**Práctica 09 – Gorilla Player y Facebook**

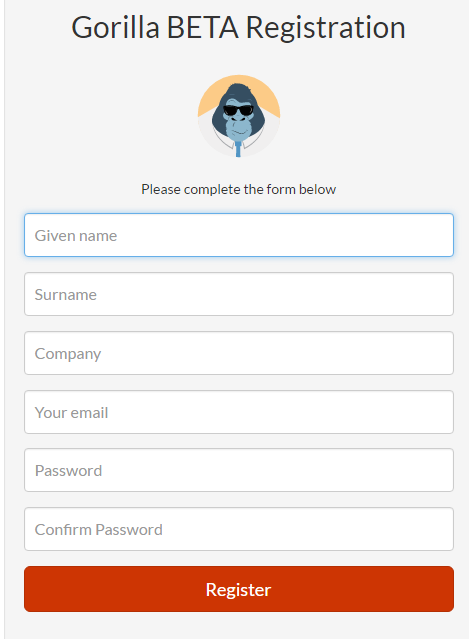
El objetivo de la primer parte de esta práctica es instalar la herramienta **Gorilla Player**, la cual nos permite obtener una vista en tiempo real del diseño **XAML** de nuestras aplicaciones de **Xamarin.Forms**, que es una de las limitantes de Xamarin pero en la cual ya se está trabajando (XAML Previewer for Xamarin.Forms, más información en <https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-previewer/>).

La segunda parte de la práctica introduce la conexión a Facebook desde nuestros proyectos de Xamarin.Forms a través de un componente. Dada la importancia de las redes sociales hoy en día, su integración en una aplicación móvil resulta fundamental para incrementar su funcionalidad, motivar a nuestros usuarios a que utilicen la app, etc.

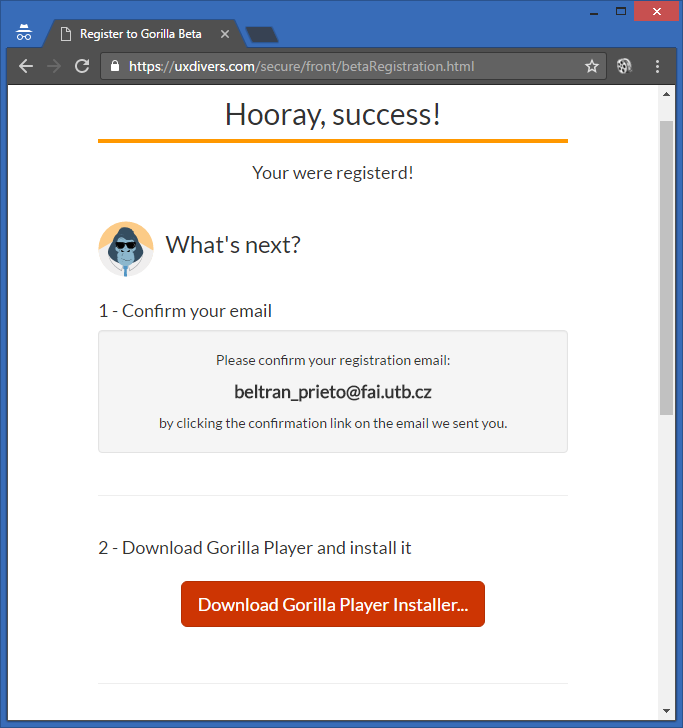
**Parte 1. Gorilla Preview**

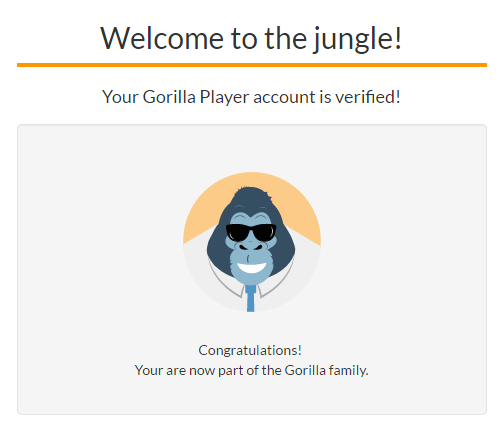
**Paso 1:** Ingresa al portal de **Gorilla Player** (<http://gorillaplayer.com/>) y da clic en el enlace **Register Now For Free** para acceder a la versión Beta de este visualizador de diseño.



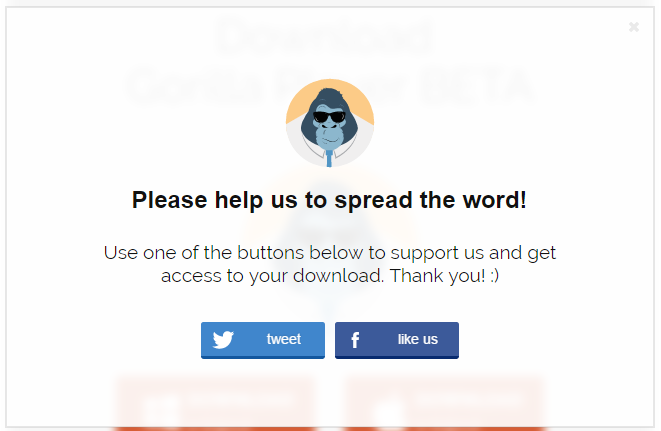


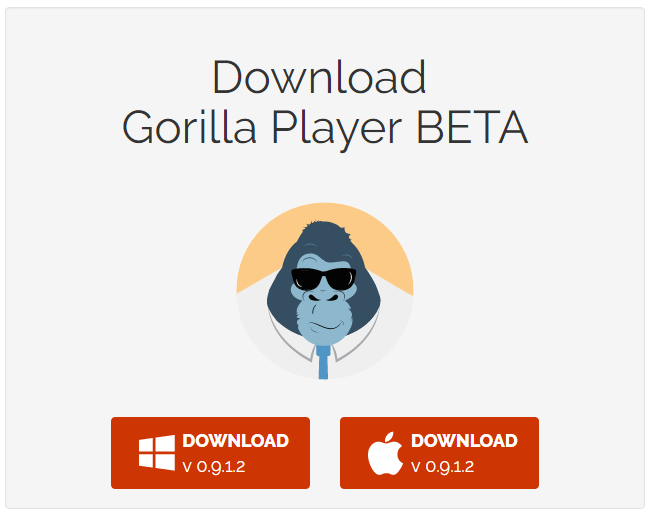
**Paso 2:** Confirma tu correo dando clic en el link de confirmación que recibirás en tu bandeja de entrada.





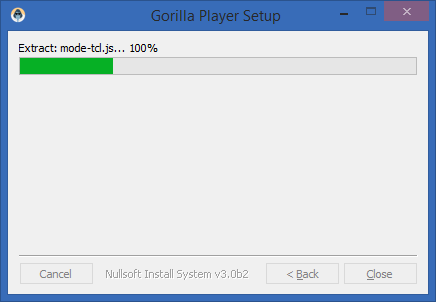
**Paso 3.** En lo que llega el correo, puedes dar clic en el botón para comenzar la descarga del instalador de Gorilla Player. Es posible que te soliciten un Tweet o Like para acceder a la descarga. Puedes cerrar la solicitud con el botón que aparece en la esquina superior derecha.

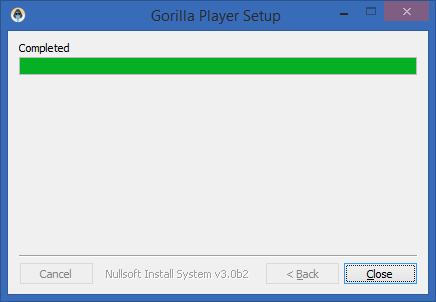




**Paso 4.** Procede a instalar la aplicación.







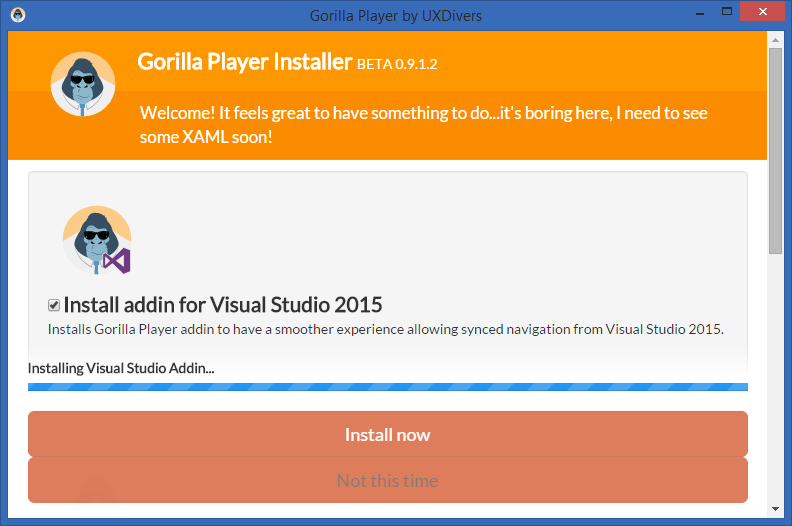
**Paso 5.** Una vez instalado el producto, acepta los términos de uso.

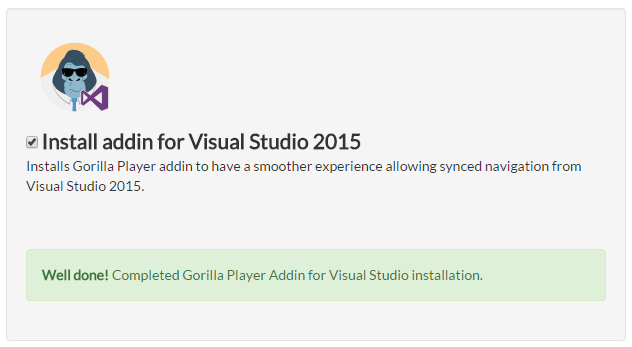


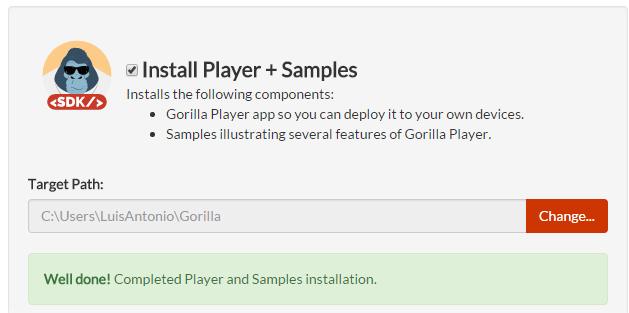
**Paso 6.** Existen 3complementos que puedes instalar, si así lo deseas:

* Un **addin** para **Visual Studio** **2015**
* Un **addin** para **Xamarin Studio**.
* **SDK, Player** y **Ejemplos**.

