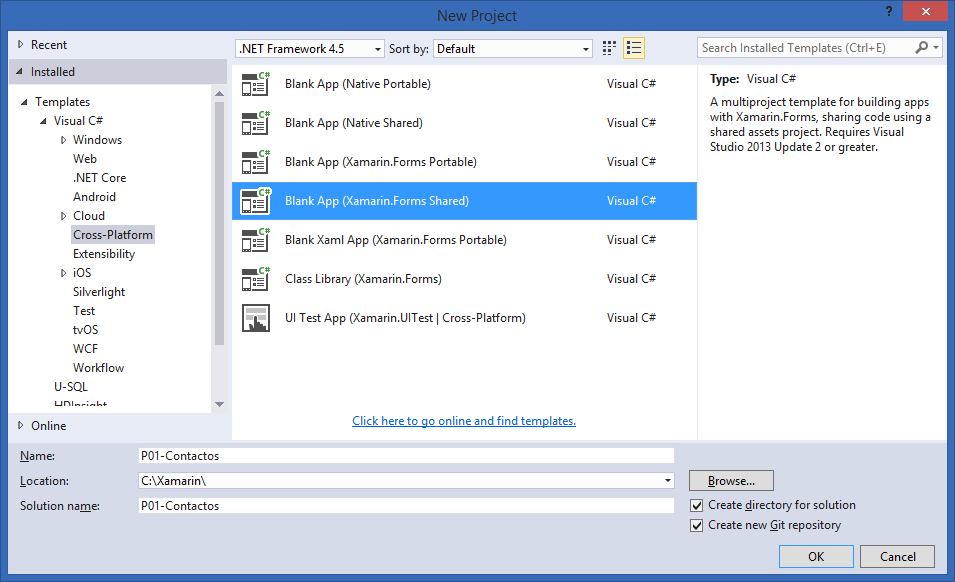
**Práctica 01 – Lista de Contactos**

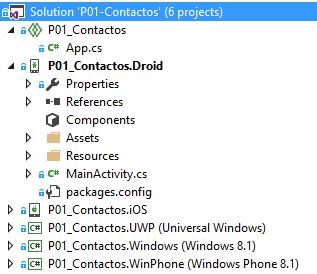
En esta práctica trabajaremos con los controles **ListView**, **Label** y **StackLayout**, además de un **ImageCell** para mostrar imágenes acompañadas de texto.

El objetivo es crear una aplicación que muestre una lista de contactos. En prácticas posteriores, añadiremos más funcionalidad a la app.

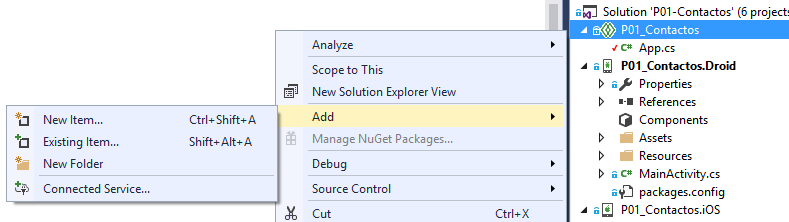
**Paso 1.** Crea el proyecto **P01-Contactos** que es de tipo **Xamarin.Forms Shared** (localizado bajo la categoría **Cross-Platform**):



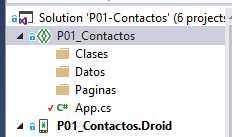
Una vez generado el espacio de trabajo, observarás los siguientes proyectos:



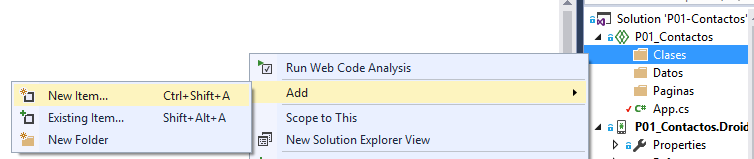
**Paso 2.** Agrega carpetas al proyecto compartido. Da clic derecho sobre el nombre del proyecto compartido (P01\_Contactos) en el Explorador de Soluciones y selecciona la opción **Agregar 🡪 Nueva carpeta**:



