

# DESENVOLVIMENTO MOBILE COM XAMARIN

COPYRIGHT © RAFAEL PADILLA

[contato@rafaelpadilla.net](mailto:contato@rafaelpadilla.net)

# RAFAEL PADILLA

## **Doutorando em Engenharia Elétrica**

- Universidade Federal do Rio de Janeiro (COPPE/UFRJ)
- Deep Learning
- Inteligência Artificial
- Processamento Digital de Imagens

## **Mestrado em Engenharia Elétrica pela UFAM**

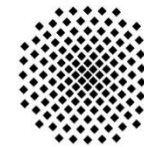
- Reconhecimento de Padrões
- Reconhecimento Facial
- Processamento Digital de Imagens

## **Graduação em Engenharia Elétrica**

- Universidade Federal do Amazonas (UFAM)
- Instituto de Automação Industrial e Engenharia de Software (Universidade Stuttgart)



UFRJ



Universität  
Stuttgart



# RAFAEL PADILLA

## **Microsoft / Instituto Nokia de Tecnologia**

- Desenvolvedor C#
- Evangelista Windows Phone
- Pesquisador ( Visão Computacional / Imaging)

## **Pace**

- Engenheiro Industrial / Engenheiro de Teste

## **Adutech**

- Desenvolvedor Sênior C#

## **Instituto de Automação Industrial e Engenharia de Software**

- Desenvolvedor C#





# RAFAEL PADILLA



[contato@rafaelpadilla.net](mailto:contato@rafaelpadilla.net)



[www.linkedin.com/in/rafael-padilla](https://www.linkedin.com/in/rafael-padilla)



<https://github.com/rafaelpadilla>

# CRONOGRAMA



**Pós-Graduação Lato Sensu:** Desenvolvimento de Software para Dispositivos Móveis

**Módulo:** Introdução à Programação com C#

**Local:** Universidade Católica do Tocantins- TO

**Carga horária:** 30 horas

**Datas:**

Parte 1: 04 e 05 de novembro de 2017

Parte 2: 18 e 19 de novembro de 2017

SÁBADO (04/11 e 18/11)	DOMINGO (05/11 e 19/11)
08:00 às 12:00h	08:00 às 12:00h
14:00 às 19:00h	-



# Xamarin



# E M E N T A X A M A R I N

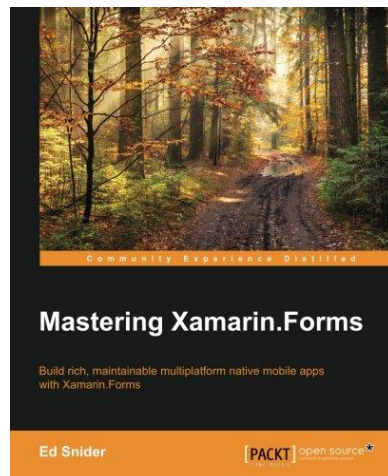
- XAML
- LayoutPadding
- LayoutMargin
- StackLayout
- ScrollView
- Grid
- AbsoluteLayout
- LayoutOptions
- Bindings
- Animations
- Navegando entre Páginas
- Acceso à camera
- GPS
- IoC container



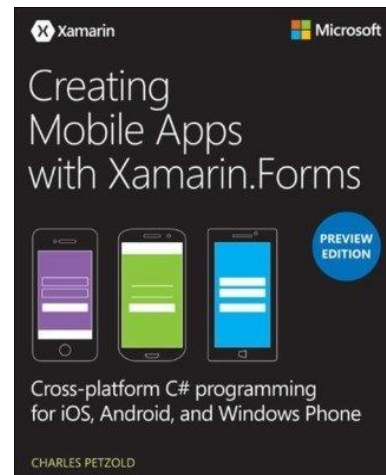
# REFERÊNCIAS BIBLIOGRÁFICAS



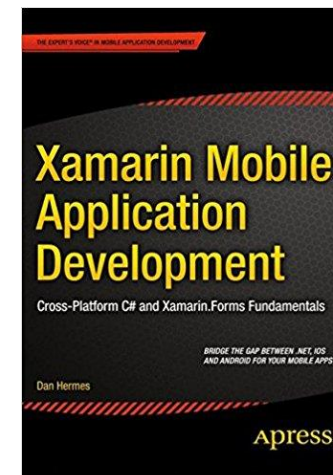
DEITEL, H. M.  
**C# Como Programar.**  
Pearson.  
1ª Edição.  
2003.



ED SNIDER.  
**Mastering  
Xamarin.Forms**  
1a Edição, 2016



PETZOLD ,CHARLES  
**Creating Mobile  
Apps with  
Xamarin.Forms**  
2a edição, 2015



DAN HERMES.  
**Xamarin Mobile Application Development:  
Cross-Platform C# and Xamarin.Forms  
Fundamentals**  
1a Edição, 2015

# AVALIAÇÕES

A avaliação dos alunos será feita através da execução **de 2 projetos**. Para cada projeto será dada uma nota de 0 a 10. A nota final será a média dos dois os projetos.

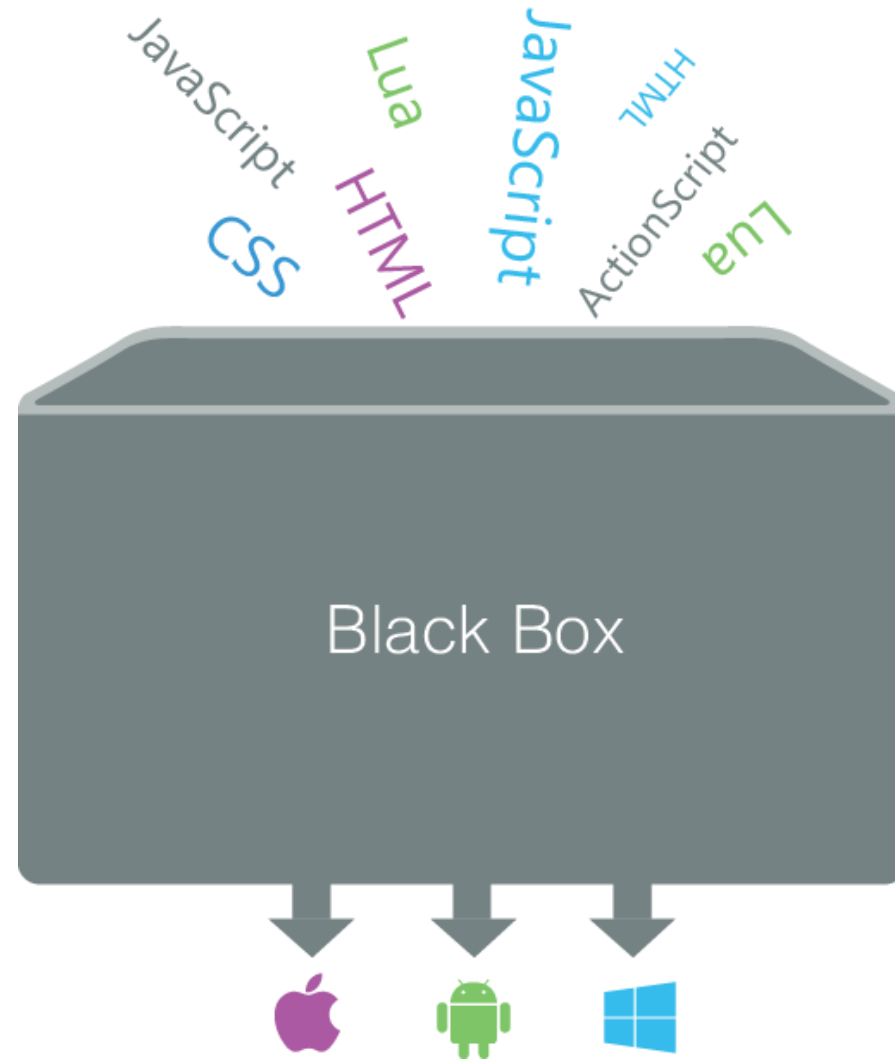
Os pontos avaliados em cada projeto serão:

- Projeto funcional com tratamento de exceções;
- Projeto consistente com as instruções;
- Código comentado;
- Funcionalidades e UI responsivas para as 3 plataformas (iOS, Android e Windows Phone)

Desafio 1 - Calculadora

Desafio 2 – Projeto Livre

# TIPOS DE DESENVOLVIMENTO



# TIPOS DE DESENVOLVIMENTO

## Nativo

O desenvolvimento nativo é o modo **tradicional** para desenvolvimento de aplicações mobile. Neste tipo, a equipe de desenvolvimento cria **um aplicativo para cada sistema operacional** no qual o app será lançado. Se uma aplicação vai ser lançada em Android, iOS e Windows OS, então serão três aplicativos diferentes.

## Multiplataforma

Também chamada de abordagem *cross-platform*, a criação de um aplicativo é feita por meio de um **único processo de desenvolvimento**, onde o resultado final serão apps lançados em plataformas diferentes (Android e iOS, por exemplo).

## Híbrido (web apps)

Estes aplicativos são desenvolvidos utilizando tecnologias web como HTML5 e CSS3 (as mesmas utilizadas para desenvolvimento de web sites).

# DESENVOLVIMENTO NATIVO

## **Vantagens:**

- Experiência nativa (dev)
- Usabilidade nativa (user)
- Desenvolvimento com SDKs e IDE do próprio sistema operacional
- Acesso total ao device fornecido pelo SDK

## **Desvantagens:**

- Pouca reutilização de código para outra(s) plataforma(s)
- Mão de obra especializada
- Custo alto
- Tempo de desenvolvimento alto
- Manutenibilidade prejudicada

# DESENVOLVIMENTO MULTIPLATAFORMA

## Vantagens:

- Compartilhamento de código
- Equipe menor
- Desenvolvimento rápido
- Redução de prazos e custo
- Manutenibilidade
- Usabilidade nativa (user)
- Desempenho idêntico

## Desvantagens:

- Desempenho prejudicado
- Componentes UI não nativos
- Limitações no uso de APIs nativas