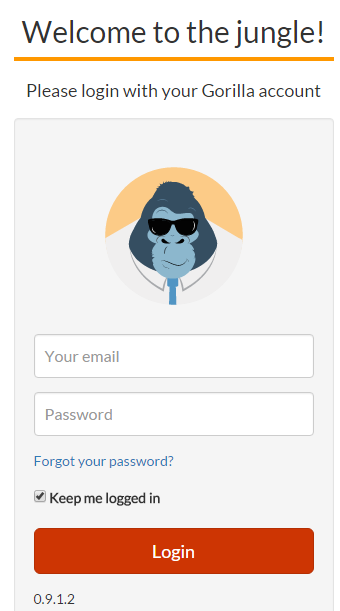
Se recomienda instalar los 3 (o al menos uno de los addins y el SDK).



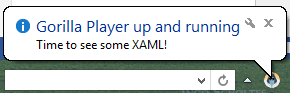


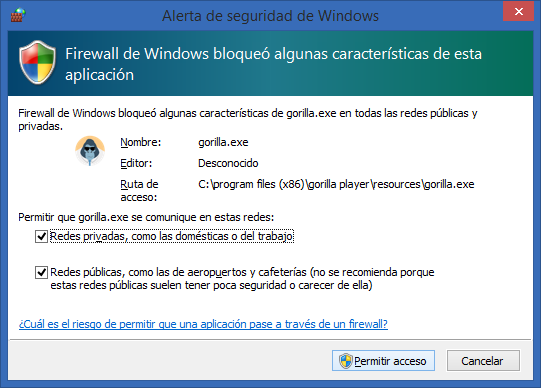


**Paso 7.** **Inicia sesión** con tu cuenta de Gorilla Player para iniciar el servicio.

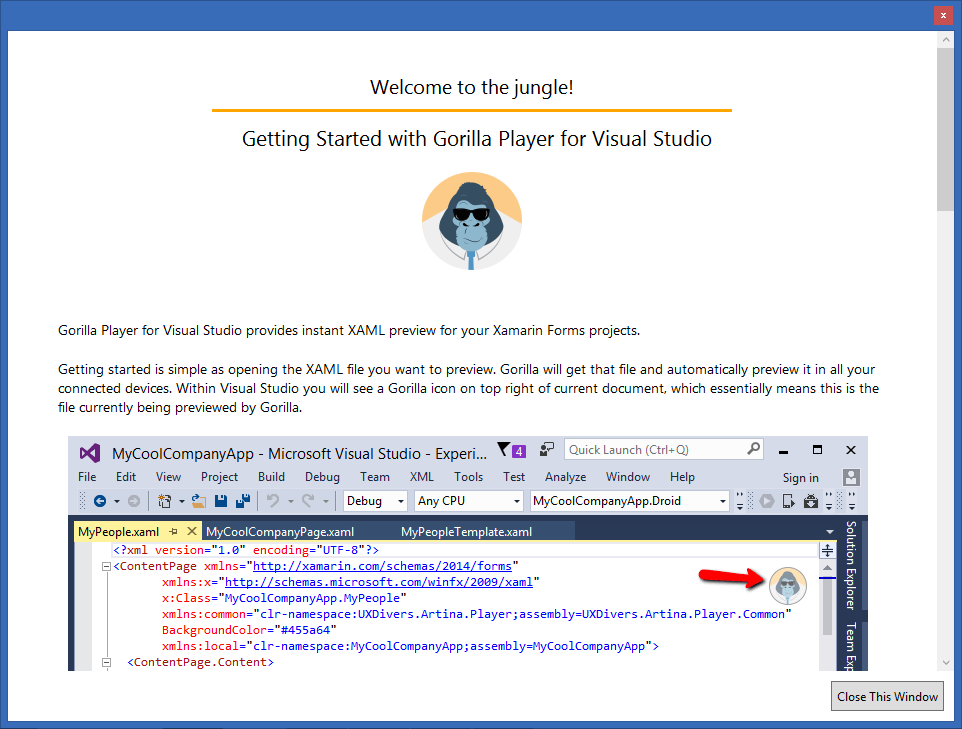


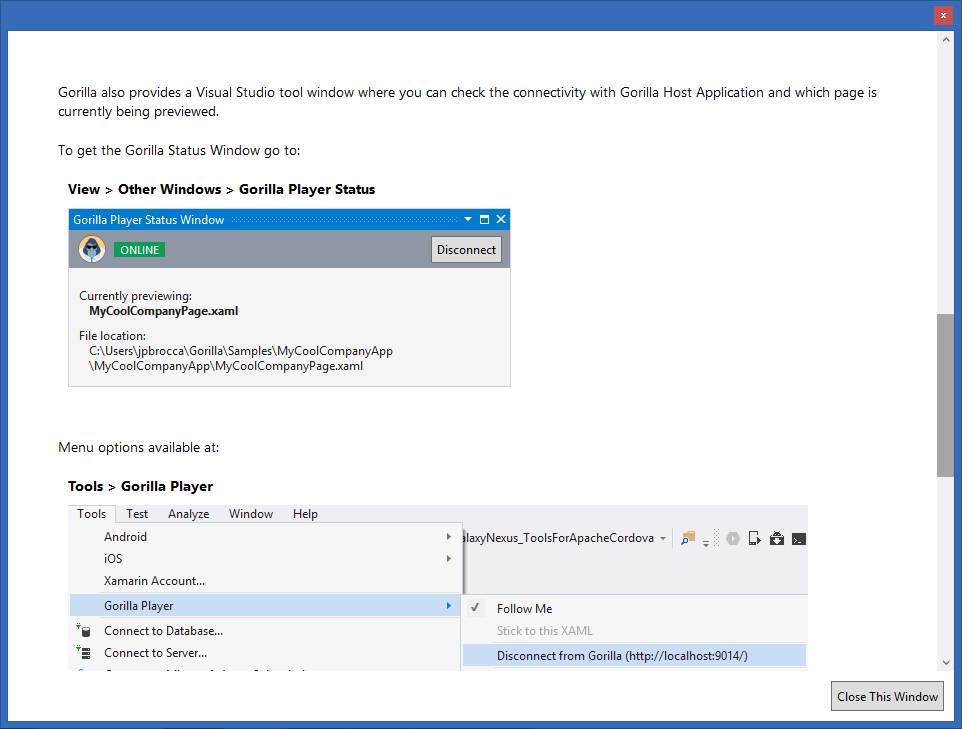
**Paso 8.** Recibirás una notificación de que el servicio está en ejecución. Es probable que Gorilla Player solicite permiso de ejecución al firewall. Autorízalo.

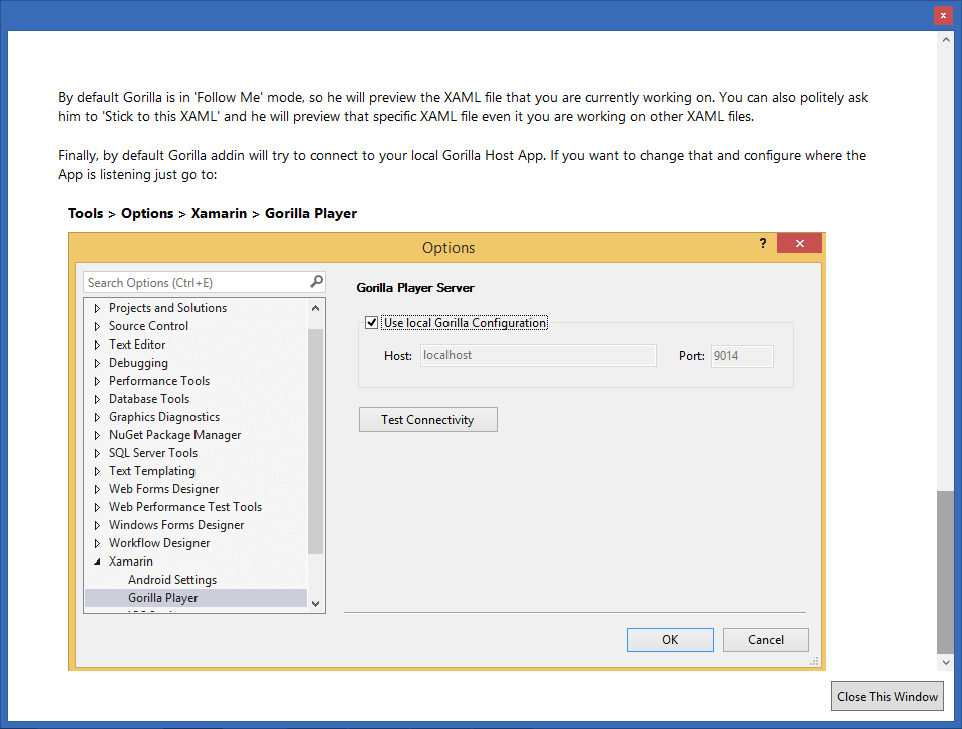




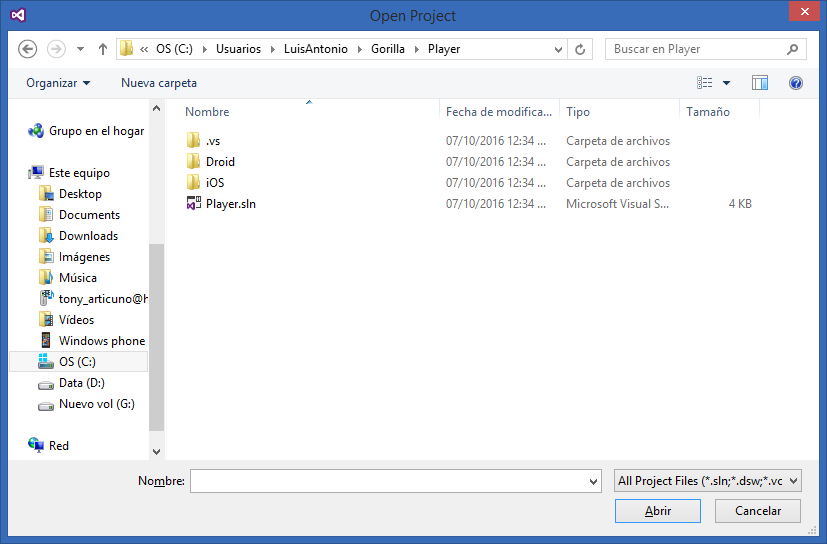
**Paso 9.** Abre Visual Studio 2015 (o Xamarin Studio). Es probable que aparezca una ventana con la ayuda inicial de este diseñador. Revísala.



