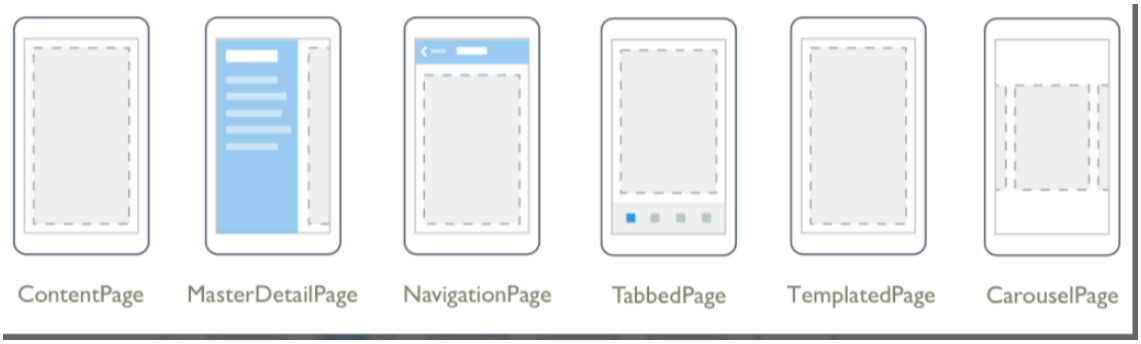
PAGES XAMARIN FORMS



CONTENT PAGE

Solo puede desplegar un único view a la vez el cual puede ser un layout como un grid, stacklayout o controlview

MasterPage

Permite administrar dos seccions de información, por un lado la sección de los datos generales es decir los datos maestros y por otra los detalles

Cambiamos Contenpage por lo siguiente:

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  xmlns:local="clr-namespace:DemoMasterDetailPage"
                  x:Class="DemoMasterDetailPage.MainPage">

    <Label Text="Welcome to Xamarin.Forms!"
           VerticalOptions="Center"
           HorizontalOptions="Center" />

</MasterDetailPage>
```

Y ahora en el codebehind

```
namespace DemoMasterDetailPage
{
    using Xamarin.Forms;
    using System;
    public partial class MainPage : MasterDetailPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

```
<?xml version="1.0" encoding="utf-8" ?>
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  xmlns:local="clr-namespace:DemoMasterDetailPage"
                  x:Class="DemoMasterDetailPage.MainPage">

    <MasterDetailPage.Master>
        <ContentPage Title="Maestro">
            <StackLayout>
                <Button Text="Soy el Maestro"></Button>
            </StackLayout>
        </ContentPage>
    </MasterDetailPage.Master>
    <MasterDetailPage.Detail>
        <ContentPage>
            <StackLayout>
                <Button Text="Soy el Detalle"></Button>
            </StackLayout>
        </ContentPage>
    </MasterDetailPage.Detail>

</MasterDetailPage>
```

Creamos dos paginas una que se llame maestro y otra detalle, ahora pegamos todo lo que teníamos en el maestro

```
<?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="DemoMasterDetailPage.Maestro"
             Title="Maestro">
    <ContentPage.Content>
        <StackLayout>
            <Button Text="Soy el Maestro"></Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Hacemos lo mismo en el 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="DemoMasterDetailPage.Detalles">
    <ContentPage.Content>
        <StackLayout>
            <Button Text="Soy el Detalle"></Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Ahora lo que vamos hacer es importar el espacio de nombres de la siguiente forma:

```
<?xml version="1.0" encoding="utf-8" ?>
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:DemoMasterDetailPage"
             x:Class="DemoMasterDetailPage.MainPage">

    <MasterDetailPage.Master>
        <local:Maestro></local:Maestro>
    </MasterDetailPage.Master>
    <MasterDetailPage.Detail>
        <local:Detalles></local:Detalles>
    </MasterDetailPage.Detail>