# DESENVOLVIMENTO WEB APP

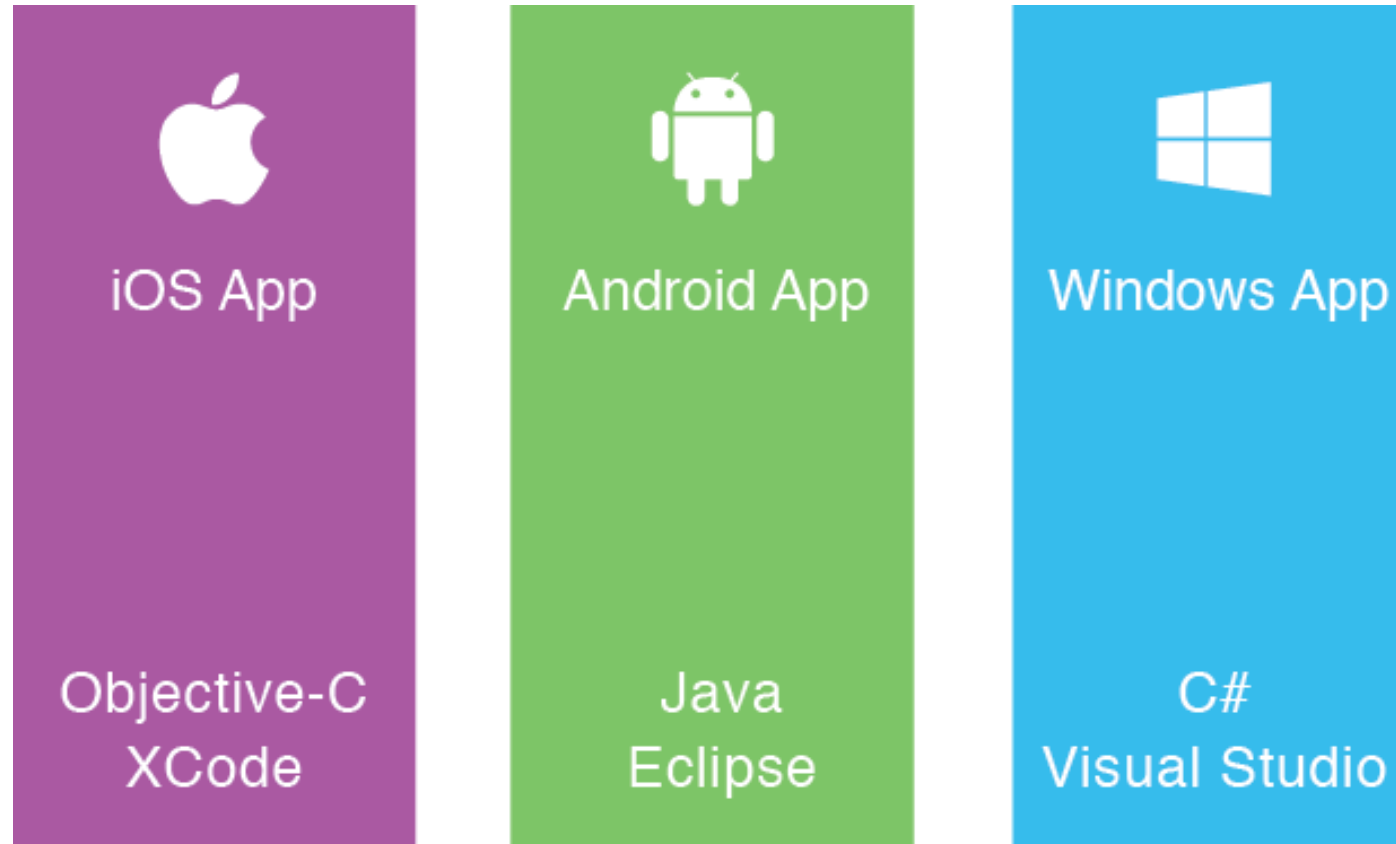
## **Vantagens:**

- Experiência única independente da plataforma
- Custo baixo
- Manutenibilidade
- Não é necessário estar nas lojas
- Não necessário dividir lucro com as detentoras das plataformas

## **Desvantagens:**

- Experiência web não nativa
- Teste em diferentes plataformas e versões
- Ajustes para cada plataforma e versão

# DESENVOLVIMENTO NATIVO





# FERRAMENTAS MULTIPLATAFORMAS



# BRAND LOVERS



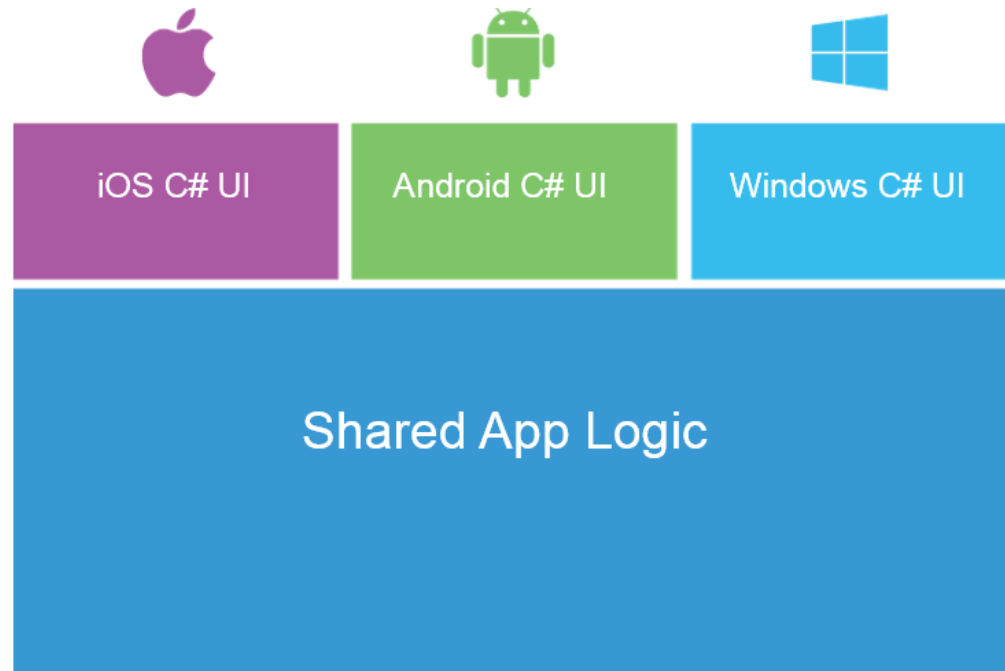
# XAMARIN



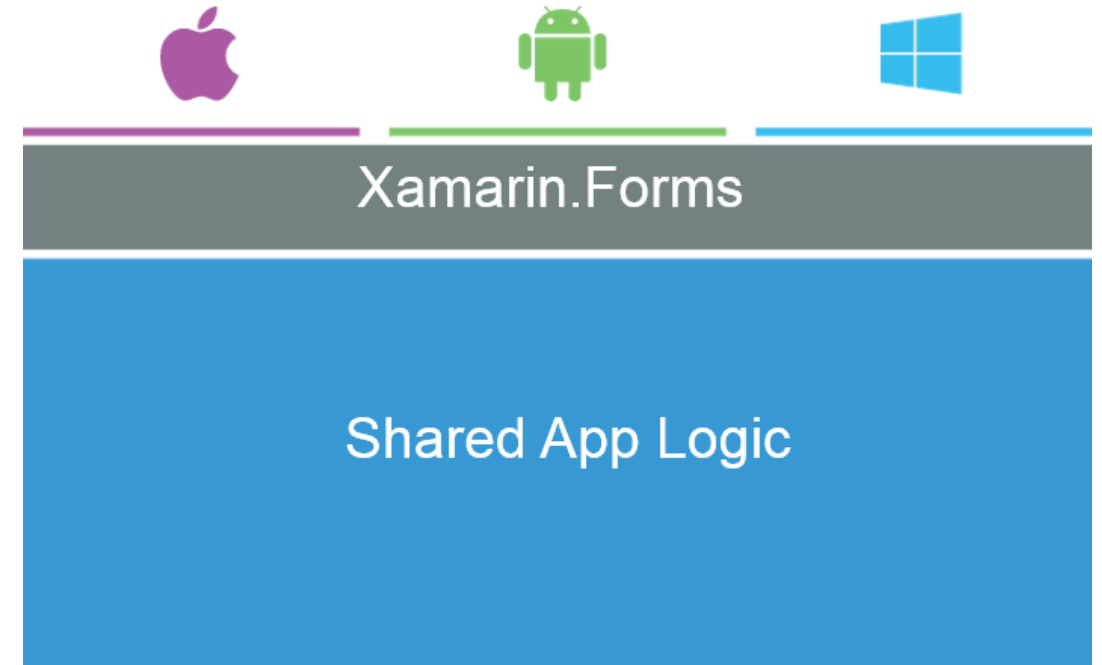
Tamarin

- Criada em maio de 2011 pelos criadores do projeto Mono (2001) – Miguel de Icaza
- Julho de 2011 os direitos do “Mono for Android” foram cedidos pela Novell para para o Xamarin
- 2012: Lançamento do Xamarin.Mac (C# para Apple OS X)
- 2012: Xamarin Studio e integração com Visual Studio
- Julho de 2013 anunciaram ter fundos de 82M USD
- Fevereiro de 2016 Microsoft anuncia a compra por 400M ~ 500M USD
- No Build 2016 Microsoft anuncia que o Xamarin SDK será open-source