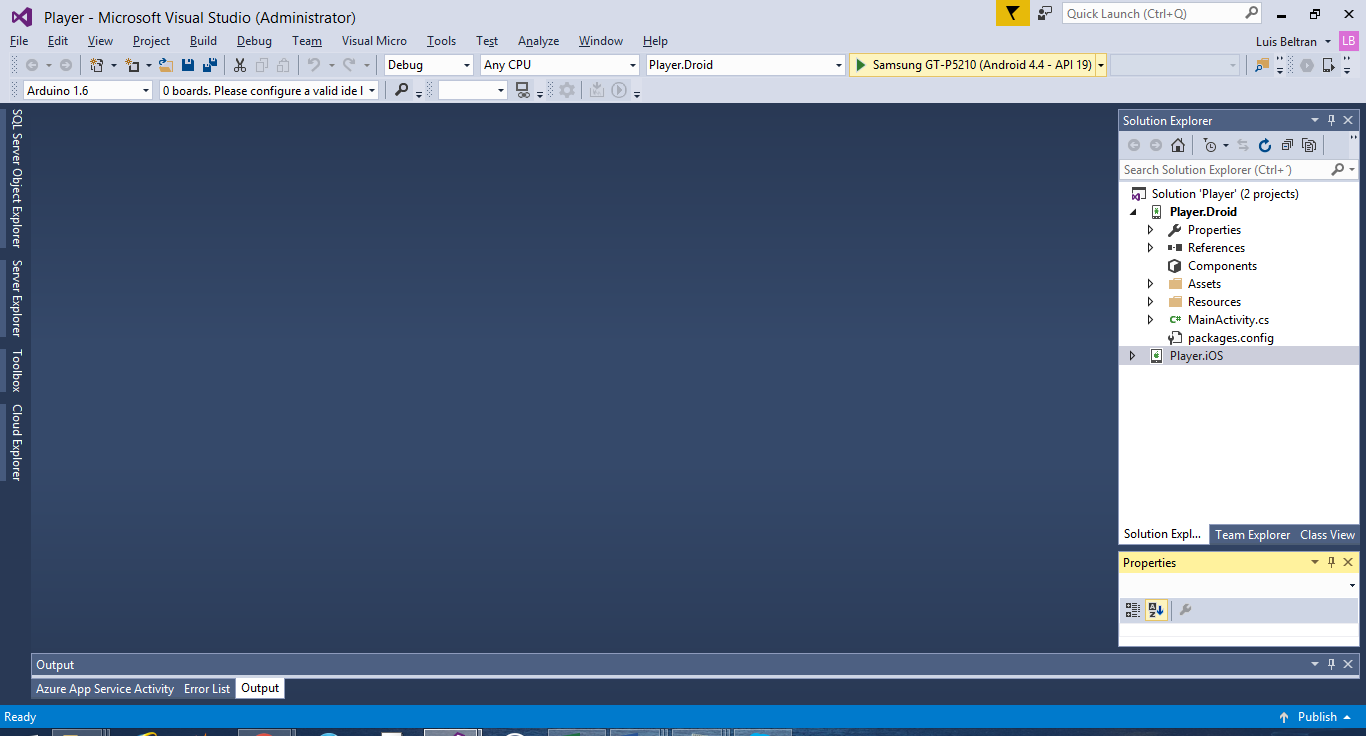


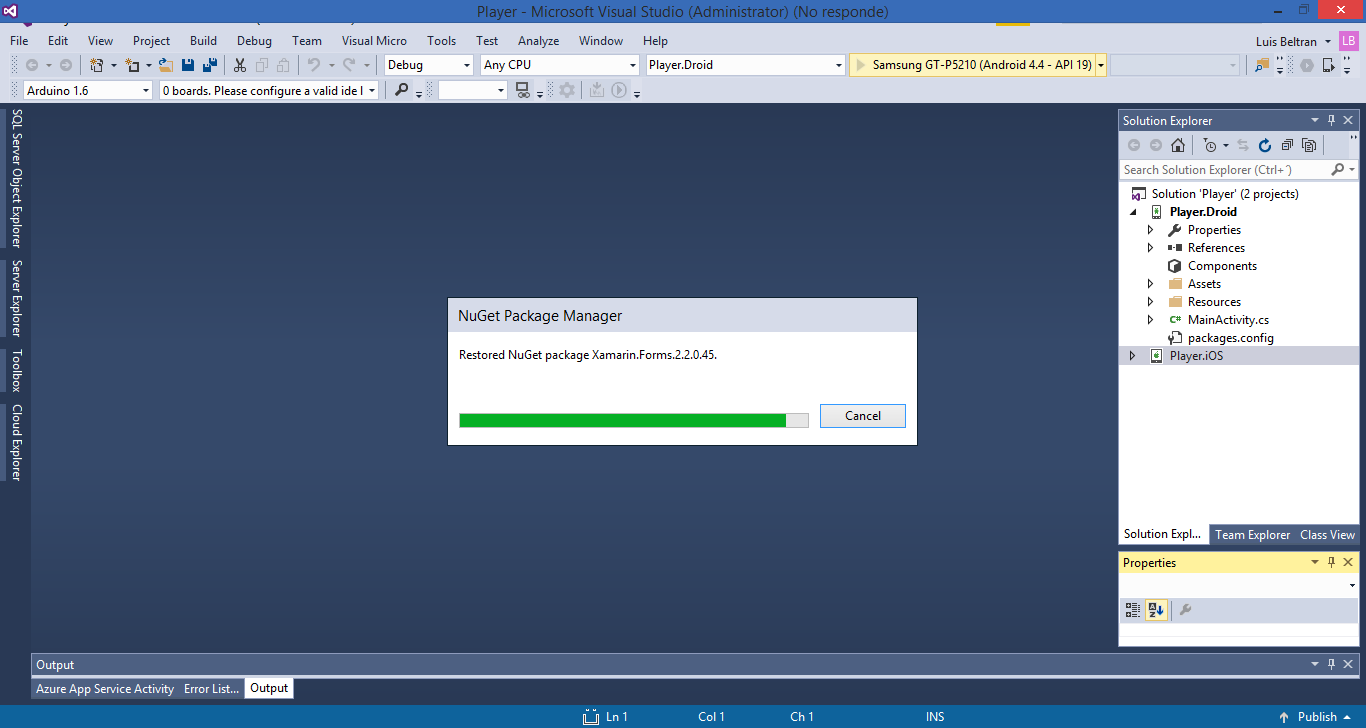


**Paso 10.** Abre el proyecto Player (revisa la ruta del **Paso 6** para localizarlo).



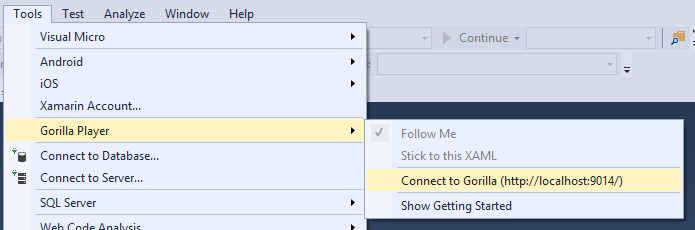
**Paso 11.** Compila y ejecuta el proyecto en tu dispositivo o emulador. Es probable que se descarguen y actualicen algunos paquetes requeridos para ejecutar el visualizador.



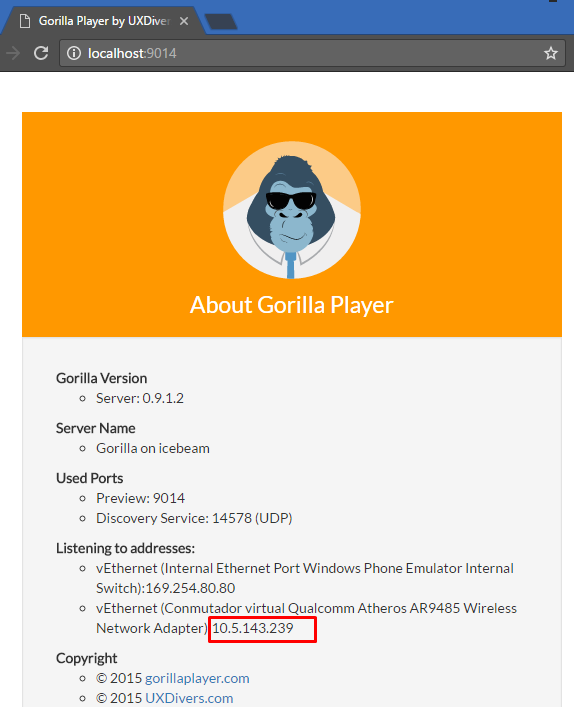


**Paso 12.** En caso de que la conexión con Gorilla Player no pueda realizarse, toma nota del puerto (y de la IP si estás probando en un dispositivo).