</MasterDetailPage>
```

Aquí nos genera un error ya que nos debemos asegurar que siempre la página maestra tenga el atributo Title

NAVIGATION

Gestiona la navegación y la experiencia de usuario de una pila de otras paginas no es un tipo de pagina que implementa una estructura de interfaz de usuario como el resto de las páginas, es decir no es un tipo de pagina que le coloquemos navigation page en su lugar se hace esto, ya que dentro del constructor de la pagina raíz se la indicamos de la siguiente forma:

```
public App ()
{
    InitializeComponent();

    MainPage = new NavigationPage(new MainPage());
}
```

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:DemoNavigationPage"
             x:Class="DemoNavigationPage.MainPage"
             Title="NAvigation Page">

    <StackLayout>
        <Button WidthRequest="300"
                HeightRequest="300"
                Text="Estos es navigation Page">
        </Button>
    </StackLayout>

</ContentPage>
```

Recordemos que siempre debemos agregarle la propiedad Title o de lo contrario la barra de navegación aparece en blanco, si le quisieras cambiar el color a la barra de navegación haríamos lo siguiente :

```
public App ()
{
    InitializeComponent();
    var navigationPage = new NavigationPage(new MainPage());
    navigationPage.BarBackgroundColor = Color.Black;
    navigationPage.BarTextColor = Color.White;
    MainPage = navigationPage;


}
```

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:DemoNavigationPage"
             x:Class="DemoNavigationPage.MainPage"
             Title="Navigation Page"
             >

    <StackLayout>
        <Label Text="Esta es una página de navegación"
               ></Label>
        <Button
               Text="Página 2"
               Clicked="SiguienteButton_Clicked"
                >
        </Button>
    </StackLayout>

</ContentPage>
```

Y el codebehind hacemos lo siguiente:Entonces el PuschAsync nos está empujando a la página , no olviden hacer el método asíncrono

```
namespace DemoNavigationPage
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();

        }
        public async void SiguienteButton_Clicked(Object sender, EventArgs args)
        {
            await Navigation.PushAsync(new Pagina2());
        }

    }
```

Como pueden ver para desaparecer la página invocamos el método Pop pero en este caso no lo utilizamos que se encuentra autoimplementado dentro del método de la mainpage

Creamos los recursos en App.xaml

```
<Application xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="DemoNavigationPage.App">
    <Application.Resources>
        <ResourceDictionary>
            <Style TargetType ="Button" >
                <Setter Property="FontSize"/>
                <Setter Property="Margin" Value="10"/>
                <Setter Property="BackgroundColor" Value="BlueViolet"/>
                <Setter Property="HorizontalOptions" Value="CenterAndExpand"/>
                <Setter Property="TextColor" Value="White"/>
            </Style>
            <Style TargetType ="Label" >
                <Setter Property="FontSize" Value="Large"/>
                <Setter Property="FontAttributes" Value="Bold,Italic"/>
                <Setter Property="HorizontalOptions" Value="CenterAndExpand"/>
                <Setter Property="VerticalOptions" Value="CenterAndExpand"/>

            </Style>

        </ResourceDictionary>
```

Pagina 2

```
<?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="DemoNavigationPage.Pagina2"
             Title="Página 2">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Esta es la página dos"
                />
            <Label x:Name="lbOpcion"
                />
            <Button x:Name="btnTest"
                    Text="Test"
                    />
            <Button x:Name="btnTres"
                    Text="Siguiente"
                    />


        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

```
[XamlCompilation(XamlCompilationOptions.Compile)]
public partial class Pagina2 : ContentPage
{
    public Pagina2 ()
    {
        InitializeComponent ();
        this.lbOpcion.Text = string.Empty;
        btnTres.Clicked += BtnTres_Clicked;
        btnTest.Clicked += BtnTest_Clicked;
    }
```

Mensajería:

