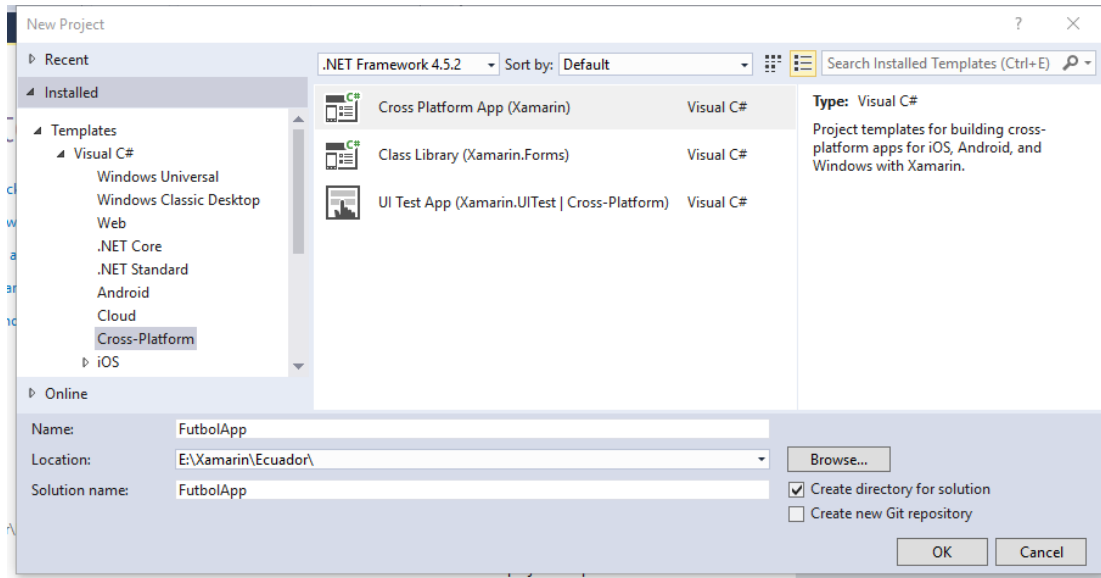


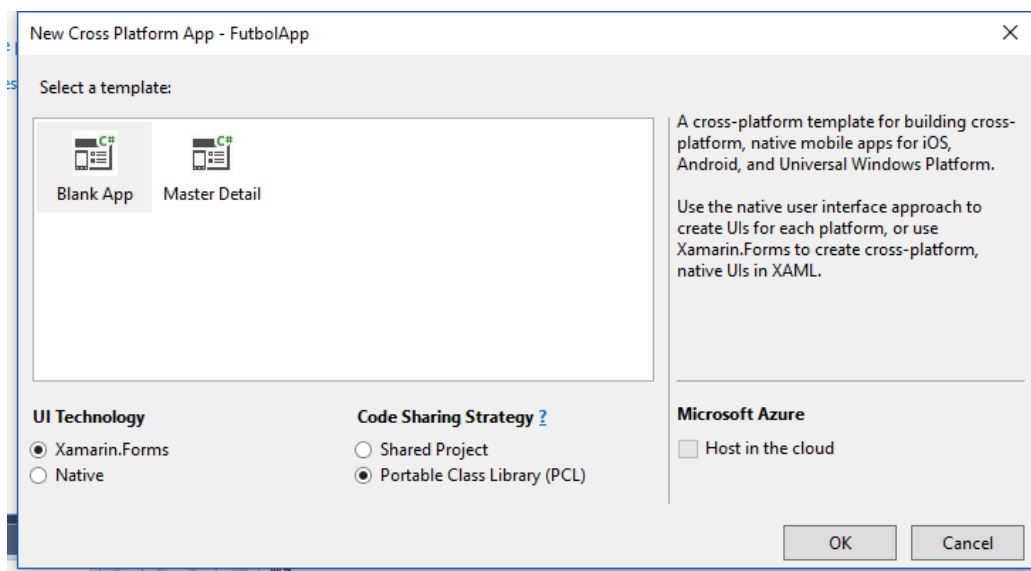
## Práctica: Almacenamiento local con SQLite en Xamarin – Autor: Luis Beltrán

En esta sesión vamos a crear una aplicación móvil multiplataforma con Xamarin que almacena información en el dispositivo mediante SQLite.

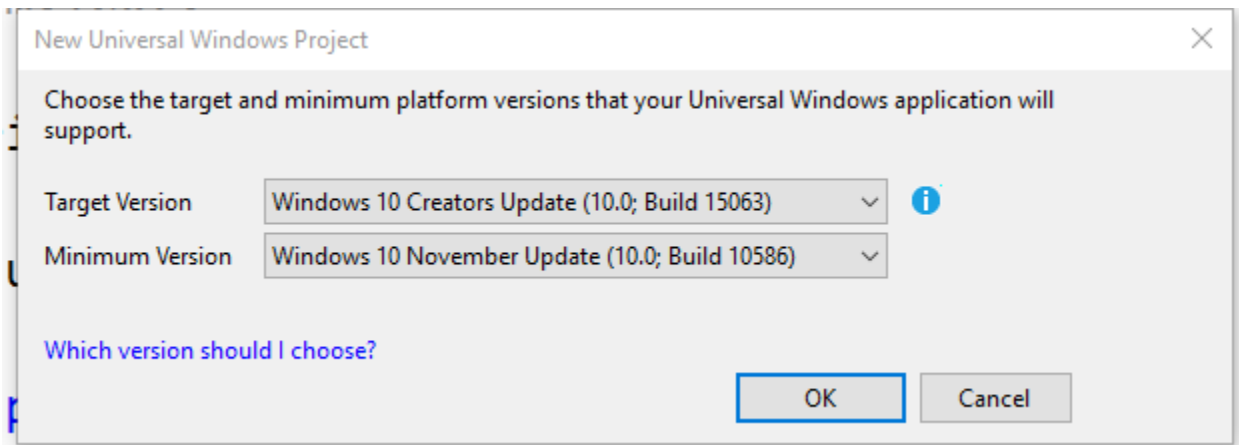
**Paso 1.** Crea un nuevo proyecto de la categoría **Cross-Platform** selecciona **Aplicación multiplataforma (Xamarin.Forms o nativa)** y coloca el nombre de proyecto **FutbolApp**. Además, la ruta del proyecto debe ser una ubicación corta para evitar problemas de ruta larga.



