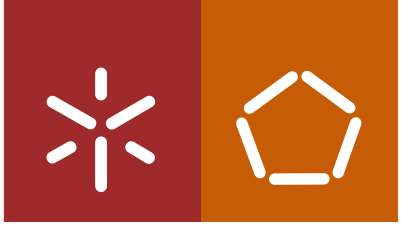
Universidade do Minho
Escola de Engenharia

Filipe Manuel Mota Miranda

Exploração e Desenvolvimento de Soluções Interoperáveis em Ambiente Hospitalar

novembro de 2016

Exploração e Desenvolvimento de Soluções Interoperáveis em Ambiente Hospitalar
Filipe Manuel Mota Miranda
UMinho | 2016



Universidade do Minho
Escola de Engenharia

Filipe Manuel Mota Miranda

Exploração e Desenvolvimento de Soluções Interoperáveis em Ambiente Hospitalar

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Informática Médica

Trabalho efetuado sob a orientação do
Professor Doutor José Manuel Ferreira Machado
e do
Doutor Júlio Duarte

novembro de 2016

DECLARAÇÃO

Nome: Filipe Manuel Mota Miranda
Endereço Eletrónico: filipemanuelmotamiranda@gmail.com
Cartão de Cidadão: 13905751

Título da Dissertação: Exploração e Desenvolvimento de Soluções Interoperáveis em Ambiente Hospitalar

Orientador: Professor Doutor José Manuel Ferreira Machado
Coorientador: Doutor Júlio Duarte

Ano de Conclusão: 2016

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica
Área de Especialização: Ramo de Informática Médica

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO.

Universidade do Minho, ____ / ____ / ____

Assinatura: _____

AGRADECIMENTOS

Em primeiro lugar gostaria de expressar a minha gratidão ao meu orientador, o Professor Doutor José Machado pela disponibilidade e ensinamentos transmitidos ao longo da presente dissertação.

Não poderia também deixar de agradecer ao Professor Doutor António Abelha, e ao Doutor Júlio Duarte pelos conselhos e disponibilidade sempre manifestados, que se revelaram vitais para a conclusão de todo o projeto. Não posso deixar de agradecer também aos Serviços de Informação do Centro Hospitalar do Porto pela atenção e disponibilidade prestada em todas as deslocações que realizei à mesma instituição de saúde.

Gostaria de agradecer aos meus pais e avós sempre presentes em todo o meu percurso através de uma palavra de incentivo e carinho determinante para o culminar deste percurso. Aos meus padrinhos e família um muito obrigado, mesmo à distância nunca deixaram de acompanhar todo o meu percurso.

Gostaria de agradecer e dedicar este projeto à minha companheira Cláudia, grande responsável pelo novo rumo da minha vida académica que culminou com a realização deste projeto. Ao meu filho Francisco um muito obrigado pelas brincadeiras e pelo seu sorriso que me deu força para terminar este projeto.

Gostaria por último de agradecer aos meus amigos, mais antigos e aqueles que fiz durante o meu percurso académico. Em especial, agradecer o apoio e incentivo daqueles que me acompanharam e trabalharam comigo de perto no laboratório.

RESUMO

Em contexto hospitalar, elaborar soluções interoperáveis remete para o conceito de *Sistemas de Informação (SI)* e a forma como estes são capazes de cooperar na partilha de informação. Aqueles que apresentam maior afinidade com a informática médica são naturalmente os *Sistemas de Informação em Saúde (SIS)*, que constituem um dos pontos mais importantes e mais sensíveis no que à prestação de cuidados de saúde diz respeito. A evolução desde o armazenamento de dados no formato de papel (*paper-based*) até à utilização de sistemas informáticos apresenta inúmeras vantagens. A maior rapidez no acesso a dados, a redução de erros (consulta, processamento e apresentação) e o diminuição de perdas de informação encabeçam sem dúvida uma lista bastante extensa de pontos positivos. Em contraponto, na altura em que muitos dos sistemas existentes foram implementados, o conceito da partilha de informação não era equacionado, resultando na criação de "ilhas de informação" sem qualquer comunicação com o exterior.

Com a evolução da Internet e da capacidade de processamento dos sistemas informáticos, a partilha de dados entre sistemas tornou-se uma constante, sendo vital a questão da interoperabilidade. Existem no entanto barreiras à implementação completa dessa ideologia motivado por divergências em formato de dados ou compatibilidade entre sistemas. Embora muito já tenha sido estudado e implementado, existem ainda no *Serviço Nacional de Saúde (SNS)* lacunas que só soluções propostas de raiz em prol da interoperabilidade semântica podem corrigir.

O setor da saúde nestes tempos de mudança deve aproveitar exemplos provenientes de outras áreas da economia, como por exemplo, a produção em larga escala onde uma nova revolução está em marcha, a Revolução Industrial 4.0. Através da introdução do conceito de indústria 4.0 desenvolveram-se *Cyber-physical System (CPS)*, onde dados relativos ao estado de uma máquina ou um processo são transmitidos pela rede com destino a unidades centrais de processamento. Estas, através de processos de *Extract, Load, Transform (ETL)*, transformam dados em informação útil acerca do estado da máquina, de uma linha de produção ou de toda ama unidade fabril, fazendo assim a ponte entre o mundo real e o virtual. Com esses dados recolhidos em tempo real, as ditas unidades centrais criam as bases para sistemas de monitorização, geralmente sob a forma de aplicações Web, permitem uma tomada de decisão mais informada por parte dos utilizadores envolvidos.

Os dispositivos móveis são hoje em dia um filão de negócio a ser explorado pelas unidades de saúde numa tentativa de aproximação ao utente e ao mesmo tempo baixando custos de comunicação, fomentando assim uma relação de confiança entre ambos. No desenvol-

vimento de uma aplicação móvel é percetível a forma de como a interoperabilidade é vital em sistemas distribuídos. A interoperabilidade física é garantida com utilização de Internet e pedidos *Hypertext Transfer Protocol (HTTP)*, a interoperabilidade sintática tira partido de estruturas como o *JavaScript Object Notation (JSON)* ou *Extensible Markup Language (XML)* para envio de dados na resposta aos diversos pedidos. Por último, a interoperabilidade semântica faz com que diferentes *layouts* sejam apresentados pela aplicação consoante os dados recebidos.

Ao longo do presente projeto, desenvolvido no âmbito da dissertação de mestrado em Engenharia Biomédica, ramo de Informática Médica foi primeiro realizado um levantamento de requisitos com base no estado atual dos SI no *Centro Hospitalar do Porto (CHP)* e mediante os resultados, foram desenvolvidos casos de estudo de forma a perceber qual o possível impacto das soluções encontradas. Quatro possíveis áreas de intervenção foram identificadas, a troca de informação seguindo a estrutura proposta pela *Health Level Seven Internation* na versão 2 relativa à troca de mensagens em contexto hospitalar, sistema de monitorização de agentes implementados na unidade de saúde (HL7 e AIDA), formulário de codificação *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* para posterior integração num sistema *Grupo de Diagnósticos Homogêneos (GDH)* e por fim a comunicação com o utente quando este está fora da unidade de saúde como forma de melhorar a imagem da unidade de saúde.

Como referido, utilizando as diretrizes *Health Level Seven International* para troca de mensagens na versão 2, foi desenvolvida uma interface do tipo *interface engine* na linguagem Python, que sustenta um *"Multi-Agent System"* (MAS) para envio e receção de mensagens. Aproveitando as diretrizes da indústria 4.0, é fundamental que estes agentes inteligentes comuniquem a cada instante o seu estado de forma a que os responsáveis conseguirem um *feedback* do estado atual de todo o sistema.

No segundo caso de estudo foi desenvolvida uma plataforma de monitorização com vista a rastrear o estado de agentes HL7 e AIDA já implementados no CHP e ainda daqueles desenvolvidos no caso de estudo anterior. Primeiro foi implementado um processo de ETL que alimenta um *Data Warehouse (DW)* onde a aplicação Web faz *queries* para recolher dados. Toda a plataforma foi desenvolvida, com base no conceito MVC com a *framework* Angularjs.

O terceiro caso de estudo apresenta uma plataforma de codificação ICD-9-CM numa abordagem à interoperabilidade semântica relacionada com representação de diagnósticos e procedimentos clínicos. Utilizando também o conceito MVC em Angularjs, a plataforma está inserida num sistema de GDH com objetivo de facilitar o trabalho do codificador, aumentar a sua produtividade, servir de barómetro ao trabalho desenvolvido e ainda categorizar os resumo de altas hospitalares. Com o GDH pode num futuro ser definido todo o financiamento hospitalar ou até compras realizadas por cada unidade.

No último caso de estudo, desenvolveu-se uma aplicação Android do enquadrada no conceito de *Electronic Health (eHealth)* do tipo *appointment remInder*. O objetivo é aproximar o paciente da unidade de saúde e evitando que este falhe o seu agendamento poupando em última instância dinheiro e recursos às unidades de saúde. Utilizou-se o design proposto pela Google para o envio de notificações o serviço *Firebase Cloud Message* também ele proposto pela Google. Mais que um lembrete a aplicação regista consultas passadas, futuras e permite fazer a confirmação da consulta. Este caso ainda se trata de um protótipo inicial para aceitação por parte da unidade de saúde.

Como métodologia de desenvolvimento em todos os casos de estudo, utiliza-se o ciclo*Design Research*. Primeiro é definido um requisito/problema que se pretende ver resolvido e todas as fases que se sucedem visam encontrar uma possível solução para tal. Como prova de conceito, escolheu-se a análise SWOT, situando-se esta, na fase de conclusão do ciclo DS. Todos os casos de estudo foram submetidos a essa análise apresentado resultados considerados positivos.

Fica assim vincada a importância da interoperabilidade no desenvolvimento de sistemas interoperáveis, sendo um conceito transversal a todas as soluções propostas, quer na fomentação direta da interoperabilidade ou no seu aproveitamento para a disseminação de informação. No primeiro caso, um sistema que favorece diretamente a partilha de informação utilizando *standards* de comunicação. No segundo, o controlo de agentes que trabalham diretamente na partilha de informação garantindo a sua segurança e rápida atuação em caso de erro. No terceiro caso, uma solução diretamente ligada à interoperabilidade semântica e por último uma aplicação que necessita de interoperabilidade para funcionar na sua plenitude.

ABSTRACT

In order to develop interoperable solutions, the Information Systems and information exchange among them are two of the basic concepts. Information Systems and for the purpose of this work Health Information Systems are without questions one of the key factors in the healthcare nowadays. The evolution from paper-based information storage to a computerized automated process presents numerous advantages like fastest information retrieval and less errors but there is a downside. The development of department or unit focused information system led to the creation and isolation of "islands of information". At the time the concept of information charing was not on the agenda.

The evolution of Internet and technology motivated the data charing across information systems and organizations. The interoperability is then a key factor under this new scenario. Although a large number of research and implementation has already occurred, there are still gaps. One the Portuguese scenario the semantic interoperability presents a tricky question that only specialized solution can solve.

The health care should look at examples from other fields, like manufacturing to seek for solutions. The Industry 4.0 implements "Cyber-Physical Systems" to get an live feed of the machine and production status. The data is provided by sensors or intelligent agent systems that extract the current status of the machine and transmit to a central unit.

The mobile industry presents nowadays a business opportunity for the healthcare sector through development of dedicated applications dedicated to approximate the patient and health care unit and reducing the costs in communication. Even at this point interoperability is vital. Physical interoperability through the establishment of communications using Internet and *Hypertext Transfer Protocol (HTTP)* requests. The syntactical interoperability with *JavaScript Object Notation (JSON)* format for information exchange, and at last the semantic interoperability for different displays with basis on different information.

The present work was developed under the scenario of the master thesis in Biomedical Engineering medical informatics branch. Intends then to present solutions in the format of study cases.

Were purposed 4 solutions. The first related with the information sharing standards through the usage of Health Level Seven message standards on the v2. format. The implementation an HL7 interface engine mixed with intelligence agents is a powerful tool to accomplish reliability and fast information exchange. The agents were implemented using Python 3.4 programming language.

The second case study, presents a monitoring system for HL7 and AIDA agents already working at *Centro Hospitalar do Porto (CHP)* and the new agents from the system presented at case study number one. The main function of this platform is present to the user the information processed in a *Extract, Load, Transform (ETL)* process implement in Python 3.4 programming language. Information like errors, agents beginning date and number of exchanged messages are present in a MVC model web application, developed using Angularjs framework.

The third case study was a specific need of the CHP. The Related Diseases Groups have been working fundamental four hospital ranking and money distribution for many years. This systems require the specific usage of *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* codification of medical release resume. So a MVC single page application was developed using Angularjs framework. The purpose is increase the number of codifications, evaluate the performance of the encoder and the health unit.

The last case study, presents a android mobile application introduced to save money to CHP in communication with patients, increase the health facility public image and prevents patients from not attend schedules. The application was developed using Google guidelines and to send notification the Firebase Cloud Message systems was adopted. The application meets the standards of *Electronic Health (eHealth)* mainly appointment remainders.

As development methodology, the Design Research cycle presents a good method to solve the requisites or problems identified previously. The last step settles if another DR is needed or if the right solution has been found. As proof-of-concept the SWOT analysis is a good and cheap solution for the purposed of the project. All the purposed solutions have a common point, the interoperability. The first case enables directly the syntactical interoperability. The second, supports and helps identifying errors in agents working directly with information exchange. The third case represents a direct approach to the semantic interoperability. The fourth, repesents a information system that depends directly of interoperability to work properly.

