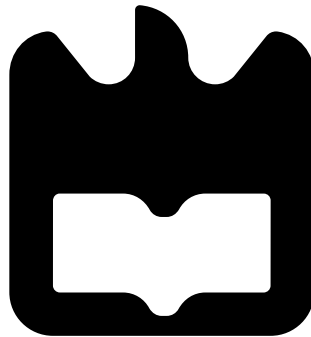




**Valter Pedro  
Lopes Vicente**

## **Gestão de Medicamentos em Dispositivos Móveis**







**Valter Pedro  
Lopes Vicente**

## **Gestão de Medicamentos em Dispositivos Móveis**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Prof. Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Prof. Doutor António Joaquim da Silva Teixeira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro



**o júri**

presidente

**Prof. Doutor António Manuel Melo de Sousa Pereira**

Professor Catedrático da Universidade de Aveiro

vogais

**Prof. Doutor Rui Pedro Sanches de Castro Lopes**

Professor Coordenador do Departamento de Informática e Comunicações da ESTG do Instituto Politécnico de Bragança

**Prof. Doutor José Luís Guimarães Oliveira**

Professor Associado da Universidade de Aveiro (orientador)

**Prof. Doutor António Joaquim da Silva Teixeira**

Professor Auxiliar da Universidade de Aveiro (co-orientador)



## **agradecimentos**

Agradeço aos meus pais por sempre terem estado ao meu lado todos estes anos, providenciando a minha educação e apoiando-me nas minhas decisões. Agradeço ao meu irmão por ser o melhor irmão que alguém pode ter. Agradeço à minha restante família pelos bons momentos e pelo fantástico sentimento de 'voltar a casa'.

A todos os meus amigos, uma palavra de apreço por me deixarem ser parte das suas vidas e por estarem ao meu lado ao longo destes anos, em todos os bons e todos os maus momentos, nas brincadeiras e nas ocasiões sérias. Os novos, os velhos, dos mais presentes aos mais ausentes, todos foram importantes para ser quem sou e estarão sempre comigo.





## Palavras-chave

Computação Móvel, Medicação, Cuidados de Saúde

## Resumo

Ao longo da vida, todos nós acabamos por ter a necessidade de tomar algum tipo de medicação. Seja de forma esporádica ou mais continuamente, tomar medicação é um processo que envolve alguma preocupação da nossa parte, tanto devido a poder ter horários específicos para a sua toma, como pelo conteúdo dos medicamentos em si e, principalmente, por esta ser tão relevante para a nossa saúde.

O advento dos dispositivos móveis nos últimos anos, especialmente dos *smartphones*, permite à população ter ao seu alcance um meio que facilita algumas tarefas do dia-a-dia e de que ajuda a debelar algumas preocupações. Contudo, dada a propagação destes, existem diversos dispositivos, com diferentes sistemas operativos, pelo que desenvolver para todos eles se torna um processo mais demorado.

Tirando partido dos mesmos *smartphones*, propõe-se o desenvolvimento de uma aplicação móvel que permita a toda a gente diminuir as preocupações que a toma de medicação cria em cada um de nós, independentemente do *smartphone* que possuam. Esta aplicação móvel (Mobile Medicine) é uma solução que se propõe a disponibilizar ao utilizador a possibilidade de obter informações sobre os medicamentos existentes no mercado português (aprovados pelo INFARMED), adicionar medicamentos à sua lista de prescrições e definir alertas para as tomas dos mesmos.



**Keywords**

Mobile Computing, Medication, Healthcare

**Abstract**

Throughout life, we all end up having the need to take some kind of medication. Whether it is something more sporadically or a continuous thing, taking medication is a process that involves some concern on our part, because it may need to be take at some specific times, we must be aware of the content of the meds themselves or, especially, since this is so relevant to our health.

The advent of mobile devices in recent years, especially smartphones, allows people to have at hand something that facilitates some day-to-day tasks and that helps quell some concerns. However, given the propagation of these, there are several devices with different operating systems, which makes the development for all of them a more time consuming process.

Taking advantage of these smartphones, it's proposed the development of a mobile application which allows everyone to decrease the concerns that taking medication creates in each one of us, regardless of the smartphone they own. This mobile application is a solution that aims to provide the user the possibility to get information on existing medicines in the Portuguese market (approved by INFARMED), add medications to his list of prescriptions and set alerts for those medicine taking.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Estrutura . . . . .	2
<b>2 Estado da Arte</b>	<b>5</b>
2.1 História e Evolução dos Dispositivos Móveis . . . . .	5
2.2 <i>Frameworks</i> de Desenvolvimento . . . . .	8
2.2.1 <i>Frameworks</i> Nativos . . . . .	8
2.2.2 <i>Frameworks</i> Multiplataforma . . . . .	9
2.2.2.1 PhoneGap . . . . .	9
2.2.2.2 Appcelerator Titanium . . . . .	12
2.3 Aplicações Relacionadas . . . . .	14
2.3.1 iPharmacy . . . . .	14
2.3.2 CR Medication Tracker . . . . .	16
2.3.3 Micromedex Drug Information . . . . .	17
2.3.4 Med Minder . . . . .	18
2.3.5 Pillbox . . . . .	19
<b>3 Análise de Requisitos</b>	<b>21</b>
3.1 Requisitos Gerais . . . . .	21
3.2 Requisitos de Funcionalidades . . . . .	22
3.2.1 Casos de Uso . . . . .	24
3.2.1.1 Definir Perfil . . . . .	24
3.2.1.2 Pesquisar Medicamento . . . . .	24
3.2.1.3 Verificar Prescrições . . . . .	25
3.2.1.4 Visualizar Contacto de Emergência . . . . .	27
3.2.1.5 Consultar Conselheiro . . . . .	28
3.2.1.6 Encontrar Farmácias . . . . .	28

<b>4</b>	<b>Desenvolvimento</b>	<b>33</b>
4.1	Arquitetura do Sistema . . . . .	33
4.1.1	<i>Frameworks</i> de Desenvolvimento . . . . .	33
4.1.2	Base de Dados de Medicamentos . . . . .	35
4.2	A Aplicação . . . . .	36
4.2.1	Usabilidade da Aplicação . . . . .	45
<b>5</b>	<b>Conclusão</b>	<b>49</b>
5.1	Sumário do Trabalho Realizado . . . . .	49
5.2	Trabalho Futuro . . . . .	50
	<b>Referências</b>	<b>51</b>

# Lista de Figuras

2.1	Os impulsionadores da computação móvel . . . . .	6
2.2	<i>Smartphones</i> com maior volume de vendas . . . . .	7
2.3	iPharmacy - Exemplos . . . . .	15
2.4	CR Medication Tracker - Exemplos . . . . .	16
2.5	Micromedex Drug Information - Exemplos . . . . .	17
2.6	Med Minder - Exemplos . . . . .	18
2.7	Pillbox - Exemplos . . . . .	19
3.1	Modelo de casos de uso da aplicação . . . . .	23
3.2	Diagrama de Sequência - Definir Perfil . . . . .	25
3.3	Diagrama de Sequência - Pesquisar Medicamento . . . . .	26
3.4	Diagrama de Sequência - Verificar Prescrições . . . . .	27
3.5	Diagrama de Sequência - Visualizar Contacto de Emergência . . . . .	29
3.6	Diagrama de Sequência - Consultar Conselheiro . . . . .	30
3.7	Diagrama de Sequência - Encontrar Farmácias . . . . .	31
4.1	Visão geral do sistema. . . . .	36
4.2	Mobile Medicine - Ecrã Inicial . . . . .	36
4.3	Mobile Medicine - Definir Perfil . . . . .	37
4.4	Mobile Medicine - Contacto de Emergência . . . . .	38
4.5	Mobile Medicine - Pesquisa . . . . .	40
4.6	Mobile Medicine - Informações do Medicamento . . . . .	41
4.7	Mobile Medicine - Adicionar às Prescrições e Definição de Alarme . . . . .	42
4.8	Mobile Medicine - Prescrições . . . . .	44
4.9	Mobile Medicine - Histórico . . . . .	44
4.10	Heurística - Visibilidade do Estado do Sistema . . . . .	45
4.11	Heurística - Controlo e liberdade do utilizador . . . . .	46
4.12	Heurística - Ajuda os utilizadores a reconhecer, diagnosticar e recuperar de erros . . . . .	46





# Lista de Tabelas

2.1	Vendas de Dispositivos Móveis a Utilizadores Finais, por Sistema Operativo, no 2Q12 (em Milhares de Unidades) [4] . . . . .	7
2.2	Comparação entre as características mais relevantes dos <i>frameworks</i> PhoneGap e Appcelerator Titanium [14] . . . . .	14



# Capítulo 1

## Introdução

### 1.1 Motivação

Nos últimos anos, o mercado de *smartphones* tem sido um dos mercados de dispositivos tecnológicos com mais rápido crescimento a nível mundial, havendo a tendência de estes dispositivos ficarem cada vez mais baratos a cada dia que passa e, assim, de se tornarem no tipo de telemóveis mais utilizado por todos. As pessoas que possuem um *smartphone* não só tiram partido de todas as opções que estão habituadas num telemóvel dito normal, como a realização de chamadas e envio/receção de mensagens, mas tendem para o ponto em que sentem a necessidade de estarem ainda mais ligadas a toda a gente, com informação facilmente acessível de forma instantânea, fazendo com que sintam que precisam de ter as suas informações e as dos seus amigos, os seus horários e as suas rotinas guardadas e acessíveis digitalmente. As empresas de desenvolvimento de software estão cientes deste facto, estando cada vez vez mais a apostar na área de dispositivos móveis, sendo que do lado dos operadores de telecomunicações, não só estes estão também cientes disso, como também sabem das possíveis vantagens e desvantagens que podem existir neste caso, uma vez que esta situação faz com que aumente a necessidade de existirem mais serviços móveis de Internet, pelo que estes operadores se devem concentrar em dar aos seus clientes mais serviços de dados e conteúdos com valor para os mesmos, em vez de se focarem só nos mais tradicionais serviços de voz e mensagens.

Quando somos crianças, tomamos medicamentos para as febres ou para a varicela e quando somos velhos, tomamos medicamentos para o reumático ou para a osteoporose. Existem muitas doenças que nos podem também afetar a qualquer altura ou acompanhar ao longo da nossa vida, como infeções, depressão, diabetes ou cancro, podendo estas ser causadas por fatores ambientais ou hereditários. Assim, quer sejamos uma pessoa jovem, de meia-idade ou idosa, em qualquer altura da nossa vida, temos de tomar algum tipo de medicação, seja para nos fazer melhorar, seja para prevenir que pioremos.

Tendo em conta estes fatores, torna-se relevante que exista uma aplicação móvel para *smartphones* que permita tirar partido das potencialidades oferecidas por este tipo de dispositivos e que, ao mesmo tempo, permita atenuar a necessidade de procura de informações e as preocupações adjacentes à toma de medicamentos, aplicação essa que se deve direccionar a todos os tipos de pessoas.

Apesar de já existirem no mercado algumas aplicações do género, é importante notar que cada país difere nas marcas que distribui e nos tipos de medicamentos que tem regulamentados (estando essa tarefa, em Portugal, a cargo do *INFARMED - Autoridade Nacional*

do Medicamento e Produtos de Saúde, I. P. [1]), não existindo nenhuma com incidência no mercado nacional.

## 1.2 Objetivos

O principal objetivo desta dissertação é desenvolver uma aplicação para *smartphones*, tendo em vista as características do mercado português, que permita a gestão de tomas e a consulta de medicamentos, com perfil e histórico de utilizador.

O resultado final deverá consistir numa aplicação móvel que permita a um utilizador, que possui um determinado perfil definido pelo mesmo, indicar quais os medicamentos que está a tomar, ser alertado de quando os deve tomar e/ou pesquisar informações úteis sobre os mesmos ou sobre qualquer outro medicamento. Neste último caso, é relevante referir que terá de ser estudado o melhor meio de se obterem as informações relativas a cada medicamento, informações essas que serão cedidas pelo INFARMED e tratadas de acordo com o objetivo pretendido para a aplicação, e assim, neste caso em específico, o utilizador terá de estar conectado à Internet para obter os dados de cada medicamento.

Como se pretende que esta aplicação seja direcionada a todas as pessoas que possuem um *smartphone* e dado que a quantidade de sistemas operativos para dispositivos móveis que existem é algo variada, esta aplicação será multiplataforma, para assim conseguir cobrir um maior espectro de possibilidades. Este facto irá envolver a necessidade de estudar os *frameworks* de programação multiplataforma para dispositivos móveis, tendo em atenção as suas vantagens mas também as suas limitações em relação aos *frameworks* nativos de cada plataforma.

## 1.3 Estrutura

Este documento está dividido em quatro capítulos principais - Estado da Arte, Análise de Requisitos, Desenvolvimento e Conclusão, podendo cada um deles ser divididos em secções para uma melhor definição dos mesmos, tal como se apresenta de seguida:

- **Capítulo 1 *Introdução***

Este capítulo descreve o contexto deste projeto, porque é que o desenvolvimento do mesmo é benéfico e quais os seus objetivos.

- **Capítulo 2: *Estado da Arte***

Este capítulo apresenta a evolução dos dispositivos móveis e como os propósitos dos mesmo foram sendo alterados ao longo dos anos. São apresentados os *frameworks* de desenvolvimento para dispositivos móveis existentes, sejam estas específicas de cada sistema operativo ou multiplataforma, e são também apresentadas aplicações semelhantes existentes no mercado.

- **Capítulo 3: *Análise de Requisitos***

Este capítulo descreve quais os requisitos gerais que se pretende para a aplicação, bem como os seus requisitos de funcionalidades entendidas como necessárias para se ter a mesma de acordo com o esperado.

- **Capítulo 4: *Desenvolvimento***

Este capítulo apresenta as escolhas efetuadas a nível de tecnologias para desenvolvimento da aplicação, a sua arquitetura geral e os detalhes de implementação da mesma de modo a atingir a aplicação final.

- **Capítulo 5: *Conclusão***

Este capítulo descreve as conclusões obtidas no término do projeto, sumarização do trabalho realizado e possíveis implementações e/ou melhoramentos futuros.



## Capítulo 2

# Estado da Arte

### 2.1 História e Evolução dos Dispositivos Móveis

Atualmente, os dispositivos móveis podem ser separados em dois grupos principais: os *smartphones* e os *tablets*. Para chegar a estes, um longo caminho foi percorrido a nível de computação móvel, pelo que neste sub-capítulo serão abordados alguns dos dispositivos que fizeram parte desta evolução.

Apesar de existir alguma discussão sobre quais foram os primeiros dispositivos móveis e que assim podem ser considerados como os impulsionadores da computação móvel, os apresentados de seguida são, hoje em dia, considerados os mais relevantes [2].

Lançado em 1983, o CASIO PF-3000 (figura 2.1a) foi o primeiro diário digital a incluir com uma calculadora e um gestor de endereços, números de telefone e nomes.

No ano seguinte, em 1984, foi lançado o Psion Organizer (figura 2.1b) com o anúncio de ser o “primeiro computador de bolso do mundo”. De entre as suas características destacavam-se um ecrã LCD que suportava 16 caracteres e a possibilidade de se programar numa linguagem tipo BASIC.

O primeiro PDA (Personal Digital Assistance), termo ainda atual, apenas surgiu em 1992, quando a Apple apresentou o Apple Newton MessagePad (figura 2.1c). Apesar da diferença de 8 anos relativa aos dispositivos anteriores, este trouxe várias inovações que foram consideradas marcantes na história da computação móvel: reconhecimento de escrita-à-mão, reconhecimento da orientação do ecrã e, a característica que mais se destacou, ecrã com reconhecimento de toque (*touch screen*), sendo que esta interação era efetuada recorrendo uma caneta *stylus*.

Devido a vários lançamentos ocorridos em paralelo, o ano de 1996 foi um dos mais importantes na história dos dispositivos móveis. Empresas como a Nokia, a Palm e a Microsoft não quiseram ficar para trás, tendo lançado os seus próprios dispositivos: a Nokia lançou o 9000 Communicator, o primeiro telefone móvel com funcionalidades de PDA; a Palm lançou o Palm Pilot, que trazia a tecnologia Graffiti, uma tecnologia revolucionária de deteção de escrita-à-mão que reconhecia caracteres maiúsculos; a Microsoft lançou, para utilização nos seus próprios dispositivos e nos de outras empresas, o sistema operativo Windows CE, tendo 4 anos depois, em 2000, lançado o PocketPC 2000, um PDA que teve bastante sucesso e que vinha com a versão 3.0 do Windows CE.