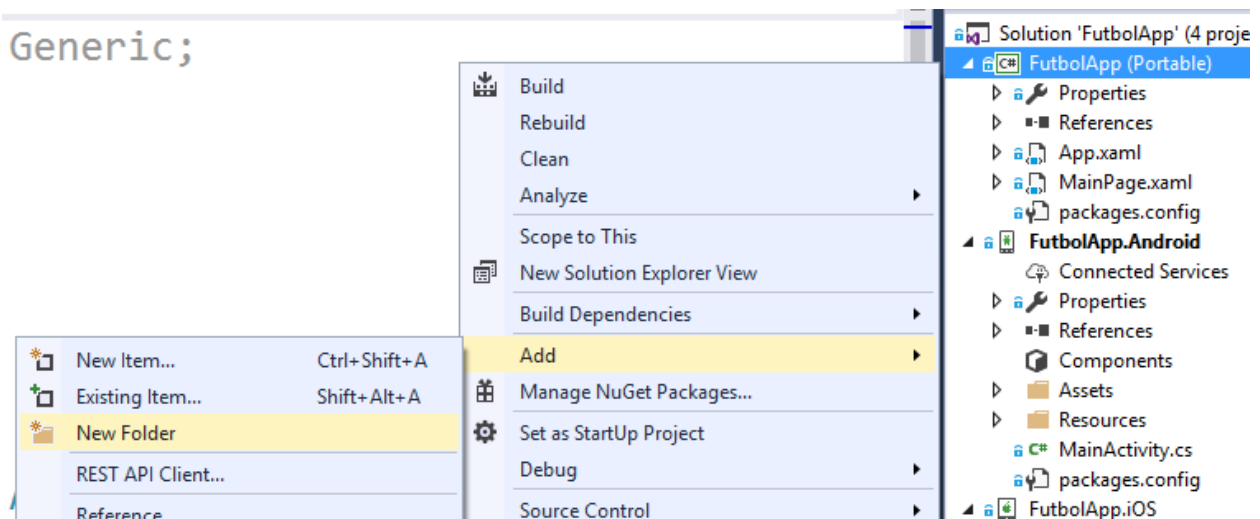
**Paso 2.** Selecciona la plantilla **Aplicación en blanco**, la tecnología de IU **Xamarin.Forms** y la estrategia de uso compartido de código **Biblioteca de clases portátil (PCL)**. Da clic en **OK**.



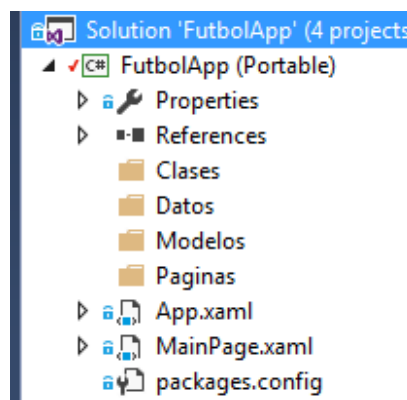
**Paso 3.** Si tienes instalado el SDK de Windows 10, aparecerá la ventana de selección del Target y Minimum Version. Selecciónalas a conveniencia, según la versión que tengas instalada.



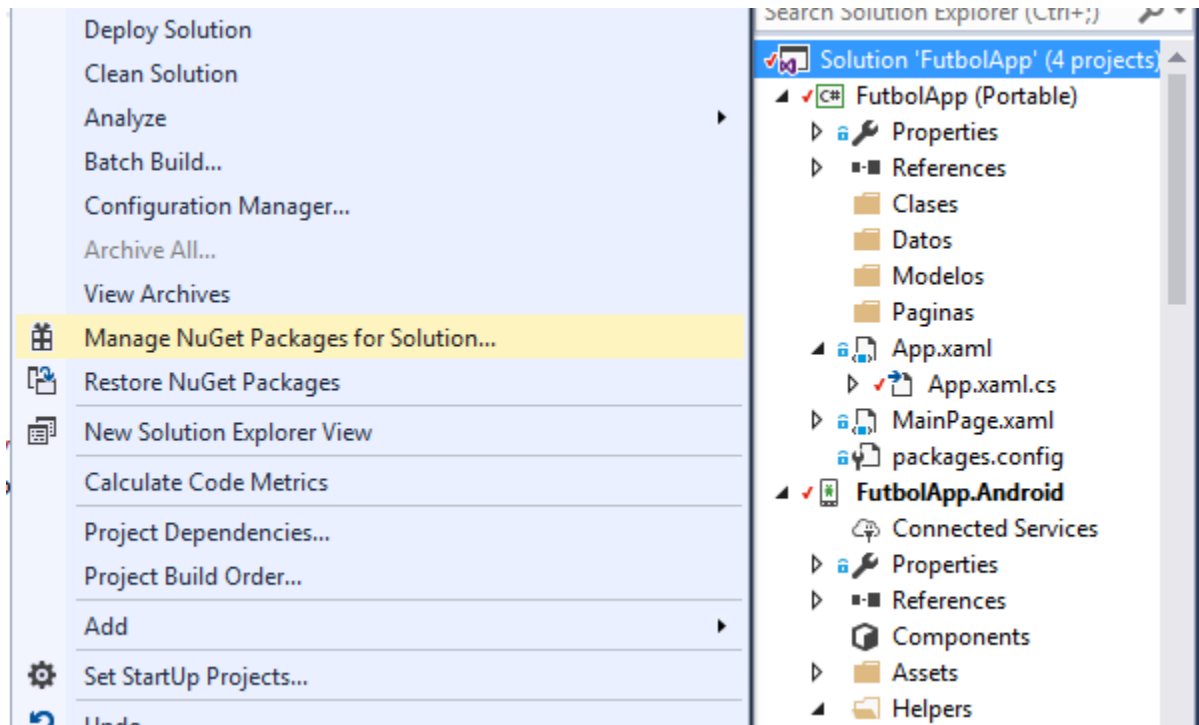
**Paso 4.** Da clic derecho en el **proyecto portable** y selecciona **Agregar → Nueva carpeta**.