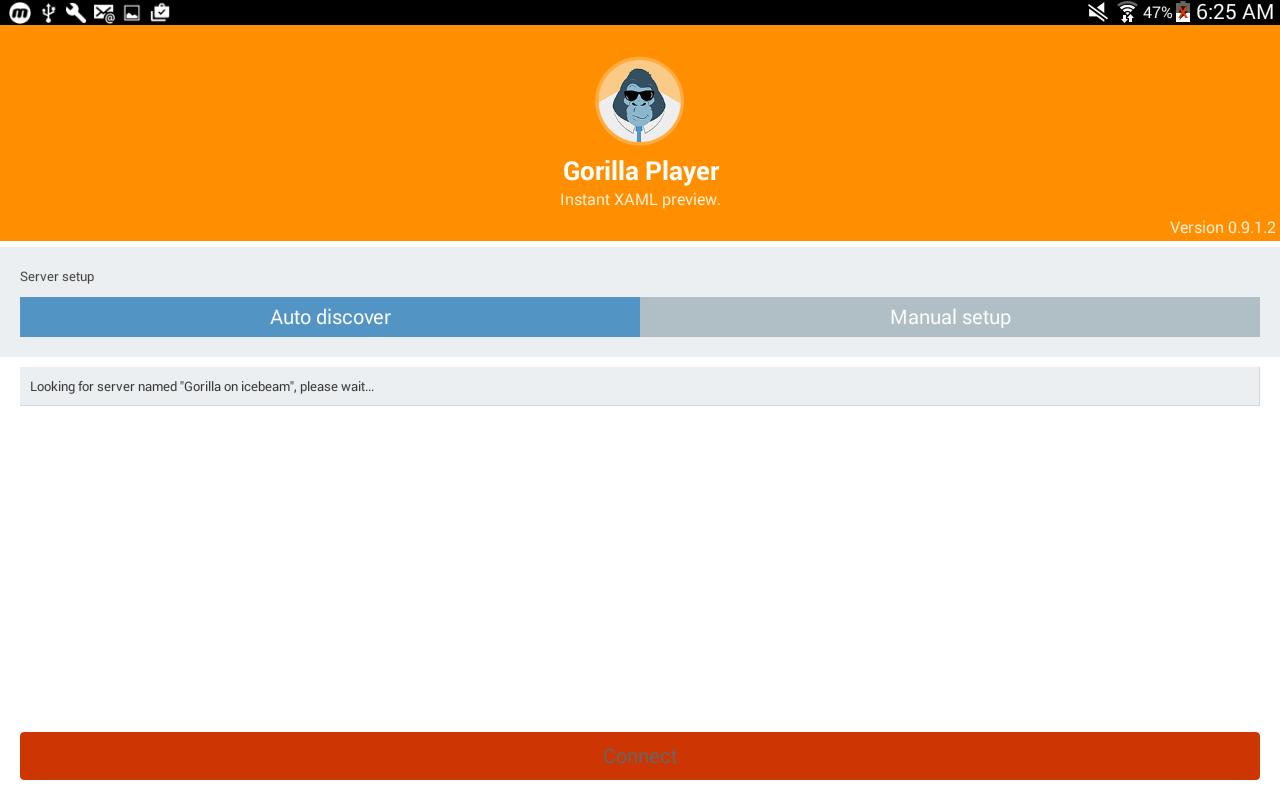
**Puerto:** Da clic en **Tools 🡪 Gorilla Player**. El enlace **Connect to Gorilla** indica el puerto de conexión.



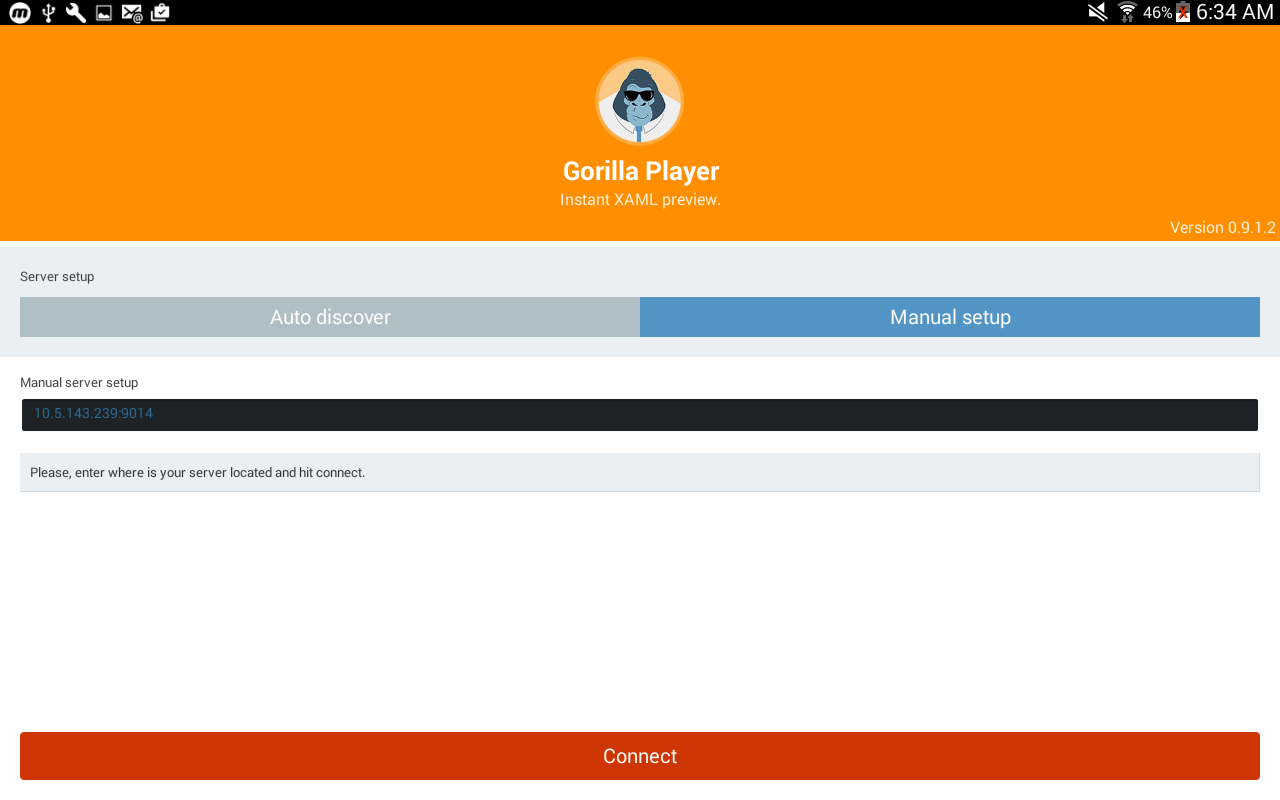
**Dirección IP:** En tu navegador, accede a <http://localhost:puerto> y revisa la IP que aparece en la conexión que estés utilizando (Ethernet, WiFi, etc).



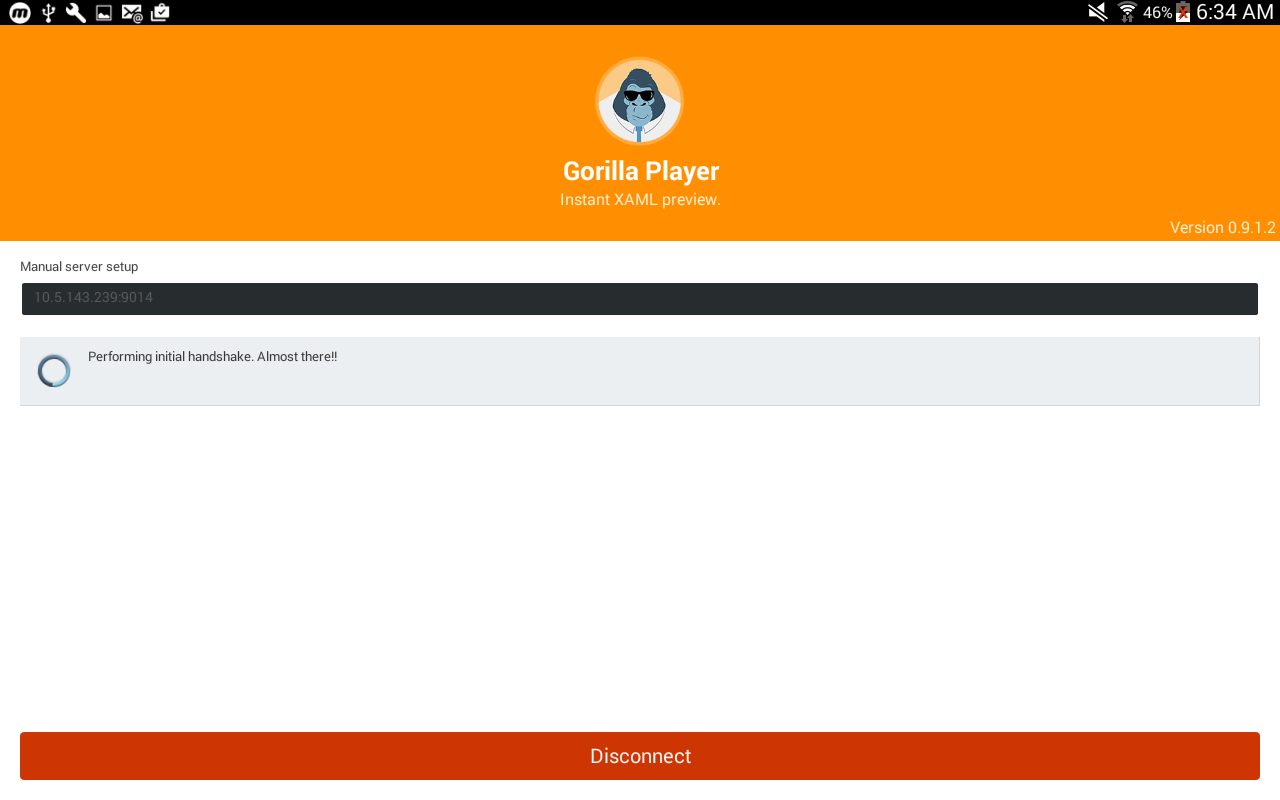
**Paso 13.** En tu dispositivo (o emulador) revisa si la conexión con Gorilla Player se pudo realizar. De manera predeterminada, Gorilla Player utiliza **Auto discover** para localizar el servicio.



