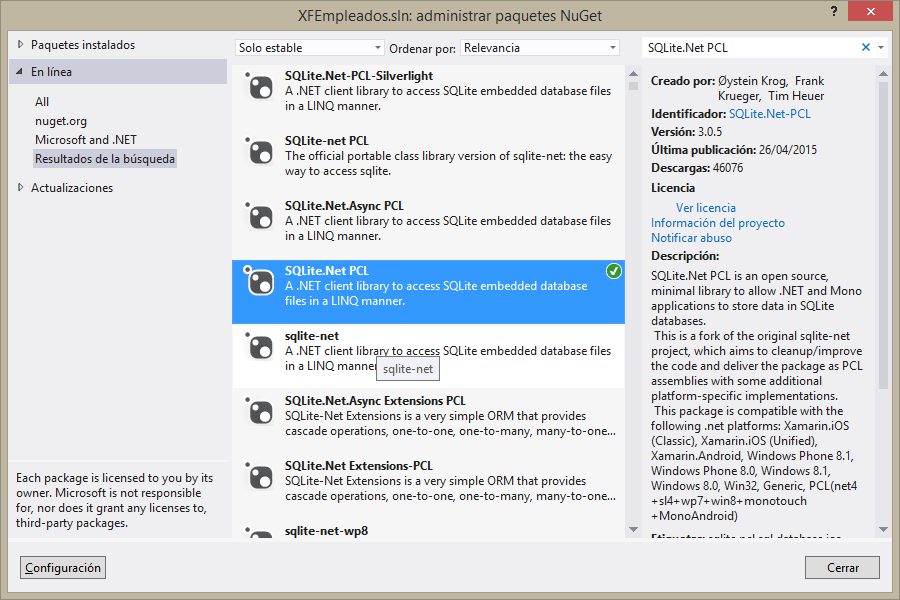
Xamarin.Forms

Guía para construir ejemplo de SQLite

Juan Carlos Zuluaga – <http://zulu-software.com>

1. Cree el proyecto portable de Xamarin.Forms, en el ejemplo lo llamaremos PreSQLite
2. Agregue a los 4 proyectos el paquete SQLite.Net PCL (Compartido, Droid, iOS y WinPhone)



1. Es MUY importante que actualice los paquetes en todos los proyectos:
2. En el proyecto compartido agregue la clase : **Empleado.cs** con:

using System;  
using SQLite.Net.Attributes;  
  
namespace PreSQLite  
{  
    public class Empleado  
    {  
        [PrimaryKey, AutoIncrement]  
        public int IDEmpleado { get; set; }  
        public string Nombres { get; set; }  
        public string Apellidos { get; set; }  
        public DateTime FechaContrato { get; set; }  
        public decimal Salario { get; set; }  
        public bool Activo { get; set; }  
  
        public string NombreCompleto {   
            get {   
                return string.Format("{0} {1}", this.Nombres, this.Apellidos);   
            }    
        }  
  
        public string FechaContratoEdited {   
            get {   
                return string.Format ("{0:yy-MM-dd}", FechaContrato);  
            }    
        }  
  
        public string SalarioEdited {   
            get {   
                return string.Format ("{0:C2}", Salario);  
            }    
        }  
  
        public override string ToString()  
        {  
            return string.Format("{0} {1} {2} {3} {4}", IDEmpleado, NombreCompleto,  
                FechaContratoEdited, SalarioEdited, Activo);  
        }  
    }

}

1. En el proyecto compartido agregue la interfaz **IConfig.cs** con el siguiente código:

using System;  
using SQLite.Net.Interop;  
  
namespace PreSQLite  
{  
    public interface IConfig  
    {  
        string DirectorioDB { get; }  
        ISQLitePlatform Plataforma { get; }  
    }  
}

1. En el proyecto iOS implemente la interfaz **IConfig** con el siguiente código:

using Xamarin.Forms;  
using SQLite.Net.Interop;  
using System;  
  