# ABORDAGENS XAMARIN

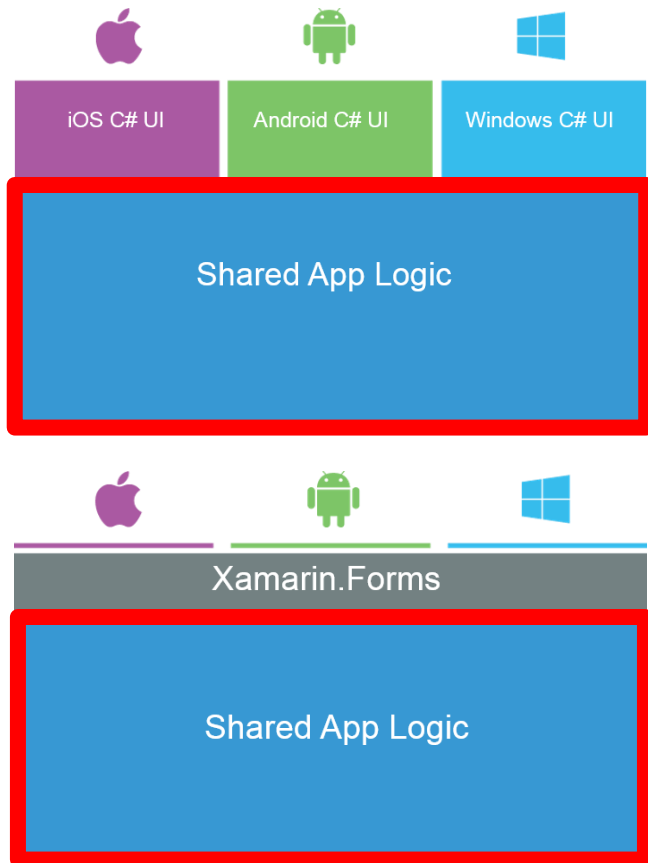


Xamarin.iOS & Xamarin.Android  
60 a 80% de reutilização do código



Xamarin.Forms  
Até 99% de reutilização do código

# FERRAMENTAS



O que colocar no Projeto Compartilhado?

- Regras de negócio
- *Parsing*
- Acesso a webservices
- Métodos que não dependem exclusivamente da plataforma

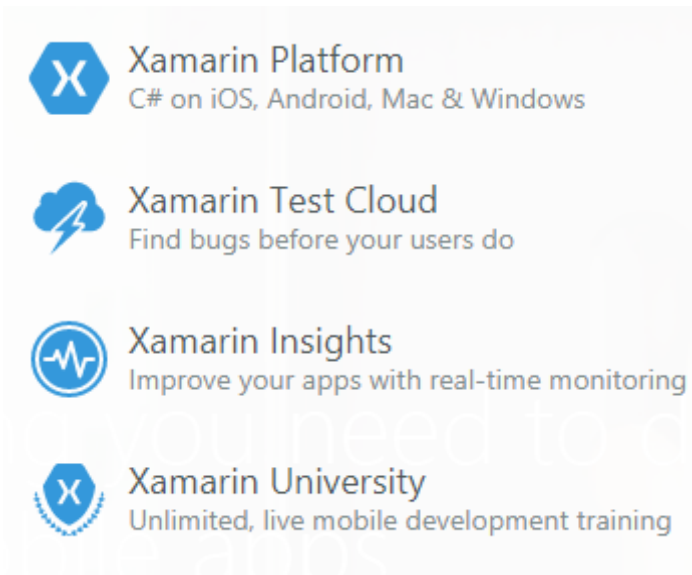
# REQUERIMENTOS



	MACOS	WINDOWS
Development Environment	VISUAL STUDIO FOR MAC	VISUAL STUDIO
Xamarin.iOS	Yes	Yes (with Mac computer)
Xamarin.Android	Yes	Yes
Xamarin.Forms	iOS & Android only (macOS in preview)	Android, Windows/UWP (iOS with Mac computer)
Xamarin.Mac	Yes	Open project & compile only <a href="#">^</a>

**Visual Studio for Mac** is replacing **Xamarin Studio**.

# FERRAMENTAS



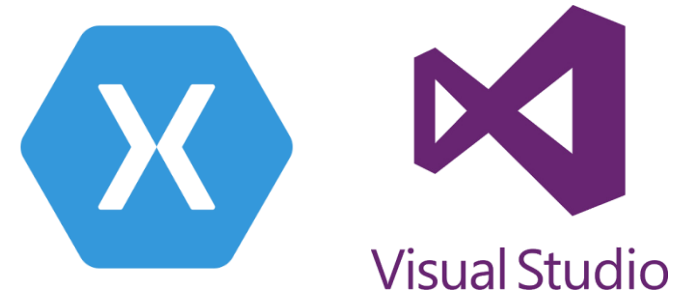
**Visual Studio Community 2017**  
<https://www.xamarin.com/download>

# FERRAMENTAS

Xamarin está disponível nas seguintes versões do Visual Studio:

- Non-express editions Visual Studio 2013 Update 2
- Visual Studio 2015 Community, Professional e Enterprise
- Visual Studio 2017 Community, Professional e Enterprise

Para desenvolver apps com Xamarin.Forms para Universal Windows Platform (UWP) é necessário no mínimo Windows 10 com Visual Studio 2015