CONTEÚDO

Acrónimos xix

1 INTRODUÇÃO 3
- 1.1 Contexto 3
- 1.2 Motivação 5
- 1.3 Objetivos 7
  - 1.3.1 Perguntas Gerais Introdutórias 7
  - 1.3.2 Caso de Estudo - Sistema de Envio E Receção de Mensagens em Formato HL7 7
  - 1.3.3 Caso de Estudo - Sistema de Monitorização de Agentes AIDA e HL7 7
  - 1.3.4 Caso de Estudo - Formulário de Codificação ICD9 8
  - 1.3.5 Caso de Estudo - Aplicação Móvel CPH_Pacientes 8
  - 1.3.6 Perguntas Gerais Conclusivas 8
- 1.4 Estrutura do Documento 10

2 ESTADO DA ARTE 11
- 2.1 Introdução 11
- 2.2 Sistemas de Informação 12
  - 2.2.1 Sistemas e Partilha de Informação em Contexto Hospitalar 15
- 2.3 Interoperabilidade 20
  - 2.3.1 Enquadramento Teórico 20
  - 2.3.2 Implementação em Contexto Hospitalar 23
  - 2.3.3 Impacto da Interoperabilidade nas Unidades de Saúde 25
- 2.4 Standard de Comunicação HL7 27
  - 2.4.1 Estrutura Linguagem Health Level Seven V.2 29
- 2.5 Standard de Representação de Informação Clínica ICD9-CM 30
- 2.6 Indústria 4.0 33
  - 2.6.1 Enquadramento Teórico 33
  - 2.6.2 Adaptação ao Contexto Hospitalar 36
- 2.7 Pervasive or Ubiquitous Computing 38
  - 2.7.1 Enquadramento Teórico 38
- 2.8 Aplicações Móveis 40
  - 2.8.1 Aplicações Móveis Área da Saúde 42
- 2.9 Conclusão 44

3 METODOLOGIAS DE INVESTIGAÇÃO E FERRAMENTAS DE DESENVOLVIMENTO 45

3.1 Introdução 45
3.2 Design Research 45
3.3 Linguagem Programação Python v.3.4.3 47
3.4 Bases de Dados 49
3.4.1 Oracle D.B. 51
3.4.2 Mysql 52
3.5 Web Services 52
3.5.1 PHP 56
3.6 Frameworks e Linguagens de Desenvolvimento Web 57
3.6.1 Angulhar JS 57
3.7 Desenvolvimento de Aplicações Móveis 60
3.7.1 Android 60
3.7.2 Java 62
3.8 Conclusão 63
4 CASO DE ESTUDO - SISTEMA DE ENVIO E RECEÇÃO DE MENSAGENS EM FORMATO HL7 65
4.1 Introdução e Enquadramento no Tema 65
4.2 Materiais 71
4.3 Metodologia 72
4.3.1 Sistema Servidor 73
4.3.2 Sistema Cliente 75
4.3.3 Controlador Local 76
4.3.4 Controlador Geral 77
4.3.5 Comunicação entre Agentes 78
4.4 Resultados e Discussão 80
4.5 Conclusão 82
5 CASO DE ESTUDO - SISTEMA DE MONITORIZAÇÃO DE AGENTES AIDA E HL7 85
5.1 Introdução e Enquadramento no Tema 85
5.2 Materiais 92
5.3 Metedologia 93
5.3.1 Processo de ETL 95
5.3.2 Plataforma WEB 99
5.3.3 Servidor RESTFUL 100
5.4 Resultados e Discussão 101
5.5 Conclusão 104
6 CASO DE ESTUDO - FORMULÁRIO DE CODIFICAÇÃO ICD9-CM 105
6.1 Introdução e Enquadramento no Tema 105
6.2 Materiais 107

6.3 Metodologia 108
6.3.1 Modelo Relacional 110
6.3.2 Aplicação Web 111
6.4 Resultados e Discussão 113
6.5 Conclusão 117
7 CASO DE ESTUDO - APLICAÇÃO MÓVEL AIDA_PACIENTE 119
7.1 Introdução e Enquadramento no Tema 119
7.2 Materiais 124
7.3 Metodologia 124
7.4 Resultados e Discussão 130
7.5 Conclusão 132
8 PROVA DE CONCEITO 135
8.0.1 Análise SWOT 135
8.0.2 Estudo de Usabilidade 143
9 CONCLUSÕES 145
9.1 Questão 1: Qual é a evolução dos sistemas de informação nas unidades de saúde e qual é a importância da interoperabilidade para estes? 146
9.2 Questão 2: Como melhorar a interoperabilidade existente nas unidades de saúde, nomeadamente no CHP? 147
9.3 Questão 3: É viável estruturar um MAS de forma a utilizar padrões *Health Level Seven International (HL7)* possibilitando assim a interoperabilidade sintática e qual é a melhor linguagem para o fazer? 148
9.4 Questão 4: Qual é o papel de um sistema de monitorização da performance hospitalar? 150
9.5 Questão 5: Quais são as melhores tecnologias que viabilizam o desenvolvimento de um sistema como aquele abordado na questão 4? 151
9.6 Questão 6: Qual é o melhor tipo de *web service* para ser implementado na unidade de saúde servindo como base para a interoperabilidade entre as bases de Dados e a aplicação? 151
9.7 Questão 7: O que representa um sistema de GDH para as unidades de saúde? 152
9.8 Questão 8: Qual é o papel ICD-9-CM no sistema GDH e quais as tecnologias que permitem ao utilizador fazer a codificação? 153
9.9 Questão 9: Qual é a melhor forma de implementar aplicações *mHealth* e em que cenários favorecem o Centro Hospitalar do Porto? 153
9.10 Questão 10: Qual a melhor forma de apresentar informações em contexto de aplicações móveis e qual o impacto da interoperabilidade neste cenário? 154

9.11 Questão 11: É viável implementar uma aplicação móvel dedicada a pacientes com vista à disseminação de informação e qual o possível impacto desta? 155
9.12 Questão 12: Qual é o futuro dos *Sistemas de Informação (SI)* e o da interoperabilidade? 155
9.13 Questão 13: De que forma é que os casos de estudo apresentados contribuem para o melhoramento da interoperabilidade a nível hospitalar? 156
9.14 Trabalho Futuro 157

Bibliografia 159

Glossário 173

A APÊNDICE A 175
A.1 Publicações 175
A.1.1 An Approach for the Semantic Interoperability 175
A.1.2 An Online-Processing Critical Patient Monitoring System- An Interoperability Overview 176

B APÊNDICE B 177
B.1 Funcionalidades da Aplicação Web 177
B.1.1 Vista Inicial 178
B.1.2 Agentes HL7 181
B.1.3 Controlo de Agentes 182
B.1.4 Agentes AIDA 183
B.1.5 Indicadores 184
B.1.6 Mensagens 186
B.1.7 Extras 188
B.2 Aplicação De Registo No Centro Hospitalar do Porto 190

LISTA DE FIGURAS

Figura 1 Sistema de informação e seus diversos componentes. 13
Figura 2 Esquema de relações entre três blocos principais das TIC. 18
Figura 3 AIDA - esquema geral de ligações. 24
Figura 4 Posicionamento dos diferentes intervenientes dentro da indústria 4.0. 34
Figura 5 Arquitetura do sistema IEEE802.15.4. 43
Figura 6 Diversas etapas do processo *Design Research*. 47
Figura 7 Modelo conceptual da arquitetura SOA. 54
Figura 8 Representação típica modelo MVC 58
Figura 9 Ciclo de vida de uma aplicação android . 62
Figura 10 Exemplo do tipo de implementação *point-to-point*. 68
Figura 11 Exemplo do tipo de implementação *interface engine*. 68
Figura 12 Esquema geral dos sistema multi agente desenvolvido 73
Figura 13 Esquema geral das interações entre agentes do tipo HL7 80
Figura 14 Ambiente dos processos ETL. 89
Figura 15 Diagrama do Ciclo de Kimball. 90
Figura 16 Esquema geral do processo ETL modulo HL7. 97
Figura 17 Esquema geral do processo ETL modulo AIDA. 98
Figura 18 Esquema de interação com a aplicação desenvolvida para o Centro Hospitalar do Porto. 100
Figura 19 Aplicação web desenvolvida em modo codificação ou recodificação. 114
Figura 20 Aplicação web desenvolvida em modo consulta. 116
Figura 21 iniciativas mHealth. 121
Figura 22 Adoção de iniciativas mHealth nas mais diversas fase de desenvolvimento. 122
Figura 23 Esquema implementado para o envio de notificações. 128
Figura 24 Resultado final da aplicação Android. 131
Figura 25 Esquema geral da análise SWOT. 136
Figura 26 Resultado final da aplicação web - *Login*. 178
Figura 27 Resultado final da aplicação web - Vista Inicial HL7. 179
Figura 28 Resultado final da aplicação web - Vista Inicial HL7. 180
Figura 29 Resultado final da aplicação web - Vista Inicial em Modo Projeção. 181

Figura 30 Resultado final da aplicação web - *Zoom* nas funcionalidades de determinado agente. 182
Figura 31 Resultado final da aplicação web - Controlo de agentes pertencentes ao MAS. 183
Figura 32 Resultado final da aplicação web - Avisos sobre agentes AIDA. 184
Figura 33 Resultado final da aplicação web - Indicadores espaço temporal de um dia. 185
Figura 34 Resultado final da aplicação web - Indicadores espaço temporal 2 a 3 dias. 186
Figura 35 Resultado final da aplicação web - Mensagens com ACK positivo. 187
Figura 36 Resultado final da aplicação web - Mensagens com erro. 188
Figura 37 Resultado final da aplicação web - Bloco de Notas. 189
Figura 38 Resultado final da aplicação web - Registo de Clientes no CHP. 190

LISTA DE TABELAS

Tabela 1 Apresentação das perguntas elaborados no questionário ao utilizador 144

ACRÓNIMOS

A

AIDA Agéncia para a Integração, Difusão e Arquivo de Informação Médica. 10, 23–25, 44, 96, 97, 99, 100, 102–104, 139, 140, 147, 148, 150, 156, 176

AIDA-BI Agéncia para a Integração, Difusão e Arquivo de Informação Médica - Business Intelligence. 176

AIDA-PCE Agéncia para a Integração, Difusão e Arquivo de Informação Médica - Processo Clínico Eletrónico. 24, 176

B

BI Business Intelligence. 49, 66, 88, 90, 91, 93, 100, 101, 104, 114, 117, 138, 139, 143, 150, 157

C

CHP Centro Hospitalar do Porto. vi, x, xiii, 3, 6, 7, 9, 25, 83, 96, 108, 125, 145, 147, 152–154, 178, 179

CPS Cyber-physical System. v, 4, 6, 35, 36

D

DW Data Warehouse. vi, 90

E

EHEALTH Electronic Health. vii, x, 21, 44, 119, 124, 145, 153

EHR Electronic Health Record. 20, 24

ETL Extract, Load, Transform. v, vi, x, 50, 89–92, 95–98, 102, 104, 108, 111, 124, 126, 128, 131, 138–140, 157

F

FCM Firebase Cloud Message. 128, 129, 131

G

GDH Grupo de Diagnósticos Homogêneos. vi, 6, 8–10, 27, 31–33, 105–107, 111, 114–117, 140, 141, 148, 152, 153, 156

H

HIS Health Information Systems. 15, 20, 21, 23, 25, 28, 65–67, 83, 145, 148, 156

HL7 Health Level Seven International. xiii, 5, 7, 10, 22, 24–29, 44, 65–71, 73, 74, 76, 81–83, 86, 87, 96–98, 101–104, 137, 138, 140, 148–150, 156

HTML Hyper Text Markup Language. 23, 55–57, 59, 63, 85, 91, 93, 100, 113, 127, 151

HTTP Hypertext Transfer Protocol. vi, ix, 52–54, 57, 91, 94, 95, 109, 124, 127–130, 151, 152, 157

I

ICD-9 International Classification Diseases 9 Revision. 21, 44, 113

ICD-9-CM International Classification Diseases 9 Revision Clinical Modification. vi, x, 6, 8–10, 22, 31, 32, 105, 108, 110, 112, 117, 141, 148, 153, 157

IM Informática Médica. 4, 15

IP Internet Protocol. 69, 74, 75, 80, 102, 137

J

JSON JavaScript Object Notation. vi, ix, 55, 59, 78, 83, 94, 101, 110, 112, 115, 149, 152, 155, 157

L

LIS Laboratory Information Systems. 66

M

MHEALTH Mobile Health. 4, 145, 147, 148

O

OMS Organização Mundial de Saúde. 31, 120

OSI Organização Internacional para a Normalização. 27, 28, 69

P

PACS Picture Archiving and Communication System. 67

PCE Processo Clínico Eletrónico. 11, 19, 66, 105, 145, 155

PHP Hypertext Preprocessor. 45, 52, 56, 100, 127, 130, 151, 153

PLSQL Procedural Language-Structured Query Language. 50, 51

PNA Presente na Admissão. 110

PNS Plano Nacional de Saúde. 15

POC Proof-of-Concept. 135, 157

POP Proof-of-Principles. 135

PRM Patient Relationship Manager. 4, 120, 123, 125, 146

R

RCE Registo Clínico Eletrónico. 19

RIS Rede Informática da Saúde. 66

S

SAM Sistema de Apoio ao Médico. 19

SAPE Serviço de Apoio à Prática de Enfermagem. 19

SI Sistemas de Informação. v, vi, xiv, 4, 8, 12–14, 17, 20–23, 34, 46, 66, 145–147, 150, 155

SIH Sistemas de Informação Hospitalar. 4, 44, 155

SIS Sistemas de Informação em Saúde. v, 11, 15, 17, 18, 66

SNS Serviço Nacional de Saúde. v, 17, 20, 32, 117

SO Sistema Operativo. 39, 61, 62, 96, 102, 125, 146, 154

SONHO Sistema de Informação para a Gestão de Doentes. 11, 19, 132

SQL Structured Query Language. 49–52

T

TI Tecnologias de Informação. 15

TIC Tecnologias de Informação e Comunicação. 5, 15, 16, 18, 20, 22, 26, 27, 44, 65

X

XML Extensible Markup Language. vi, 22, 23, 53, 55, 56, 78, 124, 126, 151, 152, 155

1

INTRODUÇÃO

A presente dissertação descreve o trabalho realizado na análise das diferentes formas de disseminação de informação em contexto hospitalar, mais concretamente no *Centro Hospitalar do Porto (CHP)*, para posterior implementação de soluções que fomentem a interoperabilidade ou dela tirem proveito. O projeto está enquadrado na dissertação de mestrado em Informática Médica do Mestrado Integrado em Engenharia Biomédica da Universidade do Minho.

O presente capítulo tem como objetivo contextualizar o trabalho desenvolvido, assim como apresentar as principais motivações para o seu desenvolvimento. Para além disso, como forma de apresentar os diversos objetivos do projeto, desenvolveu-se uma lista de perguntas que servem de mote para a dissertação e no capítulo da conclusão, o que se pretende é que estas sejam respondidas de forma clara e objetiva. A derradeira secção do capítulo apresenta a estrutura do documento com o intuito de facilitar a leitura do mesmo.

1.1 CONTEXTO

A quantidade de informação recolhida em contexto hospitalar, apresentou na última década um crescimento abrupto em quantidade e qualidade. Contudo nem sempre a abundância significa qualidade, por vezes o produto final não consegue transmitir a informação correta no momento oportuno, por atrasos na recolha/processamento de dados ou simplesmente por erros de má comunicação, processamento ou interpretação [1].

Diariamente ao nível hospitalar, assiste-se a um enorme volume de dados a circular através dos canais internos de informação. A título de exemplo, só nos Estados Unidos da América, a quantidade de dados gerados por ano, chegou em 2011 aos 150 Exabyte [2]. Estes números demonstram a responsabilidade e a importância de qualquer sistema com a função de enviar, receber ou processar dados em contexto hospitalar. Existe assim um conjunto de propriedades/características base que lhes permitem desempenhar o seu papel de forma eficiente. A fiabilidade e disponibilidade encabeçam uma lista de atributos à qual se deve juntar a "disponibilidade" de dados, a especificidade da informação sem nunca deixar de parte a segurança.