Es el poder de mostrar al usuario ciertas cajas de dialogo como respuesta a alguna solicitud, o para ser notificado de alguna condición de la lógica de la aplicación como estamos haciendo con el DisplyAlert en la parte inferior. La clase Page incluye los métodos DisplyAlert() el cual ya habíamos visto y el DisplayACtionSheet() , el cual es un método asíncrono que permite mostrar una caja de dailogo con una lista de opciones para que el usuario seleccione alguna de ellas. La desventaja es que la lista de opciones solamente pueden ser cadenas, este método recibe en primera instancia, el título, el string del botón de cancelar, el tercer parámetro que es aconsejable dejarlo en null, y finalmente la lista de opciones que vienen manejándose como un arreglo

```
public MainPage()
{
    InitializeComponent();
    DisplayActionSheet()
}
```

```
(awaitable) Task<string> Page.DisplayActionSheet(string title, string cancel, string destruction, params string[] buttons)

Displays a native platform action sheet, allowing the application user to choose from several buttons.
Usage:
 string x = await DisplayActionSheet(...);

title: Title of the displayed action sheet. Must not be null.
```

```
private async void BtnTest_Clicked(object sender, EventArgs e)
{
    this.lbOpcion.Text = string.Empty;
    var resp = await DisplayAlert("Test", "Esta seguro de realizar el test", "Si", "No");
    if (resp)
    {
        await DisplayAlert("Inicio de Test", "Super bienvenido","Aceptar");
        var opcion = await DisplayActionSheet("Opciones",
                                              "Cancelar",
                                              null, "Opcion 1", "Opcion 2", "Opcion 3");
        lbOpcion.Text = opcion.ToString();

    }
    else
    {
        await DisplayAlert("Test Fallido", "En otra ocasión sera", "Aceptar");
    }
}
```

```
private async void BtnTres_Clicked(object sender, EventArgs e)
{
    this.lbOpcion.Text = string.Empty;
    await Navigation.PushAsync(new Page3());
}
```

Pagina 3

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="DemoNavigationPage.Page3"
             Title="Página 3">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Esta es la página tres"
                />
            <StackLayout Orientation="Horizontal">
                <Button x:Name="btnInicial"
                    Text="Inicio"
                    />
                <Button x:Name="btnAtras"
                    Text="Anterior"
                    />
                <Button x:Name="btnModal"
                    Text="Siguiente"
                    />
            </StackLayout>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

```
namespace DemoNavigationPage
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class Page3 : ContentPage
    {
        public Page3 ()
        {
            InitializeComponent ();
            this.btnInicial.Clicked += BtnInicial_Clicked;
            this.btnModal.Clicked += BtnModal_Clicked;
            this.btnAtras.Clicked += BtnAtras_Clicked;
        }
```

Entonces aquí vemos que para retroceder en una página utilizamos PopAsync(), para mostrar una página modal PushModalAsync(), y para regresar a la primera página PopToRootAsync()

```
    private async void BtnAtras_Clicked(object sender, EventArgs e)
    {
        await Navigation.PopAsync();
    }

    private async void BtnModal_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushModalAsync(new Page4());
    }

    private async void BtnInicial_Clicked(object sender, EventArgs e)
    {
        await Navigation.PopToRootAsync();
    }
}
```

Pagina 4

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="DemoNavigationPage.Page4">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Esta es una página Modal"
                />
            <Button x:Name="btnCerrar"
                    Text="Cerrar"
                    >
            </Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Finalmente como deseamos cerrar la página Modal utilizamos PopModalAsync()

```
namespace DemoNavigationPage
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class Page4 : ContentPage
    {
        public Page4 ()
        {
            InitializeComponent ();
            btnCerrar.Clicked += BtnCerrar_Clicked;
        }

        private void BtnCerrar_Clicked(object sender, EventArgs e)
        {
            Navigation.PopModalAsync();
        }
    }
}
```