**Visual Studio Community 2017**  
<https://www.xamarin.com/download>



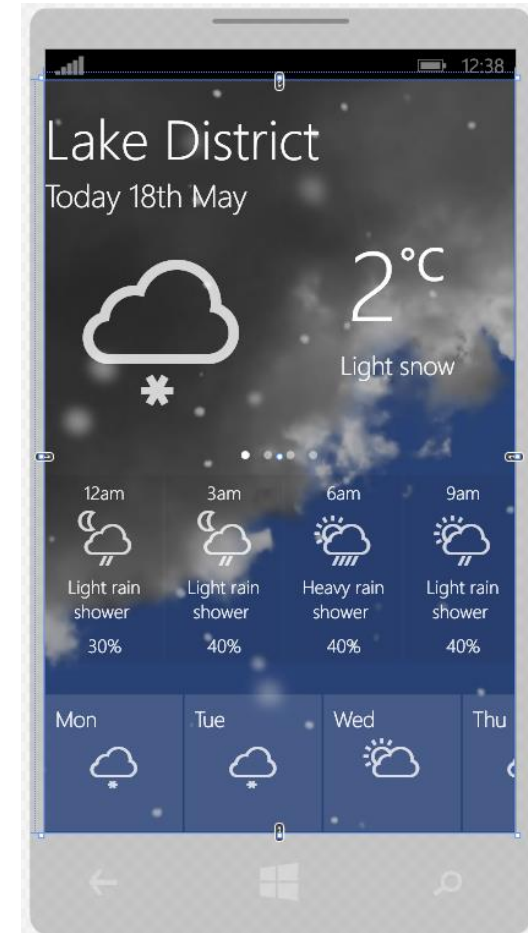
# XAML

## Extensible Application Markup Language

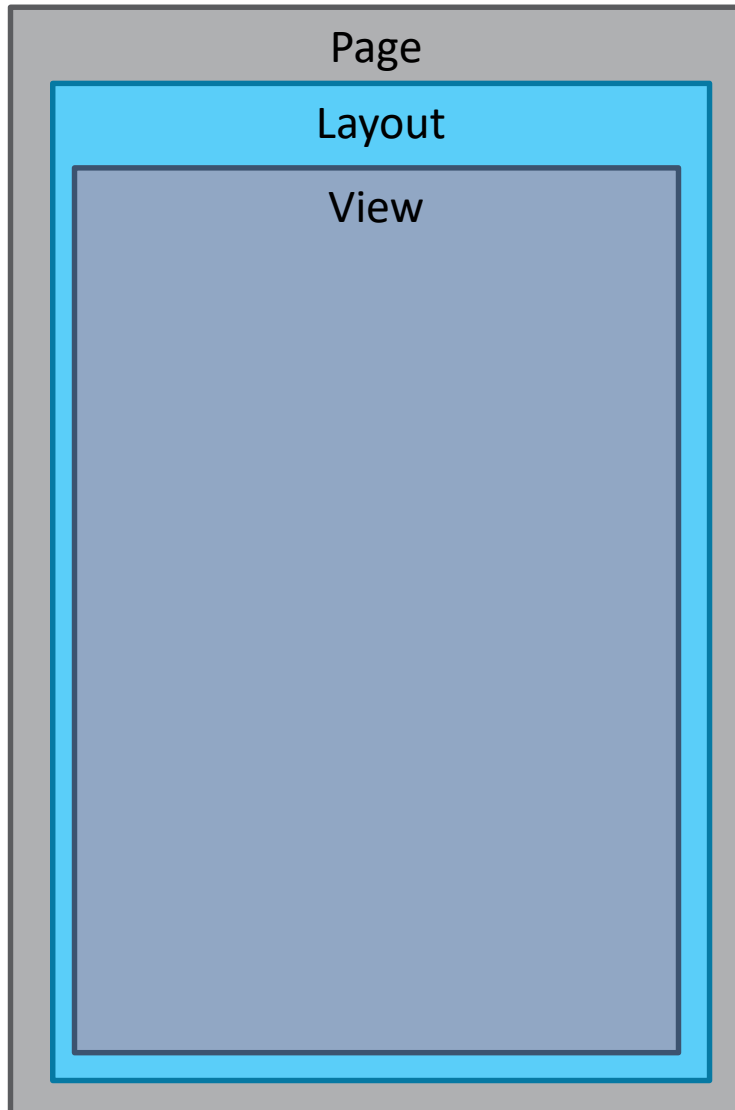
```
xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:designData="clr-namespace:Silkweb.Mobile.MountainWeather.DesignData;assembly=Silkweb.Mobile.Mount
xmlns:converters="clr-namespace:Silkweb.Mobile.MountainWeather.WinPhone.Converters"
SupportedOrientations="Portrait" Orientation="Portrait"
mc:Ignorable="d"
shell:SystemTray.IsVisible="True"
d:DataContext="{d:DesignInstance Type=designData:SampleData, IsDesignTimeCreatable=True}"
Title="{Binding Title}" Margin="-1,0,1,0">

<Grid>
  <Grid DataContext="{Binding Items[0]}">
    <Image Stretch="UniformToFill" Source="../Assets/Backgrounds/wsymbol_0011_light_snow_showers.jpg"
    <Grid Background="#33000000" />
  </Grid>

  <Grid Style="{StaticResource pageStyle}">
    <Grid.Resources>
      <ResourceDictionary>
        <DataTemplate x:Key="periodTemplate">
          <Grid Style="{StaticResource periodOuterStyle}" VerticalAlignment="Top">
            <Grid Background="{StaticResource periodColor}">
              <StackPanel Style="{StaticResource periodStyle}">
                <TextBlock Text="{Binding Period}" Style="{StaticResource periodTextStyle}"
                <TextBlock Text="{Binding WeatherCode.Glyph}" Style="{StaticResource s
                <Grid Style="{StaticResource periodDescriptionStyle}">
                  <TextBlock Text="{Binding WeatherCode.Description}" Style="{StaticR
                </Grid>
                <TextBlock Text="{Binding Probability}" Style="{StaticResource periodTe
              </StackPanel>
            </Grid>
          </Grid>
        </DataTemplate>
```



# ESTRUTURA DO XAMARIN FORM



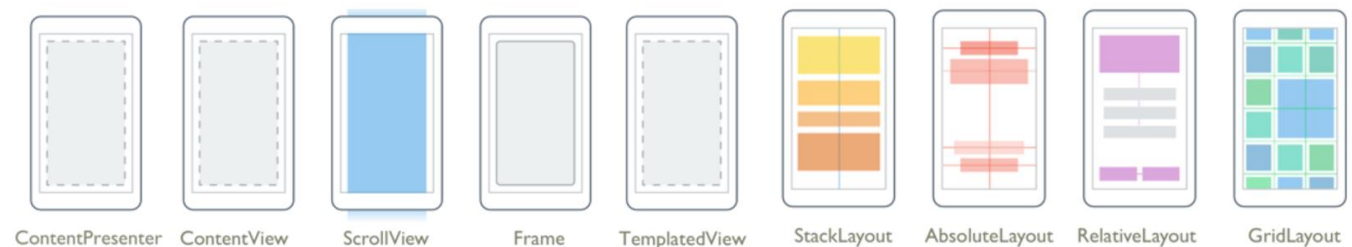
## Page

Toda área visível na tela. Contém apenas **um único** filho.



## Layout

É um container para outros Layouts ou Views. Os componentes são organizados com um Layout.

