

Mestrado em Engenharia Informática  
Estágio  
Relatório Final

# Android Knowledge Graph Development

Daniel Marques Freire Jorge  
djorge@student.dei.uc.pt

*Orientadores:*  
Alexandre Miguel Pinto  
José Miguel Carvalho Larguesa Grade

3 de Julho de 2013



**FCTUC** DEPARTAMENTO  
DE ENGENHARIA INFORMÁTICA  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



# Agradecimentos

À Bárbara.

Aos meus pais Salvador e Maria, ao Adriano, ao Tiago e ao Ivo, por anos de aprendizagem e construção pessoal a cada minuto de convivência.

Ao Miguel, ao Professor Alexandre e a todos os colegas com quem partilhei esta etapa da minha vida (quase) profissional.



# Resumo

A Maisis, com o desenvolvimento da aplicação OObian Android, colmatará o vazio em que se encontra, neste momento, o ambiente OObian na área mobile, podendo apresentar, finalmente, não só uma forma evoluída de criação e contextualização de informação, como de consumo, tratamento e apresentação do conhecimento presente nas suas estruturas de conhecimento.

Para além de ser uma componente nova no ambiente OObian, o OObian Android possibilitará correlacionar o consumidor da informação com a informação propriamente dita, localizando-o entre georeferências, contextualizando-o em assuntos do seu interesse, alertando-o para conteúdo novo, permitindo uma consulta de práticas de utilização dos utilizadores sobre a estrutura de conhecimento.

O desenvolvimento da componente Android do OObian estará também relacionado com o desenvolvimento de uma componente de Analytics. Esta componente tirará partido de tecnologia de filtragem de dados para que seja facilmente consultado o tipo de utilização que é dado sobre a estrutura de conhecimento, pelos seus utilizadores. Esta componente será uma mais-valia para o ecossistema OObian, uma vez que esta poderá ser reutilizada por outros tipos de cliente (Mobile ou Web).

**Palavras chave:** Android, OObian, Ontologia, Analytics, Mobile.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Enquadramento . . . . .	4
1.1.1	Contexto Organizacional . . . . .	4
1.1.2	Contexto Económico e Social . . . . .	4
1.1.3	Contexto Tecnológico . . . . .	5
1.2	Motivação . . . . .	6
1.2.1	OObian Android . . . . .	6
1.2.2	OObian Analytics . . . . .	7
1.3	Objectivos . . . . .	7
<b>2</b>	<b>Estado da arte</b>	<b>9</b>
2.1	Desenvolvimento de Aplicações Android . . . . .	9
2.1.1	Sistemas Operativos e Dispositivos Móveis . . . . .	9
2.1.2	Android – Uma breve descrição . . . . .	9
2.1.3	Abordagem Nativa . . . . .	10
2.1.4	Abordagem Web-Based . . . . .	12
2.1.5	Abordagem Híbrida . . . . .	14
2.2	Avaliação de Frameworks . . . . .	15
2.2.1	Análise Genérica . . . . .	15
2.2.2	Análise Qualitativa . . . . .	16
2.2.3	Análise Funcional . . . . .	19
2.2.4	Considerações . . . . .	20
2.2.5	Análise final . . . . .	20
2.3	Analytics . . . . .	21
2.3.1	A importância da monitorização . . . . .	21
2.3.2	O conceito . . . . .	22
2.3.3	Ferramentas . . . . .	23
2.3.4	Considerações . . . . .	24

<b>3</b>	<b>Definição de requisitos</b>	<b>25</b>
3.1	Metodologias de Análise de Requisitos . . . . .	25
3.1.1	ISO/IEC 9126 . . . . .	25
3.1.2	FURPS . . . . .	26
3.1.3	FURPS+ . . . . .	27
3.1.4	Metodologia de análise de requisitos a usar . . . . .	27
3.2	Análise de Requisitos . . . . .	28
3.2.1	Funcionalidades Macro . . . . .	28
3.2.2	Requisitos . . . . .	29
3.3	Descrição do Sistema . . . . .	35
3.3.1	Navegação . . . . .	35
3.3.2	Prototipagem . . . . .	36
3.3.3	Casos de uso . . . . .	38
<b>4</b>	<b>Arquitectura e desenho</b>	<b>45</b>
4.1	Fundamentos . . . . .	45
4.2	Perspectiva Funcional . . . . .	46
<b>5</b>	<b>Desenho da solução e decisões tecnológicas</b>	<b>49</b>
5.1	OObian Android . . . . .	49
5.1.1	Desenho da Solução . . . . .	50
5.1.2	Decisões Tecnológicas . . . . .	53
5.2	OObian Analytics . . . . .	56
5.2.1	Desenho da Solução . . . . .	56
5.2.2	Decisões Tecnológicas . . . . .	58
<b>6</b>	<b>Desenho de interface</b>	<b>59</b>
6.1	OObian Android . . . . .	59
6.2	OObian Analytics . . . . .	62
<b>7</b>	<b>Testes</b>	<b>63</b>
<b>8</b>	<b>Planeamento</b>	<b>71</b>
8.1	Plano de Estágio . . . . .	71
8.2	Análise de Riscos . . . . .	72
8.3	Trabalho Futuro . . . . .	73



API	<i>Application Programming Interface</i>
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
HTTP	<i>Hypertext Transfer Protocol</i>
ISO	<i>International Organization for Standardization</i>
IEC	<i>International Electrotechnical Commission</i>
JPA	<i>Java Persistency API</i>
JSON	<i>JavaScript Object Notation</i>
MEI	Mestrado em Engenharia Informática
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>
UC	Universidade de Coimbra
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>

Tabela 1: Acrónimos



# Lista de Tabelas

1	Acrónimos . . . . .	ix
2.1	Análise qualitativa das frameworks . . . . .	17
2.2	Linguagens utilizadas nas frameworks . . . . .	17
2.3	Vantagens e desvantagens do uso do Page Tagging[1] . . . . .	22
2.4	Vantagens e desvantagens do uso do Logging[1] . . . . .	23
3.1	Requisitos ISO/IEC 9126 . . . . .	26
3.2	Requisitos FURPS[2] . . . . .	26
3.3	Requisitos adicionais FURPS+[2] . . . . .	27
3.4	Requisitos mínimos para executar o OOBian Android . . . . .	31



# Lista de Figuras

2.1	Fragmentação do ambiente Android[3]	10
2.2	Exemplo de uso do Google Analytics	23
2.3	Exemplo de uso do Piwik	24
3.1	Funcionalidades Macro da aplicação OObian Android	28
3.2	Mockup de ecrã geral da aplicação e de granularidade mínima	36
3.3	Mockup de ecrã de pesquisa	37
3.4	Mockup de itens gerais da aplicação	38
3.5	Protótipos de ecrã de login e inicial	39
3.6	Protótipo de ecrã de perfil de utilizador	40
3.7	Protótipo de ecrã de filtragem	41
3.8	Protótipos de ecrãs de pesquisa e granularidade mínima	42
3.9	Protótipo de ecrã de mapa	43
4.1	Arquitectura geral do OObian Android	46
4.2	Funcionalidades da aplicação OObian Android	47
4.3	Comunicação entre o OObian Android e o OObian	48
5.1	Listview	50
5.2	GridView	50
5.3	Actionbar	51
5.4	Diagrama de base de dados da aplicação OObian Android	53
5.5	Tempos de serialização + deserialização (nanosegundos)	55
5.6	Tamanho de objecto serializado (Byte)	56
5.7	Diagrama de base de dados do módulo OObian Analytics	58
6.1	Ecrã inicial da aplicação OObian Android	59
6.2	Navegação a partir da raiz da estrutura de conhecimento	60
6.3	Vista de navegação em instância	61
6.4	Módulo OObian Maps	61
7.1	Questionário sobre navegação livre	66

*LISTA DE FIGURAS*

1

7.2	Questionário sobre navegação contextualizada . . . . .	67
7.3	Questionário sobre padrões de usabilidade . . . . .	68



# Capítulo 1

## Introdução

Vivemos num mundo no qual a informação é crucial nas decisões que tomamos no nosso dia-a-dia. Todos os dias somos estimulados pelos mais variados tipos de informação que, à primeira vista, parecem não se relacionar ou associar. Pois bem, toda esta informação tem, pelo menos, um ponto em comum: o seu consumidor. É o consumidor de toda esta informação que atribui o verdadeiro significado a tudo aquilo que observa, a tudo aquilo que procura. É o consumidor de toda esta informação que escolhe instintivamente se "Java" é uma ilha ao largo do Pacífico ou uma linguagem de programação, se "Queen" é uma banda ou a sua majestade, se "Football" é jogado com as mãos ou com os pés.

Toda esta associação de ideias é efectuada tão naturalmente pelo ser humano que se tornou um desafio transportar este comportamento para uma máquina. Apesar de parecer uma simples transferência de saber, esta troca está indubitavelmente relacionada com as mais variadas formas de apresentação de informação e, conseqüentemente, com as mais variadas formas de captação dessa mesma informação, o que levará à rápida exaustão de qualquer sistema informático ou semelhante que a pretenda simular. Assim, surgiu a necessidade da utilização de ferramentas flexíveis de representação e navegação em estruturas de conhecimento. O OObian apresenta-se como uma dessas ferramentas e, como gestor de estruturas de conhecimento, pretende catalogar toda e qualquer informação dispersa e associá-la para que seja facilmente consultada, encontrando pontos de intersecção entre si e o utilizador.



## 1.1 Enquadramento

### 1.1.1 Contexto Organizacional

Este trabalho encontra-se integrado na unidade curricular de Dissertação/Estágio do Mestrado em Engenharia Informática, administrada pelo Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

A realização da unidade curricular de Dissertação/Estágio decorre na Maisis, Information Systems, Lda, sediada em Aveiro. Tendo surgido em 1994, a sua actividade inicial passava pela prestação de serviços. Ao longo dos anos, foi visto como oportunidade a aposta na área da Investigação e Desenvolvimento, surgindo assim vários estudos que culminaram com a criação de sistemas de gestão de conhecimento baseado em ontologias.

A Maisis tem como missão desenvolver soluções tecnológicas, destinadas a servir de forma diferenciada cada cliente.

Entre os parceiros de negócio da Maisis, contam-se: Microsoft, PT-Inovação, InovaRia, ADI – Agência de Inovação, Divultec, Martifer, ColepCCL, Gestamp Aveiro, Primavera, Gestin, Barco.

Os clientes finais da Maisis englobam tanto os parceiros de negócio da Maisis que utilizem os produtos desenvolvidos nesta empresa, como todos aqueles que usufruem de todos os serviços prestados por eles.

### 1.1.2 Contexto Económico e Social

O principal mercado-alvo do OObian é o segmento empresarial que necessite de uma forma rápida de representação de grandes quantidades de informação relacionada. Qualquer empresa que esteja dependente do corelacionamento de informação poderá tirar o melhor partido de uma forma tão inovadora de gestão de conhecimento como a que é oferecida pelo ambiente OObian.

Pretende-se também que o mercado alvo se alargue e passe a incluir qualquer pessoa que pretenda aceder a informação presente em repositórios de conhecimento mais generalistas como redes sociais e plataformas colaborativas de conhecimento.

A população afectada por este tipo de tecnologia inclui todos aqueles que pretendam, de uma forma rápida, prática, simples e intuitiva, aceder a informação muitas vezes dispersa, correlacionando-a.

### 1.1.3 Contexto Tecnológico

Actualmente, é notório o crescimento da utilização de dispositivos móveis para o acesso a informação. Este fenómeno caracteriza-se sobretudo pelo facto de se tornar simples e imediato o que até então necessitava de um posto de trabalho físico, um computador de grandes dimensões ou, no limite, de um computador portátil. Os smartphones, sendo dispositivos de pequeno porte, vêm abrir um novo segmento tecnológico explorado por empresas como a Google (com o Android), a Apple (com o iOS) e a Microsoft (com o Windows Phone).

Os equipamentos a utilizar no desenvolvimento de uma aplicação Android, bem como no desenvolvimento de uma aplicação para outro ecossistema deverão incluir um dispositivo móvel de ambas as gamas (smartphone e tablet), podendo variar consoante os dispositivos-alvo que se pretendem alcançar com os desenvolvimentos.

As linguagens de programação variam com a framework de desenvolvimento escolhida. De uma forma geral, o Java prevalece (como mais à frente teremos oportunidade de observar, todos os controlos nativos são desenvolvidos/convertidos em/para Java). Com outras opções poder-se-á desenvolver utilizando HTML5, C++, Actionscript, JavaScript, JQuery, entre outras.

O protocolo de comunicação entre o OObian Android e o OObian Server passará pela troca de conteúdo serializado através do consumo de serviços web disponibilizados do lado do servidor. Estes serviços web serão monitorizados por um módulo de análise de dados e operações que será desenvolvido no âmbito deste projecto recorrendo a métodos de captura de eventos disponibilizados por bibliotecas de linguagem Java .

O desenvolvimento de uma aplicação Android não se limita à utilização de um kit de desenvolvimento único (Android SDK). Com vista a simplificar o desenvolvimento para todos aqueles que não têm o Java como competência técnica ou como forma de permitir a inclusão de um segmento de developers integrado na área de desenvolvimento web, foram criadas, ao longo dos tempos, outras alternativas. Entre estas alternativas, salientam-se três grupos: frameworks nativas, frameworks web-based e frameworks híbridas.

Para a análise do estado-de-arte, foram consideradas as seguintes frameworks de abordagem nativa:

- Android SDK;
- Adobe Flex;
- Marmalade;
- Mono.

Para uma abordagem web-based, foram consideradas as seguintes frameworks:

- Sencha;
- The M Project;
- jQuery;
- iUI;
- SproutCore;
- Wink.

As frameworks híbridas estudadas foram as seguintes:

- PhoneGap;
- Titanium Mobile.

## 1.2 Motivação

### 1.2.1 OObian Android

A Maisis, com o desenvolvimento da aplicação OObian Android, colmatará o vazio em que se encontra, neste momento, o ambiente OObian na área mobile, podendo apresentar, finalmente, não só uma forma evoluída de criação e contextualização de informação, como de consumo, tratamento e apresentação do conhecimento presente nas suas estruturas de conhecimento.

Os parceiros da Maisis que utilizem o OObian tirarão partido de toda uma nova abordagem ubíqua que não estava disponível até ao desenvolvimento desta nova componente.

Os clientes finais da aplicação OObian Android serão todos os actuais e futuros clientes da versão desktop da aplicação OObian. Neste momento a aplicação OObian encontra-se integrada na montra TICE e está referenciada como estrutura de conhecimento de eleição da DBpedia. A partir daqui, os utilizadores da aplicação OObian Android poderão aceder aos conteúdos disponibilizados por estas plataformas de uma forma móvel, bem como contextualizar essa informação consoante as variáveis que estão associadas ao seu consumidor (georeferencia actual do utilizador, histórico de pesquisas, histórico de operações, entre outras).

A população em geral poderá tirar partido destes desenvolvimentos tendo em conta que será possível, de uma forma célere, aceder a uma estrutura de

conhecimento que não se limitará àquilo que se procura, mas que também incluirá informação que, à partida, será do interesse do utilizador.

Para além de ser uma componente nova no ambiente OObian, o OObian Android possibilitará correlacionar o consumidor da informação com a informação propriamente dita, localizando-o entre georeferências, contextualizando-o em assuntos do seu interesse, alertando-o para conteúdo novo, permitindo uma consulta de práticas de utilização dos utilizadores sobre a estrutura de conhecimento.

### 1.2.2 OObian Analytics

O desenvolvimento da componente Android do OObian estará também relacionado com o desenvolvimento de uma componente de Analytics. Esta componente tirará partido de tecnologia de filtragem de dados para que seja facilmente consultado o tipo de utilização que é dado sobre a estrutura de conhecimento, pelos seus utilizadores. Esta componente será uma mais-valia para o ecossistema OObian, uma vez que esta poderá ser reutilizada por outros tipos de cliente (Mobile ou Web).

Sendo, de certa forma, um desenvolvimento paralelo à aplicação OObian Android, a criação da componente OObian Analytics tornou-se um foco importante no desenrolar do planeamento deste projecto de Mestrado. Devido ao crescimento de tipos de clientes que o OObian, no seu todo, suporta, a adição de mais uma forma de interacção com as estruturas de conhecimento oferecidas pelo OObian tornaram imperativo que fosse possível monitorizar a actividade que está associada a uma grande fonte de informação.

Com a criação do OObian Analytics, espera-se que seja possível equipar o ecossistema OObian com uma ferramenta capaz de medir, coleccionar, analisar e reportar, tanto a informação propriamente dita, como o tipo de informação de que se trata, passando pelo contexto que leva ao consumo dessa informação, ou até mesmo pela motivação de tal operação.

Espera-se que o OObian Analytics seja capaz de dar ao seu utilizador uma perspectiva métrica de todo o ecossistema OObian para que, de um ponto de vista de gestão, possam ser melhoradas as formas de apresentação e uso de informação; e de um ponto de vista de um utilizador comum, possa haver a percepção de como a informação é consumida.