Em 2001, a Microsoft apresenta o Compaq Tablet PC, um dispositivo que, devido à sua portabilidade, estava orientado para o ambiente empresarial, começando assim a divulgação do



(a) CASIO PF-3000



(b) Psion Organizer



(c) Apple Newton MessagePad

Figura 2.1: Os impulsionadores da computação móvel

termo *tablet*, embora, como mais tarde se veio a perceber, não tenha sido muito bem sucedido. Apesar desse aspeto, há a destacar que o Tablet PC possuía o Windows XP como sistema operativo, tendo sido a primeira vez que um sistema operativo orientado a computadores de secretária foi usado em dispositivos móveis. Para além das características base, incluía um sensor de rotação do ecrã e a sua interação era efetuada com toque (usando uma *stylus*), ao contrário dos habituais teclado e rato.

A mesma Microsoft viria a lançar em 2005 o Windows Mobile 5.0, um novo sistema operativo orientado para *smartphones* e baseado no anterior Windows CE.

Em 2007, naquilo que é considerado por muitos como uma verdadeira revolução no mercado dos dispositivos móveis, a Apple mostrou ao mundo um novo sistema operativo orientado para estes dispositivos, o iOS, vindo este associado a um novo *smartphone*, o iPhone. Com um ambiente bastante funcional, aliado a um design inovador, tanto do sistema operativo, como do dispositivo em si, foi um *smartphone* com enorme sucesso, tendo vendido cerca de 6 milhões de unidades até ao momento que foi descontinuado (na sua versão original)

Nesse mesmo ano, apareceu um outro sistema operativo orientado a dispositivos móveis, o Android, desenvolvido em conjunto pela Open Handset Alliance (um consórcio formado por 86 empresas de hardware, software e telecomunicações) e pela Google, sendo esta última a verdadeira detentora do sistema. O primeiro dispositivo equipado com este sistema operativo apenas surgiu no ano seguinte, em Outubro de 2008, mas no final de 2010 tornou-se o líder em *smartphones*, ultrapassando o Symbian, da Nokia, que durante anos tinha mantido a liderança.

Dois anos após o lançamento do primeiro dispositivo com Android, em Outubro de 2010, a Microsoft anuncia o Windows Phone 7, sucessor do Windows Mobile, mas que teve bastante dificuldade em singrar num mercado já dominado pelo Android e pelo iOS. Contudo, nos últimos meses tem sido um sistema cada vez mais popular e que está a ganhar mais quota de mercado, sendo esperado que com o lançamento da nova versão, Windows Phone 8, este aumento continue e que comece a ganhar uma quota de mercado significativa em relação aos atuais líderes [3].

A verdadeira guerra do *tablets* iniciou-se em 2010 quando a Apple lançou o iPad. Equipado com o mesmo sistema operativo usado no seu *smartphone*, o iOS, e tomando toda a experiência obtida com o desenvolvimento deste para o iPhone e pelo facto de este ser desenhado tendo em vista os dispositivos móveis, conseguiu revolucionar e evoluir este segmento bastante esquecido. Ainda nesse mesmo ano, a Samsung lançou o Galaxy Tab, um *tablet* equipado com uma versão modificada do Android, para assim ser melhor direcionado a estes dispositivos



com maior ecrã e com outras valências.

Seguindo o sucesso enorme do iPad, a Apple viria a lançar em 2011 o iPad 2 e em 2012 o iPad de nova geração. A Samsung lançou também uma nova versão do seu *tablet*, o Galaxy Tab 10.1, e até aos dias de hoje várias empresas de tecnologia se têm juntado a este mercado, lançando as suas próprias versões. Quase todas elas usam o sistema operativo da Google, o já referido Android, que entretanto teve as suas versões adequadas também a este mercado dos *tablets*. De entre as que não usaram o sistema operativo Android, destacam-se a RIM, que usou o seu sistema operativo BlackBerry no BlackBerry Tablet OS e a HP, que no seu HP TouchPad, usou o webOS.

Tabela 2.1: Vendas de Dispositivos Móveis a Utilizadores Finais, por Sistema Operativo, no 2Q12 (em Milhares de Unidades) [4]

Sistema Operativo	2Q12 (Unidades)	2Q12 % Mercado (%)	2Q11 (Unidades)	2Q11 % Mercado (%)
Android	98,529.3	64.1	46,775.9	43.4
iOS	28,935.0	18.8	19,628.8	18.2
Symbian	9,071.5	5.9	23,853.2	22.1
Research In Motion	7,991.2	5.2	12,652.3	11.7
Bada	4,208.8	2.7	2,055.8	1.9
Microsoft	4,087.0	2.7	1,723.8	1.6
Others	863.3	0.6	1,050.6	1.0
<b>Total</b>	<b>153,686.1</b>	<b>100.0</b>	<b>107,740.4</b>	<b>100.0</b>

Em 2012 (ver tabela 2.1), os sistemas operativos para dispositivos móveis que lideraram as vendas mundiais foram, destacadamente, o iOS e o Android, sendo ambos usados em *smartphones* e em *tablets*. De entre os *smartphones* equipados com estes sistemas operativos, os líderes de venda e, igualmente, os modelos mais poderosos e considerados topo de gama, são o Apple iPhone 5 e o Samsung Galaxy S3.



(a) Apple iPhone 5



(b) Samsung Galaxy S3

Figura 2.2: *Smartphones* com maior volume de vendas

Ainda assim, dentro dos valores de vendas mundiais, é de destacar o aumento das vendas

de dispositivos com Windows Phone da Microsoft - cresceram cerca de 115% num só ano [5], apesar de estes apenas se encontrarem no mercado dos *smartphones*, o que acaba por tornar esta percentagem ainda mais considerável, dado que o mercado dos *tablets* contribui bastante para as elevadas vendas dos sistemas operativos líderes.

*Smartphones* e *tablets* partilham a maior parte das suas características, em que as óbvias diferenças são a funcionalidade de telefone dos primeiros e o maior ecrã e resolução dos segundos. Estes dispositivos possuem bastantes funcionalidades, como clientes de email, ferramentas de organização, ligações de dados e/ou Wi-Fi, GPS e câmaras de alta resolução, podendo assim ser usados em contexto de negócio ou de simples entretenimento. Apesar de a maioria dos utilizadores acabar por não tomar partido diretamente de todas essas funcionalidades ou de nem sequer terem conhecimento das mesmas, ao existirem, permitem um sem fim de opções a serem exploradas de forma eficiente por empresas e programadores.

## 2.2 *Frameworks* de Desenvolvimento

### 2.2.1 *Frameworks* Nativos

Cada sistema operativo para dispositivos móveis possui um *framework* próprio de desenvolvimento, estando este assente sobre uma determinada linguagem de programação. Além da linguagem de programação, estes *frameworks* apenas são passíveis de serem utilizados, oficialmente, num IDE (*Integrated Development Environment*) suportado, tanto para o desenvolvimento do código da aplicação como para fazer *deploy* da mesma para o emulador e/ou para o dispositivo móvel.

Assim, dos sistemas operativos que se destacam: o Android usa Java como linguagem de programação e o Eclipse como IDE - é o IDE oficialmente suportado e recomendado, mas pode também ser usado Netbeans com um *plugin* próprio, sem quaisquer restrições; o iOS usa Objective-C como linguagem de programação e o Xcode como IDE; o Windows Phone usa C# como linguagem de programação (a partir da versão 8, o mesmo suporta também C++) e o Microsoft Visual Studio como IDE. Relativamente a Android, é possível programar aplicações para este sistema operativo usando qualquer um dos sistemas operativos para *desktop/laptop* mais utilizados - Microsoft Windows, Mac OS X e Linux -, dado que, não só o seu IDE de referência (e o alternativo) corre em todos esses SOs, como faz também parte da filosofia que a Google tem para o sistema operativo Android, que é a de que este não limite os programadores de acordo com o sistema operativo que usam. Já no caso do iOS e do Windows Phone, as aplicações têm necessariamente de ser desenvolvidas nos sistemas operativos para *desktop/laptop* das respetivas empresas, no Mac OS X da Apple e no Windows da Microsoft, dado que os IDEs associados aos *frameworks* do desenvolvimento apenas correm nesses sistemas.

Uma vez que são construídos com uma finalidade bem definida, estes *frameworks* e as suas funções, quando corretamente utilizados, permitem às aplicações tirar o máximo partido do sistema operativo a que se destinam. Sendo nativos, vão naturalmente poder aceder a todas as funcionalidades características do sistema de uma forma fácil e apropriada, como por exemplo, usar o seu sistema de notificações nativo, tirar partido da geolocalização do dispositivo, utilizar a sua lista de contactos e fazer a gestão da mesma, aceder à câmara ou usar o armazenamento interno. Todas estas possibilidades serão realizadas com a maior eficiência, dado que o resultado final de uma aplicação estará a correr diretamente sobre o sistema operativo, na camada destinada a aplicações.

A maior desvantagem na utilização destes *frameworks* acaba por ser as diferenças existentes entre os mesmos. Se um programador pretender disponibilizar a sua aplicação nas diferentes plataformas, necessita de programar a mesma em cada um dos diferentes *frameworks*, raramente reutilizando código, dado que as linguagens são diferentes (e, nalguns casos, bem distintas) e o modo de funcionamento dos sistemas operativos também. Esta situação faz com que o tempo para o desenvolvimento de uma aplicação se torne mais extenso do que o que, na maioria dos casos, seria pretendido. Além deste facto, é com naturalidade que surge uma outra desvantagem, pois deve-se ter em consideração que dadas as linguagens de programação diferentes utilizadas por cada *framework*, um programador terá de dominar todas elas. Isto faz com que tenha de as dominar todas à partida ou tenha de despende tempo a aprender, ou ainda, no caso de uma equipa de desenvolvimento, leva a que tenham de existir diferentes programadores para cada *framework*, o que aumenta os custos de desenvolvimento.

### 2.2.2 *Frameworks* Multiplataforma

Existem diversos *frameworks* multiplataforma que podem ser utilizados para o desenvolvimento de uma aplicação. Estes *frameworks* variam entre si, nomeadamente ao nível das linguagens de desenvolvimento que utilizam, dos sistemas operativos móveis para os quais conseguem disponibilizar as suas aplicações e no modo como essas aplicações vão funcionar no sistema operativo.

Deve-se, no entanto, destacar que estes *frameworks* existem e têm o seu espaço no mercado pois todos eles pretendem facilitar aos programadores/empresas o desenvolvimento de aplicações multiplataforma. Todos eles se focam numa determinada linguagem (ou conjunto de linguagens normalmente associadas, como HTML e Javascript) para fazerem aquilo a que se propõem, evitando assim a necessidade de serem necessários conhecimentos em múltiplas linguagens e múltiplos sistemas.

As desvantagens destes *frameworks*, são sempre características do *framework* a considerar e do seu modo de funcionamento, não havendo propriamente uma desvantagem que seja transversal a todos os *frameworks* multiplataforma.

Os dois *frameworks* que são mais utilizados são o PhoneGap e o Appcelerator Titanium, pelo que também são os que possuem uma maior comunidade, um maior suporte na resolução de problemas e uma maior quantidade de comparações entre si [6]. Assim, no seguinte subcapítulo, destacam-se estes dois *frameworks*.

#### 2.2.2.1 PhoneGap

O *framework* PhoneGap [7] assenta sobre uma filosofia de “*Code once, run everywhere*”, ou seja, desenvolve-se o código uma vez e este irá correr em qualquer lugar - no caso, em qualquer sistema operativo móvel. Isto torna-se possível porque este ambiente de desenvolvimento permite que aplicações web baseadas em HTML sejam implementadas e instaladas como se fossem aplicações nativas num sistema operativo. Estas aplicações tiram partido da disponibilização por parte dos sistemas operativos de um *web view*, que pode ser utilizado por aplicações nativas para mostrar conteúdo HTML (local ou remoto). O PhoneGap utiliza esta funcionalidade para carregar, assim que a aplicação é lançada, uma página web que é assim apresentada ao utilizador como se fosse uma aplicação nativa, podendo de seguida interagir com ela. Este tipo de aplicação funciona de uma forma encapsulada utilizando o dito *web view* como base, pelo que são aceites pelo sistema operativo como verdadeiramente nativas

e podem então ser instaladas como qualquer outra - inclusive, a partir das habituais lojas oficiais de disponibilização de aplicações.

Para além deste encapsulamento, uma das grandes facilidades do PhoneGap é um extenso conjunto de funcionalidades que permitem aceder às funções nativas dos dispositivos e do sistema operativo, que não estão acessíveis por normais aplicações web a trabalhar sobre o browser, como o acesso à câmara, aos contactos do utilizador, ao sistema de armazenamento interno ou aos sensores do dispositivo.

Uma vez que as aplicações desenvolvidas com PhoneGap utilizam páginas web baseadas em HTML, é de salientar que também suportam Javascript e, inclusive, todos os acessos às funcionalidades nativas são realizadas usando esta linguagem. Neste caso é o próprio *framework* que trata internamente das chamadas de Javascript, sejam elas às funcionalidades nativas ou relativas à aplicação desenvolvida pelo programador.

As vantagens deste *framework* é que acaba por funcionar em qualquer sistema operativo que suporte um *web view*, e dadas as funcionalidades nativas simplifica bastante o processo de desenvolvimento. Além disso, se se considerar que o PhoneGap é um *framework* que está bem desenhado e adequado a cada um dos sistemas operativos a que se destina, um programador não precisa de se preocupar com as particularidades que afetam o desenvolvimento de aplicações nativas, sendo esse processo mais abstrato para a equipa de desenvolvimento.

Em termos de desvantagens, é de notar que a qualidade da interface de uma aplicação desenvolvida em PhoneGap dependerá da qualidade do *web view* e do motor de renderização do sistema operativo. Nas versões mais atuais dos principais sistemas operativos, estes motores já são bastante desenvolvidos, pelo que esta torna-se cada vez menos uma questão relevante.

Uma desvantagem que ainda é verdadeiramente real, prende-se com o facto de que ao correr sobre um *web view*, e logo, sob o motor do browser, uma aplicação desenvolvida em PhoneGap não irá ter uma execução tão fluída que se encontra, por norma, numa aplicação nativa. Assim, esta desvantagem terá de ser ponderada aquando da utilização de PhoneGap para desenvolvimento, porque dependerá sempre dos requisitos da aplicação e do modo como a mesma funcionará - por exemplo, para uma aplicação que necessite de muito processamento, usar PhoneGap não será a melhor opção.

Ainda relativamente à utilização de um *web view*, torna-se claro para um programador que a interface da aplicação não será nativa, uma vez que esta irá estar assente sob HTML. Pode-se tentar criar uma interface que seja parecida com a interface nativa, mas é uma tarefa mais difícil e o utilizador final irá certamente notar diferenças. Contudo, apesar de esta razão poder parecer uma desvantagem à primeira vista, esta pode também ser considerada uma vantagem, uma vez que ao possuir uma interface mais genérica, um utilizador que use a mesma aplicação em diferentes dispositivos, com diferentes sistemas operativos, irá ter menos dificuldades em passar de um dispositivo para o outro, dado que já está familiarizado com a interface da aplicação [8].

De modo a facilitar a integração nos dispositivos e/ou o desenvolvimento de interfaces, as aplicações desenvolvidas em PhoneGap utilizam, numa considerável parte dos casos, outros *frameworks* que se focam no desenvolvimento de soluções que permitam uma melhor navegação em dispositivos móveis. O objetivo primordial deste tipo de *frameworks* começou por ser o desenvolvimento de páginas web adequadas a dispositivos móveis, tendo foco na sua usabilidade e nos standards web. Contudo, dado o potencial dos *frameworks* do género PhoneGap, têm-se também virado para o suporte e adequação a este tipo de aplicações. Assim, de entre esses *frameworks*, destacam-se o jQuery Mobile e o Sencha Touch [9], cujas características se abordam de seguida.

## jQuery Mobile

O *framework* jQuery Mobile [10] destina-se a criar interfaces, baseadas em HTML5 puro, adequadas a *smartphones* e *tablets*, tomando os ensinamentos e o suporte do bastante estabilizado e muito utilizado jQuery.

Possuindo este *framework* várias vantagens, destaca-se claramente a utilização da filosofia “escreva menos, faça mais”, pois é dos *frameworks* existentes, aquele que mais sistemas operativos distintos suporta: Android, iOS, Windows Phone, Blackberry, bada, HP webOS, Nokia Symbian e MeeGo. Assim, alguém que desenvolva usando jQuery Mobile, pode escrever aplicações ou web sites que sejam funcionais numa miríade de dispositivos, dos mais populares aos menos utilizados.

A implementação de código usando jQuery Mobile é do tipo *markup-driven*, o que faz com que o mesmo funcione à base de páginas HTML5, com utilização de Javascript e jQuery quando necessário, o que torna isto uma vantagem, dado que é uma abordagem largamente conhecida e com bastante suporte.

