



UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA

Mestrado em Engenharia Informática

**Mobilidade no Contexto Hospitalar: Registo online e offline
de intervenções de enfermagem CIPE no local de prestação
de cuidados**

Filipe Sezões

Orientador

Prof. Vitor Manuel Beires Pinto Nogueira

Évora, 13 de Outubro de 2012

Mestrado em Engenharia Informática

**Mobilidade no Contexto Hospitalar: Registo online e offline
de intervenções de enfermagem CIPE no local de prestação
de cuidados**

Filipe Sezões

Orientador

Prof. Vitor Manuel Beires Pinto Nogueira

Sumário

Os registos de enfermagem são uma das fontes de informação mais valiosas relativamente a uma boa prestação de Cuidados de Saúde, pois permitem fazer uma avaliação mais correcta do estado de saúde do doente assim como da sua evolução. Esta dissertação discute a temática dos registos de enfermagem associados a tecnologias móveis e como estas podem favorecer esses mesmos registos.

Para tal procedeu-se ao desenvolvimento de uma aplicação móvel para o sistema Android, o Hcis Mobile, que permite efectuar registos de Intervenções de Enfermagem CIPE no local de prestação dos Cuidados de Saúde e que se pretende integrar com o HP-HCIS, um sistema de informação integrado para o sector da saúde. Um dos objectivos desta aplicação é que possibilite o registo das Intervenções tanto num contexto *online* como *offline*, pelo que se procedeu à implementação de uma Base de Dados no dispositivo móvel que guarde a informação relativa aos registos e que posteriormente possa ser enviada.

*Mobility in hospitalar context: online and offline
registration of ICNP nursing interventions at the care
providing place*

Abstract

The nursing records are one of the most valuable information sources for a good Health Care provision, as they allow a better evaluation of the patient health status as well as its evolution. This dissertation discusses the nursing records topic associated with mobile technologies and how these technologies can assist those records.

In this sense a mobile application for Android, the Hcis Mobile, was developed, allowing the ICNP Nursing Interventions record to be made in the same place where the healthcare is provided, and which is intended to integrate with the HP-HCIS, an integrated information system for the health sector. An objective of this application is for it to allow both online and offline record of the interventions, so it was implemented a database in the mobile device to store the records information, which can be sent later.

Aos meus pais

Ao meu irmão

Aos meus avós

Aos meus amigos

Agradecimentos

Gostaria de agradecer, em especial, aos meus pais e ao meu irmão que sempre me apoiaram.

Gostaria igualmente de agradecer ao Prof. Vitor Nogueira por todo o acompanhamento e disponibilidade prestadas durante o desenvolvimento desta dissertação.

Quero também agradecer ao João Rodrigues que esteve sempre disponível durante o desenvolvimento da aplicação e sem o qual nada disto teria sido possível.

Por fim quero agradecer a todos os meus amigos que me ajudaram nesta fase.

Conteúdo

Sumário	i
Abstract	iii
Lista de Conteúdo	xi
Lista de Figuras	xiv
Lista de Tabelas	xv
1 Introdução	1
1.1 Enquadramento Geral: Os Registos em Enfermagem	1
1.2 Objectivos do Trabalho	3
1.3 Estrutura da Dissertação	4
2 Estado da Arte	7
2.1 Mobilidade no Contexto da Saúde	7
2.2 Sistemas e Aplicações Móveis	10
2.2.1 Serviços e Aplicações Destinadas à Gestão de Registos Clínicos Electrónicos	10
2.2.2 Serviços e Aplicações Destinadas ao Utilizador	19
2.3 Sistemas Operativos Móveis	22
2.3.1 Windows Phone	24
2.3.2 bada	25
2.3.3 BlackBerry OS	26
2.3.4 Symbian	26
2.3.5 iOS	27

2.3.6	Android	28
3	Enfermagem e a Linguagem CIPE	33
3.1	Evolução Histórica dos Cuidados de Enfermagem	33
3.2	Linguagem CIPE	37
4	HP-HCIS	43
4.1	Modelo Funcional	44
4.2	Módulos HP-HCIS	46
4.3	Segurança	47
5	Metodologia de Desenvolvimento	49
5.1	Metodologia Scrum	51
5.1.1	Equipas Scrum	52
5.1.2	Time-Boxes	53
5.1.3	Artefactos	54
5.2	Metodologia <i>Scrum</i> no desenvolvimento do Hcis Mobile	55
5.2.1	Fases do Projecto	56
6	Hcis Mobile	59
6.1	Comunicação Web Service	60
6.1.1	Web Services: O que são?	60
6.1.2	Web Service HP-HCIS Enfermagem	65
6.2	Armazenamento da Informação	70
6.2.1	SQLite	72
6.2.2	Base de Conhecimento	73
6.2.3	Plano de Cuidados	77
6.2.4	Plano de Trabalho	78
6.3	Funcionalidades	80
6.3.1	Login/Autenticação	80
6.3.2	Carregamento e Listagem dos Planos de Cuidados	81
6.3.3	Visualização do Plano de Trabalho	82
6.3.4	Confirmação de Acções de Enfermagem	83
6.3.5	Envio dos Planos de Cuidados	84
6.3.6	Circuito Funcional	85
7	Implementação	87

7.1	Ambiente de Desenvolvimento e Testes	87
7.2	Estrutura do Código e Descrição das Classes	88
7.2.1	<i>Package</i> hp.hcis.nursery.bd	88
7.2.2	<i>Package</i> hp.hcis.nursery.ws	89
7.2.3	<i>Package</i> hp.hcis.nursery.pojo	89
7.2.4	<i>Package</i> hp.hcis.nursery.ui	90
7.3	Interface Gráfica	96
7.3.1	Cores	97
7.3.2	Botões	97
7.3.3	Menus	100
7.3.4	Menus de Opções	106
7.4	Internacionalização	108
7.4.1	Idioma	108
7.4.2	Estrutura Interna de Dados	109
7.5	Testes	109
7.5.1	Testes de <i>Stress</i>	110
7.5.2	Testes Funcionais	110
7.5.3	Testes em Dispositivos Físicos	112
8	Conclusões e Trabalho Futuro	115
8.1	Trabalho Futuro	116
	Referências bibliográficas	124
A	Interface Web Service	127
B	Schema da Base de Dados	145

Lista de Figuras

2.1	Aplicação ALERT [®] para iPad	13
2.2	Exemplos da Interface Gráfica	16
2.3	Arquitectura do Validus inTouch[Validus Medical Systems, 2009]	17
2.4	Exemplos da Interface do DIAMEDIC [Diamedic©, 2011]	20
2.5	Look4MyHealth - Esquema de funcionamento	22
2.6	Arquitectura do sistema SMS-Saúde [Jesus, 2011]	23
2.7	Arquitectura do iOS	27
2.8	Arquitectura do Android	28
3.1	Simbolo da Enfermagem [Lyavelino, 2007]	36
3.2	Evolução do modelo da CIPE Beta 2 para a CIPE 1.0 [Garcia and Nóbrega, 2009]	40
6.1	Camada dos protocolos Web Service	61
6.2	Relação entre Diagnósticos, Intervenções e Acções	73
6.3	Características padrão das Acções	74
6.4	Tipos de Objectos Clínicos	75
6.5	Transacções e Observações Clínicas	75
6.6	Tabelas <i>hospitals</i> e <i>gender</i>	76
6.7	Alguns componentes do Plano de Cuidados	77
6.8	Alguns componentes do Plano de Cuidados	78
6.9	Secção de Confirmação de Acções pertencentes a Diagnósticos	79
6.10	Secção de Confirmação de Acções pertencentes a Actividades	79
6.11	Menu de Login	81

6.12	Autenticação	81
6.13	Menu HCIS	82
6.14	Menu de Listagem dos Planos de Cuidados	82
6.15	Menu Plano de Trabalho	83
6.16	Circuito de Estados de uma Acção	84
6.17	Menu Confirmação de Acções	84
6.18	Menu de Envio	85
6.19	Circuito Funcional	86
7.1	Botão Login	100
7.2	Estilo dos Botões do Menu HCIS	100
7.3	Menu de Login	101
7.4	Evolução da interface do Menu HCIS	102
7.5	Evolução da interface do Menu Listagem dos Planos de Cuidados	102
7.6	Evolução da interface do Menu Plano de Trabalho	104
7.7	Diferentes visualizações do Menu Confirmação de Acções	105
7.8	Menu Observações Clínicas	106
7.9	Menu de Envio	107
7.10	Menu de Opções	108

Lista de Tabelas

6.1	Mensagens correspondentes a cada método do Web Service	67
6.2	Correspondência entre tabelas Hcis Mobile e HP-HCIS da Base de Conhecimento	76
6.3	Correspondência entre tabelas Hcis Mobile e HP-HCIS do Plano de Cuidados	78
6.4	Correspondência entre tabelas Hcis Mobile e HP-HCIS do Plano de Trabalho	79
7.1	Componentes de cada <i>estilo</i> definido para os botões	100
7.2	Correspondência entre os Menus Hcis Mobile e os Menus de Opções	107

Capítulo 1

Introdução

1.1 Enquadramento Geral: Os Registos em Enfermagem

O registo dos Cuidados de Saúde tem sido uma temática de discussão desde o início da Enfermagem como profissão. Já em 1885, Florence Nightingale¹, durante a Guerra da Crimeia², onde foi prestar auxílio aos soldados britânicos, deparou-se com um cenário de grande deficiência relativamente aos registos no Hospital de Scutary. Na sua avaliação aos registos existentes constatou que não se faziam estatísticas precisas sobre quase nenhum dos aspectos relativos aos tratamentos, ou evolução do estado de saúde dos doentes, verificando ainda a não existência de qualquer tipo de estatísticas, nomeadamente qual a faixa etária onde ocorre o maior número de óbitos. Numa carta escrita a Sidney Herbert, Secretário de Estado para a Guerra do Reino Unido, chegou a escrever: *"Geralmente os nossos registos são tão pobres que, muitas vezes, o único registo existente é: morreu um homem em determinado dia"* [Costa, 2009]. Nightingale pretendia que as enfermeiras registassem toda informação relativa aos Cuidados prestados, como relatos da evolução do estado de saúde, observações essenciais, descrições claras, pormenores indispensáveis e estatísticas elucidativas, pois só assim era possível fazer uma avaliação assertiva da evolução do doente e optar pelo tratamento correcto.

A conquista da inclusão dos registo de Enfermagem como parte essencial na prestação de Cuidados de Saúde foi um marco importante que favorece o bem estar do doente e a própria qualidade dos Cuidados prestados. Com isto, os registos requerem uma atenção especial por parte dos enfermeiros, pois é necessário obedecer a determinadas regras para

¹Considerada a Mãe da Enfermagem moderna, ver Capítulo 3.

²Guerra que decorreu entre 1853 e 1856 que pôs frente a frente o Império Russo contra uma coligação formada pelo Reino Unido, França e Turquia.

que os mesmo sejam válidos. É a informação existente nos registos de Enfermagem que perdura no tempo, possibilitando análise futuras. Segundo [Costa, 2009] a importância dos registos não se deve a um aspecto específico mas ao conjunto de vários. Para o autor a importância dos registos de enfermagem pode ser dividida em quatro aspectos:

- **Qualidade dos Cuidados Prestados:** Através dos registos é possível ter acesso a todas as intervenções que o doente já esteve sujeito, independentemente do local onde ocorreram. Com esta informação, é possível efectuar diagnósticos médicos e de enfermagem mais assertivos e fundamentadas, pois os registos indicam qual a resposta do doente aos tratamentos prestados e auxiliam na alteração de prescrições. Através dos registos é possível individualizar os Cuidados, pois os mesmos ajudam a compreender a conduta do utente e retratam a evolução física e mental ao longo do tempo, assegurando a continuidade dos Cuidados e tornando claro para toda a equipa quais as necessidades específicas do doente.
- **Organização dos Cuidados Prestados:** Uma boa qualidade dos registos permite aos gestores avaliar as necessidades da unidade relativamente ao número de enfermeiros existentes. São também fundamentais para um bom trabalho em equipa, servindo inclusive para estimular esse tipo de trabalho. Através dos registos é possível aos enfermeiros de turnos seguintes terem conhecimento do que ocorreu com determinado doente, podendo assim prestar-lhe os Cuidados adequados. É fundamental que exista confiança entre todos os elementos da equipa, sendo os registos uma forma de reforçar essa mesma confiança, melhorando a comunicação entre todos os enfermeiros que prestam Cuidados ao mesmo doente.
- **Investigação Científica em Enfermagem:** A autonomia dos enfermeiros é um factor importante para uma prestação de Cuidados de qualidade. Um bom enfermeiro não deve apenas agir por intuição ou por ordem médica. Os registos de enfermagem permitem uma maior autonomia aos enfermeiros, pois constituem uma poderosa fonte de informação que serve como fundamentação para uma determinada opção que seja necessária tomar. Constituem também uma inigualável fonte de informação que pode ser utilizada em trabalhos de pesquisa, e quanto mais específica for a informação anotada, mais útil será no desenvolvimento de novos métodos de trabalho. Para além disso fornecem uma quantidade inestimável de dados estatísticos que posteriormente irão ser utilizados por outros Técnicos de Saúde.
- **Campo Jurídico:** Devido à cada vez maior exigência dos seus direitos por parte da população, a utilização de registos permite protecção legal por parte do enfermeiro, protegendo-o contra autos. Através da consulta aos registos efectuados pode-se garantir se o Plano de Cuidados foi correctamente seguido ou se houve negligência por parte do responsável. A ausência de registos demonstra, por si só, incúria, negligência ou encobrimento relativo às prestações de Cuidados. A utilização de registos num processo legal é permitida, constituindo um documento legal.

Existem, portanto, algumas regras e boas práticas para os registos serem considerados válidos e de qualidade. Os mesmos têm que seguir a política hospitalar, não podem estar rasurados, apenas em caso de erro e devem indicar isso mesmo, não devem existir linhas em branco nem possuírem entrelinhas. Devem apenas conter os dados de forma precisa, concisa e clara e nunca julgamentos ou conclusões não fundamentadas, como também não devem conter dados generalizados, como, por exemplo, "*sem queixas*". Devem ser feitos periodicamente e ordenados por ordem cronológica, registando sempre a data e hora, assim como possuírem as unidades de medida, caso existam, para evitar determinados erros. É também importante nunca escrever antecipadamente nem deixar registos importantes para o final do turno, pois podem ficar esquecidos.

Actualmente, na maioria dos casos, os registos, na hora em que são prestados, são efectuados em papel, obrigando o enfermeiro a levar consigo constantemente toda a informação relativa aos doentes em questão. Se por alguma razão se esquecer de um Plano de Trabalho no seu posto fixo é necessário voltar atrás, o que implica a perda de tempo que podia ser utilizada para prestar Cuidados de Saúde essenciais. Para além disso, é prática frequente, só se inserir no sistema informático a informação relativa aos registos no fim do turno, o que pode levar que o enfermeiro introduza dados incorrectamente, pois pode não conseguir perceber os registos que tirou enquanto efectuou a prestação de Cuidados devido à pressão que pode existir durante esse período, como se pode esquecer completamente de introduzir os dados. Para além da possibilidade de erro ao introduzir os dados, ao fazê-lo só no fim do turno leva a que outros intervenientes nas prestações de Cuidados, sejam eles médicos ou outros enfermeiros, só tardiamente vão ter acesso a esses mesmos registos. Outro aspecto que deve ser tido em atenção está relacionado com o facto de, ao registar os dados numa folha, a mesma pode-se perder, ficando assim, o enfermeiro, sem qualquer registo das prestações efectuadas.

1.2 Objectivos do Trabalho

Este projecto, desenvolvido em parceria com o Observatório HP na Universidade de Évora, pretende fornecer uma alternativa moderna, e diferente das utilizadas até agora, aos enfermeiros para o registo de intervenções de Enfermagem.

Para tal, foi desenvolvida uma aplicação móvel, para o sistema operativo Android, que permite aos enfermeiros registar todas as intervenções que prestem durante a execução do seu trabalho clínico. Esta aplicação tem o objectivo de interagir com o HP-HCIS, uma solução integrada de gestão de todas as actividades existentes numa Organização de Saúde, seja ela uma pequena clínica ou um grande centro hospitalar³, podendo efectuar o registo de intervenções tanto num contexto *online* como *offline*. É essencial que a aplicação funcione num contexto *offline*, pois poderão ocorrer situações em que não será possível ao dispositivo efectuar uma comunicação com o servidor onde irão ficar armazenados todos os registos. Sendo os registos uma fonte de informação essencial relativamente à evolução

³Mais detalhes sobre esta solução estão disponíveis no Capítulo 4.

do estado de saúde do doente, os mesmo não podem ser perdidos por dificuldades de comunicação. Para a realização deste projecto foi necessário efectuar a implementação de um *Web Service* que permitisse fazer a comunicação entre a aplicação desenvolvida e o HP-HCIS. Com esta alternativa ao modo de registar as intervenções pretende-se que o registo das mesmas seja melhorado e efectuado no momento exacto em que se realizem, não correndo o risco de ficarem esquecidas ou introduzidas com informação incorrecta. Outra factor importante relaciona-se com a disponibilidade da informação. Ao se registar as informações no momento em que ocorrem permite-se que todos os outros responsáveis clínicos tenham acesso à mesma imediatamente, factor que pode influenciar no melhoramento da prestação de Cuidados que o doente necessita.

Como existe sempre uma grande resistência à mudança, principalmente num meio onde a entrada da tecnologia nem sempre é bem vista pelos seus membros, tentamos implementar um *look and feel* e *workflow* bastante semelhantes aos existentes no HP-HCIS. Com isto, pretendemos reduzir o tempo necessário de habituação à nova aplicação.

1.3 Estrutura da Dissertação

Esta dissertação está dividida em 7 capítulos.

O Capítulo **1 (Introdução)** pretende contextualizar o leitor para o tópico relativo aos registos na área da Enfermagem, através de uma explicação concisa de qual a sua utilidade e como podem ajudar e influenciar na prestação de Cuidados de Saúde. Enumeramos algumas regras e *bons hábitos* que se devem ter para efectuar registos de qualidade e válidos. Neste Capítulo estão, igualmente, enunciados os objectivos que pretendemos atingir com este trabalho assim como a estrutura desta dissertação.

O segundo Capítulo, **2 (Estado da Arte)**, começa por abordar a temática da mobilidade no contexto da saúde, enumerando os principais conceitos e termos utilizados. Posteriormente é feita uma descrição do estado da arte em aplicações móveis existentes na área da saúde e os diversos objectivos que pretendem atingir. Por último é feita uma breve descrição do estado actual na área de dispositivos e sistemas operativos móveis.

No terceiro Capítulo, **3 (Enfermagem e a Linguagem CIPE)**, pretende-se sensibilizar o leitor para os aspectos evolutivos da história da Enfermagem, assim como fazer uma breve descrição do que é a Linguagem CIPE, referindo o objectivo para a qual foi criada, assim como características importantes da mesma.

O quarto Capítulo, **4 (HP-HCIS)**, fornece informação sobre o HP-HCIS, uma solução integrada direccionada para o contexto das Organizações de Saúde. Neste capítulo encontram-se informações sobre o modelo funcional do mesmo, quais os módulos que o compõem assim como algumas soluções existentes relacionadas com a segurança.

A metodologia adoptada para o desenvolvimento deste projecto foi uma metodologia *Agile*, mais especificamente a metodologia *Scrum*. No Capítulo **5 (Metodologia de Desen-**

volvimento) estão enunciadas as principais regras que compõem este método para desenvolvimento de *software*. Como na realidade não foram seguidas todas as *boas práticas* do *Scrum*, procedemos à explicação do que não foi seguido e porquê. Por fim, é feita uma listagem de todas as etapas em que se dividiu o desenvolvimento da aplicação.

No Capítulo **6 (Hcis Mobile)** são referidos todos os aspectos referentes ao desenvolvimento da aplicação. Irão ser explicados os conceitos do *Web Service* implementado e a estrutura da *Base de Dados* utilizada. Por fim são explicadas as funcionalidades existentes na aplicação e o *workflow* que a mesma permite.

Os detalhes sobre a implementação da aplicação estão descritos no Capítulo **7 (Implementação)**. Neste Capítulo é explicado como a aplicação foi implementada, descrevendo a estrutura de classes e ficheiros da mesma assim como os testes de desempenho efectuados.

Por último, no Capítulo **8 (Conclusões e Trabalho Futuro)** são apresentadas as principais conclusões desta dissertação assim como o trabalho futuro a realizar.

Capítulo 2

Estado da Arte

2.1 Mobilidade no Contexto da Saúde

O termo utilizado, na área da saúde, para definir as práticas médicas que utilizam dispositivos e infraestruturas móveis é **mHealth** (*mobile healthcare*, em português Cuidados Médicos Móveis). Uma das definições, que pode ser encontrada em [Institute, 2011], é: *”The delivery of health care services via mobile communication devices such as cell phones.(...)”*. Esta definição, actualmente, deve ser alargada pois mHealth compreende todo o tipo de actividades que envolvam a troca de informação relacionada com cuidados médicos através de meios de comunicação móveis e novas tecnologias multimédia. Isto inclui os telemóveis, tablet’s, PDA’s e a maioria das infraestruturas wireless existentes.

Um conceito que facilmente se confunde com mHealth é o conceito de **eHealth**(*electronic healthcare*). eHealth está relacionado com a utilização de processos electrónicos e comunicações que fornecem suporte a cuidados de saúde, fazendo o armazenamento, gestão e avaliação dos dados. Enquanto que mHealth pode ser visto como o meio de acesso à informação respeitante aos cuidados de saúde de saúde em questão.

A área respeitante à mHealth tem sido alvo de um profundo investimento e desenvolvimento nos últimos anos. O que, inicialmente, tinha como principal público alvo os profissionais da área da Saúde, com o desenvolvimento tecnológico passou a ser ponto de interesse de um público mais vasto. Para além dos médicos e enfermeiros, também os próprios doentes, e a população em geral, começaram a olhar de maneira diferente para as potencialidades que estes novos mecanismos podem oferecer.

Um dos grandes factores, para esta evolução no mercado, está relacionado com facto de cada vez mais pessoas conseguirem ter acesso a dispositivos que possuem grandes poten-

cialidades, associando a isso o crescimento das ligações de dados móveis (*Wi-Fi*, *3G*, etc.) e da utilização crescente de SMS's. Com estas tecnologias torna-se mais fácil a comunicação, independentemente da localização e da hora. Para se verificar desenvolvimento na secção de mHealth, é essencial que tal desenvolvimento também se verifique na área da eHealth. Um exemplo relevante é a utilização de *Electronic Medical Records* (EMR's), que consistem em armazenar o histórico do doente (eHealth), e possibilitam o acesso aos dados do doente pelos diversos especialistas (mHealth).

Actualmente divide-se o mercado de mHealth em vários nichos [mobileStorm, 2011], dando-se uma maior relevância aos seguintes:

- **Monitorização Geral** A monitorização de características relativas à saúde engloba vários parâmetros, desde a medição do nível de glucose, até à pressão arterial, entre outros. Actualmente é necessário anotar todas essas medições e posteriormente levá-las ao médico para que este as possa analisar. Com a evolução das infraestruturas e da tecnologia móvel será possível esses dados serem adquiridos e enviados automaticamente para o médico, ou outro agente responsável, instantaneamente e assim prover uma mais rápida análise, diagnóstico e aplicação de medidas terapêuticas.
- **Sistemas Pessoais de Resposta de Emergência** Este conceito praticamente manteve-se inalterado desde a sua origem. A primeira evolução está associada ao facto de ser possível comunicar, através de redes sem fios, com uma rede existente. Como por exemplo comunicar com o 112 no caso de um alerta ser accionado. Com o envolvimento das novas tecnologias pretende-se que não seja dado, apenas, o alerta como também sejam enviadas outras informações. Informações essas que podem fornecer a localização e medições de funções básicas de vida.
- **TeleMedicina** TeleMedicina consiste na capacidade de enviar e receber informação médica de forma interactiva para que seja possível a realização de uma consulta, e em certos casos fornecer suporte a práticas operatórias ou exames. A TeleMedicina já existe muito antes do desenvolvimento das tecnologias móveis, mas com o surgimento destas o seu desempenho pôde ser melhorado. As câmaras existentes actualmente, nos dispositivos móveis, já possuem uma elevada qualidade de imagem, permitindo assim uma boa recolha de dados para serem analisados pelos especialistas que estão noutra local. Para além disso, com o surgimento das redes 4G a comunicação entre os intervenientes torna-se mais eficiente, potenciando uma melhor qualidade nas consultas à distância.
- **Aplicações de Saúde e Fitness** Com a evolução dos SmartPhones as aplicações móveis direccionadas para a saúde e fitness começaram a proliferar, o que tornou óbvio o crescimento do sector relacionado com mHealth. Essas aplicações permitem aos dispositivos funcionar como aparelhos de cuidados médicos ou servirem como meios de comunicação ou armazenamento de registos pessoais de saúde. As potencialidades destas aplicação são inúmeras e incalculáveis. Por exemplo, o iOS da Apple permite desenvolver aplicações que possibilitem a sincronização de aparelhos médicos

com o telemóvel, através de USB ou Bluetooth. Outro exemplo, é a categoria "Saúde e Fitness" no **Google Play**¹, o que demonstra as inúmeras aplicações já existentes dentro desta área para aquele sistema operativo móvel e a procura existente por parte dos utilizadores.

- **Segurança em Mensagens Móveis** A utilização de SMS foi uma das primeiras formas de mHealth. O facto de ser uma tecnologia à qual os utilizadores já estavam bastantes habituados facilitou a sua introdução e fixação. No entanto, a sua utilização no sector médico encontrou diversos entraves, relativamente a segurança, privacidade e conformidade HIPAA². Actualmente, diversas instituições utilizam o sistema de mensagens para realizar várias acções automaticamente, tais como marcação de consultas, envio de notificação e alterações de datas, entre outras. Pretende-se que no futuro, por exemplo, para além de ser possível avisar o doente ou médico que determinados exames ou análises já estão concluídas fornecer também um *link* seguro onde o interessado possa aceder à informação do exame no local onde está, não sendo necessário deslocar-se até ao local onde o exame foi realizado, ou esperar que este lhe seja enviado para uma determinada localização.
- **Registos Clínicos Electrónicos** Uma das aplicações mais directas das novas tecnologias na medicina está relacionada com a criação de registos clínicos electrónicos, **RCE**³, tendo como objectivo substituir os registos clínicos em papel. Um registo clínico contém todos os dados relativos ao paciente em questão, feitos pelo médico como por outros profissionais de saúde. Este registo contém os dados relativos à história clínica, evolução diária, diagnósticos apurados, tratamentos efectuados, evolução do estado de saúde, complementados com testes laboratoriais e relatórios de diagnósticos (radiografias, endoscopias, etc.). Nos registos em papel, os dados são introduzidos de forma manuscrita por diversas pessoas, e toda a informação clínica é anexada ao processo, muitas vezes de forma desordenada. Sempre que se pretende aceder à informação relativa ao doente é necessário ir buscar o processo "físico" e procurar naquele amontoado os dados que se desejam avaliar. Tudo isto torna o acesso à informação muito complexo. Tendo em conta as lacunas existentes nos registos em papel, torna-se necessário informatizar todo esse processo. Com essa informatização muitas outras potencialidades podem ser atingidas. As principais vantagens da utilização de registos clínicos electrónicos são: auxiliar na prestação de cuidados de saúde, auxiliar a decisão clínica, avaliar a qualidade dos cuidados prestados, fazer a gestão e planeamento dos recursos de saúde, auxiliar na investigação e auxiliar na educação médica.

Tendo em atenção que o pretendido, com este trabalho é a implementação de uma aplicação móvel, que permita a consulta e actualização de informação clínica durante a prestação

¹Loja de aplicações para Android.

²*Health Insurance Portability and Accountability Act* - Lei Norte-americana que, entre outros aspectos, estabelece as normas relativas à privacidade da informação de saúde, ajudando a prevenir o uso impróprio de registos médicos pessoais [of Health & Human Services, 2011].

³Em inglês, **EMR**: *Electronic Medical Records*.

de cuidados de enfermagem no local, podemos assumir que o sector de mHealth em que está englobado é o último descrito.

2.2 Sistemas e Aplicações Móveis

As tecnologias para a computação móvel tiveram o seu "boom" a partir dos anos 90, o que possibilitou o acesso à informação a partir de qualquer localização, inclusive em movimento, e no momento exacto que se deseja. Este factor permitiu o surgimento de inúmeros sistemas e serviços para os utilizadores.

Os dispositivos móveis estão em constante evolução, tornando-se cada vez mais populares, e em certos casos considerados indispensáveis no quotidiano da população. Para além de comunicarem entre si, esses dispositivos permitem, também, ligação à Internet e com as redes fixas tradicionais. Tudo isto é suportado pela cada vez maior cobertura por parte das infraestruturas, entre as quais as redes *Wireless* e as redes de telemóveis (GSM, 3G, 4G). Com estas novas formas de comunicar o desenvolvimento de sistemas móveis tornou-se inevitável.

Existem diversas aplicações desenvolvidas para dispositivos móveis relacionadas com a área da saúde. No entanto, a grande maioria delas são apenas mini-aplicações relacionadas com um aspecto específico dentro da área. Essas aplicações vão desde contador de calorias a manuais de medicamentos, passando por outras que ensinam a fazer Reanimação Cardi-respiratória, por exemplo. Estes programas podem ser encontradas nas diversas fontes de aplicações consoante o dispositivo móvel que se possua, podendo ser oficiais (**Google Play**, **Apple Store**, etc. . .) ou não-oficiais (**YAAM**⁴, por exemplo).

Em seguida iremos expor algumas aplicações existentes para dispositivos móveis existentes actualmente no mercado, separando-as em dois tipos. As que se destinam à gestão de registos clínicos (as mais relevantes para o nosso projecto) e aquelas que tem como principal objectivo ajudar o utilizador do dispositivo móvel, ou seja, destinadas ao utilizador.

2.2.1 Serviços e Aplicações Destinadas à Gestão de Registos Clínicos Electrónicos

Os RCE possuem toda a informação médica e clínica relativa aos pacientes, assim como os cuidados de enfermagem que lhe estão destinados. É indispensável, para que se consigam realizar de forma correcta e atempada os cuidados, que o enfermeiro, ou outro profissional, tenham acesso a esse documento quando estão junto do doente. Outro factor que deve ser tomado em atenção está relacionado com o tempo que se leva desde que se administraram os cuidados até à altura que estes são introduzidos no sistema. O que se verifica actualmente, na maioria dos casos, é que o prestador de cuidados necessita aceder ao sistema

⁴ Yet Another Android Market - <http://yaam.mobi/>.

informático antes de iniciar a sua ronda, tomar notas do que tem que realizar a cada doente, durante a prestação de cuidados ir anotando os dados de cada paciente e só no fim do turno é que introduz esses dados no registo electrónico. As aplicações destinadas à gestão de RCE pretende facilitar o acesso e gestão da informação, permitindo que o enfermeiro, ou outro prestador de cuidados, consiga aceder a partir de um dispositivo móvel ao registo do doente quando se encontrar junto a ele e actualize o registo no instante em que os cuidados são prestados.

Glantt - Soluções de mobilidade na área dos cuidados de saúde

A solução proposta pela **Glantt**⁵, pretende fornecer aos enfermeiros um sistema móvel com o qual seja possível aumentar a produtividade do seu trabalho. Deseja-se que as tarefas a realizar sejam efectuadas de forma mais célere, eficiente e segura. Estes factores levam a uma melhoria de processos, proporcionados por uma maior facilidade de comunicação e a não utilização de papel, possíveis devido ao uso da aplicação.

As principais funcionalidades, implementadas pela solução são [glantt, 2011]:

- Registo de consumos
- Localizador de pacientes por RFID⁶
- Preparação da administração
- Administração a pacientes
- Alertas para enfermeiros

De modo a complementar esta solução, pode-se integrar os módulos referentes a **Supply Chain** e **Warehouse Management** [glantt, 2011] adaptados à área hospitalar. O módulo *Supply Chain* controla o fluxo dos diversos materiais utilizados dentro da instituição, simplificando os processos e garantindo que o percurso de todos os materiais, entre os diversos sistemas, é devidamente seguido. Na área hospital, o módulo anteriormente referido, tem como principais objectivos o controlo unidose, a integração com prescrições médicas e a integração com robots. Por sua vez, o módulo *Warehouse Management* tem como principal objectivo o controlo dos stocks e garantir uma melhor gestão do espaço físico do armazém. Para tal, pretende implementar as seguintes funcionalidades:

- Integração Barcode Scanning e RFID
- Gestão de Lotes e números de série

⁵<http://www.glantt.com/>.

⁶**Radio-Frequency IDentification** - É um método de identificação automática que utiliza etiquetas RFID. Estas etiquetas permitem recuperar e armazenar dados remotamente através de sinais de rádio. [Wikipédia, 2011c].

- Suporte Multi-Armazém
- GeoReferenciação dos Armazéns
- Gestão de Localizações dos Artigos
- Impressão de etiquetas
- Gestão de entradas e saídas
- Transferências de stocks
- Writeoffs
- Gestão de níveis de stock/Pontos de Encomenda
- Inventariação

Projecto Hercules - Solução de Supply Chain e Patient Care

Este projecto foi encomendado, à Glintt, pelo **BES Saúde**, para ser utilizado no Hospital da Luz, em Lisboa. Tem como principal objectivo tornar o hospital *livre de papéis* centralizando-se nos PDA's como a principal ferramenta a ser utilizada. As funcionalidades que lhe estão incumbidas são:

- Gestão de stocks (supply chain completo) ao nível dos consumíveis e farmácia
- Controlo e administração aos pacientes (Patient Care)
- Recepção de artigos
- Picking para armazenamento
- Picking para Serviços
- Supervisão e autorização de recepção
- Quebras de stock
- Gestão de Localizações de artigos
- Inventariação
- Sistema de reposição automática

ALERT[®]

O sistema ALERT[®] é uma solução informática para locais de prestação de cuidados de saúde, que efectua o registo, a interligação, a reutilização e a análise de toda a informação clínica. É composto por vários módulos que vão desde a marcação de consultas até a ferramentas de *Business Intelligence* que fornecem uma perspectiva mais alargada sobre as instituições de saúde. Na área da mobilidade o ALERT[®] tem vindo a desenvolver, desde 2010, uma aplicação móvel para iPad, o **ALERT[®] para iPad** [alert, 2006], Fig.2.1. Para além desta aplicação, o ALERT[®] já utiliza tecnologias móveis que permitem diversos modos de disponibilizar e receber informação por parte dos profissionais de saúde. Entre esses serviços encontra-se a notificação de consultas, acções de workflow e alertas por SMS, Beep e PDA [ALERT[®], 2011a].

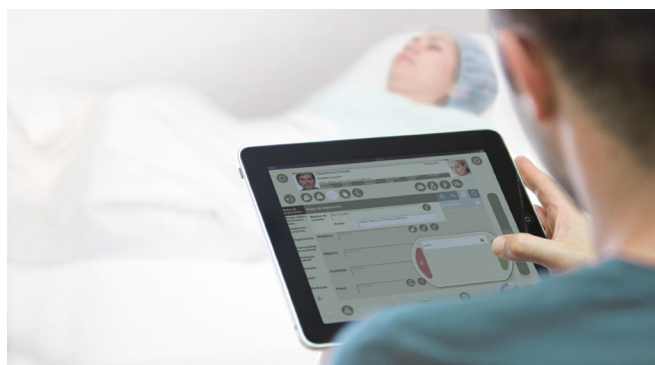


Figura 2.1: Aplicação ALERT[®] para iPad

O módulo do ALERT[®] que é relevante para o projecto é o relativo ao internamento hospitalar, o ALERT[®] INPATIENT. Este software clínico e de gestão das unidades de internamento proporciona as seguintes vantagens [ALERT[®], 2011b]:

- Visão global de todos os episódios, horas e alertas
- Acompanhamento de cada intervenção
- Análise das actividades de todos os profissionais de saúde
- Reconciliação medicamentosa mais fácil
- Evita a redundância nos procedimentos médicos e meios auxiliares de diagnóstico
- Redução do tempo de permanência, utilizando alertas para avisar os profissionais da existência de tarefas em atraso
- Gestão da alocação dos recursos técnicos e humanos
- Promoção da eliminação de toda a informação clínica registada em papel

- Promoção de uma análise e avaliação objectivas das operações do departamento através de ferramentas de análise e de tomada de decisão
- Adopção de conceitos inovadores
- Utilização da Certificação Internacional para a Prática de Enfermagem (CIPE)
- Indicadores para elaboração de gráficos
- Calculadora do número de horas de cuidados de enfermagem necessárias
- Associação automática de conteúdos
- Gestão de alertas

As vantagens, referidas anteriormente, são possíveis graças às diversas funcionalidades que o software permite [ALERT®, 2011b]:

- Resumo do episódio de urgência
- Nota de entrada
- História completa do paciente
- Hábitos alimentares
- Alocação e gestão de camas
- Posicionamentos
- Avaliações de risco
- Horas de cuidados de enfermagem estimadas
- Transferência de responsabilidade
- Transferência de serviço
- Lista de espera de internamento
- Criação de processos cirúrgicos e de internamento
- Medicina Física e de Reabilitação

ValidusTMinTouchTM

Este sistema, desenvolvido pela *Validus Medical Systemns, Inc*⁷, tem como principal objectivo facilitar o acesso à informação e dar instruções para o tratamento dos doentes (*CPOE*⁸), por parte do médico responsável [Validus Medical Systems, 2009]. O sistema está directamente integrado com todos os outros serviços existentes no hospital: serviços de admissão, farmácia, radiologia e laboratório. O uso deste sistema permite, ao responsável, conseguir sempre aceder à informação do doente independentemente da localização. Com isto pretende-se que o tratamento dos doentes seja a principal preocupação do médico, diminuindo o tempo que poderia demorar a fornecer essas ordens através dos métodos convencionais (escrever numa folha que posteriormente seria entregue à pessoa responsável pelo tratamento/exame).

Tendo em atenção que o principal objectivo é permitir a emissão de ordens clínicas, este sistema possui as seguintes características [Validus Medical Systems, 2009]:

- **Cobertura Clínica Completa** O sistema fornece mais de 150 formulários de ordens clínicas, incluindo medicações, testes de laboratório e de diagnóstico, cuidados respiratórios e intervenções intravenosas.
- **Smart Order EntryTM** O Validus inTouch foi desenhado tendo em consideração o esquema de trabalho dos médicos, conseguindo-se, através do Smart Order Entry, aumentar a velocidade do processo, pois permite guardar ordens introduzidas e recuperá-las imediatamente, através de frases contextualizadas. Essas frases estão sempre prontas a serem reutilizadas, aumentando assim o processo de introdução de ordens. Para tal é utilizado, como ponto de partida, um conjunto de ordens próprias do hospital. As ordens podem ser adaptadas a um médico específico e partilhadas com os outros grupos clínicos.
- **Reduzir Erros Utilizando Ordens Estruturadas** Todas as ordens estão estruturadas, possuindo campos que são de preenchimento completo. Com isto pretende-se que toda a informação indispensável seja introduzida. No caso de existir alguma falha, a ordem fica destacada e o médico é informado quais os campos que não foram preenchidos.
- **FirstDataBankTM** É possível a integração com a base de dados de medicamentos FirstDataBank⁹, o que vai auxiliar o médico na escolha certa da dosagem e informar quais os efeitos que podem surgir caso se prescrevam vários medicamentos diferentes. Pode-se também sincronizar o Validus inTouch com o *stock* do hospital, evitando a prescrição de medicamentos inexistentes.

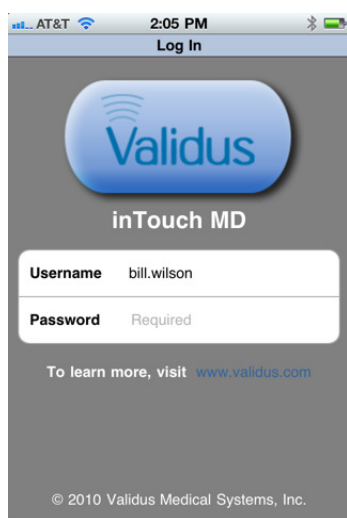
⁷<http://www.validus.com/>.

⁸ *Computerized Physician Order Entry* - processo electrónico que permite dar instruções médicas para o tratamento dos doentes. Essas ordens são comunicados para os funcionários responsáveis ou para os departamentos do hospital [Wikipédia, 2011a].

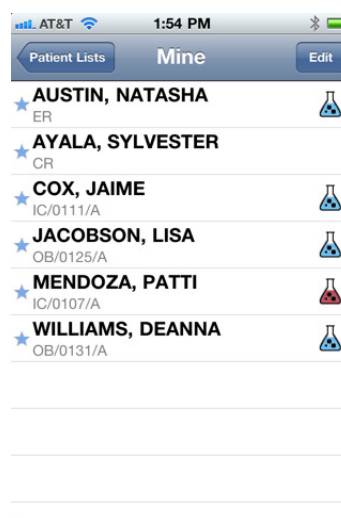
⁹ Base de dados que possui informação sobre medicamentos.