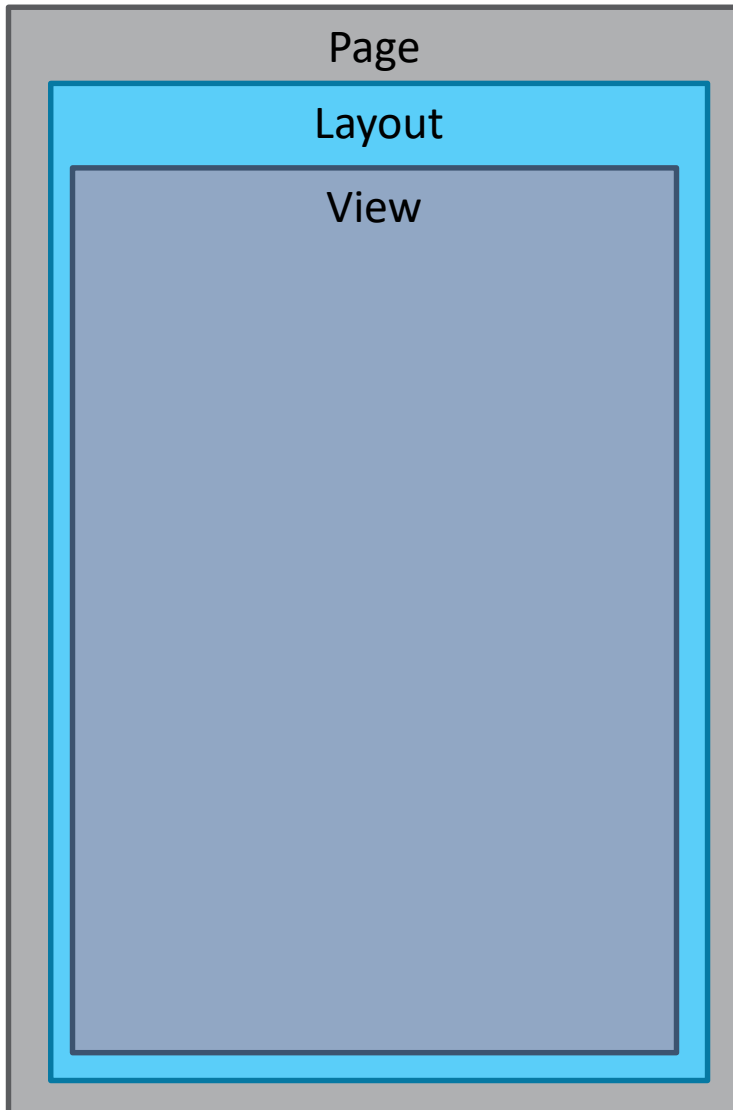


## View

São os objetos visuais (elementos de UI como buttons, labels, text entry boxes, etc).



# ESTRUTURA DO XAMARIN FORM



## Page

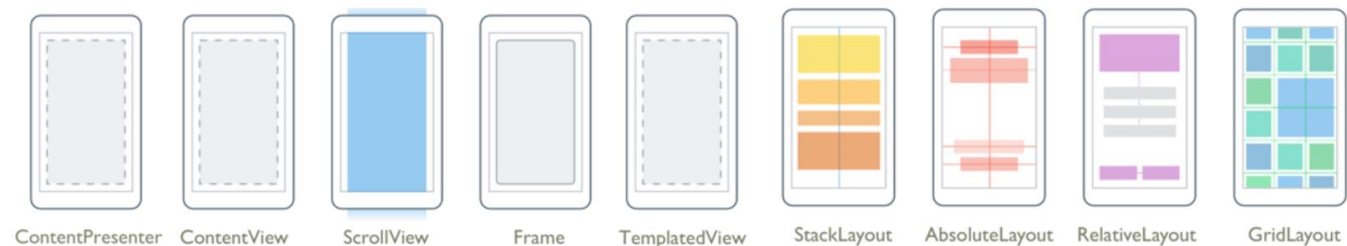
Toda área visível na tela. Contém apenas **um único** filho.



No projeto [Code 01] identifique na Page01.xaml cada um destes elementos!

## Layout

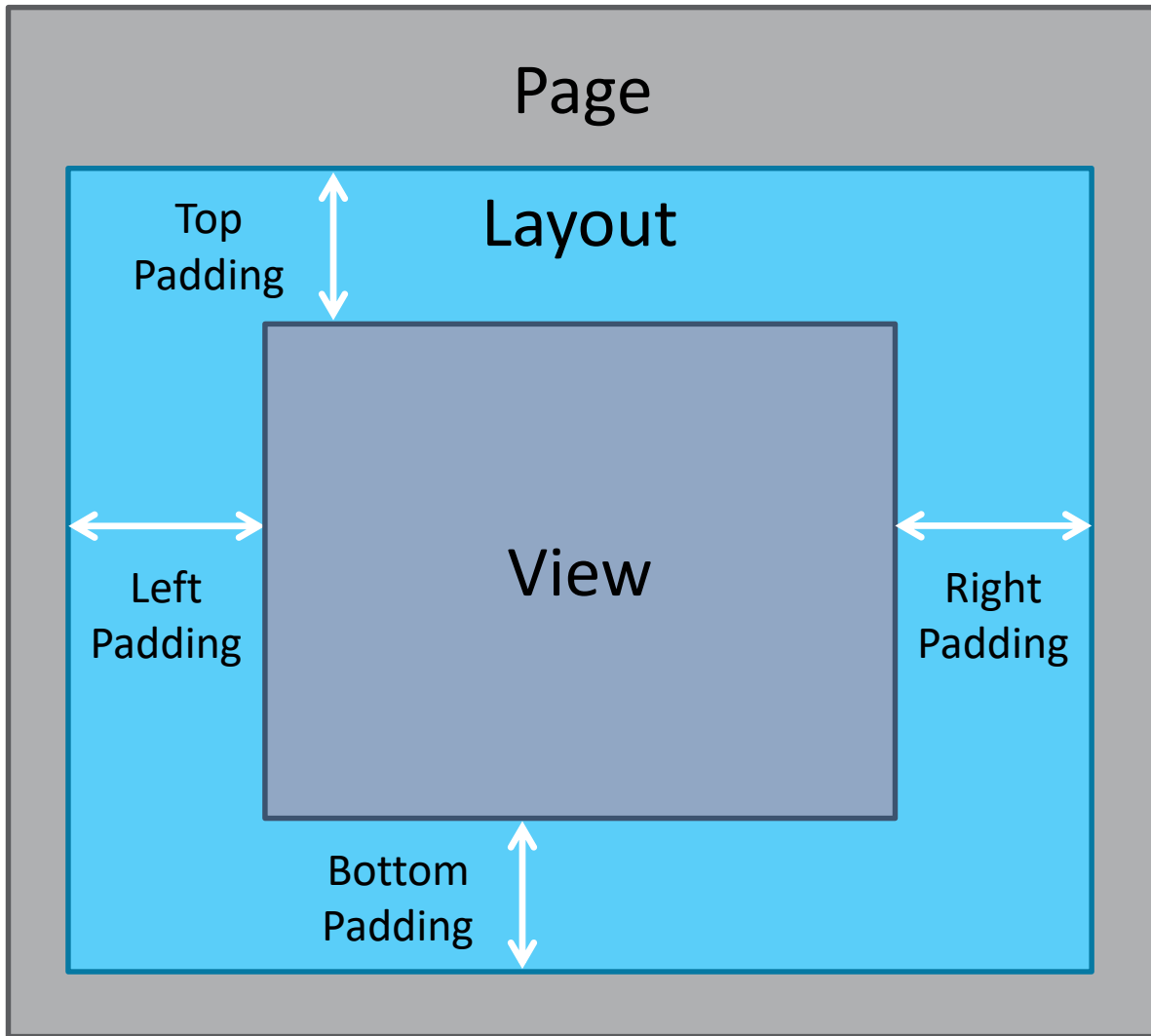
É um container para outros Layouts ou Views. Os componentes são organizados com um Layout.



