

programação cross-platform com

xamarin

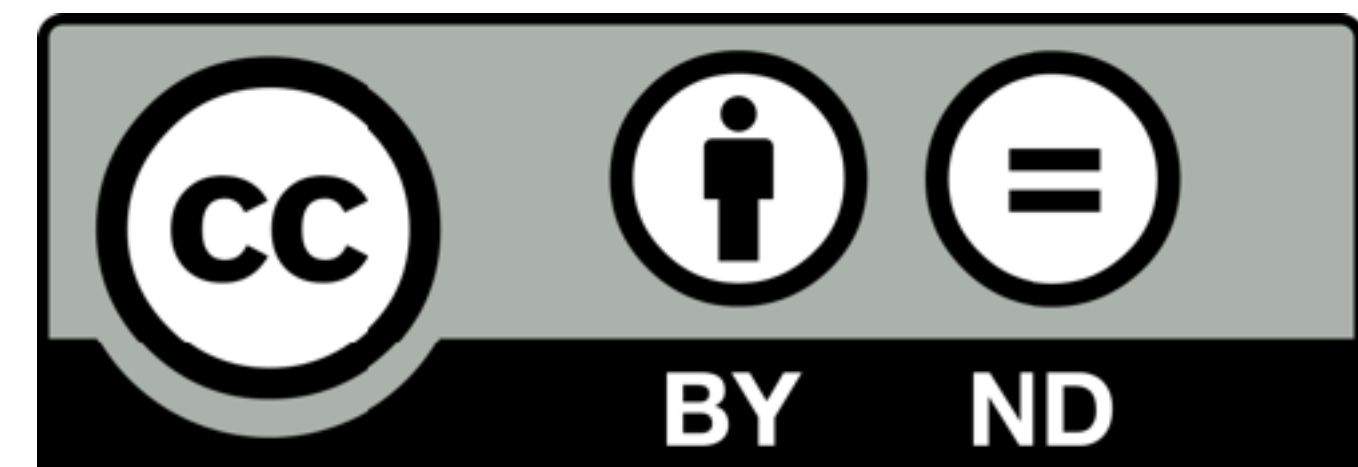


daniel ferreira
@dfsdaniel

Licença



- **Compartilhar:**
 - Você tem o direito de copiar e redistribuir o material em qualquer suporte ou formato para qualquer fim, mesmo que comercial.
- **Atribuição:**
 - Você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças foram feitas. Você deve fazê-lo em qualquer circunstância razoável, mas de maneira alguma que sugira ao licenciante a apoiar você ou o seu uso.
- **Sem derivações:**
 - Se você remixar, transformar ou criar a partir do material, você não pode distribuir o material modificado.
- **Licença CC BY-ND 4.0:**
 - https://creativecommons.org/licenses/by-nd/4.0/deed.pt_BR



Quem sou eu?



@dfsdaniel

- **DANIEL FERREIRA DA SILVA**
- **Graduado em Ciência da Computação (UFPE)**
- **Mestrando em Engenharia de Requisitos (UPE)**
- **MCP, MCTS Web, MCPD, MCTS WPF**
- **Microsoft Student Partner (2016-2010)**
- **Engenheiro de Sistemas CESAR**
- **Instrutor Windows Phone/C#/Xamarin:**
 - Quality
 - CESAR School
 - iDez (João Pessoa)
 - FA7 (Fortaleza)
 - Fatepi (Piauí)
 - Unibratec
 - FG

Bibliografia

- Creating Mobile Apps with Xamarin.Forms
 - Charles Petzold
 - Microsoft Press
 - 2016





Introdução

Programação mobile



- Desktops vão ficar apenas para o **meio profissional** específico.
- **Mobilidade** hoje é fundamental em todos os dispositivos (IoT)
- **Maior quantidade:** Smartphones com Android
- **Maior tempo gasto:** iPhones
- Corre por fora: Windows Phone e Windows 10 Mobile (??)

Desafios



- **Diferentes paradigmas de interface**
 - Cada sistema possui peculiaridades de como interagir e fazer determinadas ações (locais de botões, menus, etc...)
- **Diferentes ambientes de desenvolvimento**
 - Para iPhones: Xcode com Mac
 - Para Androids: Android Studio em diferentes plataformas
 - Para Windows: Visual Studio com PC

Desafios



- **Diferentes interfaces de programação**
 - Um simples **toggle button**:
 - iPhone: `UISwitch`
 - Android: `Switch`
 - Windows: `ToggleSwitch`
- **Diferentes linguagens de programação**
 - Para iPhones: Objective-C/Swift
 - Para Androids: Java
 - Para Windows: C#

C# e .NET



- Relevada pela Microsoft em 2000, C# é uma linguagem de programação **relativamente nova** (comparada ao Objective-C e Java).
- C# é **fortemente tipada** e **orientada a objetos**. Foi influenciada pelo Java e pelo C++ (porém bem mais limpa e direta)
- Desde sua primeira versão, C# dá suporte a **Propriedades** e **Eventos**, o que ajuda bastante na programação de interfaces de usuário.
- Totalmente ligada ao **.NET Framework**:
 - Math, Debugging, Reflection, Collections, Globalization, File I/O, Networking, Security, Threading, Web services, Data handling, XML e JSON, etc...

E o Xamarin?



- Assim que o .NET foi anunciado em Junho/2000, a companhia **Ximian** resolveu criar o **Projeto Mono (open-source)**, que era um compilador de C# e um .NET Framework que eram compatíveis com **plataformas Linux**.
- Mais tarde a Ximian foi adquirida pela Novell.
- **Em 2011**, os fundadores da Ximian fundaram **a Xamarin**, a qual continuou contribuindo com o Mono porém inserindo a possibilidade de programação **cross-platform mobile**.

E o Xamarin?



- Em 2014 o .NET com o C# se mostra bem promissor para o futuro e diversas empresas começam a investir numa versão do **.NET Compiler de código aberto**. A **.NET Foundation** é criada com o objetivo de ajudar nessa tarefa.
- Um dos principais apoiadores da .NET Foundation foi **a Xamarin**.
- Em **março de 2016**, a Microsoft adquire a Xamarin com o objetivo de trazer todo o desenvolvimento cross-plataform com .NET para o Visual Studio.

E o Xamarin?



- O coração do Xamarin é composto basicamente de 3 grandes bibliotecas .NET:
 - **Xamarin.Mac** (que evoluiu a partir do MonoMac Project)
 - **Xamarin.iOS** (que evoluiu do MonoTouch)
 - **Xamarin.Android** (que evoluiu do Mono for Android - MonoDroid)
- Todas as bibliotecas são versões .NET dos códigos nativos de **Mac, iOS e Android APIs**.
- Programadores utilizando essas bibliotecas podem desenvolver para essas plataformas utilizando C#, com a vantagem de ter acesso a todo o .NET Framework e seus benefícios.