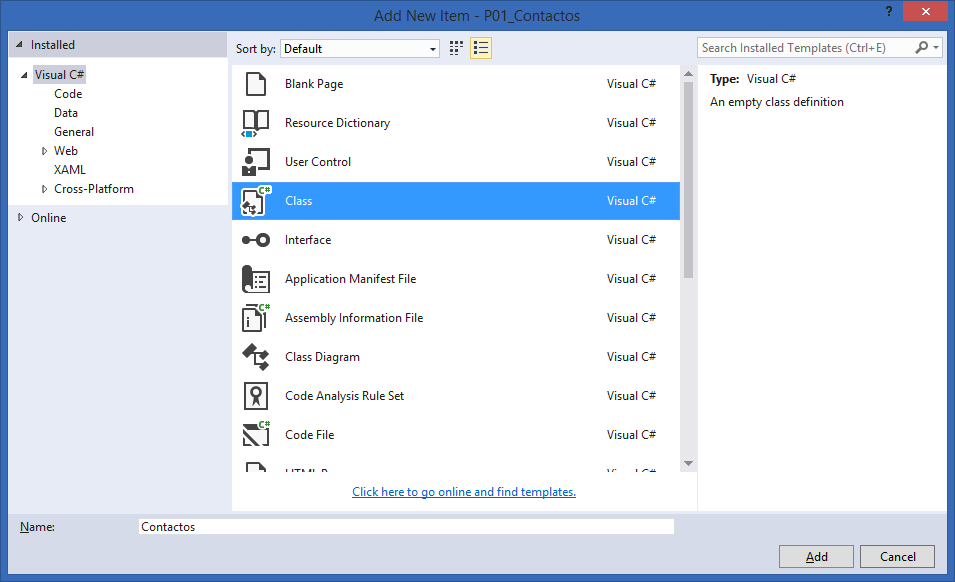
El nombre de la carpeta será **Paginas**. Repite el proceso para agregar las carpetas **Datos** y **Clases**.



**Paso 3.** Sobre la carpeta **Clases** da clic derecho y selecciona la opción **Agregar 🡪 Nuevo elemento**:



Agrega una **Clase** llamada **Contactos**:



**Paso 4.** A continuación se muestra el **código** de la clase **Contactos.cs**. Básicamente, se agregan 5 propiedades de tipo string (ID, Nombre, Foto, Telefono y Correo); además, contiene un **método** **estático** llamado **ObtenerContactos** que devuelve una lista que contiene 5 contactos:

using System;

using System.Collections.Generic;

using System.Text;

namespace P01\_Contactos.Clases

{

public class Contactos

{

public string ID { get; set; }

public string Nombre { get; set; }

public string Foto { get; set; }

public string Telefono { get; set; }

public string Correo { get; set; }

public static List<Contactos> ObtenerContactos()

{

Contactos c1 = new Contactos()

{

ID = System.Guid.NewGuid().ToString(),

Nombre = "Juan Perez",

Correo = "juan.perez@yahotgmail.com",

Telefono = "1234567890",

Foto = "juan.jpg"

};

Contactos c2 = new Contactos()

{

ID = System.Guid.NewGuid().ToString(),

Nombre = "Pepe Lopez",

Correo = "pepe.lopez@yahotgmail.com",

Telefono = "5557771234",

Foto = "pepe.jpg"

};

Contactos c3 = new Contactos()

{

ID = System.Guid.NewGuid().ToString(),

Nombre = "Miguel Sanchez",

Correo = "miguel.sanchez@yahotgmail.com",

Telefono = "6083942874",

Foto = "miguel.jpg"

};

Contactos c4 = new Contactos()

{

ID = System.Guid.NewGuid().ToString(),

Nombre = "Memo Suarez",

Correo = "memo.suarez@yahotgmail.com",

Telefono = "3095768329",

Foto = "memo.jpg"

};

Contactos c5 = new Contactos()

{

ID = System.Guid.NewGuid().ToString(),

Nombre = "Pedro Gonzalez",

Correo = "pedro.glez@yahotgmail.com",

Telefono = "9045679421",

Foto = "pedro.jpg"

};

List<Contactos> lista = new List<Contactos>();

lista.Add(c1);

lista.Add(c2);

lista.Add(c3);

lista.Add(c4);

lista.Add(c5);

return lista;

