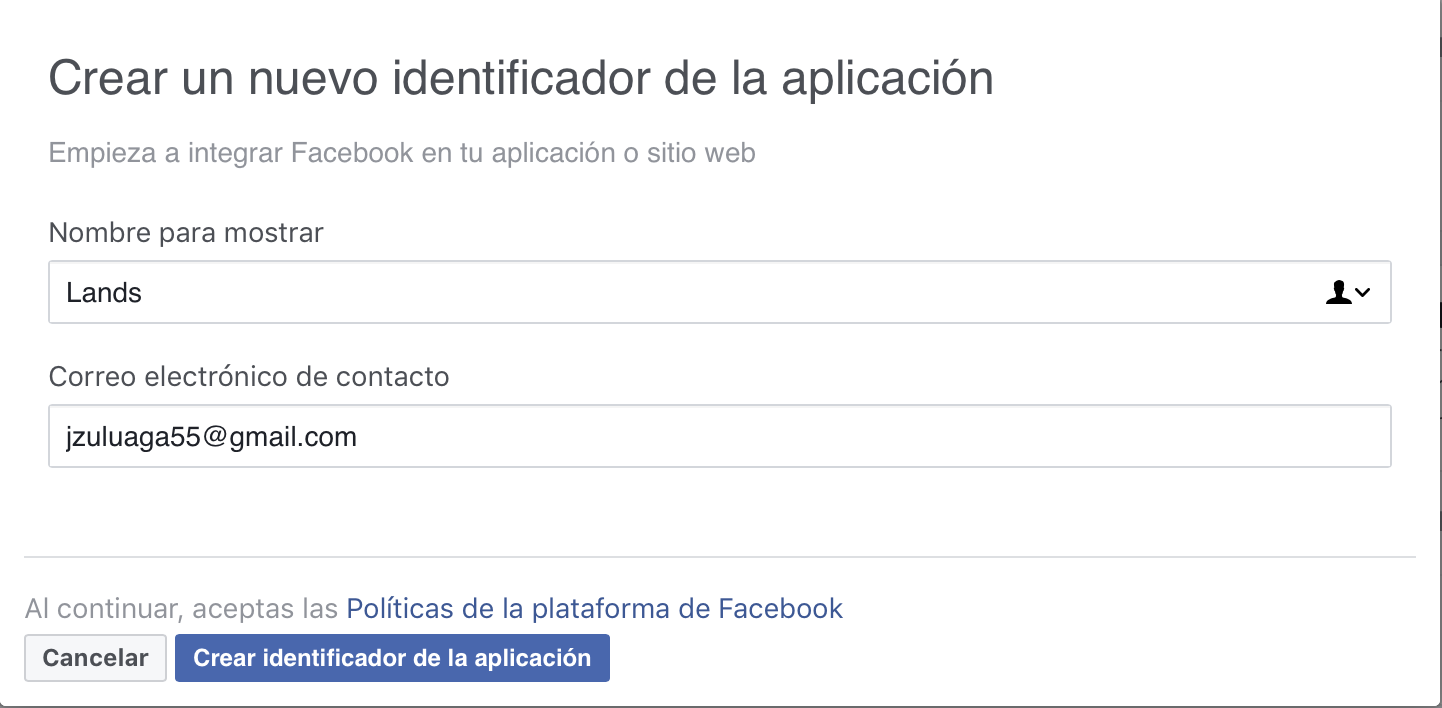
Facebook Login with Xamarin – V2 (MVVM) 

1. Ingrese a <https://developers.facebook.com/apps> y crea una nueva App



Ingresa los dastos del captcha:



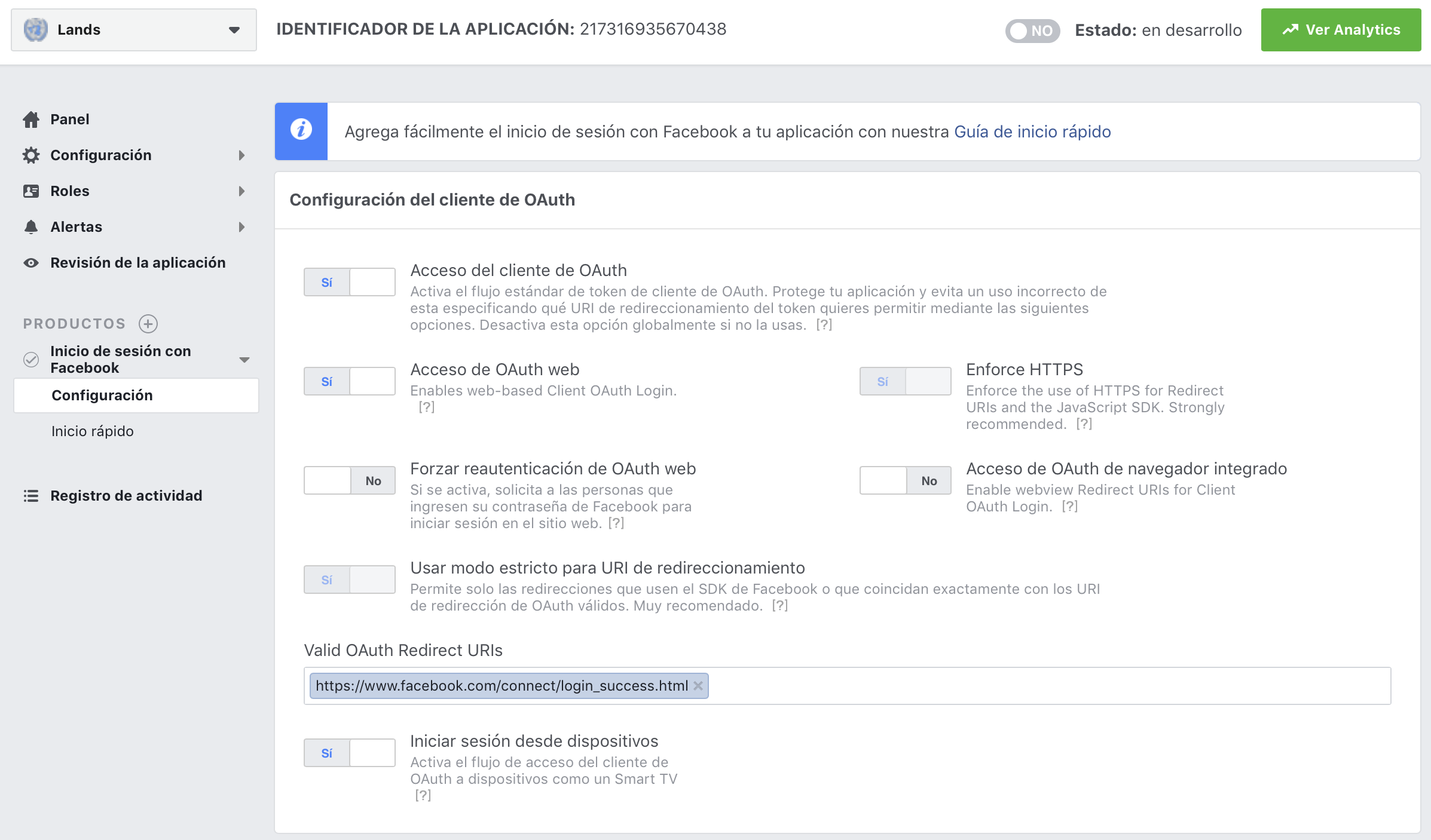
Ve a inicio de sesión:



Luego a configuración básica y llena la URL de política de privacidad, puede ser con: <http://www.zulu-software.com> y opcional coloca un ícono de tamaño 1024 x 1024, y selcciona una categoría en nuestro caso, colocamos: Educación:



Luego ve a inicio de sesión con Facebook, configuración y coloca la URL de retorno en: <https://www.facebook.com/connect/login_success.html> y los siguientes suiches:



Por último ve a revisión de la aplicación y colócala pública:



Ten el identificador de la App a la mano que lo vamos a necesitar más adelante.