- **Reconhecimento de Voz** Outra das características presentes nesta aplicação está relacionado com o modo de introduzir os dados. Para além da possibilidade de *touch screen*, está também presente o reconhecimento de voz. Isto permite aumentar a rapidez com que se introduzem as ordens por parte dos médicos, pois basta dizer o que se pretende, por exemplo o nome de um medicamento, para que a informação seja introduzida.
- **Plataformas Suportadas e Mobilidade** A aplicação cliente deste sistema, o Validus inTouch MD, foi desenvolvida para ser usada em PDA's e *smartphones* que se possam conectar à rede, através de ligações *Wi-Fi* ou redes de dados celulares. É compatível com vários modelos de plataformas móveis, incluindo telemóveis *Windows Mobile* e *iPhones*. A interface foi desenvolvida com a preocupação de ser completamente eficaz em ecrãs reduzidos, como pode ser verificado na Fig. 2.2 [iTunes App Store, 2011].

A arquitectura cliente-servidor do sistema suporta, também, *web browsers*, o que permite a sua utilização em PCs *desktop*, portáteis ou tablets. O acesso ao sistema pode ser feito tanto dos postos de enfermeiros ou da sala do médico, como remotamente a partir de casa ou enquanto se viaja.



(a) Ecrã de Login



(b) Lista de Pacientes

Figura 2.2: Exemplos da Interface Gráfica

- **Arquitectura do Sistema** Para além de todas as componentes já faladas, o sistema possui ainda mais duas componentes, o Validus inTouch Print StationTM e o Validus inTouch ConsoleTM. Com o primeiro é possível as ordens serem impressas na estação correspondente à localização do doente em questão, enquanto que o segundo permite aos enfermeiros aceder a toda a informação importante relativa ao paciente. Um esquema completo de toda a arquitectura pode ser visto na Fig. 2.3.

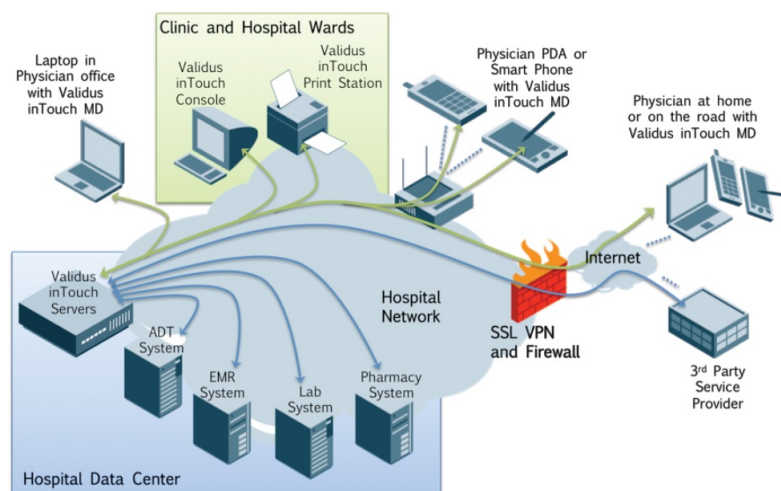


Figura 2.3: Arquitectura do Validus inTouch[Validus Medical Systems, 2009]

- **Segurança** Os dados clínicos são informação confidencial e privada que não podem ser visualizados por terceiros. O Validus inTouch apenas realiza comunicações utilizando o protocolo SSL, encriptando toda os dados, privados ou não, tanto seja através de redes por cabos como *wireless* ou redes de telemóveis. Outro factor está relacionado com a autenticação. Para ser possível a conexão com o sistema, e posterior consulta de informação ou introdução de ordens, é necessário o dispositivo possuir um cartão de memória SD (*Secure Digital*), pessoal a cada médico e que possui um número de série único, e que o médico introduza a password relativa a esse mesmo cartão. Para garantir que a informação não seja captada por terceiros, nenhuma informação é guardada no dispositivo, pois este pode ser roubado ou pode-se perder. Sempre que se pretende realizar alguma operação é necessário conectar-se ao servidor onde a informação está alojada e consultá-la.

eKlinik Healthcare Solutions

Este serviço¹⁰ não é propriamente um gestor de Registos Clínicos como o anterior, é antes um serviço *Web Based* que funciona como um gestor de soluções Hospitalares e Clínicas, Cuidados de Saúde, Laboratórios e Farmacêuticas.

Existem várias versões deste serviço, cada um destinado a um fim específico:

- **eKlinikCHIRO** Sistema de gestão de Quiroterapia¹¹
- **eKlinikDC** Sistema de gestão de Clínicas de Diálise
- **eKlinikDENTA** Sistema de gestão de Práticas Dentárias

¹⁰<http://www.eklinik.com/>.

¹¹Uma prática médica focada na relação entre a estrutura do corpo, normalmente a espinhal medula, e as suas funcionalidades.

- **eKlinikLAB** Sistema de gestão de Informação Laboratorial
- **eKlinikMC** Sistema de gestão de Informação de Centros Médicos
- **eKlinikMD** Sistema de gestão de Clínicas
- **eKlinikMMC** Sistema de gestão de Maternidades
- **eKlinikPHYSIO** Sistema de gestão de Fisioterapia
- **eKlinikSCREEN** Sistema de gestão de Dados do Paciente
- **eKlinikRPIS** Sistema de gestão de produtos farmacêuticos
- **eKlinikVET** Sistema de gestão de Práticas Veterinárias
- **eKlinikVIS** Sistema de gestão de Vendas

Este serviço pode ser utilizado em telemóveis, PDA's, tablet's, netbook's e PC's fixos desde que possuam ligação à Internet. Para tal é necessário adquirir uma licença e posteriormente aceder ao site <http://www.eklinik.com/> para que se possa aceder aos serviços adquiridos.

Relativamente à versão **eKlinikSCREEN**, que é aquela que se enquadra no âmbito do projecto, as funcionalidades permitidas são:

- **Registo de Pacientes**
- **Seleção Aleatória de Paciente**
- **Gestão de Marcações**
- **Gestão de Resultados de Exames**
- **Gestão de Filas**
- **Comunicação**
- **Sistema de Informação Executiva**
- **Sistema de Administração**

iPatientCare

O *iPatientCare*, da **MCS** - Medical Communication System Inc., é uma solução EHR (*Electronic Health Records*) e Gestão de Práticas integrada que possibilita tornar os cuidados primários mais acessíveis, contínuos, compreensivos, centrados na família, coordenados, compassivos e mais eficientes. Com isto a MCS pretende tornar o paciente o centro do sistema de cuidados de saúde [MCS, 2010].

Este sistema é uma solução baseada em *Cloud Computing* disponível como uma solução SaaS¹² hospedada remotamente e como uma solução Intranet hospedada localmente.

Fazem parte deste serviço três sistemas destinados a aplicações móveis: o *iPatientCare Charts*, o *iPatientCare eRx* e o *iPatientCare Charge Capture* [MCS, 2011]. Em seguida iremos fornecer uma breve explicação das duas primeiras, pois a última, apesar de ser uma aplicação móvel na área da saúde, está relacionada com questões de facturação e pagamentos, o que não é relevante para este projecto.

iPatientCare Charts Esta aplicação contém modelos bastante intuitivos que permitem a introdução de informações clínicas no local de prestação de cuidados. Pode ser utilizada em dispositivos Palm, PocketPc, Tablet PC e SmartPhones. Com a utilização desta ferramenta é garantido a legibilidade da informação, reduz os custos de transcrição e assegura a conformidade de facturamento. O *iPatientCare Charts* fornece um relatório completo dos registos clínicos do paciente tanto ao lado da cama, através de dispositivos moveis ligados via *Wireless*, como em computadores *desktop* através de um *Web Brower*. Os modelos fornecidos permitem o registo de notas de evolução, progresso, consultas, admissão e alta clínica. Para além desses registos fornece um registo clínico completo, onde figuram as notas clínicas, os relatórios de exames, as listas de problemas e medicação, o históricos de imunizações, as alergias e o histórico familiar e social. Os alertas clínicos pró-activos são mostrados com cor vermelha o que facilita a pesquisa de problemas por parte do médico.

iPatientCare eRx Esta aplicação facilita a introdução de ordens por parte dos médicos. Para isso utiliza alertas activos e passivos baseado em dados provenientes de informação clínica, o que permite a eliminação de erros de prescrição de medicamentos aumentando a segurança do paciente. A administração de medicação facilita a "identificação positiva de pacientes", com isto tenta-se garantir que o paciente certo, recebe a medicação correcta com a dosagem correcta, no momento exacto, da maneira apropriada. Esta aplicação funciona sobre os mesmo dispositivos e da mesma forma que a *iPatientCare Charts*, fornecendo os mesmo registos clínicos.

2.2.2 Serviços e Aplicações Destinadas ao Utilizador

Nesta secção iremos expor alguns sistemas e aplicações que tem como principal objectivo auxiliar o utilizador, que tanto pode ser um paciente como um profissional, nas suas tarefas pessoais relacionadas com a saúde. Desde aplicações que permitem efectuar pesquisas sobre medicamentos a meios de comunicar com os centros hospitalares.

¹² *Software-as-a-Service*.

Epocrates Rx

Esta aplicação médica, gratuita, da *Epocrates, Inc.*, é uma das mais conhecidas a nível mundial da área da saúde. Está disponível para Palm, Windows Mobile, iPhone, iPod touch, BlackBerry e Android. Este software possui várias indicações referentes a medicamentos, entre as quais os seus efeitos, a dosagem, o preço de mercado, efeitos secundários, farmacologia e indicações de segurança [Epocrates®, 2011].

Para além desta aplicação, a Epocrates®, possui ainda outras aplicações móveis gratuitas (Epocrates MedTools, Epocrates CME, Epocrates MOBILE RESOURCE CENTERS) e pagas (Epocrates Rx Pro, Epocrates Essentials e Epocrates Essentials Deluxe). Estima-se que cerca de 45% dos médicos nos E.U.A. utilizem as aplicações Epocrates para o tratamento de doentes e aumentarem a produtividade [Epocrates®, 2011] na realização das suas actividades profissionais.

DIAMEDIC

Diamedic Diabetes LogBook é uma aplicação destinada ao controlo da diabetes. Através desta aplicação pode-se guardar as leituras de glucose, as injeções de insulina, medicamentos que se estão a tomar, entre outras funcionalidades. É possível enviar um e-mail directamente para o médico, com os dados previamente guardados, para que este possa avaliar o estado da doença. Outro aspecto interessante da aplicação é o facto de permitir, ao próprio utilizar, a consulta de gráficos, com os quais pode visualizar como a doença tem evoluído ao longo do tempo. A aplicação é altamente adaptável, permitindo utilizar os dados pré-definidos relativos aos medicamentos, por exemplo, ou alterá-los de acordo com as preferências do utilizador. Possui vários tipos de gráficos, pode-se configurar qual a unidade em que se deseja ver as leituras de glucose (mmol/L ou mg/dL) e possui também a capacidade de trocar dados com um servidor FTP para armazenar a informação. Exemplos da interface do DIAMEDIC podem ser vistos na Fig.2.4.



Figura 2.4: Exemplos da Interface do DIAMEDIC [Diamedic©, 2011]

Look4MyHealth

O Look4MyHealth é um sistema, da ISA IntelliCare¹³, que permite fazer Monitorização Remota de Sinais Vitais (**MRSV**), tanto em ambiente hospitalar como doméstico. A MRSVA permite uma detecção precoce de situações de potencial risco.

Meio Hospitalar Através de sensores, colocados no corpo, são recolhidos diversos sinais vitais (electrocardiograma, temperatura, movimento, entre outros) que, após serem processados localmente, são enviados, por GPRS, Wi-Fi ou rede com fios, para a unidade de saúde. Este envio é feito em tempo real ou sempre que as condições de alarme definidas forem atingidas. Os dados podem ser visualizados por um profissional de saúde, o próprio paciente ou um familiar através da Internet num computador ou PDA.

Meio Doméstico O sistema recolhe dados a partir de medidores de peso, monitores de pressão arterial e de outros aparelhos de monitorização de bio-sinais. Dependendo de cada utilizador, a informação é enviada periodicamente, um dado numero de vezes por dia ou semana, e enviada para um centro de dados. Se as condições de alarme forem atingidas, um profissional vai avaliar os dados e verificar se é necessária alguma intervenção. Os benefícios de utilizar este sistema são: uma permanente monitorização da saúde do utilizador, envio de alarmes (SMS ou e-mail) se algo anormal ocorrer, possibilidade de diagnósticos precoces, prevenir situações de risco, reduzir os custos com os cuidados de saúde, aumentar a autonomia e qualidade de vida do utilizador, fornecer mais dados ao utilizador e aos profissionais de saúde, reduzir o número de idas a centros de saúde, os profissionais estão prontos a agir em caso de situações urgentes e permite um fácil acesso aos dados através de um acesso à Internet. Além do já referido, o sistema garante confidencialidade e segurança dos dados e é fácil de usar. A nível técnico, este sistema pode realizar a comunicação através de uma rede sem fios, utilizando tecnologia Wi-Fi ou 3G, ou então através de uma rede Ethernet. O sistema permite a configuração de diversos dispositivos Bluetooth. O histórico médico dos pacientes e a configuração dos dados e dos alarmes é possível recorrendo a uma página Web [IntelliCare, 2011].

SMS-Saúde

A solução SMS-Saúde utiliza a tecnologia SMS para realizar a comunicação entre o Hospital e os seus utentes, o que faz com que a comunicação se efectue de modo mais rápido e eficiente. Este serviço foi desenvolvido numa parceria entre a TMN e a Movensis, tendo sido implementado, pela primeira vez, no Hospital Santa Maria (HSM), em Lisboa. Este projecto foi integrado com a inauguração do novo *Contact-Center* do HSM que utiliza serviços de Internet, Telefone e SMS para realizar a comunicação com os Utentes.

¹³<http://www.isa.pt/intellicare/>.



Figura 2.5: Look4MyHealth - Esquema de funcionamento

Funcionamento

- A marcação da consulta é realizada, via Internet, com o próprio Hospital pelo Centro de Saúde.
- O HSM alerta o utente, até três dias antes da data e hora da consulta, para estar presente no Serviço, enviando uma SMS.
- O utente responde, via SMS, confirmando ou não da sua presença.

Este sistema permite uma melhor gestão de toda a burocracia inerente à marcação de consultas, facilitando a comunicação entre o Utente e o Hospital. Com isto aumenta a rapidez com que se efectuem as marcações e permite reduzir o número de faltas de presença nas consultas por parte dos utentes. Esse número de faltas provoca constrangimentos tanto à própria unidade hospitalar como aos utentes que esperam por uma consulta.

As principais funcionalidades estão relacionadas com a capacidade de envio e recepção de SMS, definição de mensagens tipo e possibilidade de programação e calendarização de envio de SMS. A utilização destas mensagens vai além da simples marcação, notificação e confirmação de consultas. Podem servir também para notificação de falta dos médicos, marcação de exames por parte do utente, envio de informação genérica, reuniões internas, convocatórias para acções e eventos e, também, como meio para publicação e divulgação de normas [Jesus, 2011].

A arquitectura do sistema SMS-Saúde pode ser visualizada na Fig. 2.6:

2.3 Sistemas Operativos Móveis

A crescente utilização e desenvolvimento de aplicações do tipo das referias anteriormente está, em grande parte, relacionada com a evolução que se tem verificado na área dos

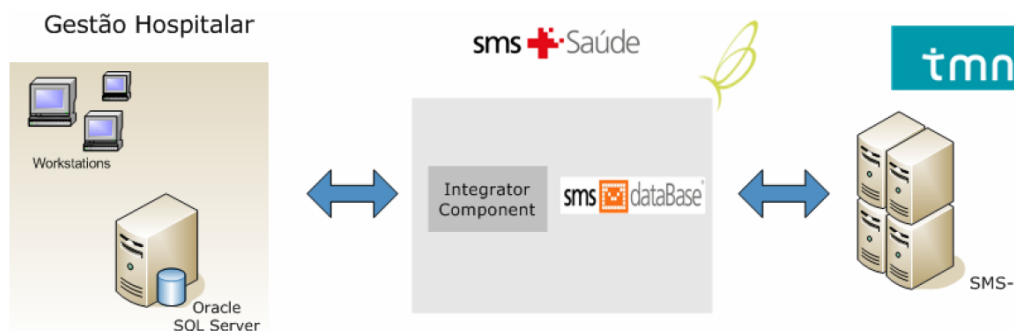


Figura 2.6: Arquitectura do sistema SMS-Saúde [Jesus, 2011]

dispositivos móveis. Se no início do seu surgimento apenas eram utilizados para efectuar e receber chamadas, actualmente é possível compará-los a um qualquer computador em termos de funcionalidades.

Através de um qualquer *tablet* ou *smartphone* é possível aceder à Internet, consultar e enviar *e-mails*, utilizar editores de texto, entre outras funcionalidades. Muitas das capacidades são possíveis graças à evolução existentes nos *Sistemas Operativos, S.O.*, destinados a plataformas móveis. Graças a todas as capacidades que estes dispositivos já possuem, aliadas às suas características móveis, as vendas de *smartphones*, segundo um estudo efectuado pela **Canalys** ultrapassaram as vendas de computadores pessoais, onde se incluem *netbooks*, *notebooks* e *desktops*. No ano de 2011 foram vendidos 487,7 milhões de *smartphones* contra 414,6 milhões de PC's [canalys, 2011].

Esta conquista nas vendas muito se deve ao facto dos S.O. móveis estarem cada vez mais completos e permitirem aos utilizadores experiências quase às idênticas às verificadas ao utilizarem PC's *comuns*. Existem diversos S.O. móveis, uns direccionados apenas para um certo tipo de aparelhos como é o caso do **iOS**, da **Apple**, que apenas é utilizado nos dispositivos da marca, iPod, iPhone e iPad, enquanto outros podem ser utilizados numa grande variedade de dispositivos, como é o caso do **Android**, que pertence à **Google**.

Na realidade, estes são os dois maiores S.O. destinados a dispositivos móveis da actualidade. Segundo o mesmo estudo da **Canalys**, o Android possui uma quota de mercado de 48,8%, que corresponde a 237,7 milhões de dispositivos vendidos com esse S.O., obtendo assim um crescimento de 244% relativamente às vendas verificadas no ano transacto. No segundo lugar desse estudo encontra-se o iOS dos quais foram vendidos 93,1 milhões de exemplares, o que constitui um crescimento de 96% relativamente a 2010 conseguindo uma quota de mercado de 19,1%. Estes dois S.O. juntos correspondem a uma fatia de 67,9 % de todas as vendas de *smartphones* registadas em 2011, ficando os outros 32% repartidos entre o **Symbian**, com 16,4% de quota de mercado, **BlackBerry OS** 10,5%, **bada** 2,7%, e **Windows Phone** 1,4%, sendo os restantes 1,1% distribuídos por outros S.O. não mencionados neste estudo.

Iremos então fazer uma breve descrição de cada um destes S.O. móveis, começando pelo que

possui uma menor quota de mercado, o Windows Phone, e terminando no Android. Iremos descrever mais pormenorizadamente o Android pois é aquele que domina actualmente o mercado e é também o alvo da aplicação desenvolvida no âmbito deste projecto.

2.3.1 Windows Phone

O Windows Phone é o sistema operativo desenvolvido pela **Microsoft** que veio substituir o *Windows Mobile*. Foi anunciado oficialmente em Fevereiro de 2010 em Barcelona durante o *Mobile World Congress 2010* e lançado oficialmente em Outubro do mesmo ano [Wikipédia, 2012d]. Apesar de ser o sucessor do Windows Mobile é totalmente incompatível com o mesmo, focando-se agora no mercado consumidor em detrimento do mercado empresarial. Actualmente está na versão 7.5¹⁴.

Um das alterações mais visíveis relativamente a todos os seus antecessores prende-se com a sua interface. O Windows Phone 7 possui uma nova interface gráfica, apelidada de *Metro*, a qual apresenta o ecrã principal dividido em **mosaicos dinâmicos** (*live tiles*, em inglês) que funcionam como atalhos para todas as funcionalidades, aplicações e recursos, que podem ser adicionados, removidos ou organizados segundo os gostos e necessidades do utilizador. Ao se carregar num desses mosaicos, são apresentados os **Hubs**, ou *Concentradores* em português. Este Hubs juntam, numa única vista, conteúdos relacionados com o objectivo de simplificar as tarefas mais usuais. Os Hubs padrão do Windows Phone são:

- **People Hub**: permite agregar os contactos de diferentes sítios como, por exemplo, Windows Live, Facebook e Gmail, incluindo os respectivos *feeds* de redes sociais.
- **Picture Hub**: facilita a partilha de imagens nas redes sociais num único passo, permitindo ainda reunir todas as imagens do utilizador através de uma integração com todos os álbuns existentes na Internet e no computador.
- **Games Hub**: integração com a *Xbox LIVE*.
- **Music + Video Hub**: permite ao utilizador aceder a todos os conteúdos multimédia: **Zune**, conteúdos do PC, serviços de música *online* e ao Rádio FM incorporado.
- **Marketplace Hub**: possibilidade de aceder a todas as aplicações e jogos instalados ou, inclusive, compra-los através do *Windows Marketplace*.
- **Office Hub**: reúne o *Outlook Mobile*, o *OneNote* e o *SharePoint Workspace* num só local, permitindo ao utilizador ler, editar e partilhar documentos.

Uma das principais características deste S.O. é a imposição de requisitos mínimos, por parte da Microsoft, aos fabricantes de dispositivos. Assim, para um dispositivo poder

¹⁴Este é o nome oficial, mas tecnicamente corresponde ao update 7.10.8107.79 [Microsoft, 2012].

utilizar o S.O. da Microsoft, tem que respeitar os seguintes requisitos [Lee and Chuvyrov, RESS]:

- **Visor 800x400 WVGA ou 480x320 HCGA:** Multitouch, com quatro ou mais pontos de toque.
- **5 botões:** Iniciar, Retroceder, Procurar, Câmara e Power.
- **CPU:** 1 GHz - ARMv7 Cortex/Scorpion ou superior.
- **GPU:** Aceleração com DirectX 9.
- **Memória:** 256 MB de RAM ou mais e memória interna de, pelo menos, 8 GB.
- **Wi-Fi**
- **GPS**
- **Acelerómetro**
- **Câmara de Alta Definição:** pelo menos 5MP, com flash.
- **Bússola Digital**
- **Sensor de Luminosidade**
- **Rádio FM**

O desenvolvimento de aplicações para o Windows Phone continua a ser efectuado nas linguagens já utilizadas pela Microsoft: **C#** e **VisualBasic** utilizando as bibliotecas base disponibilizadas pela *framework* **.NET**. As aplicações podem ser desenvolvidas através dos *softwares* **Silverlight**, utilizado normalmente para aplicações empresariais e jogos simples em *2D*, ou **XNA**, utilizado essencialmente para desenvolver jogos, permitindo criar jogos em *3D*.

2.3.2 bada

Este S.O. é desenvolvido pela *Samsung* e foi lançado oficialmente em 2010. Actualmente encontra-se na versão 2.0 [samsung, 2011] e tem como principal objectivo transformar os utilizadores de telemóveis Samsung em utilizadores de *smartphones*. Segundo [samsung, 2011] o bada nasceu da plataforma proprietária da Samsung que se pretendia tornar mais *inteligente*, para isso foram adicionadas melhorias, tais como o *multitouch* e gráficos *3D*. Foi também melhorada ao nível da sua interface e dando-lhe a capacidade de efectuar o *download* e instalação de aplicações.

A arquitectura deste S.O., segundo [Dorokhova et al., 2009], é constituída por 4 *layers* (camadas):

- **Kernel:** kernel Linux ou kernel **RTOS** (Real Time Operating System), dependente do hardware.
- **Device Layer:** fornece os serviços nucleares, tais como gestão do sistema e de segurança, gráficos, comunicação de dados e voz e gestão de áudio e vídeo.
- **Service Layer:** fornece serviços focados nas funcionalidades que são fornecidos pelos motores de aplicações e componentes *Web Services* interligando-os com o servidor bada.
- **Framework Layer:** apenas este *layer* pode exportar a API aberta C++. A *framework* consiste numa *framework* de aplicações e funções exportadas pelos *layers* adjacentes.

2.3.3 BlackBerry OS

Este S.O., de código fechado e proprietário, é desenvolvido pela Research In Motion (RIM) direccionado para os seus *smartphones*, os **BlackBerry**. A primeira versão foi lançada em 1999 para o Pager BlackBerry 580 [Wikipedia, 2012b], estando actualmente na versão 7.1 que se encontra nos modelos Bold 9900/9930, Curve 9350/9360/9370 e nos Torch 9850/9860 e 9810 [Limited, 2012].

A linguagem de programação utilizada para desenvolver aplicações para o BlackBerry OS é o Java, que podem ser desenvolvidas por programadores externos recorrendo à API disponibilizada. Possui ainda uma loja virtual onde se podem obter aplicações, a **BlackBerry App World**, que está disponível em todo o mundo deste 1 de Abril de 2009 [Dorokhova et al., 2009].

2.3.4 Symbian

Mantido actualmente pela **Accenture** este S.O., sucessor do **Symbian OS** e do **Nokia Series 60**, encontra-se na sua versão **Symbian Belle** lançada em 2011, que é um *update* da versão Symbian³. Ao contrário do seu antecessor Symbian OS, desenvolvido pela **Symbian Ltd.** que era formada, entre outras, pela Nokia, Sony Ericsson e Motorola, possui uma componente de interface baseada no *S60 5th Edition* [Wikipedia, 2012d].

Este sistema operativo foi idealizado de maneira a gerir os recursos escassos existentes nos dispositivos móveis, especialmente bateria e memória. Possui uma arquitectura de *microkernel* que contem apenas as funcionalidades mais básicas e mínimas de modo a maximizar a robustez e a disponibilidade para permitir responder em tempo real. Entre essas funcionalidades encontra-se o escalonamento, a gestão de memória e dos *drivers* do dispositivo, ficando a gestão de rede e de ficheiros noutras camadas da arquitectura.

É possível a instalação de aplicações efectuadas por terceiros, podendo estas ser desenvolvidas em várias linguagens de programação. A linguagem *standard* é o C++, utilizando

Qt, mas é possível programar aplicações em Symbian C++ (antigo *standard*), Python, Java ME, Flash Lite, Ruby e .NET, entre outras.

2.3.5 iOS

Anunciado em 9 de Janeiro de 2007 na **Macworld Conference & Expo** e lançado em Junho do mesmo ano, este S.O. encontra-se agora na sua versão 5.1 [Ribeiro, 2011]. O iOS apenas é utilizado nos dispositivos da Apple, nomeadamente no iPhone, iPad, e iPod Touch, não sendo permitido utilizar em hardware de terceiros. Uma das principais características deste S.O., derivado do Mac OSX, relaciona-se com a revolução que originou na maneira do utilizador interagir com o dispositivo. A interface do utilizador está baseada em conceitos de manipulação directa e multi-toque, permitindo ao utilizador interagir directamente com o ecrã do dispositivo através de vários gestos, como *tocar*, *deslizar* e *efeito pinça*¹⁵ [Wikipedia, 2012c].

A arquitectura do iOS é composta por quatro camadas, como pode ser visto na Fig. 2.7 [Ribeiro, 2011].

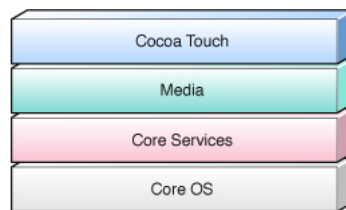


Figura 2.7: Arquitectura do iOS

A camada **Core OS** é a camada mais baixa do iOS e corresponde a uma versão reduzida do núcleo do Mac OS X. As principais funções desta camada são, entre outras, a gestão do funcionamento do sistema operativo, memória, bateria, luminosidade da tela e segurança.

Na camada **Core Services** é onde se localizam todos os serviços essenciais que são utilizados pelas aplicações, tais como gestão de chamadas (em dispositivos que tenham essa funcionalidade, como o iPhone), protocolos de comunicação e de base de dados, serviços de localização e gestão dos *Model-view-controller*, MVC, das aplicações.

A camada **Media** é responsável pelos gráficos, áudio, vídeo e tecnologias associadas tais como *OpenGL*, *Quartz* e *OpenAL*.

Por fim, a camada **Cocoa Touch** trata da parte visível para o utilizador. É responsável para apresentação da interface e por captar os *inputs* que podem ser através do toque como de sensores, como por exemplo o acelerómetro. É igualmente responsável pelo controlo das câmaras frontal e posterior.

¹⁵O *efeito pinça* consiste em juntar e separar dedos, que consoante o número de dedos utilizados é obtida uma resposta diferente por parte do dispositivo.

Inicialmente o desenvolvimento de aplicações nativas para o iOS era apenas efectuado pela *Apple* mas desde 2008, quando foi lançada a primeira versão do **iOS SDK**, é possível a terceiros desenvolverem aplicações. O SDK é composto por vários *softwares*, sendo eles o **XCode** como ambiente de desenvolvimento e que só corre em computadores *Mac*, **Interface Builder** para criar as interfaces, a ferramenta **Instruments** que permite analisar o consumo de memória e desempenho da aplicação e o **Simulator** que permite simular como a aplicação vai ser executada no dispositivo. A programação de aplicações para iOS pode ser efectuada em três linguagens diferentes: **C**, **C++** e **Objective C**, sendo esta última a mais recomendada pela *Apple*.

2.3.6 Android

A primeira versão comercial deste S.O. de código-aberto, versão 1.0, foi lançada em 22 de Outubro de 2008, sendo o *HTC Dream (G1)* o primeiro *smartphone* a utilizá-la [Wikipedia, 2012a]. Inicialmente começou a ser desenvolvido pela **Android Inc.** que foi adquirida, em 2005, pela *Google* que posteriormente se aliou a muitas outras companhias, onde se pode destacar a *Intel*, *HTC*, *LG*, *Motorola*, *Nvidia* e a *Samsung Electronics*, criando a **Open Handset Alliance** que tinha como objectivo desenvolver *standards* abertos para dispositivos móveis. O anúncio deste consórcio foi efectuado dia 5 de Novembro de 2007, no mesmo dia que foi revelado o seu primeiro produto, o *Android*.

Desde a primeira versão foram lançadas muitas outras para corrigirem *bugs* e adicionarem novas funcionalidades, estando actualmente na versão *4.0 Ice Cream Sandwich* anunciada a 19 de Outubro de 2011 [Wikipedia, 2012a].

Arquitectura

A arquitectura do *Android* está dividida em 5 camadas, Fig. 2.8 [Android, 2012]:

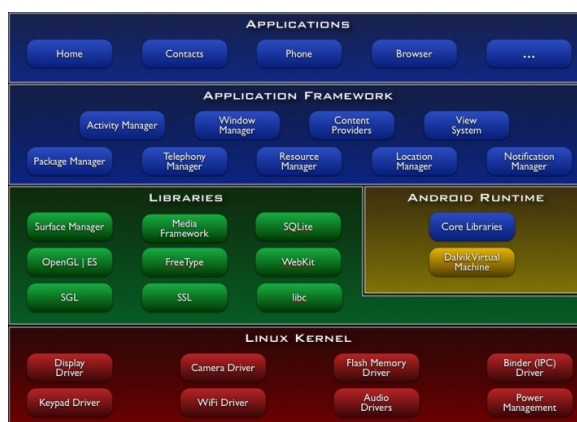


Figura 2.8: Arquitectura do *Android*

Na camada inferior encontra-se o **kernel** do *Android*. O *kernel* permite criar uma abs-

tracção entre o hardware e o restante *software*. Utiliza o Linux versão 2.6 para suportar os serviços essenciais do sistema, como segurança, gestão de memória, processos, rede e modelos de *drivers* [Android, 2012].

Na camada de **Runtime** estão incluídas as bibliotecas do núcleo e a máquina virtual Dalvik. As bibliotecas aí existentes fornecem a grande maioria das funcionalidades disponíveis na linguagem Java. A máquina virtual é utilizada para executar os arquivos em formato Dalvik executável (**.dex**) otimizando-os para baixo consumo de memória, sendo que cada aplicação Android é executada num processo individual na sua própria instância da máquina virtual Dalvik. Os arquivos **.dex** correspondem a classes Java compiladas transformadas pela ferramenta *dx* já incluída. Para as funcionalidades de gestão de memória, controlo de *threads* e outras de baixo nível a Dalvik invoca o *kernel* do Linux [Android, 2012].

A camada das **Libraries** contem um grupo de bibliotecas C/C++ que, para além de serem utilizadas por diversos componentes do sistema, estão completamente visíveis aos programadores. Entre outras, encontram-se nesta camada as bibliotecas que permitem suporte de áudio e vídeo, *Media Libraries*, e de manipulação da Base de Dados, *SQLite*¹⁶ [Android, 2012].

Os programadores possuem total acesso às API's utilizadas pelas aplicações do núcleo. A estrutura das aplicações é desenhada de modo a simplificar a reutilização de componentes. Na camada **Application Framework** encontra-se um conjunto de serviços e sistemas que auxiliam na criação e gestão das aplicações, entre os quais o *View System* que é um conjunto de *Views* (listas, grelhas, botões, etc.) que podem ser usados na criação de uma aplicação, *Content Providers* que possibilitam partilhar os seus próprios dados, *Resource Manager* que permite aceder a recursos como *strings*, imagens e ficheiros de *layout* e *Activity Manager* que gere o ciclo de vida das aplicações [Android, 2012].

Na última camada encontram-se as **Applications**. O Android possui um conjunto de aplicações por defeito onde se inclui um cliente de *email*, um programa de SMS, calendario, mapas e *browser*, todas escritas na linguagem Java [Android, 2012].

Desenvolvimento

Para se desenvolverem aplicações, o Android fornece um *kit* de desenvolvimento de *software*, o **Android SDK**. Este *kit* possui várias ferramentas essenciais e úteis para o desenvolvimento de aplicações: um *debugger*, bibliotecas, um emulador, documentação e tutoriais.

As aplicações para Android são escritas em Java, sendo função do Android SDK compilar o código juntamente com os dados e os ficheiros *resource* para um pacote Android que é um ficheiro único com a extensão **.apk**. Cada aplicação Android é composta por um

¹⁶O sistema de gestão de Base de Dados é abordado na Subsecção 6.2.1.

ou mais componentes (*Activities*, *Services*, *Content Providers* e *Broadcast Receivers*) que possuem funções distintas no comportamento da aplicação e que devem ser declarados no ficheiro de manifesto, o **AndroidManifest.xml**. Para além dos componentes já referidos, uma aplicação Android é ainda composta por elementos visuais, *layouts* e *widgets*, e outros ficheiros de recursos (*resources*), como as imagens utilizadas e os ficheiros necessários para a internacionalização.

Componentes

- **Activities:** são componentes que fornecem um ecrã com o qual o utilizador pode interagir.
- **Services:** estes componentes não fornecem qualquer interface visual. São utilizados para permitir à aplicação correr processos em *background*.
- **Content Providers:** permitem fornecer dados a outras aplicações, estejam eles alojados em ficheiros, na Base de Dados ou em qualquer outro lugar de armazenamento persistente.
- **Broadcast Receivers:** é um componente que permite responder a eventos lançados pelo sistema ou por outras aplicações.

Para se realizar a comunicação entre os componentes são utilizadas mensagens denominadas **Intents**. Os *Intents* são um conjunto de dados estruturados que são imprescindíveis para o componente que os recebe, pois indicam a acção que é necessária executar como os dados sobre os quais deve ser executada. Para além da informação destinada ao componente específico possuem ainda informação relevante para o Android, indicando qual a categoria do componente de destino do *Intent*, por exemplo.

Ficheiro de Manifesto

O ficheiro **AndroidManifest.xml** é utilizado para informar o Android quais são os componentes pertencentes à aplicação. Todos os componentes devem ser declarados neste ficheiro que fica localizado na directoria raiz do projecto. Ao não se declaram os componentes desta forma o sistema não vai ter conhecimento dos mesmos e eles nunca irão ser executados.

Para além de servir para se declarem os componentes da aplicação é igualmente utilizado para outras funcionalidades necessárias à correcta execução da aplicação. Entre essas funcionalidades destaca-se a identificação de permissões que a aplicação necessita (aceder à Internet ou aos contactos, entre outros), declarar o nível mínimo de API necessário para que a aplicação possa ser executada e declarar o *hardware* e *software* utilizado ou requerido pela aplicação (acesso à câmara ou serviços de *bluetooth*, por exemplo).

Interface do Utilizador (*User Interface*)

A interface de uma aplicação Android é construída a partir de objectos do tipo *View* e *ViewGroup* que descendem todos da classe *View*. Os objectos *View* serve de base para as subclasses denominadas *widgets* que correspondem a objectos com os quais o utilizador pode interagir (botões, caixas de texto, etc.). Por sua vez, os objectos *ViewGroup* correspondem aos *layouts*.

Layouts Os *layouts* permitem formar a interface do utilizador organizando as *views* da maneira desejada sendo essa organização efectuada através de um esquema de *árvore*, normalmente num formato XML. Cada elemento dessa árvore corresponde a um *View* ou a um *GroupView* onde os elementos *View* corresponde a folhas e os *GroupViews* a nós que contém outras *views*. Os *GroupView* definem o modo como os outros filhos *views* se estruturam. Existem várias formas de estruturas sendo as mais comuns as estruturas do tipo *LinearLayout*, *RelativeLayout*, *TableLayout* e *GridLayout*.

Widgets Os *widgets* são objectos do tipo *View* que possibilitam a interacção entre o utilizador e a aplicação. O Android já fornece um conjunto abastado de *widgets*, como botões, caixas de selecção e campos de entrada de texto, que permitem uma rápida construção da interface. Apesar de já existirem vários *widgets* por defeito é possível utilizar *widgets* diferentes, personalizando os existentes ou, inclusive, através da implementação de novos elementos.

Dialogs Os *Dialogs* são pequenas janelas que são exibidas por cima da *Activity* que está a ser executada ganhando o *foco* da aplicação. São utilizados, normalmente, para exibir notificações ou para a execução de pequenas tarefas que estejam relacionadas com a aplicação. Existem diferentes tipos de *Dialogs* já pré-definidos no Android, no entanto é possível criar um novo descrevendo o seu *layout*.

Menus Os menus permitem ao utilizador aceder a funcionalidades e configurações da aplicação. O tipo de menu mais usual é acedido a partir da tecla *Menu* existente nos dispositivos mas também é possível aceder a menus ao pressionar durante um período de tempo mais prolongado um item da interface, que correspondem aos *Menus Contextuais*. Estes menus possuem igualmente uma hierarquia de *views*, no entanto não cabe ao programador essa implementação pois a mesma fica a cargo do Android. É apenas necessário definir os métodos *onCreateOptionsMenu()* ou *onCreateContextMenu()* na implementação da *Activity* onde são indicados quais os itens que se pretendem introduzir no menu correspondente, sendo igualmente possível definir quais os itens desejados através de um ficheiro XML.

Recursos da Aplicação

Os ficheiro XML referido na subsecção anterior, a 2.3.6, nomeadamente os ficheiros respeitantes aos *layouts* e aos *menus* fazem parte dos ficheiro de recursos da aplicação. Para além desses ficheiro existem muitos outros tipos de ficheiros de recursos, onde destacamos os ficheiros de *strings*, *colors* e *drawables*.

Estes ficheiros ficam separados dos ficheiros Java que constituem o código fonte de cada aplicação, ficando situados na directoria *res/* organizados em sub-directorias consoante o seu tipo. Ao proceder desta forma torna-se possível prover a aplicação de recursos alternativos consoante as configurações do dispositivo, nomeadamente tamanhos de ecrã e diferentes idiomas.

Layout Como já foi referido, os *layouts* definem a estrutura da interface do utilizador de uma determinada *Activity*. Estes ficheiros estão disponíveis na directoria *res/layout/*.

Menu Situados na directoria *res/menu/*, são os ficheiros que contem a definição dos menus da aplicação indicando quais são os itens que compõem cada menu.

Drawable Os *drawables* são os recursos gráficos a que a aplicação tem acesso. Eles podem ser tanto imagens como ficheiros XML. Estes ficheiros encontram-se na directoria *res/drawable/* e alguns dos tipos existentes são:

- **Bitmap File:** corresponde a uma imagem do formato **.png**, **.jpg** ou **.gif**.
- **StateListDrawable:** é um ficheiro XML que define quais as imagens a apresentar consoante o estado do objecto, por exemplo se um botão está ou não pressionado.
- **ShapeDrawable:** define uma forma geométrica, incluindo a sua cor e gradiente.

String O ficheiro *strings.xml*, localizado na directoria *res/values/*, contem o mapeamento de todas as *strings* utilizadas na aplicação associando-as com um **Id** específico. Consoante a configuração do idioma do dispositivo é possível indicar o conjunto de *strings* correspondentes ao mesmo, bastando para isso colocar o ficheiro com as *strings* dos vários idiomas em pastas *values* personalizadas, possibilitando assim que uma aplicação esteja disponível em várias linguas. Por exemplo, ao se colocar as *strings* na pasta *values-es* é fornecida à aplicação que se o dispositivo estiver no idioma castelhano são essas as *strings* que se devem utilizar.