TabbedPage

Primero cambiar el content page por el tabbedPage

```
<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
            xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
            xmlns:local="clr-namespace:DemoTabbedPageDetail"
            x:Class="DemoTabbedPageDetail.MainPage">

    <TabbedPage.Children>
        <ContentPage Title="Datos Básicos">
            <StackLayout HorizontalOptions="Center"
                         VerticalOptions="Center">
                <Label Text="Estos son los datso básicos"></Label>
            </StackLayout>
        </ContentPage>
        <ContentPage Title="Intereses">
            <StackLayout HorizontalOptions="Center"
                         VerticalOptions="Center">
                <Label Text="Estos son los intereses"></Label>
            </StackLayout>
        </ContentPage>
        <ContentPage Title="Ocupaciones">
            <StackLayout HorizontalOptions="Center"
                         VerticalOptions="Center">
                <Label Text="Estos son las ocupaciones"></Label>
            </StackLayout>
        </ContentPage>
    </TabbedPage.Children>

</TabbedPage>
```

Y cambiamos la forma de heredar del mainpage

```
public partial class MainPage : TabbedPage
{
    public MainPage()
    {
        InitializeComponent();
    }
}
```

CarouselPage

Primero heredamos de carouselpage en el main page

```
namespace DemoCarouselPage
{
    public partial class MainPage : CarouselPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

Y agregamos las correspondientes imágenes a las páginas que ya conocemos

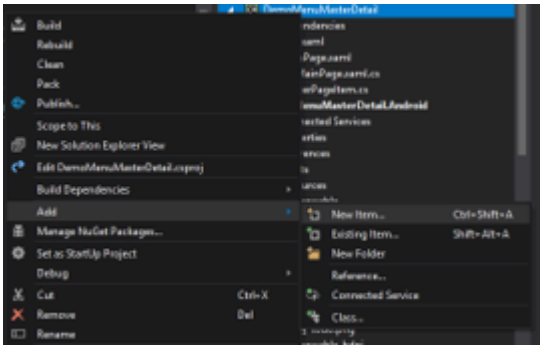
```xml
<?xml version="1.0" encoding="utf-8" ?>
<CarouselPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xam
              xmlns:local="clr-namespace:DemoCarouselPage"
              x:Class="DemoCarouselPage.MainPage">

    <ContentPage>
        <StackLayout HorizontalOptions="Center"
                     VerticalOptions="Center">
            <Image Source="github.png"></Image>
        </StackLayout>
    </ContentPage>
    <ContentPage>
        <StackLayout HorizontalOptions="Center"
                     VerticalOptions="Center">
            <Image Source="githubDesktop.png"></Image>
        </StackLayout>
    </ContentPage>
    <ContentPage>
        <StackLayout HorizontalOptions="Center"
                     VerticalOptions="Center">
            <Image Source="gitlab.png"></Image>
        </StackLayout>
    </ContentPage>
    <ContentPage>
        <StackLayout HorizontalOptions="Center"
                     VerticalOptions="Center">
            <Image Source="xamarin.png"></Image>
        </StackLayout>
    </ContentPage>

