Desarrollo Móvil con Xamarin

2 - Introducción a Xamarin.Forms

Contenido

- ¿Qué es Xamarin.Forms?
- Componentes de Xamarin.Forms
  - Pages
  - Views
  - Layout
- Acceder a APIs de plataformas específicas

¿Qué es Xamarin.Forms?

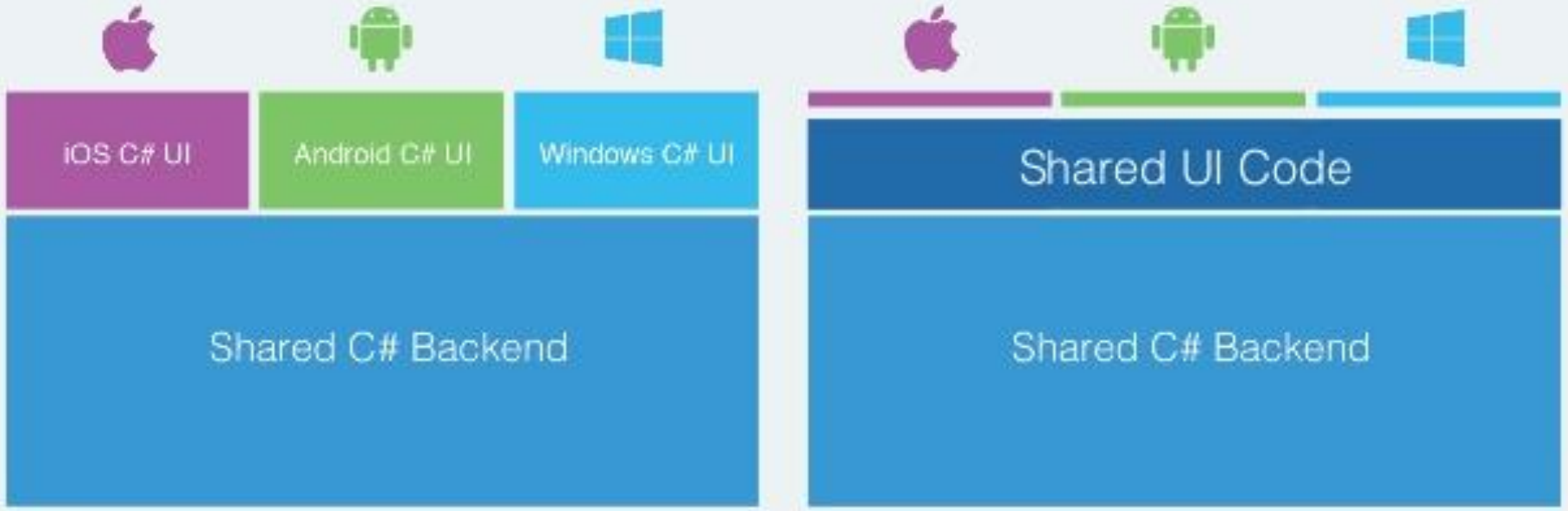
- En el enfoque tradicional de Xamarin, se comparte principalmente el código de la lógica de negocios y estructuras de datos.

- En el enfoque de Xamarin.Forms, además de compartir el código de la lógica de negocios y estructuras de datos, también se comparte el código de la interfaz del usuario (UI).

- Esto se logra al definir la UI utilizando controles que son independientes de las plataformas (Android, iOS, Windows).

Xamarin + Xamarin.Forms

Xamarin



- En tiempo de ejecución, cada plataforma transforma los controles definidos en Xamarin.Forms a controles nativos.
  - Xamarin.Forms Button:
    - Android → Widget Button
    - iOS → UI Button.
    - Windows → XAML Control Button

- Xamarin.Forms no siempre será el mejor enfoque para el desarrollo de una aplicación.