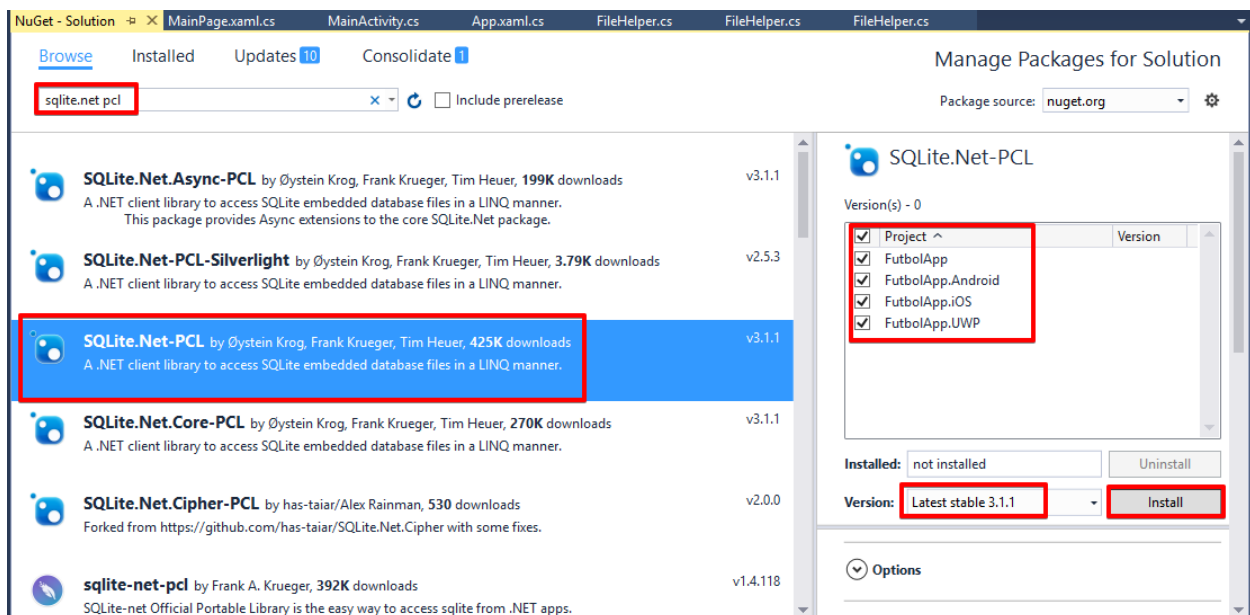
**Paso 5.** Agrega las carpetas Clases, Datos, Modelos y Paginas.



**Paso 6.** Da clic derecho en la solución y selecciona la opción **Administrar paquetes NuGet** para la solución...



**Paso 7.** Agrega los siguientes paquetes Nuget: **SQLite.Net-PCL**



## SQLiteNetExtensions.Async

The screenshot shows the NuGet Package Manager interface. The search bar contains 'sqlite.net.async extensions pcl'. The results list includes 'SQLite.Net.Async-PCL', 'SQLiteNetExtensions.Async', 'System.Threading.Tasks.Extensions', 'Extensions', and 'System.Reflection.Extensions'. The 'SQLiteNetExtensions.Async' package is highlighted. The right pane shows the package details for 'SQLiteNetExtensions.Async', including a table of projects and versions.

Project	Version
<input checked="" type="checkbox"/> FutbolApp	
<input checked="" type="checkbox"/> FutbolApp.Android	
<input checked="" type="checkbox"/> FutbolApp.iOS	
<input checked="" type="checkbox"/> FutbolApp.UWP	

Installed: not installed Uninstall

Version: Latest stable 1.3.0 Install

## SQLiteNetExtensions

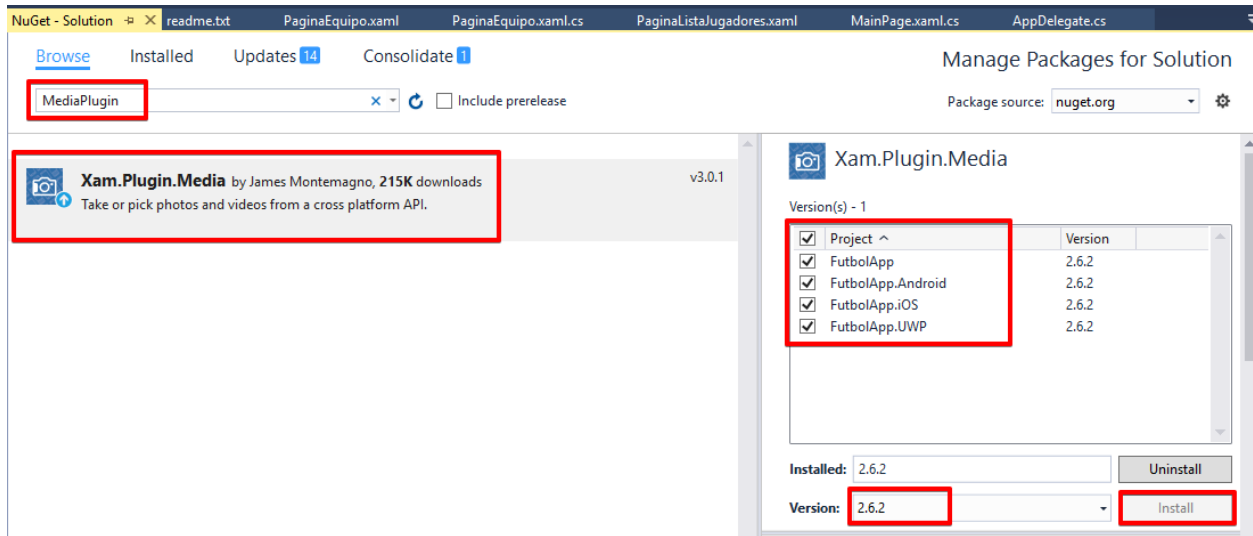
The screenshot shows the NuGet Package Manager interface. The search bar contains 'sqlite.net extensions-pcl'. The results list includes 'SQLiteNetExtensions', 'SQLiteNetExtensions.Async', 'ComponentTrak.SQLite.NET', 'sqlite-net-wp8', and 'SQLite.Net-PCL-Silverlight'. The 'SQLiteNetExtensions' package is highlighted. The right pane shows the package details for 'SQLiteNetExtensions', including a table of projects and versions.