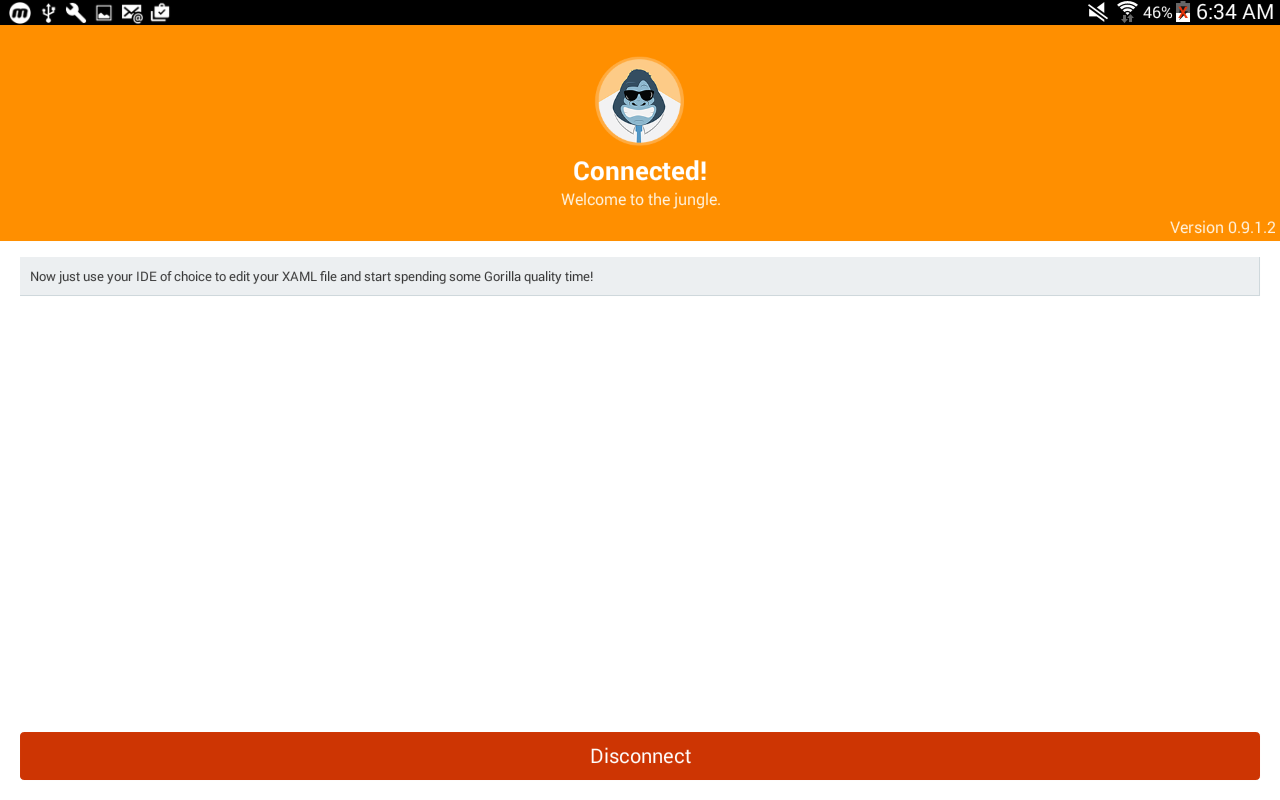
En caso de que no se pueda realizar la conexión (timeout), da clic en el botón Manual setup y coloca **localhost:puerto** (si estás desde el emulador) o **ip:puerto** (si utilizas un dispositivo físico). Da clic en **Connect**.



Un mensaje de handshake inicial indica que la conexión ha sido exitosa. Solo tendrás que esperar un momento más.

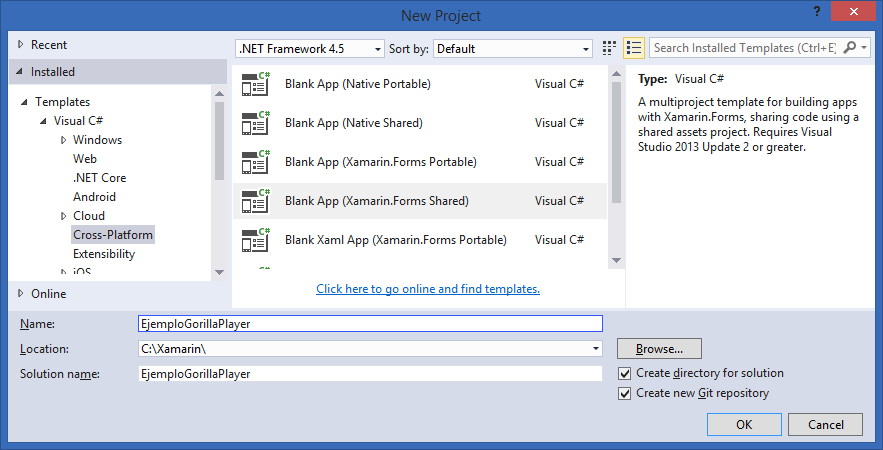


¡Listo! La conexión ha sido efectuada con éxito. Ahora, ya puedes crear un proyecto y visualizar el XAML de tu app.

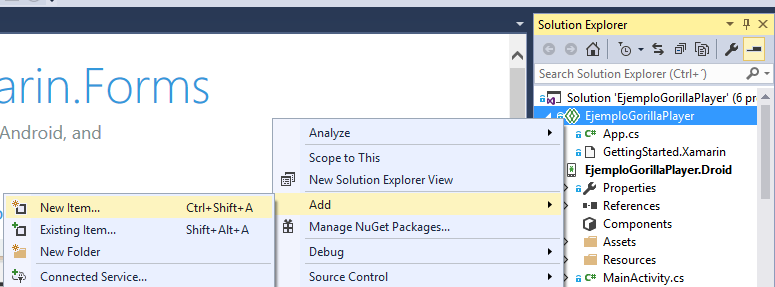


**NOTA: No cierres la app Player. Es necesario que siga en ejecución para visualizar el XAML de las aplicaciones que generes.**

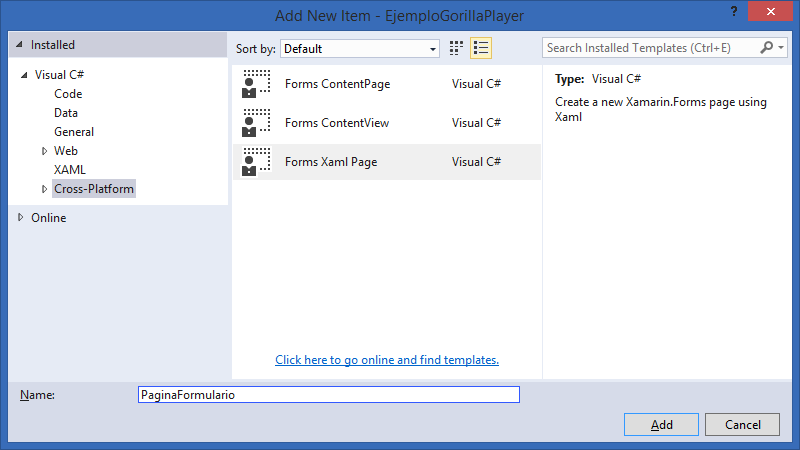
**Paso 14.** Crea un nuevo proyecto de tipo **Xamarin.Forms Shared** llamado **EjemploGorillaPlayer**.



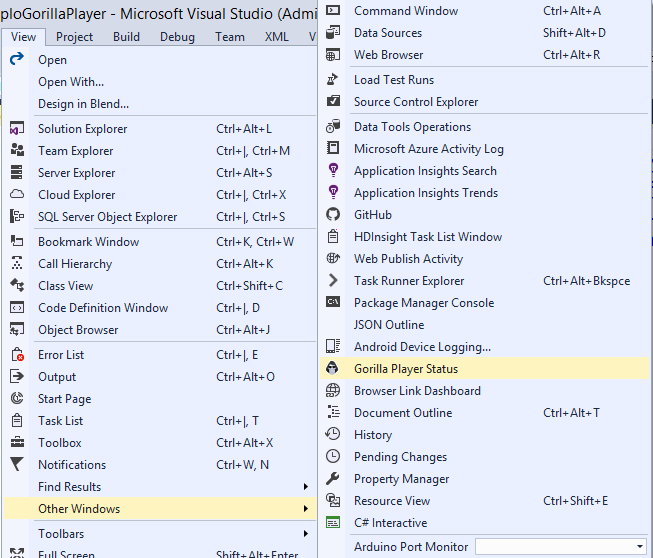
**Paso 15.** Directamente agregaremos una nueva página a nuestro proyecto. Da clic derecho sobre el proyecto **EjemploGorillaPlayer** y selecciona **Agregar 🡪 Nuevo elemento**.



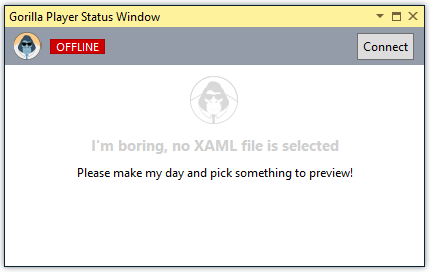
Selecciona **Forms.Xaml Page** en la categoría **Cross-Platform.** El nombre de la página es **PaginaFormulario**.

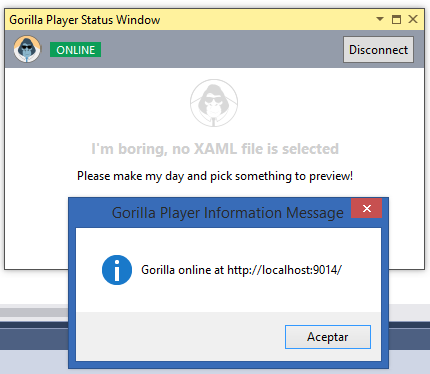


**Paso 16.** En caso de que el ícono de Gorilla Player no aparezca del lado derecho del XAML, inícialo seleccionando **Ver 🡪 Otras ventanas 🡪 Gorilla Player Status.**



**Paso 17.** Da clic en el botón **Connect**.





**Paso 18.** Escribe el siguiente código XAML.

<StackLayout BackgroundColor="White">

<Label TextColor="Blue" Text="Formulario de datos" FontSize="20" />

<Entry x:Name="txtNombre" WidthRequest="200" Placeholder="Escribe tu nombre..." />

<StackLayout Orientation="Horizontal">

<Label TextColor="Black" Text="Eres..." />

<Picker HorizontalOptions="FillAndExpand">

<Picker.Items>

<x:String>Mujer</x:String>

<x:String>Hombre</x:String>

</Picker.Items>

</Picker>

</StackLayout>

<StackLayout Orientation="Horizontal">

<Label TextColor="Black" Text="¿Cual es tu fecha de cumpleaños?" />

<DatePicker Format="dd-MM-yyyy" HorizontalOptions="Start"/>

</StackLayout>

<ActivityIndicator IsRunning="True"/>

<StackLayout Orientation="Horizontal">

<Label TextColor="Red" Text="Progreso" />

<ProgressBar Progress=".4" WidthRequest="200" />

</StackLayout>

<StackLayout Orientation="Horizontal">

<Label TextColor="#6622CC" Text="Calificación" />

<Slider x:Name="slider" Maximum="10" Minimum="5" Value="7" WidthRequest="200"/>

<Label BindingContext="{x:Reference slider}"

Text="{Binding Value, StringFormat='Tu valor es {0}'}"

FontAttributes="Bold" FontSize="Large" TextColor="Green"

HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

</StackLayout>

<StackLayout Orientation="Horizontal">

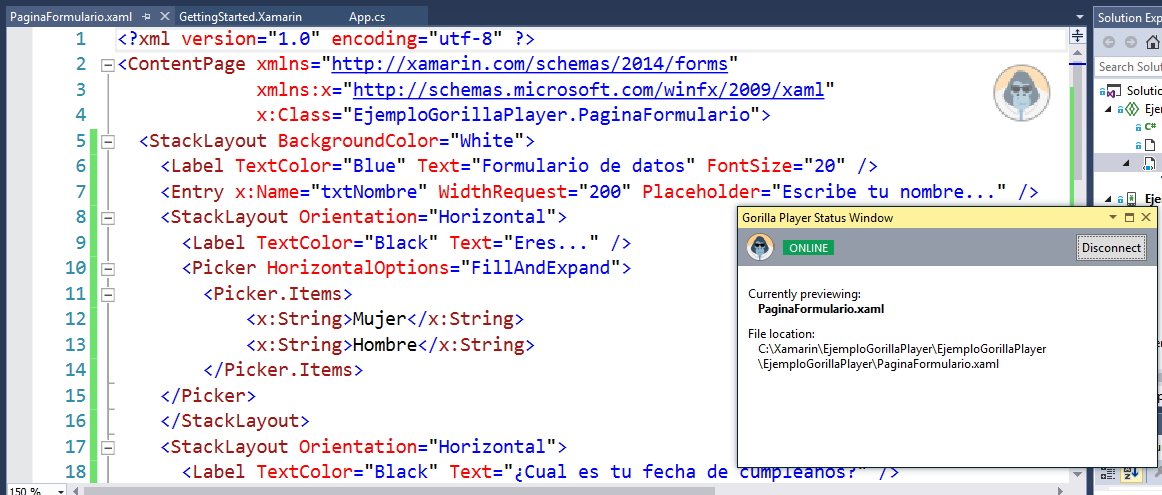
<Label TextColor="Lime" Text="¿Te gusta el futbol?" />

<Switch IsToggled="True"/>

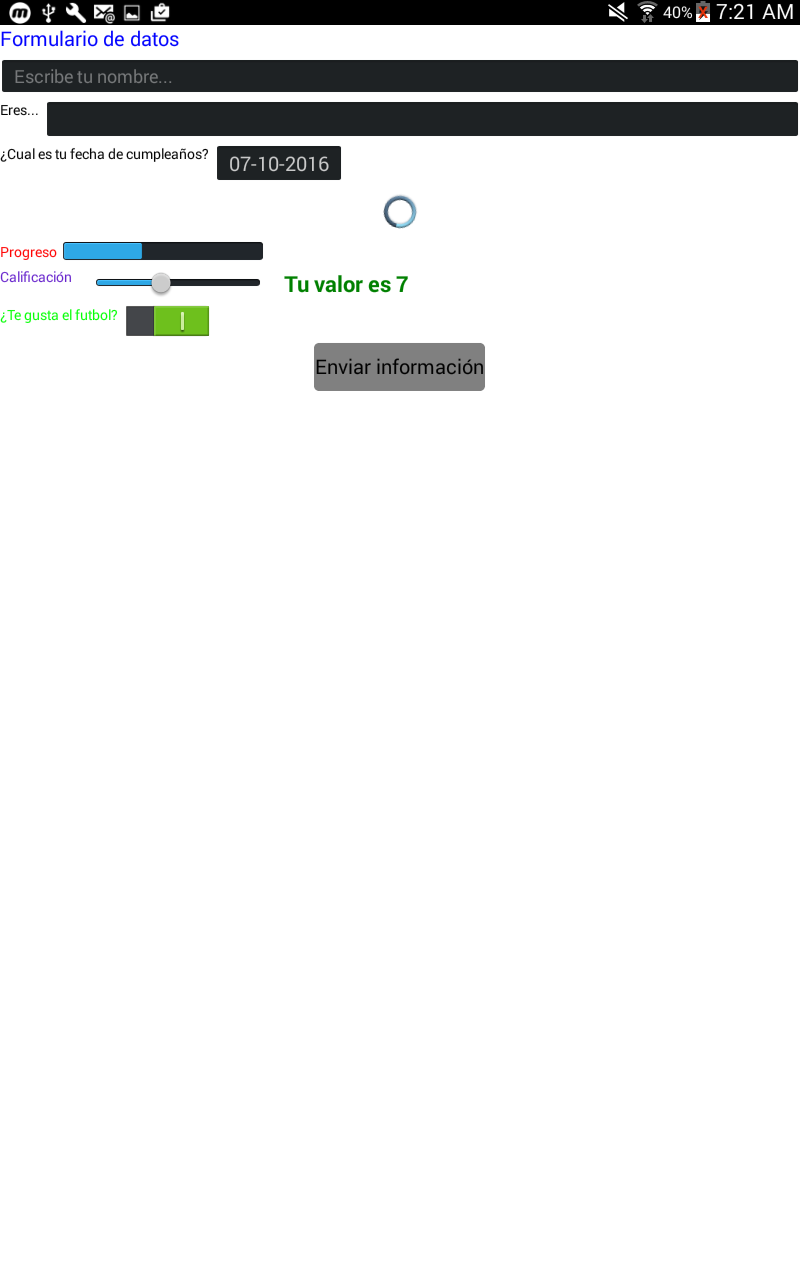
</StackLayout>

<Button TextColor="Black" BackgroundColor="Gray" Text="Enviar información" HorizontalOptions="Center"/>

</StackLayout>



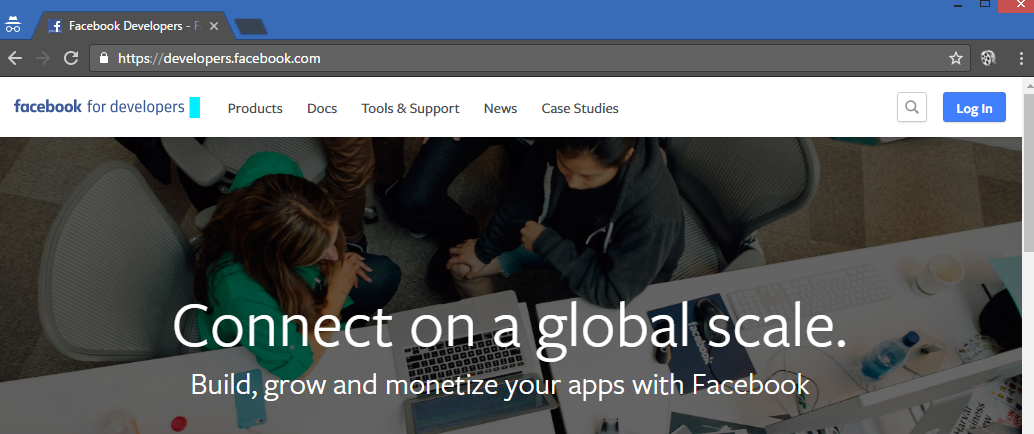
Cada vez que guardes el archivo, observarás que el diseño aparece en el emulador o dispositivo sin necesidad de ejecutar la aplicación actual.



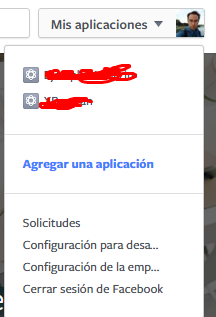
**¡Listo! En esta práctica has instalado, configurado y realizado una prueba inicial de GorillaPlayer. En una sesión posterior, se demostrará el uso del binding (enlace de datos) en tiempo de diseño, para demostrar su uso en el diseño de Listas.**

**Parte 2- Facebook**

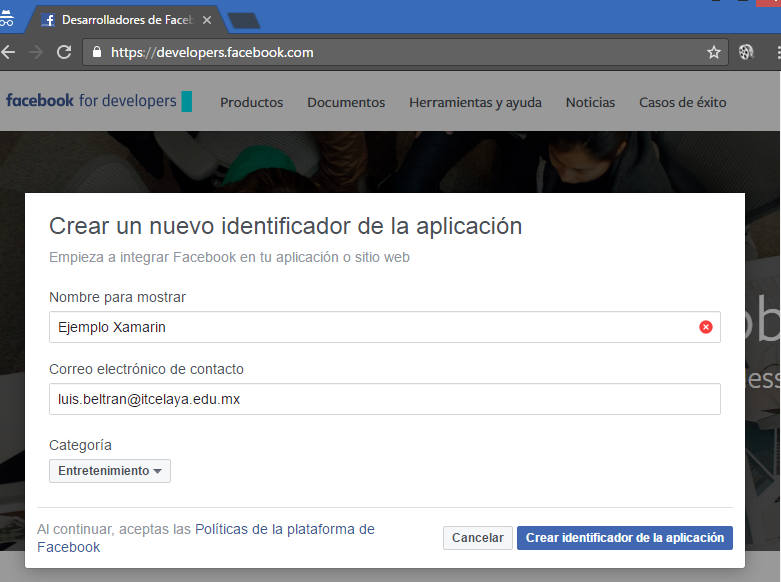
**Paso 19.** Ingresa a <https://developers.facebook.com/> y regístrate (si no lo has hecho). Inicia sesión.



**Paso 20.** Da clic en **Mis aplicaciones** y selecciona **Agregar una aplicación**.



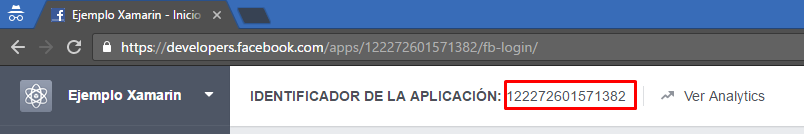
**Paso 21.** Crea un **nuevo identificador de aplicación** llenando el formulario. Ingresa un **nombre**, **correo electrónico de contacto** y selecciona una **categoría**. Al finalizar, da clic en el botón **Crear identificador de la aplicación**.



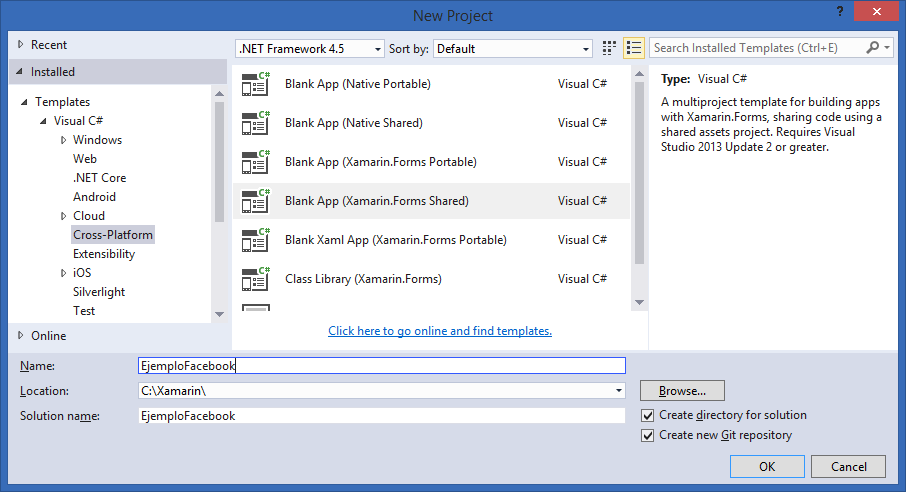
**Paso 22.** Selecciona Empezar en **Inicio de sesión en Facebook**. En la pantalla de **configuración del cliente de OAuth** rellena el campo **URI de redireccionamiento** escribiendo la siguiente dirección: <http://www.facebook.com/connect/login_success.html>



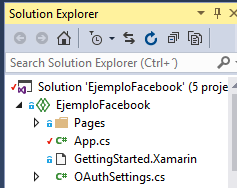
**Paso 23.** En la parte superior ubica el **id de la aplicación**. Cópialo en un bloc de notas, lo utilizaremos más adelante.