## 1.3 Objectivos

Situamo-nos numa época em que a facilidade de acesso a informação está fortemente associada à utilização de dispositivos móveis que proporcionam

uma experiência mais imersiva e contextualizada. Surgiu então a necessidade de transportar as potencialidades do OObian para um ambiente móvel de grande expansão: o Android. A integração de uma componente ubíqua ao OObian permitirá, para além do simples consumo de informação, contextualizar de uma forma espaço-temporal a navegação na estrutura de conhecimento, potenciando as funcionalidades existentes no OObian. O desenvolvimento desta aplicação está associado a uma componente de análise de dados (OObian Analytics), que trará valor acrescentado aos serviços disponibilizados pelo OObian como estrutura de conhecimento.

Toda a envolvente do desenvolvimento desta aplicação centrar-se-á no ecossistema disponibilizado pelo sistema operativo Android; na utilização de web-services que permitam a transmissão de dados entre o dispositivo móvel e o servidor; na aplicação de Data-(De)Serialization como forma de otimizar essas transmissões; na aplicação de técnicas de captura de dados e operações que permitam a recolha de informação relacionada com os utilizadores da aplicação.

A componente OObian Analytics necessitará da criação de uma infraestrutura paralela do lado do servidor OObian que seja capaz de fazer uma captura de informação, operações e contexto de ações sobre a estrutura de conhecimento. Pretende-se, com esta adição às funcionalidades do OObian, que qualquer tipo de cliente seja capaz de consumir esta informação, podendo consultá-la de uma forma simples e categorizada.

As ferramentas a utilizar neste tipo de desenvolvimento são as mais variadas e vão desde aplicações de desenho e planeamento de usabilidade, frameworks de desenvolvimento aplicacional, IDE's e emuladores para testar a aplicação num leque mais alargado de dispositivos móveis.

# Capítulo 2

## Estado da arte

### 2.1 Desenvolvimento de Aplicações Android

#### 2.1.1 Sistemas Operativos e Dispositivos Móveis

O facto de se pretender ter acesso a informação de uma forma completamente inovadora e sem barreiras geográficas está inevitavelmente associado a uma limitação de recursos, seja em termos de memória, de processamento ou de armazenamento. O aumento da necessidade, ao nível da portabilidade, do acesso à informação levou a que as grandes software houses adaptassem os seus produtos a dispositivos anfitriões com menores capacidades que os típicos computadores. Assim, empresas como a Google, a Apple, a Microsoft e a Samsung, entre outras, decidiram investir no desenvolvimento de sistemas operativos que suportem e sejam suportáveis por smartphones e tablets.

#### 2.1.2 Android – Uma breve descrição

O sistema operativo desenvolvido pela Google, o Android, teve o seu início de disseminação em 2008. Desde esse momento que esta plataforma se destacou pela sua versatilidade, usabilidade e customização, o que a levou a ser amplamente adoptada pelas mais diversas marcas de dispositivos móveis. A par deste sistema operativo, é disponibilizado também um centro de partilha (gratuita ou paga) de aplicações: o Google Play. Este repositório contém já mais de 700000 aplicações e encontra-se neste momento à frente dos números de aplicações disponibilizadas pela AppStore da Apple (650000 aplicações) e pela Microsoft (120000 aplicações). Mais tecnicamente, o Android é baseado no kernel Linux e trabalha sobre o Dalvik (Dalvik Virtual Machine) que, de uma forma sintética, corre executáveis Dalvik previamente convertidos de uma linguagem intermédia (Java, na maioria dos casos). Este formato de

executável permite o funcionamento de aplicações em dispositivos de fracos recursos de hardware (memória e processamento).

O Android encontra-se fragmentado em diversas versões, contendo cada uma delas um leque de funcionalidades que foram sendo adicionadas à medida que os próprios dispositivos móveis foram evoluindo.

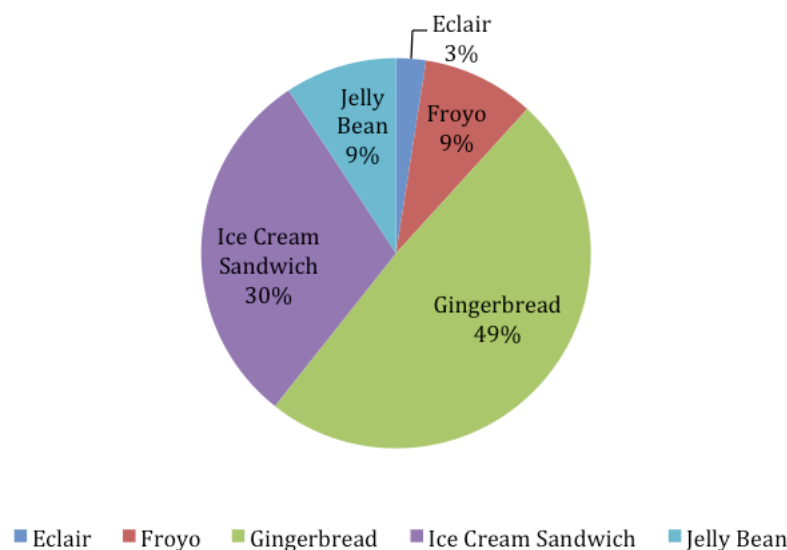


Figura 2.1: Fragmentação do ambiente Android[3]

Conforme é possível observar na Figura 2.1, a versão GingerBread (2.2) é a mais utilizada até à data, sendo suportada pela maioria dos dispositivos móveis (smartphones e tablets) low-end.

Actualmente, de entre todas as versões do sistema operativo Android, podem-se destacar as distribuições Honeycomb (Android 3.0 a 3.2), que são suportadas nativamente apenas por tablets. A título de curiosidade, desde que foi apresentada a versão Ice Cream Sandwich (Android 4.0) que é possível a sua inclusão tanto em smartphones como em tablets, minimizando assim a fragmentação de usabilidade ou de desenvolvimento.

### 2.1.3 Abordagem Nativa

Como referência, uma abordagem nativa inclui todas as frameworks que compilam/geram código nativo dos dispositivos. Para esta abordagem, efectuei a análise das seguintes frameworks:

- Android SDK <sup>1</sup>;
- Adobe Flex <sup>2</sup>;
- Marmalade <sup>3</sup>;
- Mono <sup>4</sup>.

Para cada uma destas frameworks foram tidos em conta vários aspectos como a licença de utilização e de comercialização de aplicação desenvolvidas com essa respectiva ferramenta, o software incluído para o seu desenvolvimento e, entre outros aspectos, a linguagem usada. Seguidamente, a análise individual de cada framework.

### **Android SDK**

A framework nativa desenvolvida pela Google possibilita o desenvolvimento de aplicações através da linguagem Java, sendo a interface do utilizador gerada através de XML. O bundle de criação de aplicações inclui, de uma forma sintética, as SDK Tools e as ADT (Android Development Tools). As SDK Tools agregam uma API de desenvolvimento, o Android Emulator e o AVD (Android Virtual Devices) Manager. As ADT cingem-se à configuração base de desenvolvimento para Eclipse, em forma de plug-in, e o Visual Layout Editor, que possibilita o desenvolvimento simplificado de user-interfaces. A simulação do ambiente Android está a cargo do Android Emulator. A utilização desta framework não apresenta qualquer tipo de encargos.

### **Adobe Flex**

A alternativa apresentada pela Adobe faz uso da linguagem Actionscript3, sendo que toda a interface é gerada com MXML. Esta framework utiliza um conjunto livre de bibliotecas, sendo, por isso, sem encargos para o seu utilizador. Para o desenvolvimento com esta framework são aconselhadas duas abordagens: utilização das bibliotecas através do IDE Eclipse, de uma forma menos assistida e utilização do Adobe Flash Builder. O Adobe Flash Builder (adaptação do IDE Eclipse com vários plugins de apoio ao desenvolvimento com Flex SDK) possibilita a criação intuitiva e facilitada das interfaces com o utilizador, bem como na programação do Actionscript3. Esta opção paga pode ser obtida a partir de 186.94 €, no momento da escrita deste relatório.

---

<sup>1</sup><http://developer.android.com/>

<sup>2</sup><http://flex.org/>

<sup>3</sup><http://www.madewithmarmalade.com/>

<sup>4</sup><http://android.xamarin.com/>



## Marmalade

Esta framework permite o desenvolvimento de aplicações através da linguagem C++, sendo que o deploy poderá ser multiplataforma. Para a sua utilização, é disponibilizado um pacote que inclui um plugin para o MS Visual Studio, o Marmalade Simulator e o Marmalade Studio UI, entre outras ferramentas de menor importância para os objectivos deste estudo. Ao longo da utilização do Marmalade, é-nos possível aceder à API nativa do Android (Android SDK). A sua aquisição para fins comerciais poderá ser feita a partir de cerca de 374.62 €, no momento da escrita deste relatório.

## Mono

Esta opção é também ela um plugin para o MS Visual Studio e faz uso do Android Emulator, incluído nas SDK Tools do Android SDK. As linguagens utilizadas são o C# e o .NET, sendo que se apresenta como uma alternativa para quem pretenda desenvolver aplicações sem ter de se adaptar a outras linguagens de programação (como o Java, que é utilizado pela API nativa do Android SDK). Esta framework é paga e pode ser obtida a partir de 299.55 €, no momento de criação deste relatório.

### 2.1.4 Abordagem Web-Based

A abordagem web-based compreende todas as frameworks que se baseiem simplesmente nos recursos oferecidos por um web-browser e por uma linguagem web. Para esta abordagem, foram consideradas as seguintes opções:

- Sencha <sup>5</sup>;
- The M Project <sup>6</sup>;
- jQuery <sup>7</sup>;
- iUI <sup>8</sup>;
- SproutCore <sup>9</sup>;
- Wink <sup>10</sup>.

---

<sup>5</sup><http://www.sencha.com/>

<sup>6</sup><http://www.the-m-project.org/>

<sup>7</sup><http://jquerymobile.com/>

<sup>8</sup><http://www.iui-js.org/>

<sup>9</sup><http://www.sproutcore.com/>

<sup>10</sup><http://www.winktoolkit.org/>

Visto que todas estas ferramentas possuem métodos de desenvolvimento semelhantes, vou omitir o facto de que dão, todas elas, uso a HTML, CSS e Javascript para criação das aplicações. É também óbvio que todas estas ferramentas permitem um deployment multiplataforma. De salientar que estas frameworks permitem uma pré-visualização das suas capacidades através de Kitchen Sinks<sup>11</sup> presentes nas suas páginas web.

### **Sencha**

Esta web-framework inclui um módulo de desenvolvimento de aplicações para dispositivos Android: o Sencha Touch. Para o apoio ao desenvolvimento, disponibilizam também as ferramentas Sencha Architect, para criar interfaces de um modo rápido e fácil; e o Sencha Animator para gerar animações em CSS3. O módulo Sencha Touch é de utilização livre sob licença comercial e a ferramenta Sencha Architect pode ser adquirida a partir de 299.55 € no momento de redacção deste relatório.

### **The M Project**

O The M Project permite o desenvolvimento de aplicações web baseadas em jQuery Mobile, suporta acesso a conteúdo em off-line de uma forma nativa, permite a persistência de dados e internacionalização. Esta ferramenta é de utilização livre e pode ser adquirida a partir da página do projecto.

### **jQuery**

O jQuery Mobile Framework permite o desenvolvimento de web-apps utilizando a linguagem jQuery; é compatível com, entre outros, Android e iOS; gera aplicações de pequenas dimensões; suporta soluções de acessibilidade (como o Text-to-Speech no Android); suporta eventos associados ao toque; é customizável. A sua utilização é livre de encargos.

### **iUI**

O iUI permite a criação de menus navegacionais e interfaces customizáveis, a utilização do acelerómetro do dispositivo móvel, gera aplicações de pequenas dimensões, e possibilita a integração de vários plugins e extensões. Esta framework é de utilização livre.

---

<sup>11</sup>O uso de uma Kitchen Sink possibilita a demonstração prática de padrões utilizados pelas frameworks de desenvolvimento. São geralmente pequenas aplicações demonstrativas.

## SproutCore

Esta framework compete, em termos de riqueza de conteúdos e de funcionalidades, com as aplicações nativas (escalabilidade, suporte para touch-events, acesso off-line, entre outras). A sua utilização também é livre.

## Wink

O Wink faz uso do JavaScript para a criação de aplicações web móveis ricas. Permite o recurso a touch-events, bem como uma grande aproximação ao ambiente nativo de cada plataforma. O recurso a HTML5 possibilita o uso de 3D CSS, georeferenciação, orientação e gestão de outros sensores do dispositivo. A utilização do código fonte da framework Wink é livre de encargos.

### 2.1.5 Abordagem Híbrida

A abordagem híbrida é caracterizada pela utilização de uma vista web capaz de introduzir recursos nativos, isto é, embeber uma página web num container que tirará partido das capacidades do dispositivo móvel. As frameworks consideradas foram:

- PhoneGap <sup>12</sup>;
- Appcelerator - Titanium Mobile <sup>13</sup>.

## PhoneGap

O PhoneGap é uma alternativa gratuita que faz uso do HTML5, do CSS3 e do Javascript para desenvolvimento das suas aplicações. O seu deployment pode ser feito para várias plataformas, incluindo Android e iOS. Possui uma biblioteca que trabalha sobre o Android SDK, ou seja, todo o código gerado é “traduzido” em código nativo. Esta é uma abordagem mista que tira partido de tecnologias web para o desenvolvimento das aplicações sem que, para isso, se dispensem muitas características físicas do dispositivo móvel, que só são acessíveis a partir da API nativa (utilização de todos os componentes do dispositivo móvel, em que se inclui o compasso, o acelerómetro, o sistema de notificações, entre outros).

---

<sup>12</sup><http://www.phonegap.com/>

<sup>13</sup><http://www.appcelerator.com/>

### **Appcelerator – Titanium Mobile**

O desenvolvimento de uma aplicação no Titanium Mobile é em tudo semelhante ao PhoneGap: dá uso a Javascript, HTML5, CSS3 e possibilita também a inclusão de snippets de código escritos em Python e Ruby. A plataforma de desenvolvimento inclui o Titanium Studio, que é constituído pelo Eclipse com o plugin Aptana Studio. O deployment poderá ser feito de uma forma multiplataforma e a execução destas aplicações poderá ser efectuada através do emulador nativo Android Emulator.

## **2.2 Avaliação de Frameworks**

### **2.2.1 Análise Genérica**

Após analisar genericamente cada uma destas frameworks, foi possível determinar em que situações se adequam cada uma destas três abordagens.

Do lado da abordagem nativa, e visto que o programador tem acesso à API nativa de desenvolvimento, é possível controlar de uma forma total todo o hardware disponibilizado pelo dispositivo móvel, bem como aceder a dados que o utilizador armazena no seu terminal. Esta abordagem permite também uma perfeita integração da aplicação no ambiente do sistema operativo em que se insere. Por outro lado, a abordagem web-based permite o que a nativa, salvo algumas excepções, não consegue: suporte multiplataforma (iOS e Android).

Com a introdução do HTML5 e de todas as suas potencialidades, uma framework web-based consegue dar uso a características comuns em aplicações nativas, como é o caso do multi-touch, da geo-referenciação, ou até mesmo de acesso off-line de dados através da cache. Ainda assim, a aproximação à experiência conseguida com um user-interface nativo fica muito aquém: os padrões de usabilidade pretendem apenas possibilitar a sua utilização de uma forma multi-plataforma, misturando conceitos de design de aplicações móveis das mais variadas plataformas. Um utilizador que está habituado aos padrões de interacção da plataforma Android poderá não estar familiarizado com padrões de interacção utilizados em plataformas como o Windows 8, ou o iOS, causando entropia na utilização.

Entre estas duas abordagens, situa-se a híbrida. Esta abordagem agrega características específicas das abordagens nativa e web na medida em que possibilita a utilização de um número limitado de controlos num container que, resumidamente, funciona como uma vista web. As aplicações são desenvolvidas com ferramentas web (edição de HTML, CSS, Javascript, entre outras) e lançadas no dispositivo após serem traduzidas para código nativo.

Esta tradução é limitada devido às capacidades que, por exemplo, o HTML5 oferece, comparando com todas as que o dispositivo suporta.

### 2.2.2 Análise Qualitativa

A análise qualitativa efectuada pretendeu avaliar cada uma das frameworks tendo como guia as seguintes métricas:

- Produtividade;
- Flexibilidade/capacidades do sistema;
- Interoperabilidade;
- Suporte;
- Preço.

A avaliação transcrita na Tabela 2.1 foi efectuada tendo como escala os valores entre 1 (Mau) e 5 (Excelente), sendo que o total é calculado a partir da média aritmética das notas obtidas. Considerar-se-á uma nota positiva como sendo maior ou igual a 3. Esta avaliação foi feita através da execução de tutoriais e respectivo desenvolvimento de código para as frameworks referidas. De salientar também que a escolha da atribuição de igual peso para cada uma das guias deveu-se ao facto de, no decorrer do desenvolvimento de código para cada uma das frameworks, me ter deparado com a estreita relação entre a qualidade e quantidade de suporte existente e a produtividade e performance de desenvolvimento de uma aplicação.

#### Produtividade

Para medir a produtividade de uma framework, foi tido em consideração a linguagem utilizada e a rapidez de adaptação que está associada a esta. Esta avaliação foi efectuada através da execução de tutoriais e demonstrações de aplicações para determinar o grau de complexidade de produção de código. Neste ponto destacaram-se as frameworks Android SDK (nota 5), Adobe Flex (nota 5) e Marmalade (nota 4). As duas primeiras utilizam o Java e a terceira o C++. Estas duas linguagens estão quase sempre presentes no percurso académico, logo a curva de aprendizagem e de adaptação às frameworks apresenta um declive residual. Na Tabela 2.2 podemos ver as linguagens suportadas por cada framework. Este foi um ponto essencial na análise da produtividade.