[assembly: Dependency(typeof(PreSQLite.iOS.Config))]  
  
namespace PreSQLite.iOS  
{  
    public class Config : IConfig  
    {  
        private string directorioDB;  
        private ISQLitePlatform plataforma;  
  
        public string DirectorioDB {  
            get {  
                if (string.IsNullOrEmpty(directorioDB)) {  
                    var directorio = System.Environment.GetFolderPath(Environment.SpecialFolder.Personal);  
                    directorioDB = System.IO.Path.Combine(directorio, "..", "Library");  
                }  
  
                return directorioDB;  
            }  
        }  
  
        public ISQLitePlatform Plataforma {  
            get {  
                if (plataforma == null) {  
                    plataforma = new SQLite.Net.Platform.XamarinIOS.SQLitePlatformIOS();  
                }  
  
                return plataforma;  
            }  
        }  
    }  
}

1. En el proyecto Droid implemente la interfaz **IConfig** con el siguiente código:

using System;  
using SQLite.Net.Interop;  
using Xamarin.Forms;  
  
[assembly: Dependency(typeof(PreSQLite.Droid.Config))]  
  
namespace PreSQLite.Droid  
{  
    public class Config : IConfig  
    {  
        private string directorioDB;  
        private ISQLitePlatform plataforma;  
  
        public string DirectorioDB  
        {  
            get {  
                if (string.IsNullOrEmpty(directorioDB)) {  
                    directorioDB = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);  
                }  
  
                return directorioDB;  
            }  
        }  
  
        public ISQLitePlatform Plataforma {  
            get {  
                if (plataforma == null) {  
                    plataforma = new SQLite.Net.Platform.XamarinAndroid.SQLitePlatformAndroid();  
                }  
  
                return plataforma;  
            }  
        }  
    }  
}

1. En el proyecto Windows Phone implemente la interfaz **IConfig** con el siguiente código:

using SQLite.Net.Interop;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Windows.Storage;

using Xamarin.Forms;

[assembly: Dependency(typeof(XFEmpleados.WinPhone.Config))]

namespace XFEmpleados.WinPhone

{

class Config : IConfig

{

private string directorioDB;

private ISQLitePlatform plataforma;

public string DirectorioDB

{

get

{

if (string.IsNullOrEmpty(directorioDB))

{

directorioDB = ApplicationData.Current.LocalFolder.Path;

}

return directorioDB;

}

}

public ISQLitePlatform Plataforma

{

get

{

if (plataforma == null)

{

plataforma = new SQLite.Net.Platform.WindowsPhone8.SQLitePlatformWP8();

}

return plataforma;

}

}

}

}

1. En el proyecto compartido agregue la clase **DataAccess.cs** con el siguiente código:

using System;  
using System.Collections.Generic;  
using System.Linq;  
using SQLite.Net;  
using Xamarin.Forms;  
  
namespace PreSQLite  
{  
    public class DataAccess : IDisposable  
    {  
        private SQLiteConnection connection;  
  
        public DataAccess()  
        {  
            var config = DependencyService.Get<IConfig>();  
            connection = new SQLiteConnection(config.Plataforma,   
                System.IO.Path.Combine(config.DirectorioDB, "Empleados.db3"));  
            connection.CreateTable<Empleado>();  
        }  
  
        public void InsertEmpleado(Empleado empleado)  
        {  
            connection.Insert(empleado);  
        }  
  
        public void UpdateEmpleado(Empleado empleado)  
        {  
            connection.Update(empleado);  
        }  
  
        public void DeleteEmpleado(Empleado empleado)  
        {  
            connection.Delete(empleado);  
        }  
  
        public Empleado GetEmpleado(int IDEmpleado)  
        {  
            return connection.Table<Empleado>().FirstOrDefault(c => c.IDEmpleado == IDEmpleado);  
        }  
  
        public List<Empleado> GetEmpleados()  
        {  
            return connection.Table<Empleado>().OrderBy(c => c.Apellidos).ToList();  
        }  
  
        public void Dispose()  
        {  
            connection.Dispose();  
        }  
    }  
}