**Paso 24.** Crea un nuevo proyecto de tipo **Xamarin.Forms Shared** llamado **EjemploFacebook**.



**Paso 25.** Crea una carpeta en el proyecto **EjemploFacebook** con el nombre **Pages**.



**Paso 26.** Agrega 3 clases a la carpeta Pages: **BaseContentPage**, **LoginPage** y **ProfilePage**. El código de cada clase se muestra a continuación:

**BaseContentPage.cs**

using Xamarin.Forms;

namespace EjemploFacebook.Pages

{

public class BaseContentPage : ContentPage

{

protected override void OnAppearing()

{

base.OnAppearing();

if (!App.Instance.IsAuthenticated)

Navigation.PushModalAsync(new LoginPage());

}

}

}

**LoginPage.cs**

using Xamarin.Forms;

namespace EjemploFacebook.Pages

{

public class LoginPage : ContentPage

{ }

}

**ProfilePage.cs**

using Xamarin.Forms;

namespace EjemploFacebook.Pages

{

public class ProfilePage : BaseContentPage

{

public ProfilePage()

{

// Using messaging center to ensure that content only gets loaded once authentication is successful.

MessagingCenter.Subscribe<App>(this, "Authenticated", (sender) => {

Content = new Label() {

Text = "Profile Page",

XAlign = TextAlignment.Center,

YAlign = TextAlignment.Center

};

});

}

}

}

**Paso 27.** En raíz del proyecto **EjemploFacebook**, agrega la clase **OAuthSettings** con el siguiente código:

namespace EjemploFacebook

{

public class OAuthSettings

{

public OAuthSettings(

string clientId,

string scope,

string authorizeUrl,

string redirectUrl)

{

ClientId = clientId;

Scope = scope;

AuthorizeUrl = authorizeUrl;

RedirectUrl = redirectUrl;

}

public string ClientId { get; private set; }

public string Scope { get; private set; }

public string AuthorizeUrl { get; private set; }

public string RedirectUrl { get; private set; }

}

}

**Paso 28.** Modifica la clase **App.cs** para que tenga el código mostrado a continuación. NOTA: Cuando se crea la instancia de **OAuthSettings**, el parámetro **clientId** tendrá el **id de la aplicación de Facebook** mostrada en el **paso 23**:

using EjemploFacebook.Pages;

using System;

using Xamarin.Forms;

namespace EjemploFacebook

{

public class App : Application {

static volatile App \_Instance;

static object \_SyncRoot = new Object();

public static App Instance {

get {

if (\_Instance == null) {

lock (\_SyncRoot) {

if (\_Instance == null) {

\_Instance = new App();

\_Instance.OAuthSettings =

new OAuthSettings(

clientId: "TU-CLIENT-ID",

scope: "",

authorizeUrl: "https://m.facebook.com/dialog/oauth/",

redirectUrl: "http://www.facebook.com/connect/login\_success.html");

}

}

}

return \_Instance;

}

}

public OAuthSettings OAuthSettings { get; private set; }

NavigationPage \_NavPage;

public Page GetMainPage(){

var profilePage = new ProfilePage();

\_NavPage = new NavigationPage(profilePage);

return \_NavPage;

}

public bool IsAuthenticated {

get { return !string.IsNullOrWhiteSpace(\_Token); }

}

string \_Token;

public string Token { get { return \_Token; } }

public void SaveToken(string token) {

\_Token = token;

// broadcast a message that authentication was successful

MessagingCenter.Send<App>(this, "Authenticated");

}

public Action SuccessfulLoginAction {

get { return new Action(() => \_NavPage.Navigation.PopModalAsync());}

}

protected override void OnStart () { }

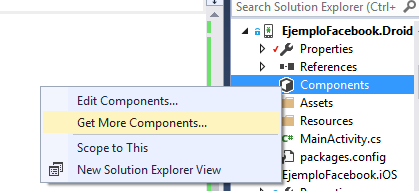
protected override void OnSleep () { }

protected override void OnResume () { }

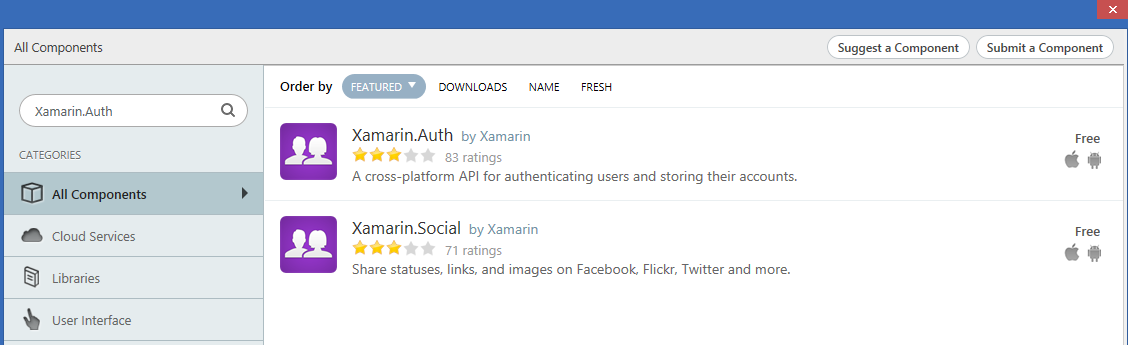
}

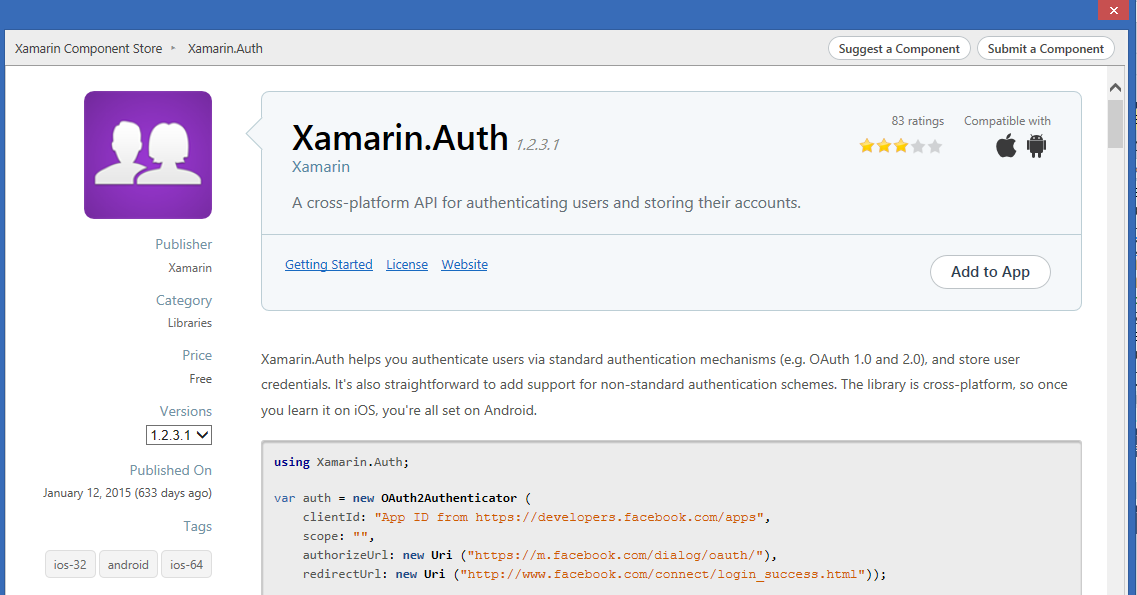
}

**Paso 29.** En el proyecto de Android, da clic derecho en **Components** y selecciona **Get more components**:

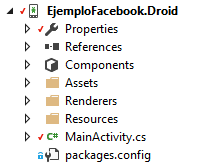


**Paso 30.** Busca **Xamarin.Auth**, da clic en el resultado que aparece y da clic en **Add to App**.





**Paso 31.** En el proyecto de **Android**, agrega la carpeta **Renderers**:



**Paso 32.** En la carpeta **Renderers**, agrega una nueva clase llamada **LoginPageRenderer**, con el siguiente código:

using System;

using Android.App;

using EjemploFacebook.Pages;

using Xamarin.Auth;

using Xamarin.Forms;

using Xamarin.Forms.Platform.Android;

using EjemploFacebook.Droid.Renderers;

[assembly: ExportRenderer(typeof(LoginPage), typeof(LoginPageRenderer))]

namespace EjemploFacebook.Droid.Renderers

{

public class LoginPageRenderer : PageRenderer

{

protected override void OnElementChanged(ElementChangedEventArgs<Page> e)

{

base.OnElementChanged(e);

var activity = this.Context as Activity;

var auth = new OAuth2Authenticator(

clientId: App.Instance.OAuthSettings.ClientId,

scope: App.Instance.OAuthSettings.Scope,

authorizeUrl: new Uri(App.Instance.OAuthSettings.AuthorizeUrl),

redirectUrl: new Uri(App.Instance.OAuthSettings.RedirectUrl));

auth.Completed += (sender, eventArgs) => {

if (eventArgs.IsAuthenticated)

{

App.Instance.SuccessfulLoginAction.Invoke();

// Use eventArgs.Account to do wonderful things

App.Instance.SaveToken(eventArgs.Account.Properties["access\_token"]);

}

else

{

// The user cancelled

}

};

activity.StartActivity(auth.GetUI(activity));

}

}

}

**Paso 33.** Finalmente, modifica **MainActivity.cs** para que tenga el siguiente código:

using Android.App;

using Android.Content.PM;

using Android.OS;

using Xamarin.Forms.Platform.Android;

namespace EjemploFacebook.Droid

{

[Activity (Label = "EjemploFacebook", Icon = "@drawable/icon", MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation)]

public class MainActivity : AndroidActivity

{

protected override void OnCreate (Bundle bundle)