Outra vantagem é o facto de o código que o suporta ser bastante leve, o que faz com que a execução de uma página/aplicação seja relativamente rápida - isto tendo em comparação outros *frameworks* do género, tanto a nível de leveza do código como tempo de execução (mesmo considerando a utilização associada do jQuery, o qual é requerido pelo jQuery Mobile) [11]. A sua interface é também bastante flexível, possuindo variados temas, em que estes facilmente se adequam ou são passíveis de alterar à medida das necessidades de um programador. O jQuery Mobile possui ainda um conjunto de valências que facilitam certas tarefas aos programadores, como filtragem de listas e gestão de eventos, e a sua documentação é bastante completa, contendo vários exemplos de utilização do *framework*.

A principal desvantagem relatada por quem desenvolve com jQuery Mobile, enquanto utilização do *framework* apenas, é que, por vezes, as suas transições são algo lentas e pouco fluídas, dependendo também do sistema operativo em causa - é influenciado pela renderização utilizada pelo browser. Isto pode ser algo inoportuno quando se possuem aplicações ou páginas web que possuam conteúdo considerável, que assim demoram mais tempo a carregar. Contudo, este é um dos aspetos - tal como outros que vão sendo apontados, como diferença de aparência entre sistemas operativos diferentes ou falhas aquando do scroll de páginas - que tem sempre vindo a ser melhorado em todas as versões do jQuery Mobile, sendo que atualmente se notam cada vez menos esses problemas.

## Sencha Touch

O Sencha Touch [12] foi o primeiro *framework* existente especificamente para desenvolvimento web em dispositivos móveis. Baseado totalmente em HTML5, CSS3 e Javascript, é utilizado para que um programador possa rápida e facilmente criar aplicações móveis e/ou páginas web, com experiência de aplicação nativa, dentro de um browser.

Em termos de vantagens, destacam-se a inclusão de um conjunto de interfaces gráficas com controlos altamente otimizados para responder ao toque, a utilização de efeitos de transição muito fluídos, a adequação da resolução ao dispositivo em que estiver a correr e o suporte bastante fiável a todo o conjunto de eventos de toque que são utilizados neste tipo de dispositivos (*tap*, *double tap*, *pinch*, *swipe* e *scroll*).

O Sencha Touch possui ainda, a nível de interface gráfica, um conjunto de temas bem desenvolvidos, que tentam imitar as interfaces nativas dos sistemas operativos que suporta,

para que a experiência de um utilizador final seja o mais normal possível e para que este sinta menos que as aplicações desenvolvidas com este *framework* não são nativas. Estes temas podem também ser alterados pelo programador para melhor se adequarem às necessidades, ou pode até o programador criar novos temas, que sejam do seu entendimento, para uma experiência mais transversal e comum em todos os sistemas operativos.

O Sencha Touch apenas suporta os sistemas operativos Android, iOS e Blackberry. Apesar de não ser de todo uma desvantagem, pois é certo que suporta os dois sistemas operativos líderes de mercado, este *framework* perde em comparação com o suporte a sistemas operativos que existe noutros *frameworks* - principalmente se comparado com o jQuery Mobile.

No que toca a desvantagens mais claras, é de considerar o tamanho que o *framework* irá ocupar numa aplicação (cerca de 370KB), o que também afeta o tempo de carregamento de aplicações (cerca de 200% mais lento, em cada ecrã da aplicação, em comparação com jQuery Mobile) [11].

Uma outra característica do Sencha Touch que se pode tornar uma desvantagem, é o facto de usar uma abordagem assente em ExtJS (puro desenvolvimento Javascript). Assim, o código desenvolvido para Sencha Touch terá poucas semelhanças com uma normal página web, o que poderá assemelhar-se mais a um sentimento de desenvolvimento nativo da aplicação, e, a não ser que se possua já conhecimentos de ExtJS, torna-se um entrave ao desenvolvimento. Terá de haver um tempo de aprendizagem para perceber o seu modo de funcionamento, enquanto que HTML e Javascript puro é algo mais habitual no dia-a-dia de um programador.

### 2.2.2.2 Appcelerator Titanium

O *framework* Appcelerator Titanium [13] pretende fornecer uma forma de desenvolver aplicações móveis de qualidade e que sejam multiplataforma. Este *framework* utiliza uma API de Javascript como base de programação e assenta em dois princípios: existem componentes e modos de funcionamento que são comuns a todos os sistemas operativos móveis, pelo que o código pode ser reutilizado; existem componentes e certas convenções que são exclusivas de cada sistema operativo e estes devem ser tidos em conta, pelo que deve acontecer um desenvolvimento específico nesse caso para manter fluída a experiência do utilizador. Considerando estes princípios, o que o Appcelerator Titanium pretende é uma filosofia de “*Code once, adapt everywhere*”, que significa que a base de uma aplicação pode ser escrita de forma igual para todos os sistemas operativos, mas que depois as particularidades de cada um sejam tidas em conta e que o código seja adaptado a cada um.

O modo de funcionamento assenta, como já referido, numa API de Javascript, unificada, que é usada para escrever aplicações nativas. Os criadores pretendem assim libertar os programadores da necessidade de aprendizagem das APIs específicas de cada sistema operativo, focando-se assim na própria API do Appcelerator Titanium, mas ainda assim, terem ao seu alcance todas as vantagens e a experiência de utilizador característica que existe numa aplicação nativa [8].

Para desenvolver com este *framework* é necessário instalar ferramentas específicas para implementação de aplicações em cada sistema operativo pretendido, mas depois desse ponto estar ultrapassado, um programador só terá de utilizar o IDE próprio do Appcelerator Titanium, o Titanium Studio. Com este programa será possível desenvolver todo o código pretendido da aplicação - possui verificação de erros, completação automática, etc. -, que será depois gerada de modo a funcionar tal como acontece no normal desenvolvimento nativo e com todas as possibilidades que esse desenvolvimento oferece: é possível utilizar os simula-

dores próprios de cada sistema operativo, com depuração de código, para testar as aplicações. Instalar as mesmas aplicações diretamente em dispositivos físicos é também possível, com um sistema próprio que trata de todas as dependências e da ligação ao dispositivo. Para lançar as aplicações nas lojas de aplicações próprias de cada sistema operativo, existe também um sistema que constrói os pacotes com as aplicações finais, prontas a serem submetidas. Desta forma, todo o desenvolvimento será como se fosse nativo, permitindo ainda tratar diversas características das aplicações nativas, como notificações, configurações ou serviços em background.

Numa aplicação desenvolvida com Appcelerator Titanium, o código Javascript não é convertido e compilado previamente para a linguagem nativa do sistema operativo que se destina, como é muitas vezes assumido por quem não conhece o *framework* totalmente, mas é simplesmente executado e adequado durante a execução da mesma aplicação. Durante a execução, existem três componentes principais, que são: o código Javascript da aplicação desenvolvida pelo programador; a implementação específica da API do Appcelerator Titanium, direcionada à plataforma de destino, em linguagem de programação nativa; um interpretador de Javascript que avalia o código durante a execução. Quando a aplicação é lançada, o código da aplicação é avaliado e os objetos em Javascript serão associados/emparelhados com os objetos análogos em código nativo. Assim, quando é chamada uma determinada função associada a um objeto no código Javascript, será invocada uma função nativa que irá criar o objeto em causa em formato nativo, e entre os quais existirá uma “ponte” que permite o acesso às propriedades e métodos do objeto nativo por parte do objeto em Javascript.

De entre as vantagens do Appcelerator Titanium, destacam-se o vasto conjunto de características e funcionalidades nativas que o mesmo tem acesso, como a utilização de componentes visuais, a integração do sistema de notificações próprio de cada sistema operativo, o armazenamento interno nativo e as interfaces de ligação com serviços online. A aparência das aplicações desenvolvidas com este *framework* são também uma vantagem, dado que não existe nenhuma emulação visual, porque quando um componente é criado, o respetivo componente nativo é-lhe automaticamente associado. Desta forma as animações e o comportamento de uma aplicação serão exatamente os mesmos de uma normal aplicação nativa. Por fim, a API de Javascript utilizado pelo Appcelerator Titanium é bastante fácil de utilizar e dado que uma boa parte dos programadores possui algum conhecimento de Javascript ou de linguagens que funcionam de forma parecida, a barreira inicial para se começar a desenvolver com este *framework* é mais facilmente quebrada.

Contudo, a maior desvantagem do *framework* é que, dada sua arquitetura de funcionamento, a sua adequação a novos sistemas operativos é uma tarefa bastante árdua. Por esse facto, o Appcelerator Titanium apenas está disponível para os sistemas operativos mais utilizados, o Android e o iOS, mas existem planos para, a curto prazo, seja disponibilizado também para Windows Phone e Blackberry.

Como a camada de abstração disponibilizada pelo Appcelerator Titanium, para comunicar com o sistema nativo, é algo extensa, existem ainda certas implementações na API que não são as mais eficazes. Por outro lado, alguns componentes da interface de utilizador não têm ainda o desempenho que se espera (principalmente, componentes que lidem com quantidades elevadas de dados). Mas, à medida que erros são detetados e resolvidos e o hardware melhora, esses são alguns dos problemas que começam a desaparecer.

Depois de apresentados os principais *frameworks* multiplataforma existentes, expõem-se na tabela 2.2, uma forma sumariada, das características mais relevantes de cada um.

Tabela 2.2: Comparação entre as características mais relevantes dos *frameworks* PhoneGap e Appcelerator Titanium [14]

	PhoneGap	A. Titanium	Observações
<b>API de Javascript</b>	✓	✓	A API de Javascript interage no PhoneGap como se tratasse de típico código web, enquanto que no Appcelerator Titanium esta não é considerada como código web, mas é sim usada para interagir com código nativo.
<b>Suporte a HTML5/CSS3</b>	✓	✗	O PhoneGap é uma aplicação web que corre como nativa num <i>web view</i> .
<b>Suporte a Standards Web</b>	✓	✗	O PhoneGap é desenvolvido como uma normal página web, estando também sujeito aos mesmos problemas de compatibilidades.
<b>Código Nativo</b>	✗	✓	O Appcelerator Titanium cria uma aplicação verdadeiramente nativa, usando a API de Javascript que mapeia as funções para código nativo.
<b>Interface Nativa/Velocidade de Execução</b>	✗	✓	A velocidade de execução no Appcelerator Titanium é limitada pelo dispositivo em si, enquanto que a mesma no PhoneGap é limitada pelo <i>web view</i> característico do dispositivo.

## 2.3 Aplicações Relacionadas

Seja para comparar, seja para replicar boas funcionalidades, ou seja para conseguir oferecer algo mais ao utilizador final, quando se pretende desenvolver uma determinada aplicação, é importante proceder a uma avaliação de aplicações da mesma área se encontram no mercado. Assim, são aqui abordadas algumas das aplicações relacionadas que são mais utilizadas ou que mais se aproximam daquilo que se pretende para a aplicação a desenvolver neste projeto, tendo em conta os diferentes sistemas operativos.

### 2.3.1 iPharmacy

Disponível para iOS e Android, a iPharmacy<sup>1</sup> foi considerada a “Melhor Aplicação em Gestão de Prescrições” no livro *Best iPhone Apps, 2nd Edition*, da editora O’Reilly [15].

É uma aplicação bastante completa, sem ser demasiado complicada de utilizar. Compara medicamentos prescritos a fim de verificar se não existe problema em tomar os mesmos em conjunto, alerta o utilizar quando os deve tomar (figura 2.3b), permite a identificação de com-

<sup>1</sup><http://medconnections.com/iparmacy/>



primidos de acordo com as suas características (cor, formato), possuindo também uma base de dados bastante extensa, desde os medicamentos mais conhecidos aos menos habituais. É ainda de destacar a existência de, para cada medicamento, uma imagem com o correspondente comprimido (figura 2.3a), para uma mais fácil identificação, bem como um conjunto de informações sobre o mesmo, desde como tomar, a possíveis reações adversas e contra-indicações.



(a) Medicamento e sua lista de opções (iOS)

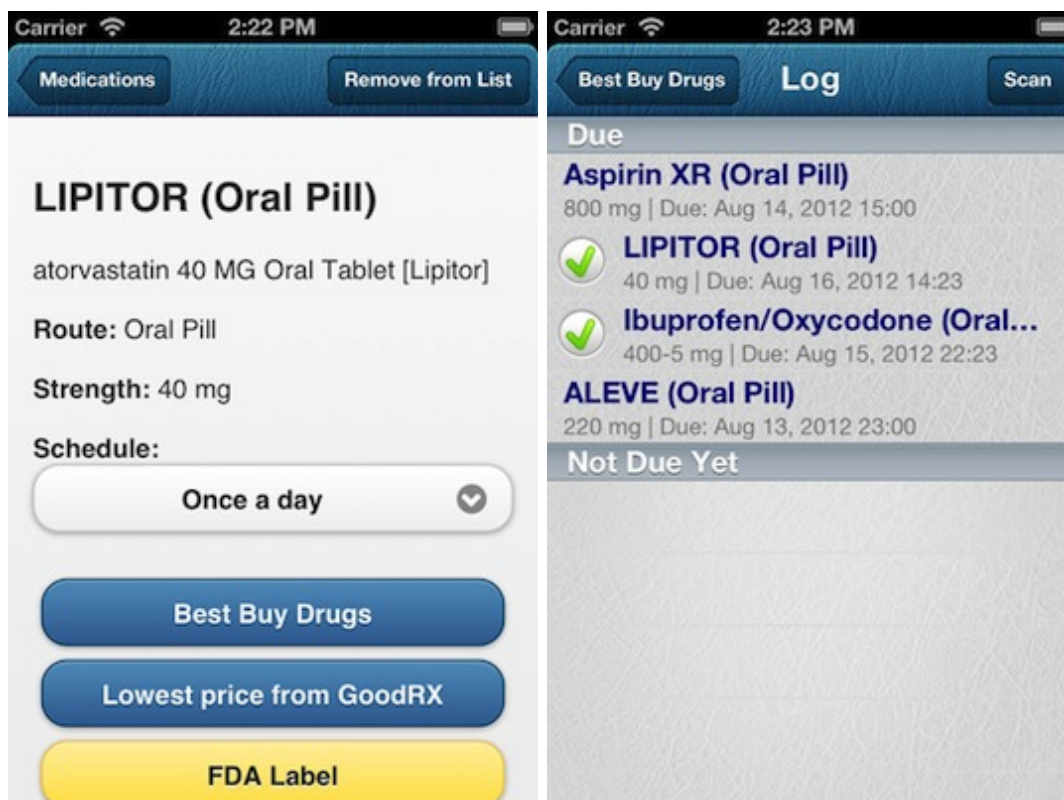
(b) Alarmes de tomas (Android)

Figura 2.3: iPharmacy - Exemplos

### 2.3.2 CR Medication Tracker

A CR Medication Tracker<sup>2</sup>, disponível para iOS, permite uma fácil organização e um fácil registo das medicações de um utilizador.

De entre as suas funcionalidades, destacam-se a possibilidade de procura por medicamentos mais baratos equivalentes ao que o utilizador pretende tomar, a informação e a definição de lembranças de toma de medicamentos (figura 2.4a), a disponibilização de informação sobre as substâncias ativas, a fácil adição de novos medicamentos à lista de medicações e a gestão de histórico (figura 2.4b).