Capítulo 3

Enfermagem e a Linguagem CIPE

3.1 Evolução Histórica dos Cuidados de Enfermagem

A história dos cuidados de enfermagem é tão antiga quanto a humanidade. A condição de sobrevivência obrigou a que, desde sempre, as práticas de cuidados de saúde tivessem um papel importante nas civilizações. Nos primórdios da civilização as práticas de saúde eram, essencialmente, na grande maioria das comunidades, praticadas por instinto e estavam associadas ao trabalho feminino. Como era responsabilidade das mulheres cuidar dos seus filhos, assumiu-se que esses cuidados deviam ser estendidos para todos os membros doentes e necessitados da sociedade [Egenes, 2009].

Quando se teve consciência que quem dominava a capacidade de curar ganhava um maior relevo e respeito pelos outros membros das comunidades, os homens começaram a assumir esse papel para si. O misticismo e o desconhecimento das causas da doença, que muitas das vezes eram vistas como magia, resultado de pecado ou castigo por parte dos deuses, fez com que a influência e poder dos conhecedores dessa *arte* se fortalecesse perante os outros elementos da sociedade [On-Line, 2012]. Os trabalhos, por parte das mulheres, estavam, sobretudo, relacionados com os partos e a actuação, ainda hoje pouco decifrada, de mulheres de classe social alta que actuavam nos templos com os sacerdotes [de Enfermagem do Rio de Janeiro, 2012].

Nesta época, onde se abordava a mística entre os conhecimentos religiosos e as práticas de saúde primitivas utilizadas pelo sacerdócio nos templos, as práticas de saúde entravam na fase do empirismo [Andrade, 2008]. Este tipo de práticas, conhecidas como *práticas de saúde mágico-sacerdotais* dura até cerca do século V a.C., sendo os conhecimentos apreendidos por transmissão oral, passados de geração em geração, da observação de outros a

prestarem cuidados e, muitas vezes, através de processos de tentativa e erro [Egenes, 2009]. Estes conhecimentos eram transmitidos nos templos, que funcionavam simultaneamente como santuários e escolas, onde se leccionavam os conhecimentos arcaicos dos conceitos de saúde. Surgiram então escolas especializadas para ensinar a arte dos cuidados de saúde, essencialmente no sul de Itália e na Sicília, que se expandiram para os grandes centros, ilhas e cidades da costa. Nessas escolas existiam várias teorias sobre o funcionamento do corpo humano, distúrbios e doenças. Essas teorias marcaram durante largos anos a evolução dos conhecimentos na área da saúde no seu modo empírico [Andrade, 2008].

Com o despertar da ciência e o surgimento da filosofia verifica-se uma evolução nas práticas de saúde, que começam, então, a basear-se em critérios mais científicos e filosóficos. Esta Era inicia-se no século V a.C. e dura até aos primeiros séculos da Era Cristã. As práticas de saúde, que até então estavam centradas em conceitos de misticismo e religiosos, começam a ser um resultado da experiência, do raciocínio lógico e da especulação filosófica, baseando-se na investigação livre e na observação dos fenómenos. Existia, no entanto, um grande entrave para uma evolução mais profunda relativamente aos cuidados de saúde: os conhecimentos anatómicos e fisiológicos relativos ao corpo humano eram bastante reduzidos. Nesta Era a prestação dos cuidados voltou-se para o homem e as suas relações com a natureza com as suas leis contínuas. Este período é identificado, pela Medicina Grega, como o período hipocrático [On-Line, 2012]. Hipócrates é considerado o pai da medicina, pois foi graças a si que a medicina se tornou uma área científica, abandonando a ideia que as doenças eram causadas pela ira e descontentamento de espíritos e deuses. A sua técnica consistia na observação do doente, ao qual era efectuado um diagnóstico de onde resultava um prognóstico. Após isso indicava e executava a terapêutica que julgava ser a correcta. O seu lema era: *não contrariar a natureza, porém auxiliá-la a reagir* [Ciências de Enfermagem, 2010]. Foi Hipócrates o impulsor do abandono das práticas associadas a dogmas místicos e religiosos em favorecimento dos métodos indutivos, de inspecção e observação.

A enfermagem como prática leiga¹ surge numa época em que as práticas de saúde eram influenciadas pelos factores sócio-económicos e políticos da Idade Média e das sociedades feudais. Este tipo de prática foi desenvolvida por membros de comunidades religiosas durante a época medieval, compreendida entre os séculos V e XIII [On-Line, 2012]. Foi durante esta fase da enfermagem que surgiu a enfermagem ligada ao Cristianismo. Os ideais da enfermagem prendiam-se com a caridade, servir os outros, e sacrifício próprio para o bem estar do outro, estando assim em consonância com os ideais cristãos. Todos estes valores perduram até aos dias de hoje como legítimos e necessários para uma prestação de Cuidados de Enfermagem eficiente.

Entre os séculos XIII e XVI verificou-se uma fase menos positiva na história evolutiva da Enfermagem. Durante esse período, que se enquadra no contexto do Renascimento e da Reforma Protestante, verificou-se um progresso social e intelectual, assim como uma nova visão sobre as práticas científicas, que coincidiu com a evolução das Universidades. No

¹Exercida por quem não possui conhecimentos.

entanto a enfermagem não acompanhou essa evolução. Continuou presa aos hospitais religiosos, estagnando na sua metodologia empírica e desarticulada, tendo ficado ainda mais frágil quando se deu a Reforma Religiosa e a Santa Inquisição. Esta época ficou intitulada como *Os Anos Negros da Enfermagem* [Pianucci, 2002]. Devido à Reforma Religiosa muitas ordens religiosas foram extintas e, conseqüente, os membros das ordens que prestavam serviço voluntário nos hospitais, maioritariamente mulheres, foram forçados a saírem dos mesmos. Por sua vez, o papel da mulher na sociedade também sofreu alterações, tendo estas que ficar em casa e obedecer aos seus maridos. Com tudo isto as mulheres viam, mais uma vez, o seu espaço no mercado de trabalho reduzir-se. Para piorar o cenário, as oportunidades que existiam eram em hospitais cujas condições eram tudo menos apetecíveis, como, por exemplo, colocarem mais que um doente em cada cama, alguns em posições invertidas, ficando os pés de um junto à cabeça do outro [Egenes, 2009]. Como não existia diagnóstico, após a admissão era possível encontrar doentes com uma perna partida a dividir cama com alguém que tivesse tuberculose. Para além das condicionantes hospitalares, a duração do horário de trabalho, os baixos salários e do *stress* que tais condições criavam, a Enfermagem deixou de ser desejável para as mulheres de classes mais elevadas. Com tudo isto começou a ser frequente deixar o cuidado dos doentes a ex-reclusas e prostitutas, mulheres cujo seu estatuto na sociedade não lhes permitia exercer outra actividade.

Com a Revolução Industrial, no século XVI, a visão sobre as práticas de saúde, nomeadamente as práticas de Enfermagem, começa a ser alterada. Muito dessa alteração deve-se à análise sob a óptica do sistema politico-económico da sociedade capitalista emergente, que começa a pretender institucionalizar a prática dos cuidados de Enfermagem. O desejo de melhorar as condições das pessoas que prestavam serviço nos hospitais levou a que grupos reformadores originassem iniciativas religiosas e sociais. Essas mudanças têm grandes efeitos, culminando com o surgimento da Enfermagem moderna no século XIX, em Inglaterra. Com o desenvolvimento da Medicina começa-se a verificar uma reorganização a nível hospitalar. Até então a Enfermagem encontrava-se numa fase obscura e à qual não se dava muita importância, mas com a reorganização hospitalar, tendo o médico como peça central, os processos de disciplina iniciam o seu progresso começando a reflectir-se na área da Enfermagem. Apesar de começar a dar os primeiros passos como ciência, a prática de Enfermagem continuou sob más condições, em grande parte devido ao elevado número de doenças infecto-contagiosas existentes e à escassez de pessoal especializado para cuidar dos doentes. Para além disso, as famílias mais abastadas continuavam a ser tratadas nas suas próprias casas, enquanto os mais pobres tornavam-se cobaias de experiências que iriam permitir um maior conhecimento das doenças em benefício de quem tinha mais posses.

A revolução na Enfermagem dá-se no chamado *Período Florence Nightingale*. Florence Nightingale é considerada por muitos como a **mãe da Enfermagem**, ela não só alterou como era praticada a enfermagem na época como criou as condições para ser reconhecida como profissão. Filha de pais ingleses, nasceu em 1820, em Florença, quando os seus pais faziam uma viagem pela Europa [Padilha and Mancia, 2005]. Filha de *boas famílias* teve uma educação privilegiada, tendo tido aulas de várias línguas, incluindo Latim, matemática, ciências e literatura antiga e moderna [Egenes, 2009]. Sendo extre-

mamente religiosa, a sua grande vontade era ajudar os doentes e os mais carenciados, com o objectivo de lhes amenizar o sofrimento. Deu os primeiros passos na Enfermagem ao frequentar um estágio no **Instituto de Diaconisas de Kaiserswerth**², na Alemanha, onde ganhou os primeiros conhecimentos dessa área, tais como regras e horários rígidos e divisão do ensino por classes sociais [Padilha and Mancia, 2005]. Outro marco importante está relacionado com a sua passagem pelo **Hôtel-dieu**³, em Paris, onde interiorizou o trabalho desenvolvido pelas Irmãs de Caridade de São Vicente de Paulo, que se preocupavam mais em formar o carácter das suas discípulas do que em ensinar conhecimentos específicos de enfermagem. Durante o tempo que passou nessa instituição observou todo o tipo de tarefas, incluindo o trabalho assistencial e administrativo que lá se realizava, como as suas regras e formas de cuidar dos doentes. A sua aprendizagem passou por tirar várias anotações, realizar gráficos e listas das actividades desenvolvidas, como na aplicação de questionários, com o objectivo de aprofundar os seus conhecimentos e melhorar a sua organização.

Em 1854 foi convidada, pelo então Secretário de Estado para a Guerra do Reino Unido, Sidney Herbert, para prestar assistência aos soldados na Guerra da Crimeia. Ao chegar lá, em conjunto com outras 38 mulheres, deparou-se com uma realidade assustadora. A taxa de mortalidade dos soldados britânicos era de 44% [Egenes, 2009], causadas na sua maioria pela total falta de recursos e pelo desconhecimento da população relativamente à maior parte das doenças. Entre as doenças identificadas por Nightingale encontravam-se a cólera, disenteria e tifo [Padilha and Mancia, 2005]. Os hospitais estavam sujos, com os soldados feridos amontoados no chão, pois não existiam camas, e não havia sabão nem medicamentos ou outros suplementos. As suas maiores medidas passaram pela reforma da higiene, saneamento e práticas de enfermagem. Para isso contratou pessoas para lavar e limpar e implementou turnos, que funcionavam inclusive de noite, nos quais as enfermeiras levavam uma lanterna a óleo, que se tornou no símbolo actual da enfermagem (Fig. 3.1).

Apesar da falta de cooperação dos oficiais militares e médicos para com as medidas desejadas por Nightingale, esta, juntamente com a sua equipa, conseguiu reduzir a taxa de mortalidade entre os soldados para 2%. O seu maior contributo esteve relacionado com as orientações dadas sobre a importância da higiene individual, colectiva e ambiental, até aí descuidadas pelos militares. Durante esse tempo que esteve na Crimeia, ela própria adoeceu, dedicando-se a partir daí a trabalhos intelectuais.



Figura 3.1: Símbolo da Enfermagem [Lyavelino, 2007]

²Instituto criado pelo pastor luterano Theodor Flidner e pela sua esposa Frederika, cuja organização era muito semelhante à praticada pelas Irmãs de Caridade de São Vicente de Paulo.

³O *Hôtel-dieu* de Paris, fundado em 651 por Saint Landry, é um dos mais antigos hospitais europeus em funcionamento [McHugh, 2006].

Pelos serviços prestados, ao regressar da guerra, recebeu um prémio do Governo Inglês, o qual utilizou para fundar, em 1859, uma Escola de Enfermagem no Hospital Saint Thomas, que serviu de modelo para muitas outras que se viriam a fundar futuramente. A sua excelência no ensino da Enfermagem deveu-se, sobretudo, à rigorosa disciplina, muito ao estilo militar absorvido por Nightingale no tempo que esteve na Crimeia, e também à exigência com que seleccionava as candidatas relativamente às suas qualidades morais. O curso tinha uma duração de um ano e consistia em aulas diárias leccionadas por médicos, pois eram os únicos que possuíam a qualificação necessária para decidir quais as funções que as enfermeiras podiam realizar. Nightingale falece a 13 de Agosto de 1910 [Ciências de Enfermagem, 2010], mas deixando o seu contributo na Enfermagem, pois foi com ela que realmente esta área da saúde deixou de ser vista como uma actividade empírica e passou a ser reconhecida como uma actividade social institucionalizada e específica.

Desde essa época até aos dias de hoje a prática de Enfermagem sofreu uma grande evolução. Actualmente, a Enfermagem vê o doente como um ser holístico⁴ [Leitão et al., 2000], que possui família, com um passado e futuro, crenças e valores que têm um papel determinante no histórico de saúde de cada um. Todos estes elementos fazem parte do indivíduo e nenhum deles pode ser desprezado. Isto faz com que a Enfermagem esteja englobada na área das Ciências Sociais e não das Ciências Naturais. Durante o exercício da sua actividade é frequente os enfermeiros depararem-se com situações para as quais não há respostas científicas, tendo que usar a sua experiência enquanto pessoas. Um dos factores que caracteriza a profissão enfermagem é que é necessário possuir conhecimentos além do empírico, tais como crenças e valores. A Enfermagem está constantemente em grande desenvolvimento, não só em termos humanos, pois são os enfermeiros quem passa mais tempo em contacto com os doentes, como também em termos técnicos. Um enfermeiro tem que possuir um elevado conhecimento em relação a procedimentos, técnicas e fármacos, que para além de ter que os conhecer tem, também, a obrigatoriedade de saber administrá-los.

Com o objectivo de auxiliar os enfermeiros, o Conselho Internacional de Enfermeiros, CIE⁵, avançou para a implementação de uma linguagem direccionada para a prática de Enfermagem. Assim, actualmente a Enfermagem possui uma linguagem própria, a Classificação Internacional para a Prática de Enfermagem (CIPE), que auxilia os enfermeiros na formulação de diagnósticos de Enfermagem, planeamento das intervenções e avaliação dos resultados consequentes.

3.2 Linguagem CIPE

Na área da Enfermagem, tem-se vindo a verificar uma preocupação cada vez maior na elaboração de sistemas de classificação partindo do princípio que as equipas de enfermagem são um grupo sócio-profissional que utilizam um vocabulário técnico muito específico e particular, ao qual se pode chamar *linguagem especial de Enfermagem* [Nóbrega and

⁴É preciso ver o doente no seu *todo* e não em partes separadas.

⁵ICN - International Council of Nurses.

Garcia, 2005].

Devido à existência dessa linguagem, nas últimas décadas, tem-se verificado uma maior preocupação por parte dos responsáveis em torná-la numa linguagem unificada, permitindo assim ser possível aceder à informação pertinente muito mais facilmente [Nóbrega and Garcia, 2009]. Para tal ser possível, chegou-se à conclusão que é necessária a aplicação de normas de terminologia para ser possível desenvolver sistemas de informação que permitam auxiliar os enfermeiros nas suas decisões clínicas. A finalidade de uma linguagem uniformizada prende-se com a capacidade que a mesma teria para descrever os cuidados de enfermagem prestados tanto ao indivíduo, como à sua família e comunidade. Teria também o propósito de possibilitar comparações de dados e demonstrar e prever tendências na prestação de cuidados de enfermagem. A existência de um *framework* que tivesse tais características foi recomendada pela Organização Mundial de Saúde ao CIE, que pretendia adicionar aos Sistemas de Classificação Internacionais de Diagnósticos e Procedimentos Médicos uma Classificação de Enfermagem que contemplasse o que os enfermeiros faziam (Acções/Intervenções de Enfermagem) consoante necessidades humanas específicas (Diagnósticos de Enfermagem) para obter os resultados esperados (Resultados de Enfermagem) [Mazoni et al., 2010]. Com base neste pedido, foi aprovado em 1989, no Congresso Quadrienal do CIE, em Seul - Coreia do Sul, uma proposta que visava a elaboração dessa mesma linguagem comum [Nóbrega and Garcia, 2005].

A primeira fase passou pelo levantamento dos termos utilizados nos Sistemas de Classificação existentes, entre os quais o sistema *North American Nursing Diagnosis Association*, **NANDA**, *Nursing Interventions Classifications*, **NIC**, *Nursing Outcomes Classifications*, **NOC**, *Omaha System* e *Home Health Care Classification*. Levantamento esse que só teve fim em 1993. Com base nesses termos foi publicado em 1996 [Nóbrega and Garcia, 2009], a *Classificação Internacional para a Prática de Enfermagem - Versão Alfa*, que permitia a configuração cruzada desses termos e que possibilitava a integração de novos termos que pudessem surgir. No seguimento foi lançada a *Versão Beta*, em 1999, a *Versão Beta 2*, em 2001, em 2005 a *CIPE Versão 1.0*, a *Versão 1.1*, em 2008, estando actualmente em vigor a versão 2.0, publicada em 2009 [Cubas et al., 2010].

A versão *Alpha* baseou-se na conceitualização dos termos identificados inicialmente e estruturando-os numa hierarquia de três pirâmides ou modelos [Silva et al., 2008]. A primeira pirâmide foi construída como uma classificação mono-axial, em cujo vértice superior apenas se encontrava um princípio geral de divisão e representava o domínio do cliente, que tanto podia ser o indivíduo ou o meio ambiente. Nesta pirâmide descreviam-se 292 Fenómenos de Enfermagem [Nóbrega and Garcia, 2009]. A segunda pirâmide descrevia as Intervenções de Enfermagem, e ao contrário da primeira subdividia-se em seis eixos, sendo, portanto, uma classificação multi-axial. Os eixos representados eram: Acção, Objecto, Método, Recurso, Lugar do Corpo e Tempo/Lugar [Mazoni et al., 2010]. A terceira pirâmide pretendia descrever os Resultados de Enfermagem, porém não chegou a ser implementada [Silva et al., 2008].

A versão *Beta* foi construída com base nas necessidades detectadas na primeira versão e

que foram comunicadas ao CIE. A principal diferença verificou-se no tipo de classificação dos Fenómenos de Enfermagem, que simboliza os factores relevantes para a prática de Enfermagem, que passou a ser multi-axial, sendo composta por oito eixos: Foco da Prática de Enfermagem, Juízo, Frequência, Duração, Lugar do Corpo, Topologia, Portabilidade e Portador. Por sua vez, a estrutura das Intervenções foi renomeada para Acções de Enfermagem, definidas como *"o desempenho dos enfermeiros na prática assistencial"* [Cubas et al., 2010], e passou a contar, também, com oito eixos: Tipo de Acção, Alvo, Recurso, Tempo, Topologia, Localidade, Via e Beneficiário [Cubas et al., 2010]. Com esta mudança a linguagem tornou-se mais flexível e com maior liberdade para formar combinações entre os conceitos. Por exemplo, os conceitos *dor* e *agudo* podem ser unificados e formar um conceito complexo *dor aguda*.

A versão *Beta 2* manteve-se quase idêntica à sua antecessora, apenas possuindo algumas correcções e modificações relacionadas com codificações e conteúdo gramatical, assim como a inserção de novos códigos, conceitos e termos. Para se construir um Diagnóstico de Enfermagem recorrem-se aos termos presentes nos eixos da Classificação dos Fenómenos de Enfermagem, para compor as Intervenções são utilizados os termos dos eixos da Classificação das Acções de Enfermagem e, por último, para se formarem Resultados de Enfermagem são utilizados os termos dos eixos da Classificação de Fenómenos de Enfermagem, como para os Diagnósticos.

Com versão *CIPE 1.0* pretendeu-se resolver os problemas de redundância e ambiguidade existentes na versão *Beta 2*, de maneira que fosse mais fácil cativar e empolgar os enfermeiros na adopção desta nova linguagem. O caminho encontrado para resolver esses problemas passou por unificar as duas estruturas existentes (Fenómenos e Acções de Enfermagem) numa única estrutura com sete eixos, Fig. 3.2: **Foco, Juízo, Cliente, Acção, Recursos, Localização e Tempo**.

Segundo o CIE [Cubas et al., 2010], as definições de cada um destes eixos são:

1. **Foco:** a área de atenção que é relevante para a Enfermagem
2. **Juízo:** opinião clínica ou determinação relacionada ao foco da prática de enfermagem
3. **Cliente:** sujeito ao qual o diagnóstico se refere e que é o recipiente de uma intervenção
4. **Acção:** um processo intencional aplicado a um cliente
5. **Recursos:** uma maneira ou um método de desempenhar uma intervenção de enfermagem
6. **Localização:** orientação anatómica e espacial de um diagnóstico ou intervenções
7. **Tempo:** o momento, período, instante, intervalo ou duração de uma ocorrência

Sendo as definições das declarações as seguintes:

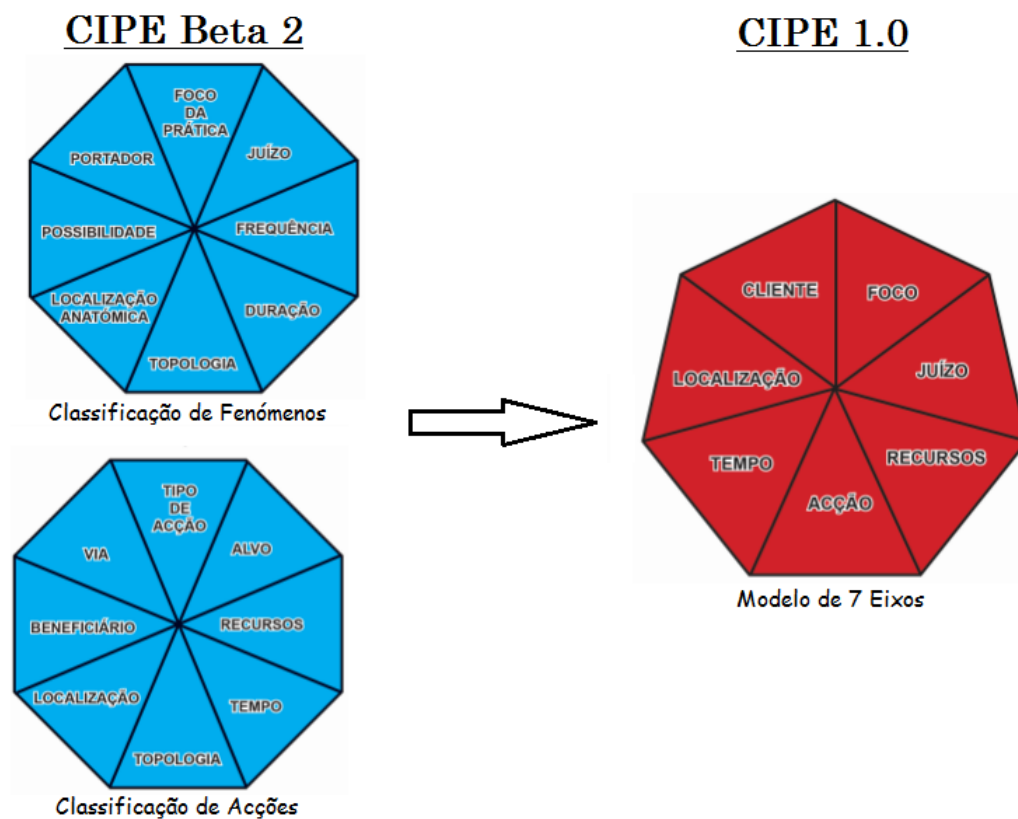


Figura 3.2: Evolução do modelo da **CIPE Beta 2** para a **CIPE 1.0** [Garcia and Nóbrega, 2009]

- **Fenómeno:** factores que influenciam o estado de saúde com características específicas (fenómenos que os enfermeiros diagnosticam).
- **Diagnóstico:** designação atribuída por uma enfermeira à decisão sobre um fenómeno que representa o foco das intervenções. Representa o estado do cliente, problemas, necessidades e potencialidades.
- **Intervenção:** acção realizada em resposta a um Diagnóstico de Enfermagem com a finalidade de obter um Resultado.
- **Resultado:** Medida ou condição de um Diagnóstico num intervalo de tempo após uma intervenção.

Para além da reestruturação, a versão 1.0 apresenta a inclusão de uma ontologia, que, recorrendo a recursos computacionais, classifica conceitos hierarquicamente, verificando quais são as combinações e restrições existentes entre os termos. Para a construção de Enunciados de Diagnósticos, Resultados e Intervenções a CIE recomenda um conjunto de regras [ICN, 2009]:

- **Diagnósticos e Resultados:**

- Têm de incluir um termo do **Eixo do Foco**.
 - Têm de incluir um termo do **Eixo do Juízo**.
 - Podem incluir termos adicionais, de acordo com a necessidade, dos **Eixos do Foco, Juízo ou de outros Eixos**.
- **Intervenções:**
 - Têm de incluir um termo do **Eixo da Acção**.
 - Têm de incluir pelo menos um **Termo Alvo**. Um termo alvo pode ser um termo de qualquer eixo, excepto do **Eixo do Juízo**.
 - Podem incluir termos adicionais, de acordo com a necessidade, do **Eixo da Acção** ou de qualquer outro Eixo.

Com o reconhecimento da importância que os Sistemas de Informação têm para a decisão, qualidade e segurança dos cuidados, a Ordem dos Enfermeiros de Portugal, associando-se à CIE, lançou, em Março de 2006, a versão oficial da CIPE em português da Primeira Edição da CIPE 1.0 [Cubas et al., 2010].

A versão **1.1** da CIPE, lançada em 2008, apenas ficou disponível na *WEB*, não estando prevista qualquer publicação impressa. Esta versão foi a primeira a estar de acordo com a norma **ISO 18.104 - Integração de um Modelo de Terminologia de Referência para Enfermagem**. Outra inovação está relacionado com a implementação de declarações de Diagnósticos, Resultados e Intervenções de Enfermagem no corpo da Classificação [Cubas et al., 2010]. Através destas declarativas começou a ser possível a criação de Catálogos de Enfermagem. Os Catálogos permitem definir um sub-conjunto de Diagnósticos, Intervenções e Resultados de Enfermagem, pré-elaborados, que se destinam a serem utilizados numa área específica da Enfermagem. Os Catálogos CIPE, após serem implementados e aprovados pela CIPE, devem ser divulgados, tanto através de apresentações e *posters* em conferências profissionais como de manuscritos para publicação. Essa divulgação deve ser eficiente, pois só assim podem ser testados e validados pelos vários países.

Actualmente, a CIPE está na versão **2.0**. Esta versão foi anunciada em Julho de 2009 aquando do Congresso do CIE em Durban, na África do Sul. Esta versão é o resultado do ajustamento que foi necessário realizar para integrar a classificação na **Família das Classificações Internacionais da Organização Mundial de Saúde**. Para além desse ajuste, foram também acrescentados mais de 400 novos conceitos.

O *Browser* CIPE 2.0 pode ser consultado na página oficial do CIE⁶, em <http://www.icn.ch/pillarsprograms/icnpr-browsers/>, ou na página da Ordem dos Enfermeiros, em <http://www.ordemenfermeiros.pt/browserCIPE/BrowserCIPE.aspx>.

⁶Está disponível em vários idiomas, incluindo o português.

Capítulo 4

HP-HCIS

O HP-HCIS é a solução proposta pela HP para um sistema de informação integrado para o sector da saúde. Esta solução teve origem em Espanha e foi implementada devido às restrições nos sistemas de gestão hospitalar *HIS1* e *HIS2*, desenvolvidos a partir do início da década de 1990 [Sebek and Mladek, 2010], baseando-se nos mesmos, de maneira a que o processo de transição fosse mais suave para os utilizadores. Esses sistemas tinham como objectivo fazer a gestão de um único hospital, essencialmente ao nível administrativo (admissões e marcações), gestão empresarial (controlo de custos) e algumas funcionalidades de pedidos e resultados.

O HP-HCIS permite um nível de exaustividade e coesão nos sistemas de informação de saúde. É perfeitamente implementável em qualquer organização, pois é uma aplicação *Web-Based*, seja ela uma pequena clínica ou um grande centro hospitalar, oferecendo um método simples e prático de coordenar e prestar Cuidados de Saúde. Esta solução, sendo um Sistema Integrado de Informação Clínica, tem como principais objectivos garantir a homogeneidade de prestações e tecnologias, assegurar a melhor prestação de cuidados tendo como referência as melhores práticas de assistência e fortalecer a utilização do Processo Clínico Electrónico, **PCE**, como o centro de todas as actividades. Para atingir esses objectivos está baseada nos seguintes princípios:

- Foco no doente
- Integração entre os vários níveis de assistência
- Coordenação entre os diferentes prestadores de cuidados
- Assistência continuada
- Segurança para o doente

O HP-HCIS permite a todos os intervenientes nos processos de saúde acederem a todas as funcionalidades do sistema, o que faz dele um verdadeiro Sistema Integrado de Saúde. Assim não existem barreiras entre os vários níveis e meios de assistência que impeçam os médicos, profissionais de saúde, enfermeiros e gestores de acederem a toda a informação que pretendam para efectuar o seu trabalho de decisões clínicas, de prestação de cuidados e de gestão. É então possível os diferentes profissionais conseguirem aceder a toda a informação relevante, a qualquer hora e qualquer lugar. Entre essas actividades pode-se nomear a possibilidade de visualizar documentos e imagens clínicas, analisar a evolução dos sinais vitais, programar uma intervenção cirúrgica e prescrever medicação.

O núcleo do HP-HCIS é o Processo Clínico Electrónico do doente que abrange toda a organização. A arquitectura utilizada para o PCE utiliza o padrão **HL7 V3 RIM**¹ [System, 2008] e incorpora todos os tipos elementos existentes num PCE: processo clínico, documentos, notas, ordens médicas, resultados de exames, consultas, internamentos, episódios de urgências, etc.

Outra particularidade do HP-HCIS é que cada organização pode definir a sua própria estrutura de documentos existentes no PCE. Qualquer tipo de documento pode ter um formulário associado², onde o utilizador regista a informação de maneira estruturada, que ficam armazenados na Base de Dados do HP-HCIS e podem ser consultados posteriormente. Através desta metodologia liberta-se o utilizador de qualquer tipo de formatação caso utilizasse um documentos comum.

O HP-HCIS possui um PCE comum a toda a organização que é orientado ao processo clínico, ou seja, cada doente possui uma lista de processos clínicos que compõem o PCE. Existe uma ligação semântica entre todos os elementos de informação (documentos, anotações, exames, etc.) e os processos clínicos, assim como a outros elementos. O HP-HCIS permite, por exemplo consultar informação recorrendo ao episódio ou ao problema. Cada elemento possui associada uma especialidade, com isto torna-se possível a cada especialista filtrar a informação de um determinado doente, seleccionando a informação que lhe interessa efectivamente. Através da utilização orientada ao processo clínico o sistema permite obter informação sobre assuntos completamente distintos, como qual o custo de um determinado tratamento, a sequência em que ocorreram as diversas prestações de cuidados enquanto o doente esteve na Organização Prestadora de Serviços de Saúde, etc.

4.1 Modelo Funcional

O modelo funcional do HP-HCIS assenta em sete componentes [García, 2008]:

- **Gestão de Prestação de Cuidados:** Neste segmento estão incorporados todos os processos de prestação de Cuidados. Esses processos podem ser Cuidados Primários,

¹ *Health Level 7* é um padrão utilizado para esquematizar a linguagem médica.

² Os formulários podem ser construídos recorrendo a uma ferramenta disponível no HP-HCIS, o Editor de Formulários.

Cuidados Especializados e Cuidados Sociais de Saúde, tendo sempre como referência o PCE e o processo de Cuidados Continuados. Este segmento contempla ainda os processos de Emergências, Saúde Mental e Dependências, estando integrado com o Repositório de Doentes, todos os Departamentos e os restantes componentes deste modelo.

- **ERP (Enterprise Resource Planning):** Planificação de Recursos. Este componente do modelo integra a secção relativa à gestão e aos processos de apoio. Esses processos são relativos à gestão Económica e Financeira, Logística e de Recursos Humanos. Os processos económico-financeiros são utilizados para permitir a criação, aprovação, execução e seguimento do que é pressuposto, possibilitando analisar os custos de cada área, departamento ou qualquer outro conceito que se pretenda visoriar. Permite igualmente a integração com a área de Cuidados em matérias de compras, possibilitando uma avaliação e monitorização de todos os gastos.
- **BI (Business Informatics):** Informação de Negócio. Neste componente estão enquadrados aspectos referentes a Modelos, Políticas e Planificações de Saúde, assim como normas. Pretende ajudar na tomada de decisões, sejam elas efectuadas por médicos, gestores clínicos, económico-financeiros ou de recursos humanos, fornecendo a informação necessária sobre todos os perfis e profissionais de uma Organização de Saúde. Essa capacidade de resposta eficaz à tomada de decisões é possível através da integração de toda a informação existente em sistemas transaccionais num único Sistema Integrado de Negócio.
- **Gestão de Conhecimento:** Possibilidade de aceder *on-line* a todos as evidências médicas sobre tratamentos, informação necessária para desempenhar uma boa actividade e para desenvolvimento profissional, por parte dos profissionais de saúde. Essa informação pode ser obtida na já referida Informação de Negócio, bem como em formações, investigações, fóruns, notícias, comunicações e Guias de Prática, como, por exemplo, através da metodologia MBE³.
- **EAI (Enterprise Application Integration):** Plataforma de integração e *workflow* entre os vários processos de negócio. Para além de realizar a integração dos sistemas e processos já existentes, é possível integrar outros sistemas que se pretendam. O *workflow*, numa organização de saúde é uma peça fundamental. Os médicos, enfermeiros e todos os funcionários colaboram estritamente pois os Cuidados de Saúde requerem muita coordenação e cooperação. O HP-HCIS possui um *Workflow Builder* que permite definir e automatizar processos de negócio específicos de cada organização. Esses processos podem envolver pessoas de várias áreas da organização. A grande maioria desses processos tem origem numa decisão clínica, tais como incluir um doente em lista de espera ou programar uma consulta de seguimento.

³Metodologia Baseada por Evidências: *é um movimento médico que se baseia na aplicação do método científico a toda a prática médica, especialmente àquelas tradicionalmente estabelecidas que ainda não foram submetidas ao escrutínio sistemático científico. Evidências significam, aqui, provas científicas* [Wikipédia, 2011b].

Todos estes processos são enviados a partir da estação de trabalho dos médicos e são incluídos, imediatamente, no processo clínico. Através da utilização do padrão HL7 é possível utilizar o HP-HCIS com qualquer outro sistema de gestão de doentes, proporcionando assim uma continuidade de Cuidados em ambientes heterogéneos.

- **Portal:** Implementação de um portal único, que pode ser acedido por cidadãos, doentes e profissionais. A interface HP-HCIS é altamente configurável, uma característica importante para um portal único que tem que satisfazer as necessidades de cada um dos seus utilizadores. Os utilizadores, que possuam permissões para tal, podem criar novos painéis de informação, combinar painéis existentes originando uma vista multi-painel, otimizando-os consoante as suas necessidade. Por exemplo, um médico de Urgências pode configurar a sua página inicial com a lista de pacientes admitidos nas urgências, enquanto um enfermeiro de internamento pode colocar, na sua página inicial, o mapa de camas do seu serviço. Toda a interface, os menus, os ícones e o idioma é personalizado, assim como a navegação entre as várias funcionalidades. Não é necessária alternar entre módulos (ver Secção 4.2) para conseguir executar todas as actividades que pretende. Por sua vez, não existem perfis associados a módulos, podendo cada utilizador possuir todas as permissões indispensáveis para as suas necessidades enquanto gestor ou prestador de Cuidados.
- **Infraestrutura:** O modelo funcional está implementado sobre uma estrutura totalmente centralizada na qual se realizam todos os processos e sistemas descritos anteriormente, cuja gestão de *hardware* e de rede é efectuada pela Organização de Saúde.

4.2 Módulos HP-HCIS

Esta solução é bastante modular e flexível, e como já foi referido, possui uma grande capacidade de integração, o que lhe permite adaptar-se a qualquer tipo de hospital ou de organização. Neste momento, o HP-HCIS está em utilização em cerca de 100 hospitais espalhados por Portugal, Espanha e Índia [HP, 2011]. Em Portugal, está activo em quatro projectos: no Centro Hospitalar de Lisboa Ocidental (CHLO), no Centro Hospitalar de Lisboa Central (CHLC), no Centro Integrado de Cirurgia de Ambulatório (CICA) no Porto, e na Unidade Local de Saúde de Matosinhos (ULSM). A sua modularidade e interoperabilidade permite aos clientes possuírem módulos dos seus sistemas de informação de outros fornecedores e ser possível manterem um sistema sólido e integrado. Esta modularização permite introduzir módulos independentes em sistemas cujo sistema principal é de outro fornecedor ou seleccionar módulos específicos para um determinado cliente. Os módulos que o HP-HCIS possui são vários, que permitem fazer a gestão de toda a actividade existente numa unidade de saúde, de onde destacamos os seguintes:

- **Internamento:** Permite efectuar a gestão administrativa dos doentes internados em qualquer área de uma Organização de Saúde. Essa gestão inclui a realização dos

processos de ingresso, alta, transferência, mudança de responsável, transferência em ambulância, gestão de camas e documentação respeitante a todos estes actos.

- **Urgências:** Efectua a gestão administrativa dos doentes que deram entrada no serviço de Urgências da unidade de saúde. Efectua a gestão dos processos de identificação do doente, motivo da ida às Urgências, unidades de observação, hospitalização, e todos os processos respeitantes à estadia do doente nessa área, tais como triagens e reencaminhamentos para o sector indicado. Faz também a gestão de todos os documentos associados, como o acesso ao processo clínico e documentos relacionados com a *estadia* nas Urgências (etiquetas, justificações, etc.).
- **Enfermaria:** Este módulo corresponde às actividades de Enfermagem prestadas enquanto o doente permanece na unidade de saúde. Aqui incluem-se a realização da Avaliação Inicial, a gestão do Plano de Cuidados, a prestação de cuidados com os respectivos registos de acções, consoante o que está estabelecido no Plano de Trabalho e todos os dados associados, assim como permite efectuar o seguimento do estado de saúde dos doentes.
- **Bloco Cirúrgico:** Permite fazer a gestão de toda a área respeitante ao Bloco Cirúrgico. Entre as funcionalidades encontram-se a programação de salas, assim como a sua manutenção. Gere também tudo o que diz respeito a marcações, cancelamentos, suspensões e confirmações de cirurgias. Outra funcionalidade neste módulo é a possibilidade de registar toda a informação respeitantes à realização da cirurgia, como, por exemplo, quais foram os intervenientes (médicos, anestesistas, enfermeiros, etc.), os tempos relativos à cirurgia, assim como todo o material utilizado.
- **Consultas Externas:** Módulo que efectua a gestão hospitalar referente às consultas externas. Permite configurar agendas de consultas por médico, especialidade ou necessidade de material. Permite ao utilizador marcar consultas automaticamente (na primeira vaga existente) ou por pesquisa de vagas. Processa, também, toda a informação relativa à consulta propriamente dita: chegada, atendimento e confirmação de realização.

4.3 Segurança

A arquitectura do HP-HCIS está em conformidade com a norma **CEN/ISO EN13606**. Esta norma têm como objectivo estabelecer uma arquitectura de informação estável e rigorosa de como o PCE deve ser manuseado. Segundo a [CIE/ISO, 2012] os principais desejos relativamente ao PCE são:

- *Preservar o significado clínico original pretendido pelo autor;*
- *Reflectir a confidencialidade dos dados como pretendido pelo autor e pelo doente;*

Para cumprir estas normas o HP-HCIS implementa vários mecanismos de segurança. Sempre que é adicionado um elemento ao PCE este é assinado explicitamente pelo seu autor ou deixado aberto temporariamente. Todos os tipos de elementos possuem um período de latência, passado o qual o elemento é fechado caso ainda esteja aberto. Após ter sido fechado ou assinado o elemento apenas pode ser versionado para corrigir erros, mas todas as versões de um determinado elemento ficam guardadas, sendo possível aceder e consultar o seu conteúdo, assim como quem e quando versionou o elemento. Outra funcionalidade de segurança está relacionada com a possibilidade de atribuir políticas de acesso que permitem estipular qual a informação, por quem, como, quando e onde pode ser vista. Outro factor implementado está relacionado com as permissões e privilégios. Como o HP-HCIS é utilizado por várias pessoas, que desempenham funções distintas implementou-se este sistema que permite seleccionar quem tem permissão e privilégio para o quê. Por fim, existe ainda um Sistema de Auditoria, que possibilita gravar a sessão completa de cada utilizador, possibilitando a necessidade futura de se ir conferir o que determinado utilizador realizou numa data passada.

