

UNIDAD 7:

PAGES EN

.NET MAUI

Páginas

- En el tema anterior comentamos que los posibles elementos raíz en un archivo XAML de MAUI son:
 - Application
 - Shell
 - ContentPage
 - TabbedPage
 - FlayoutPage
 - NavigationPage
 - ResourceDictionary
- Nos centraremos en los tipos de páginas (pages) que podemos encontrar en .NET MAUI

} PAGES

ContentPage

- Es la página más sencilla.
- Solamente puede tener un único elemento raíz.
- Ese hijo puede tener múltiples hijos.
- En ella se suele colocar como hijo un elemento “Grid” o un “StackLayout”

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MyMauiApp.MyPage"
    Title="MyPage"
    BackgroundColor="White">
    <StackLayout>
        <Label Text="Hola Mundo"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
    </StackLayout>
</ContentPage>
```



TabbedPage

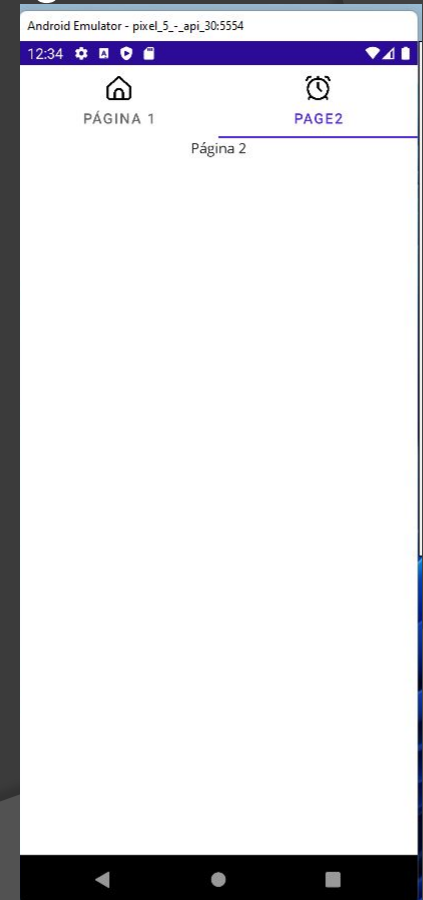
- En ella usaremos Tabs para visualizar una de las páginas.
- Se recomienda para 2-5 páginas.
- Tenemos que cambiar el tipo de clase heredada en el Code Behind a TabbedPage

```
<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"  
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
  xmlns:local="clr-namespace:_07_Pages.Views"  
  x:Class="_07_Pages.Views.TabbedPage1"  
  Title="TabbedPage">
```

```
<local:Page1></local:Page1>  
<local:Page2></local:Page2>
```

```
</TabbedPage>
```

Tenemos que añadir esa línea para poder usar local:



TabbedPage

- Los únicos hijos de este tipo de páginas son ContentPage o NavigationPage.
- Es incompatible con Shell.
- La localización de las pestañas será:
 - IOS: parte de abajo y con iconos.
 - Android: parte de arriba y con iconos. Se puede colocar en la parte de abajo especificando un cambio para esta plataforma.

```
<TabbedPage ...
```

```
xmlns:android="clr-namespace:Microsoft.Maui.Controls.PlatformConfiguration.AndroidSpecific;assembly=Microsoft.Maui.Controls"
```

```
    android:TabbedPage.ToolbarPlacement="Bottom">
```

```
    ...
```

```
</TabbedPage>
```

- Windows: parte de arriba sin iconos.

NavigationPage

- Es incompatible con Shell.
- Nos da la posibilidad de navegar entre
 - diferentes páginas.
- Lo hace mediante una pila LIFO.



Last In-First Out

NavigationPage

- Para crear una NavigationPage debemos hacer lo siguiente:
- Abrir el archivo App.Xaml.CS
- Cambiar la línea
 - `MainPage = new new MainPage();`
- Por
 - `MainPage = new NavigationPage(new MainPage());`
- De esta forma MainPage será la primera página en la pila de navegación (Root Page).
- Si queremos cambiar propiedades, podemos instanciar primero un objeto de tipo NavigationPage, cambiarle sus propiedades y después asignárselo a MainPage.
BarBackground, BarBackgroundColor, BackButtonTitle, etc.
- Para navegar (al hacer click en un botón, por ejemplo):
`await Navigation.PushAsync(new DetailsPage());`

NavigationPage

- InsertPageBefore: inserta una página detrás de una especificada:



- RemovePage: elimina una página especificada de la pila

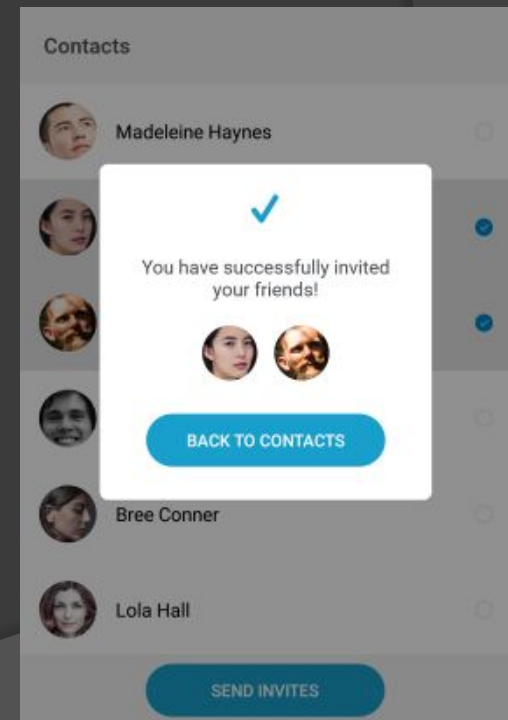


NavigationPage

Navegación MODAL

- Una página puede ser mostrada de forma modal.
- Modal significa que el usuario debe interactuar con la nueva ventana antes de volver a la anterior.
- La página anterior se queda “bloqueada”.
- El usuario puede volver pulsando el botón Back y saldrá de la pila.
- Para abrir una página:

```
await Navigation.PushModalAsync(new DetailsPage());
```



NavigationPage

Pasar datos

Tenemos dos formas:

- Usando constructor por parámetros

```
Contact contact = new Contact
{
    Name = "Jane Doe",
    Age = 30,
    Occupation = "Developer",
    Country = "USA"
};
...
await Navigation.PushModalAsync(new DetailsPage(contact));
```

NavigationPage

- Usando BindingContext

```
Contact contact = new Contact
{
    Name = "Jane Doe",
    Age = 30,
    Occupation = "Developer",
    Country = "USA"
};

await Navigation.PushAsync(new DetailsPage
{
    BindingContext = contact
});
```

XAML

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MyMauiApp.DetailsPage"
    Title="Details">
    <StackLayout>
        <Label Text="{Binding Name}" />
        <Label Text="{Binding Occupation}" />
    </StackLayout>
</ContentPage>
```

FlyoutPage

- Es incompatible con Shell.
- Contiene dos elementos:
 - Una página Flyout o flotante
 - Una página de detalles
- Lo normal es que se use para mostrar un menú y seleccionar páginas que queramos abrir en la parte “Detalle”
- Una de las propiedades más importantes es FlyoutLayoutBehavior que define la forma de comportarse el menú