Após anos de desenvolvimento, os *Sistemas de Informação Hospitalar (SIH)* estão atualmente focados nas necessidades daqueles que de forma direta ou indireta, estão ligados aos cuidados de saúde (sistemas de apoio à decisão, comunicação, entre outros). Para a perceção de lacunas e a apresentação de soluções inovadoras em muito tem contribuído a *Informática Médica (IM)*. A comunidade científica afeta a este ramo da informática tem desempenhado um papel preponderante na investigação e no desenvolvimento de soluções que proporcionem uma tomada de decisão mais sensata e esclarecida, contribuindo em parte para a redução do erro associado às diversas atividades de uma unidade hospitalar.

Os SIH formam então sistemas de elevada complexidade onde impera a heterogeneidade, uma vez que comunicam em diferentes formatos e artefactos. Existe portanto um "isolamento" da informação dentro de determinados pontos de uma unidade hospitalar, com origem nos primórdios dos SIH . Nessa altura o foco não era a partilha de informação como atesta o trabalho de D. E. Detmer intitulado *"Information Technology for Quality Health Care: a Summary of United Kingdom and United Stated Expenses"*, ao afirmar que na altura a maioria dos *Sistemas de Informação (SI)* eram desenvolvidos e implementados a pensar numa unidade ou máquina em específico [3].

É imperativo desenvolver soluções originais que permitam a integração e a interoperabilidade destes sistemas, ajudando a terminar com problemas do passado e acima de tudo permitam a partilha e transformação de dados em informação . A integração visa recolher informação proveniente de diferentes origens. Já o segundo ponto remete para a interoperabilidade, atestando dessa forma a importância de dois ou mais sistemas comunicarem entre SI, de forma a que aquele que recebe informação seja capaz de a perceber e utilizar no futuro. De acordo com Sandra V. Jardim em *"The Eletronic Health Record and Contribution to Healthcare Information Systems Interoperability"*, a interoperabilidade é obrigatória para que a comunicação entre sistemas funcione corretamente [4].

Existem outras áreas da economia que podem servir de exemplo para ideias originais, nomeadamente os princípios da indústria 4.0 onde as máquinas comunicam entre si e com operador criando um sistema *Cyber-physical System (CPS)* [5]. A unidade de produção funciona então como uma rede social com o objetivo de reduzir custos, aumentar produtividade e diminuir tempos de resposta em caso de erro. Os dados recolhidos podem ser utilizados para ajudar a uma decisão mais informada ou simplesmente para o controlo e deteção de erros. Adaptar esta ideologia a sistemas de agentes inteligentes é então um dos pontos do presente projeto, servindo posteriormente de base a um sistema de monitorização e controlo de agentes.

A partilha de informação, com a introdução de dispositivos móveis mudou drasticamente, juntando isso com evolução da Internet, determinada pessoa pode estar contactável 24 horas por dia. É então interessante estender a partilha de informação para fora das fronteiras físicas da unidade de saúde numa abordagem ao *Mobile Health (mHealth)* e *Patient*

Relationship Manager (PRM). Mais uma vez sem interoperabilidade sintática e semântica tal extensão não é de todo viável.

Este projeto incide na exploração e desenvolvimento de soluções interoperáveis diretamente ligadas a algumas áreas da partilha de informação, umas mais visíveis outras fazem parte do *background* das unidades de saúde que dependem ou trabalham em prol da interoperabilidade. Assim sendo, são abordados aspetos relacionados com a estrutura em que a informação é partilhada (interoperabilidade sintática) através das diretrizes propostas pela *Health Level Seven International (HL7)*, formas de promover a interoperabilidade semântica como é o caso de codificações clínicas utilizando ICD-9, controlo e monitorização de agentes e processos e por fim partilha de informação para o exterior da unidade de saúde utilizando dispositivos móveis.

1.2 MOTIVAÇÃO

Num mundo cada vez afeto às tecnologias e em particular às *Tecnologias de Informação e Comunicação (TIC)*, desenvolver soluções interoperáveis é facilitado em parte pelo desenvolvimento tecnológico e das ferramentas disponíveis para a produção de Software. Parte da motivação deste projeto está na implementação de novas abordagens a sistemas já implementados assim como aproveitar novas *frameworks* e metodologias na correção de lacunas existentes.

Embora o grande foco dos sistemas de informação e de apoio à decisão sejam os profissionais de saúde que contactam diretamente com o paciente é imperativo também a existência de ferramentas destinadas ao *background* das unidades de saúde (gestores de bases de dados, serviços de informação, gestores de sistemas, etc.). A Faculdade de Medicina da Universidade do Porto destaca no seu relatório sobre sistemas de informação em saúde, que determinado sistema pode estar orientado para os diversos tipos de utilizadores, entre os quais [6]:

- Pessoal administrativo;
- Profissionais de Saúde - médicos,enfermeiros entre outros;
- Gestores - administração clínica, diretores serviço;
- Utentes.

Da lista anterior resultam possíveis pontos onde a partilha de informação e interoperabilidade podem precisar de soluções inovadoras para corrigir lacunas ou erros provenientes do passado. Sabendo das possíveis áreas de intervenção é necessário saber que tipo de acções podem ser executadas, para tal Júlio Duarte e seus colaboradores atestam que os

padrões segundo qual os dados podem ser transmitidos, estão divididos em três blocos principais [7]:

- Comunicação;
- Representação de Informação Clínica;
- Imagem Médica.

Aproveitar abordagens de sucesso noutras áreas faz parte da motivação do trabalho, podendo a indústria 4.0 representar uma boa fonte de ideias para criação de soluções inovadoras. O CPS pode ser utilizado na criação de uma solução híbrida, adaptada ao primeiro ponto da lista anterior, ou seja, a partilha de informação de acordo com *standards* de comunicação propostos pela *Health Level Seven International*. Explorar o conceito de MAS é algo aliciante, com provas dadas em diversas áreas como por exemplo leilões *online*. Detetar o erro o mais cedo possível ou prevenir que este aconteça é motivação em qualquer área, muito mais quando estão a ser transacionados grandes volumes de dados. No CHP, existem agentes e sistemas com a função de ajudar na integração de informação clínica, sendo assim de vital importância monitorizar o seu comportamento a todo e qualquer instante.

Com a quantidade de *frameworks* e soluções para produção de conteúdo Web é cada vez mais simples desenvolver aplicativos atrativos e fluídos, contudo nem sempre se revelam intuitivos e com capacidade de ajudar na execução de tarefas. Neste contexto, a interoperabilidade semântica utiliza sistemas de codificação bastante complexos, daí que soluções que facilitem o trabalho daqueles que os utilizam seja vital. A utilização de códigos *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* para codificação de resumos de alta está na base dos *Grupo de Diagnósticos Homogêneos (GDH)*. Existe atualmente carência de aplicações web intuitivas e de fácil utilização dedicadas especificamente à codificação clinica, daí que constitua uma motivação para esta dissertação.

A comodidade e o bem estar de pacientes é uma preocupação constante dos serviços de saúde, com a crescente utilização de dispositivos móveis as unidades de saúde devem pensar em desenvolver aplicações vocacionadas para estes, por forma a manter um contacto mais estreito com os seus utentes.

Subjacente às motivações apresentadas está a criação de soluções que contribuam direta ou indiretamente para o aumento da qualidade de prestação dos cuidados de saúde. Por outro lado o desenvolvimento de soluções que fomentem a interoperabilidade vai afetar positivamente todo e qualquer sistema de informação, devido à estreita relação entre ambos.

1.3 OBJETIVOS

Como visto na secção anterior a presente dissertação tem diversas motivações dispares que conduzem a diferentes objetivos relacionados com a partilha de informação e subjacente interoperabilidade.

Os objetivos assumem um papel preponderante no desenrolar de todo o projeto, estes podem então ser vistos como questões que servem posteriormente de *guideline* ao processo de resolução e para os quais vão ser procuradas soluções. Existem questões de desenvolvimento transversais a todo o projeto e outras que servem de mote ao desenvolvimento dos casos de estudo.

1.3.1 *Perguntas Gerais Introdutórias*

QUESTÃO 1: Qual é a evolução dos sistemas de informação nas unidades de saúde e qual é a importância da interoperabilidade para estes?

QUESTÃO 2: Como melhorar a interoperabilidade existente nas unidades de saúde, nomeadamente no CHP?

1.3.2 *Caso de Estudo - Sistema de Envio E Receção de Mensagens em Formato HL7*

QUESTÃO 3: É viável estruturar um MAS de forma a utilizar padrões HL7 possibilitando assim a interoperabilidade sintática e qual é a melhor linguagem para o fazer?

1.3.3 *Caso de Estudo - Sistema de Monitorização de Agentes AIDA e HL7*

QUESTÃO 4: Qual é o papel de um sistema de monitorização da performance hospitalar?

QUESTÃO 5: Quais são as melhores tecnologias que viabilizam o desenvolvimento de um sistema como aquele abordado na questão 4?

QUESTÃO 6: Qual é o melhor tipo de *web service* para ser implementado na unidade de saúde servindo como base para a interoperabilidade entre as bases de dados e a aplicação?

1.3.4 *Caso de Estudo - Formulário de Codificação ICD9*

QUESTÃO 6: Qual é o melhor tipo de *web service* para ser implementado na unidade de saúde servindo como base para a interoperabilidade entre as bases de Dados e a aplicação?

QUESTÃO 7: O que representa um sistema de GDH para as unidades de saúde?

QUESTÃO 8: Qual é o papel ICD-9-CM no sistema GDH e quais as tecnologias que permitem ao utilizador fazer a codificação?

1.3.5 *Caso de Estudo - Aplicação Móvel CPH_Pacientes*

QUESTÃO 9: Qual é a melhor forma de implementar aplicações *mHealth* e em que cenários favorecem o Centro Hospitalar do Porto?

QUESTÃO 10: Qual a melhor forma de apresentar informações em contexto de aplicações móveis e qual o impacto da interoperabilidade neste cenário?

QUESTÃO 11: É viável implementar uma aplicação móvel dedicada a pacientes com vista à disseminação de informação e qual o possível impacto desta?

1.3.6 *Perguntas Gerais Conclusivas*

QUESTÃO 12: Qual é o futuro dos SI e o da interoperabilidade?

QUESTÃO 13: De que forma é que os casos de estudo apresentados contribuem para o melhoramento da interoperabilidade a nível hospitalar?

A perguntas introdutórias visam o enquadramento dentro de cada cenário abordado, posteriormente estas estão na génese daquelas que vão servir de *guideline* em pontos específicos do trabalho. Cada uma das perguntas tem um propósito ou objetivo subjacente que deve ser esmiuçado:

OBJETIVO 1: Perceber qual a evolução dos SI e em que ponto da evolução ocorreram os erros que contribuíram para a heterogeneidade entre eles. Perceber qual é a situação atual e quais as possíveis áreas de intervenção;

OBJETIVO 2: Primeiro perceber o que de facto é a interoperabilidade. Depois saber o ponto de situação atual e em conjunto com a questão anterior perceber quais são as possíveis áreas de intervenção/contributo para o presente trabalho;

OBJETIVO3: Investigar as funcionalidades e qual a é capacidade de um MAS para ser aplicado na troca de mensagens em formato HL7 v.2. Indagar sobre a aplicabilidade em contexto hospitalar e quais as propriedades que cada agente deve apresentar com vista a tal aplicação. Perceber qual é a melhor linguagem de programação para tal implementação;

OBJETIVO 4: Perceber como funciona um sistema de monitorização e quais as diversas estruturas que o sustentam. Perceber no contexto do CHP qual é o enquadramento de um sistema deste tipo;

OBJETIVO 5: No mercado atual de *frameworks* e mediante as necessidades existentes, perceber qual ou quais são as melhores soluções. Perceber também, quais as tendências atuais dos sistemas de monitorização no mundo empresarial;

OBJETIVO 6: Mediante a questão SOAP Vs. REST qual é a melhor solução conforme as necessidades e os meios existentes;

OBJETIVO 7: Indagar sobre o que é um sistema GDH e qual é a importância deste para o CHP. Perceber qual ou quais são os seus constituintes e ainda as necessidades existentes;

OBJETIVO 8: O ICD-9-CM é um constituinte do sistema GDH, daí tentar perceber qual a verdadeira função e como se processa a codificação para posterior desenvolvimento de uma aplicação Web;

OBJETIVO9: Investigar de entre as soluções *mHealth*, aquele que melhor cumpre as necessidades do CHP;

OBJETIVO 10: Mediante a informação a disponibilizar perceber quais as tendências das grandes empresas produtoras de *software* e como construir a aplicação que provoque interesse por parte do utilizador;

OBJETIVO 11: Mediante a confidencialidade de dados e qualidade dos mesmos, tentar perceber o que é necessário realizar para conseguir desenvolver um aplicativo para transportar informação para o exterior da unidade de saúde;

OBJETIVO 12: Mediante a literatura e todo trabalho desenvolvido tentar apontar um rumo para a interoperabilidade ao nível hospitalar;

OBJETIVO 13: Objectivo desta questão é apresentar ao leitor um resumo de todo o trabalho ao nível de soluções interoperáveis e que forma contribuem para o aumento da interoperabilidade;

1.4 ESTRUTURA DO DOCUMENTO

A presente dissertação está dividida em 10 capítulos: Introdução, Estado de Arte, Metedologias de Investigação e Ferramentas de Desenvolvimentos, Caso de Estudo - Sistema de Envio de Mensagens em Formato hl7, Caso de Estudo - Sistema de Monitorização de Agentes *Agéncia para a Integração, Difusão e Arquivo de Informação Médica (AIDA)* e HL7, Caso de Estudo - Formulário de Codificação ICD-9-CM, Caso de Estudo - Aplicação Móvel AIDA_Paciente, Prova de Conceito, Conclusões.

INTRODUÇÃO faz um enquadramento geral de todos os temas abordados assim como quais as motivações que sustentam todo o trabalho desenvolvido. Levanta assim questões que se pretende ver respondidas com execução do trabalho;

ESTADO DE ARTE através de referências bibliográficas, faz o enquadramento teórico de todo o trabalho desenvolvido. É importante também perceber ao nível da produção cientifica qual é o estado atual do trabalho desenvolvido;

METODOLOGIAS DE INVESTIGAÇÃO E FERRAMENTAS DE DESENVOLVIMENTO apresenta de uma forma geral todas as ferramentas utilizadas ao longo do presente projeto incluindo a metodologia de desenvolvimento utilizada;

CASO DE ESTUDO - SISTEMA DE ENVIO E RECEÇÃO DE MENSAGENS EM FORMATO HL7 apresenta o enquadramento, a metodologia, os materiais e os resultados alcançados com a implementação de um sistema MAS para o desenvolvimento de uma interface HL7 na versão 2 relativa à troca de mensagens clínicas;

CASO DE ESTUDO - SISTEMA DE MONITORIZAÇÃO DE AGENTES AIDA E HL7 apresenta o enquadramento, a metodologia, os materiais e os resultados alcançados com a implementação de um sistema de monitorização numa unidade de saúde;

CASO DE ESTUDO - FORMULÁRIO DE CODIFICAÇÃO ICD9 apresenta o enquadramento, a metodologia, os materiais e os resultados alcançados com a implementação de uma plataforma de codificação ICD-9-CM em conjunto com um sistema GDH.

