



**Marlon Silva Carvalho**  
**@marlonscarvalho**  
**marlon.carvalho@gmail.com**



<http://developer.android.com/reference/java/lang/Thread.html>

# **Tarefas Assíncronas**

## **Tarefas “Bloqueantes” não são mais...**

**...permitidas na UI Thread a partir do Android 4.X**



<http://developer.android.com/reference/java/lang/Thread.html>

# Tarefas Assíncronas

**Antes, era possível usar mas...**

**...bloqueava a tela do usuário! Precisamos usar tarefas assíncronas!**



<http://developer.android.com/reference/java/lang/Thread.html>

# **java.lang.Thread**

## **As Threads existem no Android...**

**...e funcionam igual a como você usa no Java "padrão".**



<http://developer.android.com/reference/java/lang/Thread.html>

# **java.lang.Thread**

## **Só tem um problema:**

**o acesso aos elementos de interface que estão na "UI Thread". Eles não são Thread Safe.**



# Prática 1

## Explorar Erro de Threads

Criar um Projeto e tentar acessar elementos da UI Thread



<http://developer.android.com/reference/android/os/AsyncTask.html>

# android.os.AsyncTask

## Utilitário que facilita a execução...

```
new AsyncTask<Void, Void, String> () {
```

**...de tarefas assíncronas.**

```
@Override
```

```
protected void onPreExecute() { }
```

```
@Override
```

```
protected String doInBackground(Void... params) {  
    return null;  
}
```

```
@Override
```

```
protected void onPostExecute(String result) { }
```

```
}.execute();
```





# Prática 2

## AsyncTask

Explorar o uso da AsyncTask





<http://developer.android.com/reference/android/os/AsyncTask.html>

# **android.os.AsyncTask**

## **Cuidado:**

**com a rotação de tela. A Activity é destruída e a AsyncTask terá uma referência inválida!**



<http://developer.android.com/reference/android/os/AsyncTask.html>

# **android.os.AsyncTask**

## **Rotação de Tela**

**O ideal é guardar uma referência para a Activity atual dentro da AsyncTask!**



<http://developer.android.com/reference/android/os/AsyncTask.html>

# **android.os.AsyncTask**

## **Cuidado:**

**AsyncTask.cancel() não cancela imediatamente a Task. Só muda o estado para "cancelado"**



<http://developer.android.com/reference/android/os/AsyncTask.html>

# **android.os.AsyncTask**

## **Cuidado:**

**No Android 4: as Tasks são executadas sequencialmente. Antes era paralelo.**



<http://developer.android.com/reference/android/os/AsyncTask.html>

# **android.os.AsyncTask**

## **Cuidado:**

**Para forçar execução em paralelo: `AsyncTask.executeOnExecutor(THREAD_POOL_EXECUTOR)`**



<http://developer.android.com/reference/android/os/Handler.html>

# **android.os.Handler**

## **Facilita a comunicação entre Threads...**

**...permite você enviar mensagem entre Threads, incluindo a UI Thread.**



# Prática 3

## Handler

Comunicar com a UI Thread usando Handlers



<http://developer.android.com/reference/android/os/Handler.html>

# **android.os.Handler**

## **Views já possuem um Handler...**

**...associado com elas. Podemos fazer View.post(Runnable)!**





# Prática 4

## Handler

Testar o uso de `View.post()`



<http://developer.android.com/reference/android/os/Handler.html>

## **android.os.Handler**

**Também pode agendar código para...**

**...execução no futuro com `postDelayed()`!**



# Prática 5

## Handler

Agendar a execução de um Runnable com o Handler