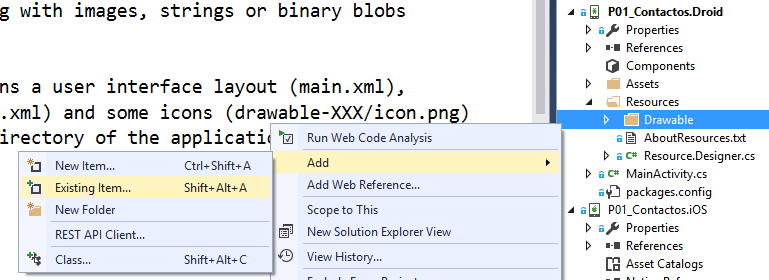
}

}

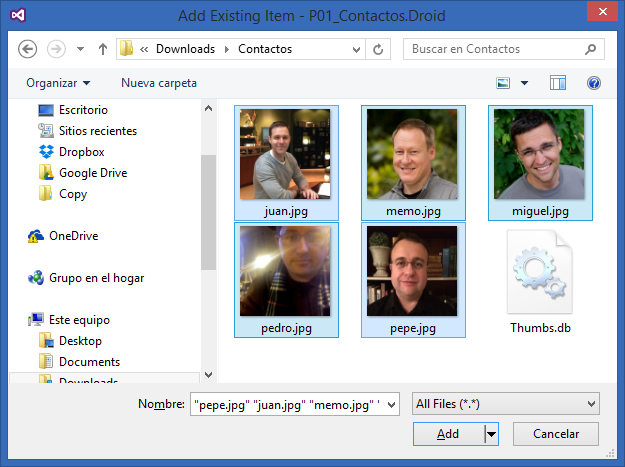
}

**Paso 5.** En el paso anterior se hace referencia a unas **imágenes** en la propiedad **Foto**. Las vamos a agregar a continuación. En Xamarin.Forms se recomienda que este tipo de recursos no se compartan en las 3 plataformas, sino que se agreguen en cada una de manera independiente (esto se debe a que cada plataforma maneja las imágenes de forma distinta, sobre todo para aspectos de resolución). Realiza los siguientes pasos en cada plataforma que tengas disponible o que desees probar:

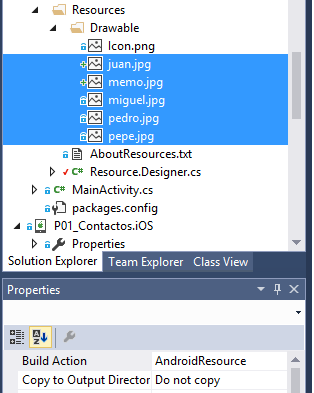
1. **Para Android:** En el proyecto **P01\_Contactos.Droid**, expande **Resources** y da clic derecho en la carpeta **Drawable**. Selecciona la opción **Agregar 🡪 Elemento existente**:



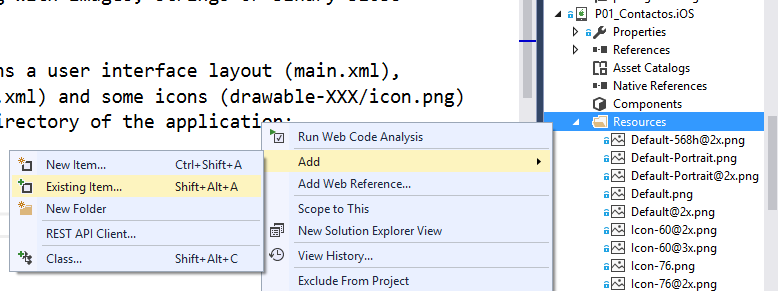
Selecciona las 5 imágenes que se incluyen en la práctica, correspondientes a nuestros contactos y agrégalas:



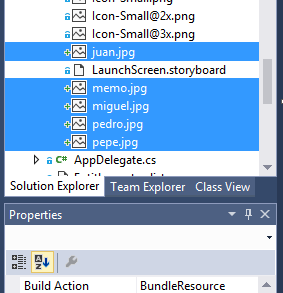
Finalmente, verifica que la acción de compilación (**Build Action**) de los 5 elementos esté establecida en **AndroidResource**.