Requisitos



- **Com Mac**

- Xcode
- Xamarin Studio / Visual Studio for Mac
 - Desenvolvimento para iOS
 - Desenvolvimento para Android

- **PC com Windows**

- Visual Studio 2015+
 - Desenvolvimento para Windows
 - Desenvolvimento para Android
 - Desenvolvimento para iOS (com Mac + Xcode instalado na mesma rede)



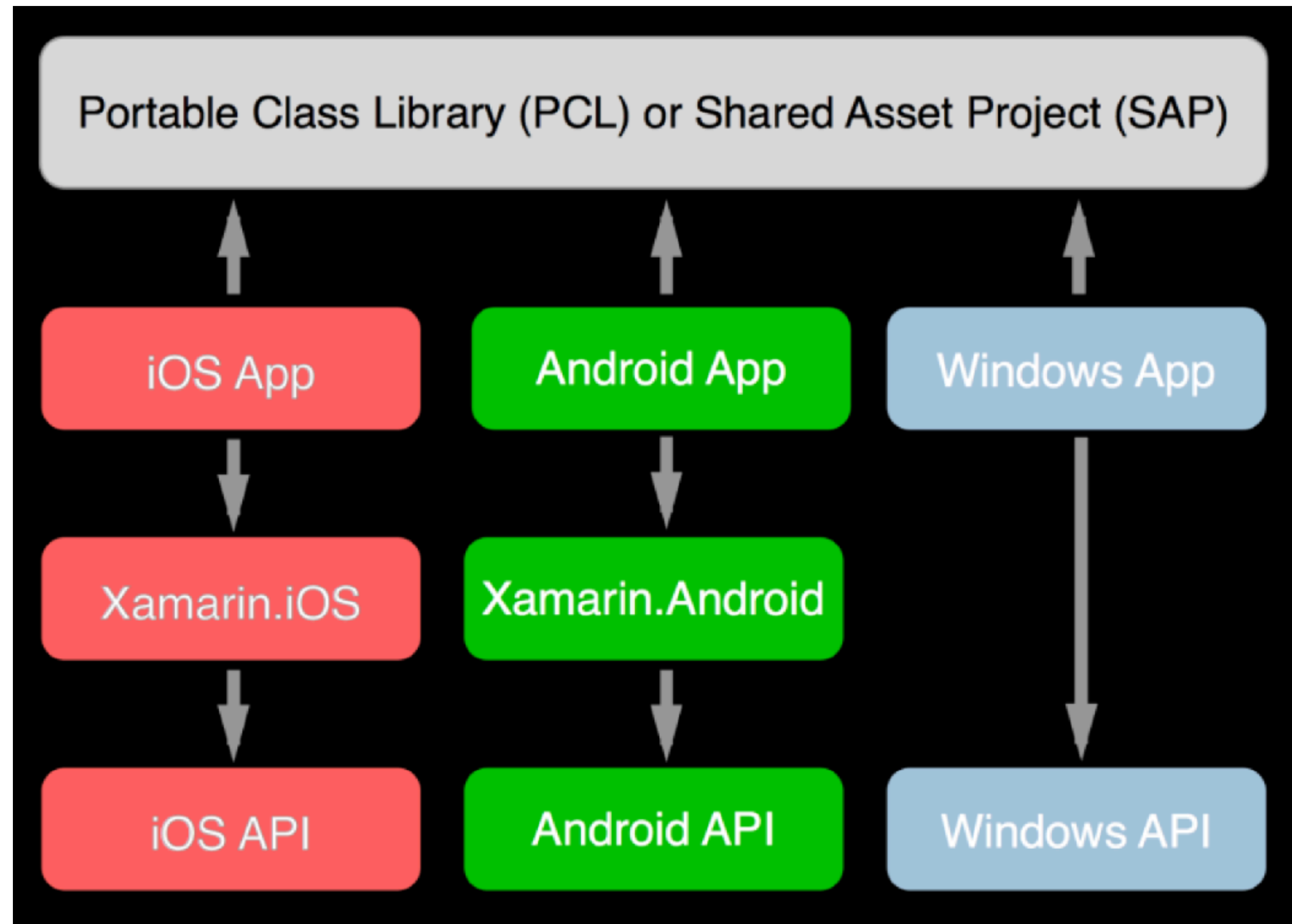
Arquitetura

Shared Code



- Arquitetura baseada em MVVM (Model-View-ViewModel).
- Todo o código **independente de plataforma** pode ser isolado.
 - **Shared Asset Project (SAP)**
 - Distribui o código para ser acessível em outros projetos.
 - **Portable Class Library (PCL)**
 - Encapsula todo o código compartilhado em DLLs.