Project	Version
<input checked="" type="checkbox"/> FutbolApp	1.3.0
<input checked="" type="checkbox"/> FutbolApp.Android	1.3.0
<input checked="" type="checkbox"/> FutbolApp.iOS	1.3.0
<input checked="" type="checkbox"/> FutbolApp.UWP	1.3.0

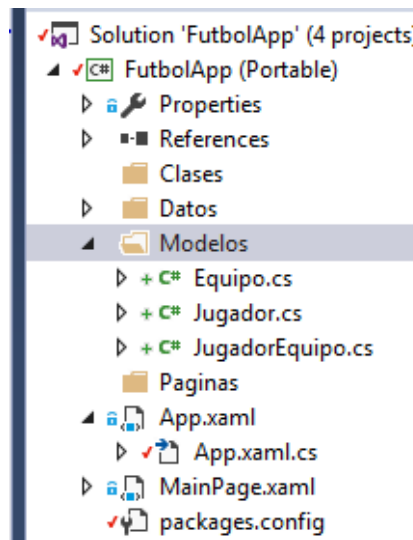
Installed: 1.3.0 Uninstall

Version: Latest stable 1.3.0 Install

## Xam.Plugin.Media



**Paso 8.** En la carpeta Modelos agrega tres clases: **Equipo**, **Jugador** y **JugadorEquipo**



**Paso 9.** El código de cada clase es el siguiente:

### **Jugador**

```
using SQLite.Net.Attributes;
using SQLiteNetExtensions.Attributes;
using System;
using System.Collections.Generic;

namespace FutbolApp.Modelos
{
    public class Jugador
    {
        [PrimaryKey, AutoIncrement]
        public int ID { get; set; }

        public string Nombre { get; set; }
        public string FechaNacimiento { get; set; } = DateTime.MinValue.ToString();
        public string Foto { get; set; }

        [ManyToMany(typeof(JugadorEquipo))]
        public List<Equipo> Equipos { get; set; }
    }
}
```

### **Equipo**

```
using SQLite.Net.Attributes;
using SQLiteNetExtensions.Attributes;
using System.Collections.Generic;

namespace FutbolApp.Modelos
{
    public class Equipo
    {
        [PrimaryKey, AutoIncrement]
        public int ID { get; set; }

        public string Nombre { get; set; }
        public string Escudo { get; set; }

        [ManyToMany(typeof(JugadorEquipo))]
        public List<Jugador> Jugadores { get; set; }
    }
}
```

## JugadorEquipo

```
using SQLite.Net.Attributes;
using SQLiteNetExtensions.Attributes;

namespace FutbolApp.Modelos
{
    public class JugadorEquipo
    {
        [PrimaryKey, AutoIncrement]
        public int ID { get; set; }

        [ForeignKey(typeof(Equipo))]
        public int IDEquipo { get; set; }

        [ForeignKey(typeof(Jugador))]
        public int IDJugador { get; set; }

        public int Numero { get; set; }
        public int Goles { get; set; }
    }
}
```

**Paso 10.** En la carpeta **Clases** agrega las clases **DetalleJugadorEquipo** y **ServicioImágenes**

## DetalleJugadorEquipo

```
namespace FutbolApp.Clases
{
    public class DetalleJugadorEquipo
    {
        public string FotoJugador { get; set; }
        public string NombreJugador { get; set; }
        public string EscudoEquipo { get; set; }
        public string NombreEquipo { get; set; }
        public int Numero { get; set; }
        public int Goles { get; set; }
    }
}
```

## ServicioImágenes

```
using Plugin.Media;
using Plugin.Media.Abstractions;
using System.Threading.Tasks;

namespace FutbolApp.Clases
{
    public static class ServicioImágenes
    {
        public static async Task<MediaFile> SeleccionarImagen()
        {
            await CrossMedia.Current.Initialize();
            var file = await CrossMedia.Current.PickPhotoAsync();
            return file;
        }
    }
}
```

```
}  
}
```

**Paso 11.** Ahora agrega una clase llamada **BaseDatos** en la carpeta **Datos**.

```
using System.Collections.Generic;  
using System.Linq;  
using SQLite.Net;  
using SQLite.Net.Interop;  
using FutbolApp.Modelos;  
using SQLiteNetExtensions.Extensions;  
using FutbolApp.Clases;  
  
namespace FutbolApp.Datos  
{  
    public class BaseDatos  
    {  
        static object locker = new object();  
        readonly ISQLitePlatform _plataforma;  
        string _rutaBD;  
  
        public SQLiteConnection Conexion { get; set; }  
  
        public BaseDatos(ISQLitePlatform plataforma, string rutaBD)  
        {  
            _plataforma = plataforma;  
            _rutaBD = rutaBD;  
        }  
  
        public void Conectar()  
        {  
            Conexion = new SQLiteConnection(_plataforma, _rutaBD,  
                SQLiteOpenFlags.ReadWrite | SQLiteOpenFlags.Create  
                | SQLiteOpenFlags.FullMutex, true);  
  
            Conexion.CreateTable<Jugador>();  
            Conexion.CreateTable<Equipo>();  
            Conexion.CreateTable<JugadorEquipo>();  
        }  
  
        #region Metodos de la tabla Jugador  
  
        public void AgregarJugador(Jugador jugador, Equipo equipo = null)  
        {  
            lock(locker)  
            {  
                Conexion.Insert(jugador);  
            }  
        }  
  
        public void ActualizarJugador(Jugador jugador, Equipo equipo = null)  
        {  
            lock (locker)  
            {  
                Conexion.Update(jugador);  
  
                if (equipo != null)
```



```

        {
            if (jugador.Equipos == null)
            {
                jugador.Equipos = new List<Equipo>();
            }

            if (jugador.Equipos.Where(x => x.ID == equipo.ID).Count() == 0)
                jugador.Equipos.Add(equipo);

            Conexion.UpdateWithChildren(jugador);
        }
    }

    public void EliminarJugador(Jugador jugador)
    {
        lock (locker)
        {
            Conexion.Delete(jugador);
        }
    }

    public List<Jugador> ObtenerJugadores()
    {
        lock (locker)
        {
            return Conexion.GetAllWithChildren<Jugador>().OrderBy(x =>
x.Nombre).ToList();
        }
    }

    public Jugador ObtenerJugador(int id)
    {
        lock (locker)
        {
            return Conexion.GetWithChildren<Jugador>(id);
        }
    }

    public List<Equipo> ObtenerEquiposJugador(int id)
    {
        lock (locker)
        {
            return Conexion.GetWithChildren<Jugador>(id).Equipos;
        }
    }

    #endregion

    #region Metodos de la tabla Equipo
    public void AgregarEquipo(Equipo equipo)
    {
        lock (locker)
        {
            Conexion.Insert(equipo);
        }
    }

