

# Desenvolvimento para Dispositivos Móveis

Gerenciadores de Layout

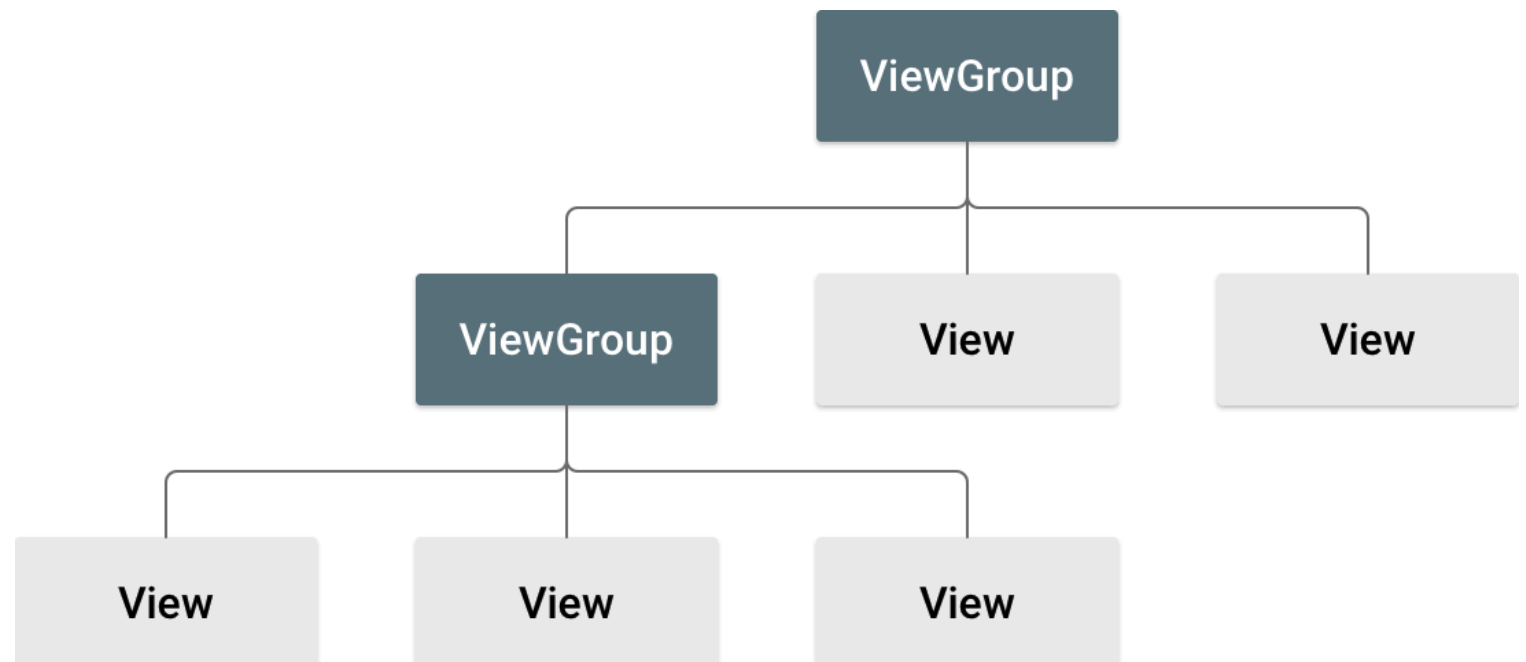
# Sumário

- Introdução
- Programação XML
- Tipos de layout
  - LinearLayout
  - RelativeLayout

# Introdução

# Introdução

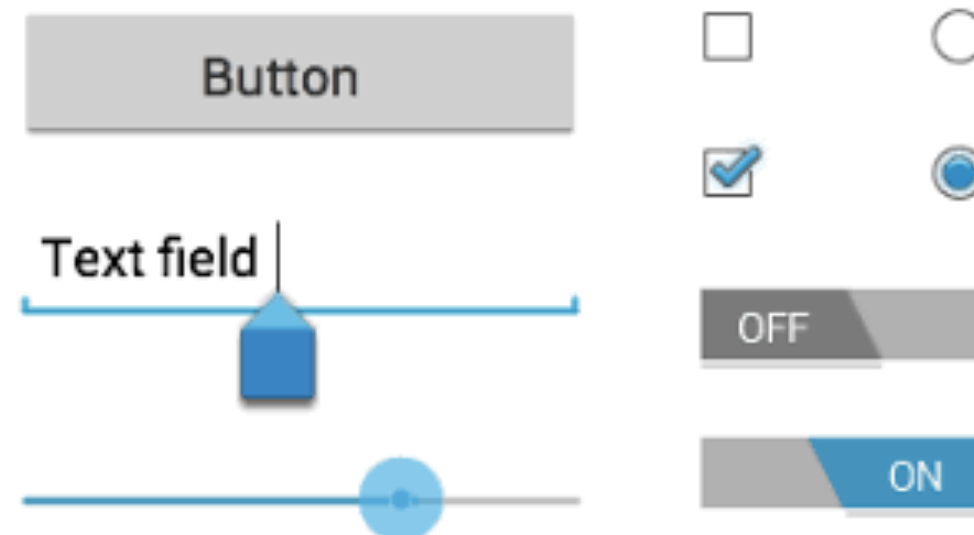
- Estrutura de uma interface do usuário no aplicativo
- Todos os elementos são criados usando uma hierarquia
  - View
  - ViewGroup