(a) Medicamento prescrito

(b) Histórico

Figura 2.4: CR Medication Tracker - Exemplos

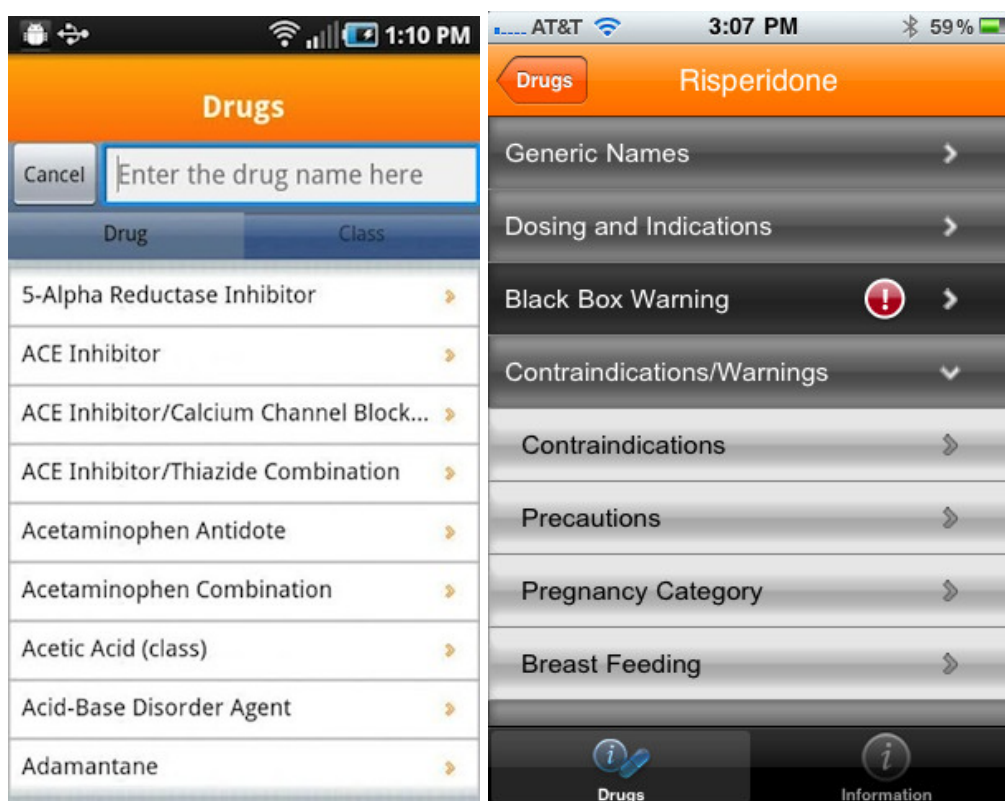
<sup>2</sup><https://itunes.apple.com/us/app/cr-medication-tracker/id553579140>

### 2.3.3 Micromedex Drug Information

Não sendo esta uma aplicação com incidência em prescrições, é relevante destacar a Micromedex Drug Information<sup>3</sup> dado que esta possui uma extensa mas concisa lista de informações sobre medicamentos.

Dá a possibilidade de procurar por mais de 4 mil termos diferentes (figura 2.5a), dando informações sobre os medicamentos mais comuns em que se encontra uma determinada substância ativa, informa sobre a terapêutica de um medicamento, a sua dosagem (seja para criança, seja para adulto), o seu tipo de administração, as suas contra-indicações e efeitos adversos, incluindo gravidez e amamentação (figura 2.5b).

Está disponível para Android e iOS e não requer qualquer ligação de dados, estando a sua base de dados incluída na própria aplicação.



(a) Procura por termos (Android)

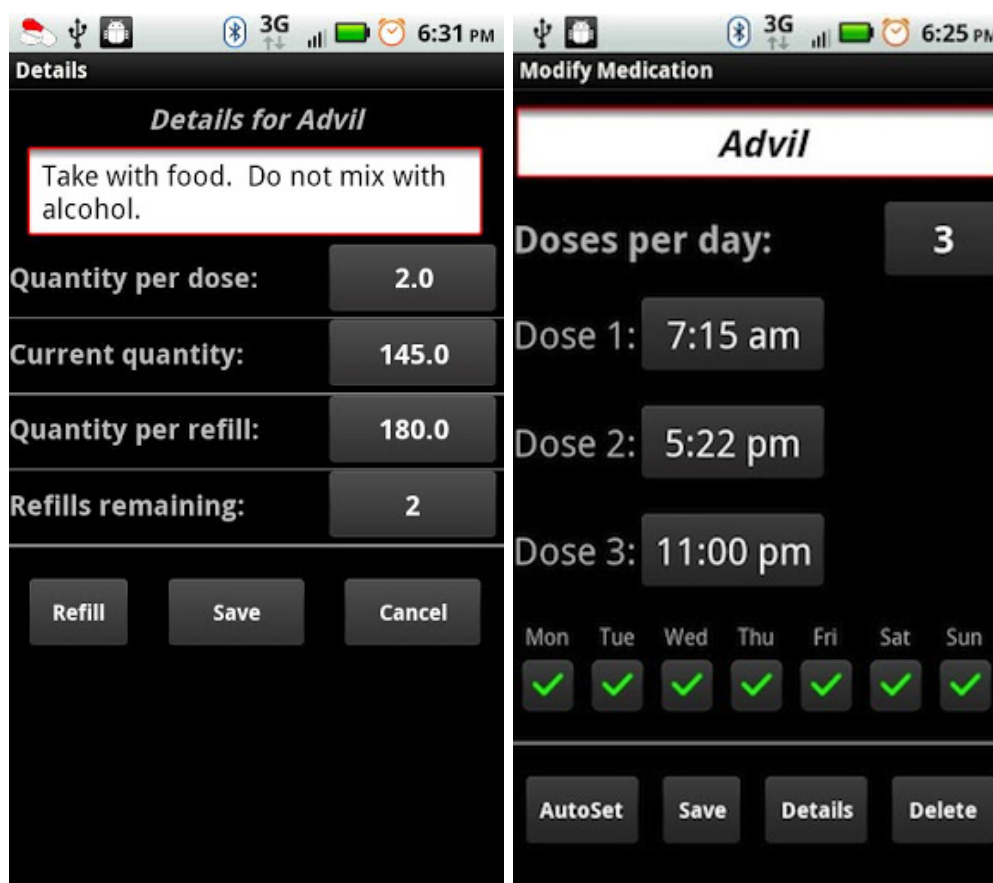
(b) Informações de um medicamento (iOS)

Figura 2.5: Micromedex Drug Information - Exemplos

<sup>3</sup><http://healthcare.thomsonreuters.com/micromedexMobile/drugInformation/index.html>

### 2.3.4 Med Minder

A aplicação Med Minder<sup>4</sup> é uma aplicação simples, direcionada ao alerta de tomas de medicamentos, disponível para Android. Sem qualquer tipo de informações próprias, deixa a cargo do utilizador a inserção dos dados do medicamento, da sua toma (figura 2.6a) e da definição dos horários dessas mesmas tomas (figura 2.6b) para posterior notificação. De destacar o facto de alertar o utilizador que uma determinada medicação está a terminar (mediante a inserção, por parte deste, da quantidade de medicação existente).



(a) Dados do medicamento e da sua toma (b) Definição de notificação de toma

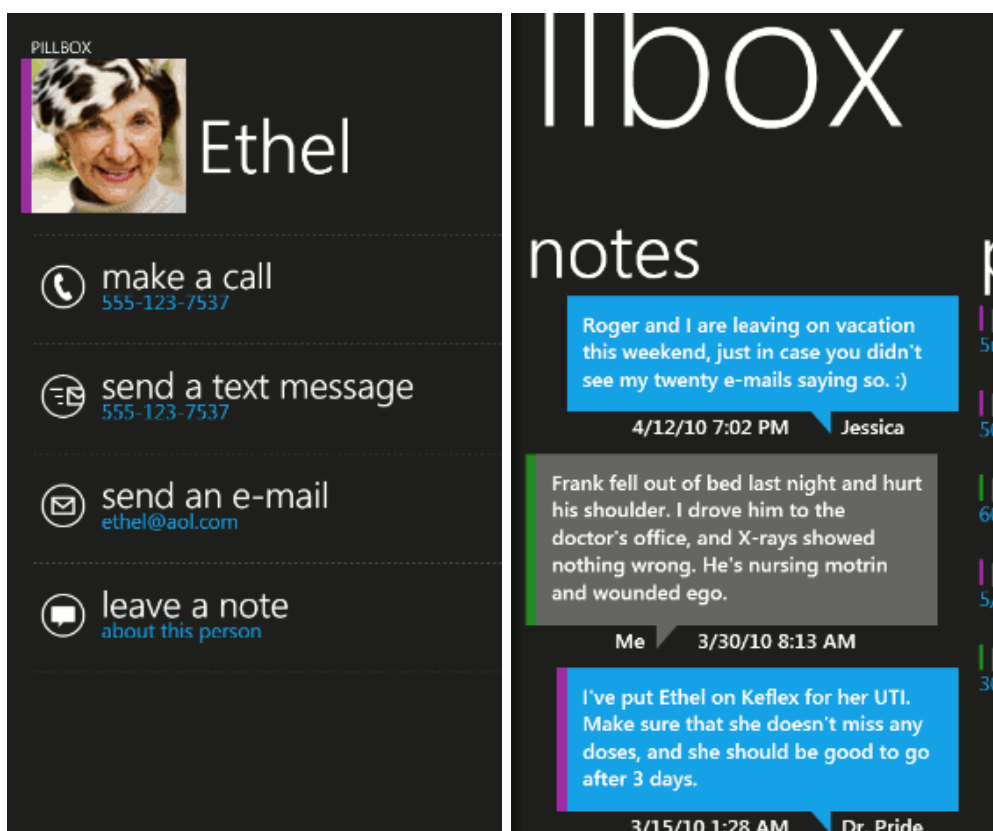
Figura 2.6: Med Minder - Exemplos

<sup>4</sup><https://play.google.com/store/apps/details?id=com.garland.medminder>

### 2.3.5 Pillbox

Disponível para Windows Phone e direcionado, principalmente, a pessoas que cuidam de outras pessoas, o Pillbox<sup>5</sup> é usado para facilitar o controlo de medicação de família e amigos.

Esta aplicação permite um conjunto de perfis associado a cada pessoa sob tratamento, contendo as informações pessoais, respetivas medicações e horas de toma. Permite a fácil comunicação com os médicos e outras pessoas que também cuidem dessa mesma pessoa, deixando notas que podem ser vistas por todos os intervenientes (sendo o armazenamento e partilha destas notas feita através do serviço Microsoft HealthVault).



(a) Dados do medicamento e da sua toma (b) Definição de notificação de toma

Figura 2.7: Pillbox - Exemplos

<sup>5</sup><http://veracitysolutions.com/pillbox>

De entre todas as aplicações aqui abordadas, salienta-se o facto de todas elas serem direccionadas para o mercado norte-americano ou então, sendo aplicações mais genéricas não usam informação específica de nenhum mercado. Assim, não existe nenhuma aplicação que seja direccionada ao mercado português, com informações em português.

## Sumário

Neste capítulo foi apresentada uma breve história e evolução da computação móvel, desde o seu início, focando as inovações principais que foram ocorrendo e os dispositivos que foram sendo lançados, até aos dias de hoje, tal como a conhecemos, nos é apresentada e a utilizamos no nosso dia-a-dia.

Foram abordados alguns *frameworks* de desenvolvimento de aplicações móveis existentes, sejam eles nativos de cada sistema operativo, sejam eles multiplataforma, analisando os aspectos em que cada um deles é vantajoso e desvantajoso, quer do ponto de vista do programador, quer também do utilizador final.

Por fim, foram analisadas algumas aplicações existentes no mercado que partilham funcionalidades com aquilo que se pretende na aplicação a desenvolver no âmbito desta dissertação, por forma a ter conhecimento do que tem sido feito e o que mais se poderá fazer.

## Capítulo 3

# Análise de Requisitos

### 3.1 Requisitos Gerais

O objetivo inicial desta dissertação era desenvolver uma aplicação para gestão pessoal de medicamentos e tratamentos, que pudesse correr num dispositivo móvel. Depois de avaliada a concorrência e estudados os principais *frameworks* de desenvolvimento, é fundamental definir os requisitos da aplicação. Esta aplicação deverá ter um público alvo bem definido, de modo a termos ideia de quais são as características que levaram à necessidade desta aplicação, o que trará de benéfico a sua criação e que considerações devem ser tomadas no seu desenho/implementação por forma a que se adeque ao público alvo.

Dado que todos nós, a qualquer altura da nossa vida, teremos de tomar algum género de medicação, podemos afirmar que qualquer cidadão faz parte do público alvo desta aplicação. Sejam pessoas jovens ou idosas, sejam mulheres ou homens, sejam pessoas com doenças esporádicas (como pneumonias, infeções ou varicela) ou doenças crónicas (como diabetes, cancro ou esclerose múltipla), ou sejam pessoas saudáveis que querem algum tipo de informação de medicamentos, esta aplicação será direcionada a todas elas.

Assim, é importante ter em conta que ao ser uma aplicação genérica, o seu funcionamento deverá ser simples e fácil de perceber, com as opções, consideradas como necessárias, rapidamente acessíveis. A aplicação deverá ser também de fácil perceção a nível visual, sem demasiados botões ou imagens que causem confusão e distraiam o utilizador do essencial.

A maioria das aplicações móveis atuais funciona tanto em *smartphones* como em *tablets*, uma vez que a maioria destes dispositivos corre o mesmo tipo de sistema operativo, sendo apenas necessário proceder a ligeiras alterações no código para se adequarem ao tipo de dispositivo final. O mesmo poderia ser feito no caso desta aplicação. Contudo, entende-se que o *smartphone*, dadas as suas características físicas e funcionais, é o tipo de dispositivo que mais se adequa ao objetivo final da aplicação, enquanto que um *tablet*, apesar de poder realizar as mesmas operações, é um dispositivo que acaba por se tornar de um uso mais doméstico ou empresarial, acabando por não andar sempre com o utilizador. De notar também que dadas as suas limitações em termos de bateria, mesmo que o utilizador ande com o *tablet* no seu percurso diário, o mesmo não está sempre ligado, o que pode invalidar eventuais alertas definidos pela aplicação.

Dada a diversidade de sistemas operativos para dispositivos móveis e considerando que se pretende que a aplicação possa ser usada pelo máximo de utilizadores possíveis, optou pelo seu desenvolvimento através de um *framework* multiplataforma. Desta forma, para além de

se conseguir cobrir um maior espectro de possibilidades, não se está dependente de um único sistema operativo e do seu sucesso ou insucesso de vendas, sendo esta uma aposta no máximo de sistemas operativos possíveis.

## 3.2 Requisitos de Funcionalidades

Considerando que a toma de certos medicamentos poderá estar dependente das características de uma pessoa, como a idade, o sexo (e, se for mulher, se está grávida), o peso, alergias ou doenças crónicas que possua, torna-se importante que a aplicação possua esta informação, para poder alertar o utilizador quando um determinado medicamento não deve ser administrado. Tal como referido anteriormente, o *smartphone* é considerado um dispositivo pessoal, pelo que a aplicação será direccionada a este tipo de utilizador. Desta forma, não se entende como necessário um sistema de perfis, pelo que apenas estará disponível a configuração de um perfil por aplicação.

Como elemento base da aplicação, é fundamental a existência de uma base de dados de todos os medicamentos disponíveis no mercado português e que assim permita ter todas as informações consideradas relevantes, como o nome, o seu código de barras, as substâncias ativas, a terapêutica a que se destina, o modo de ingestão, o tipo de embalagem em que é disponibilizado, se necessita ou não de receita médica, se é medicamento genérico e se possui algum tipo de comparticipação (e qual a percentagem da mesma) pelo Estado Português. Esta base de dados, dada a quantidade de informações que possui, terá um tamanho considerável, pelo que deverá estar alojada num servidor, funcionando como um serviço que será contactado quando o utilizador necessitar de obter as respetivas informações e adicionar o medicamento à sua lista de medicamentos a tomar.

Com o acesso à informação dos medicamentos, o utilizador pode então adicionar um determinado medicamento à sua lista de prescrições, podendo estas ser, efetivamente, resultantes de receita médica ou da toma de um medicamento não-sujeito a receita médica mas que o utilizador considera relevante. Associado a esta adição, estará uma das mais relevantes características da aplicação, que será a possibilidade de o utilizador definir alertas para a toma desses medicamentos, podendo assim ser lembrado, no momento certo, que deve tomar os medicamentos, sem ter de andar preocupado com a hora de ingestão de cada um.

As prescrições possuem uma data de início e, normalmente, uma determinada data de fim, pelo que se torna relevante manter um histórico das mesmas. Este facto é interessante na ótica do utilizador, que assim poderá verificar quais os medicamentos que tomou e quando os tomou, podendo também vir a ser, em caso de um qualquer problema que surja, uma forma de os médicos detetarem uma qualquer reação adversa posterior.

Dado que esta é uma aplicação que tem por objetivo incidir na área da saúde e que pode ocorrer alguma situação em que um utilizador se sinta mal, é relevante a existência de um contacto definido como de emergência, facilitando assim a comunicação mais atempada com o mesmo quando eventual situação se venha a verificar. Este cenário é principalmente crítico em pessoas com doenças crónicas ou cuja doença tenha a possibilidade de ressurgir repentinamente.

Por vezes, há medicamentos que não são aconselhados a tomar simultaneamente. Desta forma, a aplicação permitirá ao utilizador cruzar dois medicamentos e verificar se a sua toma em conjunto é aconselhada ou não.

Um utilizador desta aplicação terá, eventualmente, a necessidade de adquirir novos me-



dicamentos. Assim, este poderá utilizar a aplicação para verificar, de acordo com a sua localização, quais as farmácias existentes nas proximidades. Como os medicamentos têm custos e os medicamentos genéricos são mais acessíveis que os ditos normais, será permitido ao utilizador pesquisar quais os genéricos de um determinado medicamento.

O diagrama da figura 3.1 representa os casos de uso pretendidos para a aplicação.