	Prod.	Flex.	Interop.	Suporte	Preço	Total
<b>Nativo</b>						
Android SDK	5	5	3	5	5	4.67
Adobe Flex	5	3	5	4	3	4.00
Marmalade	4	4	5	3	3	4.00
Mono	3	4	4	3	3	3.50
<b>Web-based</b>						
Sencha	3	4	5	4	4	3.83
The M Project	3	2	5	3	5	3.50
jQuery	3	2	5	3	5	3.50
iUI	3	2	5	3	5	3.50
SproutCore	3	3	5	4	5	3.83
Wink	3	4	5	3	5	3.83
<b>Híbrido</b>						
PhoneGap	3	4	4	4	5	3.83
Titanium Mobile	3	3	4	3	5	3.67

Tabela 2.1: Análise qualitativa das frameworks

	C++	Java	HTML5	JScript	.NET	Obj. C	ActionScript3	C#
<b>Nativo</b>								
Android SDK		X						
Adobe Flex							X	
Marmalade	X							
Mono					X			X
<b>Web-based</b>								
Sencha			X	X				
The M Project			X	X				
jQuery			X	X				
iUI			X	X				
SproutCore			X	X				
Wink			X	X				
<b>Híbrido</b>								
PhoneGap				X	X			
Titanium Mobile			X	X		X		

Tabela 2.2: Linguagens utilizadas nas frameworks

### **Flexibilidade / capacidades do sistema**

Esta métrica pretende medir a capacidade que cada framework tem para tirar partido de todo o software e hardware que o dispositivo possui. Uma nota 1 significa que a framework utiliza de uma forma muito pouco satisfatória as características e componentes disponibilizados pelo dispositivo móvel; uma nota 5 significa uma utilização total de todas as propriedades do dispositivo móvel. Como é possível observar na tabela, a maioria das frameworks de desenvolvimento de aplicações web-based obtiveram notas pouco satisfatórias visto estarem limitadas ao que o HTML5 e/ou ao que o browser do dispositivo permite. Nas soluções nativas, o destaque vai para a fraca prestação do Adobe Flex. Isto deveu-se ao facto de aplicações desenvolvidas com esta plataforma estarem limitadas a versões recentes do Android (apenas a partir da versão 2.2).

### **Interoperabilidade**

A interoperabilidade de cada framework está associada à possibilidade de reutilização de código, seja entre dispositivos com o mesmo sistema operativo (smartphones e tablets), seja com sistemas operativos diferentes (Android e iOS); e também à possibilidade de integração de serviços externos ao dispositivo (suporte para SOAP, REST e webservices).

O Android SDK perde alguns pontos pelo simples facto de ser totalmente nativo, impossibilitando a sua reutilização em outros dispositivos que não corram Android. Por outro lado, o facto de ter nota positiva deve-se à facilidade de não ser necessária qualquer duplicação de código para desenvolver a mesma aplicação para dispositivos de dimensões diferentes, bem como suportar nativamente o acesso a webservices e conseguir consumir serviços SOAP e REST através de bibliotecas desenvolvidas pela comunidade.

Todas as outras frameworks, ou permitem o deployment multiplataforma, ou são web-based (correm em qualquer máquina com um browser e acesso à internet, o que leva a terem melhor prestação que o Android SDK).

### **Suporte**

Hoje em dia, o sucesso no desenvolvimento de uma aplicação está profundamente associado à quantidade e qualidade de informação disponível sobre o desenvolvimento com determinada framework. Esta informação é geralmente providenciada pela comunidade de utilizadores dessas ferramentas e, por vezes, pelos developers dessas mesmas frameworks.

Neste aspecto, o Android SDK (nota 5) apresentou-se como a opção mais rica em conteúdo disponibilizado, seja na forma de tutoriais e vídeos dedica-

dos ao assunto, seja na forma de um acompanhamento permanente por parte de um grande número de developers de aplicações Android nas mais diversas páginas web.

No segundo lugar encontram-se as frameworks que possuem um grande suporte por parte dos seus criadores (Adobe Flex) ou uma comunidade substancial (Sencha, SproutCore e Wink).

### Preço

Naturalmente, o preço de utilização de uma ferramenta de desenvolvimento de aplicações Android terá de ser tido em conta para a sua escolha. Foi proposto pela Maisis, no início deste estágio, que se encontrasse um equilíbrio entre o preço da ferramenta a utilizar, a facilidade de utilização e a constante actualização da mesma em relação à plataforma alvo.

Conforme é possível observar na tabela, as frameworks de utilização gratuita apresentam notas irrepreensíveis. Quanto às restantes, há que salientar o facto de se conseguir determinadas regalias com a sua aquisição (como é o caso do acesso ao suporte do Marmalade), ou o acesso a ferramentas adicionais que apoiam o desenvolvimento (o Architect do Sencha).

### 2.2.3 Análise Funcional

Com a análise funcional de cada uma das frameworks, foi-me possível identificar quais as que se apresentaram como as melhores opções para o desenvolvimento de aplicações.

Os testes de desenvolvimento efectuados passaram pela criação de ecrãs de aplicação e respectiva navegação entre eles, integração de dados externos (provenientes tanto de bases-de-dados como de ficheiros XML), de API's externas (Google Maps, Bing Maps), de web-views; e detecção e tratamento de eventos despoletados pelo toque.

No conjunto para a abordagem nativa, as frameworks Android SDK e Adobe Flex foram as que se evidenciaram, não só pelas potencialidades demonstradas e facilidade de desenvolvimento, como pela qualidade do suporte conseguido, seja fornecido pela comunidade, como pelos developers das frameworks.

O Android SDK, como plataforma mais madura, tem vindo a sofrer profundas melhorias, seja em termos de ferramentas de desenvolvimento, seja pelo contínuo suporte por parte da Google, seja pela inigualável quantidade de informação existente.

A opção da Adobe, o Flex, juntamente com o Adobe Flash Builder, oferece vantagens em termos de usabilidade em relação a uma aplicação nativa



desenvolvida com o Android SDK, perdendo apenas na compatibilidade com os dispositivos móveis (o Adobe Air, plataforma que suporta o deployment das aplicações desenvolvidas pelo Flex, só se encontra disponível a partir da versão 2.2 do Android e para um limitado número de dispositivos). Por este facto, esta opção foi deixada de ser tida em consideração, visto que limita muito o número de dispositivos alvo para a nossa aplicação.

De salientar também a qualidade do Marmalade, que fica aquém das duas primeiras pelo simples facto de ser uma opção bastante cara e de o suporte estar directamente associado ao licenciamento obtido. Apesar disso, o facto de ser suportado nativamente por um vasto leque de sistemas operativos faz desta uma opção a ter em consideração. O Marmalade mostrou-se principalmente indicado para desenvolvimento de aplicações de entretenimento em que não é prejudicial para a experiência do utilizador o facto de os interfaces serem diferentes dos usados pelo sistema operativo.

Para a abordagem web-based, a framework Sencha mostrou-se bastante robusta, visto possuir melhores ferramentas de desenvolvimento que as outras e poder trabalhar em conjunto com o PhoneGap para obter melhores resultados em termos de usabilidade e de integração com o dispositivo anfitrião.

#### 2.2.4 Considerações

Após efectuar as Análises Genérica, Qualitativa e Funcional, é possível salientar, do lado das soluções nativas, o Android SDK, o Adobe Flex e o Marmalade; e do lado das soluções web-based o Wink, o Sencha e o SproutCore.

O facto de se pretender desenvolver uma aplicação que ofereça ao utilizador uma experiência que tire partido de todo o hardware e software disponível pelo dispositivo anfitrião sem o prejuízo de o seu desenvolvimento necessitar de uma maior abrangência de conhecimentos, faz com que a abordagem nativa seja o caminho a tomar.

De entre as opções de desenvolvimento nativo, e como o Adobe Flex se revela limitado em termos do número de dispositivos que poderão acolher aplicações desenvolvidas com esta ferramenta, decidiu-se completar este estudo tendo apenas em conta o Android SDK e o Marmalade. Estas duas frameworks apresentaram-se ao longo destas análises como as soluções mais consistentes.

#### 2.2.5 Análise final

Tendo em conta os estudos efectuados sobre todas as frameworks relevantes para o desenvolvimento de aplicações Android, é possível afirmar que o An-

droid SDK se evidenciou como a plataforma que melhores condições apresenta.

O facto de ser uma solução sem custos associados à obtenção de ferramentas de desenvolvimento de aplicações; de possuir uma enorme comunidade com um vasto know-how baseado na experiência; de oferecer suporte por parte dos developers do sistema operativo; da integração de eficazes ferramentas de debugging, de criação de interfaces com o utilizador, de simulação do ambiente Android (apesar de ser de uma forma condicional para tablets); faz do Android SDK a framework que mais garantias oferece para um desenvolvimento célere. Este aspecto também é potenciado pelo facto de se utilizar a linguagem Java.

De salientar também a facilidade de integração de conteúdos exteriores à framework, na forma de bibliotecas, que aumenta as potencialidades de desenvolvimento e alarga o leque de conteúdos a ser integrados numa aplicação.

A framework Android SDK possibilita também uma total e sempre actualizada compatibilidade com os dispositivos anfitriões, proporcionando ao utilizador uma experiência mais rica de interacção com o smartphone ou tablet, minimizando o tempo necessário para que este se adapte às suas características. Leva também a que o hardware seja sempre reconhecido e utilizado, seja qual for a versão da API que o dispositivo suporta ou utiliza. Sempre que se verifique alguma incoerência entre padrões de usabilidade entre APIs de desenvolvimento, será necessário o desenvolvimento de bibliotecas que possibilitem a total concordância na experiência obtida entre dispositivos de diferentes versões; podendo também passar pela inclusão de bibliotecas já existentes e disponibilizadas pela/para a comunidade.

## 2.3 Analytics

### 2.3.1 A importância da monitorização

Informação sobre informação - esta é a premissa que cada vez se torna mais importante no que toca ao desenvolvimento de uma ideia, de um negócio, do início de uma empresa, do início do desenvolvimento de um produto, da sua comercialização, da sua aceitação e, finalmente, da sua consagração. Mas como poderá ser possível seguir estas etapas o mais rapidamente possível sem que a ideia original se perca pelo caminho, sem que aquela ideia deixe de estar por trás daquele grande produto?

A Informação que acompanha um produto é o factor primordial. Sem a Informação sobre a informação, será impossível saber se o mercado alvo pretende consumir determinado recurso. Poderá igualmente responder a per-

guntas como quando, como e onde o pretende fazer.

Com toda esta informação, a gestão de um produto ou negócio é comparativamente facilitada relativamente a soluções carentes de informação: ajusta-se um ecrã inicial para apresentar produtos que mais atraem o mercado-alvo, propõe-se operações que são mais utilizadas pela maioria dos utilizadores, monitoriza-se a expansão geográfica dos serviços prestados, a área de interesse de determinada faixa etária... Resumidamente, obtém-se Data Insight.

### 2.3.2 O conceito

A análise de dados web tira partido de tecnologia de filtragem de dados para que seja de fácil percepção a informação, o tipo de dados, as operações e o contexto em que determinado utilizador acede a determinado recurso.

As formas mais comuns de análise de dados são:

- Page Tagging (Marcação de páginas);
- Logging (Registos de utilização).

#### Page Tagging

Esta técnica de análise de dados executa do lado do cliente e recorre a tags (snippets geralmente de Javascript) que invocam métodos de captura de informação do utilizador.

Vantagens	Desvantagens
Permite ultrapassar servidores de proxy e caching	Incapaz de ultrapassar firewalls, caso a tag esteja restringida
Permite capturar eventos invocados do lado do cliente (Javascript, Flash,...)	Informação extremamente volátil
Permite capturar dados relacionados comércio-electrónico	Incapaz de monitorizar quantitativamente transferências de dados
Captura de informação do utilizador efectuada em tempo-real	

Tabela 2.3: Vantagens e desvantagens do uso do Page Tagging[1]

#### Logging

O Logging é uma técnica associada à inclusão de módulos (desenvolvidos interna ou externamente a uma empresa) no servidor de determinada plataforma, que são responsáveis pela captura e persistência de dados relativos a transacções de informação.

Vantagens	Desvantagens
Fácil acesso a histórico de dados e operações	Limitada por proxy/caching
Sem restrições impostas por firewalls	Incapaz de capturar eventos client-side
Permite monitorizar qualitativa e quantitativamente transferências de dados	Necessidade de recursos de armazenamento de dados
Permite monitorizar acessos de todos os tipos (web, mobile, etc.)	

Tabela 2.4: Vantagens e desvantagens do uso do Logging[1]

### 2.3.3 Ferramentas

#### Page Tagging

Para demonstrar as funcionalidades associadas a uma ferramenta de análise de dados web para este tipo de técnica, escolhi o conhecido Google Analytics. O Google Analytics<sup>14</sup> é um serviço disponibilizado pela Google que gera estatísticas sobre visitas a determinadas páginas na internet, o que faz deste um produto tendencialmente dirigido ao Marketing.

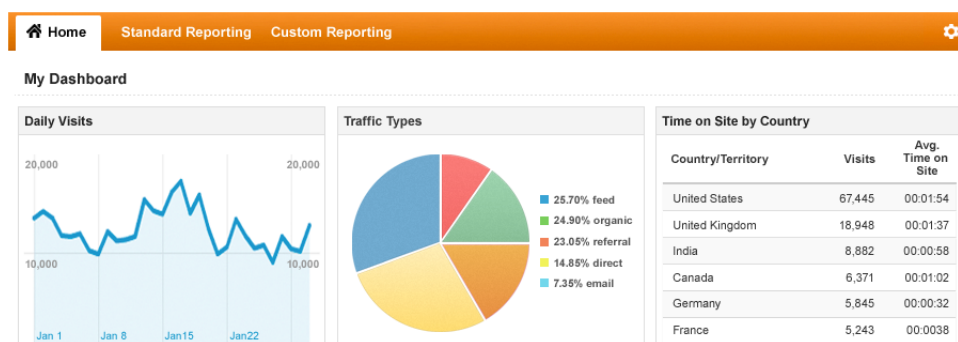


Figura 2.2: Exemplo de uso do Google Analytics

Este sistema fornece um vasto leque de ferramentas de análise, como relatórios em tempo real de acontecimentos sobre determinada página monitorizada, relatórios personalizáveis em que o gestor defini quais as métricas a usar na captura de informação, variáveis personalizáveis em que o gestor é quem define o que qualificável e quantificável; permite uma segmentação do tráfego de operações; permite uma localização do utilizador na estrutura navegável de uma dada página.

<sup>14</sup><http://www.google.com/analytics/>

## Logging

Para a técnica de Logging, a ferramenta escolhida foi o Piwik<sup>15</sup>. O Piwik é uma ferramenta open-source que trabalha sobre um servidor web PHP/MySQL.

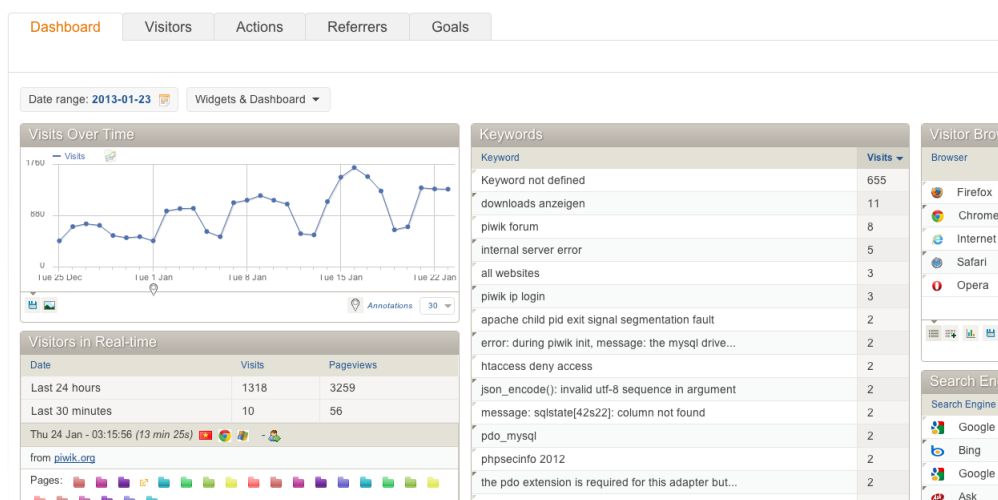


Figura 2.3: Exemplo de uso do Piwik

O Piwik possibilita a consulta de informação capturada através de relatórios enviados por mail em tempo real, possibilita a inclusão de plugins, possui uma API para acesso aos seus serviços web e é facilmente integrável em sistemas existentes.

### 2.3.4 Considerações

A utilização de técnicas de captura de dados no âmbito do desenvolvimento do módulo OObian Analytics procederá à inclusão de elementos adicionais à camada de serviços do servidor OObian. Assim, o módulo OObian assentará sobre a técnica de Logging, em detrimento da técnica de Page Tagging. A técnica de Logging permitirá proceder ao levantamento de dados consultados, operações realizadas sobre dados, utilizadores que consultaram determinada informação, localização espaço-temporal da consulta de determinada informação, bem como outras características de transferência de informação incluídas no Data Insight.