Arquitetura

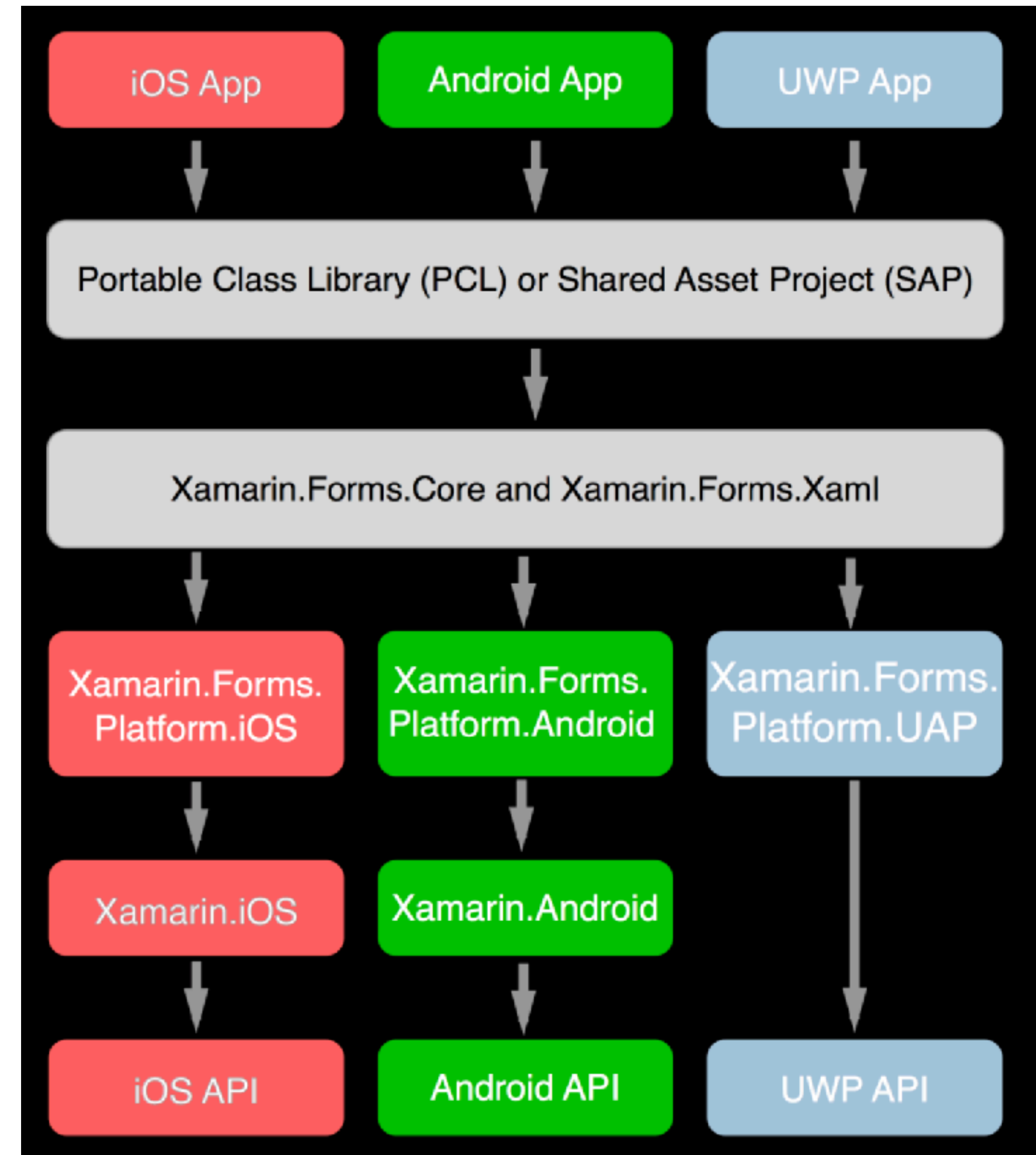
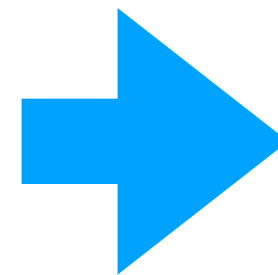
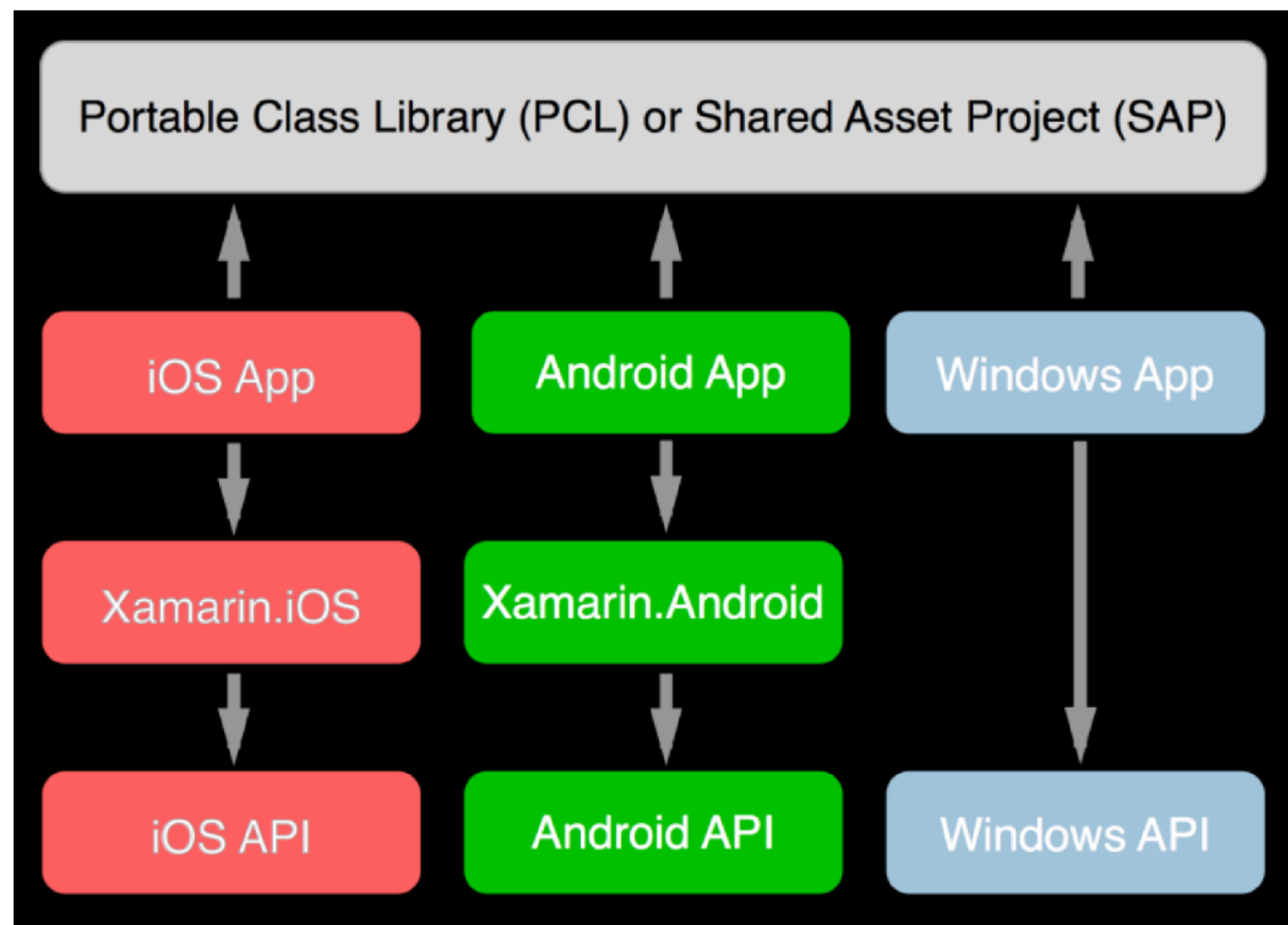