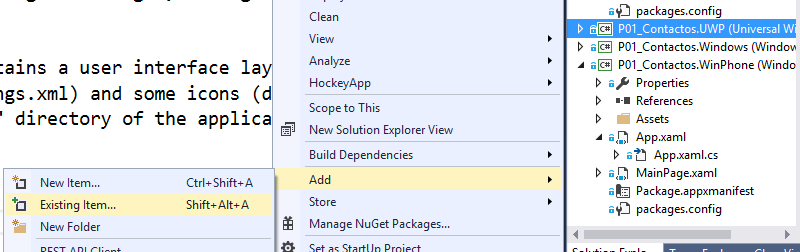
1. **Para iOS:** En el proyecto **P01\_Contactos.iOS**, da clic derecho en la carpeta **Resources** y selecciona la opción **Agregar 🡪 Elemento existente**:



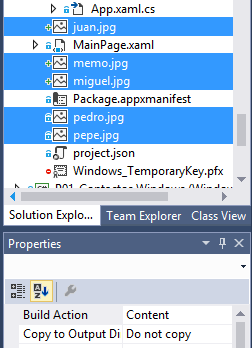
Selecciona las 5 imágenes y verifica que la acción de compilación esté establecida en **BundleResource**.



1. **Para Windows Phone, Windows y UWP:** Da clic derecho en el nombre del proyecto respectivoy selecciona la opción **Agregar 🡪 Elemento existente**:

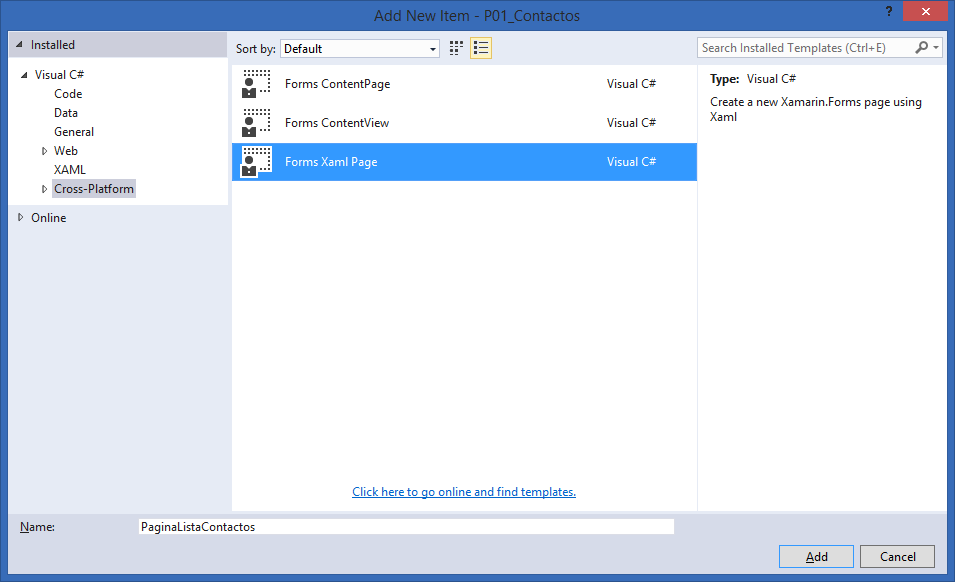


Selecciona las 5 imágenes y verifica que la acción de compilación esté establecida en **Content**.



**Paso 6.** Regresando al proyecto compartido P01\_Contactos, da clic derecho sobre la carpeta **Paginas** y selecciona **Agregar 🡪 Nuevo elemento**.

Selecciona **Forms Xaml Page** (en la categoría Cross-Platform) para crear una nueva página que incluya diseño (XAML) y lógica (C#), escribe el nombre **PaginaListaContactos**:



**Paso 7.** A continuación se muestra el código **XAML** de dicha página. Se incluye un **StackLayout** que muestra sus elementos uno debajo del otro (esto se llama apilamiento). Dichos elementos son un **Label** (etiqueta, a manera de título) y un **ListView** (lista para mostrar los contactos).

Cada elemento contiene propiedades para modificar su aspecto; por ejemplo el **Padding** del StackLayout es el espacio de separación alrededor del control y **FontSize** representa el tamaño del texto. Para el caso del **ListView** llamado **lsvContactos**, se utiliza su propiedad **ItemTemplate** para establecer que cada elemento será incluido dentro de un **ImageCell**, que es una tipo de celda utilizada para mostrar una imagen acompañada de 2 textos (un título y un detalle).

