**2ª Serie**

**Programação em Dispositivos Moveis**

Ricardo Nunes - 31656

João Silvestre – 32766

14/05/2012

*Instituto Superior Engenharia de Lisboa*



Engenheiro: Pedro Pereira

Engenharia Informática e de Computadores

Semestre de Verão 2011/2012

Índice

[Introdução 2](#_Toc324794188)

[Estrutura do Trabalho 2](#_Toc324794189)

[UserInfoActivity 2](#_Toc324794190)

[AsyncUserInfo 2](#_Toc324794191)

[UserInfoPullService 2](#_Toc324794192)

[Conclusão 2](#_Toc324794193)

# Introdução

Neste trabalho foi pedida a implementação de uma aplicação em Android, que serve de cliente de serviços do tipo do Twitter.

Neste documento encontra-se explicado todas as decisões e aspectos de implementação, que se considerem relevantes. Aqui encontram-se documentadas as decisões e o objectivo que se pretende atingir com a tomada destas.

# Estrutura do Trabalho

## UserInfoPullService

O *bounded* *service* responsável por devolver os dados necessários pela *UserInfoActivity*. Como corre num processo separado, apenas aceita pedidos através de mensagens. Implementa uma operação (GET\_USER\_INFO) que tem como retorno um *bundle* com os 4 campos pedidos.

Por cada novo pedido recebido obtém o utilizador actual de *TwitterAsync* (apesar de correr num contexto estático diferente ambos partilham as mesmas *preferences*, daí termos ambos a conectar-se com o mesmo utilizador) e usa os métodos disponíveis nesta classe (*Twitter*.*User*) para obter o número de status, subscrições e subscritores associados ao utilizador actual.

## UserInfoActivity

Esta *activity* é responsável por mostrar o nome de utilizador, número de status, subscrições e subscritores do utilizador activo na aplicação e é apenas acessível pelo menu da aplicação.

Os dados são obtidos de forma assíncrona usando *AsyncUserInfo*, pedidos após invocação do método *onResume* desta *activity*. De maneira a pedir os dados é necessário enviar um *GetUserInfoCompletedListener* (*listener* a notificar quando os dados forem retornados de *UserInfoPullService*) que, nesta implementação, afecta o valor das *TextView* presentes no formulário com o valor correspondente.

## AsyncUserInfo

Esta classe é a ponte entre a *UserInfoActivity* e o *UserInfoPullService*, responsável pela interacção com o serviço através do envio e recepção de mensagens.

Após receber um novo pedido de *UserInfoActivity*, guarda o *GetUserInfoCompletedListener* recebido, gera e envia uma *Message* para o *UserInfoPullService* e retorna de imediato. Após receber a resposta de *UserInfoPullService* o *bundle* presente na resposta é interpretado e o *listener* é notificado.

# Conclusão

Todos os objectivos propostos pelo enunciado foram atingidos, no entanto, algumas das soluções precisam de uma melhoria, principalmente o tratamento de excepções. A forma como o menu foi implementado também poderá apresentar alguns desafios no futuro dado que obriga a usar herança, e esta é “limitada” em Java.

No entanto, acredita-se ter-se conseguido uma solução modular e que permitirá a implementação de novas funcionalidades facilmente no futuro.