Xamarin Forms

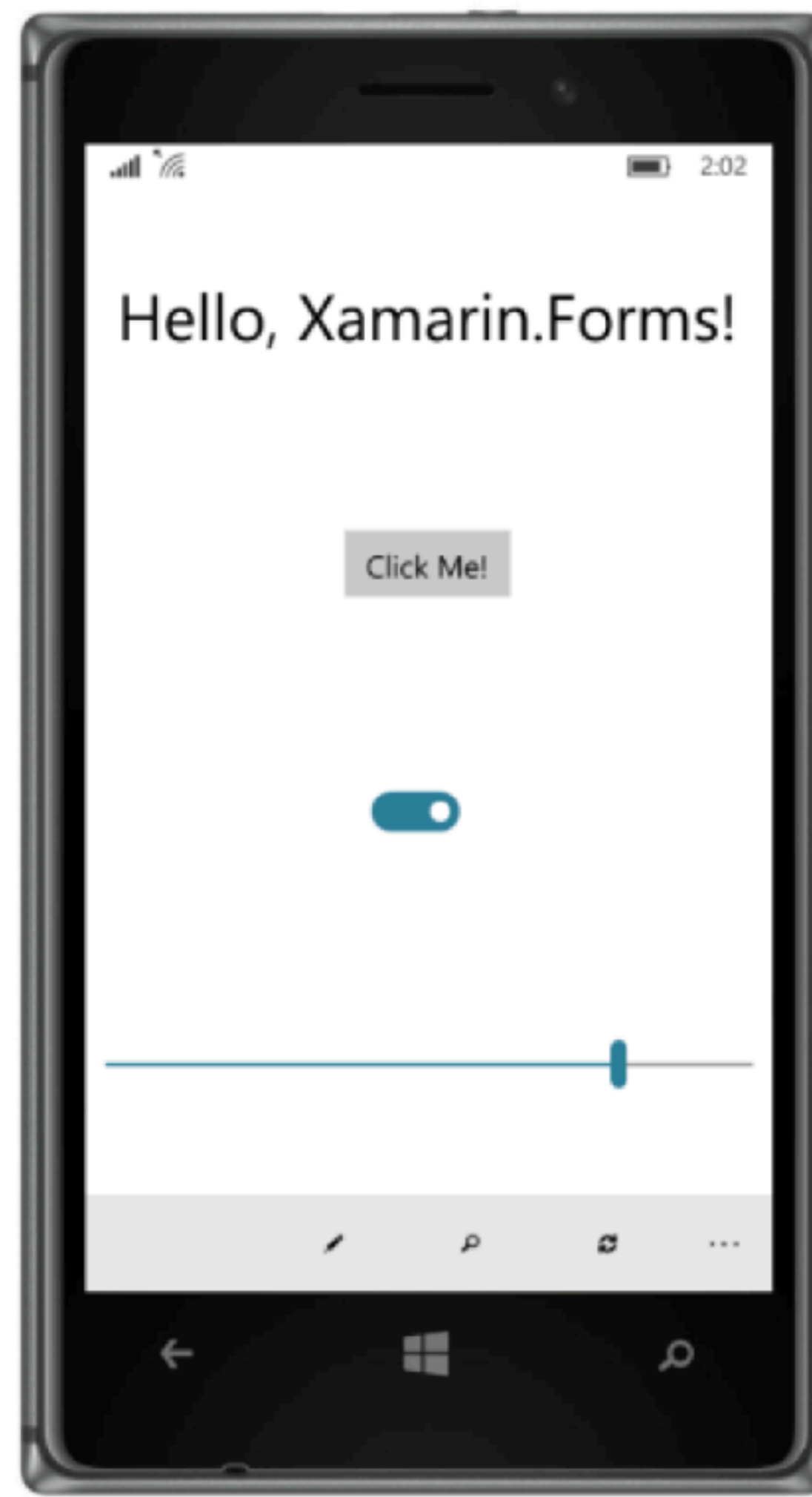
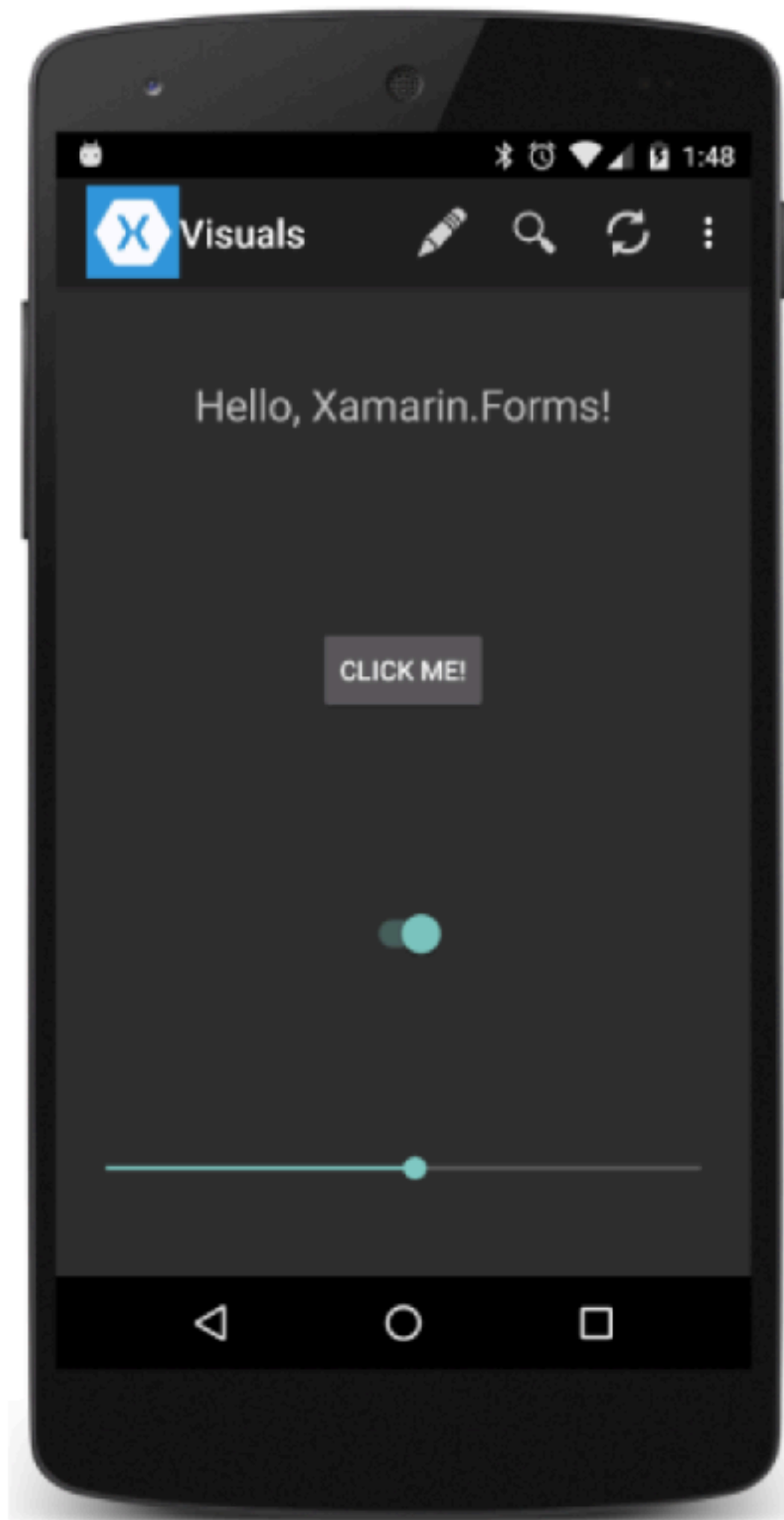
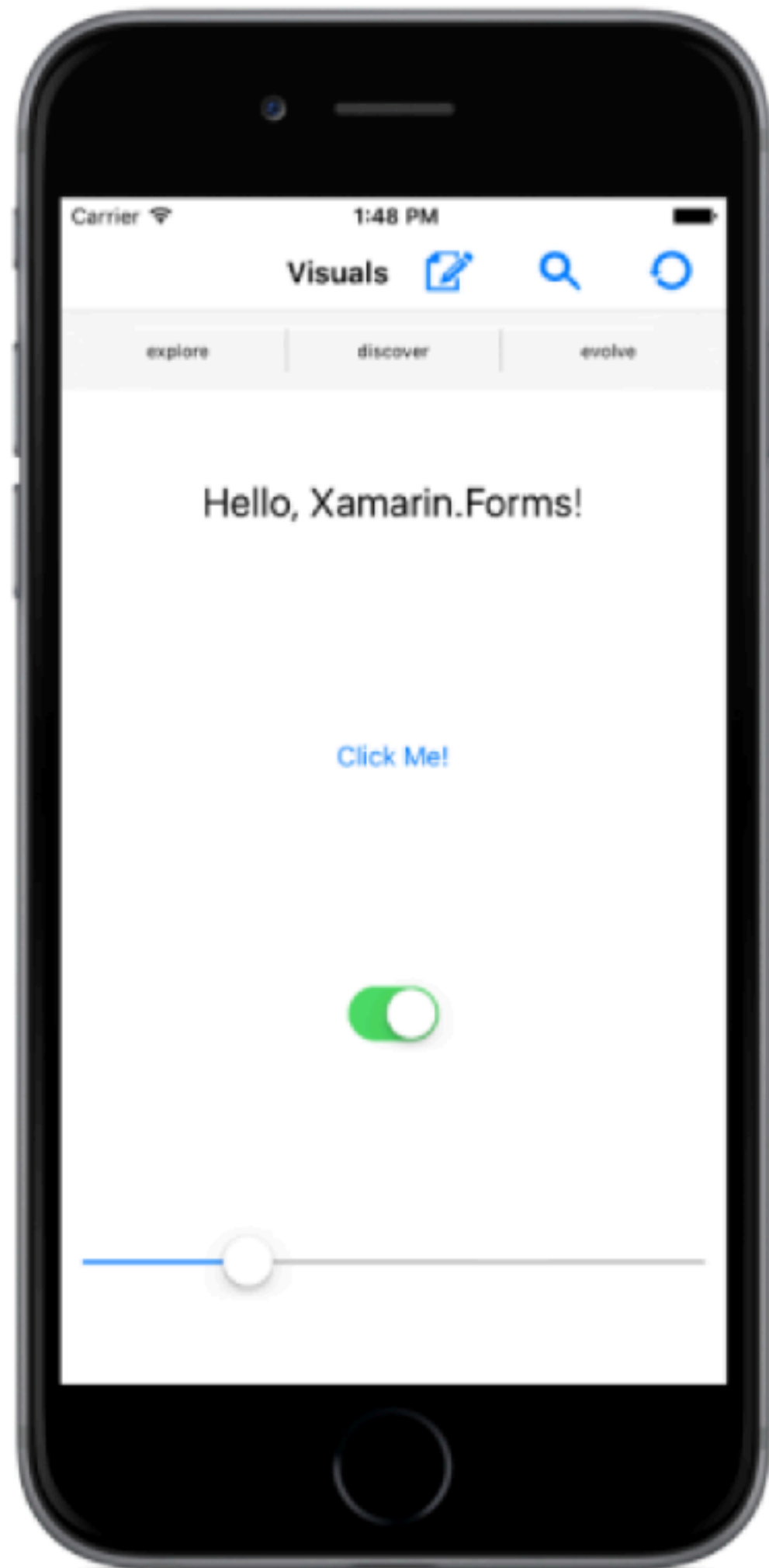


