

# Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Semestre de Inverno 2016/2017

## YAWA

### Programação em Dispositivos Móveis



Trabalho elaborado pelo Grupo 11:

Tiago Oliveira - 40842

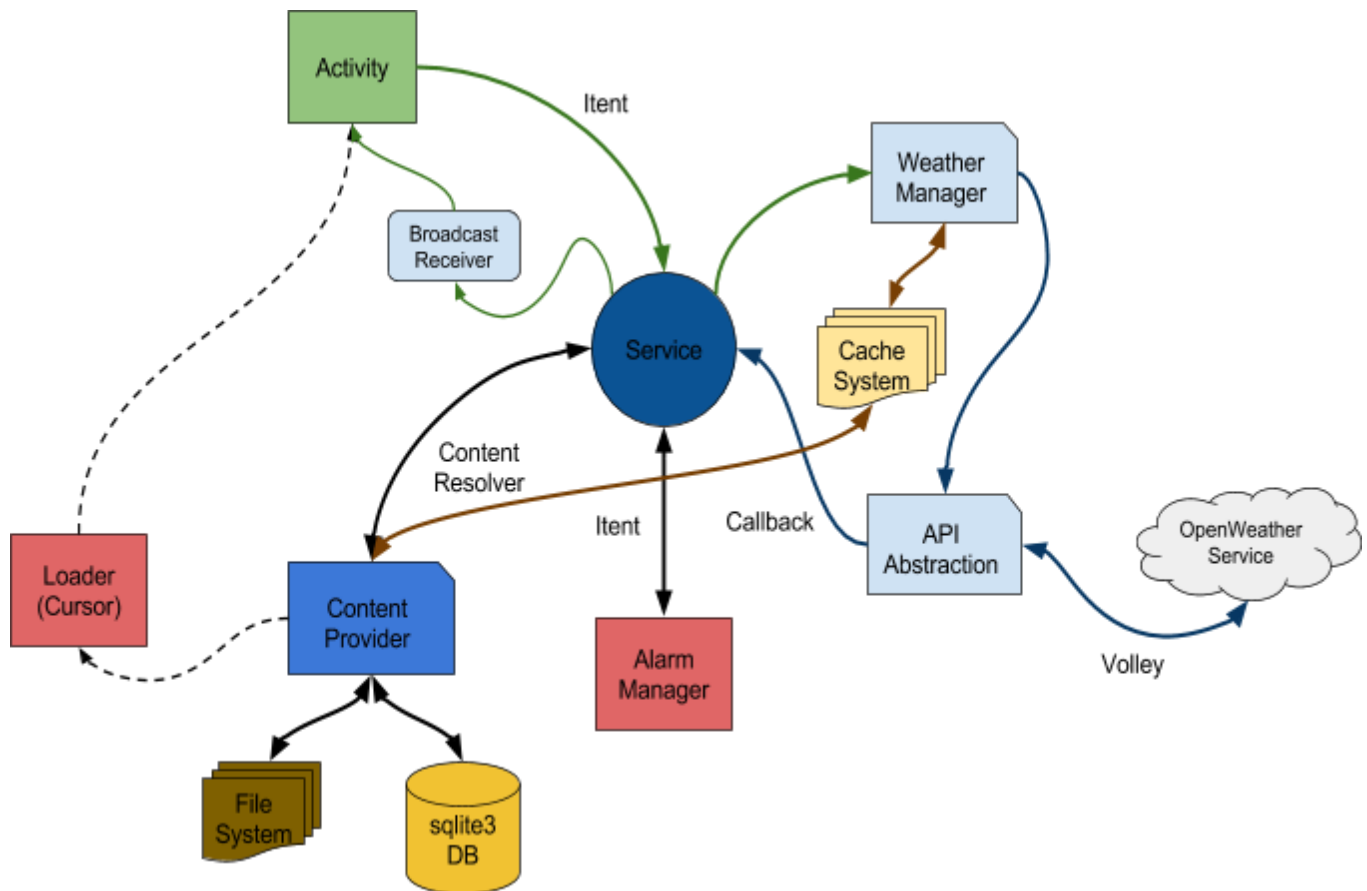
Cláudio Prates - 40908

Daniel Rosa - 40909

# Índice

<b>Diagrama</b>	<b>3</b>
<b>Service</b>	<b>3</b>
<b>WeatherManager</b>	<b>3</b>
<b>Abstração da API</b>	<b>4</b>
<b>Sistema de Cache</b>	<b>4</b>
<b>Actualização de Dados na GUI</b>	<b>4</b>
<b>Organização da GUI</b>	<b>5</b>