Capítulo 5

Metodologia de Desenvolvimento

Para a implementação deste projecto optámos por utilizar uma metodologia de desenvolvimento *Agile*, o **Scrum**¹.

As metodologias ágeis para o desenvolvimento de *software* nasceram da necessidade dos programadores encontrarem uma solução para os problemas detectados nas metodologias tradicionais de desenvolvimento de software. Um exemplo de metodologia tradicional é o modelo em Cascata que é composto por actividades sequenciais. Essas actividades constam de levantamento de requisitos, análise, projecto, implementação, testes e manutenção [Soares, 2004]. Este tipo de metodologia, que vê uma aplicação como um todo, não possibilita os melhores resultados, pois é bastante complicado definir todas as necessidades da aplicação antes do início da sua implementação. No modelo em Cascata, o resultado de cada etapa é utilizado como base para a etapa posterior. Se numa das etapas iniciais existir uma falha, apenas detectada numa etapa posterior, ou uma alteração de requisitos, o projecto necessita de ser readequado, o que pode significar um custo enorme, tanto em termos financeiros como de qualidade do produto final. As metodologias ágeis pretendem fornecer uma alternativa a estes tipos de desenvolvimento.

As metodologias ágeis são cada vez mais utilizadas no desenvolvimento de *software*. Estas metodologias pretendem desburocratizar as actividades inerentes ao desenvolvimento, utilizando conceitos de ciclos iterativos e entregas rápidas e constantes de *software* a funcionar [Cavalcanti et al., 2009]. Estes conceitos vão de encontro às necessidades dos clientes, pois cada vez mais é necessário adequar o produto final àquilo que o mercado quer. Como o mercado está em constante mutação e evolução torna-se imperativo que as alterações

¹Na realidade, foi utilizada uma metodologia *pseudo-Scrum*, pois a mesma não cumpria todos os requisitos do *Scrum* puro. Mais detalhes desta metodologia, assim como as diferenças existentes na metodologia aplicada, podem ser encontradas nas Subsecções 5.1 e 5.2.

aos requisitos inicialmente pensados e planeados sejam possíveis e o seu custo minimizado. Assim, é também possível adaptar e equipar o projecto com novas tecnologias que vão surgindo.

Historicamente, pode-se estabelecer o surgimento oficial do movimento das metodologias ágeis em 2001², quando dezassete especialistas em desenvolvimento de *software*, que representavam as metodologias *Extreme Programming*, *Scrum* e *Crystal*, entre outras, se reuniram e identificaram doze princípios comuns a todas elas [Soares, 2004] [Beck et al., 2001]:

1. *A nossa maior prioridade é, desde as primeiras etapas do projecto, satisfazer o cliente através da entrega rápida e contínua de software com valor*
2. *Aceitar alterações de requisitos, mesmo numa fase tardia do ciclo de desenvolvimento. Os processos ágeis potenciam a mudança em benefício da vantagem competitiva do cliente*
3. *Fornecer frequentemente software funcional. Os períodos de entrega devem ser de poucas semanas a poucos meses, dando preferência a períodos mais curtos*
4. *O cliente e a equipa de desenvolvimento devem trabalhar juntos, diariamente, durante o decorrer do projecto*
5. *Desenvolver projectos com base em indivíduos motivados, dando-lhes o ambiente e o apoio de que necessitam, confiando que irão cumprir os objectivos*
6. *O método mais eficiente e eficaz de passar informação para e dentro de uma equipa de desenvolvimento é através de conversa pessoal e directa*
7. *A principal medida de progresso é a entrega de software funcional*
8. *Os processos ágeis promovem o desenvolvimento sustentável. Os promotores, a equipa e os utilizadores deverão ser capazes de manter, indefinidamente, um ritmo constante*
9. *A atenção permanente à excelência técnica e um bom desenho da solução aumentam a agilidade*
10. *Simplicidade – a arte de maximizar a quantidade de trabalho que não é feito – é essencial*
11. *As melhores arquitecturas, requisitos e desenhos surgem de equipas auto-organizadas*
12. *A equipa reflecte regularmente sobre o modo de se tornar mais eficaz, fazendo os ajustes e adaptações necessárias*

²Efectivamente, as metodologias ágeis já existiam antes de 2001, no entanto só após essa data surgiu um mecanismo que as definiu.

Dessa reunião surgiu a **Aliança Ágil**, a qual elaborou um manifesto, o *Agile Manifesto* [Beck et al., 2001]. Da elaboração desse manifesto, tendo como base os doze princípios enumerados anteriormente, resultaram quatro premissas que são os fundamentos das metodologias ágeis [Fadel and Silveira, 2010] e que fazem parte do próprio manifesto [Beck et al., 2001]:

Manifesto para o Desenvolvimento Ágil de Software

*”Ao desenvolver e ao ajudar outros a desenvolver software,
temos vindo a descobrir melhores formas de o fazer.
Através deste processo começámos a valorizar:*

*Indivíduos e interacções mais do que processos e ferramentas
Software funcional mais do que documentação abrangente
Colaboração com o cliente mais do que negociação contratual
Responder à mudança mais do que seguir um plano*

*Ou seja, apesar de reconhecermos valor nos itens à direita,
valorizamos mais os itens à esquerda.”*

As metodologias ágeis são metodologias que se recomendam para situações onde a mudança de requisitos são constantes e seja necessário entregar constantemente os resultados ao cliente. Devido às suas características de adaptação e flexibilidade são as adequadas para estes cenários.

A sua aplicação consiste em dividir o desenvolvimento em ciclos, ou iterações, curtos, no fim dos quais o cliente recebe uma nova versão do produto, a funcionar, com as alterações desejadas. Com isto, a cada ciclo podem ser redefinidos os requisitos que se desejam implementar, reduzindo os riscos, e conseqüentemente os custos, do projecto.

Enquanto as metodologias tradicionais mantêm o foco na documentação e no cumprimento rigoroso de processos, as metodologias ágeis pretendem centrar-se no desenvolvimento propriamente dito e nas relações entre os participantes.

5.1 Metodologia Scrum

O aparecimento de metodologias ágeis começou a ser notório na década de 1990, com influência nas práticas da indústria japonesa, principalmente pelos métodos de trabalho implementados pelas companhias Honda e Toyota [Carvalho and Mello, 2009]. A metodologia *Scrum* enquadra-se nesta área de metodologias, tendo sido criado para gerir projectos de produção automobilística e de produtos de consumo [Fadel and Silveira, 2010].

Esta metodologia foi desenvolvida por Jeff Sutherland em 1993, tendo como base o artigo *The new product development game* de Takeuchi e Nonaka, de 1986. Nesse artigo os seus autores evidenciaram as vantagens de se utilizarem equipas pequenas e multi-disciplinares na obtenção de melhores resultados [Carvalho and Mello, 2009]. No entanto a sua definição só foi formalizada em 1995, por Ken Schwaber, que consolidou este método como uma metodologia de desenvolvimento de *software* no mundo todo [Fadel and Silveira, 2010].

O seu nome, *Scrum*, tem origem numa comparação entre os programadores e os jogadores de *Rugby*. *Scrum* é uma acção praticada pelos jogadores de *Rugby* que consiste numa rápida reunião antes de realizarem qualquer jogada. Nesta modalidade desportiva é necessário jogar em conjunto, como uma unidade única, mas onde cada jogador desempenha um papel específico e todos lutam pelo mesmo objectivo. Assim também são as equipas de desenvolvimento que adoptam a metodologia *Scrum*.

Esta metodologia é composta por um conjunto de regras e práticas de gestão que devem ser seguidas para se garantir o sucesso do projecto, não oferecendo qualquer tipo de prática para a fase de desenvolvimentos. Segundo [Oliveira and Lima, 2011] são seis as características base do *Scrum*: flexibilidade dos resultados, flexibilidade dos prazos, equipas pequenas (entre 6 a 10 elementos), revisões frequentes, colaboração entre os membros e orientação com objectos de comportamento bem definido. Para que todas estas características sejam cumpridas [Schwaber, 2009] definiu três pilares:

1. **Transparência:** é necessário garantir que todos os aspectos relativos ao sucesso do trabalho estejam visíveis e sejam conhecidos. Com isto pretende-se que, quando alguém inspecciona o trabalho, o que está visível corresponde à realidade, ou seja, se algo está pronto tem que estar realmente pronto
2. **Inspeção:** é realizada com o objectivo de verificar se existe algo que possa prejudicar os resultados da equipa
3. **Adaptação:** caso seja detectado algum factor que possa prejudicar o trabalho da equipa, o mesmo tem que ser combatido de imediato para o resultado não ser insatisfatório

O *Scrum* é constituído por três elementos [Schwaber, 2009]: **Equipas Scrum**, **Time-Boxes** e **Artefactos**.

5.1.1 Equipas Scrum

Numa equipa de *Scrum* existem três papéis: o **Scrum Master**, o **Product Owner** e a **Equipa**. A função do *Scrum Master* consiste em garantir que o *Time Scrum* está a aderir e a respeitar as regras do *Scrum*. Tem, também, como função remover os impedimentos, caso estes surjam, melhorar a produtividade da Equipa, promover a comunicação e impedir que factores externos atrapalhem a produtividade. O *Product Owner* (*P.O.*) representa

o cliente dentro da equipa *Scrum*. Tem como principal responsabilidade acompanhar o *Return of Investment*, saber quais são as necessidades e desejos do cliente, organizar o *Product Backlog*, priorizando-o, e aceitar ou rejeitar os resultados apresentados pela Equipa. Por último, a Equipa é quem realmente desenvolve. As suas responsabilidades consistem em realizar todos os itens existentes no *Sprint Backlog*, ter capacidade de auto-gestão, aplicar boas práticas de engenharia e apresentar o resultado do seu trabalho no final do *Sprint*.

Cada um dos elementos da *Equipa Scrum* é apelidado de **porco**, enquanto que qualquer outro interveniente é apelidado de **galinha**. Os *porcos* estão comprometidos com o produto ao passo que as *galinhas* apenas estão envolvidas. As *galinhas* não podem dizer aos *porcos* como estes devem proceder relativamente ao seu trabalho.

5.1.2 Time-Boxes

Time-Boxes são eventos com duração fixa. Esses eventos são:

Reunião de Planeamento da Versão para Entrega onde são definidos o plano e os objectivos de maneira que a *Equipa Scrum* e o resto da organização compreenda. O plano da versão para entrega define a meta da versão, quais são as prioridades do *Product Backlog* e qual é o conteúdo da versão. É também estabelecido o prazo de entrega e o custo da mesma. Uma das vantagens ao se utilizar *Scrum* é o facto do produto final de entrega ser desenvolvido iterativamente, onde em cada iteração é desenvolvido um pedaço do mesmo. Quando o produto já possuir incrementos suficientes, que lhe potenciem valor, o produto é entregue.

Sprint consiste numa iteração com duração fixa. Um *Sprint* é formado pelo *Planeamento do Sprint*, desenvolvimento, *Revisão do Sprint* e, por último, pela *Retrospectiva do Sprint*. É obrigação do *Scrum Master* garantir que nada importuna o decorrer normal do *Sprint*. Os *Sprints* são eventos cíclicos que ocorrem um após o outro, sem tempo de pausa entre os mesmos.

Reunião de Planeamento do Sprint, que corresponde ao início de cada iteração. Essa reunião é separada em duas partes. A primeira consiste na apresentação, por parte do *P.O.*, dos itens mais prioritários do *Product Backlog*. Nessa apresentação é definida qual a funcionalidade a ser desenvolvida durante o *Sprint*. Tendo em conta os itens existente no *Product Backlog* é função da Equipa seleccionar quais consideram ser possíveis de realizar durante o *Sprint*. Na segunda parte do planeamento a Equipa discute como irá proceder para transformar os itens seleccionados num novo incremento ao produto. Se durante essa reunião se verificar que o trabalho necessário ultrapassa ou fica aquém da capacidade da equipa é possível renegociar a selecção do *Product Backlog* com o *P.O.*

Revisão do Sprint, consiste na apresentação ao *P.O.* das funcionalidades desenvolvidas durante o *Sprint*. Durante essa apresentação a Equipa pode ser questionada sobre algum

pormenor relativo ao desenvolvimento e também dos problemas que existiram e como os mesmos foram ultrapassados. Consoante o que foi apresentado, o *P.O.* aceita ou rejeita os desenvolvimentos efectuados pela Equipa e, conseqüentemente, o resultado do *Sprint*.

Retrospectiva do Sprint, corresponde à última etapa do *Sprint*. Aqui, encorajada pelo *Scrum Master*, a Equipa deve analisar como decorreu o *Sprint* e se o mesmo cumpriu as regras que o caracterizam. Tem como principal objectivo salientar o que correu bem e quais as dificuldades encontradas. Se forem detectados aspectos negativos é necessário propor formas de os combater caso eles voltem a surgir em *Sprints* posteriores.

Reunião Diária. Esta reunião deve ocorrer todos os dias durante o *Sprint* no mesmo horário e no mesmo local. Durante essa reunião cada elemento da Equipa deve responder a três perguntas:

1. *O que foi feito?*
2. *O que se vai fazer?*
3. *Existe algum impedimento?*

Através destas perguntas, e as respectivas respostas, a equipa fica com uma melhor visão do estado actual do *Sprint*. Essa reunião deve ter uma duração de 15 minutos e ser feita de pé. Com isto pretende-se que a Equipa discuta, realmente, os pontos referentes ao *Sprint* e não tenha oportunidade para divagar por outros assuntos externos ao mesmo. Caso algum membro identifique algum impedimento é dever da Equipa encontrar solução para o resolver.

5.1.3 Artefactos

Os artefactos existentes no *Scrum* correspondem ao **Product Backlog**, **Sprint Backlog**, **Burndown da Versão** e **Burndown do Sprint**.

O **Product Backlog** contém todos os requisitos do produto e faz parte das responsabilidades do *P.O.*, que tem de o manter disponível e definir as prioridades de cada item existente. Numa primeira fase apenas contém os requisitos iniciais detectados e desejados para o produto, no entanto a sua evolução é constante. Essa evolução está relacionada com as alterações que se detectem no próprio produto e no contexto da sua utilização. Enquanto o produto existir também o *Product Backlog* existe. Com isto pretende-se que seja sempre possível alterar, adicionar ou modificar itens de modo a garantir o sucesso do produto. Este artefacto está ordenado por prioridade. Quanto mais alta for a prioridade de um item mais urgente é o seu desenvolvimento. Para além da prioridade, cada item possui, também, uma descrição detalhada sobre o que se pretende fazer. É importante que cada item esteja o mais detalhado possível, pois essa informação permite a Equipa estimar o mais correctamente possível cada item.

O **Burndown da Versão** é um gráfico onde se registam as somas das estimativas de esforço restante dos itens do *Product Backlog* no decorrer de topo o processo. Estas estimativas são calculadas durante a *Reunião de Planeamento da Versão para Entrega* e quando novos itens forem adicionados. Essa estimativa é feita unicamente pela Equipa, sendo sua responsabilidade.

O **Sprint Backlog** é formado pelos itens do *Product Backlog* seleccionados para serem realizados durante um *Sprint* com a finalidade de entregar um incremento do produto. Esses itens devem ser decompostos em tarefas para que assim a Equipa possua uma melhor noção do que é necessário realizar. Este *Backlog* deverá conter todas as tarefas assim como o tempo estimado e realmente gasto na realização das mesmas. Pretende-se fornecer, através do *Sprint Backlog* uma visão detalhada sobre a evolução do *Sprint*. Apenas a Equipa pode alterar o *Sprint Backlog* e este é da sua total responsabilidade.

O **Burndown da Versão** corresponde ao gráfico onde pode ser visualizado quanto tempo já foi gasto e quanto falta gastar para se concluírem todas as tarefas existentes no *Sprint Backlog*. Esse gráfico é construído diariamente através da soma de todas as expectativas restantes para o *Sprint*. Com isto pretende-se que a Equipa tenha uma noção como está a correr o *Sprint* e quanto esforço é ainda necessário para se concluir.

5.2 Metodologia *Scrum* no desenvolvimento do Hcis Mobile

Como referido anteriormente, adoptámos a metodologia *Scrum* na realização deste projecto. No entanto o *Scrum* utilizado não corresponde ao *Scrum puro* descrito em 5.1. Apesar de existir uma Equipa de desenvolvimento, um *Product Owner* e um *Scrum Master* estes não se encontravam dentro dos padrões exigidos pelo *Scrum*. A *Equipa Scrum* era apenas constituída por dois elementos, enquanto o recomendável é que seja formada por seis a dez elementos. Como o número de elementos era reduzido foi necessária uma adaptação, donde um só elemento realizou as tarefas de *Scrum Master* e de *Product Owner*, enquanto que o outro elemento formava a Equipa de desenvolvimento. Apesar destas restrições quase todas as outras práticas *Scrum* foram adoptadas. Foi definido um *Product Backlog* no início do projecto, e em cada *Sprint* foram delineadas tarefas para a realização do mesmo. Para cada *Sprint*, existiu sempre um *Planeamento de Sprint* e uma *Revisão de Sprint*, no entanto nunca foi realizada uma *Retrospectiva de Sprint*. Apesar de acontecerem com bastante frequência, as *Reuniões Diárias* onde era avaliado a evolução do *Sprint* raramente foram presenciais, tendo-mos recorrido essencialmente a meios electrónicos, de onde destacamos a comunicação via *email* e a utilização da plataforma **Redmine** através da qual era efectuada a gestão do *Product Backlog* e do *Sprint Backlog*.

5.2.1 Fases do Projecto

O desenvolvimento do projecto foi constituído por seis *Sprints*, cuja duração oscilou entre duas e três semanas, à excepção do *Sprint* 6 que durou quatro semanas. Cada *Sprint* possuiu uma funcionalidade específica, dentro da qual foram definidos os itens que deveriam estar finalizados até ao término do mesmo. Em seguida pode-se visualizar o que foi realizado em cada *Sprint*:

Sprint 1: Carregamento do Plano de Cuidados (Diagnósticos de Enfermagem)

- Construir a Base de Conhecimento (Diagnósticos de Enfermagem com Objectos Clínicos) no dispositivo móvel
- Carregar dados da Base de Conhecimento no dispositivo móvel
- Construir simulador de Web Service HP-HCIS Enfermagem
- Carregar dados no simulador de Web Service HP-HCIS Enfermagem
- Construir BD do plano de cuidados no dispositivo móvel
- Fazer chamada ao Web Service HP-HCIS Enfermagem para carregamento dos Planos de Cuidados dos doentes
- Carregamento dos dados do plano de cuidados na BD móvel

Sprint 2: Carregamento do Plano de Cuidados (Actividades Interdependentes, Diagnósticas e Não Diagnósticas)

- Actualização simulador Web Service
- Carregar dados simulador Web Service
- Alteração da BD do plano de cuidados
- Hcis Mobile - Alterações chamada Web Service e carregamento da Base de Dados do Plano de Cuidados
- Hcis Mobile - Analisar interface com utilizador
- Hcis Mobile - GUI - Listar doentes
- Colocar *sources* no repositório

Sprint 3: Gestão *offline* das acções de enfermagem no Plano de Trabalho (sem Objectos Clínicos)

- *Refactoring* do código do Hcis Mobile
- Criar funcionalidade de configuração do Hcis Mobile
- Colocar projecto soapUI no repositório
- Gestão de Acções (sem Objectos Clínicos) - Actualizar BD do Plano de Cuidados
- Gestão de Acções (sem Objectos Clínicos) - Criar GUI
- *Refactoring* Hcis Mobile GUI

Sprint 4: Gestão *offline* das Acções de enfermagem no Plano de Trabalho (Objectos Clínicos)

- Criar BD Móvel para registo dos Objectos Clínicos
- Aplicar *refactoring* dos nomes das tabelas da Base Conhecimento
- Efectuar novo carregamento da Base de Conhecimento
- Efectuar alterações ao *layout* do Hcis Mobile
- Hcis Mobile - Criar funcionalidade de registo de Anotações
- Hcis Mobile - Criar funcionalidade de registo de Transacções

Sprint 5: Envio das confirmações registadas no Plano de Trabalho

- Apresentar detalhes da Acção já confirmada ou anulada
- Efectuar pequenas alterações no registo de Objectos Clínicos
- Alterar Página Inicial
- Envio das Confirmações e Registos - Criar página com a lista de doentes
- Envio das Confirmações e Registos - Envio via *Web Service*

Sprint 6: Autenticação via Web Service no HP-HCIS

- Autenticação - Criar Página de Login para o Hcis Mobile
- Autenticação - Efectuar o Login no Hcis Mobile
- Autenticação - Implementação da funcionalidade Logout para o Hcis Mobile
- *Refactoring* Hcis Mobile GUI

Capítulo 6

Hcis Mobile

Como referido no Capítulo 4, o HP-HCIS possui vários módulos aplicativos. A aplicação Hcis Mobile fornece uma solução móvel que incide unicamente no módulo de Enfermagem, mais concretamente no registo de intervenções de Enfermagem.

A escolha recaiu sobre este módulo, para ser o pioneiro na implementação de uma solução de mobilidade do HP-HCIS, devido à constatação de uma necessidade recorrente na prestação dos cuidados de saúde executados pelos enfermeiros. Para serem prestados cuidados de qualidade é essencial que o enfermeiro possua todo o conhecimento necessário sobre o Plano de Trabalho no instante em que está a executar esses mesmos cuidados. No entanto é importante que essa informação seja o mais concreta e explícita possível.

Actualmente, o enfermeiro, pode optar por duas vias: ou leva consigo todos os Planos de Trabalho relativos aos doentes ou está constantemente a voltar ao seu posto fixo onde pode consultar esses mesmos Planos de Trabalho. Nenhuma dessas opções potencia a qualidade de prestação de cuidados por parte do enfermeiro. Se na primeira é obrigado a transportar consigo um conjunto enorme de processos, a segunda restringe-lhe o tempo que tem para efectuar a sua função. Outro factor importante está relacionado com a importância de se desejar manter o mais actualizado possível o Plano de Trabalho de cada doente. Com esta solução é possível submeter no sistema a informação sobre a realização das acções de enfermagem no instante em que elas são efectuadas¹. Com isto impede-se, não só que sejam introduzidos dados incorrectos ou tardiamente, mas também que os outros prestadores de cuidados, sejam eles outros enfermeiros ou os médicos responsáveis, saibam em *tempo real* o estado clínico do doente.

Neste capítulo iremos abordar as características da aplicação Hcis Mobile, tanto a nível

¹Caso exista ligação à rede o enfermeiro pode submeter a informação logo após a ter registado.

de implementação como de funcionalidades.

6.1 Comunicação Web Service

A comunicação entre as aplicações Hcis Mobile e HP-HCIS é realizada através da tecnologia **WebService**. Optou-se por este método devido a todas as potencialidades e facilidades de integração que esta tecnologia potencia na área de comunicação entre plataformas distintas.

6.1.1 Web Services: O que são?

Com o *boom* da Internet a integração e a interoperabilidade entre sistemas e plataformas diferentes tornou-se num dos principais desafios na área da Tecnologia da Informação [Zanuz et al., 2007]. Foi devido a esta necessidade que surgiram os Web Services, cujo principal objectivo passa pela integração de sistemas díspares, através da Internet, utilizando protocolos padrão que permitem a independência das plataformas e das linguagens de programação nas quais esses sistemas foram desenvolvidos. Através dos Web Services torna-se possível a qualquer aplicação cliente aceder a funcionalidades disponibilizadas por outra aplicação num servidor remoto sem ser necessário preocupações de compatibilidade entre as mesmas. O conceito Web Service pode ser definido de várias formas, uma forma mais genérica existente na literatura [Silva, 2002] é: *"uma interface que descreve uma colecção de operações que são acessíveis pela rede através de um mecanismo de mensagens XML standard"*.

Apesar desta tecnologia poder parecer algo completamente revolucionário, nada possui de novidade. A possibilidade de aceder a métodos, alojados num servidor a partir de outros clientes, através de chamadas remotas já foi amplamente estudada e para o qual foram desenvolvidas novas arquitecturas de comunicação. Dentre essas arquitecturas salientam-se a arquitectura **CORBA**² e o padrão **RMI**³ da Sun [Neto, 2005]. Uma das principais causas apontadas para a não padronização destas arquitecturas passou pela necessidade de ser necessário reescrever código e adaptação à forma de trabalho das novas arquitecturas. Outro aspecto pelo qual o RMI não foi amplamente aceite deveu-se ao facto de ser apenas possível utilizar em chamadas entre objectos Java, o que impedia que uma aplicação desenvolvida em **C**, por exemplo, pudesse utilizar os métodos disponibilizados [Freire, 2002]. Sendo os Web Service altamente modulares, auto-descritivos, com acesso a partir de um **URL**, não específicos de plataformas ou de linguagens de programação e que possibilitam as aplicações interagirem entre si sem ser necessário qualquer intervenção, por parte do utilizador, são neste momento a principal solução para os problemas de integração de

²Common Object Request Broker Architecture: um protocolo standard definido pelo **Object management Group** que permite a troca de informação entre aplicações escritas em diferentes linguagens e em plataformas diferentes.

³Remote Method Invocation: uma interface Java que permite a utilização de métodos de objectos remotos através de chamadas no estilo RPC (*Remote Procedure Call*).

aplicações [Lopes and Ramalho, 2004]. A possibilidade de re-utilizar código já existente, independentemente da linguagem em que tenha sido escrito, e transformá-lo numa funcionalidade que pode ser utilizado por outra linguagem é um dos factores que evidencia as potencialidades dos Web Services. Como quase todos os protocolos que constituem esta arquitectura são descritos utilizando linguagem XML, a implementação dos protocolos da arquitectura Web Service nas diferentes plataformas encarrega-se de efectuar a conversão transparente dos dados, permitindo que a comunicação entre programas clientes e servidores seja possível, mesmo nos casos em que a base de desenvolvimento de ambos seja completamente distinta.

Como já foi referido, os WebServices utilizam protocolos *standard* aceites e adoptados pelo mercado. Desses protocolos destacam-se o **HTTP**, **SOAP**, **WSDL** e **UDDI**. Uma ilustração da camada de protocolos pode ser visualizada na Fig. 6.1 [Becker et al., 2001]. Iremos, em seguida, fornecer uma explicação mais detalhada do que são estes protocolos assim como sobre a linguagem XML.

Outros padrões ainda não definidos
Universal Description Discovery Integration (UDDI)
Web Services Description Language (WSDL)
Simple Object Access Protocol (SOAP)
Extensible Markup Language (XML)
Protocolos Internet (TCP/IP , HTTP, SMTP)

Figura 6.1: Camada dos protocolos Web Service

Linguagem XML

A linguagem XML é um padrão reconhecido pelo **W3C**⁴ que tem sido desenvolvida e especificada pelo **XML Working Group**. Surgiu em 1996 e é considerado padrão desde Fevereiro de 1998 [Becker et al., 2001].

O XML, estruturalmente, é bastante semelhante ao HTML, no entanto, enquanto este último serve, essencialmente, para exibição, a principal função do XML consiste em descrever e estruturar informação. Para tal ser possível o XML faz uso de *marcadores* não pré-definidos, pois assim possibilita-se criar *tags* consoante as necessidades específicas de cada projecto.

Um ficheiro XML é nada mais que um ficheiro de texto formado por *marcadores* aninhados⁵ e delimitados. Um ficheiro XML é composto por *elementos*. Um elemento consiste numa marcação de início e uma marcação de fim, que delimitam o conteúdo desse elemento. Todo o ficheiro XML dever possuir, no mínimo, um elemento *raíz*. Os elementos relacionam-se com outros elementos, podendo inclusive conter outros elementos como conteúdo, formando assim uma estrutura de árvore entre os elementos. Para além de outro elemento, o

⁴World Wild Web Consortium: uma comunidade internacional que desenvolve standards abertos de maneira a garantir o crescimento a longo prazo da Web [W3C, 2012].

⁵Elementos que estão contidos dentro de outros elementos.

conteúdo de um elemento pode ser também texto, um conjunto de texto e outros elementos, ou até ser vazio (o elemento pode não possuir conteúdo de qualquer tipo). Para além do conteúdo os elementos podem conter atributos que fornecem informação extra sobre o elemento.

É indispensável que os documentos XML estejam correctamente formados, pois se não o estiverem poderão não ser bem interpretados pelas aplicações, tanto cliente como servidor, e conseqüentemente não ser possível utilizá-los para os fins a que se destinam. Para validar os documentos existem os *DTD*'s⁶. Estes documentos definem quais são os elementos permitidos em determinado XML, assim como a estrutura do próprio documento.

HTTP

Os Web Services assentam substancialmente em transacções efectuadas na Internet, portanto torna-se lógico que utilize as tecnologias e standards da mesma. Entre essas tecnologias encontram-se os protocolos de transporte/comunicação, dos quais podemos evidenciar os protocolos **TCP/IP**, **HTTP** e **SMTP**. Outros protocolos podem, também, garantir a troca de mensagens entre os Web Services, tais como o **FTP**, **MQ** ou **IIOP**, no entanto, devido às suas características, nomeadamente a sua ubiquidade, o HTTP tornou-se standard *de facto* na comunicação entre Web Services [Silva, 2002].

O HTTP⁷ é um tipo de protocolo de transporte bastante utilizado na Internet devido à sua objectividade e rapidez [Schepke and Viana, 2007], o que torna o seu uso ideal em ambientes distribuídos e cooperativos. Este protocolo baseia-se no paradigma de comunicação de pedido/resposta. A aplicação cliente estabelece uma ligação com o servidor enviando o pedido que, após analisar esse mesmo pedido, envia a resposta. A conexão deve ser estabelecida antes de cada pedido de cliente e encerrada após a obtenção da resposta. Inicialmente o protocolo HTTP servia unicamente para transferência de ficheiros tendo sido remodelado para satisfazer funcionalidades de *RPC*'s [Schepke and Viana, 2007].

Remote Procedure Call's são um conjunto de regras que especificam formas de codificar e decodificar os dados trocados entre duas aplicações, normalmente utilizado através de uma rede. Realizar uma invocação a um método noutra aplicação necessita que o sistema a chamar, a codificação dos parâmetros e a forma de receber e decodificar a resposta, fazendo com que seja possível utilizar num sistema específico, estejam correctamente definidos.

⁶Document Type Definition.

⁷Hypertext Transfer Protocol.

SOAP

SOAP⁸ tem origem num mecanismo de RPC proposto por Dave Winer, em 1998, baseado em XML [Becker et al., 2001] sendo actualmente um padrão recomendado pelo W3C. Inúmeras empresas adoptaram este protocolo, dentre as quais se destacam a HP, a Microsoft e a IBM.

Este protocolo faz uso de HTTP para comunicar entre as aplicações. Um dos principais factores que levou o SOAP a utilizar o protocolo HTTP prende-se com o facto deste ser suportado por todos os servidores Web. Para além disso, ao utilizar HTTP, as restrições de segurança/*firewalls* existentes relativamente aos sistemas convencionais de RPC, como por exemplo a arquitectura CORBA, não se verificam. A estrutura de uma mensagem SOAP encontra-se é formada por [Becker et al., 2001]:

- **SOAP Envelope:** onde se encontram definidos os *namespaces* utilizados pela mensagem, incluindo **xmlns:SOAP-ENV**, **xmlns:xsi** (*Xml Schema for Instance*) e **xmlns:xsd** (*Xml Schema for Datatype*). Para além dos *namespaces* inclui também o conteúdo da mensagem.
- **SOAP Header:** este elemento é opcional. Serve para armazenar informação relacionada com a autenticação, transacção e contabilização.
- **SOAP Body:** aqui encontram-se os dados sobre os métodos e parâmetros do pedido, ou a resposta retornada. Caso seja utilizado num contexto RPC este elemento contém um elemento único onde está o nome do método chamado, os seus parâmetros e qual o serviço a ser utilizado.

WSDL

O padrão WSDL⁹ é uma linguagem baseada em XML que permite descrever os Web Services. É visto como a API de Web Service, podendo ser comparado com uma interface Java no protocolo RMI.

O WSDL possibilita que a aplicação cliente saiba quais são as funcionalidade disponibilizadas pelo serviço, tais como os seus métodos, os parâmetros necessários e o formato das mensagens enviadas e de retorno. Para além disso possui também informação sobre onde o serviço está alojado e como se pode invocá-lo.

Sendo escrito através da linguagem XML, este documento é dividido em secções constituídas por vários elementos, sendo o principal o elemento **<definitions>**. Este elemento é a raiz de qualquer documento WSDL, fornece informação respeitantes a um ou mais serviços, e possui outros elementos dentro de si, dos quais se destacam:

⁸Simple Object Access Protocol.

⁹Web Service Description Language.

- **<message>** e **<portType>**: definições dos métodos disponibilizados pelo serviço em questão.
- **<type>**: caso seja necessário, permite definir tipos de dados estruturados.
- **<binding>**: como invocar os métodos.
- **<service>**: onde o serviço se encontra alojado.

Todos os elementos possuem o atributo *name*, o seu identificador. Se se utilizar o atributo *targetNamespace* no elemento **<definitions>**, que deve ser feito por uma questão de boas práticas, todos os nomes dos elementos são definidos no *namespace*¹⁰ de destino.

Segundo a literatura [Zanuz et al., 2007], a descrição de um serviço WSDL pode-se ser separada em duas partes, a *descrição abstracta* e a *descrição concreta*. Através da *descrição abstracta* são fornecidas as características da interface, desassociando-se das tecnologias utilizadas para alojar ou transmitir as mensagens. Essa informação encontra-se na *descrição concreta*, que define quais os protocolos de transporte e onde o serviço se encontra alojado. Com isto pretende-se manter a integridade do serviço mesmo que sejam efectuadas modificações na plataforma que o sustenta.

UDDI

Este protocolo nasceu da necessidade de se conseguir armazenar e localizar Web Services na rede. O UDDI¹¹ encontra-se em constante desenvolvimento pelo consórcio UDDI.ORG¹². Este consórcio é formado por empresas de renome mundial, entre as quais a Microsoft, a IBM e a Ariba [Becker et al., 2001].

O UDDI é formado por um conjunto de registos que fornecem informações respeitantes a uma organização ou entidade. Estes registos podem ser usados por qualquer entidade que pretenda disponibilizar informação sobre o seu negócio ou por quem queira aceder à informação disponibilizada. Com isto pretende-se acelerar a interoperabilidade e a utilização de Web Services para troca de informação. Para se poder utilizar eficazmente o UDDI, este possui três funções principais:

1. **Publicação:** cada organização pode publicar os seus serviços
2. **Descoberta:** possibilita a descoberta de um serviço específico por parte do cliente ou invocador do serviço

¹⁰Os *namespaces* são espaços para nomes, definidos no interior dos documentos XML. No caso dos WSDL's, os *namespaces* são usados para maximizar a taxa de reutilização dos componentes de um documento WSDL, utilizando-se de atributos para fazer referência a outros elementos, seja dentro ou fora do documento.

¹¹Universal Description Discovery and Integration.

¹²<http://www.uddi.org/>.

3. **Ligação:** permite que se estabeleça uma ligação com um determinado serviço e que se interactue com o mesmo

A estrutura de um registo UDDI é semelhante à estrutura de uma lista telefónica [Silva, 2002]: possui *páginas brancas*, onde se pode encontrar informação relativas à organização, como o nome, o contacto e identificadores, *páginas amarelas*, que são compostas por um índice de serviços e produtos, e *páginas verdes*, onde se pode encontrar informação relativa a transacções, descrições do serviço e invocação de aplicações.

6.1.2 Web Service HP-HCIS Enfermagem

O Web Service HP-HCIS Enfermagem possibilita a comunicação entre o Hcis Mobile e a aplicação HP-HCIS. Este Web Service foi criado no âmbito deste projecto e não se encontra por enquanto integrado no HP-HCIS, pelo que a sua utilização foi simulada recorrendo ao soapUI¹³. Posteriormente iremos fornecer uma explicação mais detalhada da implementação do projecto soapUI.

WSDL Hcis Mobile

O WSDL, que está disponível na íntegra no Anexo A, foi criado para possibilitar a comunicação entre o Hcis Mobile e o HP-HCIS. Como já foi referido os WSDL's possuem, como raíz, o elemento <definitions>. Encontram-se definidos neste elemento os atributos *targetNamespace* ("http://www.hphis.com/HcisNurseryPT/") e *name* ("HcisNurseryPT"). Para além destes atributos estão também definidos alguns prefixos para serem utilizados no documento, dos quais destacamos o *xmlns:hcis-nursery-pt* = "http://www.hphis.com/HcisNurseryPT/".

Dentro deste elemento encontram-se definidos os elementos <service>, <binding>, <portType>, <types> e <message>. No <service> podemos encontrar a informação sobre a localização, na rede, onde se encontra o serviço que se pretende utilizar, assim como qual o elemento <binding> respectivo deste serviço:

```
<wsdl:service name="HcisNurseryPT">
  <wsdl:port binding="hcis-nursery-pt:HcisNurseryPTSOAP" name="
    HcisNurseryPTSOAP">
    <soap:address location="http://www.hphis.com/" />
  </wsdl:port>
</wsdl:service>
```

O elemento <binding> indica como invocar os métodos. Os métodos disponíveis são:

- **Login:** utilizado para validar o Login na aplicação.

¹³Software da que permite simular um servidor de Web Service.

- **getPatientCarePlanByNHC**: permite obter o Plano de Cuidados do doente consoante o NHC fornecido.
- **getPatientCarePlanByNurseryUnit**: permite obter os Planos de Cuidados dos doentes consoante o quarto fornecido.
- **getPatientCarePlanByNurseryRoom**: permite obter os Planos de Cuidados dos doentes consoante a Unidade de Enfermaria.
- **uploadPatientCarePlan**: utilizado para enviar os dados registados.
- **getHospitals**: utilizado para obter os hospitais aos quais o utilizador se pode conectar.

Para cada método existe um elemento no `<binding>`. Todos esses elementos são praticamente iguais, apenas altera o valor do atributo `soapAction` consoante o método ao qual está associado. O valor do atributo `type` permite saber qual o elemento `<portType>` que possui as definições dos métodos disponíveis. Essas definições constam da indicação das mensagens de *input* e *output* para cada método. É possível visualizar em seguida, um excerto destes elementos¹⁴:

```
<wsdl:binding name="HcisNurseryPTSOAP" type="hcis-nursery-pt:HcisNurseryPT">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="Login">
    <soap:operation soapAction="http://www.hphis.com/HcisNurseryPT/Login"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  ...
</wsdl:binding>

<wsdl:portType name="HcisNurseryPT">
  <wsdl:operation name="Login">
    <wsdl:input message="hcis-nursery-pt:LoginRequest"/>
    <wsdl:output message="hcis-nursery-pt:LoginResponse"/>
  </wsdl:operation>
  ...
</wsdl:portType>
```

Para cada operação existem dois elementos `<message>`, uma mensagem de *input* e outra de *output*. A tabela 6.1 indica a correspondência entre cada método e as mensagens respectivas.

¹⁴Como já foi referido, no Anexo A pode ser visualizado o WSDL completo.

Método	Tipo	Mensagem
Login	Input	LoginRequest
	Output	LoginResponse
getPatientCarePlanByNHC	Input	getPatientCarePlanByNHCRequest
	Output	getPatientCarePlanByNHCResponse
getPatientCarePlanByNurseryUnit	Input	getPatientCarePlanByNurseryUnitRequest
	Output	getPatientCarePlanByNurseryUnitResponse
getPatientCarePlanByNurseryRoom	Input	getPatientCarePlanByNurseryRoomRequest
	Output	getPatientCarePlanByNurseryRoomResponse
uploadPatientCarePlan	Input	uploadPatientCarePlanRequest
	Output	uploadPatientCarePlanResponse
getHospitals	Input	getHospitalsRequest
	Output	getHospitalsResponse

Tabela 6.1: Mensagens correspondentes a cada método do Web Service

Por fim, existe ainda o elemento <message>. Este elemento possui o *schema* de todos os tipos de dados estruturados permitidos neste Web Service. Devido ao elevado número de tipos distintos existentes, que podem ser consultados no Anexo A e julgamos serem de fácil compreensão, não irão ser detalhados neste capítulo.

soapUI

Como este Web Service não está integrado no HP-HCIS, foi necessário recorrermos ao soapUI para simularmos a interação com o HP-HCIS. O soapUI é uma ferramenta *open source* que tem como principal função testar o funcionamento de Web Services. A sua primeira versão foi lançada em 2005 [soapUI.org, 2012], estando actualmente na versão 4.0.1¹⁵. Inicialmente era um produto da Eviware, empresa que foi adquirida em 2011 pela SmartBearSoftware [Growth, 2012].

Esta ferramenta é desenvolvida em Java e utiliza a interface gráfica Java Swing [Wikipédia, 2012c], o que possibilita o seu uso em várias plataformas. Tem como principal objectivo facilitar o teste, e também a implementação, de Web Services recorrendo a uma interface gráfica bastante simplificada. As principais características desta aplicação são [Manso, 2010]:

- **Suporte de múltiplos protocolos:** SOAP, REST, HTTP, JMS, AMF, JDBC.
- **Suporte de *standards*:** WS-I Integration, WS-Security, WS-Security, SOAP 1.1 e 1.2.
- **Simulação de Serviço** (ou *MockServices*): permite simular interfaces que ainda não estão disponíveis, a partir de um WSDL.