1. Cree la vista **LoginFacebookPage**:

<?xml version="1.0" encoding="UTF-8"?>  
<ContentPage   
    xmlns="http://xamarin.com/schemas/2014/forms"   
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"   
    xmlns:i18n="clr-namespace:Lands.Helpers"  
    x:Class="Lands.Views.LoginFacebookPage"  
    Title="{i18n:Translate Login}">  
    <ContentPage.Content>  
    </ContentPage.Content>  
</ContentPage>

1. Cree los modelos del **FacebookResponse** (también debes crearlo en el proyecto **Domain**)

namespace Lands.Models  
{  
    using System.Collections.Generic;  
    using Newtonsoft.Json;  
  
    public class DataFacebook  
    {  
        [JsonProperty(PropertyName = "is\_silhouette")]  
        public bool IsSilhouette { get; set; }  
  
        [JsonProperty(PropertyName = "url")]  
        public string Url { get; set; }  
    }  
  
    public class Picture  
    {  
        [JsonProperty(PropertyName = "data")]  
        public DataFacebook Data { get; set; }  
    }  
  
    public class Cover  
    {  
        [JsonProperty(PropertyName = "id")]  
        public string Id { get; set; }  
  
        [JsonProperty(PropertyName = "offset\_y")]  
        public int OffsetY { get; set; }  
  
        [JsonProperty(PropertyName = "source")]  
        public string Source { get; set; }  
    }  
  
    public class AgeRange  
    {  
        [JsonProperty(PropertyName = "min")]  
        public int Min { get; set; }  
    }  
  
    public class Device  
    {  
        [JsonProperty(PropertyName = "hardware")]  
        public string Hardware { get; set; }  
  
        [JsonProperty(PropertyName = "os")]  
        public string Os { get; set; }  
    }  
  
    public class FacebookResponse  
    {  
        [JsonProperty(PropertyName = "name")]  
        public string Name { get; set; }  
  
        [JsonProperty(PropertyName = "picture")]  
        public Picture Picture { get; set; }  
  
        [JsonProperty(PropertyName = "cover")]  
        public Cover Cover { get; set; }  
  
        [JsonProperty(PropertyName = "age\_range")]  
        public AgeRange AgeRange { get; set; }  
  
        [JsonProperty(PropertyName = "devices")]  
        public List<Device> Devices { get; set; }  
  
        [JsonProperty(PropertyName = "gender")]  
        public string Gender { get; set; }  
  
        [JsonProperty(PropertyName = "is\_verified")]  
        public bool IsVerified { get; set; }  
  
        [JsonProperty(PropertyName = "locale")]  
        public string Locale { get; set; }  
  
        [JsonProperty(PropertyName = "link")]  
        public string Link { get; set; }  
  
        [JsonProperty(PropertyName = "first\_name")]  
        public string FirstName { get; set; }  
  
        [JsonProperty(PropertyName = "last\_name")]  
        public string LastName { get; set; }  
  
        [JsonProperty(PropertyName = "id")]  
        public string Id { get; set; }  
    }  
}

1. Modifique el controlador de usuarios en el API y agruegue el siguiente método**,** publique el nuevo API:

[HttpPost]

[Route("LoginFacebook")]

public async Task<IHttpActionResult> LoginFacebook(FacebookResponse profile)

{

try

{

var user = await db.Users.Where(u => u.Email == profile.Id).FirstOrDefaultAsync();

if (user == null)

{

user = new User

{

Email = profile.Id,

FirstName = profile.FirstName,

LastName = profile.LastName,

ImagePath = profile.Picture.Data.Url,

UserTypeId = 2,

Telephone = "...",

};

db.Users.Add(user);

UsersHelper.CreateUserASP(profile.Id, "User", profile.Id);

}

else

{

user.FirstName = profile.FirstName;

user.LastName = profile.LastName;

user.ImagePath = profile.Picture.Data.Url;

db.Entry(user).State = EntityState.Modified;

}

await db.SaveChangesAsync();

return Ok(true);

}

catch (DbEntityValidationException e)

{

var message = string.Empty;

foreach (var eve in e.EntityValidationErrors)

{

message = string.Format("Entity of type \"{0}\" in state \"{1}\" has the following validation errors:",

eve.Entry.Entity.GetType().Name, eve.Entry.State);

foreach (var ve in eve.ValidationErrors)

{

message += string.Format("\n- Property: \"{0}\", Error: \"{1}\"",

ve.PropertyName, ve.ErrorMessage);

}

}

return BadRequest(message);

}

catch (Exception ex)

{

return BadRequest(ex.Message);