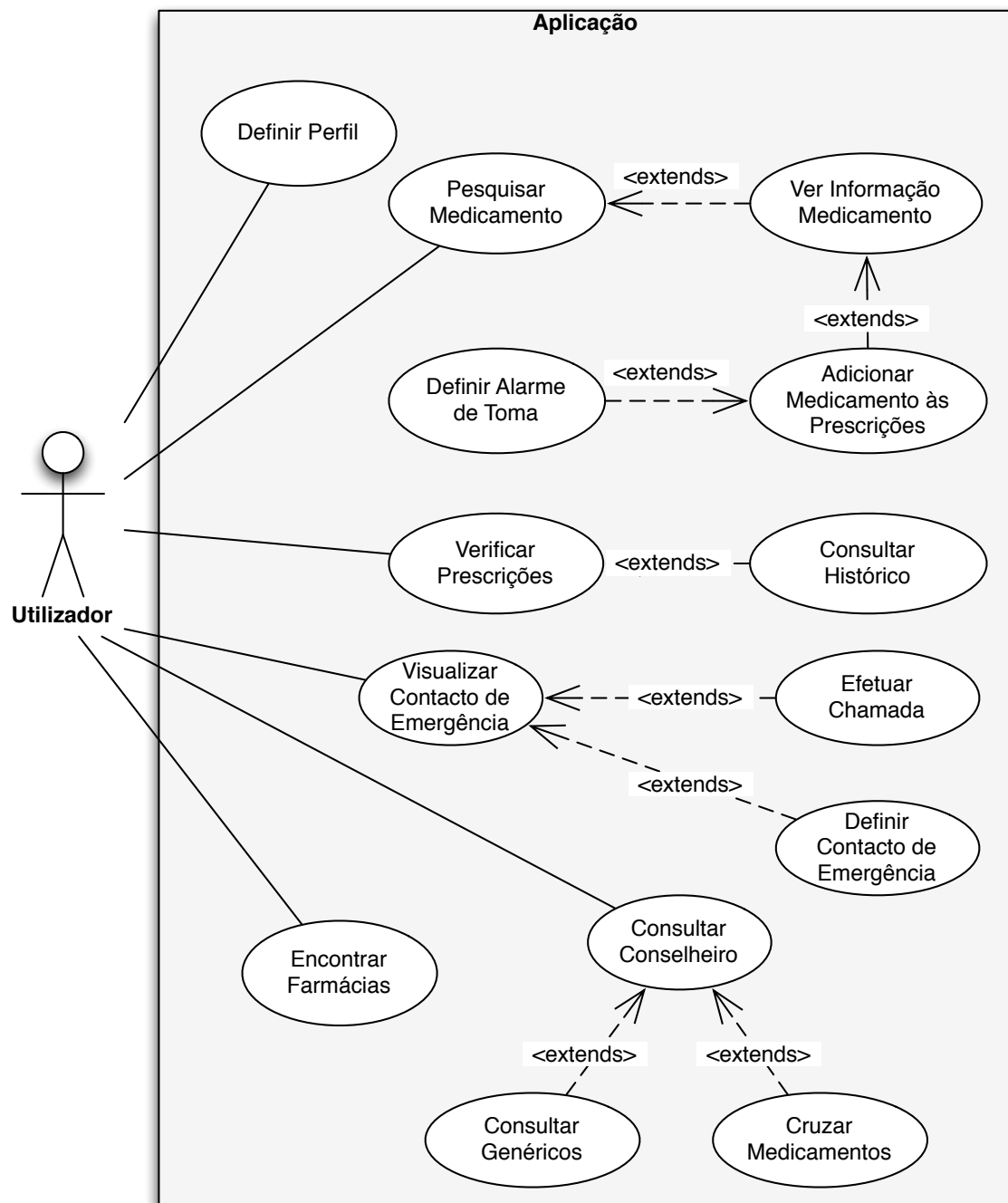


Figura 3.1: Modelo de casos de uso da aplicação

### 3.2.1 Casos de Uso

#### 3.2.1.1 Definir Perfil

Este caso de uso ocorre quando o utilizador pretende definir (ou redefinir) o seu perfil. Os dados a introduzir incluem a idade, o sexo (se for mulher, se está grávida ou não), o peso e alergias ou doenças crónicas que possua.

##### Pré-condições

- Nenhuma.

##### Sequência Básica de Eventos

1. O utilizador acede ao perfil.

Se já existirem, a aplicação carrega os dados previamente inseridos pelo utilizador.

2. O utilizador (re-)introduz os seus dados.
3. O utilizador guarda os seus dados.

##### Diagrama de Sequência

Ver figura 3.2.

#### 3.2.1.2 Pesquisar Medicamento

Este caso de uso ocorre quando o utilizador pretende pesquisar sobre um medicamento. Quando as informações sobre o medicamento são obtidas, o utilizador pode optar por apenas as ver ou por adicionar o medicamento à sua lista de prescrições. Quando o medicamento é adicionado a essa lista, é dada ao utilizador a hipótese de definir um alarme que o alerte para a toma do medicamento em causa.

##### Pré-condições

- Tem de existir uma ligação de dados ativa (WiFi/3G).

##### Sequência Básica de Eventos

1. O utilizador acede à pesquisa.
2. O utilizador introduz os parâmetros de procura do medicamento (nome ou código de barras).
3. A aplicação acede ao *webservice* para obter as informações do medicamento em pesquisa.
4. A aplicação apresenta ao utilizador as características do medicamento.

Se o utilizador quiser, pode adicionar o medicamento à sua lista de prescrições.

A aplicação permite ao utilizador definir um alarme para a toma do medicamento.

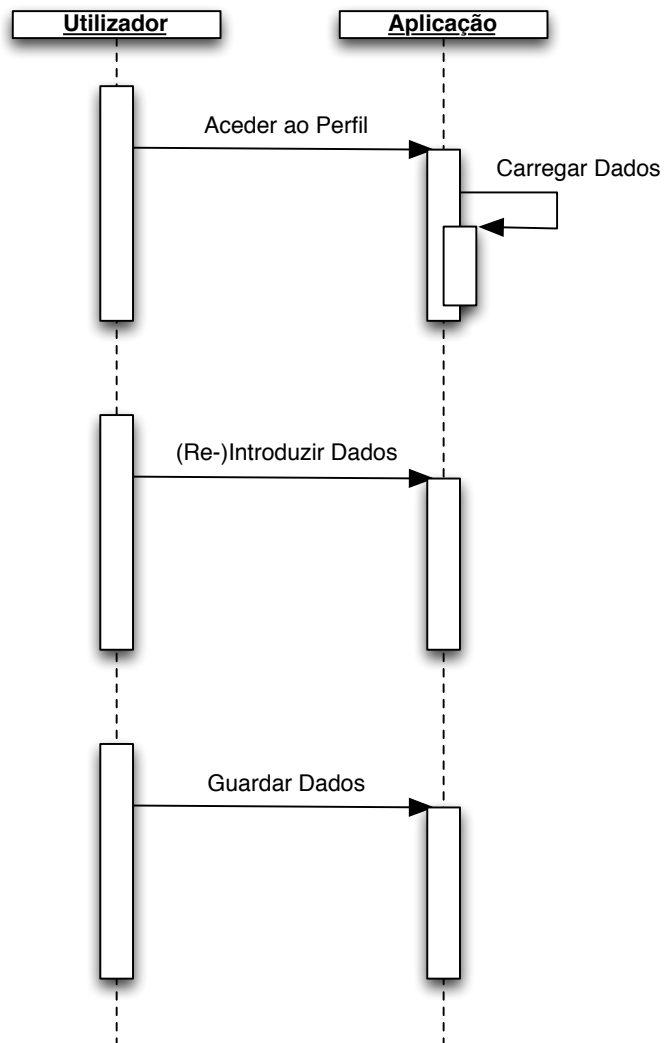


Figura 3.2: Diagrama de Sequência - Definir Perfil

### Diagrama de Sequência

Ver figura 3.3.

#### 3.2.1.3 Verificar Prescrições

Este caso de uso ocorre quando o utilizador pretende verificar a sua lista de prescrições atuais, tendo também dentro deste caso de uso a hipótese de verificar o seu histórico de prescrições.

#### Pré-condições

- Nenhuma.

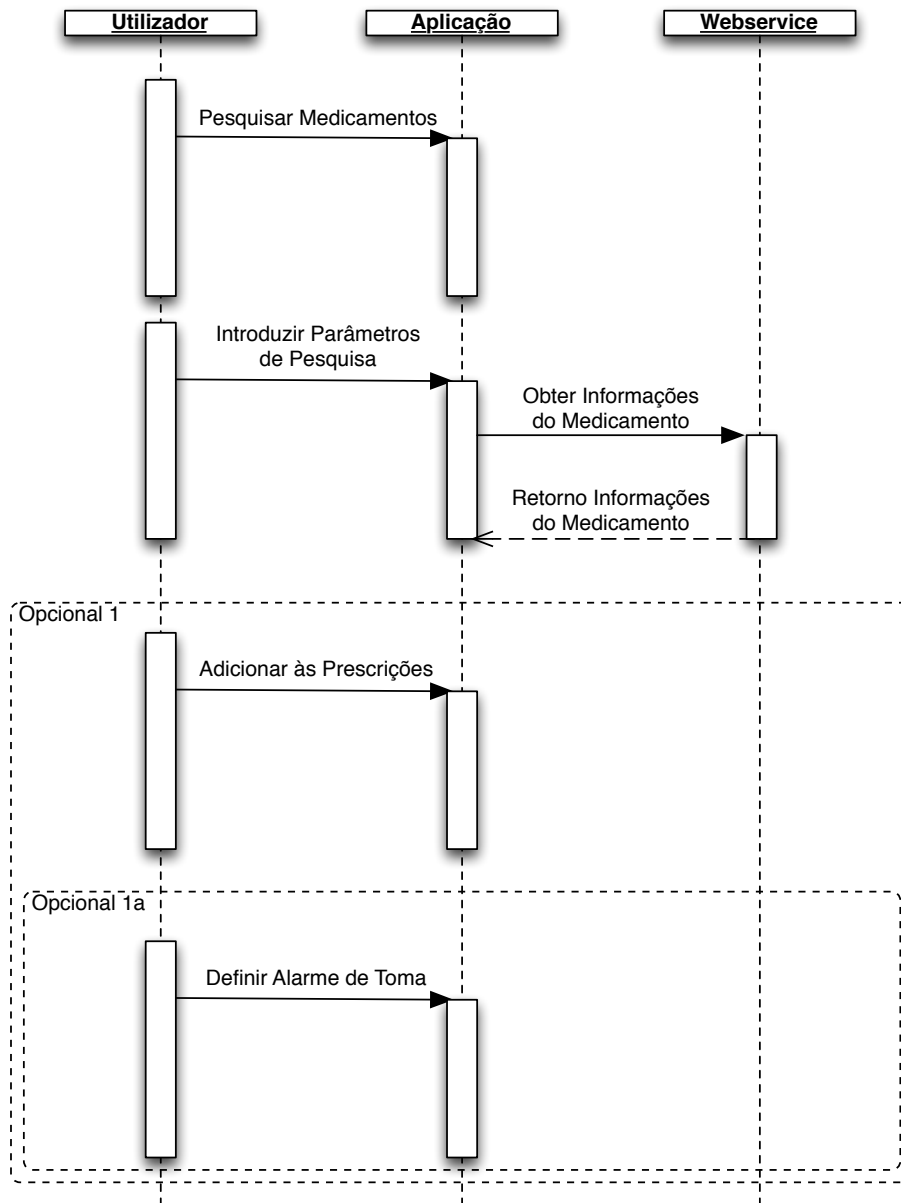


Figura 3.3: Diagrama de Sequência - Pesquisar Medicamento

### Sequência Básica de Eventos

1. O utilizador acede à sua lista de prescrições.
2. Quando estas existem, a aplicação carrega as prescrições do utilizador. Caso não existam, é mostrada uma mensagem indicando que não existem prescrições de momento.
  - O utilizador pode consultar individualmente cada um dos medicamentos prescritos.
  - O utilizador pode consultar o seu histórico de prescrições.

## Diagrama de Sequência

Ver figura 3.4.

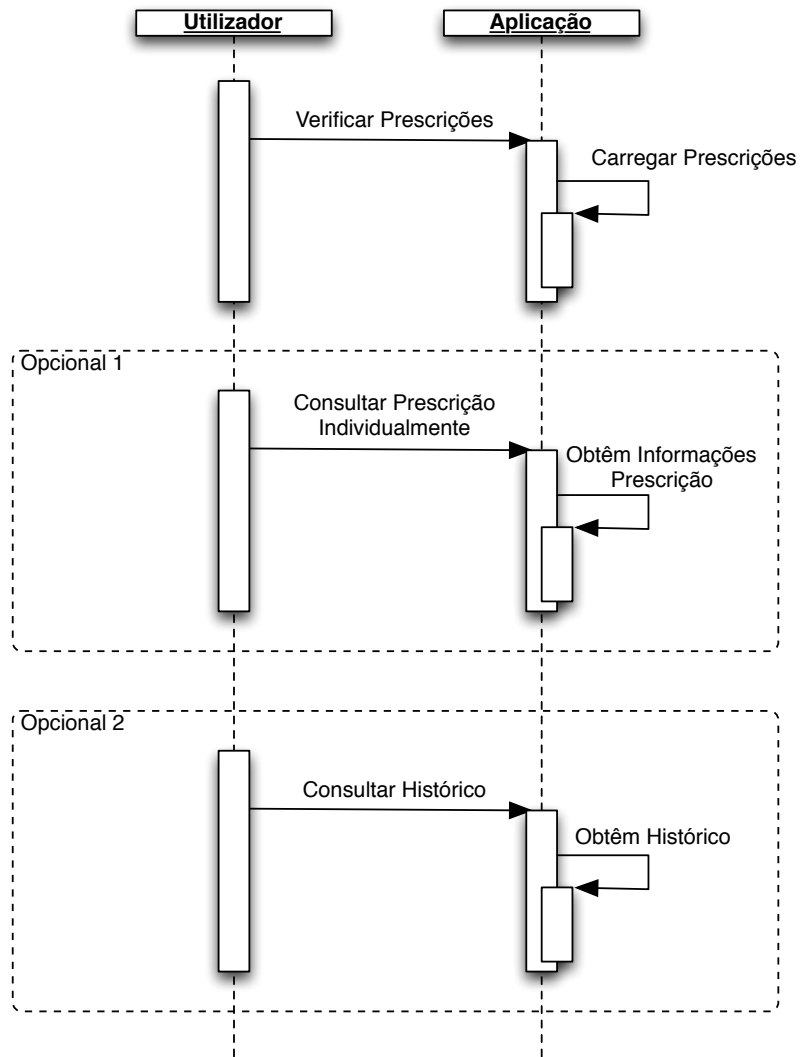


Figura 3.4: Diagrama de Sequência - Verificar Prescrições

### 3.2.1.4 Visualizar Contacto de Emergência

Este caso de uso ocorre quando o utilizador pretende visualizar qual é o seu contacto de emergência. Se o mesmo não está definido, o utilizador tem a hipótese de o definir, se o mesmo já está definido, o utilizador tem a hipótese de o alterar por outro contacto.

#### Pré-condições

- Nenhuma.

### Sequência Básica de Eventos

1. O utilizador acede ao contacto de emergência.

Se já existir, a aplicação carrega o contacto previamente definido pelo utilizador.

2. O utilizador pode carregar no contacto para iniciar imediatamente uma chamada (se este existir) ou definir/alterar o contacto.

Se optar por definir/alterar o contacto, é carregada e mostrada ao utilizador a sua lista de contactos, da qual poderá escolher um para ser o seu contacto de emergência.

### Diagrama de Sequência

Ver figura 3.5.

#### 3.2.1.5 Consultar Conselheiro

Este caso de uso ocorre quando o utilizador procura obter conselhos sobre determinado medicamento. Permite que o utilizador realize duas operações distintas: procurar medicamentos genéricos de um determinado medicamento; cruzar dois medicamentos de modo a verificar se a interação entre esses será ou não aconselhada.

#### Pré-condições

- Tem de existir uma ligação de dados ativa (WiFi/3G).

### Sequência Básica de Eventos

1. O utilizador acede ao conselheiro.
2. O utilizador escolhe uma das opções disponíveis: consultar genéricos ou cruzar medicamentos.

Se pretender consultar genéricos, o utilizador submete na aplicação o nome do medicamento principal, sendo contactado o *webservice* para obter a lista de genéricos associados.

Se pretender cruzar medicamentos, o utilizador deverá inserir na aplicação quais os medicamentos que pretende cruzar, que contactará o *webservice* e este facultará as informações relativas ao cruzamento das duas medicações.

### Diagrama de Sequência

Ver figura 3.6.