¹⁵A versão 4.0.1 é a versão estável à data de escrita deste documento, existindo já a versão Beta 4.5.0.

- **Testes Funcionais:** esta é a funcionalidade base que deu origem ao desenvolvimento do soapUI. Permite verificar se serviço está a funcionar correctamente, através de pedidos ao servidor e validação das respostas.

Para o desenvolvimento do Hcis Mobile, o soapUI foi utilizado como o *simulador do serviço*, portanto a funcionalidade mais explorada desta aplicação foram os Mock Services. A sua utilização foi bastante útil pois, ao criar o projecto, é possível definir a criação de respostas padrão que respeitam a estrutura do WSDL. Para além disso, é possível configurar, através de *scripts* **Groovy**¹⁶, várias respostas possíveis para determinado método. Consoante os parâmetros enviados na mensagem de entrada retorna uma mensagem específica.

Mecânica do Web Service Hcis Mobile

Como referido anteriormente, existem seis métodos possíveis de serem invocados através do Web Service. Para cada um deles está definida uma mensagem de *input*, o pedido, e uma mensagem de *output*, a resposta. Iremos então proceder à explicação do que compõe cada mensagem, organizando-as pelos métodos a que pertencem:

1. **getHospitals**

- (a) **getHospitalsRequest:** possui um elemento do tipo *String*, que não possui qualquer conteúdo. Serve unicamente para fazer o pedido.
- (b) **getHospitalsResponse:** é composta por um elemento do tipo *HospitalListType* que é constituído por uma lista de *Hospital*, sendo este último elemento do tipo *String* que identifica cada um dos hospitais aos quais o utilizador se pode conectar.

2. **Login**

- (a) **LoginRequest:** cada mensagem possui um elemento do tipo *AuthenticationType*, que é composto por cada um dos elementos necessários para o Login ser efectuado - *login*, *password* e *hospital*.
- (b) **LoginResponse:** a mensagem possui o elemento *LoginResponseStatusType*, cujo valor indica se a autenticação foi correctamente validada, a partir de um conjunto de valores pré-definidos.

3. **getPatientCarePlanByNHC**

- (a) **getPatientCarePlanByNHCRequest:** possui um elemento do tipo *AuthenticationType*, para ser realizada a autenticação do utilizador que efectua o pedido, e um elemento do tipo *String*, cujo valor indica o *nhc* do doente desejado.

¹⁶É uma linguagem de programação orientada a objectos que pode ser utilizada como uma linguagem de *scripting* na plataforma Java. É linguagem dinâmica que possui semelhanças com Python, Ruby e Perl [Wikipédia, 2012a].

- (b) **getPatientCarePlanByNHCRresponse**: esta mensagem é composta pelos elementos do tipo *PatientCarePlanListType* e *GetPatientCarePlanResponseStatusType*. Enquanto o último serve apenas para indicar se houve algum problema ao tentar obter os Planos de Cuidados, como, por exemplo, a autenticação ser inválida, o primeiro corresponde a uma lista¹⁷ constituída por elementos do tipo *PatientCarePlanType*. Este elemento é formado pelos elementos *PatientCarePlanInfo* (do tipo *PatientCarePlanInfoType*), *NurseryDiagnosticList* (*NurseryDiagnosticListType*), *InterdependentActivityList* (*InterdependentActivityListType*), *DiagnosticActivityList* (*DiagnosticActivityListType*) e *NonDiagnosticActivityList* (*NonDiagnosticActivityListType*). No primeiro elemento podemos encontrar informação sobre o doente, *Patient* do tipo *PatientType*, a Unidade de Enfermaria onde se encontra internado, *NurseryUnit* do tipo *NurseryUnitType*, os utilizador que criaram *CreationUser* e modificaram *ModificationUser* o Plano de Cuidados, do tipo *UserType*, e as datas de início *InitialDate* de modificação *ModificationDate* e de fim *FinalDate* se tiver sido planeada, todas do tipo *xsd.dateTime*. O tipo *NurseryDiagnosticList* corresponde a uma lista de Diagnósticos de Enfermagem, onde cada Diagnóstico, *NurseryDiagnosticType*, possui uma lista de Intervenções (*NurseryInterventionType*) que por sua vez possui uma lista de Acções (*NurseryActionType*) Por sua vez os elementos do tipo *InterdependentActivityListType*, *DiagnosticActivityListType* e *NonDiagnosticActivityListType* correspondem a listas de *NurseryActionType*'s. Os elementos *NurseryActionType*'s possuem toda a informação sobre cada uma das Acções que são necessárias prestar aos doentes.

4. getPatientCarePlanByNurseryUnit

- (a) **getPatientCarePlanByNurseryUnitRequest**: semelhante à mensagem *getPatientCarePlanByNHCRrequest*, apenas difere no elemento do tipo *String*, que neste caso indica a Unidade de Enfermaria desejada.
- (b) **getPatientCarePlanByNurseryUnitResponse**: semelhante à mensagem *getPatientCarePlanByNHCRresponse*, difere apenas no facto de poderem existir vários elementos do tipo *PatientCarePlanType*.

5. getPatientCarePlanByRoom

- (a) **getPatientCarePlanByRoomRequest**: semelhante à mensagem *getPatientCarePlanByNHCRrequest*, apenas difere no elemento do tipo *String*, que neste caso indica o quarto desejado.
- (b) **getPatientCarePlanByRoomResponse**: semelhante à mensagem *getPatientCarePlanByNHCRresponse*, difere apenas no facto de poderem existir vários elementos do tipo *PatientCarePlanType*.

6. uploadPatientCarePlan

¹⁷Neste caso específico, carregamento por *NHC*, esta lista possui no máximo apenas um elemento.

- (a) **uploadPatientCarePlanRequest**: esta mensagem contém dois elementos, sendo o primeiro relativo à autenticação do utilizador. *AuthenticationType*, já evidenciado na análise de mensagens anteriores, e o segundo relativo ao Plano de Cuidados que se deseja enviar *UploadPatientCarePlanType*. Este último elemento é composto pelo elemento *Patient*, que indica o doente ao qual pertence o Plano de Cuidados que se deseja enviar¹⁸, e pelo elemento *NurseryActionConfirmations*, que corresponde à lista de acções confirmadas respeitantes ao Plano de Cuidados em questão. Os elementos dessa lista são do tipo *NurseryActionConfirmationType*. Cada elemento deste tipo possui informação sobre a acção e sua confirmação, Os dados existentes neste elemento indicam o código da acção (*ActionCode*), se acção foi ou não realizada (*Executed*), se foi anulada (*Nullified_PT*), a data a que ocorreu a confirmação (*Date*), os motivos de não realização (*Motive*), ou de anulação (*NullificationReason_PT*), a nota associada caso possua (*Note_PT*), o utilizador que a confirmou (*User*), a Unidade de Enfermaria onde se encontra o doente (*NurseryUnit*), assim como os valores do Objecto Clínico associado à confirmação (*ClinicalRegister*), caso exista.
- (b) **uploadPatientCarePlanResponse**: serve apenas informar se o envio foi efectuado correctamente ou não.

6.2 Armazenamento da Informação

Uma das principais características da aplicação Hcis Mobile é o facto de poder ser utilizado num contexto offline (ou seja, sem qualquer ligação à estrutura física onde está guardada a informação existente). Para tal ser possível, foi necessário encontrar um modo de armazenar toda a informação indispensável para um uso eficiente da aplicação desenvolvida.

Uma das hipóteses ponderadas, no início do desenvolvimento, consistia em fazer o *download* de toda a informação necessária para os Plano de Cuidados que iriam ser utilizados num determinado período, ficando a mesma na memória do dispositivo. Esta hipótese foi descartada pois iria ter custos elevados, não só na *performance* da aplicação (pois iria sobrecarregar o sistema) como também na comunicação entre o servidor central, onde a informação está armazenada, e o dispositivo móvel. Caso fosse necessário carregar um número elevado de Planos de Cuidados com toda a informação relativa ao mesmo, principalmente informação padrão de diagnósticos, intervenções e acções, a transferência de todos esses dados iria ser longa e penosa para o utilizador. Outro inconveniente desta abordagem prendia-se com a não persistência da informação que poderia ser útil e reutilizada posteriormente. Sendo alocada na memória a informação apenas ia estar disponível em *run-time* (enquanto a aplicação estivesse a ser executada). Quando o dispositivo fosse desligado ou a aplicação fechada, propositadamente ou devido a um contratempo, por exemplo um *crash* da aplicação, toda a informação armazenada iria ser perdida.

¹⁸O envio de Plano de Cuidados é atómico, ou seja, em cada mensagem apenas é enviado um Plano de Cuidados.

De forma a garantir a persistência dos dados optámos por recorrer à criação e utilização de uma Base de Dados. A plataforma Android, dentre uma série de inovações e facilidades, traz suporte nativo ao SQLite. O SQLite é uma poderosa biblioteca de Base de Dados baseado em SQL (*Structured Query Language*) que permite controlar diversas Bases de Dados que podem conter inúmeras tabelas, actuando como um *mini-SGDB*¹⁹. Uma explicação mais detalhada do SQLite poderá ser encontrada na Subsecção 6.2.1.

Podemos dividir os tipos de informação em dois grupos:

- **Informação estática:** todo o tipo de informação que é necessário para o correcto funcionamento da aplicação.
- **Informação dinâmica:** dados que são introduzidos, removidos e alterados durante a prestação dos cuidados de saúde.

Fazendo uso do SQLite, para armazenar a informação estática criámos uma **Base de Conhecimento** no dispositivo móvel semelhante à existente na Base de Dados da aplicação HP-HCIS. Sendo a capacidade de armazenamento uma das principais adversidades existentes num dispositivo móvel, pois é limitada e bastante inferior à existente em terminais fixos, decidimos fazer uma migração seleccionada da informação indispensável, em detrimento de se fazer uma migração completa dos dados. Para isso foi efectuada uma análise de quais os campos que efectivamente são utilizados e retirámos aqueles cuja necessidade não é imperiosa.

Através da Base de Conhecimento é possível, por exemplo, aceder à informação "tipo" de cada acção, tal como a sua descrição, o tipo de acção e se possui ou não objecto clínico, entre outras características. Tendo esta informação guardada no dispositivo é apenas necessário ser fornecido o código da Acção através do Web Service. Se não existisse esta Base de Conhecimento era necessário passar toda a informação relativa a cada uma das Acções existentes no Plano de Cuidados dos doentes sempre que fosse necessário fazer o carregamento do Plano de Cuidados.

Para além desta "secção" de informação estática existente, foram implementados mais dois conjuntos de tabelas, um conjunto relativo ao Plano de Cuidados e outro relativo ao Plano de Trabalho. Estas duas "secções" contêm os dados respeitantes aos Planos de Cuidados e Planos de Trabalho de cada doente. Os dados dos Planos de Cuidados são obtidos através da informação enviada pelo Web Service, enquanto que os dados do Plano de Trabalho resultam das prestações de cuidados de saúde efectuadas consoante o Plano de Cuidados.

Nas subsecções seguintes iremos fornecer uma explicação sobre o SQLite e a estrutura da Base de Dados implementada.

¹⁹Segundo [Healey, 1991], *Os SGDB são um conjunto de programas utilizados para armazenar, manipular e obter dados de uma Base de Dados*. Existem vários tipos diferentes de SGDB's, desde sistemas pequenos usados em computadores pessoais até sistemas enormes que correm em *mainframes*.

6.2.1 SQLite

Quando se encontrava ao serviço da Marinha dos Estados Unidos, no desenvolvimento de software utilizado no lançamento de mísseis, Dwayne Richard Hipp sentiu a necessidade de utilizar um sistema de Base de Dados cuja complexidade fosse inferior ao utilizado até então. O software original funcionava sobre o HP-UX²⁰ e utilizava uma Base de Dados Informix. Como, mesmo para um gestor de Base de Dados experiente, a instalação ou upgrade desse tipo de Base de Dados demorava cerca de um dia, Hipp e um colega discutiram sobre a ideia de criar um sistema simples de Base de Dados SQL que utilizasse o **GNU Database Manager** como *back-end*, fosse fácil de utilizar, estivesse incorporado no próprio programa e que não necessitasse de qualquer outro software pré-instalado no sistema. Hipp começou então a desenvolver o SQLite, cuja versão 1.0 foi liberada em Agosto de 2000 [Owens, 2006].

Ao contrário da maioria dos SGBD's existentes, o SQLite não necessita de comunicar com processos externos e independentes para aceder, manipular ou introduzir dados. Ele próprio é uma biblioteca que se torna parte integral da aplicação. A aplicação que incorpora essa biblioteca acede, ela mesma, às funcionalidades do SQLite através de chamadas de funções simples, o que permite reduzir a latência da Base de Dados pois não é necessário efectuar comunicações entre processos. Toda a Base de Dados (incluindo o schema, índices, triggers e os próprios dados) é guardada num só ficheiro no disco da máquina onde a aplicação está a correr. Esse ficheiro é multi-plataformas, podendo ser livremente copiado entre sistemas de 32 e 64 bits sem problemas [SQLite, 2012].

Segundo o WebSite oficial do SQLite [SQLite, 2012], as principais características do mesmo são:

- **Auto-contido:** não necessita de bibliotecas externas ou do próprio sistema operativo, o que faz com que a sua utilização seja bastante apreciada em aplicações que necessitem ser executadas em plataformas o mais distintas possíveis.
- **Independente de uma estrutura cliente-servidor:** a grande maioria dos SGBD's funcionam como um servidor externo à aplicação, ao qual esta se liga através de mecanismos de comunicação entre processos. Como já foi referido, no SQLite o acesso à Base de Dados é feito através da leitura e escrita em ficheiros armazenados no disco.
- **Dispensa configurações:** o SQLite não necessita ser instalado nem configurado. Para além disso, também não possui nenhum processo que precise ser iniciado ou parado.
- **Possui uma estrutura transaccional:** uma Base de Dados transaccional possui os requisitos *ACID* (todas as alterações e *queries* são **A**tómicas, **C**onsistentes, **I**soladas e **D**uráveis).

²⁰Sistema Operativo desenvolvido pela HP e cuja primeira versão ficou disponível em 1984. É uma variante do sistema Unix fortemente baseada no System V [Wikipédia, 2012b].

No contexto de aplicações para dispositivos móveis, a falta de recursos computacionais, tais como memória e capacidade de armazenamento, inerentes a esses mesmos dispositivos (como por exemplo telemóveis e PDAs), combinado com a falta de suporte, faz com que o uso do SQLite seja mais eficaz, em comparação com os SGDB's normalmente utilizados noutros contextos. A simplicidade de administração, implementação e manutenção do SQLite são factores chaves para que o seu uso seja cada vez mais frequente.

O código do SQLite é *open source* e de domínio público. A base do código continua a ser desenvolvida por uma equipa internacional de programadores que trabalham unicamente neste projecto. Os principais objectivos consistem em melhorar e expandir as capacidades do SQLite, nomeadamente melhorar o seu desempenho e fiabilidade, mantendo a compatibilidade com as versões anteriores [SQLite, 2012].

6.2.2 Base de Conhecimento

A Base de Conhecimento contém a informação padrão de cada Diagnóstico, Intervenção e Acção, assim como se relacionam entre eles. Cada Diagnóstico possui um conjunto de Intervenções, que por sua vez possui um determinado grupo de Acções. Por sua vez cada Intervenção pode pertencer a mais que um Diagnóstico e cada Acção a mais que uma Intervenção. No caso do CIPE o mapeamento entre Acções e Intervenções é feito de 1:1, a uma Intervenção corresponde apenas uma Acção. Estas relações, assim como a constituição de cada tabela, podem ser visualizadas na Fig. 6.2.

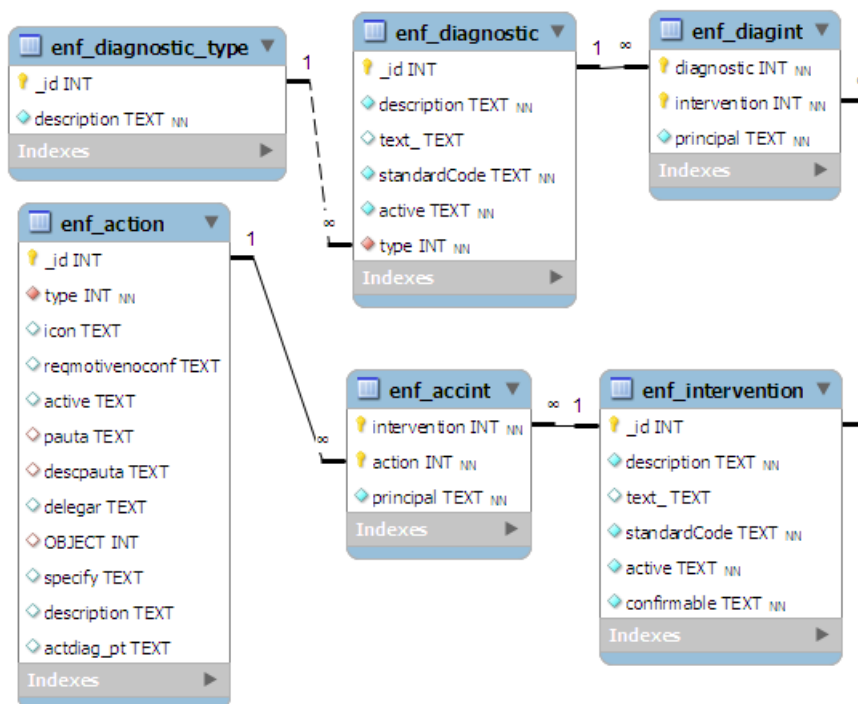


Figura 6.2: Relação entre Diagnósticos, Intervenções e Acções

As Acções são um dos componentes mais importantes da Plano de Cuidados, pois são essas Acções que, efectivamente, irão ser aplicadas aos doentes. Cada Acção tem características próprias, tais como o seu tipo (horária, contínua, pontual ou a pedido), a sua frequência (no caso de ser horária é indicado de quanto em quanto tempo deve ser realizada) e se possui Objecto Clínico associado. A Fig.6.3 ilustra as tabelas que fornecem toda a informação necessária à constituição das Acções.

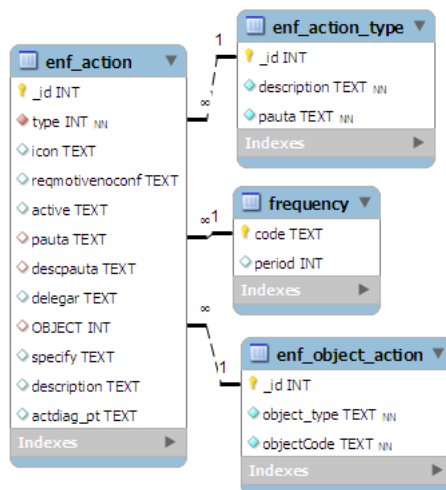


Figura 6.3: Características padrão das Acções

Algumas Acções possuem Objectos Clínicos. Os Objectos Clínicos são tipos de informação que se pode associar às Acções aquando da sua confirmação. Podem ser de vários tipos, no entanto no Hcis Mobile apenas foram implementados objectos do tipo **Anotações** e **Transacções**. As Anotações, como o próprio nome indica são notas específicas que se podem adicionar à confirmação da Acção. Uma Acção que possui um objecto clínico do tipo Anotação é a Acção *Vigiar pele periférica ao estoma*, que possui a Anotação do tipo *Notas associadas (Enfermagem)*. O outro tipo de objecto clínico implementado são as Transacções. As Transacções são conjuntos de Observações Clínicas relacionadas com a Acção. A associação entre a Acção e o Objecto Clínico correspondente é feita entre as tabelas *enf_object_action* e *doc_annotations_type*, no caso das Anotações, ou *doc_tx_type*, no caso das Transacções. O tipo de objecto é indicado na coluna *object_action.object_type*. Essa associação é feita entre o código do objecto na tabela *enf_object_action* (*object_action.object_code*) e os ID's da tabela respectiva: *doc_annotations_type._id* ou *doc_tx_type._id*, ver Fig. 6.4).

Enquanto as Anotações são um tipo de objecto simples (servem apenas para inserir uma anotação) as Transacções são objectos mais complexos. Cada Transacção possui um certo número de Observações Clínicas e a mesma Observação Clínica pode pertencer a Transacções distintas. As observações podem ser de vários tipos, entre os quais:

- **Boolean**: valor lógico Verdadeiro ou Falso, Sim ou Não.
- **Inteiro**: representa um número inteiro.

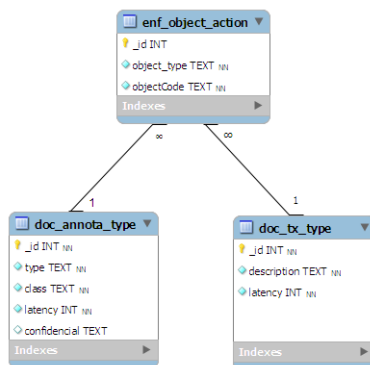


Figura 6.4: Tipos de Objectos Clínicos

- **Decimal:** representa um número decimal.
- **Dicionário:** Valores pré-definidos num dicionário.
- **Hora:** representa uma hora.
- **Texto:** permite inserir texto livre.

Se a observação for do tipo **Dicionário** vai ter associada um *Dicionário* que está associado a uma determinada Especialidade e que possui um conjunto de itens.

Para além de todas estas particularidades a observação pode ter também associada uma unidade para os seus valores (p.e. g/l). O modo como é feito o relacionamento entre as Transacções, as Observações e as características destas está esquematizado na Fig.6.5.

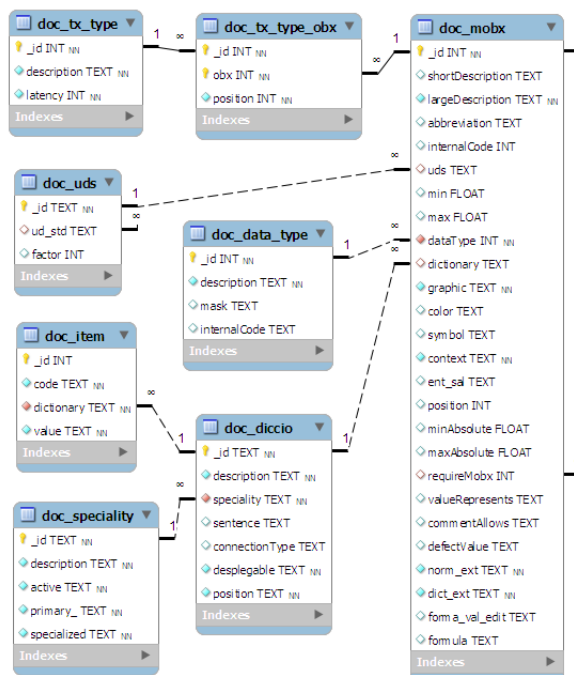


Figura 6.5: Transacções e Observações Clínicas

Para além das tabelas referidas acima, fazem ainda parte da Base de Conhecimento as tabelas onde são guardados os hospitais (tabela *hospitals*) e os géneros (tabela *gender*). Estas tabelas, e a sua composição, podem ser vistas na Fig.6.6.

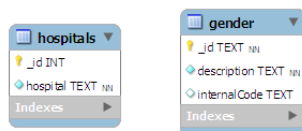


Figura 6.6: Tabelas *hospitals* e *gender*

Todos os dados destas tabelas são introduzidos quando se instala a aplicação, permanecendo imutáveis, à excepção da tabela *hospitals*, cujo carregamento depende do Centro Hospitalar a que se estiver ligado, e que pode ser actualizada frequentemente.

Apesar da Base de Conhecimento estar baseada na Base de Dados HP-HCIS, os nomes das tabelas e dos seus campos foram traduzidos para inglês por uma questão de internacionalização e coerência. A Tabela 6.2 indica a correspondência entre as tabelas do Hcis Mobile e da aplicação HP-HCIS²¹.

Hcis Mobile	HP-HCIS
doc_annota_type	doc.tipo_anota
doc_data_type	doc.tipodato
doc_diccio	doc.diccio
doc_item	doc.item
doc_mobx	doc.mobx
doc_speciality	doc.especialidad
doc_tx_type	doc.tipotx
doc_tx_type_obx	doc.tipotx_obx
doc_uds	doc.uds
enf_action	enf.accion
enf_action_type	enf.tipoaccion
enf_diagint	enf.diagint
enf_diagnostic	enf.diagnostico
enf_diagnostic_type	enf.tipo_diagnostico
enf_intervention	enf.intervencion
enf_object_action	enf.objetoaccion
frequency	om.frecuencia
gender	sexo
hospitals	hopital

Tabela 6.2: Correspondência entre tabelas Hcis Mobile e HP-HCIS da Base de Conhecimento

²¹A tabela *frequency* apenas utiliza o campo *codigo* da tabela referência tendo o período sido calculado com base na descrição da tabela referência.

6.2.3 Plano de Cuidados

Os Planos de Cuidados de cada doente são obtidos através do Web Service. Estes dados, apesar de permanecerem estáticos, não são considerados Base de Conhecimento pois são relativos ao doente e ao longo do internamento podem sofrer modificações.

Através da visualização da Fig. 6.7, constatamos que cada Plano de Cuidados está associado a um doente, a uma unidade de enfermaria e a um ou dois utilizadores, o utilizador que criou o Plano de Cuidados em questão e, caso exista, o último utilizador que o alterou. No esquema da figura acima mencionada pode-se verificar a ligação entre as tabelas do Plano de Cuidados com tabelas da Base de Conhecimento.

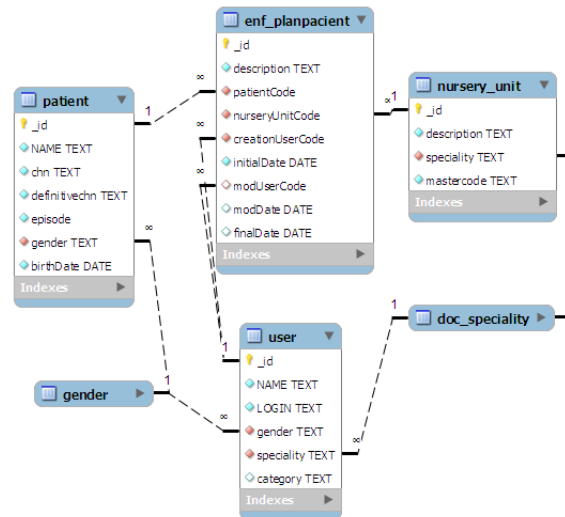


Figura 6.7: Alguns componentes do Plano de Cuidados

Contudo, a parte mais relevante do Plano de Cuidados é compreendido pelas Acções que se devem administrar ao doente em questão. Essas Acções estão agrupadas em dois sectores diferentes. Acções pertencentes a Diagnósticos e Acções pertencentes a Actividades. As Actividades estão divididas em três grupos: Interdependentes, Diagnósticas e Não Diagnósticas. A tabela *enf_plan_diaglist* possui informação sobre os Diagnósticos pertencentes ao Plano. Esta tabela possui uma ligação de 1:n com a tabela *enf_plan_diag_int*, onde estão armazenadas as Intervenções pertencentes a um determinado Diagnóstico. A tabela onde são guardadas as Acções é a *enf_plan_int_acc*. Cada elemento desta tabela está relacionado com um elemento da tabela das Intervenções do plano. Existe ainda a tabela *enf_acc* que contém as Acções que pertencem a Actividades do Plano. As tabelas *enf_plan_int_acc* e *enf_acc* são bastante semelhantes, alterando apenas na chave estrangeira, na primeira é o código da Intervenção respeitante enquanto na segunda é o código do Plano. Para além deste aspecto a segunda tabela possui mais um campo que indica qual o tipo de Actividade a que pertence, o campo *type_act* (ver Fig. 6.8). As correspondências

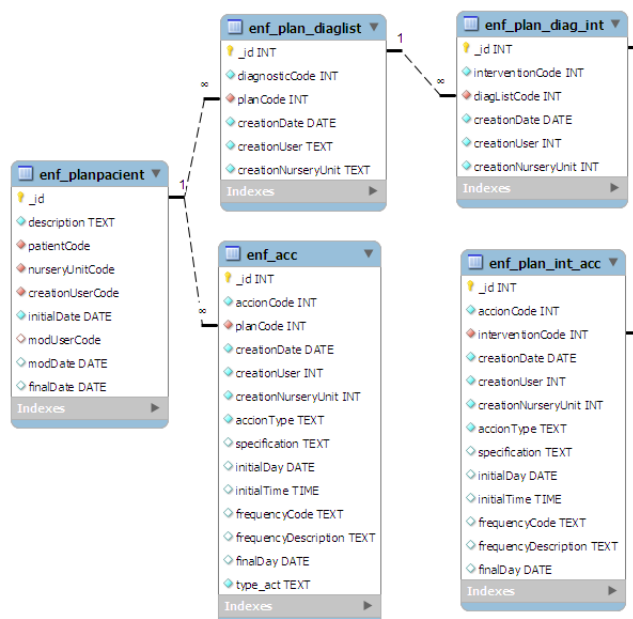


Figura 6.8: Alguns componentes do Plano de Cuidados

entre as tabelas Hcis Mobile e HP-HCIS encontram-se na Tabela 6.3²².

Hcis Mobile	HP-HCIS
enf_planpatient	enf_plandoente
patient	doente
nursery_unit	unidade enfermaria
user	sgi_persona
enf_plan_diaglist	enf_plan_diag
enf_plan_diag_int	enf_plan_int
enf_plan_int_acc	enf_plan_int_acc
enf_acc	enf_plan_accion

Tabela 6.3: Correspondência entre tabelas Hcis Mobile e HP-HCIS do Plano de Cuidados

6.2.4 Plano de Trabalho

A parte da Base de Dados relativa ao Plano de Trabalho irá conter a informação sobre os cuidados prestados. As tabelas correspondentes a este sector guardam a informação da confirmação das Acções, tais como o valor da confirmação (Realizada, Não Realizada ou Anulada), o motivo da não realização ou anulação (caso seja este o estado da confirmação) e a data e hora da confirmação. As Figs. 6.9 e 6.10 ilustram as relações existentes entre estas tabelas. Para além destes dados, relativos à confirmação, são também guardados os valores dos objectos clínicos, Anotação ou Transacção, com as respectivas Observações Clínicas,

²²Estas tabelas servem apenas para referência, não existindo um mapeamento directo entre elas.

no caso das acções os possuírem. Existem duas tabelas de confirmação, uma respeitante às Acções pertencentes a Diagnósticos e a outra respeitante às Acções pertencentes a Actividades, com as respectivas associações às tabelas dos Objectos Clínicos.

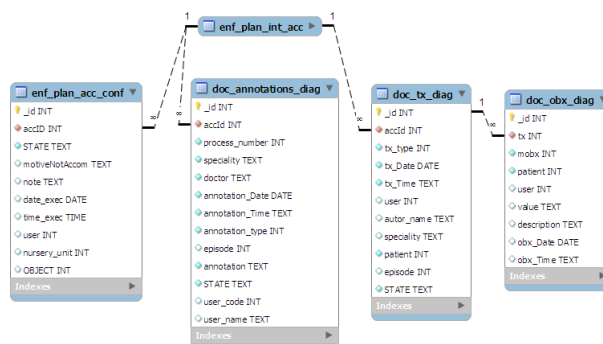


Figura 6.9: Secção de Confirmação de Acções pertencentes a Diagnósticos

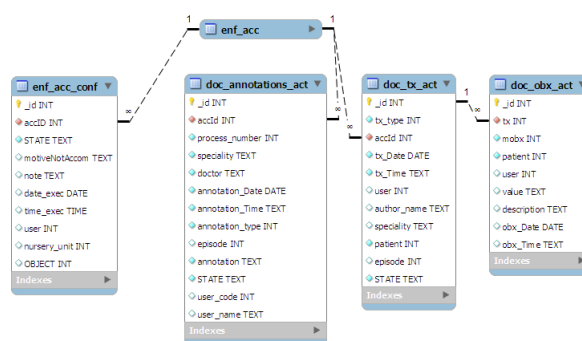


Figura 6.10: Secção de Confirmação de Acções pertencentes a Actividades

Para concluir, a correspondência entre as tabelas Hcis Mobile e HP-HCIS, referente ao Plano de Trabalho, pode ser vista na Tabela 6.4²³

Hcis Mobile	HP-HCIS
enf_acc_conf	enf_conf_accion
enf_plan_acc_conf	enf_plan_ac.conf
doc_annotations_diag	doc_annotaciones
doc_annotations_act	
doc_tx_diag	doc_tx
doc_tx_act	
doc_obx_diag	doc_obx
doc_obx_act	

Tabela 6.4: Correspondência entre tabelas Hcis Mobile e HP-HCIS do Plano de Trabalho

O Schema SQL referente à Base de Dados Hcis Mobile pode ser consultado no Anexo B.

²³Em exemplo do que acontece com o Plano de Cuidados, a correspondência entre as tabelas é apenas indicativa.

6.3 Funcionalidades

A nível funcional a aplicação Hcis Mobile está dividida em cinco módulos, sendo eles os seguintes:

- **Login/Autenticação**
- **Carregamento e Listagem dos Planos de Cuidados**
- **Visualização do Plano de Trabalho**
- **Confirmação de Acções de Enfermagem**
- **Envio dos Planos de Cuidados**

Em seguida procedemos à explicação do que consiste cada um destes módulos.

6.3.1 Login/Autenticação

Tendo em conta que a aplicação foi desenvolvida para ser utilizada em dispositivos móveis, que podem ser utilizados por vários utilizadores distintos, torna-se imperioso que quem os está a utilizar seja realmente quem diz ser. Para além disso existe sempre o perigo de se perder o dispositivo e é necessário proteger o acesso à informação clínica privada dos doentes. Caso o acesso à aplicação não estivesse protegida qualquer pessoa podia aceder aos dados dos doentes sem possuir permissões para tal.

Outro aspecto bastante importante para o qual se utiliza a autenticação está relacionado com a confirmação das Acções. Cada Acção realizada ao doente deve possuir um responsável associado. Utilizando o mecanismo de autenticação garante-se que apenas utilizadores autorizados possam efectuar registos ficando os mesmos associados a quem os introduziu no sistema.

O Login é indispensável para um utilizador conseguir aceder a todas as outras funcionalidades da aplicação. Ao se aceder à aplicação é sempre necessário realizar o Login, que caso não seja efectuado correctamente não permite que seja realizada mais nenhuma operação. A única acção que não necessita de Login é o carregamento, na Base de Dados, dos Hospitais aos quais o utilizador se pode ligar.

O Menu de Login, Fig. 6.11, possui três campos: Utilizador, Password e Hospital. Cada utilizador possui uma conta pessoal no sistema HP-HCIS, sendo as credenciais utilizadas no Hcis Mobile as mesmas da aplicação HP-HCIS. O utilizador introduz os seus dados que são enviados, via Web Service, ao sistema central. Caso o Login esteja correcto é permitido o acesso à aplicação.

Para além do menu de Login, a Autenticação é também utilizada sempre que se pretende comunicar com o servidor central, mais concretamente quando se efectua o carregamento



Figura 6.11: Menu de Login

ou envio dos Planos de Cuidados. Sempre que é exibida uma janela para se concretizar a Autenticação, Fig. 6.12, é apenas necessário que o utilizador introduza a sua *password*, estando os restantes campos já pré-preenchidos com os dados relativos ao Utilizador e Hospital indicados aquando do Login. Apesar destes campos já estarem previamente preenchidos podem ser alterados. No caso do carregamento existe ainda mais um campo para o utilizador indicar o valor da opção que seleccionou para efectuar o carregamento (ver a Subsecção 6.3.2).



(a) Carregamento

(b) Envio

Figura 6.12: Autenticação

6.3.2 Carregamento e Listagem dos Planos de Cuidados

Após efectuar o Login correctamente é exibido ao utilizador o menu principal do Heis Mobile, Fig. 6.13. Esse menu possibilita ao utilizar, entre outras, a opção de efectuar os downloads dos Planos de Cuidados que iriam ser necessários durante a prestação dos Cuidados de Saúde. Existem três tipos distintos de carregamentos: por **NHC**²⁴, por **Quarto** ou por **Unidade de Enfermaria**. Em todas estas opções é pedido ao utilizador a sua autenticação (Fig. 6.12a) e o valor desejado (NHC, código do quarto ou código da unidade, respectivamente).

²⁴ Número de História Clínica, sendo este único por doente.

No caso de escolher a opção *Carregamento Por NHC* é apenas efectuado o carregamento do Plano de Cuidados do doente correspondente ao doente que possua esse NHC, caso a opção escolhida seja uma das restantes é efectuado o carregamento dos Planos de Cuidados de todos os doentes que estejam no quarto ou na unidade de enfermaria pretendida.

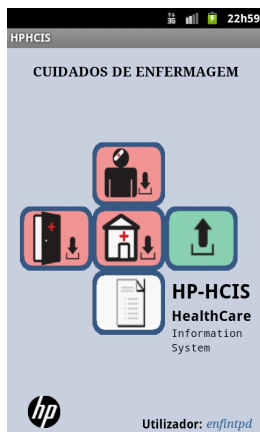


Figura 6.13: Menu HCIS

Após se ter realizado o carregamento dos Planos de Cuidados desejados é possível aceder à listagem dos mesmos a partir do menu principal. Essa listagem, Fig. 6.14, permite ao utilizador seleccionar o doente ao qual se pretende introduzir informação relativa às prestações de cuidados prestadas.

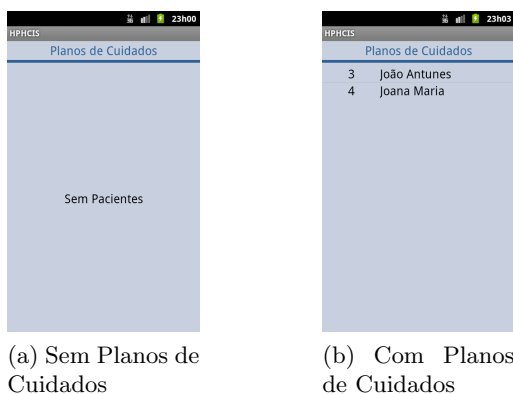


Figura 6.14: Menu de Listagem dos Planos de Cuidados

6.3.3 Visualização do Plano de Trabalho

Ao se seleccionar um doente, a partir do Menu Listagem dos Planos de Cuidados, Fig. 6.14, é exibido o Plano de Trabalho relacionado, Fig. 6.15. Nesse menu, para além da identificação do doente (nome, sexo e idade, perceptíveis através do ícone existente no canto superior direito do ecrã) é fornecida a indicação de quais Acções são necessárias efectuar ao doente.

As Acções estão associadas a um de dois tipos: **Diagnósticos** ou **Actividades**. No separador dos **Diagnósticos** são listados todos os Diagnósticos pertencentes ao Plano de Trabalho e quais as Acções que os constituem. Relativamente às Actividades estas podem ser de três tipos diferentes: **Diagnósticas**, **Não Diagnósticas** ou **Interdependentes**. Dentro de cada uma dessas secções encontram-se as Acções correspondentes.



(a) Separador Diagnósticos

(b) Separador Actividades

Figura 6.15: Menu Plano de Trabalho

Por sua vez, relativamente à periodicidade à qual devem ser prestadas, as Acções podem ser de dois tipos: **Contínuas** ou **Periódicas**. As Acções Contínuas são representadas com um tracejado enquanto as Periódicas possuem a próxima hora a que devem ser realizadas.

6.3.4 Confirmação de Acções de Enfermagem

As Acções são os elementos mais importantes do Plano de Trabalho. É através das Acções que o enfermeiro sabe quais são, efectivamente, os cuidados que o doente necessita, e em que altura. As Acções possuem quatro estados distintos:

- **Estado Inicial:** a Acção apenas está neste estado quando não foi realizada nenhuma confirmação sobre a mesma.
- **Confirmada Realizada:** quando a Acção foi realizada pelo enfermeiro.
- **Confirmada Não Realizada:** quando a Acção foi marcada como não realizada.
- **Anulada:** quando a confirmação de uma Acção foi anulada.

O esquema da Fig. 6.16 ilustra o *circuito de vida* de uma Acção. Inicialmente todas as Acções estão no seu estado inicial, ou seja, ainda nenhuma confirmação foi efectuada sobre as mesmas. Deste estado apenas podem passar para o estado **Confirmada Realizada** ou **Confirmada Não Realizada**. Se uma Acção for confirmada como Não Realizada é necessário introduzir o motivo pela qual não foi efectuada. Se por sua vez for marcada como Realizada pode ser necessário introduzir mais informação sobre a mesma, se esta

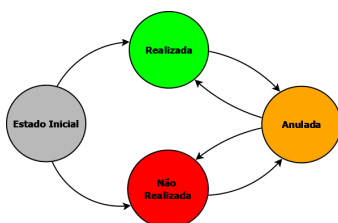


Figura 6.16: Circuito de Estados de uma Acção

possuir um Objecto Clínico associado. Estando num estado de Confirmada a Acção apenas pode passar para o estado **Anulada**, de onde pode voltar para um dos dois estados de confirmação descritos anteriormente. Para além da opção para se escolher o estado pode ser também introduzida uma pequena nota associada à confirmação ou anulação da Acção, como pode ser visto na Fig. 6.17, que fornece um exemplo de uma Acção no seu estado inicial.