CASO DE ESTUDO - APLICAÇÃO MÓVEL CPH_PACIENTES apresenta o enquadramento, a metodologia, os materiais e os resultados alcançados com a implementação de uma aplicação móvel e quais os benefícios para a unidade de saúde;

PROVA DE CONCEITO apresenta a análise SWOT de todos os casos de estudo apresentados assim como um questionário elaborado no caso do sistema de monitorização;

CONCLUSÕES para além de um resumo geral, faz a análise dos maiores contributos de cada caso de estudo e no final, responde às perguntas de investigação levantadas na introdução.

2

ESTADO DA ARTE

Ao longo do presente capítulo são apresentados todos os pressupostos teóricos considerados relevantes para a compreensão do trabalho desenvolvido assim como o estado atual das tecnologias e dos sistemas abordados. Embora sejam um pouco dispares, todos os temas estão relacionados com o trabalho desenvolvido, sendo assim relevantes ao enquadramento e apresentação do cenário atual nas instituições de saúde e investigação cientifica.

2.1 INTRODUÇÃO

A ordem pela qual as temáticas surgem é idêntica em todas as secções, apresentando um conceito mais geral do tema e de seguida focando-se nos pontos de interesse para o projeto.

Num projeto que pretende investigar e propor novas abordagens para a partilha de informação em contexto hospitalar servindo-se da interoperabilidade como meio para o alcançar é imperativo começar pela apresentação e ponto de situação dos sistemas de informação. Posto isto, aborda-se a questão dos *Sistemas de Informação em Saúde (SIS)* como são os casos do *Processo Clínico Eletrónico (PCE)* e do *Sistema de Informação para a Gestão de Doentes (SONHO)*. De realçar a importância de olhar não só para o panorama português mas também para trabalhos científicos internacionais, como ponto de partida para novas implementações.

Ao longo do capítulo, ficará demonstrada a importância da interoperabilidade para a partilha de informação e quais os pontos mais sensíveis, que carecem naturalmente de melhorias. É então dedicada uma secção à questão da interoperabilidade (definições e contextualização), no final dessa, são apresentadas diretrizes Health Level Seven e ICD9, que constituem dois marcos no que a interoperabilidade em sistemas de saúde diz respeito.

Desde 2011 surgiu na indústria uma nova revolução industrial ou *industry 4.0* como é apelidada pelo governo alemão. Nomeadamente as temáticas de *Cyber-Physical Systems* e *Medical Cyber-physical Systems* devem ser esmiuçadas encontrando pontos comuns e divergentes com os SIS. Foi então dedicada uma secção para apresentação do tema e definição de conceitos.

Nas últimas secções do capítulo são abordados os conceitos de *pervasive and ubiquituous computing* e *mobile health* com foco particular na soluções para dispositivos moveis (*mhealth*).

2.2 SISTEMAS DE INFORMAÇÃO

Possuir dados armazenados nos mais diversos formatos (papel ou formato digital) representa uma fonte importante de informação num mundo onde cada vez mais "informação e conhecimento são poder". Os *Sistemas de Informação (SI)*, apresentam-se assim como ferramentas importantes no processamento e apresentação da informação por forma a gerar conhecimento, em concordância com o sector da sociedade onde está inserido.

A propósito de conhecimento, Peter Drucker em 1959 introduziu o termo *knowledge worker* para alguém que no seu dia a dia cria, modifica e partilha conhecimento dentro de um sistema organizacional [8]. Este autor vaticinou ainda que o aparecimento deste novo conceito de trabalho, iria motivar alterações no paradigma da sociedade (*knowledge society*) onde o conhecimento e educação seriam pedras basilares. Passados mais de 50 anos, a visão de Drucker sobre os *knowledge workers* revelou-se acertada, sendo aceite pela grande maioria da comunidade cientifica. Já a visão deste para a sociedade não é completamente aceite existindo até diversos trabalhos que contrapõem esta visão [8].

As primeiras publicações por parte da comunidade cientifica sobre a temática dos sistemas de informação datam de 1960 e desde então um vasto caminho foi percorrido, motivado em parte pelas evoluções tecnológicas [8]. Um SI de acordo com Laudon e Laudon, é um conjunto de componentes inter-relacionados que cooperam para angariar, recuperar, processar armazenar e distribuir informações com a grande finalidade de facilitar o planeamento, o controlo, coordenação, a análise e o processo decisório das organizações onde está implementado [9].

Qualquer SI inclui uma entrada (input) onde existe a captação de dados em estado bruto, provenientes de fontes internas ou externas à organização [9]. O processamento refina estes dados para a forma mais adequada ao contexto, conduzindo à saída ou output destes. É importante reter que os sistemas podem apresentar um mecanismo de *feedback* para autocontrole de todas as operações em curso. A figura 1 representa o esquema geral dos componentes de um sistema de informação em qualquer formato. Todas as etapas apresentadas anteriormente estão contempladas na figura.

Um SI com base em unidades computacionais é uma combinação de Hardware, Software e *telecommunications networks* que formam um conjunto de componentes inter relacionados com o objectivo primordial de recolher, manipular e armazenar ou disseminar dados [10, 8].

Dados consistem em factos em bruto que por si só não significam muito para além de um inteiro ou uma *string*. Mas se estes dados forem organizados na forma correta (envolvidos num contexto), podem ter grande importância tornando-se assim informação [8]. Segundo

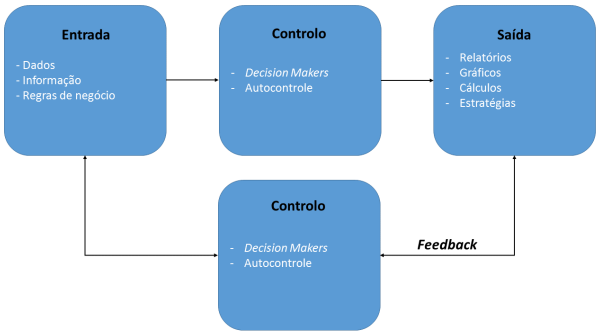


Figura 1.: Sistema de informação e seus diversos componentes (adaptado de [9]).

Ralph Satair e *George Reynolds*, os dados podem ser vistos como os estrados que suportam os carris numa linha férrea [10]. Por si só cada peça não apresenta significado aparente mas se forem ligadas a outras, cria-se um sistema de transporte, onde diferentes destinos são alcançados consoante a disposição dos constituintes. A mesma analogia pode então ser aplicada aos sistemas de informação, sendo importante reter que para organizar as diferentes peças que constituem os carris é necessário um conhecimento prévio do destino ou neste caso *(knowledge)*.

Numa abordagem um pouco diferente, Daniel Q. Chen e seus colaboradores em *"Information Systems Strategy: Reconceptualization, Measurment and Implications"*, constatam que sistemas de informação são constituídos pela arquitetura da tecnologia de informação, dados, sistemas de aplicação e pessoal responsável pela utilização de tais sistemas [11]. Também podendo ser incluídos nos SI tarefas de gestão de uma determinada organização criando sistemas híbridos entre componentes técnicas e atividades humanas [11].

Como abordado, os *Knowledge workers* são então as pessoas responsáveis por criar, utilizar e disseminar conhecimento sendo geralmente ligados a áreas como engenharia, ciências ou negócios [10]. *Knowledge management system* é então um grupo de pessoas, procedimentos, *software*, base de dados e ainda dispositivos, móveis ou não, utilizados com o intuito de criar, armazenar e utilizar todo o *knowledge* e experiência de determinada organização [10, 8].

Toda a informação para ser considerada valiosa deve conter algumas características consideradas importantes [10, 11, 8]:

ACESSÍVEL: com a autorização indicada toda a informação deve estar à distância de um *click*;

PRECISA: sem a existência de erros e informação desnecessária, geralmente estes são eliminados num processo de transformação;

COMPLETA: com base no conhecimento à priori, todos os factos importantes devem estar presentes;

ECONÓMICA: o custo de produzir informação deve ser relativamente reduzido;

FLEXÍVEL: a informação deve ser capaz de chegar a diferentes entidades em diferentes cargos e zonas, daí ter que se adaptar muito facilmente a novas situações;

RELEVANTE: para cada caso só deve aparecer informação especifica e não outra, que pode atrapalhar o processo de tomada de decisão;

CONFIÁVEL: os utilizadores devem confiar na informação que lhes é fornecida, devendo para tal numa primeira fase conquistar a sua confiança;

SEGURA: utilizadores não autorizados não devem poder consultar informação;

SIMPLES: toda a informação deve estar num formato simples e percetível por todos;

ATEMPADA: a informação deve ser entregue no momento oportuno, representando algo interessante nesse instante;

VERIFICÁVEL: toda a informação deve pode ser passível de ser investigada para averiguar se está de facto correta.

A estratégia de implementação dos sistemas de informação tem sido extensamente abordada, em parte, devido ao facto de ser extremamente complicado inserir um novo sistema deste tipo no seio de um organização [11, 12]. Nem sempre essa implementação corre de acordo com o esperado, existindo mesmo um grande histórico de insucessos a nível mundial [13].

Em *"Valuing Information Systems Investments"* Valacich e Schneider abordaram a temática de como as empresas avaliavam o investimento em SI e como estas beneficiam com essa implementação [14]. Essencialmente os possíveis ganhos podem ser agrupados em três grandes categorias, automatização de processos, informação ou aprendizagem e ainda definição de estratégias cada vez mais consistentes [14]. Relativamente ao primeiro ponto, a informação deve permitir aos agentes envolvidos completar as tarefas mais rapidamente. Com as definições corretas o sistema deve permitir aos seus utilizadores uma aprendizagem diária do trabalho a ser desenvolvido, acabando assim por beneficiar o primeiro ponto. Por último, o melhoramento de definição de novas estratégias acaba por ser a grande vantagem para as organizações [14]. Claro que para existir uma mudança de estratégia deve primeiro existir uma aprendizagem com os resultados obtidos, daí os três pontos estarem interligados.

Os tópicos do parágrafo anterior estão assim de certa forma subentendidos nos seguintes pontos:

- Diminuição de custos;
- Ajuda na tomada de decisão;
- Melhoramento do planeamento e controlo de operações;

- Informação com elevado grau de precisão, fiabilidade, e disponibilidade;
- Rápido processamento de tarefas;
- Maior facilidade de comunicação internas e externas à organização;
- Melhor utilização dos recursos disponíveis.

Tecnologias de Informação (TI), são então todos os meios tecnológicos controlados por ou para informação [8]. A comunicação (fios, cabo ou Wireless) é também ela um meio de tratar informação surgindo assim as *Tecnologias de Informação e Comunicação (TIC)* ou ICT (*Information and Communication Technology*), que englobam as TI e ainda quaisquer formas de comunicação, completando assim o conjunto *hardware*, *software* e telecomunicações.

Hoje em dia as TIC penetraram em quase todas as áreas de negócio e aspetos da vida quotidiana. A área da saúde não é exceção, representado um enorme filão de negócios para muitas empresas do setor público ou privado [1]. Neste sentido, a *Informática Médica (IM)* aborda temas do tratamento de dados com o intuito de gerar informação e posteriormente o conhecimento em medicina e saúde. De acordo com A. Hashman em *"A Systematic View on Medical Informatics"* a sua génese deve-se à tentativa de entendimento do papel das tecnologias de informação e comunicação na área da saúde [15]. Como objectivos da IM, poder ser apresentados dois, encontrar soluções para problemas de tratamento de dados, geração de informação e conhecimento [15]. E em segundo lugar entender quais a *guidelines* gerais do processamento de dados em medicina para gerar mais conhecimento e como referido anteriormente automatizar e resolver outros problemas existentes [15].

É então imperativo para o âmbito desta dissertação abordar os sistemas de informação na saúde mais concretamente em Portugal, tentado perceber o seu passado, o presente e quais as metas para o futuro. Só percebendo o estado atual dos sistemas é que podem ser levantados problemas e áreas de intervenção.

2.2.1 *Sistemas e Partilha de Informação em Contexto Hospitalar*

Sistemas de informação na saúde (SIS) ou *Health Information Systems (HIS)* são sistemas complexos de processamento de dados, informação e conhecimento que como próprio nome indica estão inseridos em ambiente hospitalar [16]. De acordo com o Gabinete Técnico do *Plano Nacional de Saúde (PNS)* no documento "Plano Nacional de Saúde" entende-se por sistema de informação em saúde, *"... não só ferramentas para as autoridades e profissionais de saúde, bem como, sistemas de saúde personalizados para utentes, tais como, o registo de saúde eletrónico do utente, a telemedicina e todo um conjunto de instrumentos de base tecnológica desenhadas para a prevenção, diagnóstico, tratamento, monitorização e gestão da saúde do cidadão."* [17]. Ao longo do documento supra mencionado é evidente o ênfase dado às tecnologias de

informação e comunicação. Na mesma linha de pensamento, Rita Espanha no documento "Adenda à Análise Especializada: Tecnologias de Informação e Comunicação", atesta a extrema importância dos sistemas de informação na área da saúde a nível nacional e internacional [18].

A comunidade científica foca atualmente atenções em perceber o ponto actual dos sistemas de informação em saúde e ainda quais os ganhos diretos e indiretos da sua implementação. No trabalho proposto por Don E. Detmer, *"Information technology for quality health care: a summary of United Kingdom and United States experiences"*, o autor atesta as melhoras substanciais com a introdução de tecnologia ligada à informação na saúde no Reino Unido e nos Estados Unidos da América [3].

Uma das maiores evoluções nos sistemas de informação ao nível da saúde e não só, foi a mudança da base em papel para o processamento e armazenamento em sistemas computacionais. As vantagens deste tipo de suporte são por demais evidentes, sendo a principal desvantagem o custo inicial de tal implementação uma vez que, *hardware*, *software* e toda uma rede deve ser criada com vista a sua implementação [16].

Nos anos 60, 70 e 80 as intenções da informática médica estavam voltadas para pequenas unidades dentro do próprio hospital (laboratórios, radiologia ou administração), nos chamados sistemas de informação departamentais [16]. No final anos 90 e década de 2000, este foco mudou e passou a tratar-se a informação como um todo dentro de uma unidade de saúde (sistemas de informação hospitalar) [16]. Atualmente o foco inside na partilha de informação, principalmente em aspetos ligados ao histórico médico de determinado paciente como forma de criar redes regionais ou até nacionais de partilha de informação [16].

Os Sistemas de Saúde são complexas redes de interação entre diversos intervenientes, sendo o potencial das TIC um meio para promoção da comunicação e partilha de conhecimento [9]. A utilização destas na Saúde em Portugal pode ser agrupada em três grandes blocos, o e-saúde ou *e-health*, Sistemas de Informação em Saúde, *Media* e comunicação em saúde [18, 17].