</CarouselPage>
```

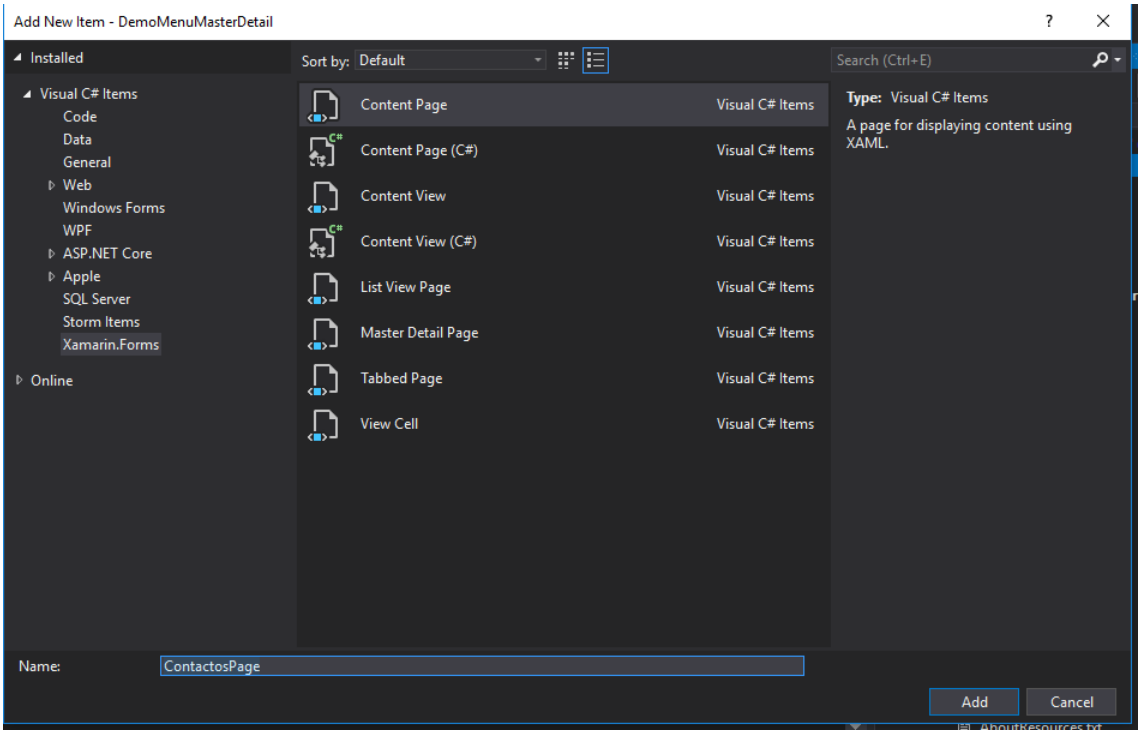
DemoMenuMasterDetail

Creamos una clase que se llame `MasterPageItem` y dentro de ella colocamos lo siguiente:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace DemoMenuMasterDetail
{
    public  class MasterPageItem
    {
        public string Titulo { get; set; }

        public string Icono { get; set; }

        public Type TipoPagina { get; set; }
    }
}
```

Creamos tres páginas: ContactosPage, ListadoPage y AlertasPage



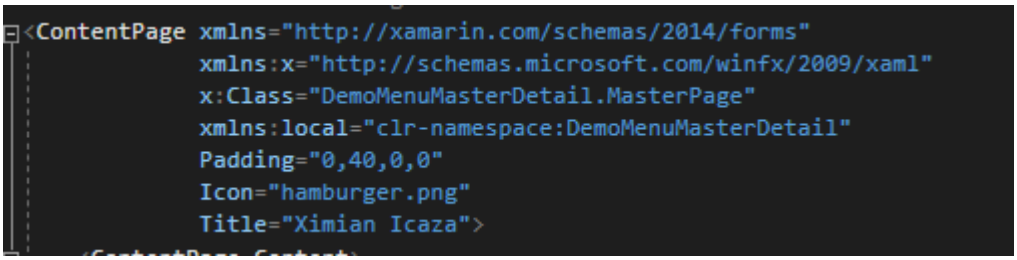


A cada una le agregamos el título y al dejamos de la siguiente forma:

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="DemoMenuMasterDetail.AlertasPage"
             Title="Alertas">
    <ContentPage.Content>
        <StackLayout>
            <Image Source="alertas.png"
                   HorizontalOptions="CenterAndExpand"
                   VerticalOptions="CenterAndExpand"/>
            <Label Text="Esta es la página de alertas"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />

        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Agregamos una nueva página y la llamamos MasterPage y hacemos lo siguiente

No olvidemos especificar el alias de local



ListView es uno de los controles más utilizamos en Xamarin Forms para mostrar la lista de los iconos entonces utilizamos ListView el tiene una propiedad que se llama Items.Source el cual hace referencia a los items que va a contener en este caso Contactos, listado, y alertas, los cuales en nuestro caso los toma como un arreglo y le especificamos que son de tipo MasterPageItem que fue la clase que acabamos de crear, en el tipo de página especificamos que página es la que deseamos llamar.