```

```

public void ActualizarEquipo(Equipo equipo)
{
    lock (locker)
    {
        Conexion.Update(equipo);
    }
}

public void EliminarEquipo(Equipo equipo)
{
    lock (locker)
    {
        Conexion.Delete(equipo);
    }
}

public List<Equipo> ObtenerEquipos()
{
    lock (locker)
    {
        return Conexion.GetAllWithChildren<Equipo>().OrderBy(x =>
x.Nombre).ToList();
    }
}

public Equipo ObtenerEquipo(int id)
{
    lock (locker)
    {
        return Conexion.GetWithChildren<Equipo>(id);
    }
}

public List<Jugador> ObtenerJugadoresEquipo(int id)
{
    lock (locker)
    {
        return Conexion.GetWithChildren<Equipo>(id).Jugadores;
    }
}
#endregion

#region Metodos de la tabla JugadorEquipo
public JugadorEquipo ObtenerJugadorEquipo(int idEquipo, int idJugador)
{
    lock(locker)
    {
        var tabla = Conexion.Table<JugadorEquipo>().ToList();
        var num = tabla.Where(x => x.IDEquipo == idEquipo && x.IDJugador ==
idJugador).Count();

        if (num > 0)
            return tabla.Where(x => x.IDEquipo == idEquipo && x.IDJugador ==
idJugador).First();
        else
            return new JugadorEquipo() { IDEquipo = idEquipo, IDJugador =
idJugador, Goles = 0, Numero = 0 };
    }
}

```

```

    }

    public DetalleJugadorEquipo ObtenerDetalleJugadorEquipo(Equipo equipo, Jugador
jugador)
    {
        lock (locker)
        {
            var jugadorEquipo = ObtenerJugadorEquipo(equipo.ID, jugador.ID);

            var detalle = new DetalleJugadorEquipo()
            {
                NombreJugador = jugador.Nombre,
                FotoJugador = jugador.Foto,
                NombreEquipo = equipo.Nombre,
                EscudoEquipo = equipo.Escudo,
                Goles = (jugadorEquipo != null) ? jugadorEquipo.Goles : 0,
                Numero = (jugadorEquipo != null) ? jugadorEquipo.Numero : 0
            };

            return detalle;
        }
    }

    public void ActualizarJugadorEquipo(JugadorEquipo jugadorEquipo)
    {
        lock (locker)
        {
            Conexion.Update(jugadorEquipo);
        }
    }
    #endregion
}
}

```

**Paso 12.** Modifica App.xaml.cs:

```

using FutbolApp.Datos;
using SQLite.Net.Interop;
using Xamarin.Forms;

namespace FutbolApp
{
    public partial class App : Application
    {
        public static BaseDatos BaseDatos { get; set; }

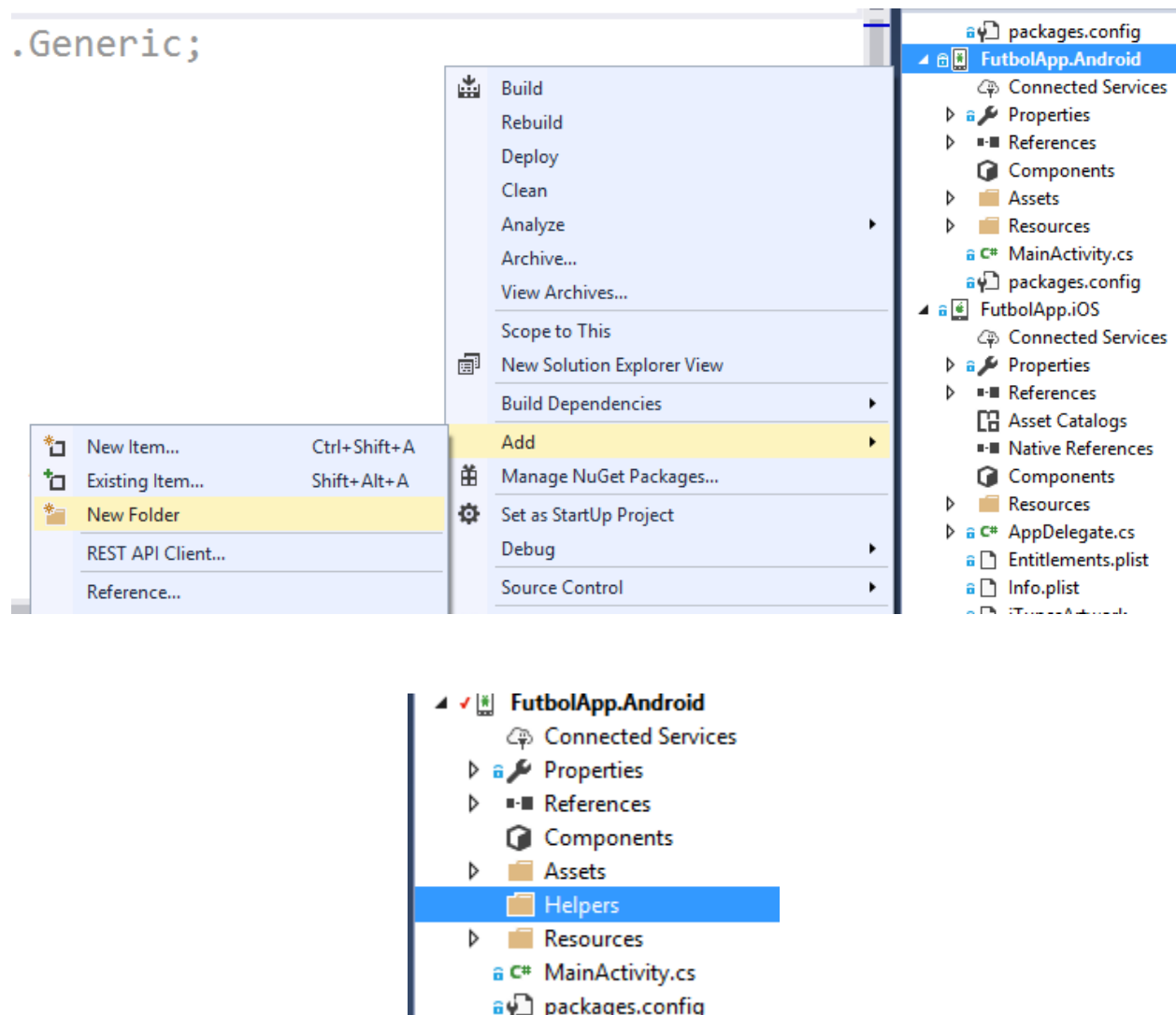
        public App(string rutaBD, ISQLitePlatform plataforma)
        {
            BaseDatos = new BaseDatos(plataforma, rutaBD);
            BaseDatos.Conectar();

            InitializeComponent();

            MainPage = new NavigationPage(new FutbolApp.Paginas.PaginaMenu());
        }
    }
}