1. Agregue la página: **HomePage.xaml** con:

<?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="PreSQLite.HomePage">  
    <ContentPage.Content>  
        <ScrollView>  
            <StackLayout Padding="10" Spacing="10">  
                <Label Text="Empleados"   
                       Font="24"   
                       HorizontalOptions="Center"   
                       VerticalOptions="Center"   
                       FontAttributes="Bold" />  
                <Entry x:Name="nombresEntry"   
                       Placeholder="Nombres" />  
                <Entry x:Name="apellidosEntry"   
                       Placeholder="Apellidos" />  
                <Entry x:Name="salarioEntry"   
                       Placeholder="Salario"   
                       Keyboard="Numeric" />  
                <StackLayout Orientation="Horizontal">  
                    <Label Text="Fecha Contratación: "   
                           VerticalOptions="Center" />  
                    <DatePicker x:Name="fechaContratoDatePicker" />  
                    <Label Text="Activo: "   
                           VerticalOptions="Center" />  
                    <Switch x:Name="activoSwitch"   
                            VerticalOptions="Center"   
                            IsToggled="True" />  
                </StackLayout>  
                <Button x:Name="agregarButton"   
                        Text="Agregar"  
                        BackgroundColor="Navy"  
                        TextColor="White" />  
                <ListView x:Name="listaListView" />  
            </StackLayout>  
        </ScrollView>  
    </ContentPage.Content>  
</ContentPage>

1. Cambie el inicio de la aplicación **App.cs** por:

        public App ()  
        {  
            MainPage = new NavigationPage(new HomePage());  
        }

Pruebe lo que lleve hasta el momento

|  |  |
| --- | --- |
|  |  |

1. Modifique el siguiente código a la página: **HomePage.xaml.cs**

using System;  
using System.Collections.Generic;  
  
using Xamarin.Forms;  
  
namespace PreSQLite  
{  
    public partial class HomePage : ContentPage  
    {  
        public HomePage ()  
        {  
            InitializeComponent();  
  
            listaListView.RowHeight = 70;  
  
            agregarButton.Clicked += agregarButton\_Clicked;  
  
            Padding = Device.OnPlatform (  
                new Thickness (10, 20, 10, 10),  
                new Thickness (10, 10, 10, 10),  
                new Thickness (10, 10, 10, 10));  
        }  
  
        protected override void OnAppearing()  
        {              
            base.OnAppearing();  
            using (var datos = new DataAccess()) {  
                listaListView.ItemsSource = datos.GetEmpleados();  
            }  
        }  
  
        private async void agregarButton\_Clicked(object sender, EventArgs e)  
        {  
            if (string.IsNullOrEmpty(nombresEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar nombres", "Aceptar");  
                nombresEntry.Focus();  
                return;  
            }  
  
            if (string.IsNullOrEmpty(apellidosEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar apellidos", "Aceptar");  
                apellidosEntry.Focus();  
                return;  
            }  
  
            if (string.IsNullOrEmpty(salarioEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar salario", "Aceptar");  
                salarioEntry.Focus();  
                return;  
            }  
  
            var empleado = new Empleado {  
                Nombres = nombresEntry.Text,  
                Apellidos = apellidosEntry.Text,  
                FechaContrato = fechaContratoDatePicker.Date,  
                Salario = decimal.Parse(salarioEntry.Text),  
                Activo = activoSwitch.IsToggled  
            };  
  
            using (var datos = new DataAccess()) {  
                datos.InsertEmpleado(empleado);  
                listaListView.ItemsSource = datos.GetEmpleados();  
            }  
  
            nombresEntry.Text = string.Empty;  
            apellidosEntry.Text = string.Empty;  
            salarioEntry.Text = string.Empty;  
            fechaContratoDatePicker.Date = DateTime.Now;  
            activoSwitch.IsToggled = true;  
            await DisplayAlert("Confirmación", "Empleado agregado", "Aceptar");  
        }  
    }  
}