<sup>15</sup><http://piwik.org/>

# Capítulo 3

## Definição de requisitos

### 3.1 Metodologias de Análise de Requisitos

#### 3.1.1 ISO/IEC 9126

A metodologia ISO/IEC 9126, criada pela ISO (International Organization for Standardization) em 1991, surgiu com a necessidade de criar um standard na avaliação de produtos de software através de modelos de qualidade. Assim, foram apuradas seis áreas de avaliação, em que são definidas as respectivas métricas:

Functionality	Reliability
Suitability	Maturity
Accuracy	Fault Tolerance
Interoperability	Recoverability
Security	
Usability	Efficiency
Understandability	Time Behaviour
Learnability	Resource Behaviour
Operability	Operability

Maintainability	Portability
Analyzability	Adaptability
Changeability	Instability
Stability	Conformance
Testability	Replaceability

Tabela 3.1: Requisitos ISO/IEC 9126

### 3.1.2 FURPS

O modelo FURPS, proposto por Robert Grady da Hewlett-Packard Co., divide as características de qualidade de um software em dois grandes grupos: requisitos funcionais e requisitos não-funcionais.

Functionality (Funcionalidade)	Reliability (Confiabilidade)
Joint of characteristics	Frequency and severity of failures
Capacities	Recovery to failures
Security	Time among failures
Usability (Usabilidade)	Performance (Desempenho)
Human factors	Velocity
Aesthetic	Efficiency
Consistency in the user-interface	Availability
Aid in line	Time of answer
Assistant	Time of recovery
Documentation of the user	Utilization of resources
Material of training	
Suportability (Suportabilidade)	
Installability	
Compatibility	
Adaptability	
Configurability	
Portability	
Testability	

Tabela 3.2: Requisitos FURPS[2]

Neste modelo, consideram-se funcionais os requisitos relacionados com a funcionalidade do sistema. Os requisitos não-funcionais são todos os que

são caracterizados por atributos de usabilidade, confiabilidade, desempenho e suportabilidade.

### 3.1.3 FURPS+

O modelo FURPS+ surgiu com a necessidade de colmatar a falha na apresentação das limitações a que o desenvolvimento de um produto de software está associado e que não estão presentes no modelo antecessor (FURPS).

Design constraints	Interface constraints
Restrictions on how a system should be designed, such as the imposition of the use of a certain technology, a certain library	External elements with the system should interact Restrictions in format, time, or other factors
Implementation constraints	Physical constraints
Standards they require Languages of implementation Integrity politics of databases Limit of resources Environments of operations	Restrictions imposed by the hardware and how it was built

Tabela 3.3: Requisitos adicionais FURPS+[2]

O “+” em FURPS+ tem como objectivo relembrar a inclusão dos requisitos de Design, Implementação, Interface e Físicos. Com este modelo, foi possível apurar quatro tipos de requisitos não-funcionais ao modelo FURPS.

### 3.1.4 Metodologia de análise de requisitos a usar

Os requisitos descritos neste documento estão organizados segundo o modelo FURPS+, tendo em conta que, com este modelo, é possível capturar todas as restrições associadas a uma aplicação móvel que irá ser executada num alargado leque de sistemas anfitrião (smartphones, tablets, diferentes capacidades de processamento, diferentes dimensões e definições de ecrã, etc.). É também proposta uma organização destes requisitos por componentes do sistema, que em conjugação com o modelo FURPS+ permite capturar os requisitos de software com uma maior facilidade e organização.



## 3.2 Análise de Requisitos

### 3.2.1 Funcionalidades Macro

Após uma análise cuidadosa, foram identificados os seguintes componentes de sistema:

- Infra-estrutura/ Geral
- Perfil do utilizador
- OObian Reader
- OObian Analytics
- OObian Maps

A versão do OObian Android é um produto trimmed-down de funcionalidades existentes no OObian Insight, com a inclusão da componente de Analytics, que será introduzida no ecossistema OObian e que poderá ser reaproveitada, tanto para o cliente web, como para outros clientes ubíquos. Foi feito um estudo das funcionalidades mais interessantes e chegou-se às seguintes Funcionalidades Macro:

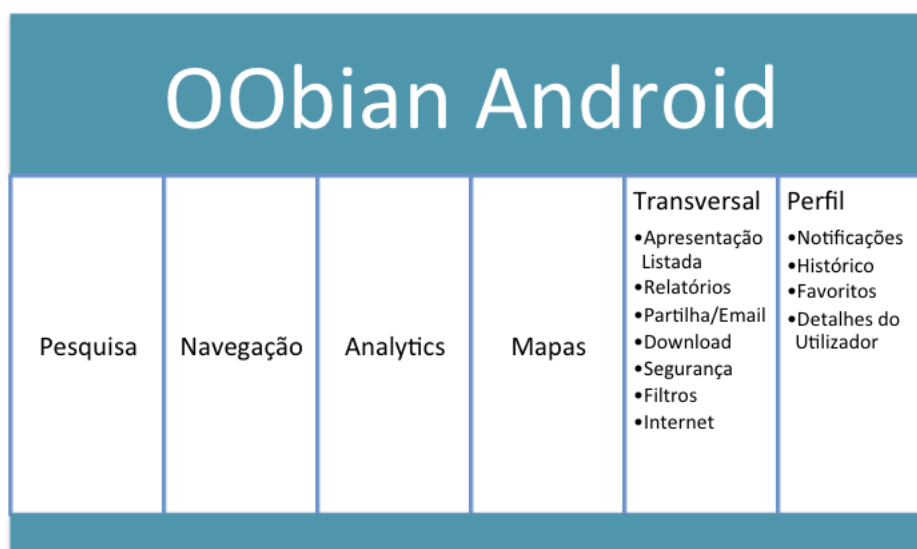


Figura 3.1: Funcionalidades Macro da aplicação OObian Android

Como já era esperado, a Pesquisa, Navegação, Analytics, Mapas e outras funcionalidades de apoio ao produto foram incluídas como obrigatórias para

o desenvolvimento do OObian Android. Estas são as funcionalidades mais valiosas do produto.

### 3.2.2 Requisitos

#### Geral

**Navegação na estrutura de conhecimento** A aplicação OObian Android deverá permitir ao utilizador uma navegação na estrutura de conhecimento: A navegação deverá seguir uma abordagem drill-up/drill-down ou por relacionamento com o nó visitado.

**Pesquisa na estrutura de conhecimento** A aplicação OObian Android deverá suportar a pesquisa de conteúdos presentes na estrutura de conhecimento: Deverá ser possível fazer pesquisas em toda a aplicação OObian Android e em qualquer momento.

**Partilha de conteúdos nas redes sociais** A aplicação OObian Android deverá permitir a partilha de conteúdos nas redes sociais: Sempre que um utilizador achar oportuno, deverá ser-lhe permitido partilhar as suas consultas na estrutura de conhecimento.

**Disponibilizar histórico de pesquisas** O OObian Android deverá criar um histórico de pesquisas para o utilizador: Deverá ser permitido ao utilizador o acesso a uma listagem de termos e conteúdos pesquisados anteriormente.

**Gerir conteúdos favoritos** A aplicação OObian Android deverá suportar a gestão de favoritos: Deverá ser facultado ao utilizador a possibilidade de adicionar ou eliminar conteúdos favoritos.

**Recepção de notificações** O OObian Android deverá recorrer a notificações: O utilizador deverá poder ser notificado sobre a actualização de conteúdos ou outros eventos relevantes.

**Garantia de segurança nas comunicações entre a aplicação e o servidor** O OObian Android deverá garantir a segurança nas comunicações com o servidor: O canal de transferência de dados entre a aplicação e o servidor deverá ser totalmente seguro. A aplicação OObian Android terá que garantir a segurança dos dados trocados com o servidor: Os dados transferidos

entre o cliente e o servidor deverão ser encriptados para se garantir a sua privacidade.

**Transparência de exceções do lado do utilizador** A aplicação OObian Android deverá tornar transparentes para o utilizador as exceções que ocorreram: O utilizador não deverá ser importunado por falhas transitórias que ocorram.

**Minimização de tempos de acesso à estrutura de conhecimento** A aplicação OObian Android deverá minimizar os tempos de acesso à estrutura de conhecimento: O utilizador deverá poder aceder a conteúdos presentes na estrutura de conhecimento no menor espaço de tempo possível. As melhores práticas de Usabilidade[4] recomendam que em média o tempo de um qualquer pedido não deverá exceder os 15 segundos, sendo o tempo ideal até aos 5 segundos.

**Disponibilização de ajuda à utilização da aplicação** O OObian Android deverá permitir ao utilizador aceder a um ecrã de ajuda: O utilizador deverá poder esclarecer as suas dúvidas em relação à interacção com a aplicação a partir de um ecrã de ajuda.

**Redução do número de interacções (teclas ou touch-events) com o dispositivo móvel para operações básicas da aplicação** A aplicação OObian Android deverá reduzir o número de interacções do utilizador com o dispositivo móvel: As operações básicas da aplicação deverão ser atingíveis com o menor número de teclas pressionadas ou toques no ecrã. As melhores práticas de Usabilidade[4] recomendam que as operações básicas de uma aplicação deverão minimizar o número de interacções com o utilizador, em média considera-se que o utilizador deverá executar uma qualquer operação básica: pesquisa e navegação em 3 ou menos interacções.

**Consistência entre o interface da aplicação e os standards do sistema operativo Android** A aplicação OObian Android deverá garantir a consistência entre o interface da aplicação e os standards do sistema operativo Android: O OObian Android deverá dar uso a patterns fornecidas pelo Android SDK para diminuir o tempo de adaptação à aplicação. Recomenda-se também seguir as melhores práticas, recorrendo a padrões de UI do Android disponíveis em <http://www.androidpatterns.com/>.

**Consistência entre o ambiente da aplicação e o experienciado através da aplicação OObian Insight** O OObian Android deverá garantir a consistência entre o ambiente da aplicação e o experienciado através da aplicação OObian Insight: A aplicação OObian Android deverá seguir a mesma linha de design da versão desktop do OObian ou uma muito semelhante que facilite a interação dos utilizadores que já utilizem o OObian Insight.

**Alargamento das possibilidades da aplicação consoante as dimensões e as características do dispositivo anfitrião** O OObian Android deverá alargar as suas possibilidades consoante as dimensões e características do dispositivo anfitrião: A aplicação deverá tirar total partido do equipamento disponível. Isto significa que um determinado conteúdo pode ter múltiplas representações consoante o tamanho do ecrã disponível. Aliás este é um dos pontos explorados por grande parte das aplicações quando se roda o dispositivo móvel.

**Requisitos mínimos para a execução da aplicação no dispositivo móvel** A aplicação deverá correr num dispositivo móvel com os seguintes requisitos mínimos:

Dispositivo	Versão do SO	Tamanho do ecrã	Sensores e Dispositivos
Smartphone	Froyo (2.2)	Normal (4 polegadas)	Acelerómetro, GPS, Acesso a rede 3G ou Wireless
Tablet	Froyo (2.2)	Large (7 polegadas)	Acelerómetro, GPS, Acesso a rede 3G ou Wireless

Tabela 3.4: Requisitos mínimos para executar o OObian Android

### Perfil do Utilizador

**Apresentação dos detalhes do utilizador** A aplicação deverá apresentar os detalhes do utilizador. O OObian Android deverá permitir ao utilizador a apresentação dos seus dados pessoais.

**Edição dos detalhes do utilizador** A aplicação deverá permitir editar os detalhes do utilizador. O OObian Android deverá permitir ao utilizador a edição dos seus dados pessoais, bem como dar-lhe controlo sobre a sua informação presente na plataforma OObian.

**Listagem e gestão de conteúdos favoritos** A aplicação OObian Android deverá listar e permitir a gestão de conteúdos favoritos: O utilizador deverá poder consultar, adicionar e remover itens da sua lista de favoritos.

**Listagem e gestão de subscrições de notificações** A aplicação OObian Android deverá listar e permitir a gestão de subscrições de notificações: O utilizador deverá poder consultar, adicionar e remover itens da sua lista de subscrições de notificações.

**Gestão de fontes de informação externa** A aplicação OObian Android deverá permitir a gestão de fontes de informação externa: O utilizador deverá poder adicionar, editar e remover as fontes de informação presentes na internet.

### Rich Knowledge Reader

**Pesquisa com janela de desambiguação de resultados em tempo real** O OObian Android deverá permitir o início de navegação a partir da pesquisa de dados presentes na estrutura de conhecimento, com o apoio de uma janela de desambiguação de resultados: O utilizador, à medida que digita na caixa de texto de pesquisa, deverá ser confrontado com uma janela de desambiguação de resultados em tempo real.

**Navegação a partir da raiz da estrutura de conhecimento** O OObian Android deverá permitir a navegação a partir da raiz da estrutura de conhecimento: O utilizador deverá ter a possibilidade de iniciar a navegação na estrutura de conhecimento a partir do seu nó inicial.

**Retorno de resultados em forma de nós da estrutura de conhecimento** A aplicação OObian Android deverá retornar os resultados em forma de nós da estrutura de conhecimento: O utilizador deverá poder navegar a partir de qualquer nó pesquisado para os seus relacionamentos.

**Relacionamento entre o nó em consulta e a estrutura de conhecimento** O OObian Android deverá garantir a existência de relacionamento entre o nó em consulta e a estrutura de conhecimento: Deverá ser sempre permitido ao utilizador consultar e navegar para nós relacionados com o visitado.

**Apresentação dos detalhes do nó em consulta** A aplicação OObian Android deverá apresentar os detalhes do nó em consulta: O utilizador deverá poder localizar o nó em consulta na estrutura de conhecimento, bem como a informação que detém e o tipo de dados de que se trata.

**Suporte para filtragem de resultados, consoante os conteúdos retornados pela pesquisa** A aplicação OObian Android deverá suportar filtragem de resultados, consoante os conteúdos retornados pela pesquisa: O utilizador poderá, caso deseje, filtrar os resultados obtidos através de uma pesquisa. Esses filtros deverão variar consoante os dados obtidos. Para além disso o utilizador poderá filtrar os resultados de uma pesquisa através de keywords. Estas keywords deverão ser usadas para uma sub-pesquisa.

**Integração com o OObian Maps para nó geo-referenciado** A aplicação OObian Android deverá permitir a integração com o OObian Maps para nós geo-referenciados: O utilizador deverá poder navegar para o OObian Maps caso o nó visitado seja geo-referenciado.

**Apresentação de conteúdos multimédia** O OObian Android deverá apresentar conteúdos multimédia: A aplicação deverá possibilitar a visualização dos elementos multimédia que estejam associados a determinado nó.

**Geração de relatórios** O OObian Android deverá possibilitar a geração de relatórios de contexto: A aplicação deverá possibilitar a geração de criação de um relatório contextual para a posição do utilizador perante a estrutura de conhecimento.

**Download de documentos** O OObian Android deverá possibilitar o download de documentos: A aplicação deverá possibilitar a descarga de documentos para o dispositivo móvel.

## OObian Analytics

**Agrupamento e apresentação de resultados de navegação a partir de todo o ecossistema OObian** O OObian Android deverá permitir a visualização de informação categorizada (agrupada por ano, classe ou atributo mais relevante) relativa à navegação na estrutura de conhecimento. Esta informação deverá ser apresentada em forma de infografia interactiva.

**Agrupamento e apresentação de resultados de navegação do utilizador** O OObian Android deverá permitir a visualização de informação categorizada relativa à navegação do utilizador na estrutura de conhecimento.

**Integração com o OObian Reader para classes e nós de consulta** O OObian Android deverá permitir a navegação para o OObian Reader a partir de resultados apresentados no OObian Analytics.

**Integração com o OObian Maps para nós geo-referenciados** O OObian Android deverá permitir a visualização de informação categorizada relativa à navegação na estrutura de conhecimento através de um mapa interactivo.

### OObian Maps

**Visualização de dados geográficos da estrutura de conhecimento** A aplicação OObian Android deverá permitir a visualização de dados geográficos da estrutura de conhecimento: O utilizador deverá poder consultar informação geo-referenciada da estrutura de conhecimento através de um mapa interactivo e de uma vista de realidade aumentada.

**Navegação no mapa** O OObian Android deverá permitir a navegação no mapa: Deverão ser oferecidos ao utilizador controlos de navegação no mapa interactivo (zoom, selecção de nós, pesquisa e filtragem).

**Navegação na vista de realidade aumentada** O OObian Android deverá permitir a navegação na vista de realidade aumentada: Deverão ser possibilitados ao utilizador controlos de navegação na vista de realidade aumentada (selecção de nós, pesquisa e filtragem).

**Deslocação do mapa para a geo-referência (localização) do utilizador** A aplicação deverá permitir a deslocação do mapa para a geo-referência do utilizador: O mapa interactivo deverá ser apresentado inicialmente nas coordenadas GPS em que o utilizador se encontra.

**Apresentação de detalhes de objecto seleccionado** O OObian Android deverá apresentar os detalhes do objecto seleccionado no mapa ou na vista de realidade aumentada: Assim que o utilizador seleccione um nó presente na vista, deverá ser apresentada informação relativa a esse mesmo nó.