#### 3.2.1.6 Encontrar Farmácias

Este caso de uso ocorre quando o utilizador pretende pesquisar quais as farmácias mais próximas do local em que se encontra.

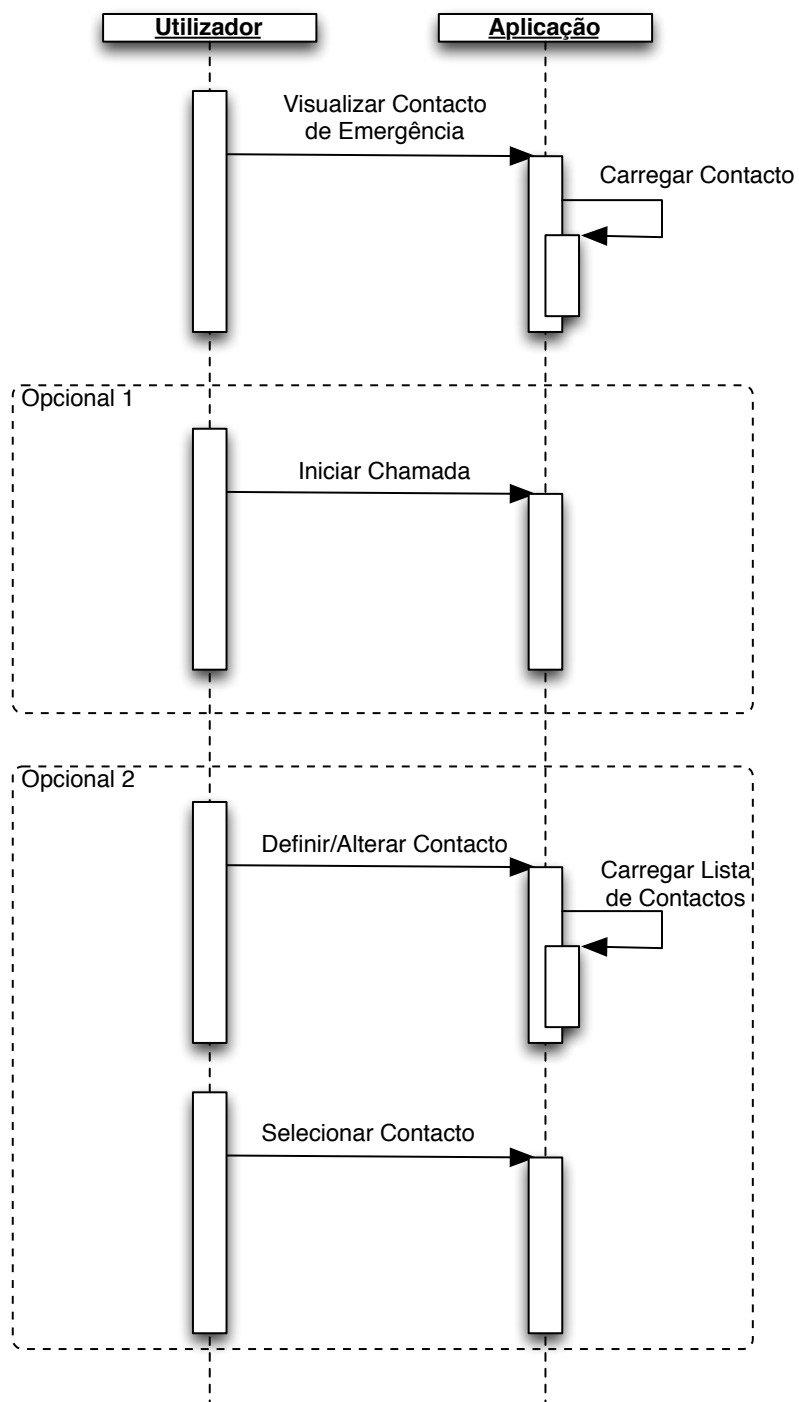


Figura 3.5: Diagrama de Sequência - Visualizar Contacto de Emergência

**Pré-condições**

- Tem de existir uma ligação de dados ativa (WiFi/3G).
- Tem de existir sensor GPS ativo.

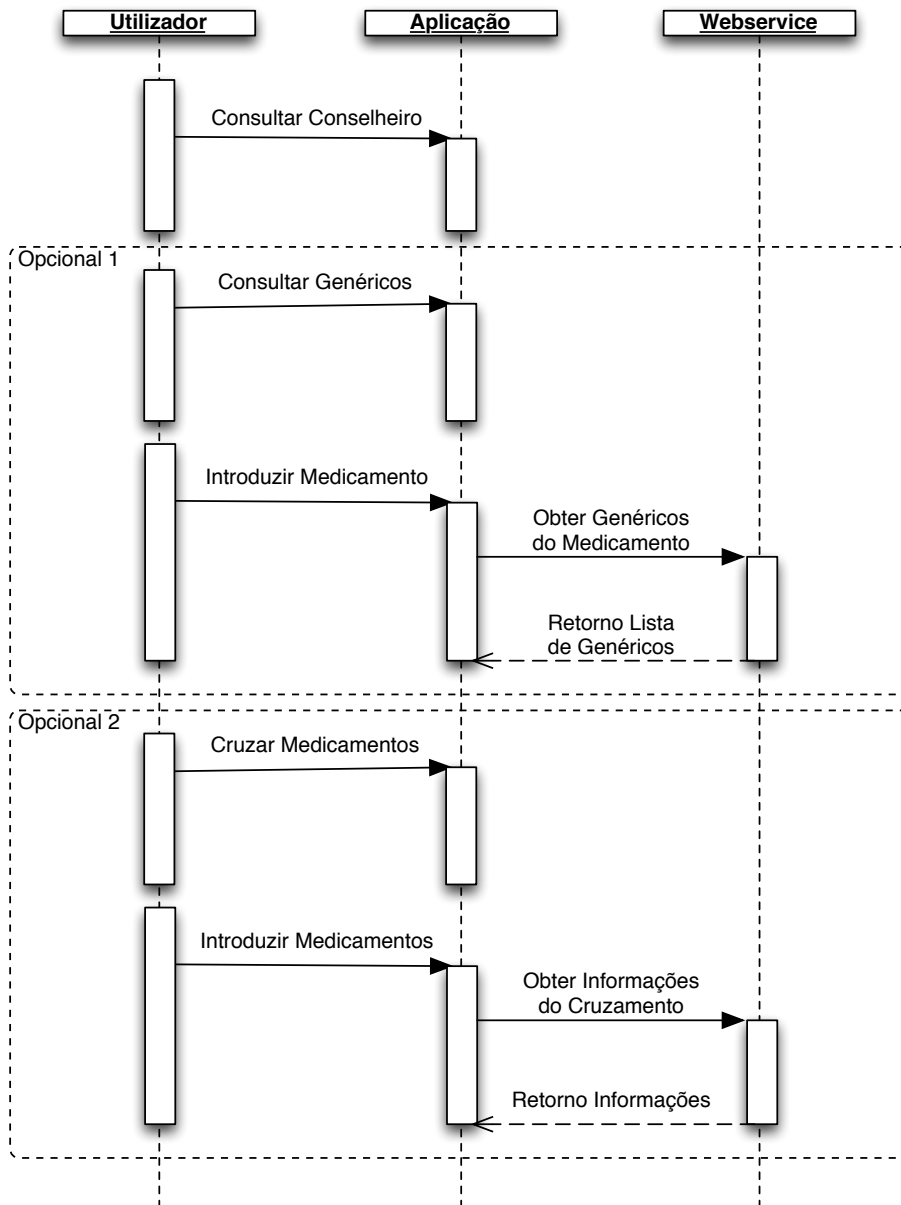


Figura 3.6: Diagrama de Sequência - Consultar Conselho

### Sequência Básica de Eventos

1. O utilizador acede à procura de farmácias.
2. A aplicação verifica a posição atual do utilizador e acede a *webservices* de localização de farmácias nas redondezas.
3. A aplicação retorna as informações dos *webservices* e apresenta-as ao utilizador.



## Diagrama de Sequência

Ver figura 3.7.

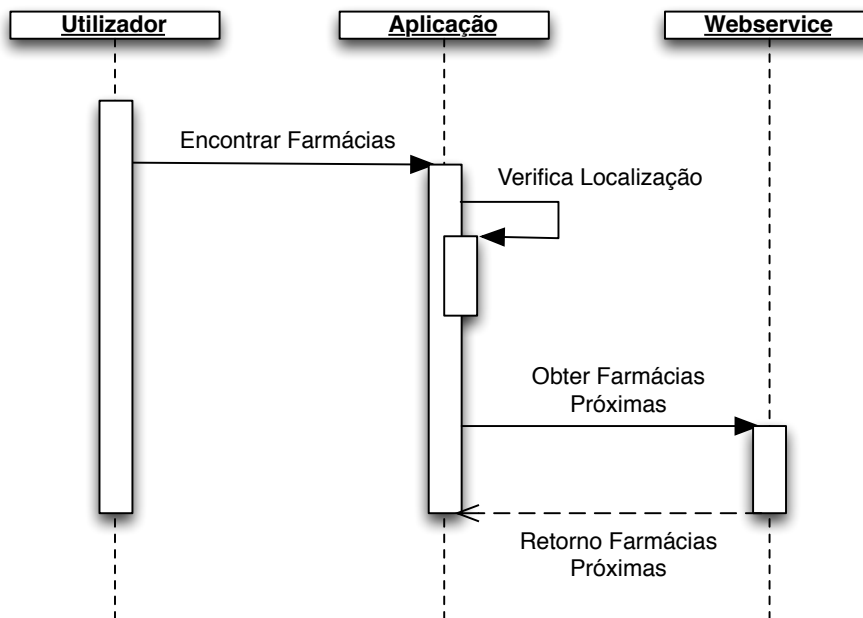


Figura 3.7: Diagrama de Sequência - Encontrar Farmácias

## Sumário

Neste capítulo foram discutidos e definidos os requisitos gerais da aplicação e os seus requisitos funcionais. Os primeiros correspondem aquilo que a aplicação deve ser no seu geral, a quem se dirige, enquanto que os segundos têm em conta aquilo que devem ser as opções presentes na aplicação e o que é que esta deve permitir ao utilizador. Nestes requisitos foram igualmente expostos os casos de uso correspondentes às funcionalidades que se pretendem implementar.



# Capítulo 4

## Desenvolvimento

### 4.1 Arquitetura do Sistema

#### 4.1.1 *Frameworks* de Desenvolvimento

No capítulos anterior apresentámos uma análise de requisitos que visa definir as funcionalidades principais da aplicação, bem como os seus utilizadores típicos. Por outro lado, no capítulo 2 discutimos distintas tecnologias que permitem o desenvolvimento de aplicação para dispositivos móveis. Assim, considerando os *frameworks* analisados, as vantagens e as desvantagens de cada um, os seus modos de funcionamento e o que estes trazem para o desenvolvimento multiplataforma, optou por se escolher o *framework* Phonegap.

As principais razões que levaram a escolher este *framework*, em detrimento do Appcelerator Titanium, são as seguintes:

- Uma das características principais que a aplicação deverá ter é o facto de poder ser utilizável a mesma no maior número possível de dispositivos móveis. Assim, com Phonegap essa característica estará coberta, dado que funciona em quase todos os sistemas operativos, enquanto que o Appcelerator Titanium apenas iria funcionar nos dispositivos com iOS, Android e Blackberry.
- Apesar de uma aplicação desenvolvida com Phonegap não funcionar de modo nativo, considerando os seus requisitos gerais e funcionais da aplicação, não irá necessitar de processamento intensivo por parte dos dispositivos em que será usada. Desta forma, o modo como o *framework* é implementado não se torna tão relevante para esta escolha.
- Depois de uma análise da documentação existente do Phonegap e também dos *plugins* desenvolvidos pela comunidade (em que alguns são posteriormente integrados no *framework*), verificou-se que com este seria possível desenvolver a aplicação na sua totalidade, com praticamente todas as funcionalidades pretendidas.
- Sendo a abordagem do *framework* baseada em HTML, que é uma linguagem familiar e largamente utilizada, entende-se que pode trazer vantagens ao nível da implementação de certas características, adequação da estrutura e na resolução de problemas que surjam.

A escolha do *framework* Phonegap implicou, por outro lado, a necessidade de recorrermos a ferramentas adicionais que facilitem e incidam na navegação em dispositivos móveis. Assim,

de entre os *frameworks* focados anteriormente, jQuery Mobile e Sencha Touch, foi selecionado o jQuery Mobile. A escolha deste em detrimento do Sencha Touch deve-se a:

- O facto deste suportar, há semelhança do Phonegap, praticamente todos os sistemas operativos móveis existentes, faz com que a sua escolha se torne relevante. Sencha Touch apenas suporta os sistemas operativos iOS, Android e Blackberry, pelo que não iria estar adequado totalmente a uma das características principais que se pretende da aplicação.
- O código que suporta o jQuery Mobile é bastante “leve” e gasta menos tempo de execução em comparação com Sencha Touch. Assim, torna-se relevante e uma mais-valia na fluidez da aplicação.
- Para além do suporte e extensas funcionalidades que a biblioteca jQuery oferece, o jQuery Mobile assenta em páginas de HTML5 e Javascript, algo habitual no dia-a-dia da maioria dos programadores. Por seu lado o Sencha Touch, dado o seu funcionamento, iria tipicamente implicar um tempo de aprendizagem de ExtJS.
- A facilidade de integração com Phonegap é maior no jQuery Mobile do que no Sencha Touch, existindo mais exemplos, tanto nos sítios oficiais destes dois sistemas como nos fóruns da comunidade de utilizadores. Desta forma torna-se mais fácil entender como os utilizar em conjunto e como resolver eventuais problemas que ocorram no desenvolvimento da aplicação.

Com a conjugação dos dois *frameworks* escolhidos, Phonegap e jQuery Mobile, obteve-se assim uma plataforma bem definida para o desenvolvimento da aplicação.

O Phonegap incide na criação da aplicação de forma a funcionar como uma normal aplicação nos diversos sistemas operativos e na disponibilização de ferramentas que tirem partido das características nativas dos mesmos, como o sistema de notificações, os serviços de armazenamento interno, a obtenção dos contactos do utilizador, a utilização da câmara ou a informação de ligações do dispositivo.

O jQuery Mobile trata da interface da aplicação, da sua adequação a cada dispositivo, da gestão e das ligações entre as páginas web que correspondem às diversas janelas da aplicação e ainda da disponibilização de um conjunto de funcionalidades que são típicas de aplicações móveis, como a filtragem de listas, a criação de formulários ou a gestão de eventos.

Estes são os principais pontos em que cada um dos *frameworks* incide diretamente. Contudo, o desenvolvimento da aplicação não é possível se não se utilizarem as linguagens HTML e Javascript, para articulação de aspetos sobre os quais os *frameworks* não incidem diretamente, como a manipulação de texto e a execução de *scripts*. A utilização da biblioteca de Javascript, jQuery [16], é também extremamente benéfica, pois disponibiliza um vasto conjunto de funcionalidades que oferecem uma grande liberdade ao utilizador a nível da criação e execução de páginas web. Esta biblioteca é, inclusive, obrigatória na utilização do jQuery Mobile, uma vez que este tira partido das suas funcionalidades.

#### 4.1.2 Base de Dados de Medicamentos

Tendo em conta que um dos objetivos da aplicação é que esta seja adequada ao mercado nacional, é bastante relevante que as informações relativas aos medicamentos sejam fiáveis. Assim, foi efetuado um pedido ao INFARMED [1] para disponibilização da sua base de dados de medicamentos, para assim tirar partido dos seus conteúdos e poder criar um protótipo final que se considere fidedigno, tendo a mesma sido gentilmente cedida por este Instituto.

A base de dados é, naturalmente, bastante extensa e completa, uma vez que tem quase todas as informações sobre os medicamentos que são aprovados pelo INFARMED. Destas informações, só algumas se tornam relevantes para esta aplicação, pelo que houve a necessidade de reduzir a base de dados ao que se considera essencial a nível de informação a utilizar.

Da análise de todas as tabelas da base de dados disponibilizada e do que cada uma contém<sup>1</sup>, consideram-se relevantes para a aplicação os seguintes atributos da base de dados:

- Nome do medicamento
- Descrição da forma farmacêutica do medicamento
- Caracterização da embalagem
- Nº de registo atribuído à embalagem, código numérico de 7 dígitos (código de barras)
- Substâncias ativas constituintes do medicamento
- Finalidade terapêutica do medicamento
- Classificação quanto ao regime de dispensa ao público do medicamento
- Descrição do grupo de produto do medicamento - normal, genérico, vacina
- Preço da embalagem do medicamento em euros
- Percentagem de comparticipação da embalagem

Assim, com estas colunas definidas como relevantes para o nosso sistema, as informações de cada uma delas foram acopladas, tendo resultado uma base de dados em XML.