Pruebe lo que lleve hasta el momento

1. Agregue la clase **EmpleadoCell.cs** con:

using System;  
using Xamarin.Forms;  
  
namespace PreSQLite  
{  
    public class EmpleadoCell : ViewCell  
    {  
        public EmpleadoCell()  
        {  
            var idEmpleadoLabel = new Label {  
                HorizontalTextAlignment = TextAlignment.End,  
                HorizontalOptions = LayoutOptions.Start,  
                FontSize = 20,  
                FontAttributes = FontAttributes.Bold,  
            };  
  
            idEmpleadoLabel.SetBinding(Label.TextProperty, new Binding("IDEmpleado"));  
  
            var nombreCompetoLabel = new Label {  
                FontSize = 20,  
                FontAttributes = FontAttributes.Bold,  
                HorizontalOptions = LayoutOptions.StartAndExpand  
            };  
  
            nombreCompetoLabel.SetBinding(Label.TextProperty, new Binding("NombreCompleto"));  
  
            var fechaContratoLabel = new Label {  
                HorizontalOptions = LayoutOptions.StartAndExpand  
            };  
  
            fechaContratoLabel.SetBinding(Label.TextProperty, new Binding("FechaContrato"));  
  
            var salarioLabel = new Label {  
                HorizontalTextAlignment = TextAlignment.End,  
                HorizontalOptions = LayoutOptions.StartAndExpand  
            };  
  
            salarioLabel.SetBinding(Label.TextProperty, new Binding("Salario"));  
  
            var activoSwitch = new Switch {  
                IsEnabled = false,  
                HorizontalOptions = LayoutOptions.End  
            };  
  
            activoSwitch.SetBinding(Switch.IsToggledProperty, new Binding("Activo"));  
  
            var line1 = new StackLayout {  
                Orientation = StackOrientation.Horizontal,  
                Children = {  
                    idEmpleadoLabel, nombreCompetoLabel   
                },  
            };  
  
            var line2 = new StackLayout {  
                Orientation = StackOrientation.Horizontal,  
                Children = {  
                    fechaContratoLabel, salarioLabel, activoSwitch,   
                },  
            };  
  
            View = new StackLayout {  
                Orientation = StackOrientation.Vertical,  
                Children = {   
                    line1, line2,  
                },  
            };  
        }  
    }  
}

1. Agregue esta línea al constructor del **HomePage.xaml.cs**:

            listaListView.ItemTemplate = new DataTemplate(typeof(EmpleadoCell));

Pruebe el impacto de los cambios

|  |  |
| --- | --- |
|  |  |

1. Agregue la página **EditPage.xaml** con:

<?xml version="1.0" encoding="UTF-8"?>  
<ContentView xmlns="http://xamarin.com/schemas/2014/forms"   
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"   
             x:Class="PreSQLite.EditPage">  
    <ContentView.Content>  
        <ScrollView>  
            <StackLayout Padding="10" Spacing="10">  
                <Label Text="Editar Empleado"   
                       Font="24"   
                       HorizontalOptions="Center"   
                       VerticalOptions="Center"   
                       FontAttributes="Bold" />  
                <Entry x:Name="nombresEntry"   
                       Placeholder="Nombres" />  
                <Entry x:Name="apellidosEntry"   
                       Placeholder="Apellidos" />  
                <Entry x:Name="salarioEntry"   
                       Placeholder="Salario"   
                       Keyboard="Numeric" />  
                <StackLayout Orientation="Horizontal">  
                    <Label Text="Fecha Contratación: "   
                           VerticalOptions="Center" />  
                    <DatePicker x:Name="fechaContratoDatePicker" />  
                    <Label Text="Activo: "   
                           VerticalOptions="Center" />  
                    <Switch x:Name="activoSwitch"   
                            VerticalOptions="Center"   
                            IsToggled="True" />  
                </StackLayout>  
                <StackLayout Orientation="Horizontal">  
                    <Button x:Name="actualizarButton"   
                            Text="Actualizar"  
                            BackgroundColor="Teal"  
                            TextColor="White"  
                            HorizontalOptions="FillAndExpand"  
                            VerticalOptions="Center" />  
                    <Button x:Name="borrarButton"   
                            Text="Borrar"   
                            BackgroundColor="Red"  
                            TextColor="White"  
                            HorizontalOptions="FillAndExpand"  
                            VerticalOptions="Center" />