**Apresentação de detalhes de elemento não presente na estrutura de conhecimento e visível na vista de realidade aumentada** O OObian Android deverá permitir a captura de imagens na vista de realidade aumentada para pesquisa gráfica: Deverão ser possibilitados ao utilizador controlos de captura de imagem na vista de realidade aumentada.

**Suporte para filtragem de resultados, consoante conteúdos retornados pela pesquisa** A aplicação OObian Android deverá suportar a filtragem de resultados, consoante os conteúdos retornados pela pesquisa: O utilizador deverá poder filtrar os resultados de pesquisa consoante os atributos desses nós. O utilizador também poderá filtrar os resultados de uma pesquisa através de keywords. Estas keywords deverão ser usadas para uma sub-pesquisa dentro dos resultados apresentados.

**Interligação com o Rich Knowledge Reader para nó geo-referenciado** A aplicação OObian Android deverá permitir a integração com o OObian Reader: O utilizador deverá poder navegar para o Reader a partir dos nós representados no mapa interactivo e na vista de realidade aumentada, ou nos detalhes de determinado nó.

## 3.3 Descrição do Sistema

### 3.3.1 Navegação

Após a identificação das funcionalidades macro por análise das funcionalidades mais usadas no OObian Insight, segue a apresentação da estrutura de navegação entre funcionalidades no OObian Android.

Assim que a aplicação é carregada, será apresentado um ecrã inicial precedido da introdução de credenciais. A partir daqui, o utilizador pode efectuar uma pesquisa, entrar no ecrã de perfil, verificar se há alguma notificação, navegar para um item favorito, navegar a partir do seu histórico de pesquisas, carregar a vista OObian Analytics ou OObian Maps. Dentro do ecrã de perfil é apresentada uma área polimórfica em que são geridos os favoritos, as subscrições de notificações, as fontes de informação externa e os detalhes do utilizador. A partir dos ecrãs de Pesquisa e de Navegação, o utilizador poderá entrar numa vista de granularidade mínima de um item, que apresenta uma área polimórfica dependente dos conteúdos associados a este. O OObian Maps tanto pode ser carregado em modo stand-alone na aplicação ou directamente relacionado com um detalhe de um item. A vista de filtragem, de favoritos e de notificações podem ser chamadas a partir das vistas



de Navegação/Pesquisa, de Detalhes ou do OObian Maps.

### 3.3.2 Prototipagem

Esta secção pretende materializar, numa fase preliminar e através de wireframes, a navegabilidade e as interacções possíveis da aplicação OOBIAN Android. O wireframing é uma técnica de prototipagem de baixa fidelidade que pretende gerar um guia visual que represente o esqueleto de uma aplicação, e que permite obter uma percepção conceptual do seu funcionamento.

#### Contexto aplicacional

No contexto aplicacional, vai ser recorrentemente dado o uso a padrões e a elementos comuns. Assim, e para que seja mais simples explicar todos esses recursos, seguem as próximas ilustrações.



Figura 3.2: Mockup de ecrã geral da aplicação e de granularidade mínima

A Action Bar suporta operações rotineiras como pesquisas, navegação, localização do ponto actual de navegação na estrutura do conhecimento, entre outras. A zona de contexto incluirá os detalhes mais relevantes de um determinado item que esteja a ser visitado (poderá ser um item ou uma classe da estrutura de conhecimento ou o cabeçalho do perfil do utilizador). Na lista de itens relacionados são apresentados todos os elementos, agrupados por classes, que estão associados ao contexto. Na parte superior desta lista encontra-se a área de classe/categoria. Esta área identificará o grupo de itens

listados. A barra de filtros estará presente ao longo de praticamente toda a aplicação e permite um rápido acesso a esta funcionalidade.

A área de navegação/pesquisa é apresentada cada vez que o utilizador efectua uma pesquisa ou navega na estrutura de conhecimento. Esta vista permite uma visualização vertical dos resultados obtidos, ou seja, é indicada para grandes quantidades de informação, em que pretendemos consultar itens de uma forma mais célere. As classes serão representadas por elementos expandíveis.



Figura 3.3: Mockup de ecrã de pesquisa

A vista de pesquisa, expandida na Action Bar a partir do toque no seu ícone, pretende oferecer ao utilizador uma rápida navegação para um ponto específico na estrutura de conhecimento, caso opte por escolher um resultado de desambiguação, ou para uma vista de navegação/pesquisa em que poderá consultar um alargado número de resultados relacionados.

O menu de contexto de aplicação pretende permitir ao utilizador o acesso rápido a tarefas de média importância (navegação para o ecrã principal da aplicação, de perfil, de notificações ou de favoritos, geração de relatórios) e é expansível através do toque-prolongado no ícone da aplicação. O item é caracterizado por uma linha de uma lista que contenha um título, uma descrição, um thumbnail e está catalogado pelo seu conteúdo. Na Figura3.4 está representado um item que está listado nos favoritos, está subscrito para notificações, possui uma geo-referência e tem documentos associados. Este código será usado para todos os itens da aplicação. Como interacções possíveis, destaca-se o toque simples no item, que levará o utilizador para o ecrã de granularidade mínima desse item; e o toque prolongado, que apresentará uma jumplist de opções associadas.

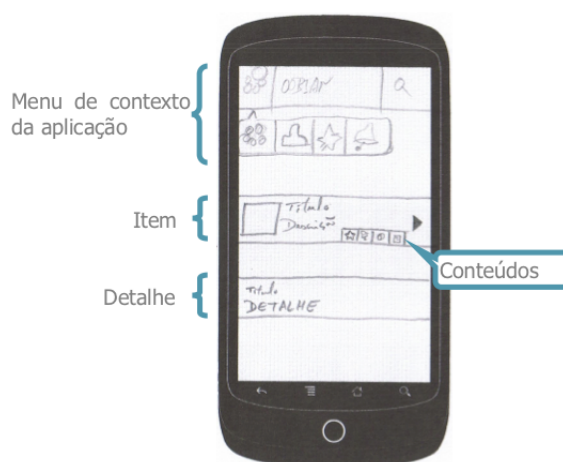


Figura 3.4: Mockup de itens gerais da aplicação

### 3.3.3 Casos de uso

Ao longo da apresentação dos casos de uso, será descrita uma situação da vida real que se aplique, acompanhada de protótipos, com vista a identificar mais facilmente o contexto em que o respectivo caso de uso está incluído. Assim, decidiu-se fazer referência a uma empresa fictícia de comércio de artigos de escritório – XPTO, Import & Export – que terá Jane Doe como responsável administrativo e John Doe como colaborador, também eles fictícios, equipados com dispositivos móveis Android com a aplicação OOBIAN Android previamente instalada. De salientar que a empresa XPTO possui o OOBIAN para gestão da sua estrutura de conhecimento.

#### Acesso ao OOBIAN Android

**Contexto real** *Associado a todas as transacções de informação dentro da XPTO, Import & Export, incluem-se as que são vitais e, imperativamente, de acesso restrito. Assim, para que a informação presente nas bases de dados da XPTO se mantenha sigilosa, John Doe terá de, numa primeira instância, efectuar login, apresentando as suas credenciais à aplicação OOBIAN Android (Figura 3.5). Sempre que pretender terminar a sessão, poderá também efectuar logout e, assim, garantir que a informação na XPTO não é acedida por outros possíveis utilizadores do seu dispositivo móvel. Assim que John Doe inicia a sessão no OOBIAN Android, é apresentada uma lista de notificações sobre conteúdos subscritos alterados, como, por exemplo, a adição de novos clientes da empresa, alteração da geo-referência de determinada empresa cliente ou uma actualização nos documentos relativos aos lucros da*

empresa para o mês anterior (Figura 3.5). John poderá também proceder ao salto para uma pesquisa efectuada no dia anterior, sobre uma tarefa que tenha ficado pendente, ou consultar rapidamente a geo-localização de um cliente que tenha sido adicionada como favorita e que pretenda visitar.



Figura 3.5: Protótipos de ecrã de login e inicial

### Perfil de Utilizador

**Contexto real** A configuração do ambiente de trabalho deverá ser personalizável consoante o utilizador da aplicação. Assim, Jane e John Doe poderão ter diferentes interesses consoante os seus papéis ou cargos na XPTO. Jane Doe, como administrativa, provavelmente terá como prioridade na sua lista de notificações as alterações relativas a documentos de cariz financeiro relacionados com os clientes e os fornecedores. Já John Doe necessitará estar esclarecido em relação aos stocks de material para o fornecimento dos clientes, o estado dos pagamentos para seguimento da assistência, etc. Todos esses pontos são configuráveis, adicionáveis ou removíveis nas listas de notificações e favoritos, navegáveis por swipe sobre a lista. Ambos poderão também alterar as fontes de informação externa (proveniente da internet) que pretendem que esteja disponível na vista de detalhe de um item. Jane poderá estar interessada em ser informada do estado financeiro de possíveis parceiros, clientes ou fornecedores para, assim, tomar decisões administrativas. John poderá ser informado acerca dos melhores produtos para determinado cenário, numa altura em que faz consultoria para um cliente.



Figura 3.6: Protótipo de ecrã de perfil de utilizador

### OObian Reader / OObian Maps

**Contexto real** De modo a facilitar a pesquisa de um limitado número de itens com atributos comuns, John e Jane poderão aplicar filtros aos seus resultados de pesquisa. De uma forma imediata, Jane poderá ter a percepção da distribuição geográfica dos seus clientes para, assim, proceder a ajustamentos relativamente à rede de distribuição dos seus produtos. John poderá assim organizar de uma forma mais efectiva o trajecto a tomar para uma visita aos clientes da XPTO, ou aceder mais rapidamente à lista de clientes que necessitam de restocking de determinado produto. Para aceder a esta vista, basta efectuar swipe para cima sobre a barra de filtros que se encontra presente na zona inferior do ecrã, presente em toda a aplicação, e aplicar os filtros pretendidos.

### OObian Reader

**Contexto real** A navegação poderá ser feita de uma forma bastante intuitiva. Caso John faça uma pesquisa sobre, por exemplo, conteúdo relativo a uma empresa que esteja referenciada por um cliente efectivo, será apresentada neste ecrã toda a informação disponível. John poderá navegar na estrutura de conhecimento da XPTO até atingir o nível que pretenda (Figura 3.8). Num segundo cenário, caso John tenha navegado para um item específico na sua navegação, poderá visualizar o conteúdo relacionado com esse item, desta vez de uma forma mais detalhada, como exemplificado na Ilustração 33. Em qualquer altura, John poderá seleccionar um dado item como favorito para posterior navegação rápida (i.e. dados de facturação relativos a um determinado cliente); adicionar como notificação dados relativos a es-



Figura 3.7: Protótipo de ecrã de filtragem

*tados de pedidos de fornecimento de um cliente para um restocking imediato, enviar uma factura electrónica a um cliente, via e-mail, entre outras.*

### OObian Analytics

**Contexto real** *Jane Doe, como responsável administrativa da empresa XPTO, pretende saber, de uma forma simples, prática e esclarecedora, todas as transacções efectuadas pela sua empresa em determinado período. Ao mesmo tempo, pretende ter uma ideia de como tem evoluído a realização de manutenção em determinada empresa cliente, podendo, assim, apostar no aconselhamento de outros produtos que serão potencialmente interessantes. Da mesma forma, pretende ter uma ideia da expansão geográfica dos clientes da XPTO nos últimos anos. Através do OObian Analytics, será possível qualificar e quantificar, não só dados pesquisados na estrutura de conhecimento, como também o tipo de operação que é efectuada sobre esses mesmos dados. Assim, é possibilitado o total controlo sobre a estrutura de conhecimento, bem como utilizar os dados obtidos para melhorar o serviço prestado.*

### OObian Maps

**Contexto real** *O OObian Maps é uma mais-valia na representação de dados geográficos. Qualquer colaborador da XPTO poderá consultar de uma forma imediata a localização de todos os seus clientes, dos seus fornecedores, o padrão de expansão da empresa, o raio de influência de campanhas publicitárias, etc. Todos estes itens poderão ser isolados através da aplicação de filtros ou efectuando uma pesquisa. Poderá também ser explorado todo o*

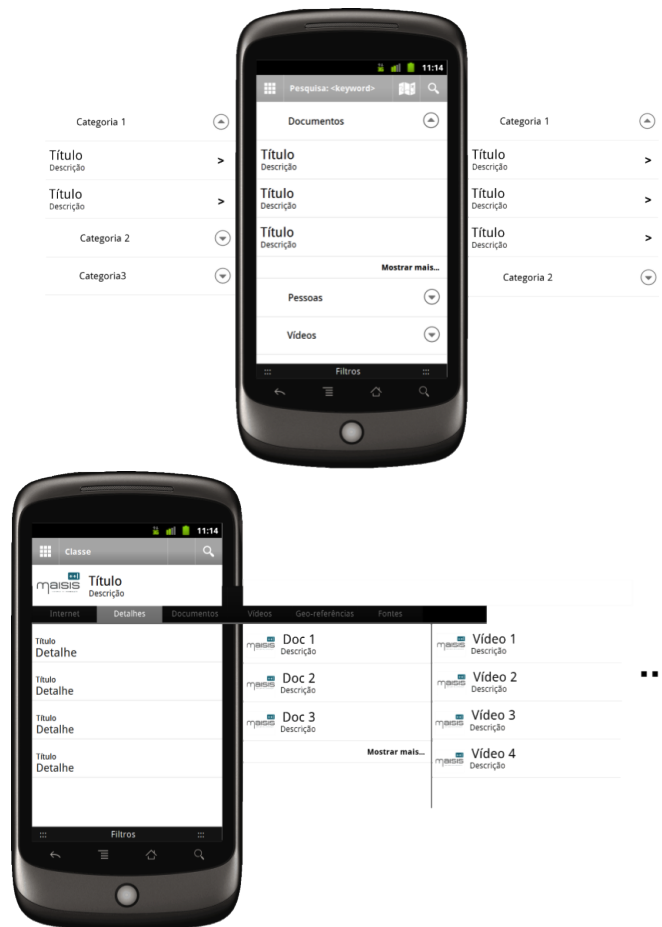


Figura 3.8: Protótipos de ecrãs de pesquisa e granularidade mínima

*mapa representado através de acções de zoom-in, zoom-out e pan, seleccionar determinado cliente etiquetado e visualizar os seus detalhes.*

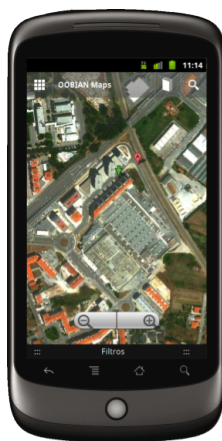


Figura 3.9: Protótipo de ecrã de mapa





# Capítulo 4

## Arquitectura e desenho

### 4.1 Fundamentos

O cliente OObian para dispositivos móveis Android tem como objectivo proporcionar uma experiência o mais próxima possível da proporcionada pelo OObian Insight (cliente web para o gestor de estruturas de conhecimento OObian).

Como foi possível observar na análise de frameworks de desenvolvimento de aplicações Android e no levantamento de requisitos para este cliente, várias foram as condicionantes que se impuseram. De uma forma sintética, optou-se por desenvolver o cliente Android através da framework nativa disponibilizada pela Google (Android SDK). Desta forma será possível tirar partido de todas as capacidades do dispositivo móvel e manter o cliente Android o mais actualizado possível, tendo em conta os ciclos de desenvolvimento deste sistema operativo móvel.

Ao nível físico, esta plataforma móvel está associada a uma notória fragmentação. Várias são as condicionantes e restrições que se apresentam: diferentes densidades de ecrã, diferentes dimensões, equipamentos com e sem módulo GPS, com e sem suporte para multi-touch, entre outras.

Ao nível lógico, os padrões usados no desenvolvimento da aplicação serão também limitados, em alguns casos. O facto de a fragmentação estar também presente a este nível leva a que se procurem soluções para os casos em que a compatibilidade entre versões do ambiente de desenvolvimento seja inexistente. Dentro do conjunto de versões da API do Android SDK que o OObian Android suportará (entre Gingerbread e Jelly Bean), surgirão incompatibilidades entre os subgrupos pré e pós Honeycomb. Estas incompatibilidades entre padrões serão ultrapassadas no decorrer do desenvolvimento da aplicação OObian Android, tendo como salvaguarda a própria retrocompatibilidade da

API, não se prevendo a sua insolubilidade.

