para importar el espacio de nombres **Microsoft.WindowsAzure.MobileServices**.

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

3. Agrega el atributo **DataTable** a la clase **Cat** para indicar el nombre de la tabla que será utilizada para obtener los datos.

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

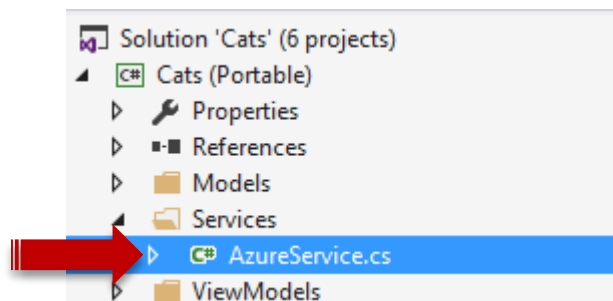
4. Agrega el siguiente código a la clase **Cat** para definir la propiedad que permitirá llevar el control de concurrencia de cada registro de la tabla.

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

#### Tarea 5. Agregar el código para consumir los datos del back-end.

En esta tarea agregarás la clase **AzureService** con el código que permitirá obtener los datos de una tabla del back-end **Azure Mobile Apps**.

1. Crea un nuevo directorio llamado **Services** en la raíz del proyecto PCL.
2. Dentro del directorio **Services**, agrega una nueva clase llamada **AzureService**.



3. Al inicio del archivo **AzureService.cs** agrega el siguiente código para importar el espacio de nombres **Microsoft.WindowsAzure.MobileServices**.

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

4. Modifica la definición de la clase para que acepte un parámetro de tipo y que sea pública

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

5. Agrega a la clase **AzureService** el siguiente código para definir una variable que almacene una instancia de **IMobileServiceClient**.

```
IMobileServiceClient Client;
```

6. Agrega a la clase **AzureService** el siguiente código para definir una variable que almacene una instancia de **IMobileServiceTable** que representará a una tabla del back-end.

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

7. Agrega ahora el siguiente código para definir el constructor de la clase **AzureService**. El código inicializa el cliente del back-end y la instancia **IMobileServiceTable** para poder realizar operaciones con la tabla del back-end. Asegúrate de asignar el URL de tu **Azure Mobile Apps** en la variable **MyAppServiceURL**.

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

8. Agrega a la clase **AzureService** el siguiente código que permitirá obtener los datos de una tabla como una colección **IEnumerable**.

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

## Tarea 6. Modificar la clase Repository.

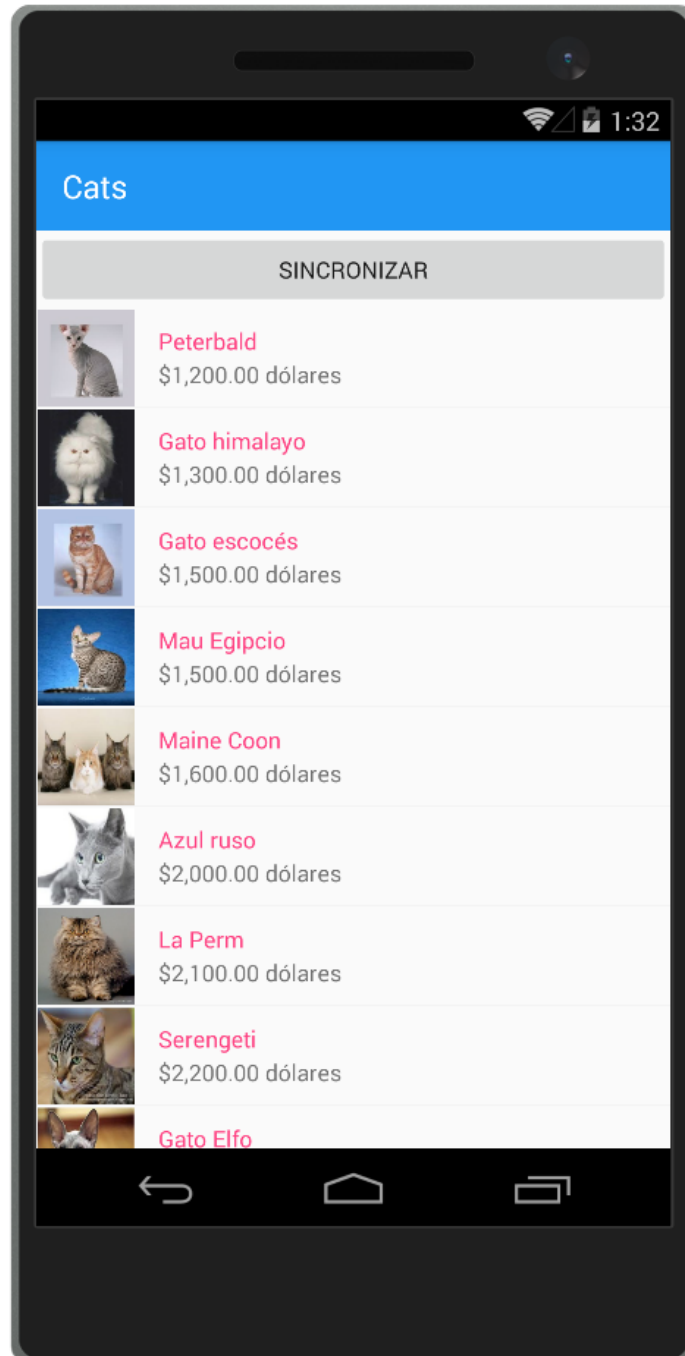
En esta tarea modificarás la clase **Repository** para que pueda obtener los datos del back-end **Azure Mobile Apps** en lugar del servicio RESTful.

1. Abre el archivo **Repository.cs**.
2. Reemplaza el código del método **GetCats** por lo siguiente.

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

3. Ejecuta la aplicación en el emulador de Android.

4. Toca el botón **Sincronizar**. Se mostrará una pantalla similar a la siguiente. Ahora los datos se están consumiendo desde el back-end **Azure Mobile Apps**.



5. Detén la aplicación y regresa a Visual Studio.
6. Prueba la aplicación en las otras plataformas.

# Resumen

---

En este laboratorio modificaste la aplicación Xamarin.Forms para consumir los datos desde una aplicación back-end hospedada en un **Azure Mobile Apps** de Microsoft Azure.

Cuando hayas finalizado este laboratorio publica el siguiente mensaje en Twitter y Facebook:

***¡He finalizado el #Lab05 del #XamarinDiplomado y conozco la forma de consumir un servicio back-end de Azure Mobile Apps desde aplicaciones Xamarin.Forms!***