- Criado em maio de 2014, Xamarin Forms permite a criação de interfaces de usuários que podem ser compiladas para iOS, Android e Windows.
- **Xamarin.Forms.Core e Xamarin.Forms.Xaml**
- Suporta:
 - iOS (iPhone, iPad, iPod Touch)
 - Android (smartphones e tablets)
 - Universal Windows Platform - UWP (Windows 10 e Windows 10 Mobile)
 - Windows Runtime API (Windows 8.1 e Windows Phone 8.1)

Xamarin Forms



Xamarin Forms



- Button
- Switch
- Slider
- **ToolBarItem**
 - UIBarButtonItem
 - ActionBar
 - CommandBar

Suporte a XAML



- XAML: **Extensible Application Markup Language**
 - Pronuncia: "zammel"
 - Linguagem para desenvolvimento de interfaces
 - Base do WPF (Windows Presentation Foundation)

```
<StackLayout Padding="10,0">
  <Label Text="Hello, Xamarin.Forms!"
    FontSize="Large"
    VerticalOptions="CenterAndExpand"
    HorizontalOptions="Center" />

  <Button Text="Click Me!"
    VerticalOptions="CenterAndExpand"
    HorizontalOptions="Center" />

  <Switch VerticalOptions="CenterAndExpand"
    HorizontalOptions="Center" />
</StackLayout>
```

Suporte a XAML



- Para ser específico em plataforma, podemos utilizar:
 - **OnPlatform***
 - **Switch + RuntimePlatform**

```
<ContentPage.ToolbarItems>
  <ToolbarItem Text="edit" Order="Primary">
    <ToolbarItem.Icon>
      <OnPlatform x:TypeArguments="FileImageSource"
        iOS="edit.png"
        Android="ic_action_edit.png"
        WinPhone="Images/edit.png" />
    </ToolbarItem.Icon>
  </ToolbarItem>
</ContentPage.ToolbarItems>
```

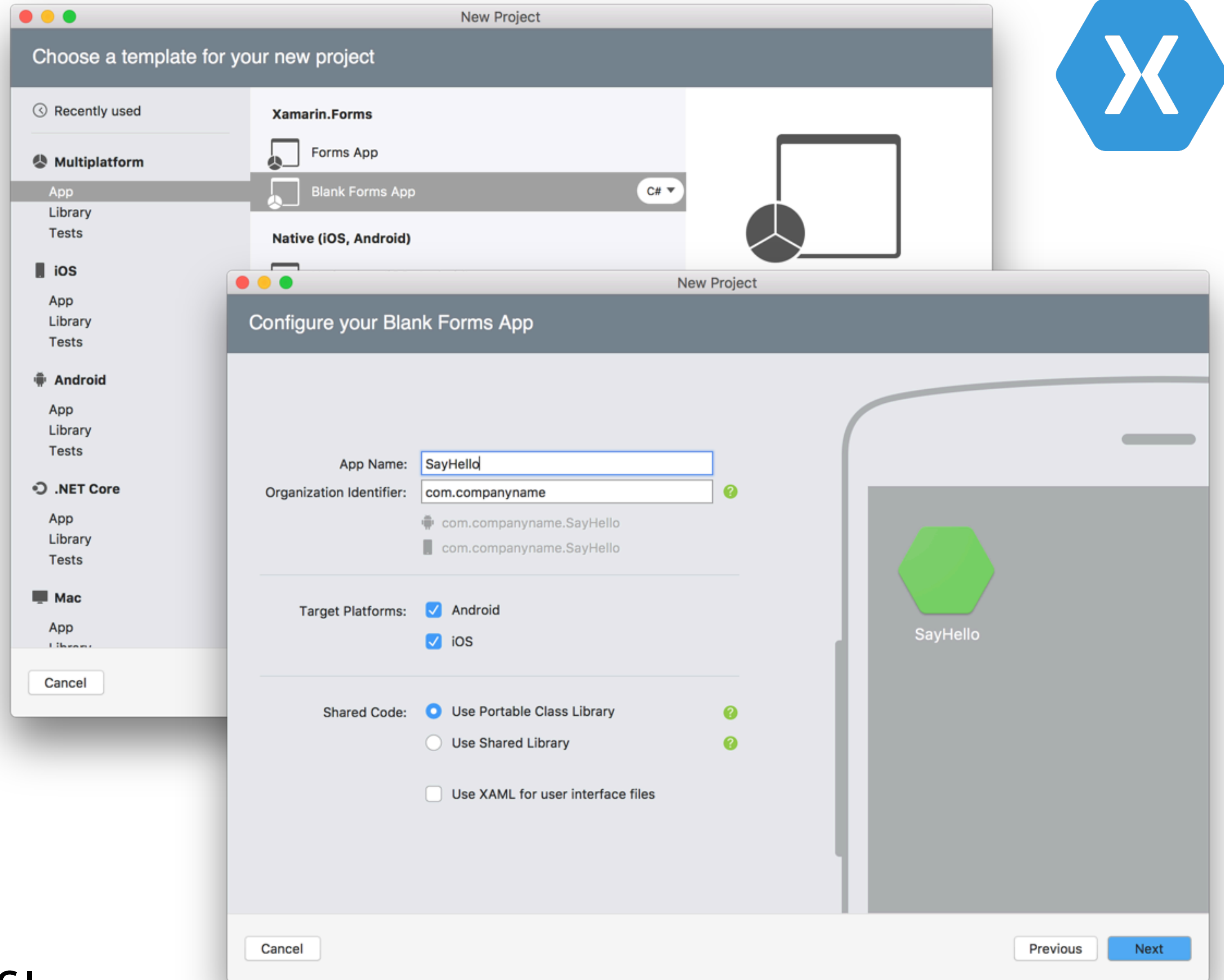
A large blue arrow pointing from the bottom left towards the `<OnPlatform>` element in the XAML code.



Say Hello

Say Hello

- File > New Solution
- Multiplatform > Blank Forms App > C#
- App Name
Organization Identifier
- Target Platforms
- **Shared Code**
 - Use Portable Class Library
 - Use Shared Library
- Use XAML for user interface files



Say Hello



- Project Name
- Solution Name
- Location
 - Create a project directory within the solution directory
- Create!

The screenshot shows the 'New Project' dialog box in Visual Studio. The title bar says 'New Project'. The main heading is 'Configure your new Blank Forms App'. The dialog is divided into two main sections: configuration on the left and a preview on the right.

