

Desenvolvimento Para Dispositivos móveis



Prof. Me. Clênio Silva
E-mail: clenio.silva@uniube.br

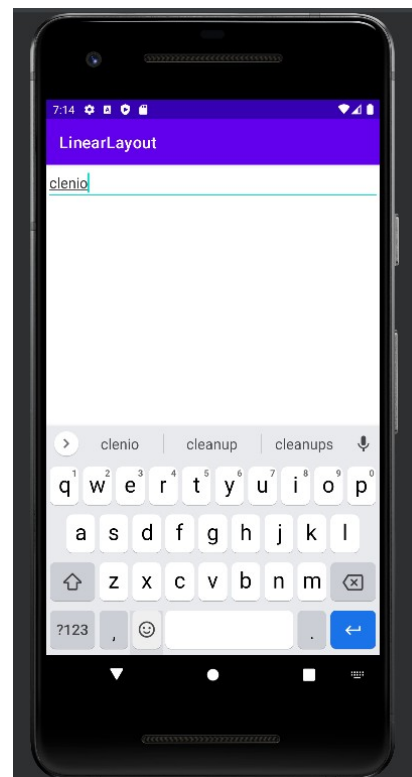
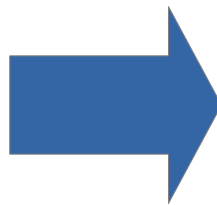
EditText

- Elemento que permite capturar entrada “input” do usuário;

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/r
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```



EditText

- O elemento **EditText** permite que possa ser especificado qual o tipo da informação que o usuário vai informar;
- Nas aplicações atuais podem ser passados número de telefone, número comum, e-mail e senha;
- Para setarmos qual o tipo de informação o **EditText** vai receber é necessário setar a propriedade:
 - **android:inputType**

EditText

- Vamos adicionar o seguinte trecho de código dentro do `LinearLayout`:

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="phone"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="number"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"/>