## Diagrama



## Service

Tem como função executar tarefas em plano de fundo para deixar a interface gráfica desimpedida ou ainda para executar tarefas sem a necessidade de uma interface gráfica, como é o exemplo das actualizações automáticas.

Com o intuito de poupar bateria, o serviço é executado apenas o tempo necessário, terminando após a conclusão de todas as pedidos.

As actividades podem receber notificações da conclusão dos pedidos, caso necessário, através de um broadcast gerado pelo serviço.

## WeatherManager

Trata-se do elemento que contém a lógica relacionada com os pedidos do tempo.

Tem como propósito retirar do Serviço a responsabilidade de por exemplo saber distinguir o que deve ser feito quando o utilizador quer saber o tempo corrente ou quer saber a previsão do tempo.

Quando o Serviço precisa fazer um pedido apenas tem de saber qual a função certa a chamar. Este elemento encarrega-se do restante processo, que inclui questionar o sistema de Caches, em pedidos que seja relevante, e interagir com a API Abstraction para despoletar pedidos ao serviço externo.

## Abstracção da API

Com o objectivo de não comprometer a aplicação com a API utilizada, foi criado um conjunto de classes que ajudam a que a aplicação seja independente.

Tarefas como a geração dos URL's, conversão da informação recebida para um formato independente que a aplicação percebe e um gerador de pedidos, são as funções desta abstracção.

## Sistema de Cache

O sistema de cache implementado tem dois níveis.

Um primeiro nível (L1), que mantém a informação utilizada mais recentemente na memória RAM para um acesso mais rápido. A implementação foi feita com uma LRUCache.

No segundo nível (L2), a informação é guardada de forma persistente, no armazenamento local do dispositivo, na forma de ficheiros.

O acesso a esses ficheiros é feita através do Content Provider da aplicação, com o objectivo desta informação poder ser partilhada com outras aplicações em versões futuras.

Quando a aplicação faz o pedido de uma imagem, antes da imagem ser descarregada, é feito um pedido ao sistema de cache.

Este recebe o pedido, consulta L1, em caso de cache miss consulta L2, e em caso de novo cache miss retorna sem sucesso, o que despoleta uma tentativa de download da imagem.

Para manter o registo das imagens que se encontram em cache, existe uma tabela na base de dados a aplicação que mantém uma lista do estado da cache.

O sistema de cache necessita ainda de alguns melhoramentos, concretamente no controlo dos ciclos de vida das imagens em cache. Embora a tabela de registo já esteja preparada, este controlo não está a ser feito.

## Actualização de Dados na GUI

Após uma actualização dos dados na Base de Dados (BD), seja ela despoletada pelo utilizador ou não, é necessário actualizar a interface gráfica com os novos dados.

Essa actualização é feita de forma quase automática através de um CursorLoader que é criado no início da/s Activities onde seja relevante, e que é notificado sempre que existam alterações na informação.

## Organização da GUI

Para organizar a GUI da aplicação optamos pela utilização de Fragmentos em algumas Activities com o objectivo de reaproveitar código e melhorar a experiência de utilização.

A MainActivity é composta por um Fragmento que mostra a informação detalhada do tempo, e é utilizado também nas previsões para mostrar a informação detalhada de um determinado dia.

Na Activity das previsões são utilizados dois Fragmentos que trocam entre si.

Um deles mostra uma lista resumida com os dias da previsão, que ao serem seleccionados pelo utilizador trocam para o fragmento que mostra os detalhes desse dia.

Este esquema permite uma troca mais rápida entre “janelas”.

A lógica de troca entre fragmentos foi deixada a cargo da Activity para retirar o máximo possível de responsabilidades dos Fragmentos.