O bloco *Electronic Healhtcare* (ver figura 2) refere-se à utilização de TIC na formação de um novo modelo de ligação entre pacientes e profissionais de saúde que assenta nos princípios da partilha de informação [19]. Para efeitos do Plano Nacional de Saúde, *e-health* entende-se por " *todas as redes de informação e comunicação sobre saúde, disponíveis online, dirigidas ao público em geral, mas também a profissionais de saúde, construídas em torno desta temática, assim como, todo o tipo de prestação de serviços, de informação, de construção de plataformas, de disponibilização de conteúdos e registo eletrónico de pacientes/utentes, mas que estão disponíveis online, para serem consultados por toda a população utilizadora da Internet, diretamente ou através de terceiros*" . Assim sendo é imperativo desenvolver ferramentas acessíveis por meios eletrônicos, que disponibilizem informação de qualidade, segura, adequada a cada situação e ainda os registos pessoais de saúde do cidadão [17]. De acordo com Rita Espanha

o conceito de e-saúde consegue maximizar a eficiência dos cuidados de saúde, qualidade dos serviços prestados, padronização da informação disponível assim como o estreitar laços entre paciente e profissional de saúde [18].

A Internet é um dos principais veículos destes sistemas e a utilização de aplicações e ferramentas *on-line* realça a importância da *Web 2.0* assim como saúde 2.0 ou Medicina 2.0 [17]. Num estudo levado a cabo por H. K. Andreassen em *"European citizens user of E-healht services: a study of seven countries"*, colocava em 2007 Portugal como um dos países com menor taxa de utilização de páginas web relacionadas com aspectos da saúde. A mesma publicação apresenta como possíveis causas, questões culturais, falta de páginas com conteúdo apropriado e fidedigno ou então a baixa idade dos utilizadores de internet que motiva um relaxamento quanto a aspectos relacionados com o estado de saúde [19].

Segundo o Alto Comissariado para a Saúde, os SIS são facilitadores diretos não só de toda a atividade clínica mas também do próprio Sistema de Saúde [17]. Destacando-se alguns os seguintes exemplos [17]:

- Face a ameaças epidemiológicas bem como catástrofes naturais e humanas, todo o sistema necessita de uma rápida reorganização;
- Fornecer dados necessários à monitorização de desempenho e estudos económicos;
- Consolidação dos índices *case-mix*;
- Dados relativos a gestão de pessoal;
- Melhorar *accountability* e processos de decisão e políticas *evidence-based*;
- Reorganização de fluxos logísticos e do aprovisionamento.

O segundo ponto diz respeito ao SI(ver figura 2), a entrada destes em Portugal, data do ano de 1994 na área da medição da produtividade [18]. A partir daí, com a reforma do *Serviço Nacional de Saúde (SNS)*, os SI foram um dos grandes impulsionadores da melhoria da acessibilidade, eficiência, qualidade, continuidade de cuidados de saúde e aumento da satisfação dos cidadãos e profissionais de saúde [20]. Ficando ciente que um implementação SI em saúde não se destina apenas a um determinado *profile* de utilizadores.

A Faculdade de Medicina da Universidade do Porto destaca no seu relatório sobre SIS , que determinado sistema pode estar orientado para os diversos tipos de utilizadores, entre os quais [6]:

- Pessoal administrativo;
- Profissionais de Saúde - médicos, enfermeiros entre outros;
- Gestores - administração clínica, diretores serviço;

- Utentes.

Os SIS devem estão assim envoltos numa filosofia de trabalho que permita a criação, gestão, atualização e utilização dos registos eletrónicos de utentes e outras informações de interesse para a unidade de saúde. Fazendo sempre uso de disposições legais que fomentem a confidencialidade, a segurança, a propriedade, a partilha e distribuição de informação. É importante que toda a instituição seja envolvida no processo informativo, optando-se assim por uma abordagem inclusiva.

O último dos três blocos referidos anteriormente, abrange os media e a comunicação em saúde (ver figura 2). De acordo com Rita Espanha, as mensagens sobre saúde divulgadas pelos media massificados, contribuem de forma determinante para a construção individual e coletiva das representações sobre saúde e unidades hospitalares [18]. A massificação da Internet criou canais informativos sem precedentes capazes de publicitar de forma positiva ou negativa aspetos de toda a sociedade, não sendo a saúde exceção [17]. Assim sendo a área da saúde não pode nem deve passar ao lado desta preciosa ferramenta de disseminação de informação clínica e até de incentivo à cidadania.

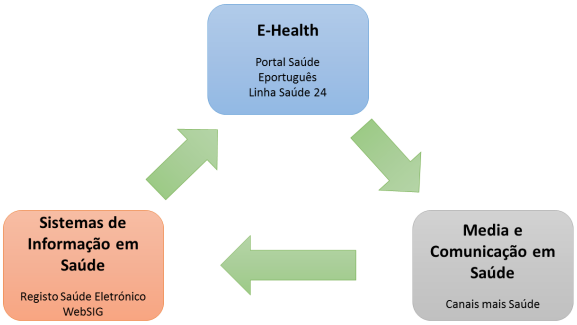


Figura 2.: Esquema de relações entre três blocos principais das TIC (adaptado de [17]).

Na figura 2 é possível observar um resumo daquilo que foi abordado anteriormente, com as relações entre os diversos blocos. No interior destes, estão alguns exemplos de sistemas implementados em Portugal.

Focando atenções ao nível da situação atual da utilização das TIC na saúde, três grandes desafios emergem: o aumento do número de sistemas *user friendly*, aumento da interoperabilidade entre sistemas e por fim o diagnóstico e o registo da utilização das tecnologias de informação e comunicação na saúde. O enquadramento dos atuais e novos sistemas com o quadro normativo vigente é primordial. Ao nível europeu, a Diretiva 2011/24/EU do Parlamento e do Conselho de 9 de Março de 2011 relativa ao Exercício dos Direitos dos Doentes em Matéria de Cuidados de Saúde Transfronteiriços de Saúde é exemplo disso mesmo existindo no artigo 14 da Diretiva referência a uma rede voluntária de autoridades responsáveis pela *e-health* [17]. Entre os objetivos desta estão a criação de aplicações com a capacidade de trabalhar em conjunto, garantindo a confiança do utilizador.

O PCE é um exemplo de um sistema de informação, introduzido com o objectivo primordial de substituir o tradicional registo em papel *Paper Clinical Process* [7]. Este registo pode então abranger todas as áreas hospitalares, desde um relatório sobre um determinado paciente, até um relatório sobre o balanço de contas semestral [21]. Segundo T. D. Gunter e N.P. Terry em *"The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions"*, um registo clínico eletrónico é *"an evolving concept defined as a systematic collection of electronic health information about individual patients or populations. It is a record in digital format that is capable of being shared across different Healthcare settings, by being embedded in network- connected enterprise-wide information systems. Such records may include a whole range of data in comprehensive or summary form, including demographics, medical history, medication and allergies, immunization status, laboratory test results, radiology images, vital signs, personal stats like age and weight, and billing information."* [22]. Da citação apresentada é possível retirar dois conceitos, primeiro o de *Registo Clínico Eletrónico (RCE)* não é estanque no tempo, existindo uma evolução por forma a acompanhar os avanços na área da tecnologia e da medicina. Segundo ponto é a capacidade da informação ser partilhada por diferentes plataformas e sistemas não sendo assim estanque a uma unidade de saúde.

Os primeiros registos de um sistema com desígnios apresentados até agora surgiram na década de 60, embora o verdadeiro foco não fosse inteiramente a informação clinica serviu como um impulso determinante para alterar o rumo dos acontecimentos [18]. Em Portugal o primeiro registo eletrónico em ambiente hospitalar, surgiu com o SONHO dedicado apenas a informação administrativa [17, 7]. Desde então surgiram entre outros, o *Sistema de Apoio ao Médico (SAM)* e *Serviço de Apoio à Prática de Enfermagem (SAPE)*. Existem mais embora com expressão muito menor do que estes supramencionados, visto muitas das vezes resultarem de aplicações produzidas dentro da própria unidade de saúde [7].

É errado colocar PCE e RCE no mesmo patamar. Este último, refere-se apenas à informação produzida num encontro clinico/paciente enquanto o primeiro, reúne toda a informação proveniente de todos os pontos de uma ou várias unidades de saúde por onde esse paciente passou facultando a análise transversal do seu histórico e atualmente bastante ligado a ferramentas de apoio à decisão [21, 1, 23]. O RCE é apenas uma função entre várias como por exemplo : fonte de informação para pessoal clínico ou sistemas de apoio à decisão; redução da frequência de perda de informação e de erro médico; melhoramento da forma de comunicação com fontes externas; entre outras. A função primária de um RCE ou EMR é o armazenamento de informação sobre a saúde e doença de um paciente após ter dado entrada na unidade hospitalar, utilizando anotações de médicos e enfermeiros sobre tratamentos, diagnósticos e processos em que paciente esteve envolvido [24].

Em muitas das unidades de saúde do Serviço Nacional de Saúde coexistem sistemas de informação locais, desenvolvidos pela própria unidade de saúde, com a aqueles apresenta-

dos anteriormente destinados a ser implementados em todo o SNS. São assim colocados entraves no estabelecimento de políticas homogéneas e transversais a toda unidade de saúde. É então necessário definir mecanismos de interoperabilidade (normas, requisitos, codificações, etc.) que diminuam os riscos de comunicação e de implementação de SI .

Existe na implementação de um novo *Electronic Health Record (EHR)* um elevado risco de falha, motivado em parte pela integração com outros HIS e ainda pelos custos inerentes a todo o processo, registando-se um maior número de insucessos do que sucessos nesta temática [1]. É natural que se existir uma inadequada interface com o utilizador ou uma má projeção do sistema em sí, as hipóteses de sucesso são diminutas [25]. O sucesso de uma implementação é decidido em ambiente prático, pelos utilizadores ao qual este se destina, segundo diversas dimensões: eficiência, efetividade, empenho por parte dos participantes, satisfação dos utilizadores e dos pacientes [25]. Uma característica importante deste tipo de sistemas é a de ter de estar em todo o lado ao mesmo tempo e com informação atualizada (ubiquidade) [7].

Um exemplo no âmbito do *e-health* é o Portal da Saúde que disponibiliza informação sobre o Sistema de Saúde, apresentando informações de relevo para o paciente como a organização do sistema de saúde, prestadores dos cuidados de saúde, redes de cuidados primários e continuados. Apresenta também informações relativas à saúde pública e legislação em vigor.

A questão da interoperabilidade é imperativa no contexto da partilha de informação. Serve não só de motivação, mas também de objetivo e a grande parte do presente projeto está apresentada em detalhe na próxima secção.

2.3 INTEROPERABILIDADE

Grande parte dos problemas na implementação das TIC tem causa bem definida, a capacidade de dois sistemas comunicar e interagir plenamente. Pela sua relevância e pertinência esta secção será inteiramente dedicada aos vários aspetos do conceito de interoperabilidade.

2.3.1 *Enquadramento Teórico*

Interoperabilidade não é um conceito somente utilizado em HIS, pode ser aplicado desde relações inter-pessoais até relações entre maquinaria pesada de uma qualquer unidade fabril. Assim sendo, podem ser encontradas diversas definições ou abordagens, que devem ser exploradas. Segundo Nancy Ide e James Pustejovsky, *"Broadly speaking, interoperability can be defined as a measure of the degree to which diverse systems, organizations, and or individuals are able to work together to achieve a common goal. For computer systems, interoperability is typically defined in terms of syntactic interoperability and semantic interoperability."* [26]. De

acordo com a citação a interoperabilidade é uma medida sobre a forma como os sistemas são capazes de trabalhar em conjunto, sendo a definição aplicável como dito anteriormente a diversas topologias diferentes. Da área da engenharia provém a seguinte definição, proposta pela organização IEEE fazendo a ponte com a área biomédica, *"interoperability is the ability of two or more systems that is used to exchange information and to use this information that has been exchanged. In the field of* Electronic Health (eHealth) *applications, Interoperability means that the ability to communicate and exchange data accurately, effectively, securely and consistently with different information technology systems, software applications, and networks in various settings and exchange data such that clinical or operational purpose and meaning of the data are preserved and unaltered"* [27]. O começo da situação, aponta para a troca de informação e para a sua utilização. Fazendo posteriormente referencia ao campo das eHealth, apontando ainda quatro pilares que considera essenciais *accurately, effectively, securely, consistently* (precisão, eficiência, segurança e consistência). Por seu turno Muhmmad Ehsan Mansoor atesta que, *"Interoperability in health care information is the capability of health information systems to work together within and across organizational boundaries"* [27] . Neste caso a definição é completamente voltada para a informação em saúde, atestando que sistemas de informação devem trabalhar em conjunto dentro e fora das unidades de saúde.

Todas as definições apresentadas anteriormente têm pontos em comum, nomeadamente quando se referem à capacidade de trabalho em conjunto (homens ou máquinas) e no caso das duas últimas, à partilha de informação em contexto de saúde. A primeira definição por ser mais lata acaba por estar subjacente às outras duas, nomeadamente quando aborda os dois tipos de interoperabilidade. Por último o facto de na definição proposta por Mansoor, este aborda a questão das fronteiras da organização marca aqui uma diferença com as demais e com uma temática que ainda não é bem aceite pela comunidade cientifica, uma vez que levanta questões relacionadas com confidencialidade e proteção de dados relativos a pacientes.

A interação entre diferentes SI pode ser divida em três níveis de interoperabilidade[27]:

FÍSICA: neste nível, apenas é necessário que o sistema de informação A, envie dados para o B, sem que este necessite de os perceber;

CONTEÚDO: a este nível existe uma preocupação com o formato e conteúdo de informação que está a ser transmitida. Garantir que os dados de um exame médico como caso de ECG (eletrocardiograma), podem ser posteriormente utilizados por outro qualquer SI é neste ponto uma preocupação. Ferramentas como Health Level Seven ou *International Classification Diseases 9 Revision (ICD-9)* (abordadas em detalhe nas próximas secções) são exemplos de ferramentas utilizadas. Muitas vezes neste nível assiste-se à implementação de bases de dados comuns para armazenar a informação nos formatos referidos anteriormente, sendo posteriormente acedida por outros HIS;

ESPECIFICAÇÃO: utilizando os padrões apresentados no ponto anterior (*Health Level Seven International (HL7)* v.2, *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)*), pode ser alcançada a interoperabilidade entre soluções de *software* que utilizem base de dados diferentes, como acontece no caso de sistemas distribuídos.

Um exemplo de aplicação da interoperabilidade resulta do trabalho de G. Vetere e M.Lenzerini em *"Models for Semantic Interoperability in Service Oriente Architectures"* [28], onde abordam a questão de que as TIC modernas são baseadas em arquiteturas SOA (service-oriented architectures). Formadas por uma rede de provedores de serviços, que podem ser requisitados por clientes existindo para tal *descriptive schemas*, nestes deve ser realizado uma pesquisa antes de efetuar determinado pedido. Os requisitantes devem então ser capazes de entender e perceber a descrição e propriedades dos serviços que vão pedir, ficando a descrição e responsabilidade disso do lado do programador aquando o desenvolvimento e recorrendo a estruturas como *Extensible Markup Language (XML)*. Há neste cenário uma camada semântica ou *semantic layer* que assegura que toda a informação contida nas mensagens é interpretada corretamente [28].