Configuration Section:

- Project Name:** A text box containing 'SayHello'.
- Solution Name:** A text box containing 'SayHello'.
- Location:** A text box containing '/Users/danielferreira/XamarinProjects' with a 'Browse...' button to its right.
- Checkboxes:**
 - ☒ Create a project directory within the solution directory.
 - ☐ Use git for version control.
 - ☒ Create a .gitignore file to ignore inessential files.
 - ☐ Add an automated UI test project. [Learn More](#)

Preview Section:

PREVIEW

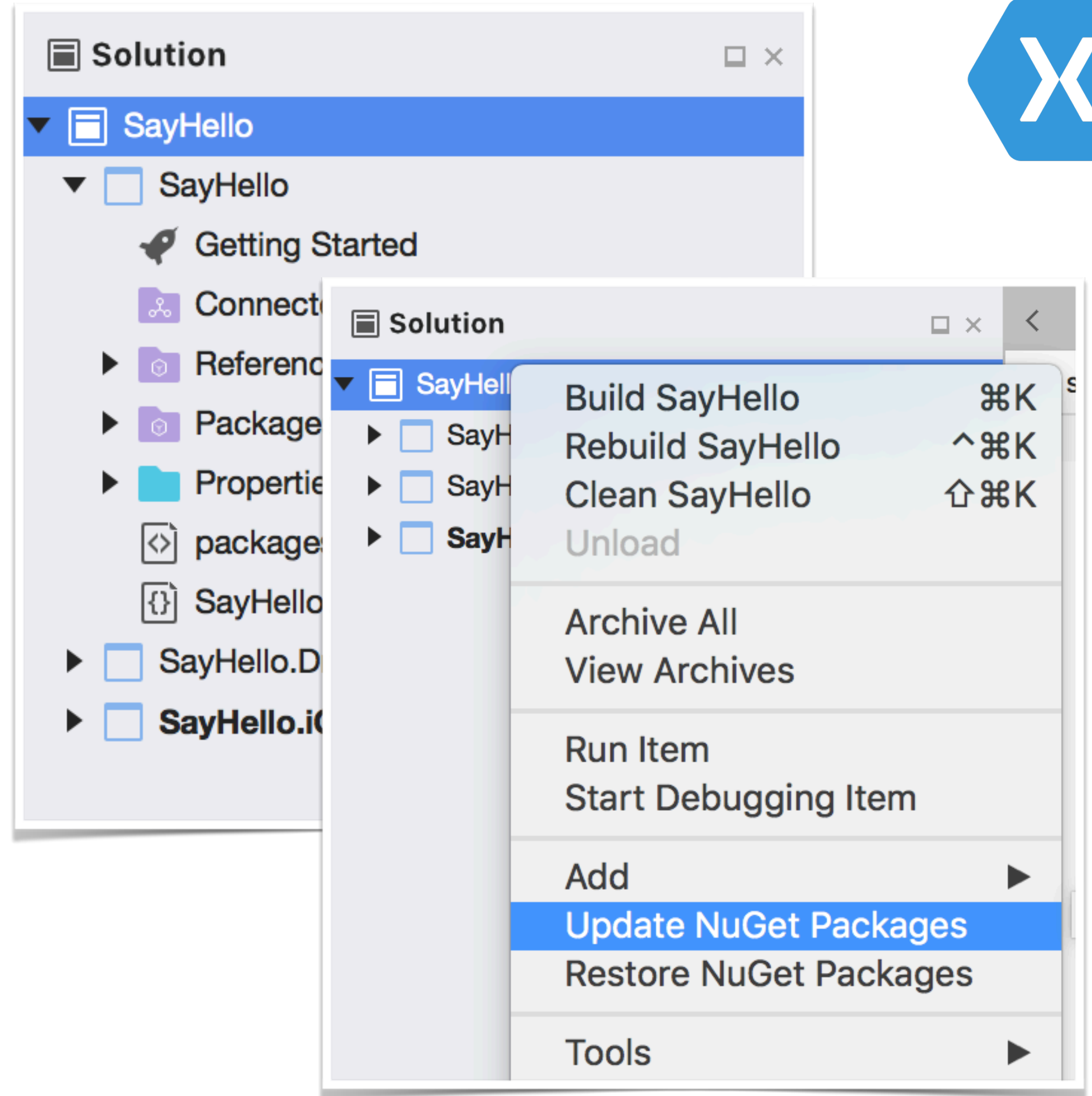
- Folder: /Users/danielferreira/XamarinProjects
 - Folder: SayHello
 - File: SayHello.sln
 - Folder: SayHello
 - File: SayHello.csproj

Buttons:

- Cancel** (bottom left)
- Previous** (bottom right, disabled)
- Create** (bottom right, highlighted in blue)

Say Hello

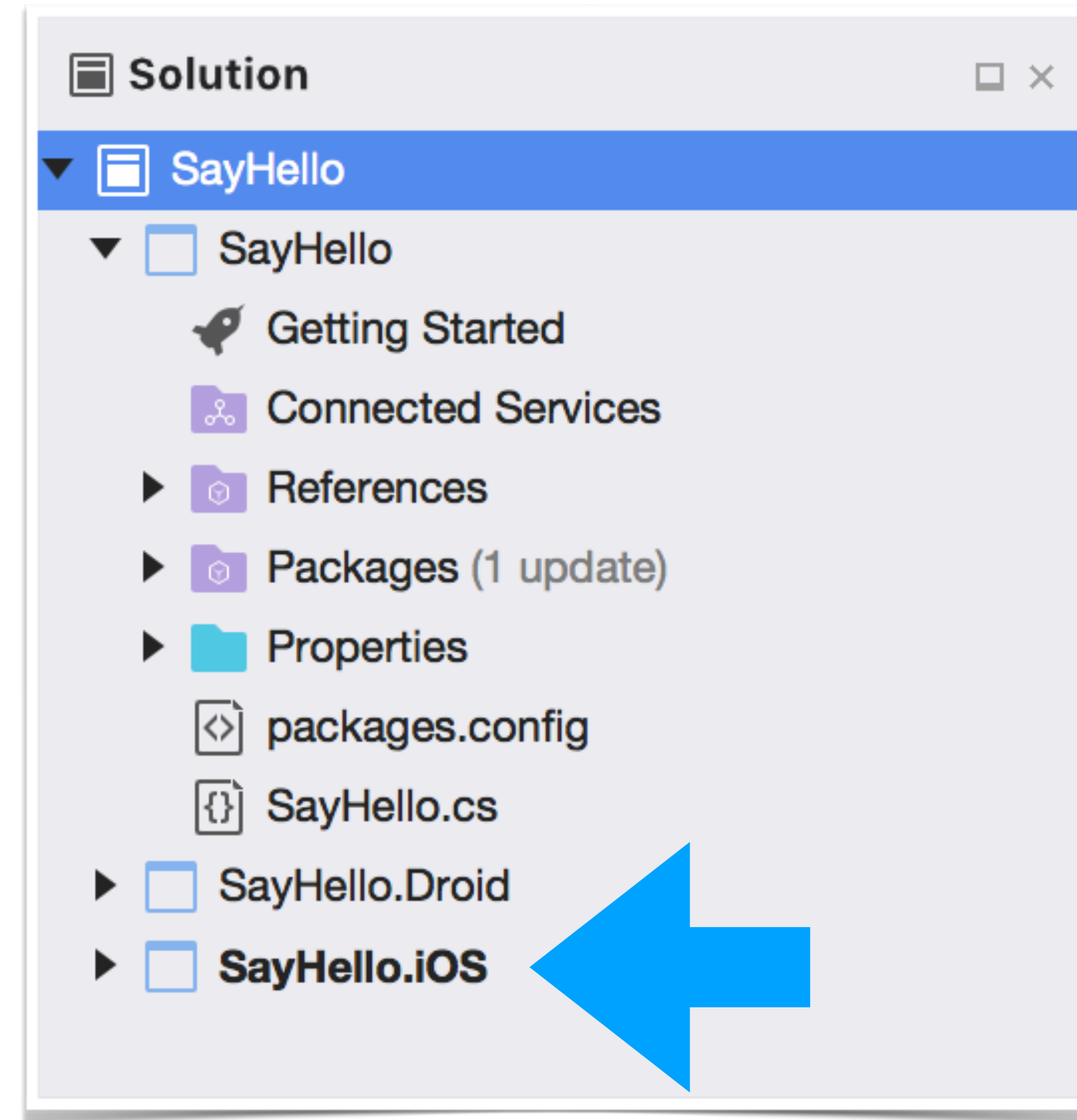
- Projetos:
 - SayHello (PCL)
 - SayHello.Droid
 - SayHello.iOS
- Sempre atualize seus projetos!
- Solução > botão direito >
Update NuGet Packages