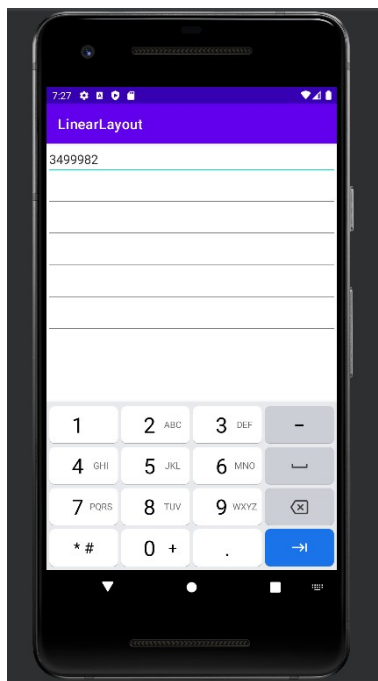
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textAutoCorrect"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiline"/>
```

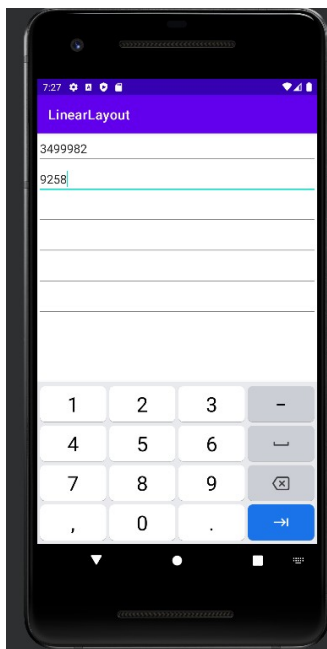
EditText

Ao setar informações no primeiro campo de entrada podemos perceber que o teclado é numérico. Isso se dá pelo fato que no primeiro **EditText** esta setado com a propriedade phone: `android:inputType="phone"`



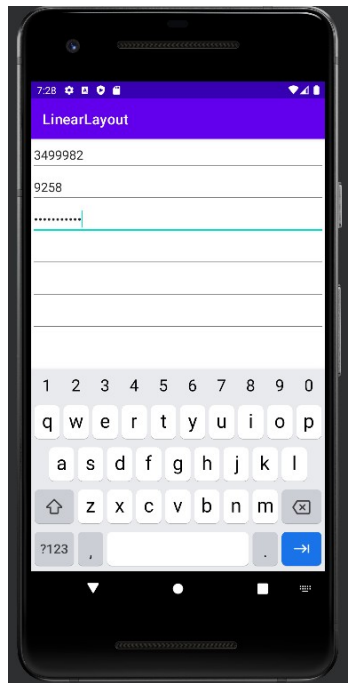
EditText

Ao setar informações no segundo campo de entrada podemos perceber que o teclado é numérico, contudo é diferente do teclado que apareceu ao setar o primeiro campo. Isso se dá pelo fato que no segundo `EditText` esta setado com a propriedade `number`: `android:inputType="number"`



EditText

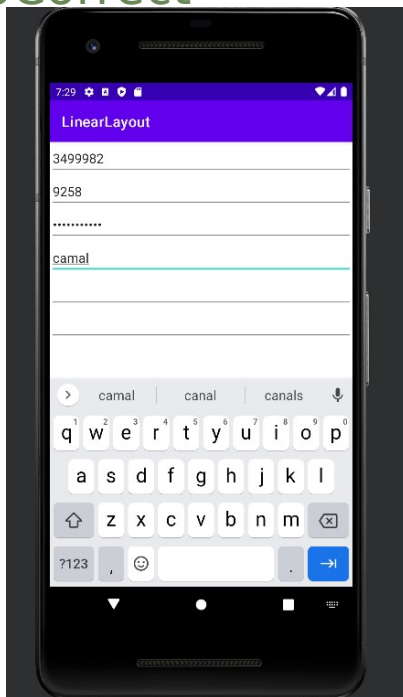
Ao setar informações no terceiro campo de entrada podemos perceber que as informações são mascaradas com Como alteramos a propriedade `inputType` do `EditText` para `textPassword` as informações ficam escondidas:
`android:inputType="textPassword"`



EditText

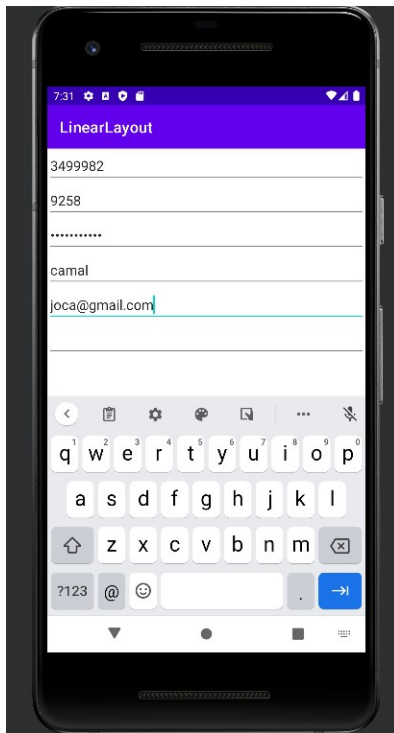
Ao setar informações no quarto campo de entrada, enquanto estamos pressionando as teclas no teclado do emulador, é possível ver que aparece sugestões de palavras. Isso porque estamos usando a propriedade `textAutoCorrect` para o `inputType`:

`android:inputType="textAutoCorrect"`



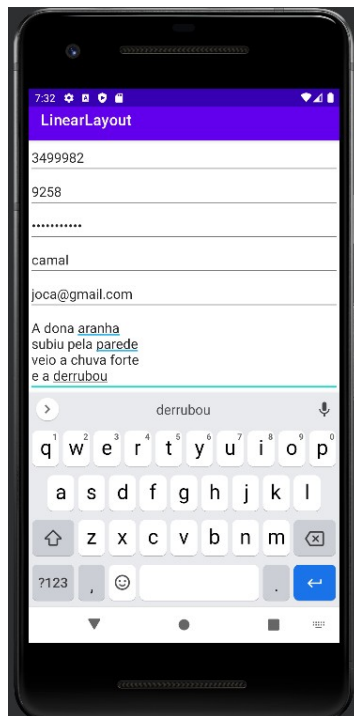
EditText

Ao setar informações no quinto campo de entrada é possível perceber que o teclado alterou no emulador alterou novamente. Nesse elemento já estamos usando a propriedade `textEmailAddress` para o `inputType`: `android:inputType="textEmailAddress"`



EditText

Já no sexto campo de entrada é possível realizar quebra de linhas dentro do input de texto. Nesse elemento estamos usando a propriedade `textMultiLine` para o `inputType`:
`android:inputType="textMultiLine"`



EditText

Além da propriedade `inputType`, temos também a propriedade `imeOptions`. Essa propriedade nos permite alterar o botão enter do teclado do emulador. O android permite as seguinte opções:

<code>actionSearch</code>	Altera a tecla enter para uma lupa informando que o texto informado no editText será para uma busca.
<code>actionDone</code>	Altera a tecla com o simbolo V (Tipo um check) de feito
<code>actionGo</code>	Altera o teclado e a tecla enter com o simbolo de enviar
<code>actionNext</code>	Altera a tecla para um seta de próximo
<code>actionSend</code>	Altera o tecla com o simbolo de enviar
<code>actionPrevious</code>	Altera o tecla com o simbolo de anterior
<code>actionNone</code>	Deixa o tecla enter como o padrão que é enter
<code>actionUnspecified</code>	A tecla fica com padrão do <code>actionDone</code>

Vamos praticar?

Altere os `EditText` anteriores e adicione a propriedade `android:imeOptions=""` e sete os valores conforme a tabela acima.

EditText

Bom, agora vamos deixar os **EditText** mais informativos adicionando uma dica para o usuário. Para isso, usaremos a propriedade **android:hint=""** e dentro das **aspas duplas** informamos o texto que queremos que apareça como dica, ex:

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="phone"
    android:hint="Digite o seu telefone"/>

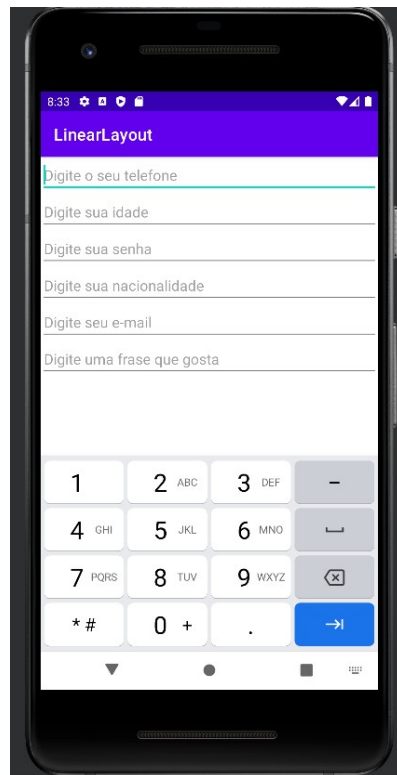
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:hint="Digite sua idade"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:hint="Digite sua senha"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textAutoCorrect"
    android:hint="Digite sua nacionalidade"/>

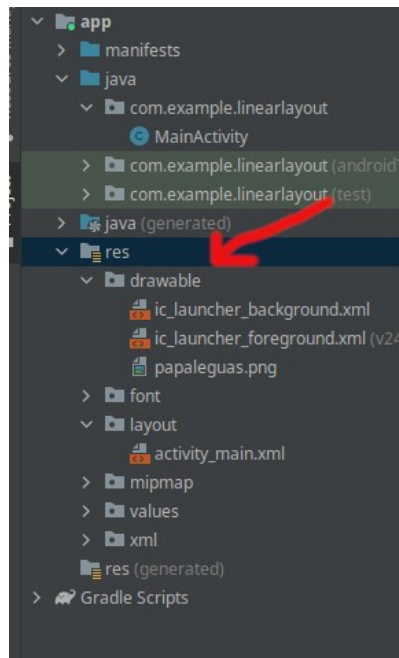
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:hint="Digite seu e-mail"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine"
    android:hint="Digite uma frase que gosta"/>
```