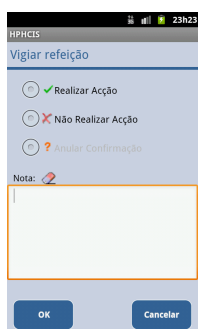


Figura 6.17: Menu Confirmação de Acções

Após se realizar uma confirmação ou anulação de uma determinada Acção, o Menu de Confirmação reflecte essa alteração. O estado actual da Acção está marcado por defeito e é exibido a hora que ocorreu a confirmação. No caso de ser Realizada e a acção possuir um Objecto Clínico associado é possível aceder aos dados registados através do ícone que aparece junto ao estado Realizada. Se, por sua vez, o estado for Não Realizada é exibido o motivo da não realização.

6.3.5 Envio dos Planos de Cuidados

Após se realizarem as confirmações das Acções com todos os dados associados é necessário que essa informação seja enviada para a Base de Dados do HP-HCIS. Esse envio pode ser feito de duas maneiras: individualmente por Plano ou em conjunto, enviando todos os Planos.

Caso se pretenda enviar um Plano específico, ou após se prestarem todos os cuidados existentes no Plano de Trabalho, é possível enviar esse plano a partir do Menu Plano de Trabalho, utilizando, para tal, a opção **Enviar** existente no menu de opções, que pede ao

utilizador para introduzir as suas credenciais (Ver Fig. 6.12b).

Se se pretender enviar mais que um plano pode-se fazê-lo acedendo ao **Menu de Envio**, Fig. 6.18. Por defeito, aparecem seleccionados apenas os Planos de Cuidados que possuam Acções que já não estejam no estado Inicial, ou seja, que estejam num estado de confirmação ou que estejam anuladas. Esta pré selecção torna-se importante para o utilizador pois assim sabe quais são realmente os Planos que necessitam de serem enviados. Ao existirem Planos cujas todas as Acções que os compõem nem tenham sofrido alterações, o seu envio é desnecessário, pois não existem dados novos para se introduzirem no sistema. No entanto é possível seleccionar quaisquer Planos existentes no Hcis Mobile. Após se efectuar a selecção de quais Planos se pretendem enviar, à semelhança do que acontece no *upload* individual do Plano, é pedido ao utilizador que introduza as suas credenciais (Fig. 6.12b). Se o envio do Plano tiver ocorrido correctamente o mesmo é eliminado do dispositivo, de modo a minimizar o espaço ocupado.



Figura 6.18: Menu de Envio

6.3.6 Circuito Funcional

O circuito *padrão* do Hcis Mobile consiste na seguinte sequência: Login, carregamento dos Planos de Cuidados, selecção de um Plano de Cuidados, confirmação das Acções respeitantes ao Plano escolhido e envio do Plano com as Acções confirmadas.

Pretende-se, no entanto, que seja fácil *navegar* por todas as funcionalidades da aplicação sem ser necessário seguir esse circuito de um modo rigoroso. Imaginemos que o utilizador pretende confirmar Acções de vários planos e só após isso enviar todos os planos confirmados. Se apenas fosse permitido navegar pelos Menus *adjacentes*, após confirmar as acções do último Plano de Cuidados o utilizador tinha que voltar dois Menus para trás, até chegar ao Menu HCIS, para ser possível aceder ao Menu de Upload. De maneira a evitar isso utilizámos o Menu de Opções, disponibilizado no Android, para que, qualquer que seja o Menu, seja possível navegar para onde se deseja. Tendo em conta o exemplo dado anteriormente, o utilizador estando no menu do Plano de Trabalho pode, através do Menu de Opções, seleccionar a opção **Enviar** que lhe permite ir imediatamente para o Menu de Envio e assim enviar todos os planos desejados. A Fig. 6.19 esquematiza todos os circuitos de navegação existentes e possíveis na aplicação.

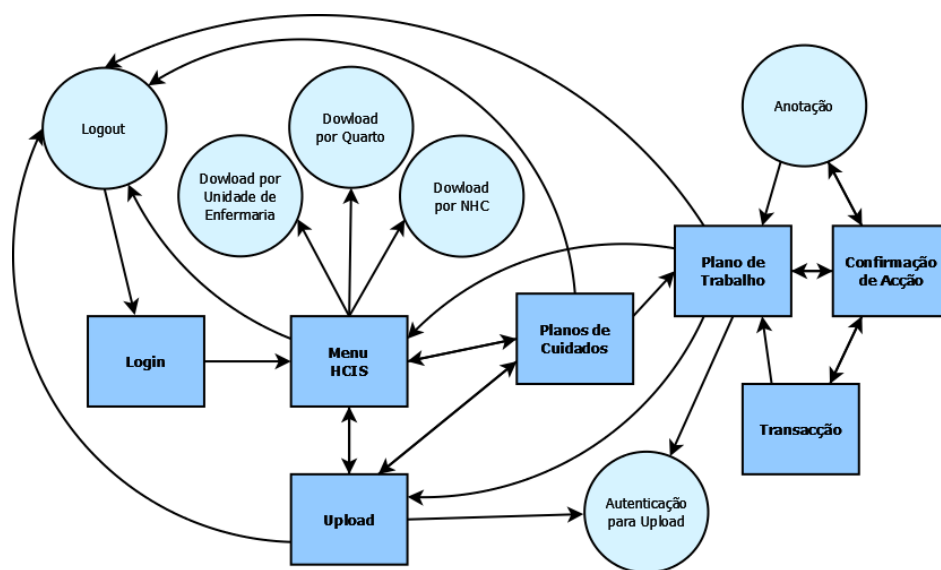


Figura 6.19: Circuito Funcional

Para além das funcionalidades descritas, existem mais duas que estão disponíveis em quase todos os Menus de Opções dos menus existentes. A possibilidade de efectuar o **Logout** da aplicação e o acesso à **Configuração do Web Service**. Sempre que é efectuada o Logout da aplicação todos os dados relativos às confirmações são eliminados do dispositivo, devido a esta particularidade sempre que o utilizador pretende efectuar o Logout, e caso existam dados por enviar, o utilizador é informado disso mesmo dando-lhe a hipótese de ir directamente para o Menu de Envio onde pode efectuar o envio de todos os planos que pretenda. É de referir que o utilizador pode sair a qualquer instante da aplicação através do botão *Home* existente nos dispositivos Android sem efectuar o Logout, podendo assim aceder a outras funcionalidades do dispositivo e não perder os dados registados até ao momento. A eliminação dos dados só se concretiza caso se efectue o Logout a partir da aplicação.

Capítulo 7

Implementação

Neste capítulo são descritos os detalhes de implementação da aplicação Hcis Mobile. É descrito o ambiente de desenvolvimento utilizado e a estrutura de classes que compõe o código fonte da aplicação, explicando as principais funcionalidades disponíveis nas mesmas. Relativamente à interface gráfica iremos ilustrar a evolução que a mesma teve desde o início do projecto, evidenciando todas as fases que passou, até chegarmos a interface actual. Por fim iremos apresentar os resultados de alguns testes efectuados.

7.1 Ambiente de Desenvolvimento e Testes

Todos os desenvolvimentos e testes deste projecto foram efectuados num laptop *HP pavilion dv5*, que possuía 4 Gb de memória RAM e um processador *Core 2Duo* de 2.53 GHz. Nesse laptop foi instalado o *Android SDK, versão 12*, integrado, através do *plugin ADT Plugin for Eclipse*, no **IDE Eclipse Helio, versão 3.6.2**. O projecto foi desenvolvido tendo como alvo a versão Android 2.3.1, razão pela qual foi necessário a criação de um dispositivo virtual (*emulador*) de Android, recorrendo à ferramenta **AVD Manager** disponível no Android SDK, tendo esse dispositivo virtual as seguintes características:

- **Alvo:** Google API Level 9 (corresponde à versão Android 2.3.1)
- **Resolução do ecrã:** 320x480
- **Cartão de Memória:** 32 GB

Para além das ferramentas e *software* descrito acima, utilizado para o desenvolvimento da aplicação na plataforma Android, foi utilizado, como já referido em 6.1.2, o *software*

`soapUI` que permitiu simular o servidor HP-HCIS.

7.2 Estrutura do Código e Descrição das Classes

Como qualquer aplicação Android, o código fonte está separado em quatro directorias:

- **bin**: nesta directoria são guardados os ficheiros Java compilados `.class` e o `APK` gerado.
- **gen**: contém o ficheiro `R.java`, gerado automaticamente.
- **res**: onde estão alojados os ficheiros de recursos da aplicação, sendo que uns são utilizados para a construção do *layout* das interfaces enquanto outros irão conter os pares de chave-valor utilizados pela aplicação. Para além dos ficheiros já referidos, é também nesta directoria que ficam guardados todas as imagens utilizadas na aplicação.
- **src**: contém todo o código fonte Java da aplicação.

Devido à necessidade de recorrermos a uma biblioteca externa procedemos à criação de uma nova directoria, a qual denominámos **lib**, e que contém a biblioteca **ksoap2-android**.

Para melhor legibilidade e estruturação do código, dividimos a implementação das classes em quatro *packages*: no *package* **hp.hcis.nursery.bd** estão contidas todas as classes relativas às operações de Base de Dados, o *package* **hp.hcis.nursery.ws** contém a classe que coordena toda a ligação *Web Service* do lado do dispositivo móvel, o *package* **hp.hcis.nursery.ui** é formado por todas as classes que instanciam *Activities* e que, conseqüentemente, correspondem aos Menus do Hcis Mobile, por último no *package* **hp.hcis.nursery.pojo** podemos encontrar todas as outras classes correspondentes a objectos necessários para a implementação da aplicação. Em seguida iremos descrever melhor as classes que compõem cada um destes *packages*.

7.2.1 *Package* hp.hcis.nursery.bd

Este *package* contém as classes **BaseDados**, **DataBaseConstructor**, **Fill1**, **Fill2**, **Fill3** e **Fill4**.

A classe **DataBaseConstructor**, como próprio nome indica, é utilizada para a construção do *Schema* da Base de Dados. inclui todos os métodos respeitantes à criação das tabelas assim como invoca os métodos que populam a Base de Conhecimento¹.

¹Ver Subsecção 6.2.2.

A classe **BaseDados** é classe que contém os métodos referentes à interacção com a Base de Dados. Esses métodos permitem efectuar consultas (*select*), actualizações (*update*), inserções (*insert*) e eliminações (*remove*) de todos os dados.

As classes **Fill1**, **Fill2**, **Fill3** e **Fill4** são classes estáticas que apenas contém um método cada uma. Cada um desses métodos contém instruções de *inserts* usadas para popular a Base de Conhecimento.

7.2.2 *Package* hp.hcis.nursery.ws

Para gerirmos toda a comunicação *Web Service* utilizamos a classe **WebService**, sendo esta a única classe existente neste *package*. Esta classe é responsável pela codificação das mensagens **SOAP** enviadas, como pelo *parser* das mensagens de resposta. Pelo facto do SDK do Android não oferecer uma solução fácil e embutida para consumir *Web services*, para conseguirmos implementar esses métodos foi necessário recorrermos a uma biblioteca externa, a biblioteca **ksoap2-android**². O *ksoap2-android* é uma biblioteca *open-source*, leve e eficiente para execuções SOAP do lado da aplicação cliente. Esta biblioteca está sob a **licença MIT**, pelo que pode ser utilizada em software livre e proprietário.

7.2.3 *Package* hp.hcis.nursery.pojo

Nesta secção da estrutura do código fonte encontram-se todos os objectos *simples* da aplicação. São considerados simples porque não possuem nenhum método mais elaborado, apenas um construtor e os métodos *getters* e *setters* referentes aos seus atributos. As classes *POJO* implementadas foras as seguintes:

- **Accao**: os objectos instanciados correspondem às Acções de enfermagem existentes nos Plano de Cuidados. Todos os seus atributos são do *tipo primitivo*. Os atributos³ existentes nesta classe são: **long** *actionCode*, **long** *actionDefinitionCode*, **String** *creationDate*, **String** *creationUser*, **String** *creationNurseryUnit*, **String** *actionType*, **String** *specification*, **String** *initialDay*, **String** *initialTime*, **String** *frequencyCode*, **String** *frequencyDescription* e **String** *finalDay*.
- **Intervencoes**: esta classe permite a construção dos objectos referentes às Intervenções do Plano de Cuidados. Cada intervenção pode possuir várias acções, pelo que foi necessário a criação de um atributo do tipo *complexo*, o atributo **accion**. É considerado *complexo* pois recorre à instanciação de um Objecto, neste caso um objecto da classe *ArrayList*. Os outros atributos existentes são: **long** *interven-*

²<http://code.google.com/p/ksoap2-android/>.

³O *significado* da grande maioria dos atributos existentes nestas classes não irá ser explicado pois julgamos que os nomes dos mesmo sejam esclarecedores.

tionCode, **String** *creationDate*, **String** *creationUser*, **String** *creationNurseryUnit* e, o já referido, **ArrayList**<**Accao**>*accion*.

- **Diagnosticos**: Esta classe corresponde aos Diagnósticos de Enfermagem. Estruturalmente é idêntica à classe **Intervencoes**. Possui um **long** *diagnosticCode*, **String** *creationDate*, **String** *creationUser*, **String** *creationNurseryUnit* e um conjunto de intervenções, **ArrayList**<**Intervencoes**>*inter*.
- **Patient**: Os doentes são representados por objectos desta classe: Os atributos que cada *doente* possui são: **long** *code*, **String** *name*, **String** *CHN*⁴, **String** *definitiveCHN*, **long** *episode*, **String** *gender* e **String** *birthDate*.
- **NurseryUnit**: representa as Unidades de Enfermaria, que são caracterizadas por **long** *code*, **String** *description*, **String** *speciality* e **String** *masterCode*.
- **User**: Esta classe permite instanciar os objectos referentes aos utilizadores da aplicação móvel, que serão os enfermeiros. Cada *enfermeiro* possui um **long** *code*, um **String** *name*, um **String** *login*, um **String** *gender*, uma **String** *speciality* e pertencem a uma determinada **String** *category*.
- **PlanoInfo**: Esta classe é utilizada para organizar a informação relativamente a cada um dos Planos de Cuidados. Um Plano de Cuidados possui um **long** *code* e uma **String** *description*. Está associada um **Patient** *patient* de uma **NurseryUnit** *nurseryUnit* específica. Indica qual o **User** *creationUser* que criou o Plano e quando, **String** *initialDate*. Caso o Plano tenha sido alterado refere quem o fez, **User** *modificationUser* e **String** *modificationDate*. Indica também qual a sua data de fim **String** *finalDate*.
- **Plano**: Por último temos a classe que faz a ligação entre todas as classes deste *package*. Cada *Plano* possui informação sobre o mesmo, **PlanoInfo** *planoInfo*, e é formado por **ArrayList**<**Accao**>*accNaoDiag*, **ArrayList**<**Accao**>*accDiag* e **ArrayList**<**Accao**>*accInterDep*.

7.2.4 *Package* hp.hcis.nursery.ui

Todos os menus existentes no Hcis Mobile correspondem a *Activities*, cujas classes estão contidas neste *package*. Para além dessas classes também estão incluídas neste *package* classes utilizadas para personalizar alguns *widgets* gráficos disponibilizados pela plataforma Android.

Relativamente às classes que estendem a classe *Activity*, ou uma subclasse dessa, irão ser identificados os métodos que as compõem assim como a sua funcionalidade. Como existem alguns métodos comuns a elas apenas irá ser descrita a funcionalidade do método uma

⁴Como julgamos que o nome deste atributo pode causar alguma incerteza sobre o que representa achamos pertinente fornecer uma identificação do seu significado: **Número de História Clínica**.

vez, referindo apenas aspectos particulares em cada uma das ocorrências posteriores que ocorram.

Nesta secção apenas desejamos explorar as classes do ponto de vista funcional, ficando os aspectos referentes aos componentes visuais para análise na secção 7.3.

A classe **Login** corresponde ao primeiro menu da aplicação, o **Menu Login**. Os métodos que a mesma disponibiliza são:

- **onCreate**: cria o **Menu Login** cujo *layout* está definido no ficheiro *login.xml*. No caso específico deste menu, verifica se existe algum *login* activo e caso exista o utilizador é imediatamente redireccionado para o **Menu HCIS**.
- **setSpinner**: actualiza a lista de Hospitais aos quais o utilizador se pode ligar, com base na informação existente na Base de Dados.
- **onCreateOptionsMenu**: método utilizado quando o utilizador carrega na tecla **Menu** existente no dispositivo Android. Apresenta o menu de opções relativa à actividade que o invoca, neste caso o **Menu Login**. As opções disponíveis neste menu de opções estão definidas no *menu.login.xml*. Ao aceder a este menu de opções o utilizador pode optar por **Configurar Webservice** ou **(Re)carregar Hospitais**.
- **onMenuItemSelected**: método invocado ao clicar numa das opções existentes no menu de opções. Se a opção seleccionada for **(Re)carregar Hospitais** é efectuada uma chamada ao *Web Service* que irá carregar, ou actualizar, a lista de hospitais existentes no dispositivo. Se, por sua vez, o utilizador seleccionar a opção **Configurar Webservice** irá ser exibido um *Dialog* onde o utilizador introduz os parâmetros necessários para efectuar a comunicação.
- **onCreateDialog**: método utilizado pela superclasse *Activity* que permite fazer a gestão de todos os *Dialogs* existentes no contexto dessa actividade.
- **wsConfigurationDialog**: instancia a janela *Dialog* utilizada para inserir os parâmetros necessários para se efectuar a comunicação *Web Service*. O *layout* relativo a este *Dialog* está implementado em *ws_config_dialog.xml*. Os valores dos parâmetros do *Web Service* ficam guardados no ficheiro *config*.
- **onLogin**: valida os dados de autenticação e, caso a autenticação seja correcta, é iniciada a *Activity* referente à classe **HPHICS**.

A classe **HPHCIS** corresponde ao menu principal da aplicação, o **Menu HP-HCIS**. A partir deste *menu* o utilizador pode optar por três opções: carregar Planos de Cuidados, visualizar os Planos de Cuidados existentes no dispositivo ou efectuar o envio dos Planos de Cuidados com as confirmações das acções. Os métodos que a mesma disponibiliza são:

- **onCreate**: o *layout* desta *Activity* está implementado no ficheiro *login.xml*.

- **onLoad**: este método é invocado quando o utilizador pretende carregar os Planos de Cuidados.
- **onCreateOptionsMenu**: a partir do menu de opções desta *Activity* o utilizador pode optar por **Configurar Webservice** ou efectuar o **Logout**. Este menu está definido em *menu_home.xml*.
- **uploadPlans**: este método inicia a *Activity CarePlansUpload*.
- **visualizePlans**: este método inicia a *Activity CarePlansList*.
- **showLogoutDialog**: instancia o *Dialog* de *Logout*. A interface deste *Dialog* está definida em *logout_dialog.xml*.
- **typeOfConsultDialog**: instancia o *Dialog* a partir do qual se podem carregar os Planos de Cuidados. A sua interface está esquematizada em *dialog_load_type*, sendo personalizado consoante o tipo de carregamento.
- **onMenuItemSelected**
- **onCreateDialog**
- **wsConfigurationDialog**

A *Activity* que permite listar os Planos de Cuidados, **Menu de Listagem dos Planos de Cuidados**, é a **CarePlansList**. Esta classe não estende uma *Activity* simples mas sim uma *ListActivity*. Os métodos existentes são:

- **onCreate**: cujo *layout* está definido em *care_plans_list.xml*.
- **fillPlans**: efectua o preenchimento da lista com os Planos de Cuidados existentes no dispositivo.
- **onCreateOptionsMenu**: a partir deste menu de opções o utilizador pode optar por efectuar o **Logout** ou ir para a *Activity CarePlansUpload*. Este menu está definido em *menu_plans_list.xml*.
- **onListItemClick**: método utilizado ao clicar num dos Planos de Cuidados listados. Inicia a *Activity CarePlanView*.
- **onMenuItemSelected**
- **onCreateDialog**
- **showLogoutDialog**

A classe **CarePlanView**, que estende a classe *TabActivity*, é utilizada para exibir o **Menu Plano de Trabalho** de cada doente. Esta classe serve de *contentor* para as classes **TabActividades** e **TabDiagnosticos**. Os métodos pertencentes a esta classe são:

- **onCreate**: cujo *layout* está definido em *care_plan_view.xml*.
- **setupTabHost**: inicializa o *TabHost* que irá conter as classes **TabActividades** e **TabDiagnosticos**.
- **addActivityTab**: adiciona uma *tab* ao *TabHost*. Neste caso irão ser adicionadas duas *tabs*.
- **createTabView**: implementa uma *View* que irá corresponder a cada uma das *tabs*.
- **fillData**: constrói o *cabecalho* desta *Activity*.

As classes **TabDiagnosticos** e **TabActividades**, são idênticas. A primeira é utilizada para representar os Diagnósticos de Enfermagem enquanto a segunda representa as Actividades. Ambas estendem a classe *ExpandableListActivity*.

- **onCreate**: cujo *layout* está definido em *care_plans_list2.xml*.
- **fillData**: método utilizado para obter os Diagnósticos ou Actividades, consoante a classe, e as acções correspondentes. É o único método cuja implementação é distinta entre as duas classes. No caso da classe **TabDiagnosticos** os valores do primeiro nível da lista correspondem aos nomes dos Diagnósticos de Enfermagem existentes no Plano de Cuidados do doente, sendo, portanto distintos entre cada um dos doentes. Relativamente à classe **TabActividades** os valores do primeiro nível são constantes, pois correspondem a cada um dos grupos existentes de Actividades: *Diagnósticas*, *Não Diagnósticas* e *Interdependentes*. Se no primeiro caso os valores são obtidos a partir de uma consulta à Base de Dados, no segundo são valores estáticos.
- **onChildClick**: método invocado ao seleccionar uma das Acções existentes no Plano de Trabalho. Inicia a *Activity* **ActionDetails**.
- **onCreateOptionsMenu**: a partir deste menu de opções o utilizador pode optar por efectuar o **Logout**, **Configurar Webservice**, enviar Planos de Cuidado, opção **Enviar** que exhibe um sub-menu onde o utilizador pode seleccionar a opção **Actual** ou **Vários** ou então voltar ao **Início**. Este menu está definido em *menu_plan_details.xml*.
- **showSendDialog**: *Dialog* exibido quando o utilizador selecciona a opção **Enviar** -> **Actual**, estando o *Layout* definido em *upload_dialog.xml*.
- **onCreateDialog**
- **showLogoutDialog**
- **wsConfigurationDialog**

A classe **ActionDetails** permite ao utilizador confirmar as Acções existentes no Plano de Trabalho, registando informações resultantes da prestação de Cuidados relacionada

com essa Acção. Se já tiver sido efectuado alguma confirmação desta acção permite ao utilizador consultar esses dados.

- **onCreate**: cujo *layout* está definido em *action_details_view_2.xml*.
- **fillData**: este método é utilizado para personalizar esta janela consoante a Acção que tenha sido seleccionada no **Menu Plano de Trabalho** com o nome e estado actual. Caso já exista alguma confirmação realizada sobre esta Acção, a informação relacionada com a mesma é exibida: Estado da Confirmação, Hora da Confirmação, Nota associada, Objecto Clínico (caso possua) e, no caso de ter sido confirmada como *Não Realizada*, o motivo pelo qual não foi realizada.
- **getActionType**: utilizado para saber se a Acção possui algum Objecto Clínico (Transacção ou Anotação).
- **getActionTime**: caso a Acção já esteja confirmada, obtém a hora de confirmação.
- **cleanNote**: apaga a Nota Associada.
- **updateActionDetails**: guarda na Base de Dados os dados do registo da confirmação.
- **showDialogAnot**: *Dialog* exibido caso se confirme a Acção como Realizada e esta tiver associado um Objecto Clínico do tipo Anotação.
- **showNotAccomplishedMotiveDialog**: *Dialog* exibido caso se confirme a Acção como Não Realizada.
- **confirmActionDetails**: método que contém a lógica associada à confirmação da Acção. Consoante o tipo de Acção e o tipo de confirmação pode ser invocado o método *showDialogAnot* ou o método *showNotAccomplishedMotiveDialog*, ser iniciada a *Activity TxObx* (quando a Acção é confirmada e possui um Objecto Clínico do tipo Transacção) ou voltar para a *Activity CarePlanView*.
- **onCancelDetails**: utilizado quando se pretende voltar ao **Menu Plano de Trabalho** sem se efectuar qualquer confirmação.

Quando uma Acção possui um Objecto Clínico do tipo Transacção, ao se realizar a confirmar dessa Acção como Realizada é iniciada a *Activity TxObx*. Esta *Activity* estende a classe *ListActivity* e lista todas as Observações Clínicas que pertencem à Transacção:

- **onCreate**: cujo *layout* está definido em *tx_obx.xml* .
- **fillPlans**: lista as Observações Clínicas correspondentes à Transacção.
- **txAssign**: guarda na Base de Dados os registos referentes às Observações Clínicas introduzidas.

- **txCancel**: cancela a inserção de Observações Clínicas, voltando para o **Menu Confirmação de Acção**.
- **showNumberInputDialog**: *Dialog* exibido quando a Observação Clínica é do tipo *Inteiro* ou *Decimal*. O *layout* respectivo está definido no ficheiro *obx_int_type*.
- **showDialogObservation**: *Dialog* exibido quando a Observação Clínica é do tipo *Texto*. Este *Dialog* instancia a subclasse *AlertDialog*.
- **showDialogTime**: *Dialog* exibido quando a Observação Clínica é do tipo *Hora*. Este *Dialog* instancia a subclasse *TimePickerDialog* padrão do Android.
- **showDicDialog**: *Dialog* exibido quando a Observação Clínica é do tipo *Dicionario*. Este *Dialog* instancia a subclasse *AlertDialog*.
- **showBoolDialog**: *Dialog* exibido quando a Observação Clínica é do tipo *Texto*. Este *Dialog* instancia a subclasse *AlertDialog*.
- **onListItemClick**: ao clicar numa Observação, consoante o seu tipo, é exibido um dos *Dialogs* mencionados.

O envio dos Planos de Cuidados, com as confirmações realizadas, é efectuado a partir do **Menu de Envio**, *Activity CarePlansUpload*. Esta *Activity* estende a subclasse *ListActivity* e possibilita escolher quais os Planos de Cuidados a enviar, aparecendo seleccionados, por defeito, aqueles que possuam Acções confirmadas.

- **onCreate**: cujo *layout* está definido em *care_plans_upload.xml*.
- **onlyWithActionsConfirmations**: método utilizado para obter os Planos de Cuidados que possuem Acções confirmadas.
- **fillPlans**: método utilizado para popular a lista com os doentes com Planos de Cuidados no dispositivo.
- **onListItemClick**: este método é invocado ao clicar num doente da lista. É utilizado para marcar e desmarcar como Plano de Cuidados a ser enviado.
- **onCreateOptionsMenu**: este menu de opções permite ao utilizador optar por efectuar o **Logout**, **Configurar Webservice** ou aceder ao **Menu de Listagem dos Planos de Cuidados**. Este menu está definido em *menu_upload.xml*.
- **selectAll**: selecciona todos os doentes listados.
- **unselecAll**: desmarca todos os doente listados.
- **upload**: invocado para se efectuar o envio dos Planos de Cuidados. Se não existirem Planos seleccionados o utilizador é informado do mesmo, se existirem é exibido o *Dialog showSendDialog*.

- **cancelButton**: permite ao utilizador cancelar o envio dos Planos de Cuidados e retornar ao **Menu HCIS**.
- **onCreateDialog**
- **showLogoutDialog**
- **showSendDialog**
- **wsConfigurationDialog**

Para além destas classes referentes a *Activities* existem outras três classes no *package ui*. Essas classes, como foi referido anteriormente, têm como objectivo personalizarem alguns componentes gráficos.

- **MenuBackground**: esta classe é utilizada para alterar o *background* dos vários Menus de Opções existentes na aplicação.
- **CustomExpandableListAdapter**: nas **TabDiagnosticos** e **TabActividades** as linhas correspondentes às Acções são implementadas recorrendo a esta classe, que possibilita alterar a cor da descrição das Acções consoante o seu estado: Sem Confirmação (estado inicial), Realizada, Não Realizada ou Anulada.
- **CustomSimpleCursorAdapter**: à semelhança da classe descrita anteriormente, esta classe personaliza as linhas referentes aos Planos de Cuidados no **Menu de Envio**, possibilitando marcar e desmarcar os Planos consoante se se desejam, ou não, enviar.

7.3 Interface Gráfica

Um dos principais objectivos de uma aplicação é que a sua *interface gráfica* não seja um entrave ao sucesso da mesma, muito antes pelo contrário. Uma boa interface gráfica deve ser o mais simples possível e que possua uma boa usabilidade, permitindo ao utilizador saber o que fazer e como o fazer no mínimo tempo possível. Para tentar minimizar o tempo de aprendizagem e de habituação do utilizador à aplicação optámos por implementar um *look and feel* gráfico e *workflow* semelhante ao existente no HP-HCIS, adaptando-o ao dispositivo Android.

Durante a implementação foram utilizados *widgets* disponibilizados pelo Android. Como os *widgets* padrão não permitiam recriar o *look and feel* HP-HCIS, foi necessário personalizar os mesmos.

7.3.1 Cores

Para se conseguir um *look and feel* idêntico ao HP-HCIS era imprescindível que as tonalidades HP-HCIS estivessem reflectidas no Hcis Mobile. Para conseguirmos obter essas tonalidades captámos *Screenshots* do HP-HCIS dos quais obtemos a codificação HTML das cores. Para efectuar a captação dessas tonalidades e respectiva conversão utilizámos a opção **Color Picker Tool** do software **GIMP**⁵.

À semelhança do que acontece com as *Strings*⁶ é possível associar uma determinada cor, na codificação HTML respectiva, com um *Id* que é posteriormente utilizado no desenvolvimento. Assim, caso pretendamos mudar uma tonalidade, basta alterar o valor do *Id* respectivo e a alteração reflecte-se em todos os contextos onde o mesmo é utilizado. O ficheiro que contém essa associação é o *color.xml* e encontra-se na directoria *res/values/*, sendo o seu conteúdo o seguinte:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="azul">#0000FF</color>
  <color name="cinzento">#bebebe</color>
  <color name="verde_mar">#6AFB92</color>
  <color name="preto">#000000</color>
  <color name="vermelho">#FF0000</color>
  <color name="branco">#FFFFFF</color>
  <color name="verde">#00FF00</color>
  <color name="amarelo">#ffff00</color>
  <color name="azul_hp">#CED7E7</color>
  <color name="azul_separador">#336699</color>
  <color name="azul_seleccao">#8EB2E4</color>
  <color name="transparent">#00000000</color>
  <color name="laranja_anulada">#FFFF6A00</color>
  <color name="verde_confirm">#FF008000</color>
  <color name="azul_separador_trans">#7F336699</color>
  <color name="azul_seleccao_trans">#7F8EB2E4</color>
</resources>
```

7.3.2 Botões

Após termos definido os tons a utilizar na aplicação, foi necessário personalizar os botões a usar. Por defeito os botões do Android possuem um tom cinzento e ao clicar apresentam um tom alaranjado. Esse esquema de cores, na maioria dos casos não nos agradava, pelo que definimos *estilos* para utilizar nos botões. Ao utilizar estilos, em vez de personalizar botão a botão, temos a vantagem de conseguir manter uma coerência entre todos os botões com funcionalidades semelhantes, pois os mesmos mantêm a cor, forma e tamanho iguais,

⁵GNU Image Manipulation Program: programa de código aberto utilizado para criação e edição de imagem.

⁶Ver Subsecção 7.4.1.

sendo possível ainda assim personalizar cada um deles com texto ou imagens referentes à sua função. No total implementámos seis estilos diferentes de botões:

- **HCISButton**: estilo utilizado pela maioria dos botões da aplicação.
- **NHCButton**: estilo utilizado pelo botão referente ao Carregamento de Planos de Cuidados por NHC, no **Menu HCIS**.
- **ROOMButton**: estilo utilizado pelo botão referente ao Carregamento de Planos de Cuidados por Quarto, no **Menu HCIS**.
- **UNITButton**: estilo utilizado pelo botão referente ao Carregamento de Planos de Cuidados por Unidade de Enfermaria, no **Menu HCIS**.
- **UPLOADButton**: estilo utilizado pelo botão referente ao Envio de Planos de Cuidados, no **Menu HCIS**.
- **VIEWButton**: estilo utilizado pelo botão referente à Listagem de Planos de Cuidados, no **Menu HCIS**.

As definições de cada um destes estilos encontram-se no ficheiro *style.xml*, na directoria *res/values/*. Ao utilizar estilos é-nos possível, também, controlar o *aspecto* do botão consoante o seu estado. Por estado entendemos se foi ou não clicado pelo utilizador. Esse controlo do aspecto consoante o estado é implementado recorrendo a um *drawable* do tipo **StateListDrawable**.

Como exemplo apresentamos o estilo *HCISButton*⁷:

```
<style name="HCISButton" parent="@android:style/Widget.Button">
  <item name="android:gravity">center_vertical|center_horizontal</item>
  <item name="android:textColor">@color/branco</item>
  <item name="android:shadowColor">@color/cinzeno</item>
  <item name="android:shadowDx">0</item>
  <item name="android:shadowDy">-1</item>
  <item name="android:shadowRadius">0.2</item>
  <item name="android:textSize">14dip</item>
  <item name="android:textStyle">bold</item>
  <item name="android:background">@drawable/hcis_button_selector</item>
  <item name="android:focusable">>true</item>
  <item name="android:clickable">>true</item>
</style>
```

Neste estilo o *background* é definido pelo *drawable* **hcis_button_selector**, sendo a definição deste a seguinte:

⁷Como se pode verificar, as cores utilizadas na definição deste estilo correspondem às definidas no ficheiro *color.xml* descrito anteriormente.

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:state_pressed="true"
    android:state_enabled="true"
    android:drawable="@drawable/hcis_button_selected" />
  <item
    android:state_focused="true"
    android:state_enabled="true"
    android:drawable="@drawable/hcis_button_selected" />
  <item
    android:state_enabled="true"
    android:drawable="@drawable/hcis_button" />
</selector>

```

Como é possível de constatar, consoante o estado do botão é exibido um *drawable* diferente. Neste caso cada um destes *drawables* corresponde ao tipo **Shape**. Como exemplo apresentamos a definição do **hcis_button**:

```

<shape xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape="rectangle">
  <corners
    android:radius="10dp"
    android:topLeftRadius="10dp"
    android:topRightRadius="10dp"
    android:bottomLeftRadius="10dp"
    android:bottomRightRadius="10dp" />
  <solid
    android:color="@color/azul_separador" />
  <stroke
    android:width="1dp"
    android:color="@color/branco" />
  <padding
    android:left="5dp"
    android:top="5dp"
    android:right="5dp"
    android:bottom="5dp" />
</shape>

```

A única diferença entre os *drawables* **hcis_button.xml** e **hcis_button_selected.xml** é referente à cor do atributo *android:color* no elemento *<solid>*, como se pode verificar no exemplo do botão **Login**, Fig. 7.1. O botão quando está Activo, sem estar pressionado, é definido por **hcis_button.xml** e apresenta uma cor azul mais escura, quando é seleccionado é definido por **hcis_button_selected.xml** e apresenta uma cor azul mais clara.

A estrutura dos restantes estilos implementados é semelhante à descrita neste exemplo, tendo apenas uma pequena alteração. Os itens de cada *StateListDrawable* que define o *background* não referem uma *Shape* mas sim uma imagem. Na Fig. 7.2 podem ser vistos



Figura 7.1: Botão Login

os botões que correspondem ao Carregamento, Listagem e Envio dos Planos de Cuidados, onde o botão superior corresponde ao botão activo e o botão inferior corresponde ao botão seleccionado.

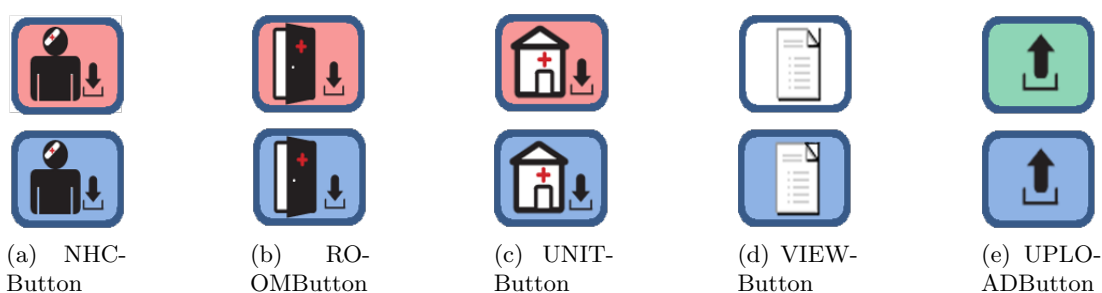


Figura 7.2: Estilo dos Botões do Menu HCIS

Todos estes cinco estilos apenas são utilizados no **Menu HCIS**. Os componentes *drawable* de cada um destes estilos pode ser consultado na tabela 7.1. Todos os ficheiros *drawable*, independentemente de serem ficheiros XML ou imagens, encontram-se na directoria *res/drawable/*.

Estilo	StateListDrawable	Imagem	
		Activo	Pressionado
NHCButton	button_nhc_selector	nhc_button	nhc.button_select
ROOMButton	button_room_selector	room_button	room.button_select
UNITButton	button_unit_selector	unit_button	unit.button_select
UPLOADButton	button_upload_selector	upload_button	upload.button_select
VIEWButton	button_view_selector	view_button	view.button_select

Tabela 7.1: Componentes de cada *estilo* definido para os botões

7.3.3 Menus

Nesta secção iremos descrever os vários menus existentes na aplicação relativamente à sua componente visual, mais propriamente à definição do seu *layout* definido nos ficheiros XML respectivos. Todos os ficheiros que definem os *layouts* encontram-se na directoria *res/layout*. Alguns destes menus sofreram alterações gráficas ao longo da evolução do projecto, pelo que nesses casos iremos apresentar essa mesma evolução. Essa evolução

deveu-se não só à adaptação do *look and feel* como também à necessidade de potenciar mais funcionalidades ao mesmo.

Menu Login

Apesar de ser o primeiro menu da aplicação, foi o último a ser desenvolvido. Devido a isso não existiu uma evolução significativa do seu layout pois já adoptou todo o *look and feel* definido para a aplicação e as funcionalidades para o mesmo já estavam bem definidas. O *layout* encontra-se definido no ficheiro *login.xml*.

Estruturalmente este menu está dividido em três secções, possui dois *RelativeLayouts* e um *LinearLayout*. No primeiro *RelativeLayout* encontra-se um widget do tipo *ImageView*, que possui como imagem o logotipo do HCIS. Este *RelativeLayout* situa-se na parte superior da janela. O *LinearLayout* localizado na parte inferior do ecrã apenas possui um *widget ButtonView*, que utiliza o estilo *HCISButton*. Na parte central da janela encontra-se o outro *RelativeLayout*, que é composto por sete *widgtes* de três tipos.

Possui três *TextView* utilizados para exibir texto, dois *EditText* utilizados para introdução de texto, neste caso concreto o **utilizador** e a **chave**, e um *Spinner* que permite escolher o hospital a que o utilizador se pretende ligar. Todos estes *widgtes* possuem um *Id* específico que permitem ser referenciados em qualquer parte do código, desde que referente ao contexto do menu em causa.

Menu HCIS

O *layout* deste menu corresponde ao ficheiro *hphcis.xml*. É composto por um *LinearLayout* e um *RelativeLayout*. No *LinearLayout* está implementado um *TextView* que possui o texto *Cuidados de Enfermagem*. O *RelativeLayout* implementado contém os outros *widgtes* que se podem visualizar: os cinco botões descritos em 7.3.2, um *ImageView* com o logotipo da HP e seis *TextView* que contém os restantes textos exibidos.

Este menu foi um dos que mais modificações teve no decorrer do desenvolvimento da aplicação. Inicialmente possuía apenas quatro botões, tendo no *background* o logotipo da HP. Esses botões não possuíam nenhum estilo associado. A segunda versão do Menu HCIS é já bastante idêntica à final, possuindo os cinco botões com um estilo semelhante, mas



Figura 7.3: Menu de Login

ao invés de possuírem imagens continham texto indicativo sobre a sua funcionalidade. O fundo desse menu já não continha o logótipo da HP mas sim uma outra imagem, que no entanto ainda não fornecia um *look and feel* HP-HCIS. Esta evolução é visível na Fig. 7.4.



Figura 7.4: Evolução da interface do Menu HCIS

Menu Listagem dos Planos de Cuidados

Este menu é dos mais simples da aplicação Hcis Mobile. Apenas possui na sua interface três *widgets*. Um *TextView* onde é exibido o título deste menu, **Planos de Cuidados**, uma *View* que utilizámos como separador de uma *ListView*, onde são exibidos os Planos de Cuidados.