- Si lo que más importa en la aplicación es el **manejo de datos**, y el diseño de la interfaz no debe ser necesariamente "pixel-perfect", entonces Xamarin.Forms es el enfoque adecuado.

- Pero, si las especificaciones del **diseño personalizado de la interfaz** son fundamentales en la aplicación, entonces el enfoque tradicional (diseñar una interfaz para cada plataforma) es el adecuado.

- Ejercicio #1 – App de Xamarin.Forms

Componentes de Xamarin.Forms

- La clase **Application** es el punto de entrada de la aplicación.
  - Es utilizada por el código de las plataformas específicas para inicializar la aplicación.
  - Siempre va "apuntar" a la primera página de la aplicación.

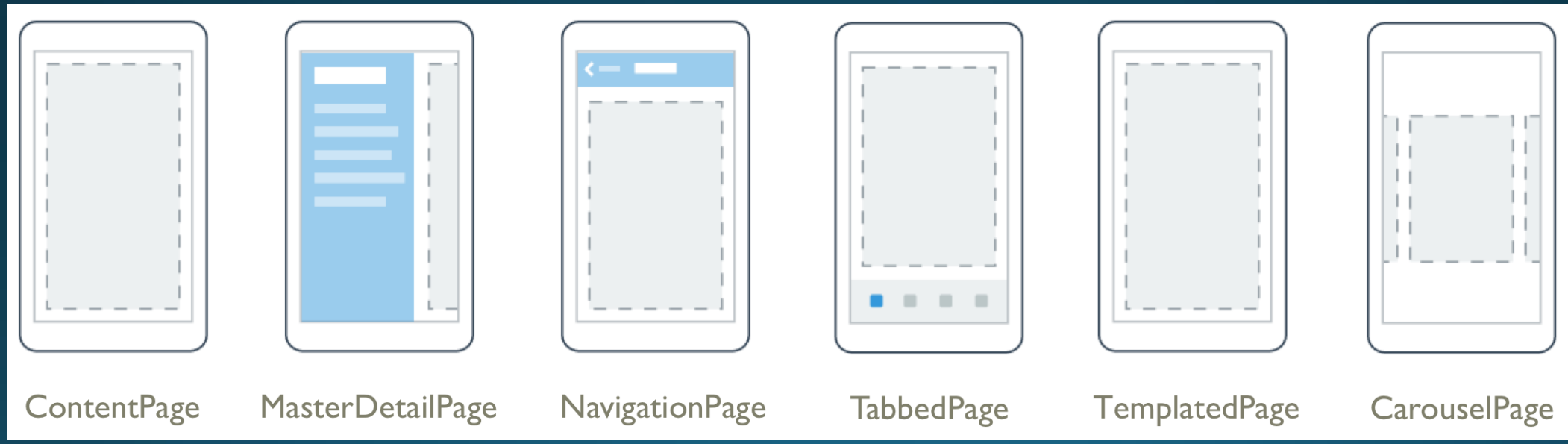
- La clase **Page** se utiliza para mostrar una pantalla al usuario.
  - Define la UI, ya sea en código (C#) o markup (XAML), y provee comportamiento para la pantalla.

Pages

- La clase **Page** define una pantalla en la aplicación.

- Un objeto **Page** representa en:
  - iOS: Un objeto **ViewController**.
  - Windows: Un objeto **Page**.
  - Android: Un funcionamiento similar a un objeto **Activity** (pero no es un objeto Activity).

- Varias clases derivan de **Page**, y cumplen diferentes propósitos.