## View

São os objetos visuais (elementos de UI como buttons, labels, text entry boxes, etc).

# LAYOUT PADDING



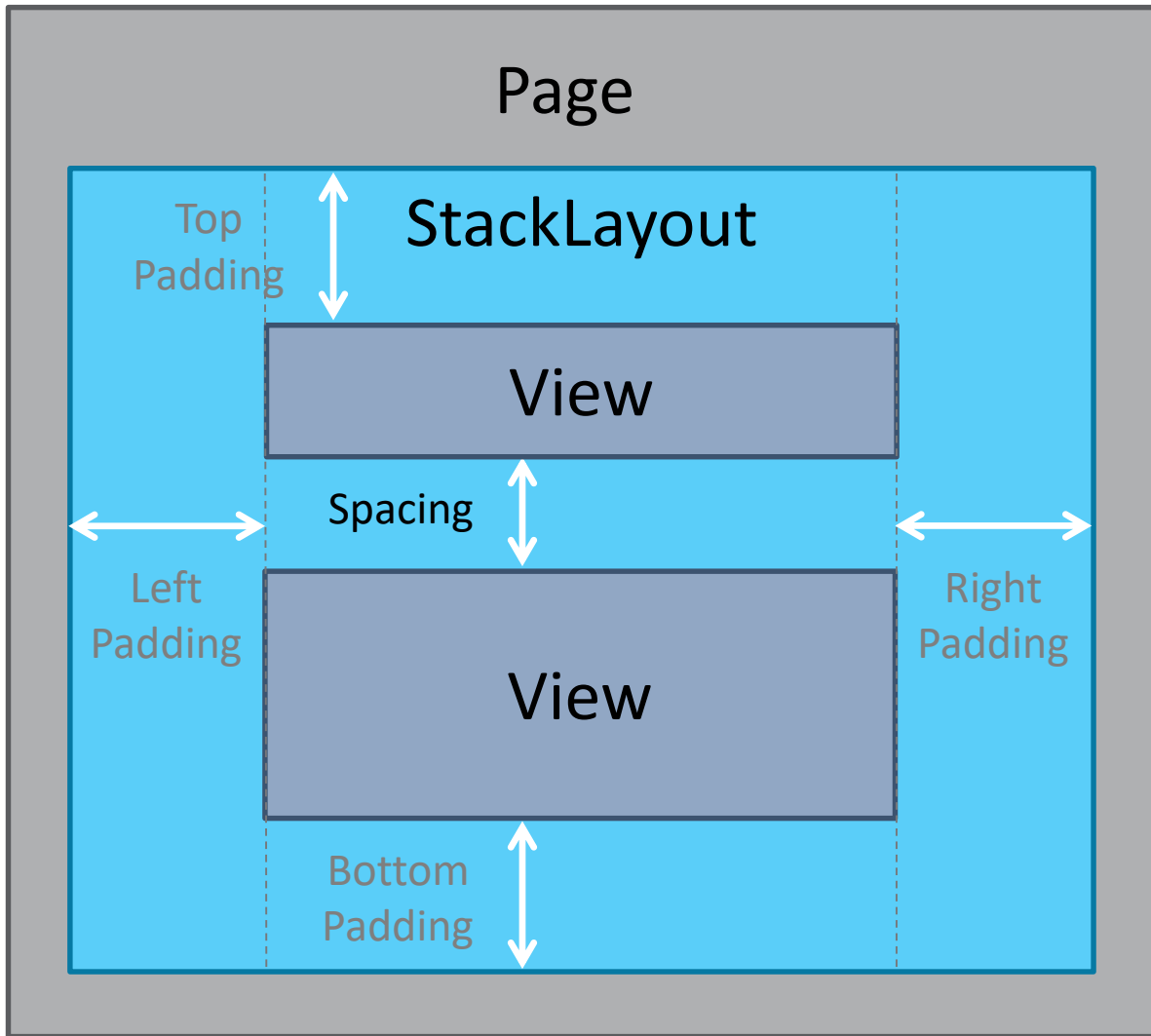
A propriedade **Padding** representa a distância entre um elemento e seus **elementos filhos**.

Padding = “left, top, right, bottom”

➡ Padding aplicado no Layout

**CÓDIGO**

# LAYOUT PADDING



A propriedade **Padding** representa a distância entre um elemento e seus **elementos filhos**.