```

**Paso 13.** En el proyecto de Android crea una carpeta llamada Helpers



**Paso 14.** Agrega una clase llamada **FileHelper**, con el siguiente código

```
using System;
using System.IO;

namespace FutbolApp.Droid.Helpers
{
    public class FileHelper
    {
        public static string ObtenerRutaLocal(string archivo)
        {
            string ruta = Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            return Path.Combine(ruta, archivo);
        }
    }
}
```

**Paso 15.** En el proyecto de **iOS** crea la carpeta **Helpers** y la clase **FileHelper**, con el siguiente código:

```
using System;
using System.IO;

namespace FutbolApp.iOS.Helpers
{
    public class FileHelper
    {
        public static string ObtenerRutaLocal(string archivo)
        {
            string folder =
Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            string libreria = Path.Combine(folder, "..", "Library");

            if (!Directory.Exists(libreria))
                Directory.CreateDirectory(libreria);

            return Path.Combine(libreria, archivo);
        }
    }
}
```

**Paso 16.** En el proyecto de **UWP** crea la carpeta **Helpers** y la clase **FileHelper**, con el siguiente código:

```
using System.IO;

namespace FutbolApp.UWP.Helpers
{
    public class FileHelper
    {
        public static string ObtenerRutaLocal(string archivo)
        {
            string ruta = Windows.Storage.ApplicationData.Current.LocalFolder.Path;
            return Path.Combine(ruta, archivo);
        }
    }
}
```

**Paso 17.** Dentro del proyecto de **Android**, modifica **MainActivity**:

```
[Activity(Label = "FutbolApp", Icon = "@drawable/icon", Theme = "@style/MainTheme",
MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize |
ConfigChanges.Orientation)]
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);

        string rutaBD = Helpers.FileHelper.ObtenerRutaLocal("futbol-v3.db3");
        LoadApplication(new App(rutaBD, new
SQLite.Net.Platform.XamarinAndroid.SQLitePlatformAndroid()));
    }

    public override void OnRequestPermissionsResult(int requestCode, string[]
permissions, Permission[] grantResults)
    {
        Plugin.Permissions.PermissionsImplementation.Current.OnRequestPermissionsResult(requestCode, permissions, grantResults);
    }
}
```

**Paso 18.** Dentro del proyecto de **iOS**, modifica **AppDelegate**:

```
public override bool FinishedLaunching(UIApplication app, NSDictionary options)
{
    global::Xamarin.Forms.Forms.Init();

    string rutaBD = Helpers.FileHelper.ObtenerRutaLocal("futbol.db3");
    LoadApplication(new App(rutaBD, new
SQLite.Net.Platform.XamarinIOS.SQLitePlatformIOS()));

    return base.FinishedLaunching(app, options);
}
```

**Paso 19.** Dentro del proyecto de **UWP**, modifica **MainPage**:

```
public MainPage()
{
    this.InitializeComponent();

    string rutaBD = Helpers.FileHelper.ObtenerRutaLocal("futbol.db3");
    LoadApplication(new FutbolApp.App(rutaBD, new
SQLite.Net.Platform.WinRT.SQLitePlatformWinRT()));
}
```

**Paso 20.** En la carpeta **Paginas** agrega las siguientes páginas de tipo **ContentPage**:

**a) PaginaMenu**

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="FutbolApp.Paginas.PaginaMenu" Title="Menu">
    <ContentPage.ToolbarItems>
        <ToolbarItem x:Name="Jugadores" Text="Jugadores" Order="Primary" Priority="0"
Clicked="Jugadores_Clicked"/>
        <ToolbarItem x:Name="Equipos" Text="Equipos" Order="Primary" Priority="1"
Clicked="Equipos_Clicked"/>
    </ContentPage.ToolbarItems>
</ContentPage>
```

```
using System;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace FutbolApp.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaMenu : ContentPage
    {
        public PaginaMenu()
        {
            InitializeComponent();
        }

        private async void Jugadores_Clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new PaginalistaJugadores());
        }

        private async void Equipos_Clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new PaginalistaEquipos());
        }
    }
}
```

**b) PaginalistaEquipos**

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="FutbolApp.Paginas.PaginalistaEquipos" Title="Equipos">
    <StackLayout>
        <ListView x:Name="lsvEquipos" ItemSelected="lsvEquipos_ItemSelected">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
```

```

                <StackLayout BackgroundColor="White" Spacing="5"
Orientation="Horizontal" Padding="5">
                    <Image Source="{Binding Escudo}" Aspect="AspectFill"
WidthRequest="70" HeightRequest="70"/>
                    <Label Text="{Binding Nombre}" LineBreakMode="WordWrap"
VerticalOptions="Center" FontSize="15" TextColor="#030303" HorizontalOptions="Start"
Margin="12,5,12,1" HorizontalTextAlignment="Start"/>
                </StackLayout>
            </ViewCell>
        </DataTemplate>
    </ListView.ItemTemplate>
</ListView>
</StackLayout>

    <ContentPage.ToolbarItems>
        <ToolbarItem x:Name="Agregar" Text="Agregar" Order="Primary" Priority="0"
Clicked="Agregar_Clicked"/>
    </ContentPage.ToolbarItems>
</ContentPage>

```

```

using System;
using FutbolApp.Modelos;

```

```

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace FutbolApp.Paginas

```

```

{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaListaEquipos : ContentPage
    {
        public PaginaListaEquipos()
        {
            InitializeComponent();
        }

        protected override void OnAppearing()
        {
            base.OnAppearing();

            lsvEquipos.ItemsSource = App.BaseDatos.ObtenerEquipos();
        }

        private async void lsvEquipos_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
        {
            try
            {
                Equipo equipo = (Equipo)e.SelectedItem;
                await Navigation.PushAsync(new PaginaEquipo(equipo));
            }
            catch (Exception ex)
            {
            }
        }
    }
}

```



```

        private async void Agregar_Clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new PaginaEquipo(new Equipo()));
        }
    }
}