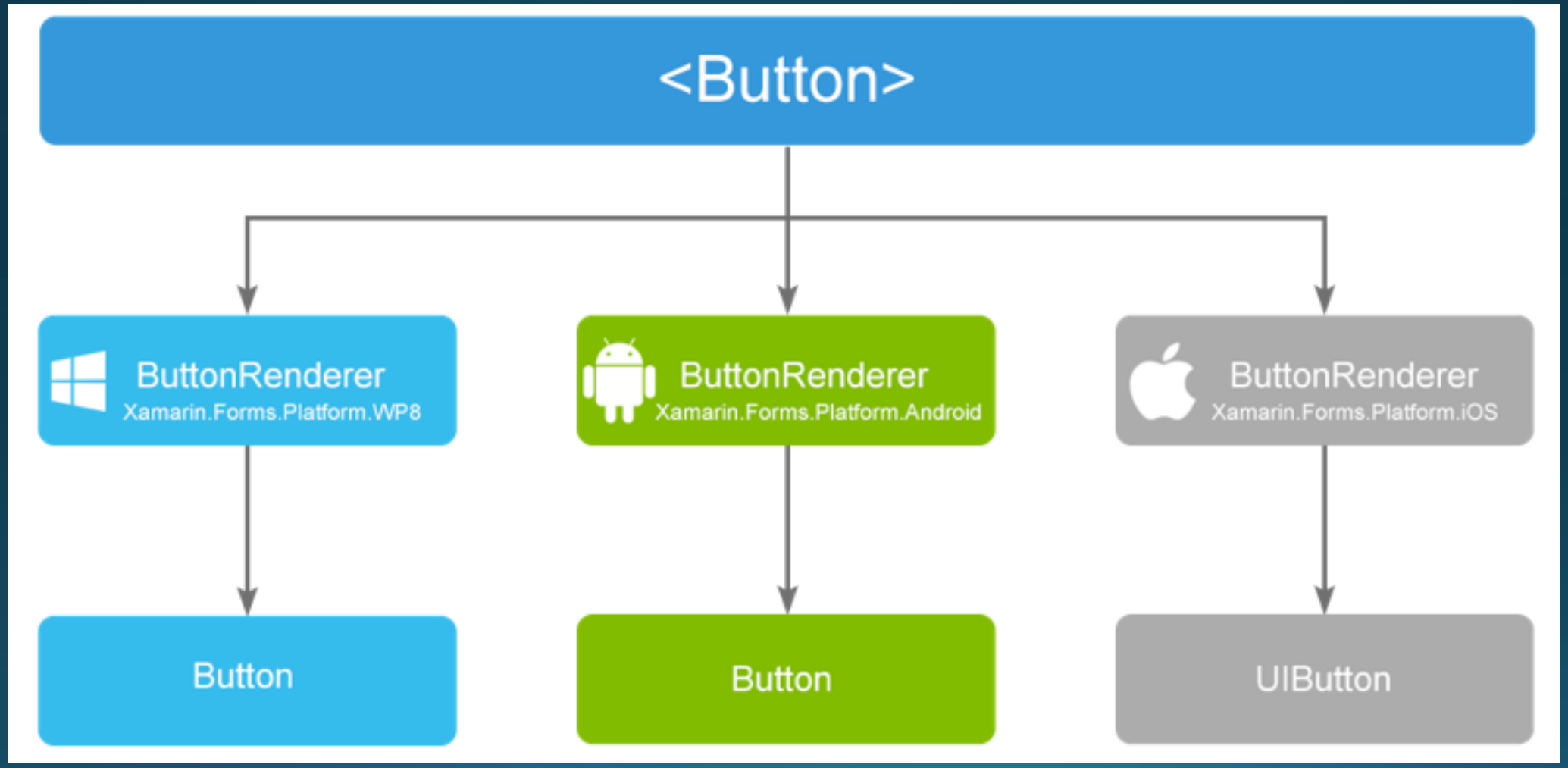


Views

- La clase **View** define los controles típicos que se tendrán en la aplicación.

- Varias clases derivan de **View**, por ejemplo:
  - Presentación: Label, Image, Map.
  - Para iniciar comandos (eventos): Button, SearchBar.
  - Para asignar valores: Slider, DatePicker, TimePicker.