O parágrafo anterior aborda a temática da *layer* ou camada, alguns autores consideram que interoperabilidade tem duas camadas (sintática e semântica), outros consideram três (técnica, semântica e processo). Pode ser observada uma breve descrição destas na seguinte lista [29, 30]:

TÉCNICA OU SINTÁTICA: representa o movimento de dados do ponto de origem até ao ponto de destino. Independente do domínio, ou seja, não existindo qualquer ligação com o conteúdo dos dados a serem trocados. Claro que esta camada está na génese de qualquer outra que possa surgir acima desta;

SEMÂNTICA: a informação deve aqui apresentar significado tanto para o sistema informático de destino como para algum humano que utilize esse sistema. Geralmente esta fase envolve códigos e outros identificadores constituindo a base para muitas soluções de interoperabilidade implementadas nível hospitalar;

PROCESSO: coordena tudo de tal forma que processos no sistema de origem e destino também estejam sincronizados com a troca de informação. Devem então estes sistemas informáticos estar enraizados de tal forma no quotidiano das organizações que a sua utilização passa a ser *ubiquitous* (permanente).

Na literatura é possível encontrar diversas formas de classificar interoperabilidade com base em modelos distintos. Entre os mais conhecidos está o *Levels of Conceptual Interoperability Model (LCIM)* focado essencialmente em graus de sofisticação da interoperabilidade presente num SI , fazendo um agrupamento em sete níveis [31]:

NÍVEL 0 - INTEROPERABILIDADE INEXISTENTE: não existe qualquer ponto de contacto entre SI para além de um operador humano;

NÍVEL 1 - INTEROPERABILIDADE TÉCNICA: partilha de informação utilizando protocolos bem definidos como *Hyper Text Markup Language (HTML)* ou XML;

NÍVEL 2 - INTEROPERABILIDADE SINTÁTICA: embora os SI não entendam o significado da informação a ser partilhada, conseguem compreender e aceitar o formato em que está a partilhada;

NÍVEL 3 - INTEROPERABILIDADE SEMÂNTICA: existe neste nível o entendimento daquilo que está a ser partilhado. O valor da informação representa o mesmo em qualquer SI que esteja inserido;

NÍVEL 4 - INTEROPERABILIDADE PRAGMÁTICA: para além de uma perceção da informação existe um conhecimento mútuo da forma como só os SI estão organizados;

NÍVEL 5 - INTEROPERABILIDADE DINÂMICA: neste momento a complexidade dos SI já é elevada uma vez que este nível consegue adaptar-se à mudança de significado da informação sempre que esta acontece;

NÍVEL 6 - INTEROPERABILIDADE CONCEPTUAL: neste nível os sistemas devem estar de acordo com as premissas e restrições do ambiente real. Podem ser implementados em diferentes ambientes estando cientes do que se passa com seus semelhantes a qualquer momento.

É importante referir que todos os níveis herdam as características dos anteriores aumentando em muito a dificuldade de implementação dos mesmo. Na próxima secção é abordada uma solução que surgiu para tentar colmatar os problemas da heterogeneidade da informação hospitalar.

2.3.2 *Implementação em Contexto Hospitalar*

A melhor palavra para descrever os sistemas de informação em saúde é heterogeneidade, motivado por anos em que o desenvolvimento se cingia a um departamento em especifico dentro da unidade hospitalar e ainda pelo facto, de estes contactarem diretamente ou estarem inseridos em equipamentos médicos [32].

Existem diversos trabalhos que promovem tanto a correção em cenários de interoperabilidade deficiente, como a implementação de novos HIS que já compreendem os conceitos abordados na secção anterior. Cada unidade de saúde é um caso muito especifico, daí que o mas o mais interessante é perceber como podem ser implementados alguns desses conceitos, de forma a avaliar as diferenças entre o antes e depois de uma implementação.

A *Agéncia para a Integração, Difusão e Arquivo de Informação Médica (AIDA)*, surgiu numa tentativa de colmatar um problemas de interoperabilidade e uniformização existente em

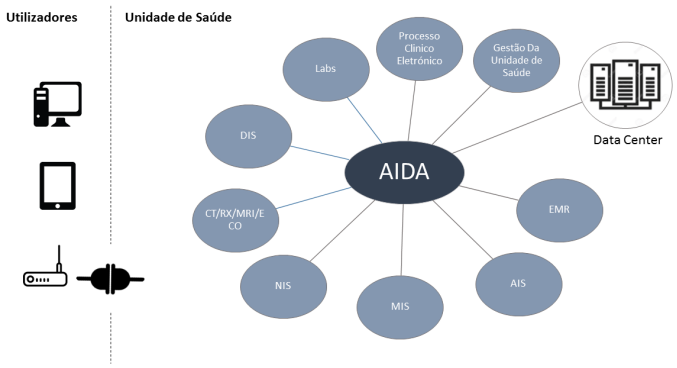


Figura 3.: AIDA- esquema geral de ligações (adaptado de [32, 7]).

sistemas clínicos e ainda reduzir a complexidade que advém da utilização de diferentes fontes de informação (foro clínico, administrativo, tesouraria, agendamentos, entre outras). Existem por vezes, enormes limitações na comunicação de informação, motivada em alguns casos pelo mau estado das infraestruturas, por limitações do foro técnico (*software* ou máquinas desatualizadas) ou até questões monetárias. A solução encontrada é integrar, difundir e agrupar esta informação num sistema dinâmico que quando necessário a consiga transmitir de forma eficaz [32].

AIDA é então uma ferramenta desenvolvida por investigadores da Universidade do Minho que utiliza tecnologias SOA e diretrizes HL7 em conjunto com sistemas multi agente (*Multi-Agent System*(MAS)) [33]. Atualmente está presente em 4 unidades hospitalares em Portugal, Centro Hospitalar do Porto, Centro Hospitalar do Alto Ave, Unidade Local de Saúde de Castelo Branco e Unidade Local de Saúde do Norte Alentejano [7]. De uma forma geral, um agente é um programa de computador capaz de acções independentes em nome do utilizador ou de si próprio [34]. Um sistema multi agente representa uma coleção de agentes que interagem entre si geralmente utilizando a Internet ou uma rede local [34].

A plataforma é então composta por sub-sistemas especializados, responsáveis por tarefas como: comunicação com sistemas heterogéneos, enviar e receber informação (relatórios médicos, imagens, receitas médicas), armazenar e gerir informação e responder a pedidos [33, 32]. Na figura 3 é possível observar alguns dos serviços com quem a AIDA interage e a forma como processa a informação para o utilizador. É ainda possível observar na figura, que uma das ligações implementadas, destina-se ao registo clínico eletrónico, funcionando como um repositório de informação que diz respeito a determinado indivíduo. *Agéncia para a Integração, Difusão e Arquivo de Informação Médica - Processo Clínico Eletrónico (AIDA-PCE)* é então um modulo da AIDA que se dedica ao EHR abordado através de uma "*subjective, objective, assessment and planing framework*" [7]. Este eixo AIDA-PCE é o aquele que apresenta maior relevância para efeitos de investigação assim como criação de ferra-

mentas de apoio à decisão no *Centro Hospitalar do Porto (CHP)* .Neste ponto, é vital definir o conceito de episódio. Episódio é então o conjunto de operações e intervenções realizados num paciente do início até final do tratamento [17].

As dificuldades de integração e de interoperabilidade dos HIS são por demais evidentes. A AIDA surge então como uma forma de resolver alguns destes problemas, devendo seu sucesso, em grande parte à flexibilidade do modelo [33]. Comparar modelos ou implementações de soluções neste campo é complicado, uma vez que cada situação depende do local onde está inserida e da quantidade de meios alocados para o projeto. Um ponto essencial é então a normalização e utilização de informação em concordância com *standards* definidos.

Ao nível de *standards*, as diretrizes propostas pela *Health Level Seven International (HL7)*, são talvez, as mais reconhecidas internacionalmente, no entanto, existem uma panóplia de soluções bastante abrangentes ao nível de mensagens (HL7), terminologias (SNOMED-CT), diagnósticos e procedimentos clínicos (ICD9), informação clínica e registo clínico eletrónico (openEHR, HL7 CDA) e ainda ao nível da imagem médica (*Digital Imaging and Communications in Medicine (DICOM)*) [35, 33, 36, 27]

Os diversos *standards*, supra mencionados, podem ser agrupados em três blocos, o primeiro o da representação e informação clínica, o segundo referente à comunicação e por último o bloco da imagem médica [37, 29].

Alguns destes sistemas vão ser abordados em maior detalhe ao longo das próximas secções. HL7 foca-se no domínio clinico e administrativo, NCPDC dedicado à área da farmácia, IEEE dispositivos médicos, ACR/NEMA dedicado à imagiologia, X12 dedicado a questões de segurança, ADA dedicado à area dentista [38]

2.3.3 *Impacto da Interoperabilidade nas Unidades de Saúde*

"Interoperability will bind together a wide network of real-time, life-critical data that not only transform but become health care" . A frase que deu inicio a esta sub-secção pertence a DJ. Brailer, sendo retirada do documento "*Overview of Healthcare Interoperability Standards*" [36]. Neste documento são abordadas diversas temáticas entre as quais os benefícios da interoperabilidade nos cuidados médicos na Irlanda. Existem outros trabalhos que apresentam e testemunham a importância do conceito de interoperabilidade. Luciana Cardoso e seus colaboradores, em *Healthcare Interoperability through Intelligent Agent Technology* aborda a temática dos agentes HL7 na prespectiva de controlo e monitorização [33]. Hugo Peixoto e seus colaboradores, apresentam *Intelligence in Interoperability with AIDA* abordando o impacto da implementação de um sistemas como AIDA nos HIS [32], por fim Ai Jin e Lars Ahlfors apresenta *Interoperability in Eletronic Medical Records*, onde aborda esta questão com o foco nos registos do paciente [39].

Atualmente qualquer implementação de uma TIC sem interoperabilidade não é viável, existindo um vasto leque de benefícios desde a área médica até à administrativa, passando inclusive pela financeira. A melhor forma de apresentar todos os benefícios encontrados na bibliografia é sob a forma de lista [36, 29, 30, 27]:

- Os pacientes beneficiam de melhores cuidados de saúde, desprovidos de erro e muito mais informados. Quando a interoperabilidade atravessar fronteiras este impacto poderá ser ainda maior;
- A utilização de meios e sistemas de prescrição eletrónica poderá evitar erros de prescrição, erros de dosagem e acima de tudo evitar a fraude medicamentosa. A utilização de sistemas de apoio à decisão é sem dúvida um trunfo na prescrição de medicamentos por exemplo;
- Profissionais de saúde têm acesso à informações atualizadas de todos os pacientes diminuindo assim a probabilidade de cometerem um erro. Os sistemas de apoio à decisão neste ponto voltam a ser uma ferramenta bastante útil. Aumenta assim também o seu bem estar no local de trabalho;
- Redução da inserção de informação por parte de profissionais de saúde e pacientes, uma vez que está tudo acessível num ponto comum;
- As companhias de seguro e outros provedores de saúde beneficiam com a redução do erro médico e com diagnósticos antecipados e corretos;
- Aumenta assim a abertura do mercado do *software* para saúde uma vez que o facto de existir um mercado único diminui o custo de produzir para diferentes ambientes;
- Aumenta da eficiência do sistema como um todo. Desde profissional de saúde que não perde tempo a procurar informação. Trabalhador dos sistemas de informação que sabe onde esta erro de determinada máquina. Funcionário da recepção que não precisa de pedir sempre mesma informações ao paciente, tudo contribui para um acelerar de processos e melhorar das condições de atendimento.

Interoperabilidade só é alcançada com a utilização de *standards* consistentes que definam de forma sintática e semântica a organização. Em algumas áreas da indústria como por exemplo a financeira, elevados graus de integração foram atingidos utilizando padrões como SWIFT [40]. Na área da saúde, organizações como HL7 facilitam o nível de integração, mas ainda assim existem lacunas no que a interoperabilidade diz respeito. Embora o tremendo investimento de tempo, recursos e fundos um pouco por todo o mundo com o objetivo de interoperabilidade conceptual, ainda raramente é atingido.

Nas próximas duas secções serão abordados dois processos normativos, o HL7 relacionado com conceitos de comunicação de informação e ICD9-CM utilizado para codificar

diversos aspetos do tratamento dos pacientes e para utilizar em *Grupo de Diagnósticos Homogêneos (GDH)*.

2.4 STANDARD DE COMUNICAÇÃO HL7

Health Level 7 (HL7) é uma organização internacional para desenvolvimento de *standards*, sem fins lucrativos presente em mais de 31 países, estando Portugal entre eles [29, 41]. Fundada à mais de 30 anos (1987), a organização pauta pelas seguintes características, *"ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services. HL7 is supported by more than 1,600 members from over 50 countries, including 500+ corporate members representing healthcare providers, government stakeholders, payers, pharmaceutical companies, vendors/suppliers, and consulting firms."* [42, 43]. Da citação anterior é possível observar um foco na partilha, integração e recuperação de informação relativa a saúde utilizando o formato eletrónico. Atesta também, a quantidade de utilizadores dos diversos quadrantes que utilizam as diretivas propostas pela organização. A missão da organização, de acordo com a sua página oficial da Internet é *" HL7 empowers global health data interoperability by developing standards and enabling their adoption and implementation."* [44]. Fazendo uma alusão clara à relação direta com a interoperabilidade.

Como referido na primeira citação, HL7 é acreditada pela *American National Standards Institute (ANSI)*, fazendo parte de um conjunto de SDO que operam na área das TIC. A organização colabora também com a ISO TC215 e CEN TC251 da comunidade europeia e ainda com a IHTSDO (SNOMED terminology) e CDISC (testes clínicos) através de SDO *Global Health Informatics Standardization* [29].

O número sete na sigla da organização deriva do sétimo nível *Organização Internacional para a Normalização (OSI)* [43, 45]:

CAMADA 7- APLICAÇÃO: permite o acesso da aplicação aos serviços *Network* como por exemplo acesso a ficheiros remotos, gestão da network e mensagens eletrónicas;

CAMADA 6- APRESENTAÇÃO: responsável pela formatação de dados para ser utilizado na camada seguinte;

CAMADA 5- SESSÃO: garante a sessão entre diferentes máquinas;

CAMADA 4- TRANSPORTE: garante a entrega de mensagens sem erro. Garante que os protocolos seguintes não se preocupem com transferência de dados;

CAMADA 3- REDE: ligado a aspetos de controlo de sub rede e qual o caminho físico que os dados devem seguir;

CAMADA 2- *DATA-LINK*: garante a transferência de dados sem erros entre camadas físicas;

CAMADA 1- FÍSICA: responsável pelo envio e receção de *bits* sem estrutura definida.