## 4.2 Perspectiva Funcional

De uma perspectiva funcional, a aplicação OObian Android será desenvolvida tendo em conta as guidelines representadas na figura seguinte:

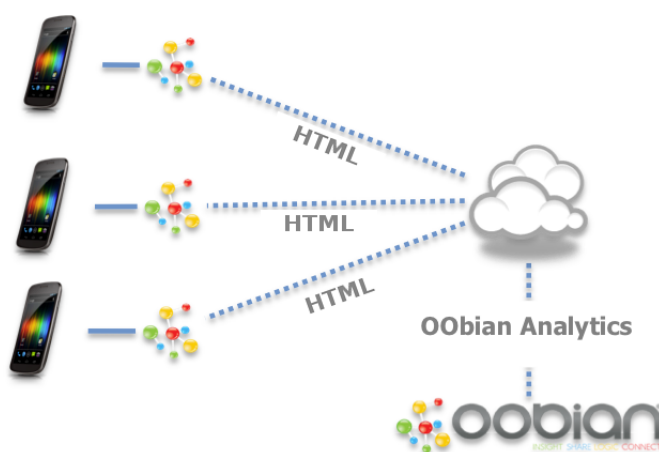


Figura 4.1: Arquitectura geral do OObian Android

A partir da ilustração acima, é possível verificar que se trata de uma aplicação que segue o modelo Cliente-Servidor. A comunicação entre o dispositivo móvel e o servidor OObian é feita através da aplicação OObian Android, que recorre a web-services para fazer a troca de informação. Esta troca de informação será constantemente monitorizada através do módulo OObian Analytics, que correrá paralelamente ao servidor OObian. Este módulo não só irá capturar os dados pedidos e enviados de/para o servidor a partir de qualquer tipo de cliente OObian (Android, iOS, Insight), como também as operações sobre esses mesmos dados. A aplicação permitirá ao utilizador efectuar pesquisas e navegar na estrutura de conhecimento, bem como consultar dados categorizados de consultas e pesquisas, gerir o seu perfil e todas as opções associadas. Algumas operações poderão ser exclusivas para determinados tipos de utilizador. Poder-se-á esquematizar de uma forma breve as funcionalidades da aplicação:

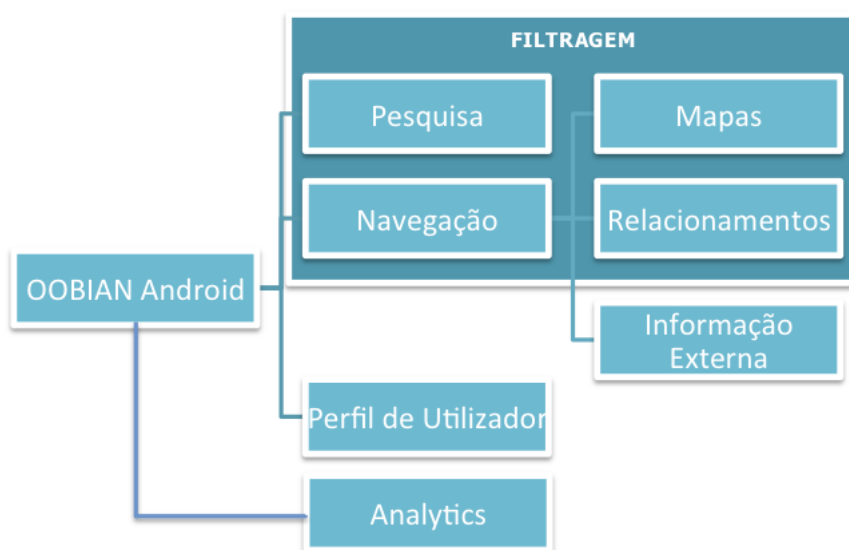


Figura 4.2: Funcionalidades da aplicação OObian Android

De acordo com os conhecimentos adquiridos com o estudo do desenvolvimento de aplicações utilizando a framework Android SDK, prevê-se que aplicação OObian Android terá como base a estrutura geral de uma aplicação Android, sendo adicionados elementos que a melhorarão, tanto em termos de desempenho, como de usabilidade. Preliminarmente, poder-se-á prever a utilização de uma classe para cada actividade: OObian Android (actividade de carregamento da aplicação), Login, Insight (ecrã principal da aplicação), Search, Details, Analytics, Maps, Profile e Bookmarks. Estas classes serão responsáveis pela interligação entre a camada de dados e a camada de apresentação.

De igual forma, serão necessários Adaptadores, visto que teremos de lidar com informação de tipos bastante particulares (classes, instâncias, detalhes de todos os tipos, imagens, etc.) que serão apresentados de formas também elas bastante particulares (listas normais e expandíveis, galerias de imagens, zonas de paginação de vistas).

A informação circulará entre a aplicação OObian Android e o servidor através de dados (de)serializados produzidos e consumidos por serviços web.

A comunicação entre a aplicação OObian Android e o servidor OObian pode ser representada pela Figura 4.3.

Em primeiro lugar, para o dispositivo móvel efectuar determinado pedido, será necessário serializar a informação a ser enviada para o servidor, compactar o pedido para que seja efectuado de forma célere e e finalmente

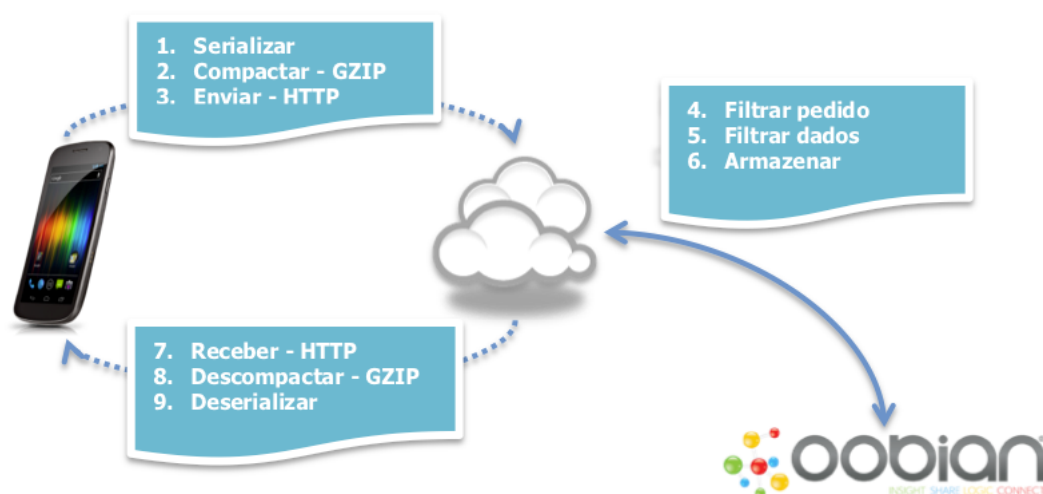


Figura 4.3: Comunicação entre o OObian Android e o OObian

enviar por HTTP.

Assim que o pedido seja feito pelo dispositivo móvel, do lado do servidor é produzida a resposta a enviar e, de uma forma igualmente transparente, é activado o módulo OObian Analytics que irá processar, tanto a informação trocada, como a operação efectuada sobre essa mesma informação. Esta informação captada pelo OObian Analytics será persistida para posterior utilização.

A resposta é enviada pelo servidor OObian.

O dispositivo móvel, ao receber a resposta do servidor OObian, procede à sua descompactação, deserialização e posterior apresentação ao utilizador.

# Capítulo 5

## Desenho da solução e decisões tecnológicas

O desenho da aplicação OObian Android e do módulo OObian Analytics tiveram, invariavelmente, que seguir um caminho traçado, seja através de padrões de desenvolvimento de uma aplicação Android, seja pela forma como a API de um serviço web é projectado e desenhado.

Assim, este capítulo servirá para explicar as decisões que foram tomadas no decorrer do desenvolvimento da aplicação OObian Android, bem como na criação da estrutura preliminar do módulo OObian Analytics.

### 5.1 OObian Android

Conforme descrito no capítulo anterior, o desenvolvimento desta aplicação Android segue os padrões disponibilizados pela Google[5]. Apesar de não se caracterizar totalmente como um modelo MVC (Model-View-Controller), o desenvolvimento Android permite construir uma aplicação em que há uma clara distinção entre as diferentes camadas de funcionalidades de cada elemento usado. Ainda assim, este modelo permite ao desenvolvedor fundir estas camadas de forma a tornar mais simples para si o processo de criação da aplicação.

De grosso modo, e de forma a resumir a forma como o OObian Android se estrutura, decidi descrever a aplicação através de três grupos: a camada de visualização, a camada de tratamento de dados e a camada de dados *per se*.

### 5.1.1 Desenho da Solução

#### Camada de Visualização

Desde o início da realização deste projecto, e ainda no âmbito do estudo das frameworks de desenvolvimento de aplicações Android (efectuado na etapa preliminar deste estágio), que se tornou claro que um dos pontos cruciais para a correcta criação da aplicação OObian Android passaria pela garantia, não só da utilização de padrões de usabilidade definidos pela Google, como também de uma retro-compatibilidade o mais abrangente possível. Para isso, e em alguns casos, foi necessário, tanto recorrer a bibliotecas disponibilizadas pela comunidade, como pelo desenvolvimento dos meus próprios padrões.

Dos padrões nativos usados no desenvolvimento do OObian Android, pode-se contar, entre outros, os seguintes:

- Listview: esta vista tem a capacidade de, a par do seu adaptador (explicado na secção seguinte), possibilitar a apresentação de informação de uma forma listada. Esta vista mostrou-se, ao longo dos tempos, como a imagem de marca de muitas aplicações Android disponibilizadas na plataforma Google Play.



Figura 5.1: Listview

- GridView: a gridview permite, de uma forma semelhante à listview, apresentar informação em mosaico. Esta vista mostra-se especialmente robusta e elementar no desenvolvimento de aplicações direccionadas para um segmento de dispositivos móveis com ecrã de maiores dimensões (tablets, algumas gamas de smartphones, televisões, etc.), tendo em conta que permite fragmentar uma listview em várias colunas.

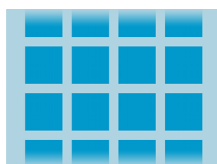


Figura 5.2: Gridview

- **WebView:** esta vista permite embeber conteúdo web dentro de uma aplicação sem que, para isso, o utilizador perca o contexto aplicacional em que esteja inserido. É, também, regularmente utilizado em aplicações desenvolvidas num paradigma web-based.
- **MapView:** este é um padrão de desenvolvimento que apetrecha uma aplicação com a API disponibilizada pelo serviço web Google Maps. Este padrão é, aliás, disponibilizado na forma de plugin, visto que a sua utilização é condicionada pela comercialização ou não da aplicação em que é inserido.
- **Actionbar:** a ActionBar veio a ser introduzida com o lançamento da versão 3.0 (Honeycomb) em que se tentou centralizar operações de contexto aplicacional num único ponto de acesso. Encontra-se no topo de um grande número de aplicações Android.



Figura 5.3: ActionBar

- **TextView:** esta vista permite introduzir elementos textuais a uma aplicação Android. Apesar de bastante simples, este padrão de desenvolvimento afigura-se como um dos sempre presentes no desenvolvimento de qualquer aplicação.

Ainda assim, foi necessário desenvolver padrões não-nativos que foram utilizados ao longo da execução da aplicação. Dentro destes padrões destaca-se:

- **InstanceView:** esta vista permite representar graficamente uma instância da estrutura de conhecimento. Incorpora, não só a informação que identifica determinada instância, como também inclui o seu menu de contexto.
- **ClusterView:** de uma forma semelhante à Instanceview, a Clusterview representa, no OObian Android, uma classe de instâncias.
- **FilterView:** a vista de filtro é um padrão desenvolvido no decorrer deste projecto com o intuito de tornar polimórfica a representação de um objecto de filtragem. Analisando a interface disponibilizada pelo cliente web OObian, verifica-se que um filtro está sempre dependente do tipo de dados a que corresponde. Assim, mostrou-se imperativo



permitir que um mesmo objecto tomasse a forma de uma *sliding-bar* de escolha de um intervalo entre números inteiros, de um *date-picker* para a escolha de determinada data, de uma classe de instâncias, ou até mesmo utilizar uma instância como filtro.

- `DetailView`: permite a representação estática de informação presente em instâncias, classes ou no perfil do utilizador.

### Camada de Tratamento de Dados

A camada de tratamento de dados reflecte todos os padrões de desenvolvimento Android responsáveis pela transformação, tanto proveniente do servidor OObian (através da camada de dados), como do utilizador (através da camada de visualização), para que possa, de forma legível, ser apresentada ao utilizador. Este tratamento é efectuado por adaptadores que, de uma forma dinâmica, mantêm actualizados os elementos da camada de visualização. De grosso modo, cada elemento da camada de visualização tem por trás um adaptador. Como exemplo, uma `ListView` tem associada um `ListAdapter`. Este `ListAdapter` permite que o conteúdo da `ListView` seja alterado pela resposta proveniente de um pedido feito pelo utilizador.

### Camada de Dados

A camada de dados inclui todos os métodos utilizados para a comunicação da aplicação com o servidor OObian e a persistência de dados.

O desenvolvimento da aplicação Android necessitou recorrentemente de persistir informação de forma a, não só aceder a informação de uma forma imediata sem recorrer a pedidos ao servidor, como também de modo a garantir um funcionamento fluido da aplicação.

Dependendo do contexto e da ontologia usada pelo servidor OObian, justifica-se, por vezes, a necessidade de autenticação de utilizadores para acesso ao conteúdo da estrutura de conhecimento. Apesar de o contexto utilizado no desenvolvimento da aplicação OObian Android ter sido um baseado na DBpedia, é também capaz de gerir credenciais de acesso e customização de perfil do utilizador. Neste cenário, foi importante desenvolver um mecanismo que possibilitasse um rápido acesso à aplicação OObian Android, não só por uma questão de conforto para o utilizador, como também de forma a poderem ser utilizadas todas as funcionalidades de pesquisa e de integração no dispositivo móvel que o sistema operativo Android permite.

Por outro lado, a persistência de informação permitiu uma gestão mais eficaz dos recursos necessários para apresentar informação ao utilizador sem

descurar o facto de que estamos perante uma aplicação que faz a gestão de conteúdo muito exigente em termos de consumo de tráfego web. Assim, foi também desenvolvido um mecanismo de cache para conteúdo gráfico (imagens e thumbnails) de forma a evitar pedidos desnecessários para que se aceda a conteúdos transmitidos anteriormente.

Mais concretamente em relação à forma como a informação é persistida num dispositivo móvel com o sistema operativo Android, o sistema de gestão de base de dados usado é o SQLite.

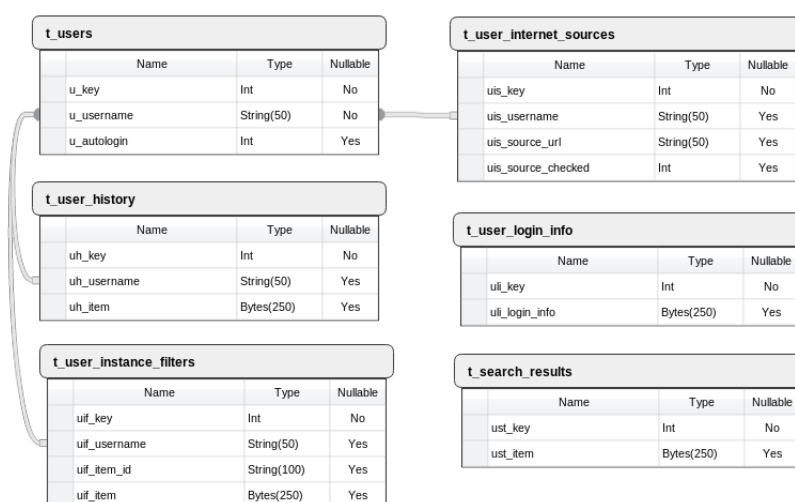


Figura 5.4: Diagrama de base de dados da aplicação OObian Android

### 5.1.2 Decisões Tecnológicas

Para além de todos elementos individuais, foi necessário recorrer a bibliotecas externas, algumas desenvolvidas pela comunidade de desenvolvedores de aplicações Android, outras disponibilizadas pela própria Google.

A ActionBar foi um destes casos. No início do desenvolvimento da aplicação OObian Android deparei-me com o facto de o Android SDK ter algumas falhas em termos de retrocompatibilidade. Alguns padrões que hoje em dia são dados como vitais para o funcionamento e aceitação de uma aplicação, não poderiam ser garantidos em versões mais antigas deste sistema operativo. A ActionBar, tendo surgido com a versão Honeycomb, não garantiu a sua compatibilidade com dispositivos móveis com versões mais antigas do Android. Assim, surgiu a urgente necessidade de adoptar uma biblioteca que desse suporte à ActionBar por parte desta gama mais antiga de dispositivos

móveis. Após alguma pesquisa, surgiu a *Actionbar Sherlock*<sup>1</sup>. Amplamente usada por relevantes aplicações Android presentes na plataforma Google Play, como é o caso do *Foursquare*<sup>2</sup> e *GitHub*<sup>3</sup>, apresentou-se como uma solução viável para o suporte deste padrão de usabilidade.

Uma biblioteca também vital para o desenvolvimento do OObian Android foi o plugin *Google Maps API*<sup>4</sup>, devido à sua componente geo-contextualizável, OObian Maps. Esta biblioteca permite incluir a API do Google Maps em qualquer aplicação Android. Ao longo do desenvolvimento, foi-me possível ter contacto com as duas versões desta API: a V1 e a V2.<sup>5</sup>

Outra funcionalidade à qual era necessário dar suporte foi a inclusão de elementos gráficos numa Mapview. Esta Mapview, alimentada pelo Google Maps API V1, não permitia, à partida, a introdução de elementos complexos (conteúdo dinâmico) nas suas camadas de apresentação e que fossem capazes de representar informação localizada, sem nunca perder a área geográfica em que se navega. Assim, foi necessário optar por uma nova biblioteca: *Android-MapviewBallons*<sup>6</sup>. Esta biblioteca permite, não só introduzir marcadores numa Mapview, como também customizar totalmente os items a apresentar. Tornou-se possível, com esta inclusão, representar, não só marcadores de agregação de instâncias georreferenciadas, como também oferecer ao utilizador informação mais detalhada sobre a instância que está a visitar.