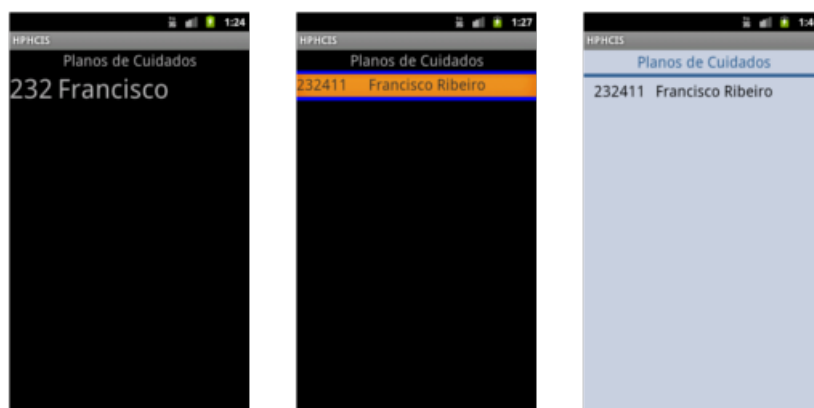


Figura 7.5: Evolução da interface do Menu Listagem dos Planos de Cuidados

Cada linha da *ListView* é, no entanto, definida no ficheiro *care_plans_line.xml*. Foi necessário a utilização desta definição de cada linha de modo a podermos personalizar o conteúdo e *background* da mesma. Em cada linha são utilizados dois *widgets TextView*, um para o NHC e outro para o nome do doente. Por sua vez o *background* é definido por um *drawable* do tipo *StateListDrawable*, que permite alterar a cor da mesma quando é pressionada. Em vez de exibir o tom laranja, típico do Android, é exibida a cor *azul_selecao* de-

finida em *color.xml*. O ficheiro onde está implementada esta personalização é o *list.bg.xml*. Este menu sofreu também algumas alterações no decorrer do desenvolvimento, principalmente ao nível das tonalidades do mesmo. Essa evolução é perceptível na Fig. 7.5.

Menu Plano de Trabalho

Este menu, em termos de interface, foi o que mais dúvidas suscitou durante o desenvolvimento. Devido à existência de dois grandes grupos de Acções de Enfermagem, as Acções pertencentes a Diagnósticos e as Acções pertencentes a Actividades, que, por sua vez, se dividem em três subgrupos, Diagnósticas, Não Diagnósticas e Interdependentes, era necessário encontrar uma forma simples e clara de as exibir ao utilizador. Inicialmente estudámos a hipótese de as exibir todas, utilizando um esquema de árvore, no entanto essa hipótese foi colocada de lado, pois no caso de Planos de Cuidados que possuíssem muitas Acções, ao exibi-las desse modo, e tendo em conta o tamanho do visor do dispositivo móvel, o utilizador apenas ia ver uma ínfima parte. Por exemplo, se se expandisse o ramo relativo aos Diagnósticos, caso precisasse de verificar quais as Acções Interdependentes, que ficariam no último ramo da árvore, era necessário *deslizar* bastante a área exibida para alcançar esse subgrupo. Optámos então por implementar um sistema de *tabs*. Desta forma, o utilizador, conseguiria aceder fácil e rapidamente a qualquer um dos grupos de Acções. Após a decisão de implementar o sistema de *tabs* foi necessário decidir a organização do mesmo.

Inicialmente implementámos um sistema de quatro *tabs*: Diagnósticos, Actividades Diagnósticas, Actividades Não Diagnósticas e Actividades Interdependentes. Para exibir as Acções pertencentes a Diagnósticos decidimos utilizar um sistema de árvore, onde o primeiro nó correspondia ao Diagnóstico cujos filhos eram as Acções correspondentes. No caso das outras acções, implementámos um sistema de lista, onde eram exibidas todas as Acções pertencentes a cada um dos grupos. No entanto essa solução não nos pareceu a mais indicada, pois o nome de cada grupo não era totalmente visível nas *tabs*. Optámos, então, por implementar um sistema de duas *tabs*, que permitia dividir os grupos de Acções em apenas dois: Diagnósticos e Actividades. Para representar cada um dos subgrupos de Actividades fizemos a analogia com o sistema de árvore implementado para os Diagnósticos, sendo o primeiro nó de cada subgrupo o nome do mesmo: Diagnósticas, Não Diagnósticas e Interdependentes. Para além da divisão das Acções, decidimos colocar a informação relativa ao doente na janela, para permitir, ao utilizador, ter a certeza que estava a visualizar o Plano de Trabalho correcto. A evolução das várias interfaces está esquematizada na Fig. 7.6

O layout final está descrito no ficheiro *care.plan.view.xml*. Este *layout* é implementando através de um *TabHost*, para permitir a utilização de *tabs*. o *TabHost* possui um único elemento, um *LinearLayout*, que por sua vez possui um *RelativeLayout*, um *TabWidget*, uma *View* e um *FrameLayout*. O *RelativeLayout* corresponde à secção com a informação do doente em causa. É composto por um *TextView*, onde é colocado o NHC e nome



Figura 7.6: Evolução da interface do Menu Plano de Trabalho

do doente, e por um *ImageView*, no qual é exibido o ícone que permite identificar o sexo e idade do doente. o *TabWidget* corresponde à implementação das *tabs*, cuja personalização está definida no ficheiro *tabs_bg.xml*. este ficheiro usa dois *StateListDrawable* para alterar o *background* e a cor do texto existente em cada *tab* consoante o estado da mesma. Os *StateListDrawable* são o *tab_bg_selector.xml* e o *tab_text_selector*. A *View*, referida anteriormente, é utilizada para exibir um separador entre as *tabs* e o conteúdo de cada uma delas, que se encontra dentro do *RelativeLayout*. O conteúdo de cada uma das *tabs*, definido em *care_plans_list2.xml* possui uma *widget* do tipo *ExpandableListView*, que implementa uma lista expansível, a árvore referida anteriormente. O *widget ExpandableListView* encontra-se, igualmente, personalizado. A personalização do elemento pai está descrito no ficheiro *care_plans_group_name.xml* que utiliza o *StateListDrawable list_bg.xml*, e cada um dos elementos filho está definido no ficheiro *care_plans_children_info.xml* que possui dois *TextView*, um para o nome da Acção e outro para o tipo de Acção (contínua ou periódica) e cujo *background* encontra-se igualmente definido por *list_bg.xml*. Outro factor que é importante referir está relacionado com a cor do nome de cada Acção. Consoante o seu estado de confirmação a cor é alterada: se estiver confirmada como realizada é exibida com a cor verde, se estiver confirmada como não realizada é exibida com a cor vermelha, se, por sua vez, tiver sido anulada uma confirmação é exibida com a cor laranja. Esta gestão de qual a cor apresentada é efectuada pela classe *CustomExpandableListAdapter*, como referido em 7.2.4.

Menu Confirmação de Acções

No *Menu Confirmação de Acções* o utilizador pode confirmar uma acção como Realizada, Não Realizada ou Anular uma confirmação. Para implementarmos esta interface recorreremos ao *widget RadioGroup*.

Esta interface possui dois *RelativeLayout*. No primeiro, estão contidos dois *TextView*, onde um possui o nome da Acção e outro a **label Nota:**, uma *View* que é utilizada como separador, um *EditText* onde se pode introduzir uma breve nota associada à confirmação

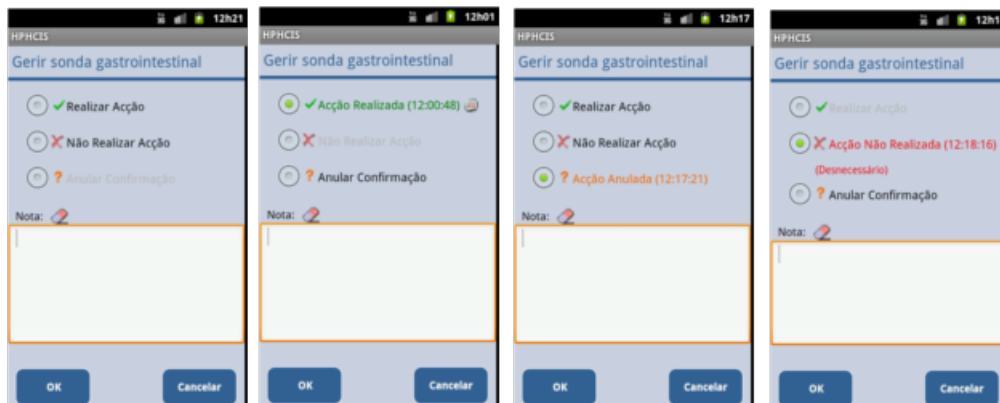


Figura 7.7: Diferentes visualizações do Menu Confirmação de Acções

e dois *ImageView*, sendo que um deles possibilita apagar o conteúdo do *EditText* referido anteriormente e o outro só é exibido quando a Acção está confirmada e possui um *Objecto Clínico* associado. Para além destes componentes, o primeiro *RelativeLayout* contém o já referido *RadioGroup*. Dentro deste grupo estão definidos três *RadioButtons*, sendo que o primeiro corresponde ao **Realizar Acção**, o segundo ao **Não Realizar Acção** e o terceiro ao **Anular Confirmação**. Estes *RadioButtons*, para além da *label* correspondente, possuem, ainda, três ícones indicativos da sua funcionalidade. Para o caso de se efectuar uma confirmação de *Não Realizar Acção* existe ainda outro *TextView* que é exibido com a indicação do motivo da não realização. O segundo *LinearLayout* apenas compreende dois *Buttons*, que são utilizados para confirmar ou cancelar o registo da confirmação.

Consoante o estado da Acção, para além das imagens e informações que devem ser ou não visíveis, as *labels* dos *RadioButtons* sofrem, igualmente, alterações. Toda essa contextualização é efectuada pela *Activity ActionDetails*⁸ e pode ser vista na Fig. 7.7.

Menu Observações Clínicas

Ao se confirmar uma Acção como Realizada, e caso esta possua um *Objecto Clínico* do tipo *Transacção*, é exibido o **Menu Observações Clínicas**. Este menu não é mais que uma listagem de todas as *Observações Clínicas* correspondentes à *Transacção* em questão. O *layout* desta janela está definido em *tx_obx.xml*. Este *layout* foi implementado recorrendo a um *TextView*, que corresponde ao nome da *Transacção*, três *Views* que são utilizados como separadores, um *LinearLayout*, um *RelativeLayout* e uma *ListView* que irá conter as várias *Observações Clínicas*. O *LinearLayout* contém dois *TextViews* que correspondem ao nome das colunas da lista, enquanto o *RelativeLayout* contém dois *Buttons* utilizados para fechar ou cancelar o registo das *Observações Clínicas*. A personalização das linhas desta lista é efectuada com base no ficheiro *obx.line.xml* possuindo, o mesmo, três *TextViews*, sendo a primeira utilizada para exibir o nome da *Observação Clínica*, a seguinte para exibir

⁸Ver Subsecção 7.2.4.

o valor registado e a última para mostrar as unidades em que esse valor foi registado. O *background* da linha, como na grande maioria das listas existentes, é implementado recorrendo ao *StateListDrawable list_bg.xml*. Na Fig. 7.8 é possível visualizar este menu, sem e com Observações Clínicas registadas.



Figura 7.8: Menu Observações Clínicas

Menu de Envio

É recorrendo às funcionalidades disponíveis neste menu que o utilizador pode efectuar o envio de todos os Planos de Cuidados com os registos efectuados. A interface deste menu está implementada através do ficheiro *care_planes_upload.xml* e é composta por um *LinearLayout*, dois *RelativeLayouts* e uma *ListView*. O *LinearLayout* possui um *TextView*, com a identificação do menu, e uma *View*, que serve de separador. Os *RelativeLayouts* são utilizados para armazenar os *Buttons*, possuindo dois cada um. No *RelativeLayout* situado na parte superior do menu estão os botões que permitem marcar ou desmarcar todos os Planos de Cuidados listados, no *RelativeLayout* situado na parte inferior estão os botões que efectuam o envio ou cancelam esta opção, retornando para o Menu HCIS. Relativamente à *ListView*, esta irá conter todos os Planos de Cuidados existentes no dispositivo. Cada linha desta lista é personalizada pelo *layout* descrito em *care_plans_upload_line.xml*, possuindo dois *TextView*, que irão conter o NHC e o nome do doente, e uma *CheckBox*, que servirá para marcar quais os Planos de Cuidados que se desejam enviar. Uma ilustração deste menu está disponível na Fig. 7.9, onde umas das imagens possui um Plano de Cuidados seleccionado e a outra não apresenta qualquer selecção.

7.3.4 Menus de Opções

Como referido em 6.3.6, utilizámos o Menu de Opções do Android para, permitir ao utilizador, aceder a outras funcionalidades que não estivessem disponíveis no menu em questão. Por exemplo, configurar o *Web Service*, efectuar *Logout* ou, inclusive, aceder a

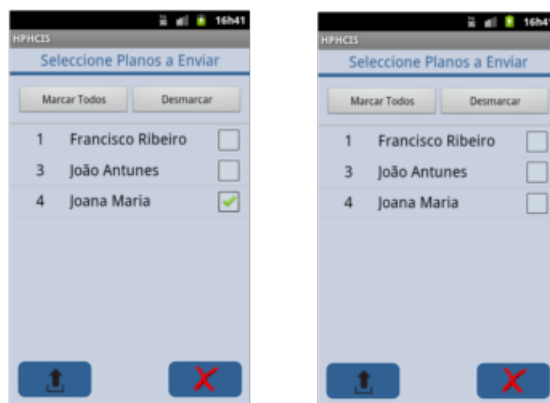


Figura 7.9: Menu de Envio

outros menus são algumas das funcionalidades que se podem aceder a partir do Menu de Opções.

Cada Menu de Opções possui uma definição própria, que indica quais as opções definidas pelo mesmo. Essas definições estão implementadas no formato XML, ficando os ficheiros relativos às mesmas na directoria *res/menu/*. Na Tabela 7.2 está representada qual a definição do Menu de Opções consoante o menu do Hcis Mobile.

Menus Hcis Mobile	Menus De Opções
Menu Login	menu_login.xml
Menu HCIS	menu_home.xml
Menu Listagem dos Planos de Cuidados	menu_plans_list.xml
Menu Plano de Trabalho	menu_plan_details.xml
Menu Envio	menu_upload.xml

Tabela 7.2: Correspondência entre os Menus Hcis Mobile e os Menus de Opções

Devido ao facto da aparência padrão do Menus de Opções, disponibilizadas pelo Android, não nos agradar, decidimos personalizar o mesmo. Essa personalização incide no *background* do mesmo tornando-o mais transparente e com uns tons azulados. Para tal implementámos o *StateListDrawable menu_button_selector.xml*. Consoante o estado do botão, se está ou não pressionado, é exibido um *drawable* do tipo *Shape* diferente, respectivamente os *drawables menu_button_selected.xml* ou *menu_button.xml*. A aparência do Menu de Opções está visível na Fig. 7.10, que corresponde ao Menu de Opções disponível no Menu Plano de Trabalho. Neste caso o botão referente à opção *Início* encontra-se pressionada.

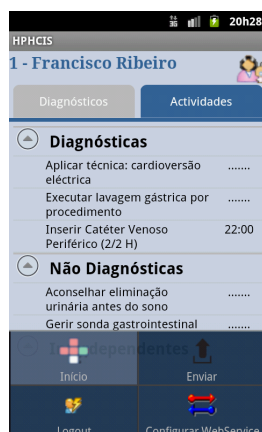


Figura 7.10: Menu de Opções

7.4 Internacionalização

Ao longo do desenvolvimento desta aplicação tivemos sempre presente características referentes à internacionalização da mesma. Um dos factores mais determinantes no sucesso de internacionalização de uma aplicação móvel é a capacidade que possui de se *adaptar* ao seu contexto, principalmente ao idioma dos seus utilizadores. Para além do idioma, tivemos em atenção que a realidade da Enfermagem não é igual a nível global. Outros países utilizam outros tipos de Linguagens de Enfermagem diferentes da CIPE, pelo que a estrutura interna de dados foi implementada com a intenção de suportar outro tipo de Linguagem de Enfermagem, no caso o NANDA NIC/NOC.

7.4.1 Idioma

o Hcis Mobile suporta três idiomas diferentes: Português, Castelhana e Inglês. Consoante o idioma do dispositivo a aplicação *adapta-se* ao mesmo, sendo o Inglês o idioma escolhido para *default*. Ou seja, quando o idioma do dispositivo não corresponde a nenhum dos três mencionados anteriormente é utilizado o Inglês.

Como já foi referido em 2.3.6, o Android permite realizar essa tarefa de um modo bastante simples e prático. Durante o desenvolvimento da aplicação todos as *Strings* utilizadas foram colocadas no ficheiro *strings.xml*, localizado na directoria **res/values/**, e para utilizá-las recorreremos ao seu **Id** associado. Para procedermos à internacionalização apenas foi necessário criar as directorias *values-pt* e *values-es*, igualmente dentro da directoria *res* e, dentro de cada uma delas, foi criado um novo ficheiro *strings.xml* que contém todas as *Strings* utilizadas na aplicação traduzidas em Português e Castelhana, respectivamente. Na directoria *values* ficou o ficheiro que contém as *Strings* em Inglês. Desta forma garantimos que sempre que o dispositivo esteja em Português ou Castelhana as mensagens exibidas estão no idioma correspondente. Caso o idioma seja outro distinto deste é utilizado o ficheiro existente na directoria *default* que faz o mapeamento entre os

Id's colocados no código e os textos que devem ser exibidos. Caso se deseje implementar outro idioma, basta traduzir o conteúdo do ficheiro *strings.xml* e colocá-lo na directoria *values-"idioma"*⁹ correspondente. Com isto evitamos ter que percorrer o código todo e substituir as mensagens pelas suas traduções. É de salientar que apenas as mensagens da aplicação possuem tradução, toda a informação existente na Base de Conhecimento faz parte de configuração da aplicação, logo o idioma em que está essa informação depende de quem a introduz e não da aplicação.

7.4.2 Estrutura Interna de Dados

Para além da preocupação referente ao idioma, durante o desenvolvimento tivemos em atenção outras realidades de Enfermagem existentes. A Linguagem CIPE não é, por enquanto, adoptada em todos os países. Por exemplo em Espanha é utilizada o sistema de classificação NANDA NIC/NOC. Uma das principais diferenças de um para outro está relacionada com a estrutura dos Diagnósticos. Na realidade CIPE estes são constituídos por Acções/Intervenções¹⁰, no entanto na realidade NANDA NIC/NOC um Diagnóstico possui várias Intervenções que por sua vez, cada uma delas, possuem várias Acções. Se num contexto CIPE se pode associar directamente uma Acção a um Diagnóstico, no caso NANDA NIC/NOC tal já não é possível pois existe um nível intermédio, o nível das Intervenções. Com base nisto, a estrutura da Base de Dados e o WSDL do Web Service contentam o nível das Intervenções, assim, caso se esteja numa realidade NANDA NIC/NOC, a aplicação já está preparada para tratar essa estrutura. Na realidade CIPE o nível das Intervenções não é mais que um elo de ligação entre um Diagnóstico e as Acções que o compõem, existindo uma ligação de 1:1 entre uma Acção e uma Intervenção. Ao termos estruturado internamente a informação deste modo apenas é necessário efectuar uma pequena alteração na camada de apresentação caso se pretenda utilizar a aplicação num contexto NANDA NIC/NOC, pois no **Menu Plano de Trabalho** a *tab* respeitante aos Diagnóstico apenas possui dois níveis, sendo necessário adaptá-la para suportar três níveis.

7.5 Testes

Para comprovar a qualidade da aplicação é necessário efectuar testes da mesma, nomeadamente testes de cariz qualitativo e de usabilidade, em ambientes reais, mais concretamente pelos enfermeiros enquanto prestam os Cuidados de Saúde. No entanto, e devido à não existência do *Web Service* que permitiria a comunicação entre o Hcis Mobile e o HP-HCIS, esses testes não foram realizados.

Apesar dos testes referidos anteriormente, imprescindíveis para aferir a qualidade da aplicação, não terem sido efectuados, foram realizados alguns testes de desenvolvimento,

⁹Após se adicionar o novo idioma é necessário recompilar o código e gerar o **apk**.

¹⁰Segundo a definição CIPE uma Intervenção é uma Acção, ver 3.2.

mais concretamente testes de *stress* e testes funcionais, ambos recorrendo ao simulador Android. Para complementar estes testes foram igualmente realizadas experiências em dois dispositivos físicos.

7.5.1 Testes de *Stress*

O Android SDK permite realizar testes de *stress* através da ferramenta **UI/Application Exerciser Monkey**. Estes testes consistem em enviar um número definido de eventos aleatórios à aplicação com o objectivo de detectar *bugs* e verificar se ocorrem *crashes*.

Foram efectuados cinco testes utilizando esta ferramenta, começando com 500 eventos em que se incrementavam outros 500 a cada novo teste, tendo o último teste 2500 eventos. Durante a realização destes testes foram detectados dois erros. O primeiro estava relacionado com o tipo de dados introduzido quando se tentava carregar os Planos de Cuidados por NHC. Durante o desenvolvimento assumimos que os NHC's eram valores do tipo *inteiro* pelo que nunca fizemos nenhuma validação do seu tipo. Ao inserir um valor que não fosse numérico a aplicação não estava preparada para resolver esse problema, razão pela qual se verificava um *crash*. O segundo problema estava relacionado com o acesso à Base de Dados, num determinado contexto era feita uma chamada à Base de Dados sem inicializar o objecto que faz essa gestão, razão pela qual se verificava outro *crash* da aplicação. Graças a estes testes esses dois problemas foram detectados e corrigidos.

7.5.2 Testes Funcionais

Para garantirmos que todas as funcionalidades da aplicação eram executadas correctamente foi criado um conjunto de testes funcionais. Estes testes foram criados recorrendo aos *frameworks* **JUnit** e **Robotium**. O JUnit facilita a criação de código para efectuar testes na linguagem Java enquanto o Robotium facilita a criação desses mesmo testes pois não é necessário conhecer o código fonte da aplicação. Por exemplo, para verificar se um determinado botão tem o comportamento desejado não é necessário conhecer o **Id** do botão, bastando apenas indicar que se deseja *clicar* no botão que possui o texto *X*.

Estes testes simulam a interacção com a aplicação por parte do utilizador, pelo que em cada um deles é necessário definir quais são as funcionalidades que pretendemos testar, assim como os parâmetros que desejamos que sejam inseridos nos diferentes campos. Para realizar a validação funcional da aplicação implementamos um conjunto testes por cada *Activity* existente no Hcis Mobile.

LoginTest

Verifica as funcionalidades do Menu Login

- **testInputCorrectLogin:** verifica se o Login é feito correctamente, fornecendo dados de Login correctos.
- **testInputInvalidLogin:** verifica se o Login é feito correctamente, fornecendo dados de Login incorrectos.

HPHCISTest

Verifica as funcionalidades do Menu HCIS

- **testLoadNHC:** verifica se é efectuado o carregamento do Plano de Cuidados do doente cujo NHC é fornecido.
- **testLoadNHCNotInsert:** valida o comportamento da aplicação ao se tentar efectuar um carregamento sem fornecer o NHC.
- **testLoadNHCInvalidLogin:** valida o comportamento da aplicação ao se tentar efectuar um carregamento com dados de Login inválidos.
- **testAllButtons:** valida o comportamento de todos os botões existentes no Menu HCIS.
- **testMenu:** valida o comportamento do Menu de Opções existente no Menu HCIS.

CarePlansListTest

Verifica as funcionalidades do Menu Listagem dos Planos de Cuidados

- **testSelectOne:** verifica se é exibido o Menu Plano de Trabalho ao se seleccionar um Plano de Cuidados.

CarePlanViewTest

Verifica as funcionalidades do Menu Plano de Trabalho

- **testTabsAndList:** valida se o comportamento desejado é obtido ao *clicar* nas *tabs* e em elementos da lista de Acções.
- **testMenuLogout:** verifica o comportamento ao se seleccionar a opção *Logout* no Menu de Opções.
- **testMenuWS:** verifica o comportamento ao se seleccionar a opção *Confirmar Web Service* no Menu de Opções.

- **testMenuHome:** verifica o comportamento ao se seleccionar a opção *Índice* no Menu de Opções.
- **testMenuSend:** verifica o comportamento ao se seleccionar a opção *Enviar* no Menu de Opções.

ActionDetailsTest

Verifica as funcionalidades do Menu Confirmação de Acções

- **testConfirmation:** verifica o comportamento de todos os botões existentes neste menu.

UploadTest

Verifica as funcionalidades do Menu Envio

- **testSelectAndUnselectPatientsAndExit:** testa o comportamento dos botões para marcar e desmarcar todos os doentes.
- **testSendWithoutPatients:** valida a resposta quando se tenta proceder ao envio sem ter Planos de Cuidados seleccionados.
- **testSendPatients:** verifica se o envio dos Planos de Cuidados é efectuado correctamente.

WorkFlow

Verifica o *workflow* típico do Hcis Mobile

- **testWorkFlow:** valida o *workflow* típico do Hcis Mobile, desde o Login, passando pela confirmação de Acções com registo de Objectos Clínicos, até efectuar o Logout.

Após realizar várias iterações destes testes, todos terminaram correctamente, pelo que podemos afirmar que, do ponto de vista funcional, a aplicação não demonstrou qualquer problema.

7.5.3 Testes em Dispositivos Físicos

Apesar dos testes no simulador terem detectado poucas falhas, sendo que as detectadas foram corrigidas, nada garante que não existissem falhas ao utilizar a aplicação em dispositivos reais.

Para verificar a sua funcionalidade instalámos o Hcis Mobile em dois dispositivos distintos. O dispositivo **A** possuía a versão Android 2.1 enquanto que o dispositivo **B** tinha a versão 2.3.3. Em ambos a aplicação foi instalada correctamente e inicializada. Com o dispositivo **A** carregamos um Plano de Cuidados, que apareceu listado, no entanto ao tentar aceder ao Menu Plano de Trabalho verificou-se sempre um *crash* da aplicação. Com o dispositivo **B** foram testados um número relevante de circuitos possíveis, tendo sido detectados, em algumas ocasiões, problemas em estabelecer a comunicação com o servidor.

O problema detectado no dispositivo **A** julgamos que esteja relacionado com a versão do Android, pois a aplicação foi desenvolvida para a versão 2.3.1 e esse problema não se verificava no dispositivo **B**. Relativamente ao problema de comunicação, e tendo em conta que os problemas eram esporádicos, julgamos que o mesmo possa estar relacionado com o sinal *wireless* que era de pouca qualidade.

Apesar dos resultados dos testes serem satisfatórios não deixa de ser essencial a aplicação ser testada num contexto real, para aferir a fiabilidade da mesma.

Capítulo 8

Conclusões e Trabalho Futuro

Através deste projecto foi implementada uma aplicação móvel que visa auxiliar a prestação dos Cuidados de Saúde por parte dos enfermeiros permitindo efectuar os registos de Intervenções CIPE durante a prestação dos mesmos. Ao se possibilitar o registo da informação relativa a cada Intervenção CIPE durante a prestação dos Cuidados aumenta-se a fiabilidade da mesma assim como fica disponível mais rapidamente para outros enfermeiros ou médicos, o que possibilita uma melhor prestação de Cuidados de Saúde.

A metodologia utilizada para este projecto veio a verificar-se bastante proveitosa. Graças ao planeamento de todas as funcionalidades que desejávamos implementar, assim como à divisão das mesmas pelos *Sprints*, permitiu que durante a duração de cada *Sprint* o foco do desenvolvimento fosse nos itens pertencentes ao mesmo, o que possibilitou que a implementação da aplicação tivesse demorado o tempo estipulado.

Um dos objectivos desta aplicação prendia-se com o facto de poder ser usada tanto num contexto *online* como *offline*. Para tal ser possível optámos por guardar todos os registos no próprio dispositivo enviando-os apenas quando existisse uma ligação disponível. Assim garante-se que todos os registos são correctamente guardados no HP-HCIS não existindo o risco de ficarem perdidos por não ser possível efectuar a comunicação aquando da introdução dos mesmos.

Um dos possíveis problemas do carregamento dos Planos de Cuidados estava relacionado com a quantidade de informação que era necessária enviar, o que poderia condicionar a qualidade da comunicação e o tempo desejado pelos enfermeiros para o carregamento dos Planos de Cuidados. Muita dessa informação iria ser redundante pois vários planos poderiam conter a mesma Acção que possui características padrão comum a todos os Planos de Cuidados, como o nome, por exemplo. Para contornar esses possíveis problemas

optámos pela criação da Base de Conhecimento que possui toda a informação padrão, sendo apenas necessário ser enviado o código da Acção a partir do qual se obtém todos os dados da mesma. Assim é possível diminuir a quantidade de informação a ser enviada através do *Web Service* e conseqüentemente o tempo de carregamento.

Como um dos principais entraves ao sucesso e boa aceitação de uma aplicação por parte dos seus utilizadores está relacionada com a resistência à mudança, ao implementar esta aplicação com um *look and feel* e um *workflow* idêntico ao já existente no HP-HCIS pretendemos que os enfermeiros vejam esta aplicação como mais um instrumento de auxílio ao seu trabalho e não como algo que os vai destabilizar e causar problemas durante a prestação dos Cuidados de Saúde.

A utilização do Android como S.O. alvo desta aplicação facilitou a implementação do *look and feel* HP-HCIS, pois possui um conjunto de *widgets* bastante vasto que, para além de permitirem vários tipos de interacção entre o utilizador e a aplicação, são facilmente personalizáveis.

Foram também efectuados alguns testes funcionais da aplicação recorrendo ao emulador Android. A partir dos resultados dos mesmo podemos afirmar que a aplicação cumpre todos os requisitos necessários para uma utilização real, sendo no entanto necessário efectuar testes num ambiente real a fim de aferir a fiabilidade e usabilidade da mesma.

8.1 Trabalho Futuro

Apesar do resultado final se ter mostrado positivo é necessário continuar a trabalhar sobre a aplicação para a conseguir tornar um instrumento ainda mais útil aos profissionais de enfermagem. Entre os possíveis trabalhos futuros que se podem efectuar sobre esta aplicação destacamos:

1. **Base de Dados:** apesar da Base de Dados cumprir com os seus objectivos, a mesma pode sofrer uma optimização ao nível do seu *Schema* com vista a melhorar o seu desempenho. Outra funcionalidade que deve ser implementada está relacionada com o carregamento da Base de Conhecimento. Actualmente apenas é possível carregar a Base de Conhecimento ao se instalar a aplicação, o que implica que caso se queira inserir novos dados é necessário efectuar a desinstalação da aplicação e voltar a instalar a mesma já com a nova informação.
2. **Criação de Acções Pontuais:** neste momento apenas se podem efectuar registos de Acções que já pertençam aos Planos de Cuidados dos Doente. Pode surgir no entanto, durante a prestação de Cuidados, a necessidade do enfermeiro realizar uma Acção específica naquele momento, uma Acção Pontual. É importante incluir essa funcionalidade na aplicação para possibilitar o enfermeiro de efectuar os registos respectivos à mesma directamente no dispositivo.

3. **Aceder aos Dados do Doente:** ao se fazer o carregamento dos Planos de Cuidados é igualmente carregada informação relativa ao doente, não existe contudo forma de se visualizar. É importante dotar a aplicação de um método de se aceder a essa informação para assim o enfermeiro conseguir visualizar mais informação relevante do doente em questão, para além do próprio Plano de Cuidados.
4. **Carregamento do Plano de Cuidados por Enfermeiro:** existem três formas de se obterem os Planos de Cuidados (NHC, Quarto e Unidade de Enfermaria), no entanto não existe a hipótese de se obter os doentes atribuídos a determinada enfermeiro. Com mais esta possibilidade o enfermeiro poderia obter de forma mais rápida e eficaz os Planos de Cuidados dos doentes que realmente lhe são destinados.
5. **Portabilidade para uma Aplicação Web:** a actual aplicação está restrita ao S.O. Android. Apesar de ser o S.O. com maior número de utilizadores actualmente é importante que todos os tipos de S.O. possam utilizar esta aplicação. A forma mais eficaz para tal é através da migração desta aplicação nativa para uma aplicação Web. Esse trabalho já se encontra a ser efectuado no âmbito de uma outra tese de mestrado da Universidade de Évora como seguimento ao trabalho desenvolvido neste projecto.

Bibliografia

- [alert, 2006] alert (2006). alert saúde. mni - médicos na internet.
- [ALERT®, 2011a] ALERT®(2011a). Alert®- a new paradigm in healthcare it. Disponível em <http://www.alert-online.com/pt/new-paradigm#node-13866> Acedido em Março de 2011.
- [ALERT®, 2011b] ALERT®(2011b). Alert®inpatient. Disponível em <http://www.alert-online.com/pt/pfh#toc-target-6>. Acedido em Março de 2011.
- [Andrade, 2008] Andrade, A. I. (2008). História da enfermagem. Disponível em http://www.notapositiva.com/trab_estudantes/trab_estudantes/areaprojecto/historiadaenfermagem.htm. Acedido em Março de 2012.
- [Android, 2012] Android, O. S. P. (2012). Android developers. Disponível em <http://developer.android.com/index.html>. Acedido em Abril de 2012.
- [Beck et al., 2001] Beck, K., Beedle, M., Bennekum, A. v., Cockbrun, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto para o desenvolvimento Ágil de software. Disponível em <http://agilemanifesto.org/iso/ptpt/>. Acedido em Março de 2012.
- [Becker et al., 2001] Becker, A. K., Claro, D. B., and Sobral, J. B. (2001). Web services e xml um novo paradigma da computação distribuída. Universidade Federal de Santa Catarina.
- [canalys, 2011] canalys (2011). Smart phones overtake client pcs in 2011.
- [Carvalho and Mello, 2009] Carvalho, B. V. d. and Mello, C. H. P. (2009). Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil scrum. In *Anais do XII Simpósio de Administração da Produção, Logística e Operações Internacionais*. SIMPOI, São Paulo/SP, 2009, Franco.

- [Cavalcanti et al., 2009] Cavalcanti, E., Maciel, T. M. d. M., and Albuquerque, J. (2009). Ferramenta open-source para apoio ao uso do scrum por equipes distribuídas. In *III Workshop de Desenvolvimento Distribuído de Software*.
- [CIE/ISO, 2012] CIE/ISO (2012). Cen/iso en13606 standard.
- [Ciências de Enfermagem, 2010] Ciências de Enfermagem (2010). Bases teóricas e históricas do cuidar. Disponível em <http://cienciasdeenfermagem.wordpress.com/bases-teoricas-e-historicas-do-cuidar/>. Acedido em Março de 2012.
- [Costa, 2009] Costa, R. E. (2009). Registos de enfermagem. Disponível em http://pt.wikipedia.org/wiki/Medicina_baseada_em_evid\u00eancia
- [Cubas et al., 2010] Cubas, M. R., Silva, S. H. d., and Mariângela, R. (2010). Classificação internacional para a prática de enfermagem (cipe®): uma revisão de literatura. *Revista Eletrônica de Enfermagem*, 12(1):186–194.
- [de Enfermagem do Rio de Janeiro, 2012] de Enfermagem do Rio de Janeiro, C. R. (2012). História da enfermagem. Disponível em <http://www.medicinaintensiva.com.br/enfermagem-historia.htm>. Acedido em Março de 2012.
- [Diamedic©, 2011] Diamedic©(2011). Disponível em <http://web.me.com/nimartin/Site/Overview.html>. Acedido em Fevereiro de 2011.
- [Dorokhova et al., 2009] Dorokhova, R., Amelichev, N., and Krinkin, K. (2009). Evaluation of modern mobile platforms from the developer standpoint. *Screen*, 3:5.
- [Egenes, 2009] Egenes, K. J. (2009). History of nursing. In Roux, G. and Halstead, J. A., editors, *Issues and trends in nursing : essential knowledge for today and tomorrow*. Jones and Bartlett Publishers.
- [Epocrates®, 2011] Epocrates®(2011). Epocrates products for mobile devices. Disponível em http://www.epocrates.com/products/comparison_table.html. Acedido em Fevereiro de 2011.
- [Fadel and Silveira, 2010] Fadel, A. C. and Silveira, H. d. M. (2010). Metodologias ágeis no contexto de desenvolvimento de software: Xp, scrum e lean.
- [Freire, 2002] Freire, H. (2002). Web services: A nova arquitectura da internet. *Developer's CIO Magazine*, (73). Disponível em <http://www.cnnt.com.br/files/webservices.pdf>, Acedido em Março de 2012.
- [García, 2008] García, I. R. (2008). *HP Sanidade - Guía de Formación Funcional*. Hewlett-Packard.

- [Garcia and Nóbrega, 2009] Garcia, T. R. and Nóbrega, M. M. L. d. (2009). Classificação Internacional para a Prática de Enfermagem: inserção brasileira no projeto do Conselho Internacional de Enfermeiras. *Acta Paulista de Enfermagem*, 22:875 – 879.
- [glintt, 2011] glintt (2011). Glossary. Disponível em <http://www.glintt.com/solucoes/mobilidade/solucoes>. Acedido em Março de 2011.
- [Growth, 2012] Growth, S. S. I. . (2012). Smartbear software acquires eviware. Disponível em <http://stockholminnovation.com/SV/793/smartbear-software-acquires-eviware>. Acedido em Março de 2012.
- [Healey, 1991] Healey, R. G. (1991). Database management systems. In Maguire, D. J., Goodchild, M. F., and Rhind, D. W., editors, *Geographical Information Systems: principles and applications*, chapter 18, pages 251 – 267. Longman Scientific & Technical. Acedido em Fevereiro de 2012.
- [HP, 2011] HP (2011). Hp healthcare information system (hcis) - enabling integrated and coordinated care.
- [ICN, 2009] ICN (2009). *Linhas de Orientação para a Elaboração de Catálogos CIPE®*. Ordem dos Enfermeiros. Tradução: Dra. Hermínia Castro.
- [Institute, 2011] Institute, G. M. W. W. H. (2011). Glossary. Disponível em <http://www.westwirelesshealth.org/index.php/resources/glossary>. Acedido em Fevereiro de 2011.
- [IntellCare, 2011] IntellCare, I. (2011). Look4myhealth - home edition. Disponível em <http://www.isa.pt/intellicare/PDFs/Look4MyHealthHE-EN.pdf> Acedido em Fevereiro de 2011.
- [iTunes App Store, 2011] iTunes App Store (2011). Validus intouch md for iphone, ipad and ipod touch. Disponível em <http://itunes.apple.com/qa/app/validus-intouch-md/id393163105?mt=8>. Acedido em Fevereiro de 2011.
- [Jesus, 2011] Jesus, P. (2011). A mobilidade nos serviços de saúde. Disponível em <http://www.i-gov.org/images/articles/911/mGOV%20Movenssis.pdf>. Acedido em Fevereiro de 2011.
- [Lee and Chuvyrov, RESS] Lee, H. and Chuvyrov, E. (APRESS). *Beginning Windows Phone 7 Development*. APRESS.
- [Leitão et al., 2000] Leitão, G. C. M., Linard, A. G., and Rodrigues, D. P. (2000). Conceitos de enfermagem segundo roy, orem e watson. *Acta Paul Enf*, 13(3):76–80.
- [Limited, 2012] Limited, R. I. M. (2012). Blackberry 7 os. Disponível em <http://us.blackberry.com/apps-software/blackberry7/>. Acedido em Abril de 2012.
- [Lopes and Ramalho, 2004] Lopes, C. J. F. and Ramalho, J. C. (2004). Web services: Metodologias de desenvolvimento.

