



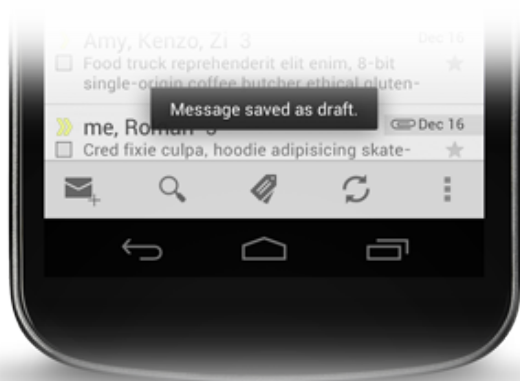
## Android - Cards de Referência

### Toast

Bem vindo ao nosso oitavo card sobre Android! :-) Vamos ver como usar o Toast para exibir mensagens para os usuários!

#### TOASTS

Os Toasts são uma forma fácil e bastante simples para você exibir para os seus usuários mensagens rápidas. Certamente, você já viu mensagens que aparecem na parte inferior da tela e que ficam lá por pouco tempo. Por exemplo veja a imagem abaixo que tiramos do próprio site do Android.



Os Toasts somem por conta própria, a partir de um determinado tempo que você mesmo determina. Você faz um Toast através da classe **android.widget.Toast** usando o método **makeText()**.

```
public class MainActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Toast toast =
            Toast.makeText(this,
                           "Mensagem",
                           Toast.LENGTH_SHORT);
        toast.show();
    }
}
```

Vamos nos aprofundar no código exibido acima. Primeiro criamos uma instância de Toast usando **makeText()** e passando como parâmetro um Contexto (no caso, a própria Activity), uma mensagem e o tempo de exibição. Na própria classe Toast já temos duas constantes de tempo. **LENGTH\_SHORT** e **LENGTH\_LONG**.

O posicionamento padrão do Toast é na parte inferior da tela, contudo, você pode modificar isto, conforme seu gosto! E você faz isto usando o método **setGravity()**. Ele recebe três parâmetros: o primeiro, uma constante que define a posição na tela, o segundo a posição no eixo X e o terceiro a posição no eixo Y.

Por último, existe um método **makeText()** que recebe como segundo parâmetro um inteiro e não uma String. Consegue visualizar o motivo? Sim, trata-se do ID do recurso que está no arquivo **/res/values/strings.xml**. Aliás, é extremamente indicado que você não coloque Strings diretamente no seu código Java, mas, sim, neste arquivo de recursos.

#### TOASTS PERSONALIZADOS

Achou o Toast padrão meio feio? Meio esquisito? Bom, você pode ter um Toast visualmente mais agradável. Só depende de você. O Android permite que você personalize o Toast. Vamos ver como?

Primeiro, vamos definir um layout para o Toast. Fazemos isso criando um arquivo XML dentro da pasta **/res/layout**. O código abaixo foi copiado descaradamente da documentação do Android. :-) Vamos observar ele com mais calma!

```
<LinearLayout
    android:id="@+id/toast_layout_root"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#DAAA"
    android:orientation="horizontal"
    android:padding="8dp" >
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="8dp"
        android:src="@drawable/ic_launcher" />
    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFF" />
</LinearLayout>
```

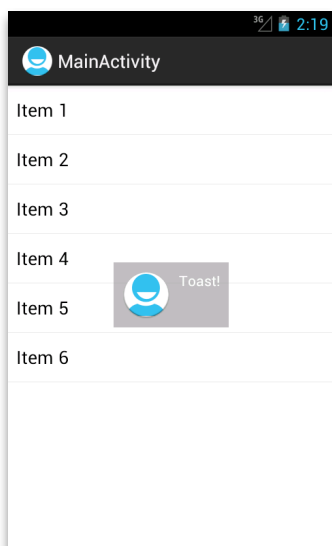
É um layout simples, no qual colocamos um `LinearLayout` com dois campos, uma imagem e um texto. Agora, precisamos usar este layout no Toast. Vamos ver o código Java.

```
public class MainActivity extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        LayoutInflater inflater = getLayoutInflater();  
        View layout = inflater.inflate(R.layout.toast,  
            (ViewGroup) findViewById(R.id.toast_layout_root));  
  
        TextView text = (TextView) layout.findViewById(R.id.text);  
        text.setText("Toast!");  
  
        Toast toast = new Toast(this);  
        toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);  
        toast.setDuration(Toast.LENGTH_LONG);  
        toast.setView(layout);  
        toast.show();  
    }  
}
```

Primeiro, precisamos de um **LayoutInflater** para fazer o serviço de traduzir o código XML do layout em algo que possa ser usado no código Java. Depois, usamos o **LayoutInflater** para “inflar” XML usando o método **inflate()**. Ele recebe dois parâmetros. O primeiro, o ID do recurso. No nosso caso, o **R.layout.toast** que criamos. O segundo, é a View raiz. Neste caso, o `LinearLayout` da listagem anterior.

Agora que temos a View “inflada” a partir do arquivo XML de layout, podemos usar ela para obter as outras Views dentro dela. Vamos usar o método **findViewById** que já estamos acostumados a usar. Obtemos o **TextView** e definimos um valor para ele. Agora, só nos resta informar para o Toast quem é a View, usando o método **setView(View)**.

Um último detalhe sobre os Toasts. Não use o construtor padrão dessa classe a não ser que você vá usar Views personalizadas. Caso contrário, prefira sempre o método estático **makeText()**.





## Sobre o Autor

### Marlon Silva Carvalho

É um programador de longa data. Já peregrinou por diversas linguagens e hoje se considera um programador agnóstico. Atualmente, está fascinado pelo mundo mobile e suas consequências para a humanidade. Está mais focado no desenvolvimento para Android, embora também goste de criar aplicativos para iOS.

Trabalha em projetos sobre mobilidade no SERPRO, tendo atuado no projeto Pessoa Física, para a Receita Federal do Brasil. Você pode encontrar ele no Twitter, no perfil @marlonscarvalho, em seu blog, no endereço <http://marlon.silvacarvalho.net> e através do e-mail [marlon.carvalho@gmail.com](mailto:marlon.carvalho@gmail.com).

## Sobre os Cards

Caso você já tenha visto os excelentes RefCards da DZone, deve ter percebido a semelhança. E é proposital. A ideia surgiu após acompanhar estes RefCards por um bom tempo. Contudo, eles são mais gerais. O objetivo destes Cards é tratar assuntos mais específicos. Trazendo informações relevantes sobre eles.

Caso tenha gostado, continue nos acompanhando! Tentaremos trazer sempre novos cards sobre os mais variados assuntos. Inicialmente, focaremos no Android.

O ícone principal deste card é de autoria de Wallec e foi obtido em seu profile no DeviantArt.

<http://wwalczyszyn.deviantart.com/>

Este trabalho usa a licença **Creative Commons Attribution-NonCommercial-ShareAlike 3.0**

<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Basicamente, você pode copiar e modificar, desde que redistribua suas modificações. Também não é permitido usar este material ou suas derivações de forma comercial.