- En tiempo de ejecución, cada plataforma convierte los controles definidos con la clase **View** a controles nativos.

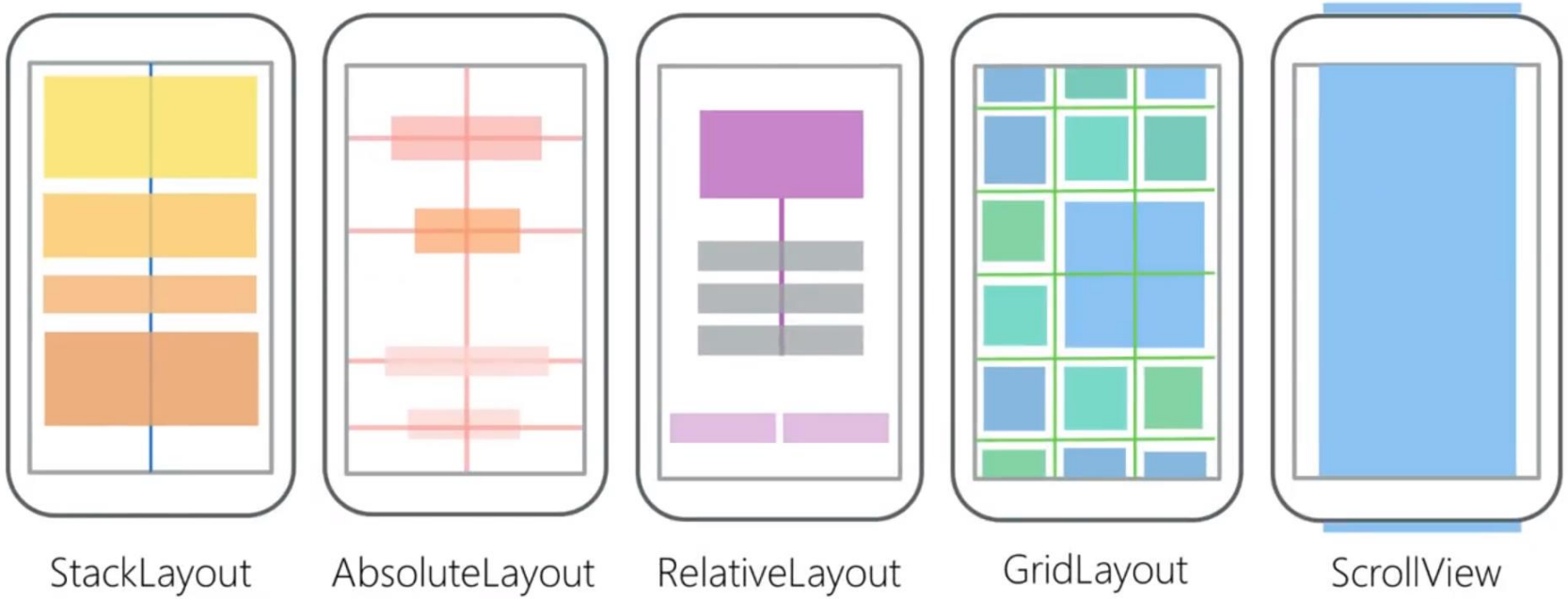

<Button>
ButtonRenderer
Xamarin.Forms.Platform.WP8
ButtonRenderer
Xamarin.Forms.Platform.Android
ButtonRenderer
Xamarin.Forms.Platform.iOS
Button
Button
UIButton