Say Hello



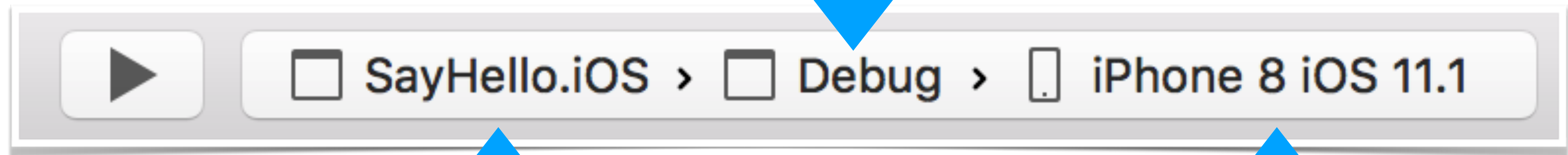
- Startup Project
 - Projeto que será executado ao clicar em RUN
- Fica em negrito
- Botão direito > **Set as Startup Project**



Say Hello



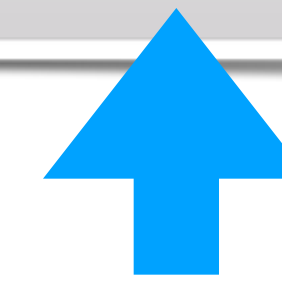
Build configuration



Startup Project



Emulador



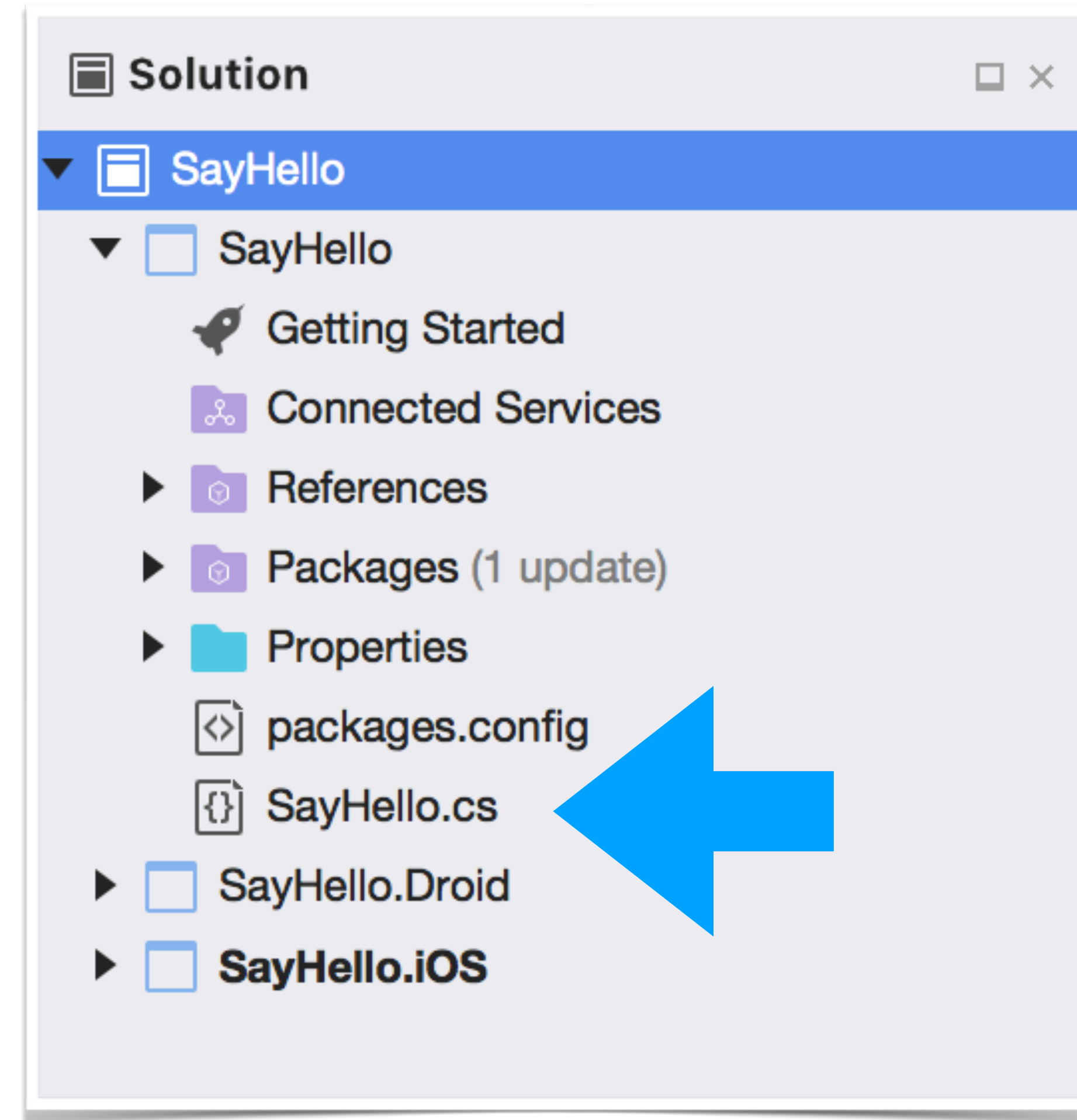
Say Hello



App.cs



- Objetivo principal é inicializar a página principal da aplicação.
- Classe do tipo `Application`.
- Pode estar associada a um arquivo `.xaml`.



App.cs



```
public class App : Application
{
    public App()
    {
        // The root page of your application
        var content = new ContentPage
        {
            Title = "SayHello",
            Content = new StackLayout
            {
                VerticalOptions = LayoutOptions.Center,
                Children = {
                    new Label {
                        HorizontalTextAlignment = TextAlignment.Center,
                        Text = "Welcome to Xamarin Forms!"
                    }
                }
            };
            MainPage = new NavigationPage(content);
        }
    }
}
```

App.cs



iOS

```
[Register("AppDelegate")]
public partial class AppDelegate : global::Xamarin.Forms.Platform.iOS.FormsApplicationDelegate
{
    public override bool FinishedLaunching(UIApplication app, NSDictionary options)
    {
        global::Xamarin.Forms.Forms.Init();
    }
}
```

Android

```
public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
{
    protected override void OnCreate(Bundle bundle)
    {
        TabLayoutResource = Resource.Layout.Tabbar;
        ToolbarResource = Resource.Layout.Toolbar;

        base.OnCreate(bundle);

        global::Xamarin.Forms.Forms.Init(this, bundle);
        LoadApplication(new App());
    }
}
```

PCL ou SAP?