- [Lyavelino, 2007] Lyavelino (2007). Portal:enfermagem. Disponível em <http://pt.wikipedia.org/wiki/Portal:Enfermagem>. Acedido em Março de 2012.
- [Manso, 2010] Manso, J. (2010). Soapui: Testes em webservices. 13^a Reunião Presencial da Comunidade NetPonto em Lisboa, Disponível em <http://www.slideshare.net/NetPonto/soapui-testes-em-webservices-5063477>. Acedido em Março de 2012.
- [Mazoni et al., 2010] Mazoni, S. R., Rodrigues, C. C., Santos, D. S., Rossi, L. A., and Carvalho, E. C. d. (2010). Classificação Internacional para a Prática de Enfermagem e a contribuição brasileira. *Revista Brasileira de Enfermagem*, 63:285 – 289.
- [McHugh, 2006] McHugh, T. (2006). *Establishing Medical Men at the Paris Hôtel-Dieu, 1500-1715*, pages 209–224. Social History of Medicine.
- [MCS, 2010] MCS (2010). Mcs' ipatientcare ehr helps physiciansin achieving level 3 recognition on ncqa ppc-pcmh project. Disponível em <http://medcomsys.com/press-release.aspx#30thsep2010>. Acedido em Fevereiro de 2011.
- [MCS, 2011] MCS (2011). Mobile point-of-care solutions. Disponível em <http://www.medcomsys.com/MobilePOC.aspx>. Acedido em Fevereiro de 2011.
- [Microsoft, 2012] Microsoft (2012). Windows phone update history. Disponível em <http://www.microsoft.com/windowsphone/en-us/howto/wp7/basics/update-history.aspx>. Acedido em Abril de 2012.
- [mobileStorm, 2011] mobileStorm (2011). 2011 mhealth report - mobile healthcare technology: Trends and innovations. Technical report, mobileStorm Inc.
- [Nóbrega and Garcia, 2005] Nóbrega, M. M. L. d. and Garcia, T. R. (2005). Perspectivas de incorporação da Classificação Internacional para a Prática de Enfermagem (CIPE®) no Brasil. *Revista Brasileira de Enfermagem*, 58:227 – 230.
- [Nóbrega and Garcia, 2009] Nóbrega, M. M. L. d. and Garcia, T. R. (2009). Classificação Internacional para a Prática de Enfermagem: instrumental tecnológico para a prática profissional. *Revista Brasileira de Enfermagem*, 62:758 – 761.
- [Neto, 2005] Neto, A. C. (2005). Web services em java com axis - teoria e prática. Disponível em <http://guj.com.br/content/articles/webservices/ws.pdf>. Acedido em Março de 2012.
- [of Health & Human Services, 2011] of Health & Human Services, U. D. (2011). Health information privacy. Disponível em <http://http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html>. Acedido em Fevereiro de 2011.
- [Oliveira and Lima, 2011] Oliveira, E. and Lima, R. (2011). Estado da arte sobre o uso do scrum em ambientes de desenvolvimento distribuído de software. *Revista de Sistemas e Computação*, 1(2).

- [On-Line, 2012] On-Line, E. (2012). Enfermagem - evolução histórica. Disponível em <http://enfermeiroonline.no.comunidades.net/index.php?pagina=1303687778>. Acedido em Março de 2012.
- [Owens, 2006] Owens, M. (2006). *The Definitive Guide to SQLite (Definitive Guide)*. Apress, Berkely, CA, USA.
- [Padilha and Mancia, 2005] Padilha, M. I. C. d. S. and Mancia, J. R. (2005). Florence Nightingale e as irmãs de caridade: revisitando a história. *Revista Brasileira de Enfermagem*, 58:723 – 726.
- [Pianucci, 2002] Pianucci, A. (2002). *Saber cuidar: procedimentos básicos em enfermagem*. SENAC.
- [Ribeiro, 2011] Ribeiro, F. c. d. N. (2011). Uma abordagem comparativa do desenvolvimento de aplicações para dispositivos móveis. Master's thesis, Universidade Federal de Pernambuco.
- [samsung, 2011] samsung (2011). What is bada. Disponível em <http://www.bada.com/whatisbada/index.html>. Acedido em Abril de 2012.
- [Schepke and Viana, 2007] Schepke, C. and Viana, V. (2007). Avaliação de desempenho de soap sobre http,smtp e beep. Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- [Schwaber, 2009] Schwaber, K. (2009). *Guia do Scrum*. Scrum Alliance.
- [Sebek and Mladek, 2010] Sebek, M. and Mladek, T. (2010). E-health: From hospital to the nation-wide system. *SZKO*.
- [Silva, 2002] Silva, A. R. (2002). Web services: Próximos passos na globalização da computação distribuída. Instituto Superior Técnico / INESC-ID Lisboa.
- [Silva et al., 2008] Silva, R. R. d., Malucelli, A., and Cubas, M. R. (2008). Classificações de enfermagem: mapeamento entre termos do foco da prática. *Revista Brasileira de Enfermagem*, 61:835 – 840.
- [soapUI.org, 2012] soapUI.org (2012). Release history. Disponível em <http://www.soapui.org/About-SoapUI/release-history.html>. Acedido em Março de 2012.
- [Soares, 2004] Soares, M. d. S. (2004). Metodologias Ágeis extreme programming e scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, 3(1).
- [SQLite, 2012] SQLite (2012). Sqlite. Disponível em <http://www.sqlite.org/index.html>. Acedido em Fevereiro de 2012.
- [System, 2008] System, H. H. C. I. (2008). Sistema integrado de información sanitaria.

- [Validus Medical Systems, 2009] Validus Medical Systems, I. (2009). ValidusTMintouchTM. Data Sheet.
- [W3C, 2012] W3C (2012). Disponível em <http://www.w3.org/>. Acedido em Março de 2012.
- [Wikipédia, 2011a] Wikipédia (2011a). Computerized physician order entry. Disponível em http://en.wikipedia.org/wiki/Computerized_physician_order_entry. Acedido em Fevereiro de 2011.
- [Wikipédia, 2011b] Wikipédia (2011b). Medicina baseada em evidências. Disponível em http://pt.wikipedia.org/wiki/Medicina_baseada_em_evid\u00eancias. Acedido em Março de 2012.
- [Wikipédia, 2011c] Wikipédia (2011c). Rfid. Disponível em <http://pt.wikipedia.org/wiki/RFID>. Acedido em Março de 2011.
- [Wikipédia, 2012a] Wikipédia (2012a). Groovy (programming language). Disponível em http://en.wikipedia.org/wiki/Groovy_programming_language. Acedido em Março de 2012.
- [Wikipédia, 2012b] Wikipédia (2012b). Hp-ux. Disponível em <http://en.wikipedia.org/wiki/HP-UX>. Acedido em Fevereiro de 2012.
- [Wikipédia, 2012c] Wikipédia (2012c). soapui. Disponível em <http://en.wikipedia.org/wiki/http://en.wikipedia.org/wiki/SoapUI>. Acedido em Março de 2012.
- [Wikipédia, 2012d] Wikipédia (2012d). Windows phone 7. Disponível em http://pt.wikipedia.org/wiki/Windows_Phone_7#cite_note-3. Acedido em Abril de 2012.
- [Wikipedia, 2012a] Wikipedia (2012a). Android (operating system). Disponível em http://en.wikipedia.org/wiki/Android_%28operating_system%29#Version_history. Acedido em Abril de 2012.
- [Wikipedia, 2012b] Wikipedia (2012b). Blackberry os. Disponível em http://en.wikipedia.org/wiki/BlackBerry_OS/. Acedido em Abril de 2012.
- [Wikipedia, 2012c] Wikipedia (2012c). ios. Disponível em <http://en.wikipedia.org/wiki/IOS/>. Acedido em Abril de 2012.
- [Wikipedia, 2012d] Wikipedia (2012d). Symbian. Disponível em <http://en.wikipedia.org/wiki/Symbian/>. Acedido em Abril de 2012.
- [Zanuz et al., 2007] Zanuz, L., Filippetto, A. S., and Crespo, S. (2007). Service-oriented architecture: Teoria e prática.

Anexos

Anexo A

Interface Web Service

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<wsdl:definitions xmlns:hcis-nursery-pt="http://www.hphis.com/
  HcisNurseryPT/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/
  soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="
  HcisNurseryPT" targetNamespace="http://www.hphis.com/
  HcisNurseryPT/">
<wsdl:types>
  <xsd:schema targetNamespace="http://www.hphis.com/
    HcisNurseryPT/">
    <xsd:element name="Login">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="authentication" type="hcis-nursery-
            pt:AuthenticationType" </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="LoginResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="status" type="hcis-nursery-
            pt:LoginResponseStatusType" </xsd:element>
          <xsd:element name="statusReason" type="xsd:string" </
            xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
</wsdl:definitions>
```

```

    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getPatientCarePlanByNHC">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="authentication" type="hcis-nursery-
        pt:AuthenticationType" />
      <xsd:element name="nhc" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getPatientCarePlanByNHCResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="patientCarePlanList" type="hcis-nursery
        -pt:PatientCarePlanListType" />
      <xsd:element name="status" type="hcis-nursery-
        pt:GetPatientCarePlanResponseStatusType" />
      <xsd:element name="statusReason" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="PatientCarePlanListType">
  <xsd:sequence>
    <xsd:element name="PatientCarePlan"
      type="hcis-nursery-pt:PatientCarePlanType" maxOccurs="
        unbounded" minOccurs="0">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PatientCarePlanType">
  <xsd:sequence>
    <xsd:element name="PatientCarePlanInfo"
      type="hcis-nursery-pt:PatientCarePlanInfoType" maxOccurs="
        1" minOccurs="1">
    </xsd:element>
    <xsd:element name="NurseryDiagnosticList"
      type="hcis-nursery-pt:NuseryDiagnosticListType" maxOccurs
        ="1"
      minOccurs="0">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="InterdependentActivityList"
  type="hcis-nursery-pt:NurseryActionListType"
  maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="DiagnosticActivityList"
  type="hcis-nursery-pt:NurseryActionListType" maxOccurs="1"
  minOccurs="0">
</xsd:element>
<xsd:element name="NonDiagnosticActivityList"
  type="hcis-nursery-pt:NurseryActionListType"
  maxOccurs="1" minOccurs="0">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NuseryDiagnosticListType">
  <xsd:sequence>
    <xsd:element name="NurseryDiagnostic"
      type="hcis-nursery-pt:NurseryDiagnosticType" maxOccurs="
        unbounded" minOccurs="1">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PatientCarePlanInfoType">
  <xsd:sequence>
    <xsd:element name="Code" type="xsd:long"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="Description" type="xsd:string"
      maxOccurs="1" minOccurs="1">
    <xsd:annotation>
      <xsd:documentation></xsd:documentation>
    </xsd:annotation>
    </xsd:element>
    <xsd:element name="Patient"
      type="hcis-nursery-pt:PatientType" maxOccurs="1"
      minOccurs="1">
    </xsd:element>
    <xsd:element name="NurseryUnit" type="hcis-nursery-
      pt:NurseryUnitType"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="CreationUser"

```

```

        type="hcis-nursery-pt:UserType" maxOccurs="1"
        minOccurs="1">
    </xsd:element>
    <xsd:element name="InitialDate" type="xsd:dateTime"
        maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="ModificationUser"
        type="hcis-nursery-pt:UserType" maxOccurs="1" minOccurs="
        0">
    </xsd:element>
    <xsd:element name="ModificationDate"
        type="xsd:dateTime" maxOccurs="1" minOccurs="0">
    </xsd:element>
    <xsd:element name="FinalDate" type="xsd:dateTime" maxOccurs
        ="1" minOccurs="0">
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryDiagnosticType">
    <xsd:sequence>
        <xsd:element name="DiagnosticCode" type="xsd:long"
            maxOccurs="1" minOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    Knowledge base diagnostic code
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="CreationDate" type="xsd:dateTime"
            maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="CreationUser" type="xsd:string"
            maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="CreationNurseryUnit"
            type="xsd:string" maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="NurseryInterventionList"
            type="hcis-nursery-pt:NurseryInterventionListType"
            maxOccurs="1" minOccurs="1">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="NurseryInterventionListType">
  <xsd:sequence>
    <xsd:element name="NuseryInterventionList"
      type="hcis-nursery-pt:NurseryInterventionType"
      maxOccurs="unbounded" minOccurs="1">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryInterventionType">
  <xsd:sequence>
    <xsd:element name="InterventionCode" type="xsd:long"
      maxOccurs="1" minOccurs="1">
    <xsd:annotation>
      <xsd:documentation>
        Knowledge base intervention code
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
    <xsd:element name="CreationDate" type="xsd:dateTime"
      maxOccurs="1" minOccurs="1">
  </xsd:element>
    <xsd:element name="CreationUser" type="xsd:string"
      maxOccurs="1" minOccurs="1">
  </xsd:element>
    <xsd:element name="CreationNurseryUnit"
      type="xsd:string" maxOccurs="1" minOccurs="1">
  </xsd:element>
    <xsd:element name="NurseryActionList"
      type="hcis-nursery-pt:NurseryActionListType" maxOccurs="1"
      minOccurs="1">
  </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryActionListType">
  <xsd:sequence>
    <xsd:element name="NurseryAction"
      type="hcis-nursery-pt:NurseryActionType"
      maxOccurs="unbounded" minOccurs="1">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryActionType">
  <xsd:sequence>

```

```

<xsd:element name="ActionCode" type="xsd:long"
  maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="ActionDefinitionCode"
  type="xsd:long" maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="CreationDate" type="xsd:dateTime"
  maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="CreationUser" type="xsd:string"
  maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="CreationNurseryUnit"
  type="xsd:string" maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="ActionType"
  type="hcis-nursery-pt:NurseryActionTypeType"
  maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="Specification" type="xsd:string"
  maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="InitialDay" type="xsd:date"
  maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="InitialTime" type="xsd:time"
  maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="FrequencyCode" type="xsd:string"
  maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="FrequencyDescription"
  type="xsd:string" maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="FinalDay" type="xsd:date"
  maxOccurs="1" minOccurs="0">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="NurseryActionTypeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Scheduled"></xsd:enumeration>
    <xsd:enumeration value="Continuous"></xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>

```



```

    <xsd:enumeration value="Conditional"></xsd:enumeration>
    <xsd:enumeration value="OnDemand"></xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="getPatientCarePlanByNurseryUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="authentication"
        type="hcis-nursery-pt:AuthenticationType">
      </xsd:element>
      <xsd:element name="nurseryUnit" type="xsd:string"></
        xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getPatientCarePlanByNurseryUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="patientCarePlanList"
        type="hcis-nursery-pt:PatientCarePlanListType">
      </xsd:element>
      <xsd:element name="status" type="hcis-nursery-
        pt:GetPatientCarePlanResponseStatusType">
      </xsd:element>
      <xsd:element name="statusReason"
        type="xsd:string">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getPatientCarePlanByNurseryRoom">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="authentication"
        type="hcis-nursery-pt:AuthenticationType">
      </xsd:element>
      <xsd:element name="room" type="xsd:string"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getPatientCarePlanByNurseryRoomResponse">
  <xsd:complexType>
    <xsd:sequence>

```

```

<xsd:element name="patientCarePlanList"
  type="hcis-nursery-pt:PatientCarePlanListType">
</xsd:element>
<xsd:element name="status" type="hcis-nursery-
  pt:GetPatientCarePlanResponseStatusType">
</xsd:element>
<xsd:element name="statusReason"
  type="xsd:string">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="PatientType">
<xsd:sequence>
  <xsd:element name="Code" type="xsd:long" maxOccurs="1"
    minOccurs="1"></xsd:element>
  <xsd:element name="Name" type="xsd:string" maxOccurs="1"
    minOccurs="1"></xsd:element>
  <xsd:element name="ClinicalHistoryNumber"
    type="xsd:string" maxOccurs="1" minOccurs="1">
</xsd:element>
  <xsd:element name="DefinitiveClinicalHistoryNumber"
    type="xsd:boolean" maxOccurs="1" minOccurs="1">
</xsd:element>
  <xsd:element name="Episode" type="xsd:long" maxOccurs="1"
    minOccurs="1"></xsd:element>
  <xsd:element name="Gender"
    type="hcis-nursery-pt:GenderType" maxOccurs="1" minOccurs
    ="1">
</xsd:element>
  <xsd:element name="BirthDate" type="xsd:date" maxOccurs="1"
    minOccurs="1"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryUnitType">
<xsd:sequence>
  <xsd:element name="Code" type="xsd:long" maxOccurs="1"
    minOccurs="1"></xsd:element>
  <xsd:element name="Description" type="xsd:string" maxOccurs
    ="1" minOccurs="1"></xsd:element>
  <xsd:element name="Speciality" type="xsd:string" maxOccurs=
    "1" minOccurs="1"></xsd:element>
  <xsd:element name="MasterCode" type="xsd:string" maxOccurs=

```

```

    " 1" minOccurs="1"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name=" UserType">
  <xsd:sequence>
    <xsd:element name=" Code" type=" xsd:long"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name=" Name" type=" xsd:string"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name=" Login" type=" xsd:string"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name=" Gender" type=" xsd:string"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name=" Speciality" type=" xsd:string"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name=" Category" type=" xsd:string" maxOccurs="1"
      minOccurs="0"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name=" GenderType">
  <xsd:restriction base=" xsd:string">
    <xsd:enumeration value=" Female"></xsd:enumeration>
    <xsd:enumeration value=" Male"></xsd:enumeration>
    <xsd:enumeration value=" Hybrid"></xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name=" uploadPatientCarePlan">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name=" authentication"
        type=" hcis-nursery-pt:AuthenticationType">
      </xsd:element>
      <xsd:element name=" uploadPatientCarePlan" type=" hcis-
        nursery-pt:UploadPatientCarePlanType"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name=" uploadPatientCarePlanResponse">

```

```

<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="status" type="hcis-nursery-
      pt:UploadPatientCarePlanResponseStatusType" </
      xsd:element>
    <xsd:element name="statusReason"
      type="xsd:string">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="AuthenticationType">
  <xsd:sequence>
    <xsd:element name="Login" type="xsd:string" </xsd:element>
    <xsd:element name="Password" type="xsd:string" </
      xsd:element>
    <xsd:element name="Hospital" type="xsd:string" </
      xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UploadPatientCarePlanType">
  <xsd:sequence>
    <xsd:element name="Patient"
      type="hcis-nursery-pt:PatientType" maxOccurs="1"
      minOccurs="1">
    </xsd:element>
    <xsd:element name="NurseryActionConfirmations" type="hcis-
      nursery-pt:NurseryActionConfirmationListType" maxOccurs=
      "1" minOccurs="1" </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryActionConfirmationType">
  <xsd:sequence>
    <xsd:element name="AccionCode" type="xsd:long" maxOccurs="1"
      minOccurs="1" </xsd:element>
    <xsd:element name="Date" type="xsd:dateTime" maxOccurs="1"
      minOccurs="1">
    </xsd:element>
    <xsd:element name="Executed" type="xsd:boolean" maxOccurs="
      1" minOccurs="1" </xsd:element>
    <xsd:element name="Nullified_PT"
      type="xsd:boolean" maxOccurs="1" minOccurs="1">
    </xsd:element>
  </xsd:sequence>

```

```

<xsd:element name="Motive" type="xsd:string" maxOccurs="1"
  minOccurs="0"></xsd:element>
<xsd:element name="Note_PT" type="xsd:string" maxOccurs="1"
  minOccurs="0"></xsd:element>
<xsd:element name="NullificationReason_PT"
  type="xsd:string" maxOccurs="1" minOccurs="0">
</xsd:element>
<xsd:element name="User"
  type="hcis-nursery-pt:UserType" maxOccurs="1" minOccurs="
  1">
</xsd:element>
<xsd:element name="NurseryUnit"
  type="hcis-nursery-pt:NurseryUnitType" maxOccurs="1"
  minOccurs="1">
</xsd:element>
<xsd:element name="ClinicalRegister" type="hcis-nursery-
  pt:ClinicalRegisterType" maxOccurs="1" minOccurs="0"></
  xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NurseryActionConfirmationListType">
  <xsd:sequence>
    <xsd:element name="NurseryActionConfirmation" type="hcis-
      nursery-pt:NurseryActionConfirmationType" maxOccurs="
      unbounded" minOccurs="0"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ClinicalRegisterType">
  <xsd:choice>
    <xsd:element name="Annotation"
      type="hcis-nursery-pt:ClinicalRegisterAnnotationType">
    </xsd:element>
    <xsd:element name="Transaction" type="hcis-nursery-
      pt:ClinicalRegisterTransactionType"></xsd:element>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ClinicalRegisterTransactionType">
  <xsd:sequence>
    <xsd:element name="Type" type="xsd:string"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="Date" type="xsd:dateTime"
      maxOccurs="1" minOccurs="1">

```

```

</xsd:element>
<xsd:element name="State"
  type="hcis-nursery-pt:ClinicalRegisterStateType"
  maxOccurs="1" minOccurs="1">
</xsd:element>
<xsd:element name="ClinicalObservations" type="hcis-nursery
-pt:ClinicalRegisterObservationListType" maxOccurs="1"
  minOccurs="1"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="ClinicalRegisterStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="B"></xsd:enumeration>
    <xsd:enumeration value="CE"></xsd:enumeration>
    <xsd:enumeration value="CL"></xsd:enumeration>
    <xsd:enumeration value="FE"></xsd:enumeration>
    <xsd:enumeration value="FL"></xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ClinicalRegisterAnnotationType">
  <xsd:sequence>
    <xsd:element name="Type" type="xsd:string"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="Date" type="xsd:dateTime"
      maxOccurs="1" minOccurs="1">
    </xsd:element>
    <xsd:element name="Text" type="xsd:string" maxOccurs="1"
      minOccurs="1"></xsd:element>
    <xsd:element name="State" type="hcis-nursery-
      pt:ClinicalRegisterStateType" maxOccurs="1" minOccurs="1"
      ></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ClinicalRegisterObservationListType">
  <xsd:sequence>
    <xsd:element name="ObservationList" type="hcis-nursery-
      pt:ClinicalRegisterObservationType" maxOccurs="unbounded"
      minOccurs="1"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ClinicalRegisterObservationType">
  <xsd:sequence>

```

```

<xsd:element name="Type" type="xsd:string" maxOccurs="1"
  minOccurs="1"></xsd:element>
<xsd:element name="Value" type="xsd:string" maxOccurs="1"
  minOccurs="1"></xsd:element>
<xsd:element name="Description" type="xsd:string" maxOccurs
  ="1" minOccurs="1"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="getHospitals" type="xsd:string">
</xsd:element>
<xsd:element name="getHospitalsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="hospitalList" type="hcis-nursery-
        pt:HospitalListType" maxOccurs="1" minOccurs="1"></
        xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="HospitalListType">
  <xsd:sequence>
    <xsd:element name="Hospital" type="xsd:string" maxOccurs="
      unbounded" minOccurs="1"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="LoginResponseStatusType">
  <xsd:annotation>
    <xsd:documentation>
      STATUS_OK - Login was validated
      STATUS_ERROR_INEXISTENT_USERNAME - Username doesn't exist
      STATUS_ERROR_INVALID_LOGIN - The password wasn't correct
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="STATUS_OK"></xsd:enumeration>
    <xsd:enumeration
      value="STATUS_ERROR_INEXISTENT_USERNAME">
    </xsd:enumeration>
    <xsd:enumeration value="STATUS_ERROR_INVALID_LOGIN"></
      xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="GetPatientCarePlanResponseStatusType">

```

```

<xsd:annotation>
  <xsd:documentation>
    STATUS_OK – all patient care plans were successfully
    downloaded

    STATUS_ERROR_CAREPLAN_DOWNLOAD_TOTALLY_UNSUCCESSFUL – no
    patient care plan could be download

    STATUS_ERROR_CAREPLAN_DOWNLOAD_PARTIALLY_UNSUCCESSFUL –
    only part of patient care plans could be downloaded
    successfully
  </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="STATUS_OK"></xsd:enumeration>
  <xsd:enumeration
    value="
      STATUS_ERROR_CAREPLAN_DOWNLOAD_TOTALLY_UNSUCCESSFUL">
  </xsd:enumeration>
  <xsd:enumeration value="
    STATUS_ERROR_CAREPLAN_DOWNLOAD_PARTIALLY_UNSUCCESSFUL">
  /xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="UploadPatientCarePlanResponseStatusType"
">
  <xsd:annotation>
    <xsd:documentation>
      STATUS_OK – all patient care plans were
      successfully uploaded

      STATUS_ERROR_CAREPLAN_UPLOAD_TOTALLY_UNSUCCESSFUL
      – no patient care plan could be uploaded

      STATUS_ERROR_CAREPLAN_UPLOAD_PARTIALLY_UNSUCCESSFUL
      – only part of patient care plans could be
      uploaded successfully
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="STATUS_OK"></xsd:enumeration>
    <xsd:enumeration
      value="STATUS_ERROR_CAREPLAN_UPLOAD_TOTALLY_UNSUCCESSFUL"

```



```

    >
  </xsd:enumeration>
  <xsd:enumeration value="
    STATUS_ERROR_CAREPLAN_UPLOAD_PARTIALLY_UNSUCCESSFUL"></
    xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="LoginRequest">
  <wsdl:part element="hcis-nursery-pt:Login" name="parameters"/>
</wsdl:message>
<wsdl:message name="LoginResponse">
  <wsdl:part element="hcis-nursery-pt:LoginResponse" name="
    parameters"/>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNHCRRequest">
  <wsdl:part element="hcis-nursery-pt:getPatientCarePlanByNHC"
    name="parameters"/>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNHCResponse">
  <wsdl:part element="hcis-nursery-
    pt:getPatientCarePlanByNHCResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNurseryUnitRequest">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getPatientCarePlanByNurseryUnit"></wsdl:part>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNurseryUnitResponse">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getPatientCarePlanByNurseryUnitResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNurseryRoomRequest">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getPatientCarePlanByNurseryRoom"></wsdl:part>
</wsdl:message>
<wsdl:message name="getPatientCarePlanByNurseryRoomResponse">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getPatientCarePlanByNurseryRoomResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="uploadPatientCarePlanRequest">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:uploadPatientCarePlan"></wsdl:part>

```

```

</wsdl:message>
<wsdl:message name="uploadPatientCarePlanResponse">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:uploadPatientCarePlanResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="getHospitalsRequest">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getHospitals"></wsdl:part>
</wsdl:message>
<wsdl:message name="getHospitalsResponse">
  <wsdl:part name="parameters" element="hcis-nursery-
    pt:getHospitalsResponse"></wsdl:part>
</wsdl:message>
<wsdl:portType name="HcisNurseryPT">
  <wsdl:operation name="Login">
    <wsdl:input message="hcis-nursery-pt:LoginRequest" />
    <wsdl:output message="hcis-nursery-pt:LoginResponse" />
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNHC">
    <wsdl:input message="hcis-nursery-
      pt:getPatientCarePlanByNHCRequest" />
    <wsdl:output message="hcis-nursery-
      pt:getPatientCarePlanByNHCResponse" />
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNurseryUnit">
    <wsdl:input message="hcis-nursery-
      pt:getPatientCarePlanByNurseryUnitRequest"></wsdl:input>
    <wsdl:output message="hcis-nursery-
      pt:getPatientCarePlanByNurseryUnitResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNurseryRoom">
    <wsdl:input message="hcis-nursery-
      pt:getPatientCarePlanByNurseryRoomRequest"></wsdl:input>
    <wsdl:output message="hcis-nursery-
      pt:getPatientCarePlanByNurseryRoomResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="uploadPatientCarePlan">
    <wsdl:input message="hcis-nursery-
      pt:uploadPatientCarePlanRequest"></wsdl:input>
    <wsdl:output message="hcis-nursery-
      pt:uploadPatientCarePlanResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getHospitals">

```

```

    <wsdl:input message="hcis-nursery-pt:getHospitalsRequest"></
      wsdl:input>
    <wsdl:output message="hcis-nursery-pt:getHospitalsResponse"></
      /wsdl:output>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="HcisNurseryPTSOAP" type="hcis-nursery-
  pt:HcisNurseryPT">
  <soap:binding style="document" transport="http://schemas.
    xmlsoap.org/soap/http"/>
  <wsdl:operation name="Login">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/Login"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNHC">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/getPatientCarePlanByNHC"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNurseryUnit">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/getPatientCarePlanByNurseryUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getPatientCarePlanByNurseryRoom">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/getPatientCarePlanByNurseryRoom"/>

```

```

    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="uploadPatientCarePlan">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/uploadPatientCarePlan" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getHospitals">
    <soap:operation soapAction="http://www.hphis.com/
      HcisNurseryPT/getHospitals" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="HcisNurseryPT">
  <wsdl:port binding="hcis-nursery-pt:HcisNurseryPTSOAP" name="
    HcisNurseryPTSOAP">
    <soap:address location="http://www.hphis.com/" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Anexo B

Schema da Base de Dados

O schema da Base de Dados, que inclui a base de conhecimento

```
CREATE TABLE hospitals (  
    _id INT PRIMARY KEY autoincrement,  
    hospital TEXT NOT NULL  
);
```

```
CREATE TABLE frequency (  
    code TEXT PRIMARY KEY,  
    period INT  
);
```

```
CREATE TABLE enf_intervention (  
    _id INT PRIMARY KEY,  
    description TEXT NOT NULL,  
    text_ TEXT,  
    standardCode TEXT NOT NULL,  
    active TEXT NOT NULL,  
    confirmable TEXT NOT NULL  
);
```

```
CREATE TABLE enf_diagnostic_type (  
    _id INT PRIMARY KEY,  
    description TEXT NOT NULL  
);
```

```
CREATE TABLE enf_diagnostic (  
    _id INT PRIMARY KEY,  
    description TEXT NOT NULL,  
    text_ TEXT,  
    standardCode TEXT NOT NULL,  
    active TEXT NOT NULL,  
    type INT NOT NULL,  
    FOREIGN KEY (type) REFERENCES enf_tipo_diagnostico(_id)  
);
```

```
CREATE TABLE enf_diagint (  
    diagnostic INT NOT NULL,  
    intervention INT NOT NULL,  
    principal TEXT NOT NULL,  
    PRIMARY KEY (  
        diagnostic ,  
        intervention  
    )  
);
```

```
CREATE TABLE enf_object_Action (  
    _id INT PRIMARY KEY,  
    object_type TEXT NOT NULL,  
    objectCode TEXT NOT NULL  
);
```

```
CREATE TABLE enf_action_type (  
    _id INT PRIMARY KEY,  
    description TEXT NOT NULL,  
    pauta TEXT NOT NULL  
);
```

```
CREATE TABLE enf_action (  
    _id INT PRIMARY KEY,  
    type INT NOT NULL,  
    icon TEXT,  
    reqmotivenoconf TEXT,  
    active TEXT,  
    pauta TEXT,  
    descpauta TEXT,  
    delegar TEXT,  
    OBJECT INT,
```

```

        specify TEXT,
        description TEXT,
        actdiag_pt TEXT,
        FOREIGN KEY (OBJECT) REFERENCES enf_object_action(_id),
        FOREIGN KEY (type) REFERENCES enf_action_type(_id)
    );

CREATE TABLE enf_accint (
    intervention INT NOT NULL,
    action INT NOT NULL,
    principal TEXT NOT NULL,
    FOREIGN KEY (action) REFERENCES enf_action(_id),
    FOREIGN KEY (intervention) REFERENCES enf_intervention(_id),
    PRIMARY KEY (
        action ,
        intervention
    )
);

CREATE TABLE gender (
    _id TEXT PRIMARY KEY NOT NULL,
    description TEXT NOT NULL,
    internalCode TEXT
);

CREATE TABLE doc_specialty (
    _id TEXT PRIMARY KEY NOT NULL,
    description TEXT NOT NULL,
    active TEXT NOT NULL,
    primary_ TEXT NOT NULL,
    specialized TEXT NOT NULL
);

CREATE TABLE doc_diccio (
    _id TEXT PRIMARY KEY NOT NULL,
    description TEXT NOT NULL,
    speciality TEXT NOT NULL,
    sentence TEXT,
    connectionType TEXT,
    desplegable TEXT NOT NULL,
    position TEXT NOT NULL,
    FOREIGN KEY (speciality) REFERENCES doc_speciality(_id)
);

```

```
);
```

```
CREATE TABLE doc_data_type (  
  _id INT PRIMARY KEY NOT NULL,  
  description TEXT NOT NULL,  
  mask TEXT,  
  internalCode TEXT  
);
```

```
CREATE TABLE doc_uds (  
  _id TEXT PRIMARY KEY NOT NULL,  
  ud_std TEXT,  
  factor INT,  
  FOREIGN KEY (ud_std) REFERENCES doc_uds(_id)  
);
```

```
CREATE TABLE doc_mobx (  
  _id INT PRIMARY KEY NOT NULL,  
  shortDescription TEXT,  
  largeDescription TEXT NOT NULL,  
  abbreviation TEXT,  
  internalCode INT,  
  uds TEXT,  
  min FLOAT,  
  max FLOAT,  
  dataType INT NOT NULL,  
  dictionary TEXT,  
  graphic TEXT NOT NULL,  
  color TEXT,  
  symbol TEXT,  
  context TEXT NOT NULL,  
  ent_sal TEXT,  
  position INT,  
  minAbsolute FLOAT,  
  maxAbsolute FLOAT,  
  requireMobx INT,  
  valueRepresents TEXT,  
  commentAllows TEXT,  
  defectValue TEXT,  
  norm_ext TEXT NOT NULL,  
  dict_ext TEXT NOT NULL,  
  forma_val_edit TEXT,  
  formula TEXT,
```



```

FOREIGN KEY (dictionary) REFERENCES doc_diccio(_id),
FOREIGN KEY (requireMobx) REFERENCES doc_mobx(_id),
FOREIGN KEY (datatype) REFERENCES doc_data_type(_id),
FOREIGN KEY (uds) REFERENCES doc_uds(_id)
);

CREATE TABLE doc_tx_type (
    _id INT PRIMARY KEY NOT NULL,
    description TEXT NOT NULL,
    latency INT NOT NULL
);

CREATE TABLE doc_tx_type_obx (
    _id INT NOT NULL,
    obx INT NOT NULL,
    position INT NOT NULL,
    PRIMARY KEY (_id ,obx),
    FOREIGN KEY (obx) REFERENCES doc_mobx(_id),
    FOREIGN KEY (_id) REFERENCES doc_tx_type(_id)
);

CREATE TABLE doc_annota_type (
    _id INT PRIMARY KEY NOT NULL,
    type TEXT NOT NULL,
    class TEXT NOT NULL,
    latency INT NOT NULL,
    confidential TEXT
);

CREATE TABLE doc_item (
    _id INT PRIMARY KEY autoincrement ,
    code TEXT NOT NULL,
    dictionary TEXT NOT NULL,
    value TEXT NOT NULL,
    UNIQUE (code ,dictionary),
    FOREIGN KEY (dictionary) REFERENCES doc_diccio(_id)
);

/*
*
*   end of the knowledge database section
*
*_____

```

```

*****
*-----*
*
*   beginning of the care plans' section
*
*/
CREATE TABLE patient (
  _id long PRIMARY KEY,
  NAME TEXT NOT NULL,
  chn TEXT UNIQUE NOT NULL,
  definitivechn TEXT NOT NULL,
  episode long NOT NULL,
  gender TEXT NOT NULL,
  birthDate DATE NOT NULL
);

CREATE TABLE nursery_unit (
  _id long PRIMARY KEY,
  description TEXT NOT NULL,
  speciality TEXT NOT NULL,
  mastercode TEXT NOT NULL
);

CREATE TABLE user (
  _id long PRIMARY KEY,
  NAME TEXT NOT NULL,
  LOGIN TEXT NOT NULL,
  gender TEXT NOT NULL,
  speciality TEXT NOT NULL,
  category TEXT
);

CREATE TABLE enf_planpacient (
  _id long PRIMARY KEY NOT NULL,
  description TEXT NOT NULL,
  patientCode long NOT NULL,
  nurseryUnitCode long NOT NULL,
  creationUserCode long NOT NULL,
  initialDate DATE NOT NULL,
  modUserCode long,
  modDate DATE,
  finalDate DATE,
  FOREIGN KEY (patientCode) REFERENCES patient(_id),

```

```

FOREIGN KEY (nurseryUnitCode) REFERENCES nursery_unit(_id),
FOREIGN KEY (creationUserCode) REFERENCES user(_id),
FOREIGN KEY (modUserCode) REFERENCES user(_id)
);

CREATE TABLE enf_plan_diaglist (
  _id INT PRIMARY KEY autoincrement NOT NULL,
  diagnosticCode INT NOT NULL,
  planCode INT NOT NULL,
  creationDate DATE NOT NULL,
  creationUser TEXT NOT NULL,
  creationNurseryUnit TEXT NOT NULL,
  UNIQUE (
    diagnosticCode ,
    planCode
  ),
  FOREIGN KEY (planCode) REFERENCES enf_planpatient(_id)
);

CREATE TABLE enf_plan_diag_int (
  _id INT PRIMARY KEY autoincrement ,
  interventionCode INT NOT NULL,
  diagListCode INT NOT NULL,
  creationDate DATE NOT NULL,
  creationUser INT NOT NULL,
  creationNurseryUnit INT NOT NULL,
  FOREIGN KEY (diagListCode) REFERENCES enf_plan_diaglist(_id)
);

/*
*
*   end of the care plans ' section
*
*-----
*
*****
*-----
*
*   beginning of the working plans ' section
*
*/

CREATE TABLE enf_plan_int_acc (
  _id INT PRIMARY KEY,

```

```

    accionCode INT NOT NULL,
    interventionCode INT NOT NULL,
    creationDate DATE NOT NULL,
    creationUser INT NOT NULL,
    creationNurseryUnit INT NOT NULL,
    accionType TEXT NOT NULL,
    specification TEXT,
    initialDay DATE,
    initialTime TIME,
    frequencyCode TEXT,
    frequencyDescription TEXT,
    finalDay DATE,
    FOREIGN KEY (interventionCode) REFERENCES enf_diag_int(_id)
);

CREATE TABLE enf_acc (
    _id INT PRIMARY KEY,
    accionCode INT NOT NULL,
    planCode INT NOT NULL,
    creationDate DATE NOT NULL,
    creationUser INT NOT NULL,
    creationNurseryUnit INT NOT NULL,
    accionType TEXT NOT NULL,
    specification TEXT,
    initialDay DATE,
    initialTime TIME,
    frequencyCode TEXT,
    frequencyDescription TEXT,
    finalDay DATE,
    type_act TEXT NOT NULL,
    FOREIGN KEY (planCode) REFERENCES enf_planpacient(_id)
);

CREATE TABLE enf_plan_acc_conf (
    _id INT PRIMARY KEY autoincrement ,
    accID INT NOT NULL,
    STATE TEXT NOT NULL,
    motiveNotAccom TEXT,
    note TEXT,
    date_exec data ,
    time_exec TIME,
    user INT,
    nursery_unit INT,

```

```

        OBJECT INT,
        FOREIGN KEY (accID) REFERENCES enf_plan_int_acc(_id)
    );

CREATE TABLE enf_acc_conf (
    _id INT PRIMARY KEY autoincrement,
    accID INT NOT NULL,
    STATE TEXT NOT NULL,
    motiveNotAccom TEXT,
    note TEXT,
    date_exec data,
    time_exec TIME,
    user INT,
    nursery_unit INT,
    OBJECT INT,
    FOREIGN KEY (accID) REFERENCES enf_acc(_id)
);

CREATE TABLE doc_annotations_diag (
    _id INT PRIMARY KEY autoincrement,
    accId INT NOT NULL,
    process_number INT NOT NULL,
    speciality TEXT NOT NULL,
    doctor TEXT NOT NULL,
    annotation_Date DATE NOT NULL,
    annotation_Time TEXT NOT NULL,
    annotation_type INT NOT NULL,
    episode INT,
    annotation TEXT NOT NULL,
    STATE TEXT NOT NULL,
    user_code INT,
    user_name TEXT,
    FOREIGN KEY (accId) REFERENCES enf_plan_int_acc(_id)
);

CREATE TABLE doc_tx_diag (
    _id INT PRIMARY KEY autoincrement,
    accId INT NOT NULL,
    tx_type INT NOT NULL,
    tx_Date DATE NOT NULL,
    tx_Time TEXT NOT NULL,
    user INT,
    autor_name TEXT,

```

```
    speciality TEXT,
    patient INT NOT NULL,
    episode INT,
    STATE TEXT NOT NULL,
    FOREIGN KEY (accId) REFERENCES enf_plan_int_acc(_id)
);

CREATE TABLE doc_obx_diag (
    _id INT PRIMARY KEY autoincrement,
    tx INT NOT NULL,
    mobx INT NOT NULL,
    patient INT NOT NULL,
    user INT,
    value TEXT,
    description TEXT,
    obx_Date DATE,
    obx_Time TEXT,
    FOREIGN KEY (tx) REFERENCES doc_tx_diag(_id)
);

CREATE TABLE doc_annotations_act (
    _id INT PRIMARY KEY autoincrement,
    accId INT NOT NULL,
    process_number INT NOT NULL,
    speciality TEXT NOT NULL,
    doctor TEXT NOT NULL,
    annotation_Date DATE NOT NULL,
    annotation_Time TEXT NOT NULL,
    annotation_type INT NOT NULL,
    episode INT,
    annotation TEXT NOT NULL,
    STATE TEXT NOT NULL,
    user_code INT,
    user_name TEXT,
    FOREIGN KEY (accId) REFERENCES enf_acc(_id)
);

CREATE TABLE doc_tx_act (
    _id INT PRIMARY KEY autoincrement,
    tx_type INT NOT NULL,
    accId INT NOT NULL,
    tx_Date DATE NOT NULL,
    tx_Time TEXT NOT NULL,
```

```
user INT,  
author_name TEXT,  
speciality TEXT,  
patient INT NOT NULL,  
episode INT,  
STATE TEXT NOT NULL,  
FOREIGN KEY (accId) REFERENCES enf_acc(_id)  
);
```

```
CREATE TABLE doc_obx_act (  
  _id INT PRIMARY KEY autoincrement,  
  tx INT NOT NULL,  
  mobx INT NOT NULL,  
  patient INT NOT NULL,  
  user INT,  
  value TEXT,  
  description TEXT,  
  obx_Date DATE,  
  obx_Time TEXT,  
  FOREIGN KEY (tx) REFERENCES doc_tx_act(_id)  
);
```