```

### c) PaginaListaJugadores

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="FutbolApp.Paginas.PaginaListaJugadores" Title="Jugadores">
    <StackLayout>
        <ListView x:Name="lsvJugadores" ItemSelected="lsvJugadores_ItemSelected">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout BackgroundColor="White" Spacing="5"
Orientation="Horizontal" Padding="5">
                            <Image Source="{Binding Foto}" Aspect="AspectFill"
WidthRequest="70" HeightRequest="70"/>
                            <Label Text="{Binding Nombre}" LineBreakMode="WordWrap"
VerticalOptions="Center" FontSize="15" TextColor="#030303" HorizontalOptions="Start"
Margin="12,5,12,1" HorizontalTextAlignment="Start"/>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>

    <ContentPage.ToolbarItems>
        <ToolbarItem x:Name="Agregar" Text="Agregar" Order="Primary" Priority="0"
Clicked="Agregar_Clicked"/>
    </ContentPage.ToolbarItems>
</ContentPage>

```

```

using System;
using FutbolApp.Modelos;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace FutbolApp.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaListaJugadores : ContentPage
    {
        Equipo Equipo;

        public PaginaListaJugadores(Equipo equipo = null)
        {
            InitializeComponent();

```

```

        Equipo = equipo;

        if (Equipo != null)
        {
            if (this.ToolbarItems.Count > 0)
                this.ToolbarItems.RemoveAt(0);
        }
    }

    protected override void OnAppearing()
    {
        base.OnAppearing();

        lsvJugadores.ItemsSource = App.BaseDatos.ObtenerJugadores();
    }

    private async void lsvJugadores_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
    {
        try
        {
            Jugador jugador = (Jugador)e.SelectedItem;

            if (Equipo == null)
            {
                await Navigation.PushAsync(new PaginaJugador(jugador));
            }
            else
            {
                await Navigation.PushAsync(new PaginaJugadorEquipo(jugador, Equipo,
true));
            }
        }
        catch (Exception ex)
        {
        }
    }

    private async void Agregar_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new PaginaJugador(new Jugador()));
    }
}

```

#### d) PaginaEquipo

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="FutbolApp.Paginas.PaginaEquipo" Title="Equipo">
    <ScrollView>
        <StackLayout Padding="10" Spacing="10">
            <Label Text="Nombre:" FontSize="Medium" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"/>

```

```

        <Entry Text="{Binding Nombre}" Placeholder="Nombre" WidthRequest="200"
FontSize="20" TextColor="Black" BackgroundColor="White" HorizontalOptions="Start"
Margin="10" HorizontalTextAlignment="Start" FontAttributes="Bold"/>
        <Button x:Name="btnEscudo" Text="Elegir escudo" BackgroundColor="LightBlue"
Margin="10" TextColor="Black" WidthRequest="200" HorizontalOptions="Center"
Clicked="btnEscudo_Clicked"/>
        <Image x:Name="imgEscudo" Source="{Binding Escudo}"
VerticalOptions="StartAndExpand" HorizontalOptions="Center" Aspect="AspectFill"
HeightRequest="150" WidthRequest="150" Margin="10"/>

        <ListView x:Name="lsvJugadores" ItemSelected="lsvJugadores_ItemSelected"
ItemsSource="{Binding Jugadores}">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout BackgroundColor="White" Spacing="5"
Orientation="Horizontal" Padding="5">
                            <Image Source="{Binding Foto}" Aspect="AspectFill"
WidthRequest="70" HeightRequest="70"/>
                            <Label Text="{Binding Nombre}" LineBreakMode="WordWrap"
VerticalOptions="Center" FontSize="15" TextColor="#030303" HorizontalOptions="Start"
Margin="12,5,12,1" HorizontalTextAlignment="Start"/>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>

    </StackLayout>
</ScrollView>

    <ContentPage.ToolbarItems>
        <ToolbarItem x:Name="Guardar" Text="Guardar" Order="Primary" Priority="0"
Clicked="Guardar_Clicked"/>
        <ToolbarItem x:Name="Eliminar" Text="Eliminar" Order="Primary" Priority="1"
Clicked="Eliminar_Clicked"/>
        <ToolbarItem x:Name="Agregar" Text="Agregar jugador" Order="Primary" Priority="2"
Clicked="Agregar_Clicked"/>
    </ContentPage.ToolbarItems>
</ContentPage>

```

```

using FutbolApp.Modelos;
using FutbolApp.Clases;
using System;

```

```

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace FutbolApp.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaEquipo : ContentPage
    {
        Equipo Equipo;

        public PaginaEquipo(Equipo equipo)
    }
}

```

```

{
    InitializeComponent();

    Equipo = equipo;
    this.BindingContext = equipo;
}

private async void btnEscudo_Clicked(object sender, EventArgs e)
{
    var escudo = await ServicioImagenes.SeleccionarImagen();
    Equipo.Escudo = escudo.Path;
    imgEscudo.Source = ImageSource.FromFile(escudo.Path);
}

private void Guardar_Clicked(object sender, EventArgs e)
{
    if (Equipo.ID > 0)
        App.BaseDatos.ActualizarEquipo(Equipo);
    else
        App.BaseDatos.AgregarEquipo(Equipo);
}

private async void Eliminar_Clicked(object sender, EventArgs e)
{
    if (await DisplayAlert("Eliminar", "¿Deseas eliminar el equipo?", "Si",
"No"))
    {
        App.BaseDatos.EliminarEquipo(Equipo);
    }
}

private async void lsvJugadores_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
{
    Jugador jugador = (Jugador)e.SelectedItem;
    await Navigation.PushAsync(new PaginaJugadorEquipo(jugador, Equipo, false));
}

private async void Agregar_Clicked(object sender, EventArgs e)
{
    await Navigation.PushAsync(new PaginalistaJugadores(Equipo));
}
}
}

```

#### e) PaginaJugador

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="FutbolApp.Paginas.PaginaJugador" Title="Jugador">
    <ScrollView>
        <StackLayout Padding="10" Spacing="10">
            <Label Text="Nombre:" FontSize="Medium" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"/>

```