Esta base de dados fica alojada num servidor e os seus dados são obtidos pela aplicação usando um *webservice*. Este *webservice* permite a procura das informações de um determinado medicamento usando o seu nome ou o seu código de barras. Quando existe informação relativa ao medicamento a procurar na base de dados, esta é retornada para a aplicação.

A figura 4.1 representa uma visão geral do sistema, considerando a aplicação e a base de dados remota.

---

<sup>1</sup>Dado que a base de dados e respetiva tabela são propriedade do INFARMED, o seu conteúdo e formato não é aqui apresentado, pelo que apenas são enumeradas as colunas que contém e que são utilizadas na implementação do sistema.

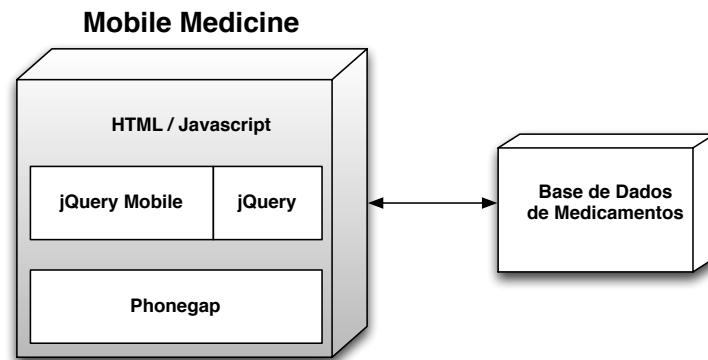


Figura 4.1: Visão geral do sistema.

## 4.2 A Aplicação

A aplicação desenvolvida, que denominamos por Mobile Medicine, é composta por um conjunto de ecrãs, com botões simples mas explicativos para navegação dentro da mesma. Quando o utilizador inicia a aplicação é-lhe apresentado o ecrã inicial (figura 4.2), que contém os botões de acesso às Prescrições, à Pesquisa, ao Contacto de Emergência e à definição do Perfil.



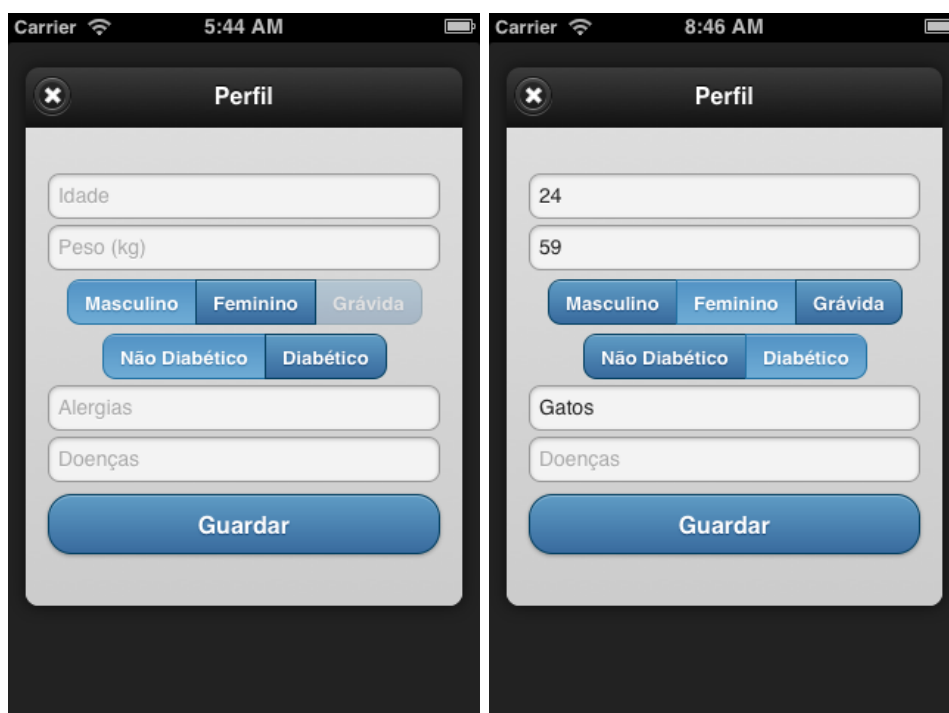
(a) iOS

(b) Android

Figura 4.2: Mobile Medicine - Ecrã Inicial

## Perfil

Quando o utilizador acede ao Perfil, a aplicação verifica se já existem dados do utilizador definidos. Caso não existam, é apresentado o perfil por omissão (figura 4.3a), sem nenhum tipo de preenchimento. Em caso afirmativo, os dados do utilizador são carregados e é feita a adequação da janela de perfil a esses dados - preenchimento dos campos de texto, modificação dos *radio buttons* (figura 4.3b). O utilizador pode fechar a janela de diálogo sem fazer qualquer tipo de introdução ou guardar os seus dados depois de preencher/alterar os campos (nenhum dos campos é obrigatório).



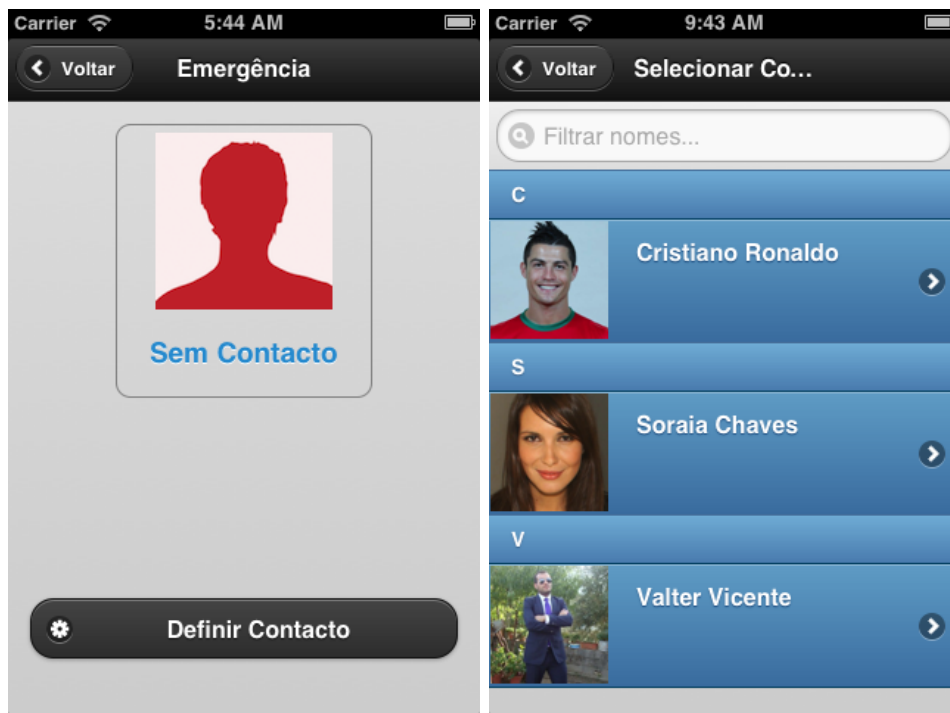
(a) Sem dados prévios

(b) Dados carregados

Figura 4.3: Mobile Medicine - Definir Perfil

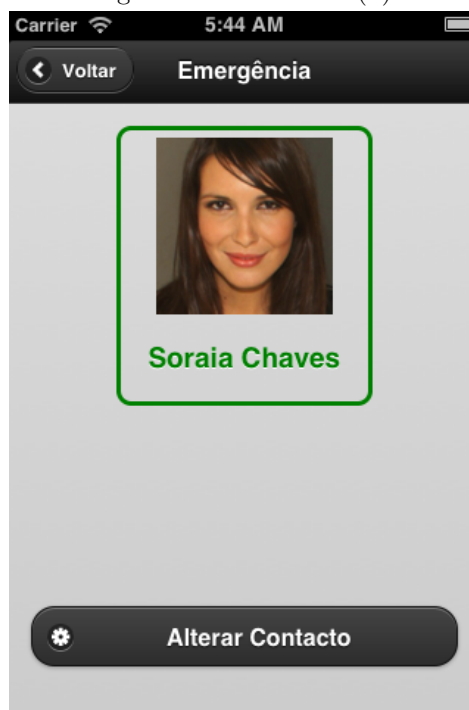
## Contacto de Emergência

Acedendo à opção de Contacto de Emergência, a aplicação verifica se existe ou não algum contacto já definido. No caso de tal não acontecer, é apresentada uma janela genérica (figura 4.4a), sem qualquer contacto passível de ser contactado e com o botão “Definir Contacto”. Se o utilizador premir este botão, será direcionado para o ecrã seguinte no qual existirá a sua lista de contactos (figura 4.4b). Aqui poderá escolher o seu contacto de emergência - o utilizador pode utilizar o campo para filtrar nomes e assim mais rapidamente encontrar o contacto pretendido. Depois de escolhido, a aplicação encaminha o utilizador para o ecrã anterior, mas em que desta vez já existirá um contacto definido, com o nome e foto associada, e tendo o botão “Definir Contacto” passado a denominar-se “Alterar Contacto” (figura 4.4c). Este último ecrã é também o tipo de ecrã que surge ao utilizador caso ele já tenha um contacto de emergência definido, quando acede a esta opção.



(a) Sem contacto de emergência

(b) Lista de contactos



(c) Contacto de emergência definido

Figura 4.4: Mobile Medicine - Contacto de Emergência



## Pesquisa

Quando um utilizador pretende pesquisar um medicamento, deverá aceder à opção Pesquisa. Nesta, são dados ao utilizador dois tipos de procura do medicamento: procurar pelo nome ou procurar pelo código de barras (figura 4.5a). Na procura por nome, o utilizador tem uma caixa de texto específica onde terá de inserir o nome do medicamento a procurar. Já na procura por código de barras, o utilizador possui duas opções: poderá introduzir o número do código de barras no campo específico ou optar por usar a câmara do seu dispositivo, que irá recorrer a um leitor de códigos de barras embutido para analisar o mesmo (figuras 4.5b e 4.5c).

Depois de o utilizador ter efetuado a pesquisa, é direcionado para a página de informações do medicamento. Esta possui o nome do medicamento pesquisado, as substâncias ativas que o compõe, a terapêutica em que o mesmo se insere, qual o formato do medicamento e a embalagem em que este se encontra. É ainda incluída informação sobre o seu grupo de medicamentos em que se insere, o seu regime de prescrição, e qual o preço atual no mercado bem como a sua percentagem de comparticipação (figuras 4.6a e 4.6b).

De destacar que o utilizador pode facilmente obter informações sobre as substâncias ativas que compõem o medicamento, bastando para isso que clique nas mesmas. A informação a apresentar sobre as substâncias ativas, é a existente na versão portuguesa da Wikipédia<sup>2</sup>, que será aberta num navegador embutido na aplicação (figura 4.6c), permitindo que quando o utilizador termine de ver as informações, bastará voltar atrás e terá ao seu dispor novamente o ecrã de informações do medicamento que pesquisou.

É aqui, na apresentação das informações de um medicamento, que o utilizador pode adicionar esse medicamento à sua lista de prescrições, bastando que clique no botão “Adicionar às Prescrições”. Quando este é clicado, é pedido para confirmar se quer mesmo adicionar às prescrições (figura 4.7a) e, em caso afirmativo, o utilizador é encaminhado para o ecrã seguinte, onde é informado que foi adicionado com sucesso às prescrições e questionado se pretende definir alarme para o medicamento (figura 4.7b). Caso o utilizador pretenda definir alarme, é-lhe apresentado o ecrã para definição do mesmo (figura 4.7c), onde poderá escolher os dias da semana da toma tal como o horário dessa toma. Para finalizar, basta pressionar o botão “Definir Alarme”, sendo notificado que essa definição foi bem sucedida e é depois encaminhado para a lista de prescrições.

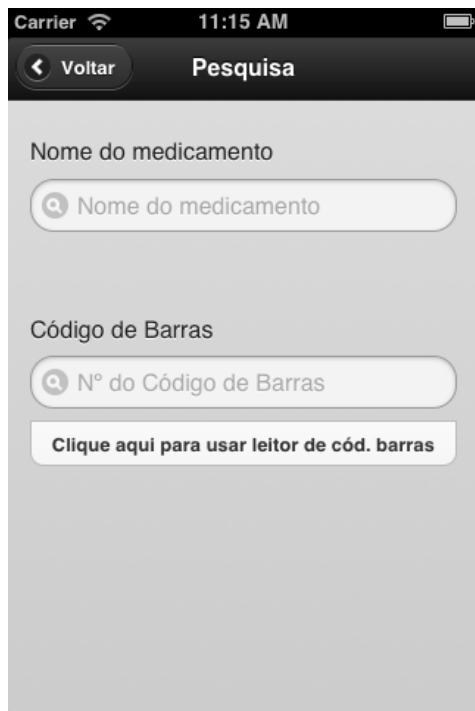
## Prescrições

Quando um utilizador pretende verificar as suas prescrições, deverá aceder à opção Prescrições. Nesta, ser-lhe-á apresentada a lista com as suas prescrições atuais, tendo também aqui o acesso ao histórico das suas prescrições, clicando no botão “Histórico” (figura 4.8a).

É possível consultar cada uma das prescrições individualmente, onde o utilizador poderá verificar quais as substâncias ativas do medicamento, a data de início da prescrição e o seu alarme de toma. O utilizador tem a liberdade de alterar aqui os alertas de acordo com quaisquer alterações que existam. Se este não quiser mais utilizar os alertas, bastará que remova a seleção dos dias na lista (figura 4.8b). Caso a necessidade de toma do medicamento prescrito termine, bastará ao utilizar pressionar o botão “Terminar Prescrição”, para esta ser removida da lista de prescrições e ser adicionada ao histórico.

---

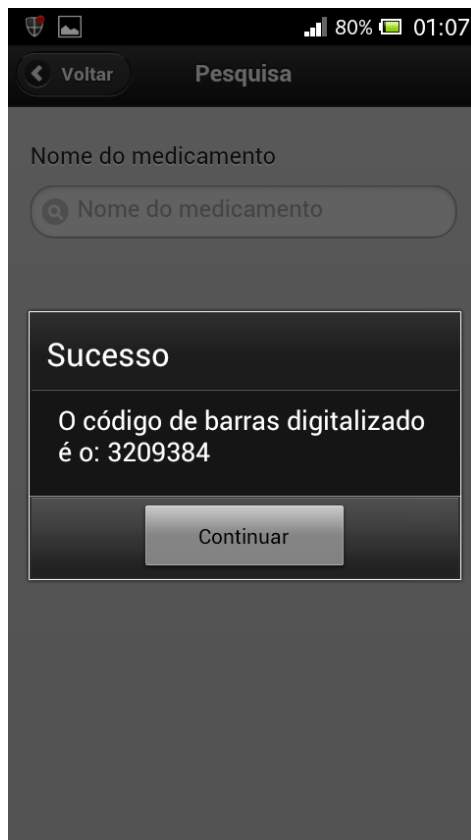
<sup>2</sup><http://pt.wikipedia.org>



(a) Ecrã de pesquisa por nome ou por código de barras

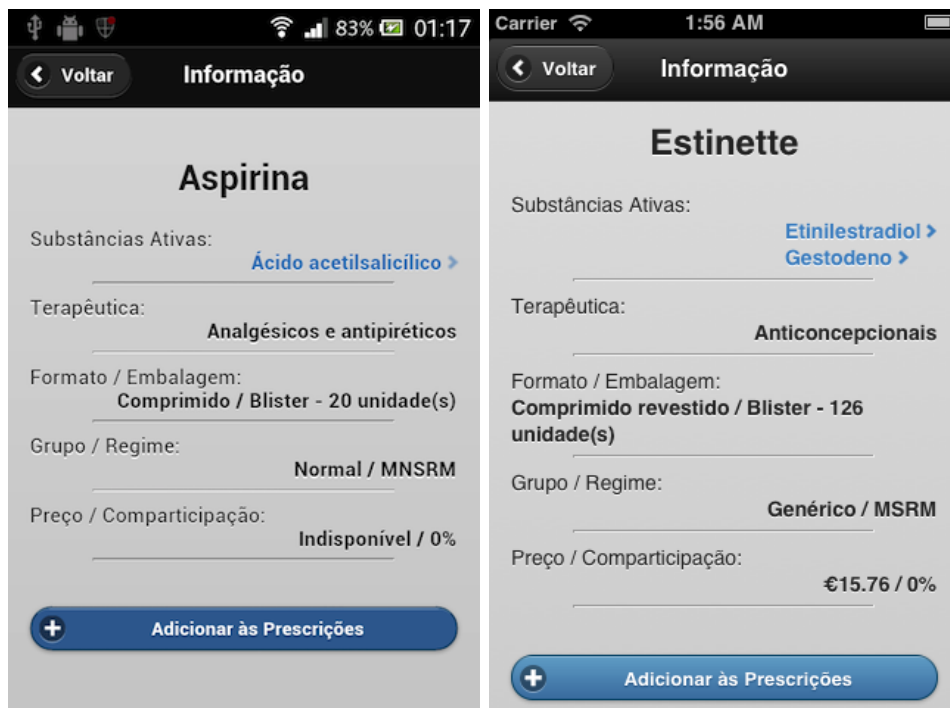


(b) Ler código de barras



(c) Código de barras lido com sucesso

Figura 4.5: Mobile Medicine - Pesquisa



(a) Resultado da pesquisa por leitura de código de barras (b) Resultado da pesquisa por nome de medicamento