O nível 7, é assim o único a lidar com aspetos de semântica e significados dos dados trocados, ficando a cargos dos níveis inferiores o encapsulamento, vital neste modelo OSI [28]. Quando os dados entram no nível sete são continuamente envolvidos por novos envelopes até atingir o nível mais baixo [45]. No ponto de receção, o mesmo processo acontece em sentido inverso.

Não podem ser comparadas implementações das diretrizes HL7, cada organização faz a implementação consoante a estrutura dos seus HIS e as condições do seu Hardware [46]. O principal objetivo é simplificar a implementação de interfaces comunicativas entre *software* de vários vendedores, simplificando o processo de comunicação de interfaces.

Nos Estado Unidos da América 90% das instituições de saúde utilizam diretrizes HL7, de três formas distintas [47]:

1. Responsáveis pela movimentação de dados clínicos ou pela criação de aplicações que partilham dados entre sistemas;
2. Entidades governamentais que pretendem a partilha de informação entre diferentes setores;
3. Área da informática médica onde são implementadas ontologias ou outros sistemas de classificação para uniformização de dados clínicos.

Embora a organização não produza *software* disponibiliza um conjunto de especificações e diretrizes que conduzem as organizações à interoperabilidade [29]. *HL7 International* considera como seus *standards* primários [43]:

- ***Version 2.x Messaging Standard***, promove a interoperabilidade em transações de informação;
- ***Version 3 Messaging Standard***, promove a interoperabilidade em transações de informação;
- ***Clinical Document Architecture (CDA)***, modelo para troca de documentos clínicos com base HL7 V.3;
- ***Continuity of Cate Document (CCD)***, especificação para troca de sumários clínicos com base HL7 V.3;
- ***Structured Product Labeling (SPL)***, informação que acompanha medicamentos com base HL7 V.3;
- ***Clinical Context Object Workgroup (CCOW)***, integração de aplicações visuais.

De todos os pontos da lista anterior sem sombra de dúvida que o mais utilizado a nível mundial é o HL7 na versão 2, através de interfaces que possibilitam a troca de mensagens nesse formato. Assim sendo na próxima secção aborda-se em maior detalhe essa especificação.

2.4.1 *Estrutura Linguagem Health Level Seven V.2*

Aquilo que começou como um modelo para comunicação intra hospital, tornou-se numa ferramenta indiscutível, servindo de modelo para a indústria da saúde quando se pretende partilhar informação. A versão 2 também conhecida como *Pipehat* foi apresentada em 1989 [48].

A primeiras versões de HL7 de *standard*, V.2, eram vagas e com pouca documentação comparado com aquelas disponíveis atualmente, a mais recente das quais, a V2.9. [48], foi criada essencialmente por especialistas em interfaces clínicas e apresentando a mesma modelação, complexidade e consistência interna do que a versão mais recente V3 [29]. Um ponto que ainda hoje é uma desvantagem é a falta de uma ontologia formal unificando assim os diversos conceitos a serem trocados pelas diversas mensagens entre diferentes interfaces [40]. As interfaces são desenhadas 80% com base em especificações HL7 e 20% de adaptações ao local onde está inserido.

Uma mensagem é assim uma estrutura hierárquica, geralmente associada a um determinado evento, capaz de despoletar ações. Esse *trigger event* está relacionado com situações especificas da prestação de cuidados de saúde que originam a troca de dados entre diferentes sistemas. A mensagem apresenta uma coleção de segmentos, com regras especificas de repetição e inclusão.

```
MSH|^~\&|AcmeHIS|StJohn|ADT|StJohn|20060307110111||ADT^A04|
MSGID20060307110111|P|2.4 EVN|A04
PID|||12001||Jones^John||19670824|M|||123 West St.^^Denver^CO^80020^USA
PV1||O|OP^PAREG^||||2342^Jones^Bob|||OP|||||||||2|||||||||||||
||||||||||20060307110111
| AL1|1||3123^Penicillin||Produces hives~Rash~Loss of appetite
```

Na exemplo acima representado, é possível observar uma mensagem que utiliza o formato HL7 V2. Cada linha representa então um segmento, contendo campos ou *fields* separados por um delimitador normalmente uma barra vertical. Um campo separado por delimitadores pode conter sub delimitadores, normalmente um assento circunflexo, para separar sub campos [49]. Os sub componentes são diferenciados por & enquanto a repetição de um componente é indicada utilizando ~ [49].

O primeiro segmento de todas as mensagens é chamado de MSH *(message header segment)*, o conteúdo deste influência a estrutura de toda a mensagem [50]. Observando o exemplo anterior, dos campos seguintes, de acordo com documentação fornecida pelo site oficial [51, 50], podem ser destacados:

FIELD SEPARATOR: indica os caracteres utilizados para separar campos ao longo da mensagem, geralmente uma barra vertical ocupando a posição 1;

ENCODING CHARACTERS: indica os caractéres de separação como é possível observar no canto superior esquerdo após o segmento MSH.

SENDING APPLICATION: identifica qual a aplicação que faz envio da mensagem;

RECEIVING APPLICATION: identifica qual a aplicação que faz a receção da mensagem;

SENDING FACILITY: identifica qual a entidade que faz a envio da mensagem;

RECEIVING FACILITY: identifica qual a entidade que faz a receção da mensagem;

MESSAGE TYPE: contém o tipo de mensagem assim como o tipo de evento que conduziu ao envio da mesma. O sistema de receção utiliza este campo para saber quantos segmentos tem a mensagem e qual o conteúdo esperado da mesma. *General Acknowledgment Message* (ACK), *Admission Discharge Transfer* ADT, *Observ. result*, *Request Patient Information* (RQI) são alguns exemplos dos tipos de mensagens;

ACCEPT ACKNOWLEDGEMENT TYPE E ACK TYPE: indica as condições em que são requeridos os ack. As opções são:

- **AL**, indicando que pretende receber sempre ack;
- **NE**, indicando que nunca pretende receber ack;
- **ER**, só pretende receber ACK em caso erro ou rejeição;
- **SU**, só pretende ser avisado em caso de sucesso.

Esta versão dos *standards* para partilha de informação no formato de mensagens, tem permitido a interoperabilidade entre sistemas de informação de unidades de saúde, registos eletrónicos de pacientes de sistemas de informação de laboratórios, farmácias entre outros, sendo atualmente adotado por grande parte das instituições de saúde em Portugal.

2.5 STANDARD DE REPRESENTAÇÃO DE INFORMAÇÃO CLÍNICA ICD9-CM

International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM) é uma adaptação por parte do Departamento de Saúde do governo dos Estados Unidos da América

aos *standards* internacionais ICD-9-CMpropostos pela Organização Mundial de Saúde para propósitos de mortalidade [52, 53].

A definição proposta pela Administração Central so Sistema de Saúde (IP) é, "ICD-9-CM é o conjunto de códigos de diagnósticos e de procedimentos utilizados para a classificação e codificação da informação de morbilidade e mortalidade para fins estatísticos e para indexação dos registos hospitalares por doença e intervenções cirúrgicas, para armazenamento e pesquisa.

Baseia-se na International Statistical Classification of Diseases, Injuries and Causes of Death, a 9ª Revisão da ICD da Organização Mundial de Saúde, que data de 1975, e da qual difere essencialmente pela introdução de subclassificações de 5 dígitos nas subcategorias (de 4 dígitos)." [54].

ICD-9-CM é alvo de desenvolvimentos constantes por parte de quatro entidades cooperantes sempre sob alçada da *Organização Mundial de Saúde (OMS)*. *AHA (American Hospital Association)* responsável pela divulgação das regras de codificação clinica, *AHIMA (American Health Information Management Association)* entidade de relevo na área da certificação da codificação nos Estado Unidos da America, *Centers for Medicare & Medicaid Services* responsável pela gestão de softwares Medicaid e Medicare e por fim *NCHS (National Center for Health Statistics)* responsável primário pelo desenvolvimento da classificação dos diagnósticos e ainda pela divulgação dos documentos *ICD-9-CM Official Guidelines for Coding and Reporting* [55].

A normalização utilizando códigos melhora a consistência e entendimento entre médicos, assim como, a eficiência do tratamento médico e da pesquisa cientifica. ICD-9-CM contem uma lista de códigos que correspondem a diagnósticos e procedimentos retirados de unidades de saúde. Estes códigos podem posteriormente ser utilizados no processo clínico eletrónico ou na codificação clínica para medição do desempenho hospitalar através de Grupos de Diagnóstico Homogéneos [54].

Este sistema de classificação foi originalmente proposto por Robert B. Fetter da Universidade de Yale no final da década de 60 como sistema de revisão de utilização, tendo como variável dependente o tempo de internamento [56]. O sucesso foi quase imediato, sendo desde 1975 aplicado a sistemas de classificação de financiamento dos hospitais. Só no ano de 1979 a Universidade de Yale reformulou os GDH, utilizando então os códigos ICD-9-CM [56]. Desde 1983 é a base do financiamento do Medicare norte americano para doentes internados utilizando o *Acute Inpatient Prospective Payment System* [57]. O grande objetivo destas alterações visaram obter um sistema de classificação de doentes em classes clinicamente coerentes e homogéneas em termos de consumo de recursos [56].

Os diagnósticos ICD-9-CM inicialmente divididos em 23 grandes categorias (GCD) foram posteriormente alargados para albergar a Infecção pelo HIV e Traumatismos Múltiplos Significativos. Estes 25 grupos foram então distribuídos por duas categorias GDH Cirúrgicos

e GDH Médicos utilizando como elementos diferenciadores as comorbilidades, idade e destino após alta [56].

De acordo com a Administração Central do Sistema de Saúde em Portugal os GDH são um sistema de classificação dedicado a doentes internados em hospitais de agudos com a missão de os agrupar em algomerados clinicamente coerentes e similares tendo em conta o consumo de recursos, permitindo assim o relacionamento entre o tipo de doente tratado e o recurso consumido [58].

Em Portugal o início do projeto GDH data de 1984 com diversos estudos de avaliação, que conduziram mais tarde,1989, à adoção do ICD-9-CM, formação de médicos codificadores, criação de um resumo de alta normalizado e aplicação informática para agrupamento de doentes e produção de relatórios de gestão(LDRG) [54, 56]. No ano de 2006 o sistema já estava na versão 21 contendo 672 grupos AP-DRG sendo também 50% dos cálculos dos orçamentos dos hospitais realizado segundo produção de internamento medida em GDH [56].

A Circular Normativa nº1/89 do Gabinete Sr. Secretário de Estado da Saúde, atesta a classificação de doentes em GDH é obrigatória sendo as primeiras tabelas utilizadas no SNS de 31 de Maio desse mesmo ano [54]. Como qualquer sistema, existe um número minímo de recolha de dados, Conjunto Mínimo de Dados (CMD), dos quais se destacam [57]:

- Sexo;
- Idade;
- Destino após alta;
- Diagnóstico Principal;
- Procedimentos;
- Causas Externas;
- Morfologia tumoral.

Da lista anterior é importante destacar o diagnóstico, as causas externas, a morfologia tumoral e os procedimentos, visto serem aqueles que vão ser codificados pelo ICD-9-CM possibilitando o seu reconhecimento em qualquer parte do globo. Naturalmente o grande objetivo de qualquer sistema que se pretenda implementar nesta área, é agrupar o episódio numa de 25 Grandes Categorias Diagnósticas (GCD) [59]. Por exemplo GDC1 representa Doenças e Perturbações do Sistemas Nervoso, GDC15 recém-nascidos e lactentes com afeções do período perinatal e GCD25 representa Traumatismos Múltiplos Significativos.

Um dos pontos que marca a diferença deste sistema para os demais é o facto de ter sido pioneiro a permitir a mediação e definição do índice *casemix* de um hospital utilizado

pela gestão clinica como suporte ao financiamento hospitalar [54]. De acordo com CIDES, *Ciências da Informação e da Decisão em Saúde* o *casemix* é, "Coeficiente global de ponderação da produção que reflete a relatividade de um hospital face aos outros, em termos da sua maior ou menor proporção de doentes com patologias complexas e, consequentemente, mais consumidoras de recursos. O ICM determina-se calculando o rácio entre o número de doentes equivalentes ponderados pelos pesos relativos dos respetivos GDH e o número total de doentes equivalentes"

ICM de cada hospital é então calculado pela ACSS com base nos dados de episódios de internamento e de ambulatório de cada hospital classificados segundo GDH.

É de vital importância para uma unidade de saúde um sistema de classificação que funcione corretamente, de fácil interação sem erros que neste podem custar verbas à unidade hospitalar.

2.6 INDÚSTRIA 4.0

Não há duvida que a indústria alemã é uma das mais competitivas a nível mundial, apresentando uma magnífica capacidade de produção em sistema distribuído, com produtos finais sólidos e robustos (produção automóvel por exemplo). Com a evolução da tecnologia surgiram novas tendências e o governo alemão não perdeu tempo e redigiu um documento para auxiliar as suas empresas na implementação de modelos que favoreçam o aumento de produtividade e redução de custos. Chamou a essas novas ideias de revolução industrial 4.0.

2.6.1 *Enquadramento Teórico*

A indústria é uma atividade económica que surgiu com a primeira revolução industrial no final do século XVIII e início século XIX, tendo por objetivo a transformação de matéria-prima em produtos comercializáveis, utilizando para tal, força humana, maquinaria e energia. As diversas revoluções industriais consideradas até ao momento, pautam por mudanças repentinas dos conceitos tecnológicos que conduzem a rápidas mudanças de paradigma. Desde a introdução de maquinaria pesada na primeira revolução industrial até à disseminação generalizada de modelos computacionais muitos anos passaram. Agora, aproveitando rápidos avanços em áreas como *Internet of Things*, interoperabilidade, sistemas embebidos e outros conceitos, uma nova mudança de paradigma está em andamento [60].

Mackinsey define esta nova era da indústria como: *" Industry 4.0 as digitization of the manufacturing sector, with embedded sensors in virtually all product components and manufacturing equipment, ubiquitous cyber-physical systems, and analysis of all relevant data. It is driven by four clusters of disruptive technologies."* [61]. O primeiro ponto a destacar é a introdução de sis-

temas embebidos na forma de sensores ou *software* capaz de desempenhar papel sensorial. O segundo não poderia deixar de ser um sistema hibrido entre físico e virtual chamado de *cyber-physical system*, por último a referência aos 4 *clusters* que segundo o autor impulsionam esta nova corrente de pensamento [61]:

1. Poder computacional e interoperabilidade entre SI . Surge neste ponto os conceitos de *big data*, sensores e sistemas embebidos impulsionados por avanços na *internet-of-things* em conjunto com *cloud technology*;

2. Poder analítico e inteligência artificial. Avanços significativos na AI e *machine learning* resultam em novas ferramentas com elevado poder comercial, principalmente ao nível das ferramentas de apoio à decisão;

3. A interação homem-máquina é um ponto de vital importância, sendo disso exemplo as implementações de *interfaces touch*, realidade aumentada, como por exemplo o *Google Glass* da Google;

4. Por último deve existir uma ponte entre mundo digital utilizando impressoras 3D ou robôs de última geração. Neste ponto os sensores fazem o sentido inverso mapeando para digital aquilo que é físico.