```

        <Entry Text="{Binding Nombre}" Placeholder="Nombre" WidthRequest="200"
FontSize="20" TextColor="Black" BackgroundColor="White" HorizontalOptions="Start"
Margin="10" HorizontalTextAlignment="Start" FontAttributes="Bold"/>
        <Button x:Name="btnFoto" Text="Elegir foto" BackgroundColor="LightBlue"
Margin="10" TextColor="Black" WidthRequest="200" HorizontalOptions="Center"
Clicked="btnFoto_Clicked"/>
        <Image x:Name="imgFoto" Source="{Binding Foto}"
VerticalOptions="StartAndExpand" HorizontalOptions="Center" Aspect="AspectFill"
HeightRequest="150" WidthRequest="150" Margin="10"/>
        <Label Text="Fecha de Nacimiento:" FontSize="Medium" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"/>
        <DatePicker x:Name="dtpFecha" Date="{Binding FechaNacimiento}"
BackgroundColor="White" TextColor="Black"/>

        <ListView x:Name="lsvEquipos" ItemSelected="lsvEquipos_ItemSelected">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout BackgroundColor="White" Spacing="5"
Orientation="Horizontal" Padding="5">
                            <Image Source="{Binding Escudo}" Aspect="AspectFill"
WidthRequest="70" HeightRequest="70"/>
                            <Label Text="{Binding Nombre}" LineBreakMode="WordWrap"
VerticalOptions="Center" FontSize="15" TextColor="#030303" HorizontalOptions="Start"
Margin="12,5,12,1" HorizontalTextAlignment="Start"/>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>

    </StackLayout>
</ScrollView>

    <ContentPage.ToolbarItems>
        <ToolbarItem x:Name="Guardar" Text="Guardar" Order="Primary" Priority="0"
Clicked="Guardar_Clicked"/>
        <ToolbarItem x:Name="Eliminar" Text="Eliminar" Order="Primary" Priority="1"
Clicked="Eliminar_Clicked"/>
    </ContentPage.ToolbarItems>
</ContentPage>

```

```

using FutbolApp.Modelos;
using FutbolApp.Clases;
using System;

```

```

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace FutbolApp.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaJugador : ContentPage
    {
        Jugador Jugador;
    }
}

```

```

public PaginaJugador(Jugador jugador)
{
    InitializeComponent();

    Jugador = jugador;
    this.BindingContext = jugador;
}

private async void btnFoto_Clicked(object sender, EventArgs e)
{
    var foto = await ServicioImagenes.SeleccionarImagen();
    Jugador.Foto = foto.Path;
    imgFoto.Source = ImageSource.FromFile(foto.Path);
}

private void Guardar_Clicked(object sender, EventArgs e)
{
    if (Jugador.ID > 0)
        App.BaseDatos.ActualizarJugador(Jugador);
    else
        App.BaseDatos.AgregarJugador(Jugador);
}

private async void Eliminar_Clicked(object sender, EventArgs e)
{
    if (await DisplayAlert("Eliminar", "¿Deseas eliminar el jugador?", "Si",
"No"))
    {
        App.BaseDatos.EliminarJugador(Jugador);
    }
}

private async void lsvEquipos_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
{
    Equipo equipo = (Equipo)e.SelectedItem;
    await Navigation.PushAsync(new PaginaJugadorEquipo(Jugador, equipo, false));
}
}

```

#### f) PaginaJugadorEquipo

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="FutbolApp.Paginas.PaginaJugadorEquipo" Title="Detalle">
    <ScrollView>
        <StackLayout Padding="10" Spacing="10">
            <StackLayout Orientation="Horizontal" Margin="10">
                <Image x:Name="imgFoto" Source="{Binding FotoJugador}"
VerticalOptions="StartAndExpand" HorizontalOptions="Center" Aspect="AspectFill"
HeightRequest="100" WidthRequest="100" Margin="10"/>
                <Label Text="{Binding NombreJugador}" FontSize="20" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"
FontAttributes="Bold"/>
            </StackLayout>
        </StackLayout>
    </ScrollView>

```

```

        </StackLayout>
        <StackLayout Orientation="Horizontal" Margin="10">
            <Image x:Name="imgEscudo" Source="{Binding EscudoEquipo}"
VerticalOptions="StartAndExpand" HorizontalOptions="Center" Aspect="AspectFill"
HeightRequest="100" WidthRequest="100" Margin="10"/>
            <Label Text="{Binding NombreEquipo}" FontSize="20" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"
FontAttributes="Bold"/>
        </StackLayout>

        <Label Text="Numero" FontSize="Medium" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"/>
        <Entry Text="{Binding Numero}" Keyboard="Numeric" WidthRequest="100"
FontSize="20" TextColor="Black" BackgroundColor="White" HorizontalOptions="Start"
Margin="10" HorizontalTextAlignment="Start" FontAttributes="Bold"/>

        <Label Text="Goles" FontSize="Medium" TextColor="Black"
HorizontalOptions="Start" Margin="10" HorizontalTextAlignment="Start"/>
        <Entry Text="{Binding Goles}" Keyboard="Numeric" WidthRequest="100"
FontSize="20" TextColor="Black" BackgroundColor="White" HorizontalOptions="Start"
Margin="10" HorizontalTextAlignment="Start" FontAttributes="Bold"/>
    </StackLayout>
</ScrollView>
<ContentPage.ToolbarItems>
    <ToolbarItem x:Name="Guardar" Text="Guardar" Order="Primary" Priority="0"
Clicked="Guardar_Clicked"/>
</ContentPage.ToolbarItems>
</ContentPage>

```

```

using FutbolApp.Clases;
using FutbolApp.Modelos;
using System;

```

```

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace FutbolApp.Paginas

```

```

{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PaginaJugadorEquipo : ContentPage
    {
        DetalleJugadorEquipo Detalle;
        JugadorEquipo JugadorEquipo;
        Jugador Jugador;
        Equipo Equipo;
        bool Nuevo;

        public PaginaJugadorEquipo(Jugador jugador, Equipo equipo, bool nuevo)
        {
            InitializeComponent();

            Jugador = jugador;
            Equipo = equipo;
            Detalle = App.BaseDatos.ObtenerDetalleJugadorEquipo(equipo, jugador);
            JugadorEquipo = App.BaseDatos.ObtenerJugadorEquipo(equipo.ID, jugador.ID);
            Nuevo = nuevo;
        }
    }
}

```

```

        this.BindingContext = Detalle;
    }

    private void Guardar_Clicked(object sender, EventArgs e)
    {
        JugadorEquipo.Goles = Detalle.Goles;
        JugadorEquipo.Numero = Detalle.Numero;

        App.BaseDatos.ActualizarJugador(Jugador, Equipo);
        App.BaseDatos.ActualizarJugadorEquipo(JugadorEquipo);
    }
}

```

Finalmente, compila y ejecuta la aplicación. Observa el resultado