Para criar uma maior fluidez na navegação da aplicação OObian Android, foi necessário recorrer a um padrão de usabilidade muito frequente: o ViewPager. O ViewPager permite navegar horizontalmente em várias vistas de uma forma intuitiva. Depois de adoptado este padrão, foi necessário tornar mais evidente, para o utilizador, aquilo que ele está realmente a visualizar. Para garantir esta contextualização, foi adoptada a biblioteca *Android-ViewPagerIndicator*<sup>7</sup>.

Outra decisão tecnológica de bastante importância para o desenvolvimento desta aplicação foi o facto de ter optado pela utilização de serviços OObian que expectassem/disponibilizassem objectos serializados em Protobuf, em detrimento da troca de objectos JSON ou XML. Esta decisão teve por base o facto de estarmos perante um contexto em que a transmissão de

---

<sup>1</sup><http://actionbarsherlock.com/>

<sup>2</sup><https://play.google.com/store/apps/details?id=com.joelapenna.foursquared>

<sup>3</sup><https://play.google.com/store/apps/details?id=com.github.mobile>

<sup>4</sup><https://developers.google.com/maps/documentation/android/>

<sup>5</sup>No capítulo do Planeamento, secção Análise de Riscos, irei explicar melhor o motivo para esta dupla abordagem.

<sup>6</sup><https://github.com/jgilfelt/android-mapviewballoons>

<sup>7</sup><https://github.com/JakeWharton/Android-ViewPagerIndicator>

dados pelos canais de informação mais recorrentes (3G ou wireless) serem actualmente limitados, tanto em termos de custos associados, como de largura de banda disponível, estando o tamanho das mensagens trocadas entre a aplicação cliente e o servidor directamente associado à utilização destes recursos. Interessa, portanto, minimizar o dispêndio de tráfego. Conforme mais à frente se poderá observar, a serialização Protobuf mostrou-se a opção mais indicada para este contexto. Outra métrica importante verificável nesta escolha passa pela utilização de recursos energéticos. Esta métrica acabou por não ser avaliada para o desenvolvimento desta aplicação por uma questão de exigência em termos temporais para a realização de um estudo desta envergadura. Teve-se sim, em conta, e para além da opinião do orientador do estágio na Maisis, um *benchmark*[6] desenvolvido por Eshay Smith criado com a necessidade de comparar vários tipos de serialização, tendo em conta tempos de serialização e deserialização de dados e tamanho do objecto serializado.

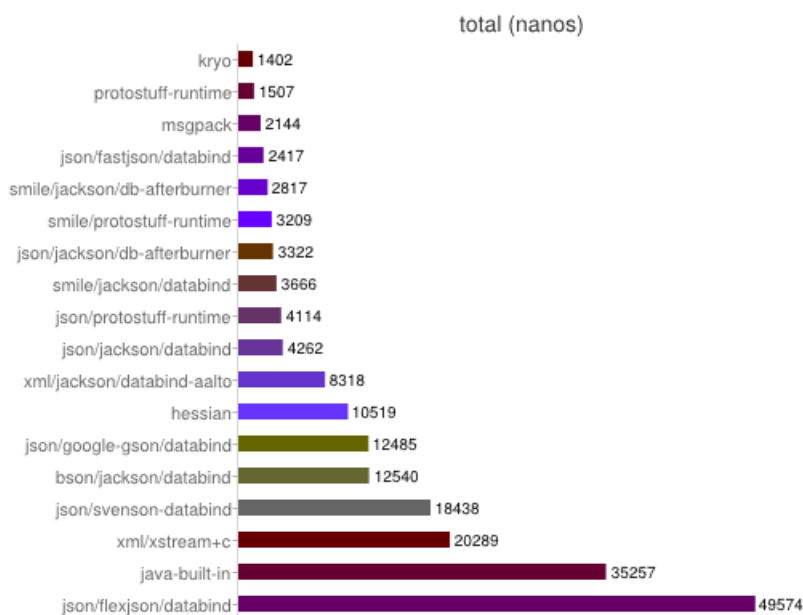


Figura 5.5: Tempos de serialização + deserialização (nanosegundos)

Este estudo visou a utilização de um sistema composto pelo sistema operativo Windows 7 de 64 bit, JVM 1.6.0\_15 JRE 32 bit e um processador Intel Core i7. Representado nas figuras 5.5 e 5.6 está um teste sobre a manipulação de um ficheiro de 719 Byte.

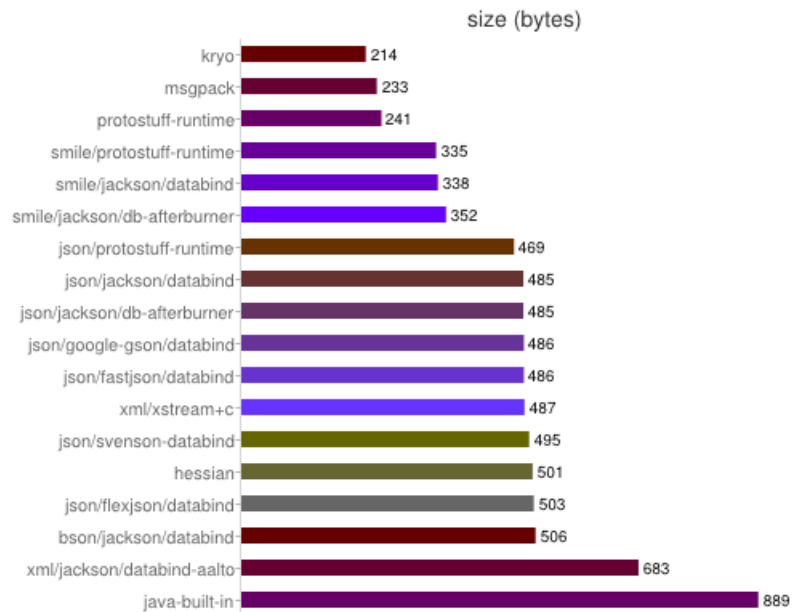


Figura 5.6: Tamanho de objecto serializado (Byte)

## 5.2 OObian Analytics

O OObian Analytics, como módulo de apresentação, tratamento e persistência de dados relativos a navegações dos utilizadores do ecossistema OObian, permite, através de um API REST, aceder a toda a informação persistida.

### 5.2.1 Desenho da Solução

#### Serviços Web

Este módulo permite, não só a qualquer aplicação cliente OObian, como de forma stand-alone, aceder a informação persistida com a navegação efectuada por todos os utilizadores na estrutura de conhecimento presente no servidor OObian. Este acesso é disponibilizado através de uma API REST que permite, não só consumir a informação disponibilizada, como também realizar a captura de informação de uma forma activa.

Os serviços disponibilizados possuem as respectivas assinaturas:

- Criar registo na tabela de logs: (POST) /webresources/logs/
- Editar registo: (PUT) /webresources/logs/
- Eliminar registo: (DELETE) /webresources/logs/id

- Obter registo por identificador: (GET) /webresources/logs/id
- Obter todos os registos: (GET) /webresources/logs/
- Obter todos os registos por data: (GET) /webresources/logs/date/from/to
- Obter todos os registos por georeferência de instância: (GET) /webresources/logs/georef/georef
- Obter todos os registos por georeferência de utilizador: (GET) /webresources/logs/usergeoref/georef
- Obter registos de instâncias mais visitadas: (GET) /webresources/logs/instances
- Obter registos de classes mais visitadas: (GET) /webresources/logs/classes
- Obter contagem de registos: (GET) /webresources/logs/count

Apesar de, para o desenvolvimento da aplicação OObian Android, se ter optado pela utilização da serialização Protobuf, todos os pedidos de criação e edição de registos no módulo OObian Analytics são efectuados através de objectos JSON. Esta escolha teve em consideração o facto de estarmos perante um desenvolvimento que é menos exigente em termos de consumo de recursos do sistema anfitrião, bem como a maior facilidade de customização dos serviços a disponibilizar. De salientar que a geração de estruturas de dados a usar na serialização Protobuf é mais complexa que para JSON, o que não se justificaria para um cenário que se apresenta tão embrionário e em que se prevêem mais alterações.

### **Integração OObian**

O módulo OObian Analytics permite ser alimentado, tanto via web-service, como por integração no servidor e utilização directa através da API do módulo. Esta API possui métodos de inserção, consulta e remoção de dados relativos a navegações na estrutura de conhecimento OObian.

Esta integração é possibilitada através da incorporação da biblioteca gerada pela compilação deste módulo e da chamada dos interfaces disponibilizados.

## Persistência de Dados

A persistência de dados é garantida através de uma base de dados MySQL, sendo a comunicação entre esta e a aplicação OObian cliente ou servidor garantida através das duas API's descritas anteriormente.

analytics_logs			
Name	Type	Nullable	
idLog	Int	No	
timestamp	DateTime	Yes	
device	String(45)	Yes	
operation	String(45)	Yes	
keywords	String(1000)	Yes	
idInstance	Int	Yes	
idClass	Int	Yes	
georef	String(45)	Yes	
userGeoRef	String(45)	Yes	

Figura 5.7: Diagrama de base de dados do módulo OObian Analytics

Numa primeira instância, e visto que o desenvolvimento deste módulo de análise de dados pretende apenas abrir a porta a novos desenvolvimentos para projectos futuros, não se entrou na exploração de mecanismos mais complexos de persistência, uma vez que a complexidade em termos estruturais para este módulo poderiam pôr em causa os objectivos principais deste estágio: o desenvolvimento da aplicação OObian Android.

### 5.2.2 Decisões Tecnológicas

Em relação a decisões tecnológicas tomadas no desenvolvimento deste módulo, pode-se contemplar a utilização de um padrão de design de software: o Façade. Com a adopção deste padrão, foi criado um interface responsável pela interligação entre a API REST, a API Java do OObian Analytics e a JPA. Esta interligação é também conseguida através de um EJB definido pela tabela da figura 5.7.

De forma a proporcionar a visualização do funcionamento deste módulo, desenvolveu-se também uma componente *web* que foi integrada na própria aplicação OObian Android. Esta componente *web* recorre a JSP's e à inclusão de bibliotecas como o *jQueryUI*<sup>8</sup>, *Javascript* e *D3.js*<sup>9</sup>. O *D3.js* é uma biblioteca desenvolvida em Javascript que possibilita a representação de dados serializados de forma facilmente consultavel pelo utilizador e recorre a HTML, CSS e SVG para renderizar a informação.

<sup>8</sup><http://jqueryui.com/>

<sup>9</sup><http://d3js.org/>

# Capítulo 6

## Desenho de interface

### 6.1 OObian Android

O desenho da interface da aplicação OObian Android seguiu os planos definidos no primeiro semestre deste estágio. Assim, seguem alguns pormenores de User Experience que acompanham a execução da aplicação OObian Android.



Figura 6.1: Ecrã inicial da aplicação OObian Android

O ecrã inicial, como em toda a aplicação, apresenta no topo a ActionBar. Esta ActionBar possibilita a navegação para o modo de navegação na estrutura de conhecimento a partir da raiz, para o modo de navegação em mapa e para o módulo de Analytics. É também apresentada uma caixa de texto para pesquisa corrida, em que, à medida que o utilizador insere texto, são apresentadas sugestões de instâncias. O restante ecrã apresenta separadores



para instâncias georeferenciadas próximas da posição actual do utilizador, histórico de navegações e, em modo de suporte para estas funcionalidades, a listagem de favoritos do utilizador e uma ligação para o ecrã de perfil de utilizador.



Figura 6.2: Navegação a partir da raíz da estrutura de conhecimento

A navegação a partir da raíz pretende oferecer ao utilizador uma exploração da estrutura de conhecimento e uma navegação no grafo do OObian de uma forma livre e sem um objectivo concreto. Esta navegação pode ser feita a partir do ecrã inicial da aplicação OObian Android. Nesta navegação, é possível partir para classes ou instâncias dentro de classes, sendo oferecido um menu de contexto para as instâncias. Ao utilizador é também possibilitada a transição para a vista de detalhe de instância ou classe, ou permanecer em vista de navegação corrida.



Figura 6.3: Vista de navegação em instância

Na vista de detalhe é possível consultar as instâncias relacionadas (para o caso das instâncias), as classes relacionadas (para o caso das classes), os detalhes da instância e demais conteúdo associado (georeferência, conteúdo multimédia, conteúdo web, etc.).

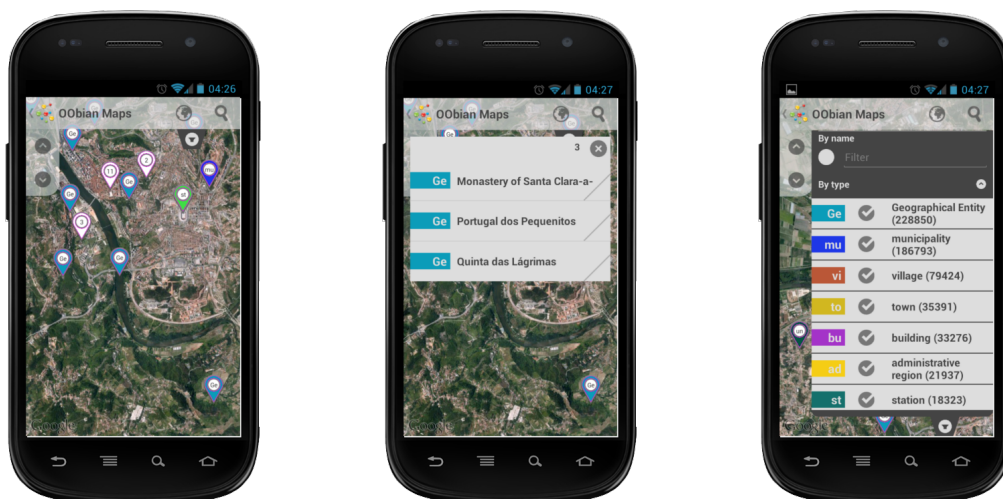


Figura 6.4: Módulo OOBian Maps

O módulo Maps do OOBian Android representa geograficamente a informação contida na estrutura de conhecimento. Todas as instâncias são agrupadas em clusters, salvo a resolução de instâncias individuais. Cada cluster ou instância é representado por um marcador no mapa. Os marcadores

numerados, relativos a clusters, activam a listagem de todas as instâncias incluídas. Os marcadores coloridos, que representam determinada instância, incluem também a classe a que pertence (representada pelos dois primeiros caracteres do nome). Este módulo é acompanhado também de uma *tab* de filtragem em que o utilizador poderá introduzir *keywords* ou seleccionar a classe de instâncias que pretende visualizar.

## 6.2 OObian Analytics

Para o OObian Analytics, e visto que se trata de um desenvolvimento paralelo ao do OObian Android, ficou definido que a interface fosse o mais simples possível, de modo a apenas ser possível verificar o tratamento da informação por parte deste módulo. Ainda assim, e por uma questão de simplificação, foi desenvolvida, neste estágio, uma web capaz de ser reutilizada, tanto por consumo directo de um browser, como através da aplicação OObian Android, via webview.

# Capítulo 7

## Testes

Os testes realizados sobre a aplicação OObian Android e sobre o módulo Analytics foram definidos através da transposição dos requisitos levantados para as aplicações e descritos nos capítulos anteriores, intersectando estes com métricas básicas de utilização de uma aplicação Android.

