# 1.1 Pages

A screenshot of a cell phone

Description generated with very high confidence

Todos los tipos de página que se describen a continuación se derivan de Xamarin.Forms Page clase. Estos elementos visuales ocupan toda o la mayoría de la pantalla. Un Page objeto representa un ViewController en iOS y un Page en la plataforma Universal de Windows. En Android, cada página ocupa la pantalla como una Activity, pero las páginas de Xamarin.Forms Activity.

## Properties

|  |  |
| --- | --- |
| Property | Value |
| BackgroundColor | Define el color de fondo |
| BackgroundImage | Define una imagen de fondo |
| HeightRequest | Define la altura de la pagina |
| Icon | Define el icono asociado a la pagina |
| InputTransparent | Indica si el elemento tuviera interacción dentro del ciclo |
| IsEnable | Define si el elemento este o no habilitado para la interacción. |
| IsVisible | Determina si el elemento es visible. |
| Navigation | Obtiene la navegación |
| Padding | Define el espacio entre el contenido y la pagina |
| Style | Determina el estilo del elemento |
| Title | Define el título de la pagina |
| ToolbarItems | Determinar los ToolbarItems |
| WidthRequest | Determina el ancho del elemento |

## Methods

|  |  |
| --- | --- |
| Methods | Value |
| DisplayActionSheet(string, string, string, string[]) | Muestra una alerta nativa con opciones de seleccion |
| DisplayAlert(String, String, String) | Muestra una alerta con un solo botón de cancelar |
| DisplayAlert(String, String, String, string) | Muestra una alerta con un botón de aceptar y cancelar |
| OnAppearing() | Se ejecuta inmediatamente antes que la pantalla sea visible |
| OnBackButtonPressed() | Con este método podemos sobrescribir que es lo que asa al presionar el botón de back. |
| OnDisappearing() | Se ejecuta inmediatamente antes de que la pantalla deje de ser visible |

# 1.2 ContenPage

Es el tipo más sencillo y común de paginas establecer el Content a una sola vista objeto, que es mas a menudo un layout como un Stacklayout, Grid o Scrollview.

## 1.2.1 Propiedades

|  |  |
| --- | --- |
| Property | Value |
| Content | Define el contenido de la pantalla a mostrar |

## 1.2.2 XAML

1. **<ContentPage** xmlns="http://xamarin.com/schemas/2014/forms"
2. xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
3. xmlns:local="clr-namespace:XamlSamples"
4. x:Class="XamlSamples.MainPage"**>**
6. **<StackLayout>**
7. <!-- Place new controls here -->
8. **<Label** Text="Welcome to Xamarin Forms!"
9. VerticalOptions="Center"
10. HorizontalOptions="Center" **/>**
11. **</StackLayout>**
13. **</ContentPage>**

# 1.3 MasterDetailPage

En MasterDetailPage administra 2 paneles. *El Master* es una página en donde podríamos mostrar un menú mediante una lista. Y el *Detail* Muestra el elemento selección en pantalla actual.

## 1.3.1 Propiedades

|  |  |
| --- | --- |
| Property | Value |
| Detail | Define la pantalla a mostrar en pantalla proveniente de las opciones del master |
| IsPreset | Define si la pantalla Master es mostrada o oculta |
| Master | Define la pantalla donde usualmente se agrega un menu |

## 1.3.2 XAML

1. **<MasterDetailPage** xmlns="http://xamarin.com/schemas/2014/forms"
2. xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
3. xmlns:local="clr-namespace:MasterDetailPageNavigation;assembly=MasterDetailPageNavigation"
4. x:Class="MasterDetailPageNavigation.MainPage"**>**
5. **<MasterDetailPage.Master>**
6. **<local:MasterPage** x:Name="masterPage" **/>**
7. **</MasterDetailPage.Master>**
8. **<MasterDetailPage.Detail>**
9. **<NavigationPage>**
10. **<x:Arguments>**
11. **<local:ContactsPage** **/>**
12. **</x:Arguments>**
13. **</NavigationPage>**
14. **</MasterDetailPage.Detail>**
15. **</MasterDetailPage>**

# 1.4 NavigationPage

Administra la navegación entre otras paginas con una arquitectura basada en la pila.

## 1.4.1 Constructor

new NavigationPage (new [RootPage]());

## 1.4.2 Propiedades

|  |  |
| --- | --- |
| Property | Value |
| BarBackgroundColor | Define el color de la barra de navegacion |
| BarTextColor | Define el color del texto dentro de la barra de navegación. |

## 1.4.3 Metodos

|  |  |
| --- | --- |
| Method | Value |
| SetHasBackButton(Page, Boolean) | Define si el boton de "back" sera mostrado. |
| OnBackButtonPressed() | Cacha el evento lanzado al presionar el botón físico "atrás" |
| SetHasNavigationBar(BindableObjet, Boolean) | Define si se muestra o no la barra de navegación la pagina. |
| Nt | Define el icono de la pagina. |

# 1.5 TabbetPage

Un tabbetpage se deriva de una clase abstracta MultiPage que permite la navegación entre paginas secundarias con pestañas.

## 1.5.1 Propiedades

|  |  |
| --- | --- |
| Property | Value |
| Children | Obtiene la lista de las paginas que definen las pantallas visibles en el tabbetpage |
| CurrentPage | Define la pagina actual mostrada en el tabbetPage |
| ItemSource | Define la fuente de las paginas para mostrar. |
| SelectedItem | Es la pagina que está actualmente seleccionada |

## 1.5.2 XAML

1. **<TabbedPage** xmlns="http://xamarin.com/schemas/2014/forms"
2. xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
3. xmlns:local="clr-namespace:TabbedPageWithNavigationPage;assembly=TabbedPageWithNavigationPage"
4. x:Class="TabbedPageWithNavigationPage.MainPage"**>**
5. **<local:TodayPage** **/>**
6. **<NavigationPage** Title="Schedule" Icon="schedule.png"**>**
7. **<x:Arguments>**
8. **<local:SchedulePage** **/>**
9. **</x:Arguments>**
10. **</NavigationPage>**
11. **</TabbedPage>**

# 2.1 Layout

A screenshot of a cell phone

Description generated with high confidence

Son usadas para construir interfaces de usuario, dentro de ellas podremos albergar otros layouts y controles.