<?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="P01\_Contactos.Paginas.PaginaListaContactos">

<ContentPage.Content>

<StackLayout Padding="5,20,10,0" >

<Label Text="Contactos" FontSize="40" HorizontalOptions="Center"/>

<ListView x:Name="lsvContactos">

<ListView.ItemTemplate>

<DataTemplate>

<ImageCell Text="{Binding Nombre}"

Detail="{Binding Telefono}"

ImageSource="{Binding Foto}"/>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</StackLayout>

</ContentPage.Content>

</ContentPage>

Observa que en el ImageCell se enlazan las propiedades Nombre, Telefono y Foto (que definimos en la clase Contactos). Esto se llama **DataBinding** (Enlace de datos). Cuando se asignen los datos al ListView, cada contacto será representado por un ImageCell, donde las propiedades anteriormente mencionadas se asignarán a los atributos del control.

**Paso 8.** Accede al código de PaginaListaContactos.xaml.cs. Este archivo representa el “code-behind” de la página. Para este caso, simplemente se generará un método llamado **MostrarContactos**, que a su vez llama al método estático **ObtenerContactos** de la clase **Contactos** para cargar una **List** de objetos; a su vez, esta lista se pasa al control **lsvContactos** en su propiedad **ItemsSource** para que se muestren en la página. El método se llama después de que se han cargado los controles (en el constructor).

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using P01\_Contactos.Clases;

namespace P01\_Contactos.Paginas

{

public partial class PaginaListaContactos : ContentPage

{

public PaginaListaContactos ()

{

InitializeComponent ();

MostrarContactos();

}

void MostrarContactos()

{

List<Contactos> lista = Contactos.ObtenerContactos();

lsvContactos.ItemsSource = lista;

}

}

}

**Paso 9.** Finalmente, modifica la página de inicio de este proyecto. En la clase **App** del proyecto compartido (P01\_Contactos) asigna una nueva instancia de **PaginaListaContactos** al objeto **MainPage**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Xamarin.Forms;

using P01\_Contactos.Paginas;

namespace P01\_Contactos

{

public class App : Application

{

public App ()

{

// The root page of your application

MainPage = new PaginaListaContactos();

}

protected override void OnStart ()

{

// Handle when your app starts

}

protected override void OnSleep ()

{

// Handle when your app sleeps

}

protected override void OnResume ()

{

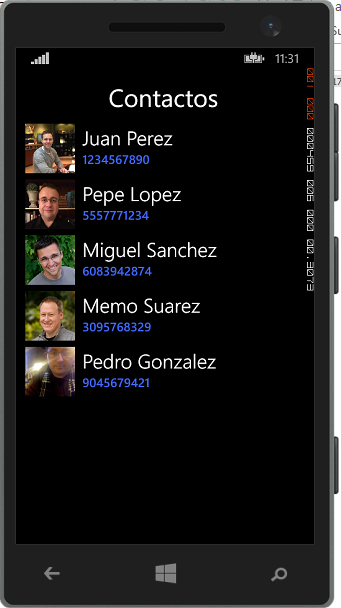
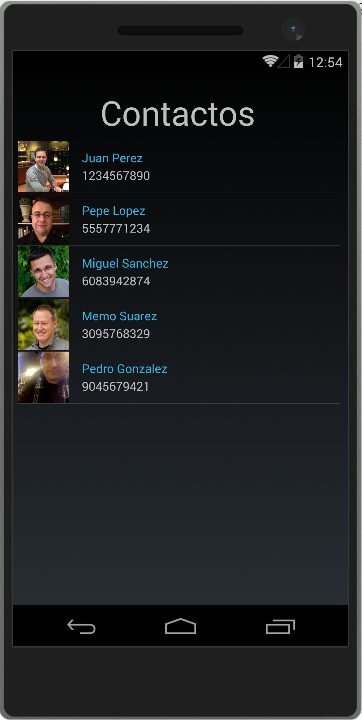
// Handle when your app resumes

}

}

}

**Paso 10.** Compila y ejecuta la aplicación. Se debe mostrar la lista de contactos

Has terminado tu primer aplicación con Xamarin.Forms. ¡Felicidades!