}

}

1. Adicione esto método al **ApiService.cs**:

public async Task<TokenResponse> LoginFacebook(  
    string urlBase,   
    string servicePrefix,  
    string controller,   
    Models.FacebookResponse profile)  
{  
    try  
    {  
        var request = JsonConvert.SerializeObject(profile);  
        var content = new StringContent(  
            request,   
            Encoding.UTF8,   
            "application/json");  
        var client = new HttpClient();  
        client.BaseAddress = new Uri(urlBase);  
        var url = string.Format("{0}{1}", servicePrefix, controller);  
        var response = await client.PostAsync(url, content);  
  
        if (!response.IsSuccessStatusCode)  
        {  
            return null;  
        }  
  
        var tokenResponse = await GetToken(  
            urlBase,   
            profile.Id,  
            profile.Id);  
        return tokenResponse;  
    }  
    catch  
    {  
        return null;  
    }  
}

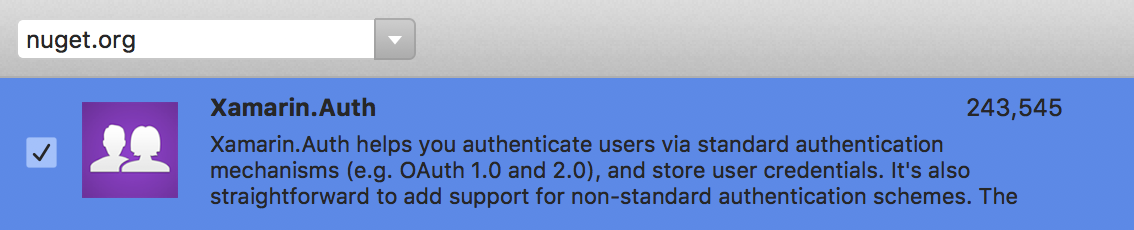
1. Modifique la propiedad **ImageFullPath** de la clase **UserLocal.cs**:

public string ImageFullPath  
{  
    get  
    {  
        if (string.IsNullOrEmpty(ImagePath))  
        {  
            return "noimage";  
        }  
  
        if (this.UserTypeId == 1)  
        {  
            return string.Format(  
                "http://landsapi1.azurewebsites.net/{0}",  
                ImagePath.Substring(1));  
        }  
  
        return ImagePath;  
    }  
}

1. Adicione estos métodos al **App.cs**:

public static Action HideLoginView  
{  
    get  
    {  
        return new Action(() => Application.Current.MainPage =   
                          new NavigationPage(new LoginPage()));  
    }  
}  
  
public static async Task NavigateToProfile(Models.FacebookResponse profile)  
{  
    if (profile == null)  
    {  
        Application.Current.MainPage = new NavigationPage(new LoginPage());  
        return;  
    }  
  
    var apiService = new ApiService();  
    var dataService = new DataService();  
  
    var apiSecurity = Application.Current.Resources["APISecurity"].ToString();  
    var token = await apiService.LoginFacebook(  
        apiSecurity,   
        "/api",   
        "/Users/LoginFacebook",   
        profile);  
  
    if (token == null)  
    {  
        Application.Current.MainPage = new NavigationPage(new LoginPage());  
        return;  
    }  
  
    var user = await apiService.GetUserByEmail(  
        apiSecurity,  
        "/api",  
        "/Users/GetUserByEmail",  
        token.TokenType,  
        token.AccessToken,  
        token.UserName);  
  
    UserLocal userLocal = null;  
    if (user != null)  
    {  
        userLocal = Converter.ToUserLocal(user);  
        dataService.DeleteAllAndInsert(userLocal);  
    }  
  
    var mainViewModel = MainViewModel.GetInstance();  
    mainViewModel.Token = token;  
    mainViewModel.User = userLocal;  
    mainViewModel.Lands = new LandsViewModel();  
    Application.Current.MainPage = new MasterPage();  
    Settings.IsRemembered = "true";  
  
    mainViewModel.Lands = new LandsViewModel();  
    Application.Current.MainPage = new MasterPage();  
}

1. Agregar a los proyectos Android y iOS el componente **Xamarin.Auth**:



1. En el **ApiService** adicionar este médoto:

public async Task<Models.FacebookResponse> GetFacebook(string accessToken)  
{  
    var requestUrl = "https://graph.facebook.com/v2.8/me/?fields=name," +  
        "picture.width(999),cover,age\_range,devices,email,gender," +  
        "is\_verified,birthday,languages,work,website,religion," +  
        "location,locale,link,first\_name,last\_name," +  
        "hometown&access\_token=" + accessToken;  
    var httpClient = new HttpClient();  
    var userJson = await httpClient.GetStringAsync(requestUrl);  
    var facebookResponse = JsonConvert.DeserializeObject<Models.FacebookResponse>(userJson);  
    return facebookResponse;  
}