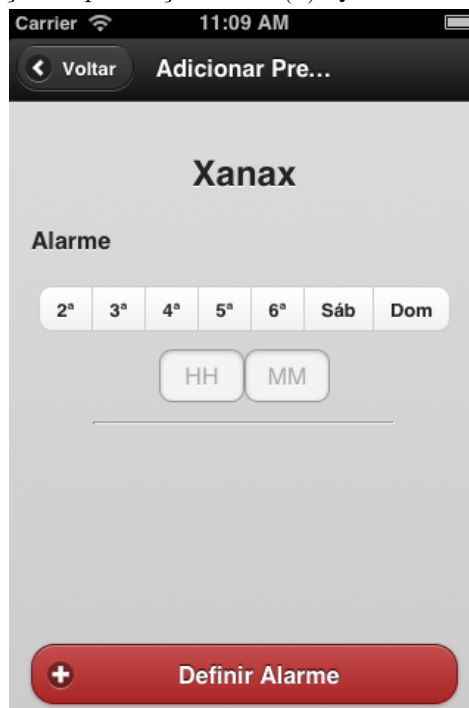


(c) Informações da substância ativa, cortesia da Wikipédia

Figura 4.6: Mobile Medicine - Informações do Medicamento



(a) Confirmar adição às prescrições (b) Questionar definição de alarme



(c) Ecrã de definição de alarme

Figura 4.7: Mobile Medicine - Adicionar às Prescrições e Definição de Alarme

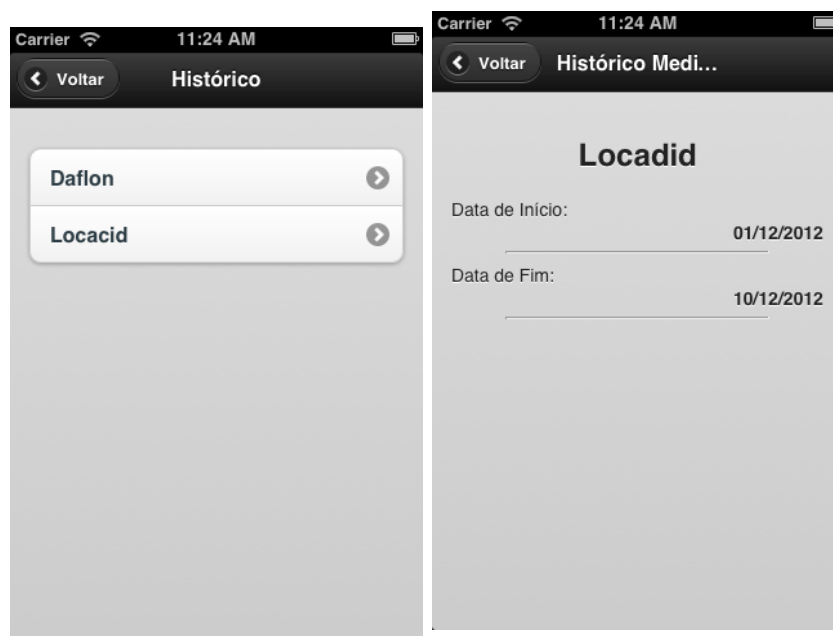
O histórico contém os medicamentos que já fizeram parte das prescrições do utilizador, ou seja, que já terminaram, sendo este histórico mostrado também no formato de lista, tal como as prescrições atuais (figura 4.9a). Ainda como estas, é também possível ao utilizador consultar cada uma delas, verificando entre que datas a toma destas ocorreu (figura 4.9b).



(a) Ecrã de prescrições

(b) Informação da prescrição

Figura 4.8: Mobile Medicine - Prescrições



(a) Histórico de prescrições

(b) Informação de prescrição passada

Figura 4.9: Mobile Medicine - Histórico

## 4.2.1 Usabilidade da Aplicação

Em 1990, Jakob Nielsen estabeleceu 10 heurísticas de usabilidade [17]. Como esta aplicação requer bastante interação com o utilizador, a sua usabilidade foi algo a ter em consideração.

A heurística “Visibilidade do estado do sistema: o sistema deve sempre manter os utilizadores informados sobre o que está a decorrer, através de feedback apropriado em tempo aceitável” é vista durante operações que necessitem de algum tempo para serem executadas. Por exemplo, quando o utilizador pesquisa por um medicamento, a obtenção dos dados pode ser demorada, pelo que o utilizador é notificado com uma imagem de carregamento (figura 4.10a) ou quando este define o alarme de toma, é apresentada uma mensagem em como este foi corretamente definido (figura 4.10b).



Figura 4.10: Heurística - Visibilidade do Estado do Sistema

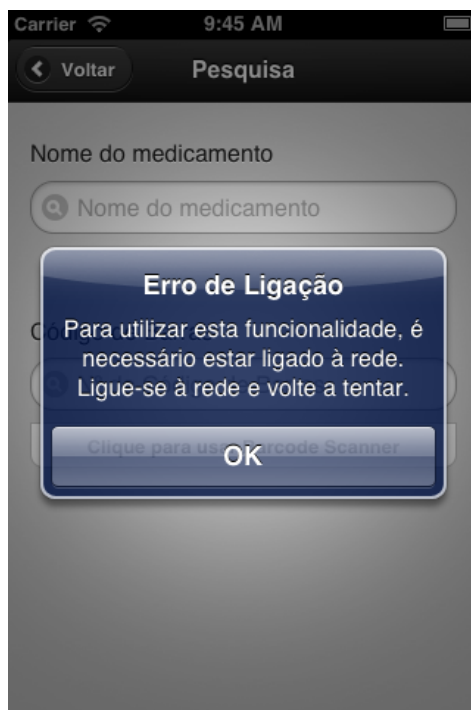
A heurística “Controlo e liberdade do utilizador: os utilizadores habitualmente escolhem funcionalidades por acidente e necessitam de uma clara ‘saída de emergência’ para saírem do estado não pretendido” está presente quando o utilizador carrega em botões que desencadeiam ações que mexem com os dados atuais da aplicação. São exemplos desta situação em que o utilizador carrega no botão para “Terminar Prescrição” (figura 4.11a) e quando este insere caracteres não-numéricos no campo de pesquisa do código de barras 4.11b).

A heurística “Ajuda os utilizadores a reconhecer, diagnosticar e recuperar de erros: as mensagens de erro devem ser escritas em linguagem corrente, indicando precisamente o problema e sugerindo uma solução construtiva” é considerada e utilizada quando o utilizador pretende usufruir da pesquisa de medicamentos e não possui qualquer ligação de dados ativa. É referida essa razão ao utilizador e aconselha-se o mesmo a ligar-se à rede para poder proceder com a pesquisa (4.12a).



(a) Confirmação da remoção da prescrição (b) Erro de inserção de código de barras

Figura 4.11: Heurística - Controlo e liberdade do utilizador



(a) Sem ligação de rede

Figura 4.12: Heurística - Ajuda os utilizadores a reconhecer, diagnosticar e recuperar de erros



## Sumário

Neste capítulo foi definida a arquitetura do sistema, incluindo a justificação das opções pelas sistemas de desenvolvimento, destacando as razões porque foram estes os escolhidos em detrimento dos outros *frameworks* analisados. Foi igualmente apresentada, ainda de uma forma sumária dadas as restrições de utilização, a base de dados de medicamentos a utilizar, desde a base de dados original do INFARMED, até à versão final a utilizar pela aplicação.

Foi também neste capítulo que foi apresentado o resultado final da aplicação, com diversas capturas da mesma em funcionamento, com a explicação pormenorizada do que ocorre em cada um dos momentos e com as opções que o utilizador tem no seu manuseamento. Foram ainda abordadas algumas das heurísticas que foram consideradas e implementadas na aplicação.



# Capítulo 5

## Conclusão

### 5.1 Sumário do Trabalho Realizado

O resultado desta dissertação é uma aplicação para gestão de medicamentos em dispositivos móveis, sendo esta aplicação direcionada ao público português e passível de ser utilizada em todos os sistemas operativos móveis relevantes no mercado. Analisando a realização desta dissertação, há marcos importantes que vale a pena salientar e que serão referidos seguidamente.

Ao analisar a história da computação móvel, verificou-se que esta foi uma área da tecnologia, como tantas outras, que inicialmente demorou a desenvolver. Contudo, com o passar dos anos, foi crescendo bastante, estando atualmente no seu maior apogeu de sempre. Torna-se relevante estar ao corrente desta situação de modo a analisar o mercado existente e definir da melhor forma os objetivos para uma aplicação, porque com o desenvolvimento da área, existe uma maior quantidade de soluções que podem ser utilizadas.

Todas as soluções que existem, apesar de diferentes, acabam por ter abordagens em comum. Assim, apesar de cada uma das soluções possuir características próprias e *frameworks* de desenvolvimento próprios, existem atualmente *frameworks* que tentam facilitar a tarefa dos programadores no desenvolvimento de aplicações móveis. De entre estes, foram estudados o Appcelerator Titanium e o Phonegap (com a posterior análise de jQuery Mobile e Sencha Touch), de forma a entender quais as vantagens e desvantagens

Como em qualquer área, quando se pretende desenvolver algo, é extremamente relevante conhecer o mercado, pelo que foram analisadas algumas aplicações relacionadas com o objetivo da nossa aplicação. Estas foram importantes não só para perceber o que existe, como também para ter ideia de algumas boas funcionalidades que são implementadas e para entender o que mais se pode fazer para beneficiar o utilizador.

Tendo em vista o objetivo proposto e as análises feitas, procedeu-se à definição dos requisitos da aplicação. Foram definidos os requisitos gerais da aplicação, tendo-se determinado em que contexto a aplicação se insere, a quem esta é direcionada e em que tipo de dispositivos irá ser usada. Por outro lado, os seus requisitos funcionais definiram de que forma a aplicação iria obter os dados que utiliza, quais as opções que se pretendia para a aplicação e qual a sequência de execução dessas opções.

Passando ao desenvolvimento, foi definida a arquitetura da aplicação, pelo que teve de se escolher um *framework* a utilizar, tendo a escolha recaído, depois de analisados os prós e os contras da sua utilização, no Phonegap. Associado a este, foi escolhido o jQuery Mobile.

Com essa escolha feita, era importante perceber que dados usar na aplicação, pelo que depois de analisar a base de dados disponibilizada pelo INFARMED, foi criada uma base de dados mais pequena, apenas com os dados considerados relevantes para a aplicação, e definida a forma como a aplicação comunica com essa base de dados.

A aplicação, denominada *Mobile Medicine*, foi então desenvolvida de acordo com os requisitos definidos e com a maioria dos casos de uso considerados, sendo aqui apresentadas no documento algumas capturas de imagem do seu funcionamento.

## 5.2 Trabalho Futuro

Tendo em conta a aplicação final obtida, é importante considerar que funcionalidades futuras poderão ajudar a melhorar a mesma.

Dado que nem todos os casos de uso pretendidos para a aplicação foram implementados, esse seria, naturalmente, dos primeiros aspetos a melhorar na aplicação.

Será sempre relevante continuar a monitorizar as atualizações que vão sendo lançadas dos *frameworks* Phonegap e jQuery Mobile, de forma a perceber se alguma dessas traz novas funcionalidades que permitam uma implementação mais adequada das opções disponibilizadas pela aplicação e, até, façam com que surjam novas ideias de opções para serem implementadas.

Recorrendo a uma base de dados com mais informações, em que se associasse as informações obtidas pelo INFARMED com outras informações sobre os medicamentos que não estão contempladas, poder-se-ia criar uma aplicação mais poderosa e com mais funcionalidades.

Recorrendo a essa base de dados, uma ideia a nível de interface seria que existisse uma imagem com a caixa/embalagem de cada medicamento, o que iria facilitar a identificação por parte do utilizador do medicamento em causa.

Já a nível de informações, e aproveitando a definição do perfil por parte do utilizador, quando este adicionasse um medicamento à sua lista de prescrições (ou até mesmo durante a procura deste), esse medicamento seria confrontado com as informações do perfil do utilizador, e caso não fosse aconselhada a sua toma, alertar o utilizador. De entre outras opções, recorrendo a mecanismos de pesquisa semântica, poder-se-ia informar o utilizador que o determinado medicamento que está a tomar pode ser a causa de ele estar a sentir um determinado sintoma.

Com o advento recente, a nível nacional, da prescrição eletrónica, poder-se-á também pensar num modelo alterado da aplicação, em que esta comunicará com essa prescrição e todos os medicamentos prescritos por um médico ao seu paciente, utilizador da aplicação, seriam automaticamente adicionados à sua lista de prescrições.

# Referências

- [1] INFARMED I. P. Apresentação. URL: [http://www.infarmed.pt/portal/page/portal/SOBRE\\_0\\_INFARMED/APRESENTACAO](http://www.infarmed.pt/portal/page/portal/SOBRE_0_INFARMED/APRESENTACAO) (acedido em 07/2012).
- [2] Gil Bouhnick. History of PDAs - Part #1. 10 de mar. de 2009. URL: <http://www.mobilespoon.net/2009/03/history-of-pdas-part-1.html> (acedido em 10/2012).
- [3] Zach Epstein. Android growth to end in 2012 as Microsoft begins to steal Google (...) 6 de jun. de 2012. URL: <http://bgr.com/2012/06/06/smartphone-market-share-2012-ios-windows-phone-idc/> (acedido em 08/2012).
- [4] Gartner Inc. Worldwide Sales of Mobile Phones Declined 2.3% in 2nd Quarter of 2012. Ago. de 2012. URL: <http://www.gartner.com/it/page.jsp?id=2120015> (acedido em 09/2012).
- [5] Rich Edmonds. Windows Phone sporting a 115% growth increase compared to 2011 (...) 8 de ago. de 2012. URL: <http://www.wpcentral.com/windows-phone-sporting-115-growth-increase-according-idc-data> (acedido em 08/2012).
- [6] Tony Lukasavage. A Deeper Look at Appcelerator and PhoneGap. 20 de abr. de 2011. URL: <http://savagelook.com/blog/portfolio/a-deeper-look-at-appcelerator-and-phonegap> (acedido em 02/2012).
- [7] PhoneGap. URL: <http://phonegap.com> (acedido em 12/2011).
- [8] Kevin Whinnery. Comparing Titanium and PhoneGap. 12 de mai. de 2012. URL: <http://developer.appcelerator.com/blog/2012/05/comparing-titanium-and-phonegap.html> (acedido em 05/2012).
- [9] Code for Concinnity. Sencha Touch vs jQuery Mobile - a first look. 28 de out. de 2011. URL: <http://cfc.kizzx2.com/index.php/sencha-touch-vs-jquery-mobile-a-first-look/> (acedido em 02/2012).
- [10] jQuery Mobile. URL: <http://jquerymobile.com> (acedido em 12/2011).
- [11] The Roveb Blog. Our Experience With jQuery Mobile and Sencha Touch. 8 de fev. de 2012. URL: <http://blog.roveb.com/post/17259708005/our-experience-with-jquery-mobile-and-sencha-touch> (acedido em 02/2012).
- [12] Sencha Touch. URL: <http://www.sencha.com/products/touch> (acedido em 12/2011).
- [13] Appcelerator Titanium. URL: <http://www.appcelerator.com/platform> (acedido em 12/2011).
- [14] Tony Lukasavage. PhoneGap is Web-based, Appcelerator is Pure Javascript. 25 de mai. de 2011. URL: <http://savagelook.com/blog/portfolio/phonegap-is-web-based-appcelerator-is-pure-javascript> (acedido em 02/2012).

- [15] iPharmacy Named the Best iPhone App for Prescription Reference. 6 de jan. de 2011.  
URL: <http://www.pr.com/press-release/288415> (acedido em 04/2012).
- [16] jQuery. URL: <http://jquery.com> (acedido em 12/2011).
- [17] J. Nielsen. «Heuristic evaluation». Em: Usability Inspection Methods. New York, NY, USA: John Wiley & Sons, 1994.