- **PCL: Portable Class Library**

- Encapsula todo o código compartilhado em DLLs.
- Não permite uso de diretivas
 - Uso do **RuntimePlatform** para chamadas específicas.

- **SAP: Shared Asset Project**

- Distribui o código para ser acessível em outros projetos.
- Uso de diretivas pre-compiladas

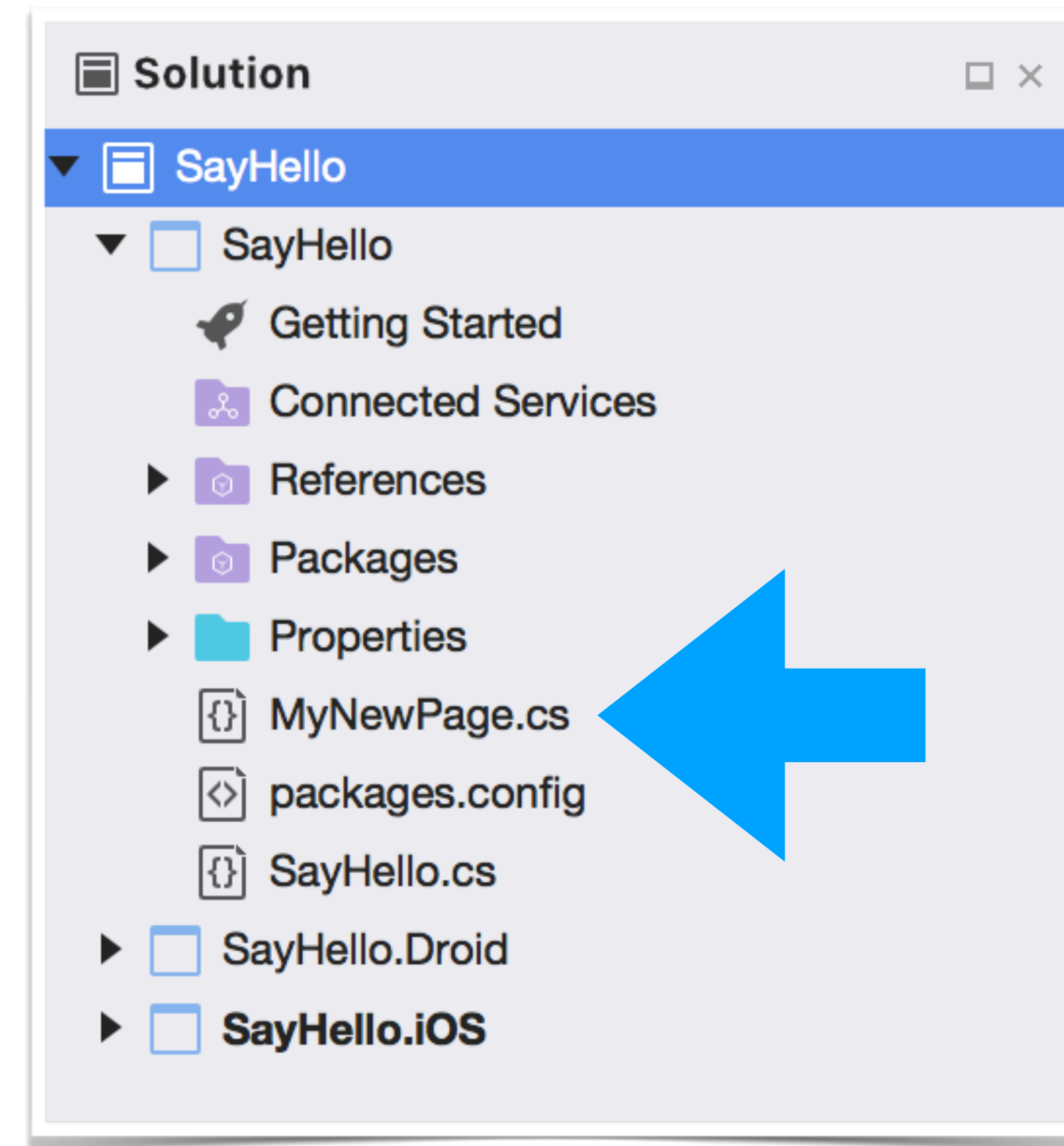
Shared Asset Project

```
#if __IOS__
    // iOS specific code
#elif __ANDROID__
    // Android specific code
#elif WINDOWS_UWP
#elif WINDOWS_APP
    // Universal Windows Platform specific code
    // Windows 8.1 specific code
#elif WINDOWS_PHONE_APP
    // Windows Phone 8.1 specific code
#endif
```

Exercício



- Adicione uma nova página no projeto **SayHello** e faça com que ela seja a página principal da sua aplicação.
- Esta nova página também deve conter um label com o texto **“ESTA É MINHA NOVA PÁGINA”**.
- No Android, o texto da nova página deve ser centralizado, e no iPhone deve estar no topo centralizado.



Device.OnPlatform



- `Device.OS` retorna um `enum` do tipo `TargetPlatform`:
 - `TargetPlatform.iOS`
 - `TargetPlatform.Android`
 - `TargetPlatform.WinPhone`
 - `TargetPlatform.Other`
- `Device.Idiom` retorna o tipo de dispositivo num `enum` do tipo `TargetIdiom`:
 - `TargetIdiom.Phone`
 - `TargetIdiom.Tablet`
 - `TargetIdiom.Desktop`
 - `TargetIdiom.Unsupported`

Device.OnPlatform



- Forma mais elegante é utilizar um método estático `Device.OnPlatform`.

```
Padding = Device.OnPlatform<Thickness>(new Thickness(0, 20, 0, 0),  
                                         new Thickness(0),  
                                         new Thickness(0));
```

Windows → iOS → Android

```
Padding = Device.OnPlatform(new Thickness(0, 20, 0, 0),  
                             new Thickness(0),  
                             new Thickness(0));
```

```
Padding = new Thickness(0, Device.OnPlatform(20, 0, 0), 0, 0);
```

```
<ContentPage.ToolbarItems>  
  <ToolbarItem Text="edit" Order="Primary">  
    <ToolbarItem.Icon>  
      <OnPlatform x:TypeArguments="FileImageSource"  
                  iOS="edit.png"  
                  Android="ic_action_edit.png"  
                  WinPhone="Images/edit.png" />  
    </ToolbarItem.Icon>  
  </ToolbarItem>  
</ContentPage.ToolbarItems>
```

Device.OnPlatform




- Infelizmente o `Device.OnPlatform` está **deprecated**. Para obter o mesmo resultado, recomenda-se agora utilizar o `enum Device.RuntimePlatform`.
- **OnPlatform** está disponível apenas para uso no XAML.

```
int paddingTop = 0;

switch (Device.RuntimePlatform)
{
    case Device.iOS:
        paddingTop = 10;
        break;
    case Device.Android:
        paddingTop = 0;
        break;
    case Device.WinPhone:
        paddingTop = 0;
        break;
}

Padding = new Thickness(paddingTop, 0, 0, 0);
```

A blue arrow pointing from the right towards the `Device.RuntimePlatform` property access in the switch statement.