Layout

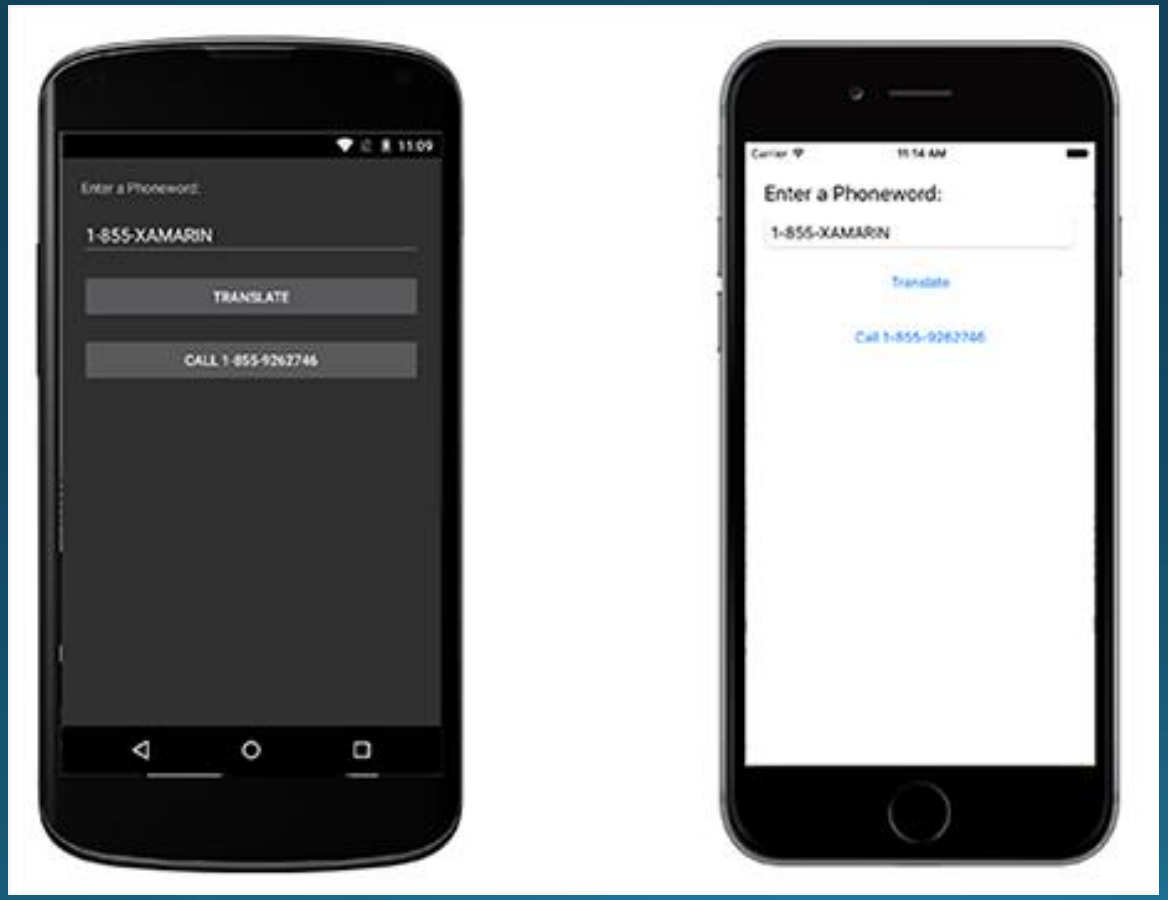
- Para crear múltiples controles (**Views**) en una pantalla, éstos deben estar definidos dentro de un contenedor.

- La clase **Layout** actúa como un contenedor tanto de controles (**Views**), como de otros contenedores (**Layouts**).

- Define el tamaño y la posición de sus elementos hijos.


StackLayout
AbsoluteLayout
RelativeLayout
GridLayout
ScrollView

- Ejercicio #2 – Phoneword



Acceder a APIs de plataformas específicas

- Si bien Xamarin.Forms provee APIs que "apuntan" a todas las plataformas, hay APIs que son específicas a las plataformas.

- En estos casos, lo ideal es utilizar las APIs específicas en cada plataforma pero, de alguna manera, invocarlas desde nuestro código compartido.

- Aquí, el uso de abstracciones (como las interfaces) es muy útil.