                </StackLayout>  
            </StackLayout>  
        </ScrollView>  
    </ContentView.Content>  
</ContentView>

1. Agregue el siguiente código a **EditPage.xaml.cs**:

using System;  
using System.Collections.Generic;  
using Xamarin.Forms;  
  
namespace PreSQLite  
{  
    public partial class EditPage : ContentPage  
    {  
        private Empleado empleado;  
  
        public EditPage (Empleado empleado)  
        {  
            InitializeComponent();  
            this.empleado = empleado;  
  
            actualizarButton.Clicked += actualizarButton\_Clicked;  
            borrarButton.Clicked += borrarButton\_Clicked;  
  
            nombresEntry.Text = empleado.Nombres;  
            apellidosEntry.Text = empleado.Apellidos;  
            fechaContratoDatePicker.Date = empleado.FechaContrato;  
            salarioEntry.Text = empleado.Salario.ToString();  
            activoSwitch.IsToggled = empleado.Activo;  
        }  
  
        private async void borrarButton\_Clicked(object sender, EventArgs e)  
        {  
            var rta = await DisplayAlert("Confirmación", "¿Desea borrar el empleado?", "Si", "No");  
            if (!rta) return;  
  
            using (var datos = new DataAccess()) {  
                datos.DeleteEmpleado(empleado);  
            }  
  
            await DisplayAlert("Confirmación", "Empleado borrado correctamente", "Aceptar");  
            *//await Navigation.PushAsync(new HomePage());*  
            await Navigation.PopAsync();  
        }  
  
        private async void actualizarButton\_Clicked(object sender, EventArgs e)  
        {  
            if (string.IsNullOrEmpty(nombresEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar nombres", "Aceptar");  
                nombresEntry.Focus();  
                return;  
            }  
  
            if (string.IsNullOrEmpty(apellidosEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar apellidos", "Aceptar");  
                apellidosEntry.Focus();  
                return;  
            }  
  
            if (string.IsNullOrEmpty(salarioEntry.Text)) {  
                await DisplayAlert("Error", "Debe ingresar salario", "Aceptar");  
                salarioEntry.Focus();  
                return;  
            }  
  
            empleado.Nombres = nombresEntry.Text;  
            empleado.Apellidos = apellidosEntry.Text;  
            empleado.Salario = decimal.Parse(salarioEntry.Text);  
            empleado.FechaContrato = fechaContratoDatePicker.Date;  
            empleado.Activo = activoSwitch.IsToggled;  
  
            using (var datos = new DataAccess()) {  
                datos.UpdateEmpleado(empleado);  
            }  
  
            await DisplayAlert("Confirmación", "Empleado actualizado correctamente", "Aceptar");  
            await Navigation.PopAsync();  
        }  
    }  
}

1. Agregue la siguiente línea al constructor de la clase **HomePage.xaml.cs**:

            listaListView.ItemSelected += listaListView\_ItemSelected;

1. Agregue el siguiente método a la clase **HomePage.xaml.cs**:

        private void listaListView\_ItemSelected(object sender, SelectedItemChangedEventArgs e)  
        {  
            Navigation.PushAsync(new EditPage((Empleado)e.SelectedItem));   
        }

Pruebe como quedo la aplicación finalizada

|  |  |
| --- | --- |
|  |  |