Por otra parte ListView contiene otra propiedad que se llama ItemTemplate en donde especificamos una plantilla de datos quienes son de tipo DateTemplate y dentro del DateTemplate siempre vamos a colocar un control tipo Cell rcordemos que los view tipo cell son:

- EntryCell
- ImageCell
- SwitchCell
- TextCell
- ViewCell: El cual nos permite establecer cualquier estructura de elementos en XAML, por esta razón colocamos el grid dentro del mismo lo cual responde a la pregunta de Oliverio.

Ahora vemos que en la imagen estamos haciendo un Binding a Icono y un Binding a Titulo, esto quiere decir que le estamos haciendo un enlace a la propiedad Icono y titulo que definimos en la parte superior cuando llamamos a la clase MasterPageItem.

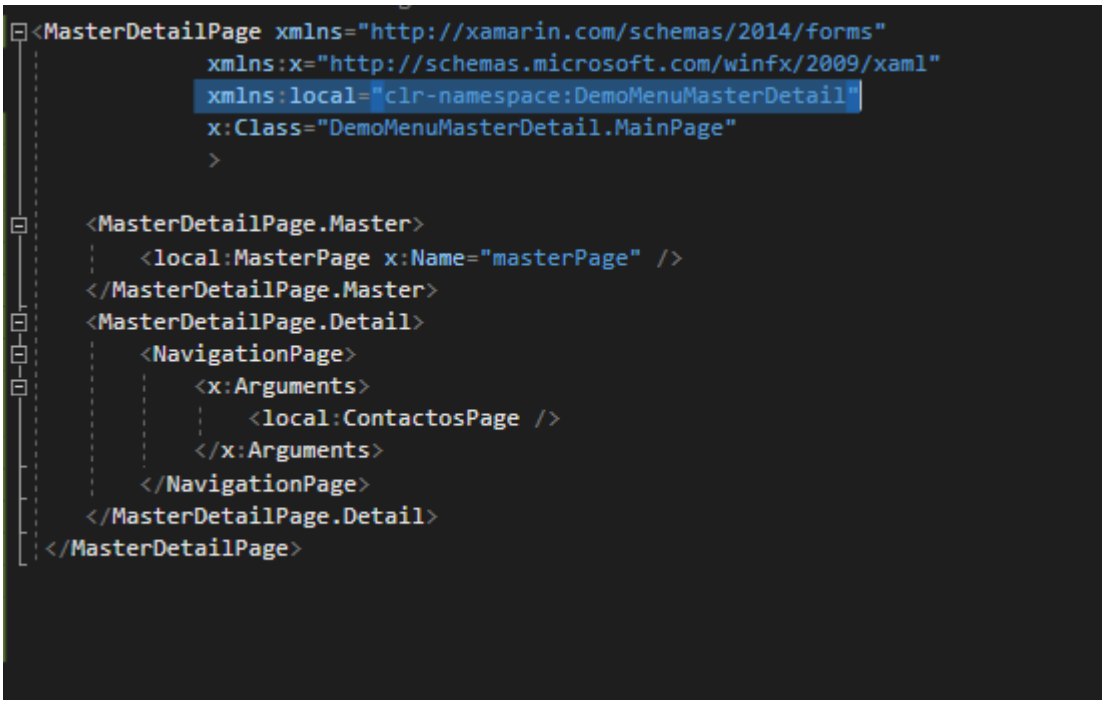
```
<ContentPage.Content>
    <StackLayout>
        <ListView x:Name="listView">
            <ListView.ItemsSource>
                <x:Array Type="{x:Type local:MasterPageItem}">
                    <local:MasterPageItem Titulo="Contactos" Icono="contacts.png" TipoPagina="{x:Type local:ContactosPage}" />
                    <local:MasterPageItem Titulo="Listado" Icono="todo.png" TipoPagina="{x:Type local:ListadoPage}" />
                    <local:MasterPageItem Titulo="Alertas" Icono="reminders.png" TipoPagina="{x:Type local:AlertasPage}" />
                </x:Array>
            </ListView.ItemsSource>
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <Grid Padding="5,10">
                            <Grid.ColumnDefinitions>
                                <ColumnDefinition Width="30"/>
                                <ColumnDefinition Width="*" />
                            </Grid.ColumnDefinitions>
                            <Image Source="{Binding Icono}" />
                            <Label Grid.Column="1" Text="{Binding Titulo}" />
                        </Grid>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>
</ContentPage.Content>
</ContentPage>
```