ImageView

- Permite a inserção de uma imagem no projeto:
 - Faça o download de uma imagem e adicione a imagem dentro da pasta **res/drawable**;



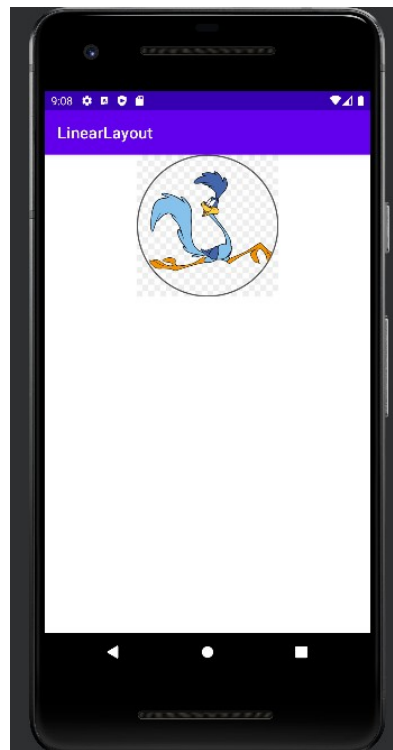
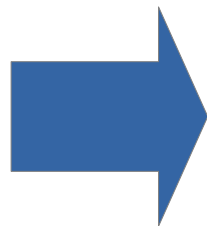
ImageView

- Permite a inserção de uma imagem no projeto:
 - Agora vamos adicionar o elemento **ImageView** no **LinearLayout**, ajustar o tamanho e emular a aplicação;

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

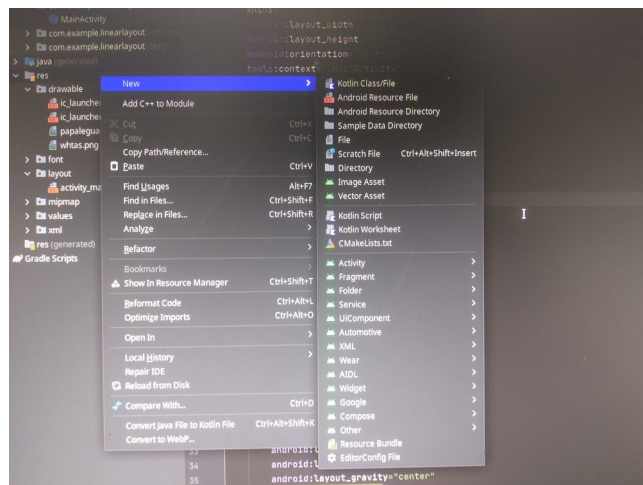
    <ImageView
        android:layout_gravity="center"
        android:layout_width="210dp"
        android:layout_height="179dp"
        android:src="@drawable/papaleguas" />

</LinearLayout>
```



ImageView

- Permite a inserção de uma imagem no projeto:
 - Existe outras formas de acionar imagens:
 - Uma delas é o uso das imagens/vetores que o android fornece
 - Ao clicar com o botão direito no diretório **res** é possível visualizar as opções para adição de imagem (**Image Asset** e **Vector Asset**):



Praticando...



Referências

LECHETA, Ricardo. R. Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK. 5ª ed. São Paulo, SP: Novatec, 2016.