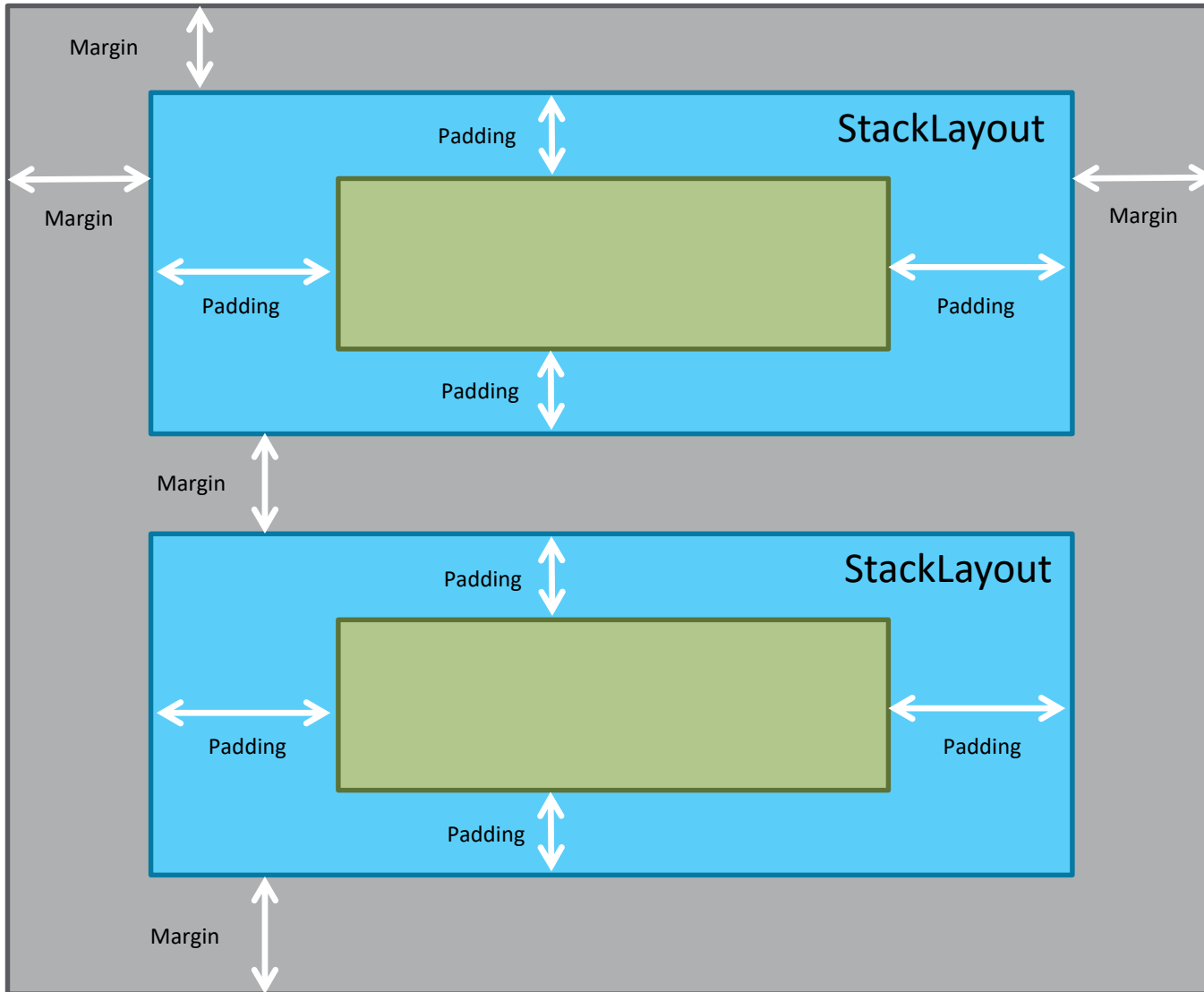
Padding = "left, top, right, bottom"

A propriedade **Spacing** representa a distância entre os elementos do StackLayout.

Spacing = "15"

**CÓDIGO**

# LAYOUT MARGIN



A propriedade **Margin** representa a distância entre um elemento e seus elementos adjacentes.

Margin = "left, top, right, bottom"

Margin = "valor"

Left = valor

Top = valor

Right = valor

Bottom = valor

Margin = "valor1, valor2"

Left = valor1

Right = valor1

Bottom = valor2

Top = valor2 ??????



# HANDS-ON (PARTE 1)

- StackLayout
- ScrollView
- Grid





# DESAFIO #1 (CALCULADORA)





# DESAFIO #1 (CALCULADORA)



Crie um projeto Xamarin.Forms e desenvolva um aplicativo seguindo as instruções abaixo:

1) Crie a UI de uma calculadora utilizando uma **Grid** para posicionar os componentes.

2) Crie os seguintes eventos:

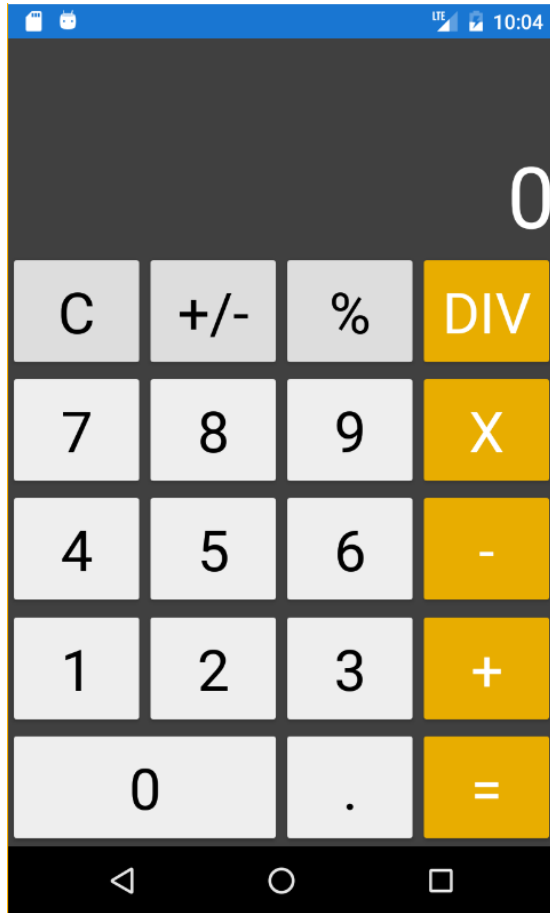
- a. Eventos para clique dos botões numéricos
- b. Eventos para clique dos botões de operações
- c. Evento para clique do botão de limpar o visor ("C")
- d. Evento para clique do botão "+/-"

3) As calculadoras possuem uma interação com o usuário de maneira particular. Por exemplo, a calculadora do emulador Android-23 x 86 trabalha com sequências de cálculos diferentemente da calculadora do Windows.

Neste desafio o comportamento da sua calculadora deve ser o mesmo da calculadora nativa do Windows e do iPhone 6.

# DESAFIO #1 (CALCULADORA)

4) Considere os seguintes casos de uso:



CT 1: 1 2 + 3 = 15 1 = 4 = 7 = 10

CT 2: 1 2 + 3 = 15 = 18 = 21

CT 3: 1 2 - 5 +/- = 17

CT 4: 1 2 + = 24 = 36 = 48

CT 5: 1 2 + = 24 + 1 = 25 = 26 3 = 4 = 5

CT 6: 1 2 = 12 3 3 = 33

CT 7: . 2 5 x . 2 5 = 0.0625 +/- = -0.015625

CT 8: . 2 5 x . 2 5 = 0.0625 +/- 1 = 0.25



Teclas pressionadas



Resultado no visor



[contato@rafaelpadilla.net](mailto:contato@rafaelpadilla.net)



[www.linkedin.com/in/rafael-padilla](https://www.linkedin.com/in/rafael-padilla)



<https://github.com/rafaelpadilla>

It's just the beginning