1. En el proyecto **Android** en la carpeta **Implementations** crear el **LoginPageRenderer** con el siguiente código:

[assembly: Xamarin.Forms.ExportRenderer(

typeof(Lands.Views.LoginFacebookPage),

typeof(Lands.Droid.Implementations.LoginPageRenderer))]

namespace Lands.Droid.Implementations

{

using System;

using System.Threading.Tasks;

using Android.App;

using Models;

using Services;

using Xamarin.Auth;

using Xamarin.Forms.Platform.Android;

public class LoginPageRenderer : PageRenderer

{

public LoginPageRenderer()

{

var activity = this.Context as Activity;

var auth = new OAuth2Authenticator(

clientId: "201105497150884",

scope: "email",

authorizeUrl: new Uri("https://www.facebook.com/dialog/oauth/"),

redirectUrl: new Uri("https://www.facebook.com/connect/login\_success.html"));

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

{

if (eventArgs.IsAuthenticated)

{

var accessToken = eventArgs.Account.Properties["access\_token"].ToString();

var profile = await GetFacebookProfileAsync(accessToken);

await App.NavigateToProfile(profile);

}

else

{

App.HideLoginView();

}

};

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

}

private async Task<FacebookResponse> GetFacebookProfileAsync(string accessToken)

{

var requestUrl = "https://graph.facebook.com/v2.8/me/?fields=name,picture.width(999),cover," +

"age\_range,devices,email,gender,is\_verified,birthday,languages,work,website,religion," +

"location,locale,link,first\_name,last\_name,hometown&access\_token=" + accessToken;

var apiService = new ApiService();

return await apiService.GetFacebookProfile(requestUrl);

}

}

}

No olvide cambiar el **clientId** por el suministrado a su aplicación en la página de desarrolladores de Facebook.

1. De manera similar, en el proyecto **iOS** crear el **LoginPageRenderer** con el siguiente código:

[assembly: Xamarin.Forms.ExportRenderer(  
    typeof(Lands.Views.LoginFacebookPage),  
    typeof(Lands.iOS.Implementations.LoginPageRenderer))]  
  
namespace Lands.iOS.Implementations  
{  
    using System;  
    using System.Threading.Tasks;  
    using Models;  
    using Services;  
    using Xamarin.Auth;  
    using Xamarin.Forms.Platform.iOS;  
  
    public class LoginPageRenderer : PageRenderer  
    {  
        bool done = false;  
  
        public override void ViewDidAppear(bool animated)  
        {  
            base.ViewDidAppear(animated);  
  
            if (done)  
            {  
                return;  
            }  
  
            var auth = new OAuth2Authenticator(  
                clientId: "1263595770461023",  
                scope: "",  
                authorizeUrl: new Uri("https://www.facebook.com/dialog/oauth/"),  
                redirectUrl: new Uri("https://www.facebook.com/connect/login\_success.html"));  
  
            auth.Completed += async (sender, eventArgs) =>  
            {  
                DismissViewController(true, null);  
                App.HideLoginView();  
  
                if (eventArgs.IsAuthenticated)  
                {  
                    var accessToken = eventArgs.Account.Properties["access\_token"].ToString();  
                    var profile = await GetFacebookProfileAsync(accessToken);  
                    await App.NavigateToProfile(profile);  
                }  
                else  
                {  
                    await App.NavigateToProfile(null);  
                }  
            };  
  
            done = true;  
            PresentViewController(auth.GetUI(), true, null);  
        }  
  
        private async Task<FacebookResponse> GetFacebookProfileAsync(string accessToken)  
        {  
            var apiService = new ApiService();  
            return await apiService.GetFacebook(accessToken);  
        }  
    }  
}

No olvide cambiar el **clientId** por el suministrado a su aplicación en la página de desarrolladores de Facebook.

1. Crea el comando en la **LoginViewModel**:

public ICommand LoginFacebookComand  
{  
    get  
    {  
        return new RelayCommand(LoginFacebook);  
    }  
}  
  
private async void LoginFacebook()  
{  
    await Application.Current.MainPage.Navigation.PushAsync(new LoginFacebookPage());  
}

1. Probemos!!!