# Introdução

## *View*

- Geralmente chamados de *widgets*
  - TextView
  - Button



# Introdução

## *ViewGroup*

- Geralmente chamados de *layouts*
  - LinearLayout
  - RelativeLayout



# Introdução

- Um **layout** pode ser declarado de **duas** maneiras:
  - Declaração em **XML**
  - **Instanciação** do layout em ambiente de execução
- O ideal é a declaração dos layouts nos **arquivos XML**
  - Separa a parte visual do comportamento
  - Criação de layouts para diferentes tamanhos de telas

# Programação XML



# Programação XML

- **Arquivo** deve conter exatamente **um** elemento na **raiz**
  - View
  - ViewGroup

# Programação XML



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Olá, sou um TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Olá, sou um Button" />
</LinearLayout>
```

# Atributos

*layout\_width e layout\_height*

- Todo componente deve possuir:
  - **layout\_width**: largura do componente
  - **layout\_height**: altura do componente

# Atributos

*layout\_width e layout\_height*

- Os **valores** possíveis são:
  - Valor absoluto em **dp** (não recomendado)
  - **wrap\_content**: redimensiona de acordo com o conteúdo
  - **match\_parent**: redimensiona de acordo com o pai

# Programação XML



**<TextView**

```
    android:id="@+id/text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Olá, sou um TextView" />
```

**<Button**

```
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Olá, sou um Button" />
```

# Atributos

*id*

- Código que **identifica** o **componente**
- É usado para:
  - Acesso via **código**
  - **Posicionamento** relativo
- Convenção:
  - `atividade_tipo_descrição`

# Atributos

## *padding*

- Espaçamento **interno**



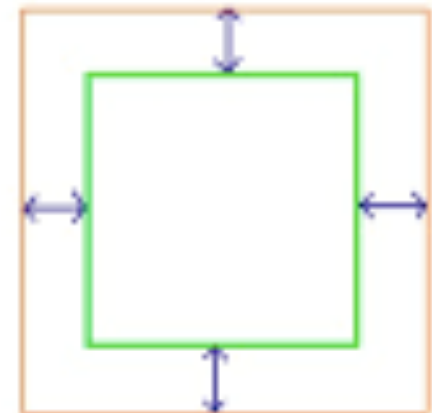
```
android:padding="20dp"
```

```
android:paddingStart="20dp"
```

```
android:paddingEnd="20dp"
```

```
android:paddingTop="20dp"
```

```
android:paddingBottom="20dp"
```



**Padding**

# Atributos

## *margin*

- Espaçamento **externo**



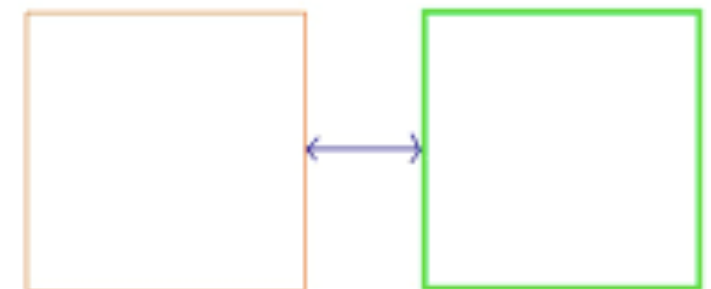
```
android:margin="20dp"
```

```
android:marginStart="20dp"
```

```
android:marginEnd="20dp"
```

```
android:marginTop="20dp"
```

```
android:marginBottom="20dp"
```



**Margin**



# Recursos

- O acesso a **recursos** através dos arquivos XML
  - @recurso/identificador



```
<TextView  
    android:id="@+id/text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:margin="@dimen/margem"  
    android:background="@color/azul"  
    android:text="@string/mensagem" />
```

# Tipos de layout

# Tipos de layout

- LinearLayout
- FrameLayout
- RelativeLayout
- TableLayout
- GridLayout
- ConstraintLayout

# LinearLayout

- Componentes são empilhados
- Orientação vertical ou horizontal



# RelativeLayout

- Posições relativas
- Construção de layouts complexos



# Referências bibliográficas



# Referência Bibliográfica

- **Declaração de layouts:** <https://developer.android.com/guide/topics/ui/declaring-layout>