

# Programação em Dispositivos Móveis

## *Etapa 1*

Estudantes:

André Domingos, 24503, 24503@alunos.isel.pt

Hugo Ferreira, 32983, 32983@alunos.isel.pt

Luís Duarte, 30713, 30713@alunos.isel.pt

16 de Abril de 2012

**keywords:** *Java, Android, Activity, Yamba*

## 1 Decisões de Implementação

Como abordagem à Programação para Dispositivos Móveis tem-se como projecto inicial o desenvolvimento de uma aplicação baseada na rede social *Twitter*<sup>1</sup>, *Yamba*<sup>2</sup>.

Esta fase inicial da implementação tem por objectivo consolidar os conhecimentos adquiridos sobre a matéria leccionada no início das aulas, destacando-se os seguintes tópicos:

- Internacionalização da aplicação;
- Conceito de *Activity*, através do desenvolvimento de quatro componentes deste tipo (*Status*, *Preferences*, *Timeline* e *Detail*);
- Execução Assíncrona em *Android*;
- Opções, Menus e preferências;
- *Tasks* e modos de lançamento das mesmas.

### 1.1 Internacionalização da aplicação

A internacionalização da aplicação é efectuada com base nos recursos da aplicação, nomeadamente no ficheiro *strings.xml*. Para permitir que a aplicação esteja traduzida para Português é criado um directório *res/values-pt/strings.xml* que permite à plataforma utilizar estes valores para *strings* consoante a língua configurada. O valor por omissão reside em *res/values/strings.xml*.

### 1.2 Execução Assíncrona

A utilização de mecanismos de execução assíncrona é obrigatória para o desenvolvimento de uma boa experiência de utilizador. No desenvolvimento da Etapa 1 são efectuadas duas operações com recurso a este mecanismo:

- Actualização do estado no Yamba;
- Leitura dos vários estados dos utilizadores do Yamba.

---

<sup>1</sup><https://twitter.com/>

<sup>2</sup><http://yamba.marakana.com/>

Para esta implementação é declarada a classe privada *UpdateStatusTask* na classe *StatusActivity* e a classe privada *UpdateTimelineTask*. Ambas estendem a classe *AsyncTask* disponibilizada pelo *Android*.

É feito o *override* dos métodos *doInBackground* e *onPostExecute*. O primeiro é invocado pela *thread* da interface gráfica e executa uma tarefa em segundo plano garantindo que a aplicação continua a "responder". O segundo método é invocado quando pelo primeiro aquando do fim da execução do código.