{

base.OnCreate (bundle);

global::Xamarin.Forms.Forms.Init (this, bundle);

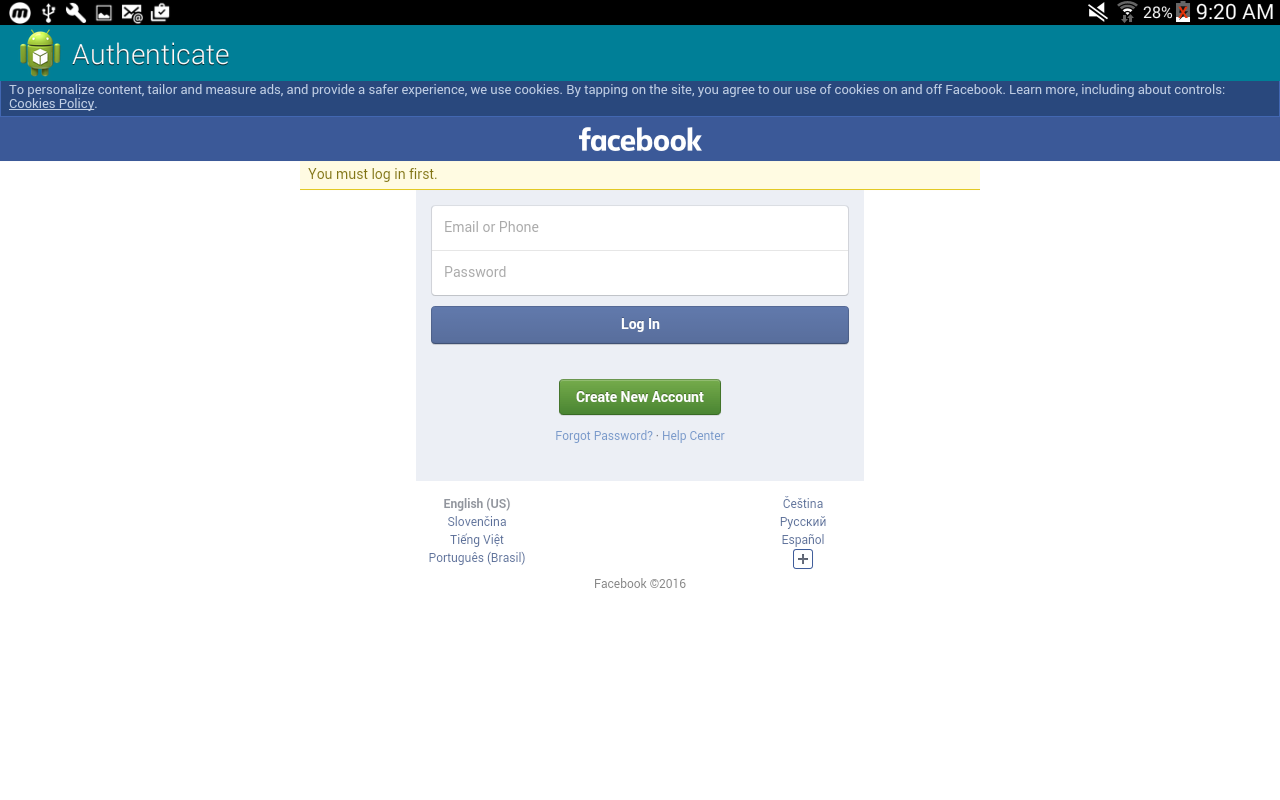
SetPage(App.Instance.GetMainPage());

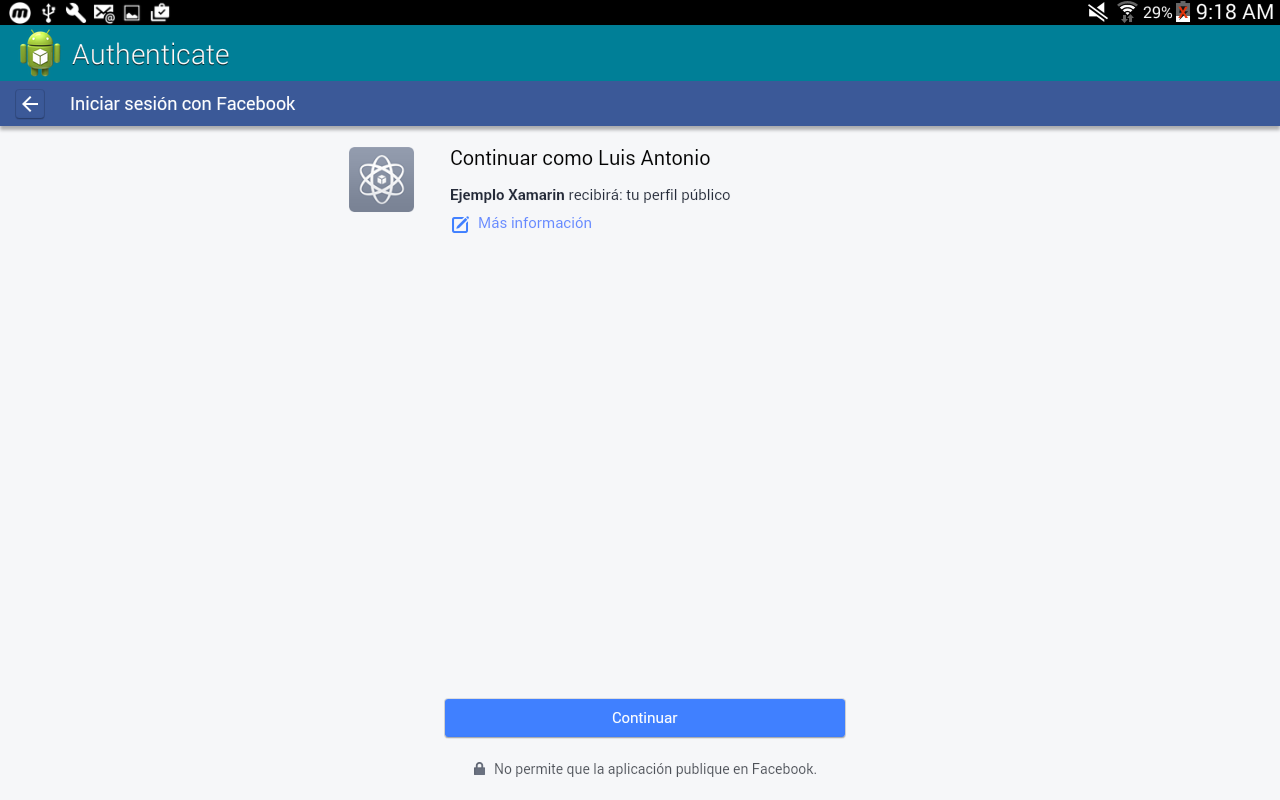
}

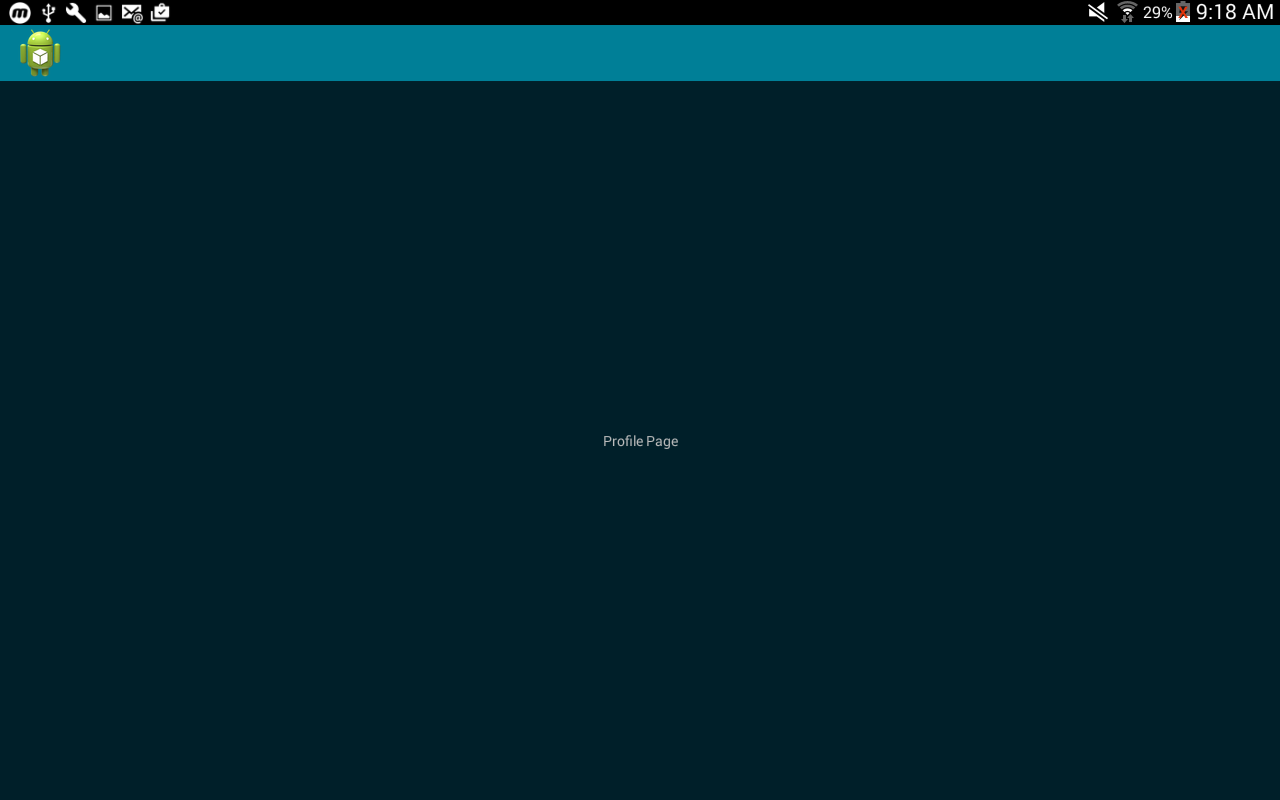
}

}

**Paso 34.** Compila y ejecuta la aplicación. Se debe mostrar la pantalla de **inicio de sesión de Facebook**. Si ingresas, recibirás una **solicitud de permiso** de la aplicación, el cual **al ser aceptado** mostrará la página **ProfilePage**.







**¡Listo! En esta práctica has realizado los pasos básicos para autentificación en Facebook con Xamarin.Forms. En una sesión posterior, se mostrará información del perfil, lista de amigos y podremos realizar acciones en Facebook desde la app.**