Segue então a lista dos testes mais relevantes realizados sobre a aplicação OObian Android:

- Instalação da aplicação - o utilizador deverá conseguir instalar a aplicação
- Iniciar aplicação - o utilizador deverá conseguir iniciar a aplicação através do menú de aplicações do dispositivo móvel
- Excepções transparentes para o utilizador - qualquer excepção que ocorra deverá ser comunicada ao utilizador só e apenas quando a acção que pretende executar se encontra limitada
- Autenticar - o utilizador deverá poder introduzir as suas credenciais de utilizador, caso o contexto em que a aplicação OObian Android esteja incluído necessite
- Iniciar aplicação com credenciais guardadas - deverá ser possível ao utilizador voltar a utilizar a aplicação OObian Android sem necessitar voltar a introduzir as suas credenciais
- Execução de operações básicas da aplicação requerem um número máximo de 3 toques - deverá ser assegurada ao utilizador a realização de tarefas comuns sem nunca exceder os três toques no ecrã do dispositivo móvel

- Pesquisa desambiguada - deverão ser apresentados ao utilizador resultados de pesquisa desambiguada à medida que introduz texto na caixa de texto de pesquisa presente no ecrã inicial da aplicação OObian Android
- Pesquisa - o utilizador deverá poder aceder à pesquisa na aplicação a partir de qualquer ponto da aplicação OObian Android
- Acções com tempos de resposta altos - o utilizador deverá ser alertado caso a acção que despoletou demorar 15 segundos, o que leva ao cancelamento da tarefa
- Listagem de Favoritos - caso o contexto em que a aplicação OObian Android esteja a ser usado, o utilizador deverá ter acesso à listagem de favoritos a partir de qualquer ponto da aplicação
- Listagem de Histórico - o utilizar deverá ter acesso à listagem de instâncias visitadas recentemente, a partir do ecrã inicial da aplicação OObian Android
- Por perto - deverá ser apresentada, no ecrã inicial da aplicação OObian Android, uma listagem de instâncias georeferenciadas próximas do utilizador
- Iniciar navegação - o utilizador deverá poder iniciar a navegação a partir da raiz da estrutura de conhecimento, no ecrã inicial da aplicação
- Iniciar mapa - o utilizador deverá poder iniciar a navegação no módulo Maps a partir do ecrã inicial da aplicação OObian Android
- Visualizar perfil de utilizador - caso o contexto o permita, o utilizador deverá poder navegar para a vista de Perfil (configurações), a partir do ecrã inicial da aplicação
- Terminar sessão - caso o contexto o permita, o utilizador deverá poder efectuar o textitlogout da aplicação
- Edição de dados do utilizador - caso o contexto o permita, deverá ser facultada ao utilizador uma vista de edição de detalhes do utilizador
- Adicionar fontes de informação - caso o contexto o permita, o utilizador deverá poder adicionar as fontes de informação utilizadas na vista de detalhe da navegação de instâncias e classes
- Remover fontes de informação

- Editar fontes de informação
- Iniciar vista de granularidade mínima - o utilizador deverá poder iniciar a vista de granularidade mínima de classe ou instância
- Navegar a partir de relações de instância - o utilizador deverá poder-navegar para instâncias relacionadas com a qual se encontra a visitar
- Paginação em relações de instância - deverá ser facultado um mecanismo de paginação para facilitar a apresentação de conteúdos de grandes dimensões
- Visualizar descrição - deverá ser apresentada uma descrição da instância ou classe visitada
- Visualizar detalhes de instância - deverá ser apresentada uma listagem de detalhes da instância visitada
- Visualizar elementos multimédia - deverá ser apresentada uma galeria de elementos multimédia associados à instância visitada
- Visualização de dados do Analytics - deverá ser facultada uma vista de apresentação de dados capturados pelo módulo OObian Analytics
- Visualizar mapa de instância georeferenciada - deverá ser apresentada uma representação da localização de uma instância georeferenciada
- Navegação por geo-referência - a representação da localização da instância deverá permitir a navegação para o módulo Maps
- Visualização de lista de documentos - caso o contexto o permita, deverão ser listados os documentos relacionados com a instância visitada
- Visualização de documento - deverá ser possível ao utilizador visualizar um documento relacionado com uma instância
- Visualização de fonte de internet - o utilizador deverá poder navegar para a página da fonte de informação apresentada na instância
- Controlo de menu de contexto de instância navegada - a instância deverá possuir um menu de contexto para operações sobre essa mesma instância
- Navegação no mapa - deverá ser possível navegar no mapa, expandir clusters e navegar para instâncias

Para além dos testes realizados no âmbito do desenvolvimento da aplicação OObian Android para apuramento de eventuais bugs que não tenham sido detectados anteriormente, foi desenvolvido um questionário para o apuramento da opinião de utilizadores da aplicação. Este questionário seguiu as directivas[7] delineadas por Raluca Budiú e Jakob Nielsen e pretende, não só medir o grau de semelhança com os restantes clientes OObian, como também a integração de padrões de usabilidade nativas do sistema operativo Android, sem esquecer a facilidade de utilização para atingir determinado objectivo.

**OObian Android**

**Utilização corrida**  
Para este grupo do questionário, pretende-se que o utilizador explore a aplicação OObian Android sem qualquer restringimento.

**Efectuar Login**  
Avalie o grau de acessibilidade à aplicação.

1 2 3 4 5

Pouco claro      Muito claro

**Navegar na aplicação e efectuar uma pesquisa**  
Avalie o grau de facilidade em aceder a esta funcionalidade.

1 2 3 4 5

Fácil      Difícil

**Efectuar uma navegação a partir da raiz de conhecimento**  
Avalie o grau de facilidade em aceder a esta funcionalidade.

1 2 3 4 5

Fácil      Difícil

**Efectuar uma navegação para uma instância**  
Avalie o grau de facilidade em aceder a esta funcionalidade.

1 2 3 4 5

Fácil      Difícil

Figura 7.1: Questionário sobre navegação livre

O levantamento das questões para este formulário tiveram em conta o facto de ser importante ter três perspectivas em relação à utilização da apli-

## OObian Android

### Utilização contextualizada

Para este grupo do questionário, pretende-se que o utilizador siga as instruções dadas com a finalidade de atingir determinado objectivo.

#### A partir do ecrã inicial, navegue para a instância "Universidade de Coimbra"

Avalie o grau de facilidade em proceder a esta operação.

1 2 3 4 5

Fácil      Dificil

#### Navegue para a representação geográfica de "Universidade de Coimbra" no OObian Maps da aplicação.

Avalie o grau de facilidade em proceder a esta operação.

1 2 3 4 5

Fácil      Dificil

#### Na classe "Universidades", filtre as relações por "Universidade de Coimbra"

Avalie o grau de facilidade em proceder a esta operação.

1 2 3 4 5

Fácil      Dificil

Figura 7.2: Questionário sobre navegação contextualizada



**OObian Android**

**Usabilidade Android**  
Neste grupo, pretende-se avaliar a utilização de padrões de usabilidade comuns em aplicações Android.

**Qual o grau de clareza das funcionalidades presentes na ActionBar?**

1 2 3 4 5

Pouco claro      Muito claro

**Qual o grau de clareza dos menus de contexto de instâncias visitadas?**

1 2 3 4 5

Pouco claro      Muito claro

**Qual o grau de clareza dos menus de contexto de instâncias relacionadas?**

1 2 3 4 5

Pouco claro      Muito claro

**Qual o grau de clareza das vistas de filtragem?**

1 2 3 4 5

Pouco claro      Muito claro

Never submit passwords through Google Forms.

Figura 7.3: Questionário sobre padrões de usabilidade

cação: uma perspectiva em que o utilizador navega na aplicação sem qualquer tipo de condicionante ou caminho proposto para determinado objectivo; uma perspectiva em que é medido o grau de facilidade com que se cumpre determinado objectivo; uma perspectiva em que se mede a inclusão da aplicação no ecossistema Android.

A disponibilização pública deste formulário apenas faria sentido caso fosse estudado um número considerável de indivíduos (mais de 15). Visto que esta meta não seria atingida em tempo útil para a sua interpretação e inclusão de resultados neste relatório, decidi que esta tarefa fosse incluída como trabalho futuro a realizar, com vista à evolução da aplicação OObian Android.

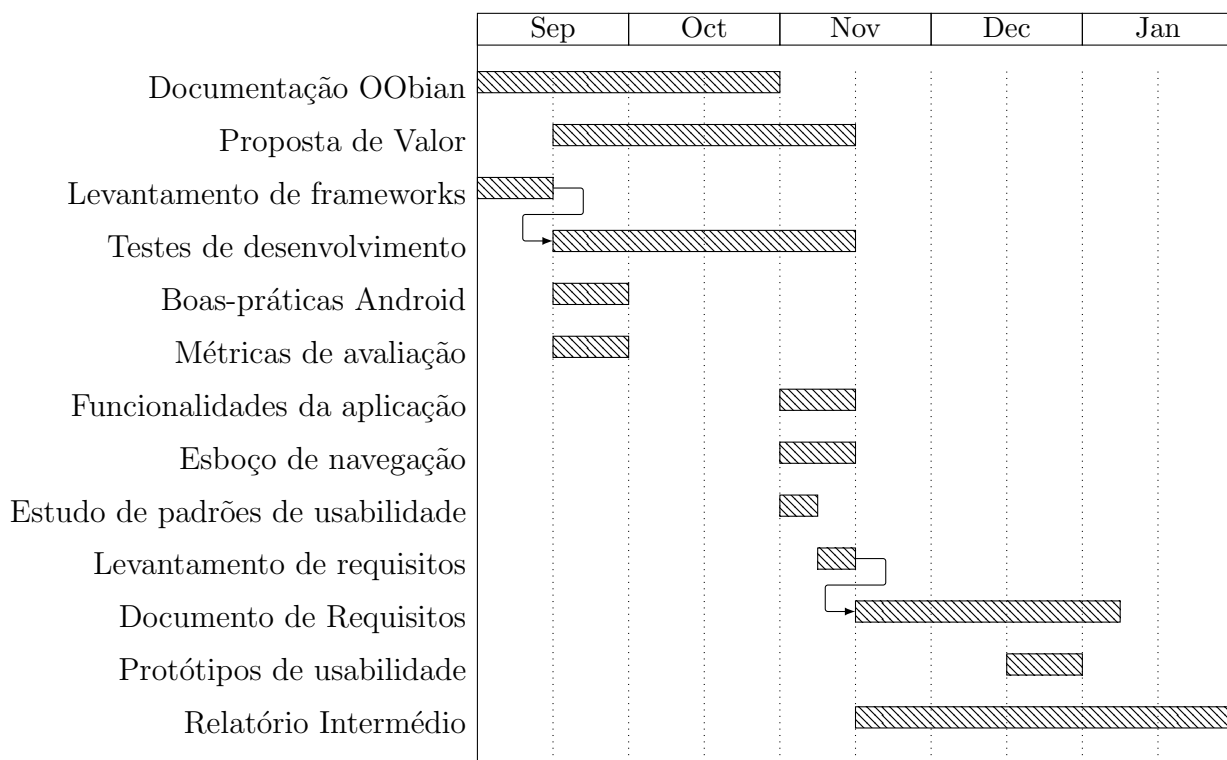


# Capítulo 8

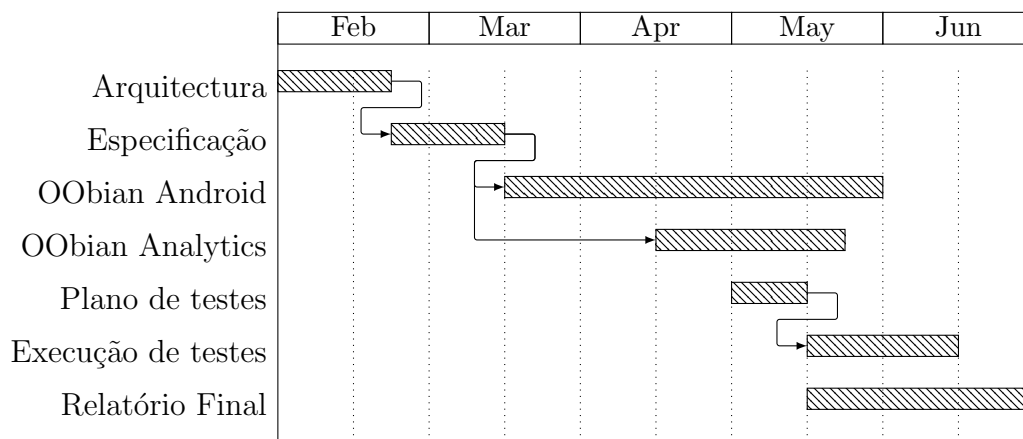
## Planeamento

### 8.1 Plano de Estágio

O diagrama de Gantt apresentado seguidamente pretende reflectir o trabalho efectuado durante o primeiro semestre do presente estágio. De salientar que ao longo do estudo do Estado de Arte o âmbito deste projecto, foram produzidos vários documentos, tanto de cariz interno (para a Maisis), como para a Universidade de Coimbra.



Durante a derradeira etapa deste estágio, o plano de trabalhos cumprido pode-se representar através do seguinte diagrama de Gantt. Constata-se, assim, que o desenvolvimento da aplicação OObian Android e da componente Analytics ocupou grande parte da segunda parte desta Unidade Curricular.



## 8.2 Análise de Riscos

Durante a fase de planeamento da aplicação OObian Android, do levantamento de requisitos funcionais e não funcionais e, sobretudo, da definição da sua arquitectura, foi possível deparar-me com diversos riscos associados ao seu desenvolvimento.

Um dos principais riscos identificados no início do estágio foi o facto de o sistema operativo Android estar associado a uma grande fragmentação, não só em termos de software, como em termos de hardware. A garantia de que uma aplicação é totalmente compatível com diversos dispositivos móveis com diversas versões do sistema operativo leva a que, por vezes, e como aconteceu neste estágio, se tenha que recorrer a bibliotecas desenvolvidas por terceiros. No âmbito deste estágio foi-me possível cobrir quase na totalidade as inconsistências levantadas por esta fragmentação do sistema operativo.

Outro tipo de fragmentação associado ao ecossistema Android é a própria fragmentação do hardware: o facto de o Android ser um sistema operativo livre leva a que o leque de dispositivos que o suporte seja vasto. De igual forma, um vasto leque de dispositivos móveis significa um vasto leque de configurações de sistemas: dispositivos com e sem GPS, com e sem acesso 3G, com e sem acesso WiFi, com diferentes densidades de ecrã. Ao longo do desenvolvimento da aplicação OObian Android, foi sempre tido em conta as possíveis limitações do dispositivo anfitrião. Como exemplo, a geolocalização do utilizador poderá ser conseguida através de localização via WiFi, ao invés

da utilização do módulo GPS.

Directamente relacionado com uma aplicação que está dependente do acesso *web* está o risco de alimentação do OObian Android como expositor de informação. Este risco encontra-se salvaguardado pelo próprio sistema operativo, através de mecanismos de gestão de rede. Da parte desenvolvida neste estágio, apenas se garante que a aplicação tenha forma de aceder à rede externa caso esta se encontre disponível.

### 8.3 Trabalho Futuro

Como trabalho futuro, e para o sucesso do desenvolvimento contínuo da aplicação OObian Android, será necessário dar seguimento às suas iterações do ciclo de vida. Em primeiro lugar será necessário dar continuidade à execução do questionário de usabilidade da aplicação. Em segundo lugar, terão de ser analisados os resultados levantados e adaptar a aplicação ao *feedback* obtido. Assim que a aplicação atingir um grau consistente de maturidade, deverá ser lançada no Google Play. A este lançamento estará associado um constante contacto com a plataforma e uma constante atenção aos seus utilizadores, visto que será a porta de entrada para o seu *feedback* imediato.

Em relação ao módulo OObian Analytics, o passo seguinte passará pelo desenvolvimento de mecanismos de persistência mais eficazes e mais abrangentes, para que se possa cobrir toda a informação gerada pela transmissão de dados feita entre o utilizador e o servidor. Melhorados também deverão ser as formas como a informação é extraída desta Data Warehouse.

Neste âmbito será também interessante desenvolver métodos de inclusão de resultados do OObian Analytics na navegação da estrutura de conhecimento do OObian, seja através da introdução de métricas de avaliação de resultados de pesquisa, ranking de instâncias ou classes, apresentação de informação mais relevante para determinado utilizador, etc.



# Bibliografia

- [1] Brian Clifton. Web traffic data sources & vendor comparison. <http://www.advanced-web-metrics.com/docs/web-data-sources.pdf>, Maio 2008.
- [2] Peter Eeles. Capturing architectural requirements. <http://www.ibm.com/developerworks/rational/library/4706.html>, Novembro 2005.
- [3] Platform versions. <http://developer.android.com/about/dashboards/index.html>, Novembro 2005.
- [4] Travis Green & Max Lin & Robert Kaplow & Johannes Kristinsson & Ryan McGee. Smart app design. [http://www.youtube.com/watch?v=FJDP\\_0Mrb-w](http://www.youtube.com/watch?v=FJDP_0Mrb-w), Maio 2011.
- [5] User interface. <http://developer.android.com/guide/topics/ui/index.html>.
- [6] Eishay Smith. Benchmark comparing serialization libraries on the jvm. <https://github.com/eishay/jvm-serializers/wiki>, Abril 2012.
- [7] Raluca Budiu e Jakob Nielsen. Mobile website and application usability. <http://www.nngroup.com/reports/mobile-website-and-application-usability/>.
- [8] Xavier Ducrohet & Tor Norbye. Android development tools. <http://www.youtube.com/watch?v=0q05KqjXTvs>, Maio 2011.
- [9] Reto Meier. Android protips: Advanced topics for expert android app developers. [http://www.youtube.com/watch?v=twmuBbC\\_oB8](http://www.youtube.com/watch?v=twmuBbC_oB8), Maio 2011.
- [10] Reto Meier & Michael Mahemoff. Html5 versus android: Apps or web for mobile development? [http://www.youtube.com/watch?v=4f2Zky\\_YyyQ](http://www.youtube.com/watch?v=4f2Zky_YyyQ), Maio 2011.



- [11] Michael Mahemoff & Paul Kinlan. Mobile web development: From zero to hero. <http://www.youtube.com/watch?v=vV85dNeGRhY>, Maio 2011.
- [12] Romain Guy & Chet Haase. Accelerated android rendering. [http://www.youtube.com/watch?v=FJDP\\_0Mrb-w](http://www.youtube.com/watch?v=FJDP_0Mrb-w), Maio 2011.
- [13] Christy Pettey & Rob van der Meulen. Market share: Mobile phones by region and country, 3q12. <http://www.gartner.com/newsroom/id/2237315>, Novembro 2012.
- [14] Roberta Cozza. Forecast: Mobile communications devices by open operating system, worldwide, 2008-2015. <http://www.gartner.com/resId=1619615>, Abril 2011.
- [15] Christy Pettey & Rob van der Meulen. Forecast: Media tablets by operating system, worldwide, 2010-2016, 1q12 update. <http://www.gartner.com/newsroom/id/1980115>, Abril 2012.
- [16] Jonas Lind. Platform x: How cross-platform tools can end the os wars. <http://www.visionmobile.com/blog/2011/06/platform-x-how-cross-platform-tools-can-end-the-os-wars/>, Junho 2011.