En el Code Behind hacemos lo siguiente, donde retornmaos la lista tal cual la nombras en x:Name

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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

namespace DemoMenuMasterDetail
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class MasterPage : ContentPage
    {
        public ListView ListView { get { return listView; } }
        public MasterPage ()
        {
            InitializeComponent ();
        }
    }
}
```

En el archivo MainPage lo convertimos en MasterDetailPage

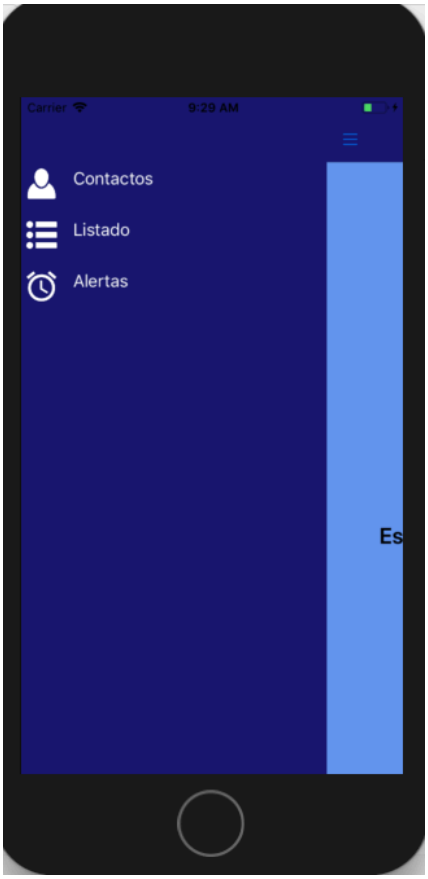
Y en el Detail Page estamos diciendo que vamos a llamar a una página ContactosPage



```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  xmlns:local="clr-namespace:DemoMenuMasterDetail"
                  x:Class="DemoMenuMasterDetail.MainPage"
                  >

    <MasterDetailPage.Master>
        <local:MasterPage x:Name="masterPage" />
    </MasterDetailPage.Master>
    <MasterDetailPage.Detail>
        <NavigationPage>
            <x:Arguments>
                <local:ContactosPage />
            </x:Arguments>
        </NavigationPage>
    </MasterDetailPage.Detail>
</MasterDetailPage>
```

Y en el Code Behind hacemos lo siguiente, finalmente generamos el evento cuando seleccionamos un elemento de la lista donde lo que hacemos es llamar a la página que acabamos de seleccionar

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;

namespace DemoMenuMasterDetail
{
    public partial class MainPage : MasterDetailPage
    {
        public MainPage()
        {
            InitializeComponent();
            masterPage.ListView.ItemSelected += OnItemSelected;
        }

        void OnItemSelected(object sender, SelectedItemChangedEventArgs e)
        {
            var item = e.SelectedItem as MasterPageItem;
            if (item != null)
            {
                Detail = new NavigationPage((Page)Activator.CreateInstance(item.TipoPagina));
                masterPage.ListView.SelectedItem = null;
                IsPresented = false;
            }
        }
    }
}
```


Carrier
9:29 AM
Contactos
Listado
Alertas
Es