Na figura 4 pode ser observada a forma como este conceito encaixa no ceio de uma indústria e de que forma pode ser implementada desde uma linha de montagem até à gestão internacional da organização. No rodapé da figura podem ser observadas as diversas empresas que contribuem e utilizam soluções relacionadas com a indústria 4.0.

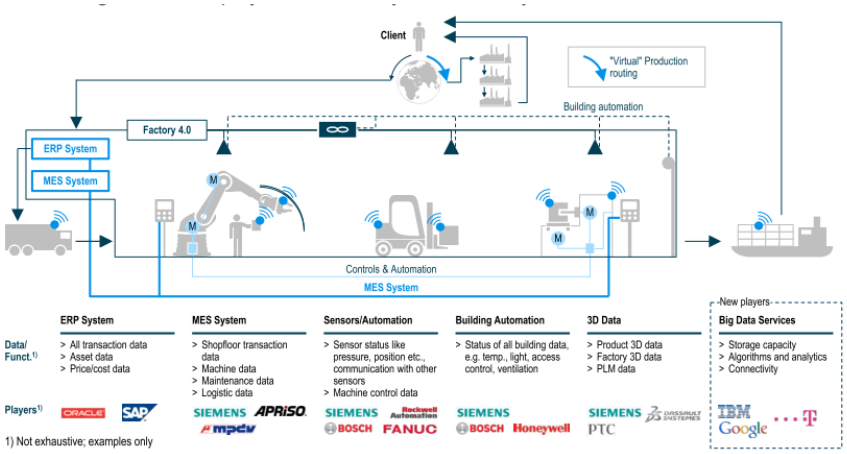


Figura 4.: Posicionamento dos diferentes intervenientes dentro da indústria 4.0 (adaptado de [62])

Leonard Sussenbach, chefe do departamento *IoT Strategy* da CISCO atesta sobre o poder da *Internet of Things* [61]:

" *Driven by the Internet of Everything, Industry 4.0 offers high potential impact at a relatively low*

cost. Every company and every industry will become a digital company resembling a technology stack – software apps enable agile processes and new revenue streams, as they turn data from connected "things" on the plant floor into real business outcomes. Investment is required in the right technologies and platforms. On the infrastructure side, the IT-centric centralized cloud model is not always complete – technical infrastructure must be tailored to work in tandem with the hyper-distributed Edge, millions of "things," running apps locally, self-learning, and in real time (e.g., the cloud might not be the best option for low-latency OT applications). " Na primeira parte da citação é referido o poder do *software* para a criação processos ágeis e de fácil adaptação a novos contextos. Fazendo posterior referência ao investimento necessário neste tipo de implementações. A adaptação das estruturas da fábrica a esta nova abordagem motiva investimentos avultados em termos monetários, recursos humanos e materiais sendo assim motivo de possíveis atrasos no avanço desta ideologia.

Cyber-physical System (CPS) toma partido de tecnologias inovadores que conectam dispositivos físicos e capacidades computacionais [63]. Ao comunicar num formato rede social uma máquina pode então revelar o seu estado e possíveis falhas ou impactos se algum parâmetro for ajustado. *5-level CPS*, também conhecida por arquitetura 5C compreende todos os níveis desde a aquisição inicial de dados até apresentação final de informação [63]. Os 5 Níveis podem então ser apresentados na forma de lista [63]:

1. ***Smart Connection:*** recolher dados fidedignos de máquinas ou sistemas informáticos é primeiro passo para uma implementação de sucesso;

2. ***Data-to-information conversion:*** nesta fase os dados devem ser trabalhados de forma a obter informação;

3. ***Cyber:*** funciona como *hub* central nesta arquitetura. Todas máquinas enviam informação para um ponto central, aumentando assim a capacidade de reunir informação extra. O histórico de máquinas pode então ser recolhido e comparado com outras dentro da mesma fábrica;

4. ***Cognition:*** a apresentação da informação ao utilizador final é ponto vital de qualquer sistema CPS, formando assim sistemas de apoio à decisão;

5. ***Configuration:*** representa o *feedback* do nível *cyber* para o espaço físico atuando como sistema de auto controlo.

De acordo com Jaap Bloem existe um benefício triplo na implementação de sistemas interligados [64]. No primeiro lugar a comunicação máquina com máquina permite uma redução de pessoal e ainda importantes contribuições para que a eficiência e segurança possam ser alcançados [64]. A introdução do conceito de manutenção preditiva com base nos relatos em tempo real do que está a acontecer e por último, informação detalhada

e permanente pode ser enviada ao cliente sobre o estado da sua encomenda e reportar atrasos caso seja necessário [64].

2.6.2 *Adaptação ao Contexto Hospitalar*

O setor da saúde representa um filão por explorar no que diz respeito à implementação de conceitos relacionados com a indústria 4.0. No artigo de opinião *Three ways the Fourth Industrial Revolution can transform healthcare*, o CEO da Novartis Joseph Jimenez indica 3 transformações que estas ideias podem trazer ao setor da saúde [65, 66]:

1. ***Incorporar a gestão de doenças no dia-a-dia***, a gestão e acompanhamento de um quadro clinico hoje em dia é muito mais que tratamento na unidade de saúde. Atualmente 80% das mortes por *Non-communicable disease (NCD)* devem-se a doenças crónicas do foro cardiovascular ou respiratório motivado em parte pela falta de empenho do paciente no tratamento. Assim num futuro não muito distante com a introdução de dispositivos ligados em network através da *IoT* algumas destas situações podem ser solucionadas. Por outro lado, a gestão de uma organização vai beneficiar diretamente de novas soluções provenientes desta áreas (CPS);

2. ***Cuidados especiais com população envelhecida***, os custos com a população envelhecida na Europa são uma preocupação constante das entidades governativas. Custa 3 a 5 vezes mais oferecer cuidados de saúde a estas faixas etárias do que mais jovens. Neste campo a robótica e inteligência artificial podem desempenhar papéis interessantes na criação de sistemas de acompanhamento e até prestação de serviços a um custo relativamente baixo;

3. ***Tentar resolver os grandes desafios da ciência ligada à área médica***, como por exemplo a cura para o cancro ou cura para doenças degenerativas pode num futuro próximo acontecer em parte devido a utilização de *software* criado para o efeito. É possível atualmente sequenciar o genoma de um indivíduo em 24 horas coisa que até bem pouco tempo era impensável. Com isto existe maior quantidade de dados sobre genoma humano possibilitando medidas preventivas e tratamentos mais individualizados.

Abordado diretamente no ponto um e de uma forma indereta em dois, está o conceito de Telemedicina. Esta promove uma melhor relação do paciente com o clínico ainda que à distância. Este conceito é apenas vocacionado para doentes, também pode ser utilizado entre clínicos, por exemplo, durante um diagnóstico, o clínico em caso de dúvidas, pode chamar outro médico especialista à distância de um *click* [66]. Numa altura em que *smartphones* são uma presença constante na vida de todos, permite que os pacientes sejam mais independentes do que em tempos passados, utilizando o seu telemóvel para estarem permanentemente em contacto com prestadores de cuidados de saúde [67].

A Telemedicina faz parte de um conjunto de conceitos englobados no chamado *mHealth*. Este promete ser um bom campo a explorar abrindo horizontes utilizando tecnologias móveis para tal [68]. Por exemplo David D. Luxton e seus colaboradores apresenta em *mHealth for Mental Health: Integrating Smartphone Technology in Behavioral Healthcare* uma solução que utiliza *smartphones* para monitorizar pacientes à distância (*behavioral health care*) [69].

Embora as soluções propostas em *"eHealth: Industry 4.0 can serve as the model for digital healthcare"* apresentem enorme potencial, motivações económicas, sociais, políticas e culturais podem criar obstáculos à implementação destas soluções [66]. Nomeadamente a disseminação de informação é um ponto que sempre gerou controvérsia quando ao mesmo tempo países colaboram para tentar criar um registo clínico eletrónico a nível europeu.

IBM Watson Health faz parte do programa Watson desenvolvido pela IBM [70]. É um sistema pioneiro, estabelecendo uma parceria entre humanos e tecnologia com o objetivo de transformar a saúde a nível global. Dotado de capacidade cognitiva (aprendizagem, raciocínio e percepção) pode ajudar na percepção de características presente em dados recolhidos que até agora poderiam passar despercebidos [70]. De acordo com site oficial 80% dos dados são invisíveis aos sistemas atuais uma vez que não estão normalizados (não estruturados). Watson consegue ultrapassar isso utilizando *software* sofisticado com capacidade de analisar 40 milhões de documentos em apenas 15 segundos [70].

Alvo de grande mudanças nas últimas décadas, o sector da saúde vai continuar a ser alvo de profundas mudanças, motivado por questões demográficas e impulsionado por desenvolvimentos tecnológicos [71]. Surgiram então novos competidores com soluções inovadoras para conquistar um mercado lucrativo como o da saúde.

Medical Cyber Physical Systems

Como apresentado na secção anterior, o sector da saúde está a sofrer mudanças abruptas na sua configuração e na forma como trata os dados e a informação, suportadas por soluções inovadoras na área dos sistemas embebidos e *networks*. A mudança de um equipamento único a realizar monitorização e tratamento de um paciente, para múltiplos pontos de recolha de informação e intervenção, de acordo com esta, é uma realidade não muito distante [63]. A estes sistemas, em conformidade com o que acontece na indústria chamar-se-à *Medical Cyber-physical Systems*.

A utilização de modelos para criação destas tecnologias é imperativa, sendo necessárias técnicas de validação, verificação e *design* inovadoras. É também previsível que a criação de legislação para este tipo de sistemas seja uma tarefa árdua [63]. As mudanças que podem acontecer nesta área podem ser categorizadas como [72]:

- **Novas Abordagens de *Software***, utilizando novas tecnologias com base em sistemas embebidos. A cirurgia robótica é um exemplo da utilização de transmissão de informação em tempo real assim como utilização de imagens de alta resolução;

- **Aumento Conectividade de Dispositivos Médicos**, com o aumento das soluções de *software* existe o aumento das ligações entre dispositivos. Hoje em dia o network prende-se essencialmente com o registo clínico eletrónico e sistemas de monitorização de pacientes;
- **Sistemas de Controlo de Níveis Fisiológicos**, a maior parte dos cenários clínicos envolvem a presença de um ou mais especialistas. Uma cirurgia envolve a presença de um anestesista responsável por controlar a administração de sedativos ao longo desta. Um sistema de monitorização capaz de controlar níveis de fármaco e quando este baixa-se de um determinado valor, despoleta-se um sistema de alerta ao anestesista ou numa solução mais inovadora, ele próprio fizesse administração de fármacos, retirava peso no trabalho deste tipo de profissionais de saúde;
- **Monitorização Continua Cuidados Saúde**, por exemplo, acompanhamento através de Telemedicina de pacientes nas suas casas, reduzindo assim custos de permanência numa unidade hospitalar.

Enquanto na indústria 4.0, a modelização procura através de modelos *Auto-Cad* simular a unidade de produção de uma unidade fabril. No caso de um sistema médico hibrido pode ser utilizado *software* para simular crescimento de tecidos, excitação de células ou atividades de gestão e monitorização da unidade hospitalar [73, 74].

Ao longo da presente secção *Industry 4.0* foram abordados temas mais gerais e ainda uma tentativa de analogia com a área clínica. Existe no entanto, um tema pertinente que deve ser abordado em maior detalhe, uma vez que na presente secção está presente nas entrelinhas. Com as implementações apresentadas a tecnologia passa a ser ubíqua na vida do ser humano daí ser abordado o conceito de *ubiquitous and pervasive computing*.

2.7 PERVASIVE OR UBIQUITOUS COMPUTING

Evoluir os sistemas computacionais de forma a que estes estejam presentes na vida quotidiana de todas as pessoas é um desafio enorme. Nos últimos 40 anos mudanças na dimensão e capacidade de processamento de diversos componentes, tornaram estes capazes e ao mesmo tempo com preço suportável por quase toda a população. Esta secção abrange então a ubiquidade de sistemas computacionais e quais as suas ligações com área da saúde.

2.7.1 *Enquadramento Teórico*

O conceito de ubiquidade em sistemas computacionais pode ser descrito por diversos prismas. De acordo com W. Keith Edwards do Instituto de Tecnologia da Georgia (Estados Unidos da América), pode ser visto como o desaparecimento do computador [75]. A

utilização de tecnologia passará a ser tão banal como utilizar um lápis ou caneta, assistindo o ser humano nas interações básicas do dia a dia [75]. Noutra abordagem Kalle Lyytinen e Youngjin Yoo atestam no seu trabalho *Issues and Challanges in Ubiquitous Computing* que *" Ubiquitous computing will help organize and mediate social interactions wherever and whenever these situations might occur."* [76]. As duas definições são próximas e concordantes, ambas com base nos trabalhos de Weiser sobre avanços nas áreas das comunicações sem fios e capacidade computacional intitulado *The Computer for the 21st Century* [77].

Para alcançar o nível de transparência pedido estes devem ser livres de administração humana, com capacidade de entrar e sair da networksendo para tal imperativa a existência de interoperabilidade entre todos os intervenientes [75].

Os avanços nesta área dependem então da *mobile* e *pervasive computing*. O primeiro refere-se à capacidade de levar a tecnologia para qualquer ponto, como acontece no caso dos *smartphones* embora não seja necessário a recolha de dados provenientes do ambiente onde se insere [75]. *Pervasive Computing* implica que o sistema obtenha do meio uma percepção da realidade, construída tendo por base a informação recolhida por modelos computacionais adequados [76]. Se o meio onde está inserido for capaz do recíproco forma-se assim um sistema inteligente auto adaptativo.

O maior desafio deste tipo de sistemas é encontrar um ponto de equilíbrio entre sistemas de mobilidade em larga escala e o conceito de *pervasive computing* [76].

A área da saúde representa um enorme filão para este tipo de sistemas, a ubiquidade nos cuidados de saúde pode ser vista como um conjunto de sensores no paciente ou ambientais que recolhem informação com vista ao aumento da qualidade de saúde física e mental do mesmo [78].

Conjuntos de pequenos sensores podem ser então desenvolvidos com intuito de recolher informações fisiológicas diretamente do paciente, podendo ser automaticamente transmitidas em tempo real para o diagnóstico de patologias ou então em caso de agentes atuadores, libertar quantidades de fármacos que resolvem e controlam o estado do paciente [78]. Nesta visão o paciente poderia estar fora da unidade de saúde, aumentando o conforto do mesmo e reduzindo os custos hospitalares [78].

O trabalho realizado por Pei-Cheng Hii e Wan-Young Chung, *A Comprehensive Ubiquitous Healthcare Solution on an Android Mobile Service* apresenta um sistema com base no *Sistema Operativo (SO) Android* que serve de plataforma de monitorização para um conjunto de sensores que coleta dados relativos ao ECG em tempo real [79].

Um trabalho mais dedicado aos prestadores dos cuidados de saúde é proposto por Charlotte Tang e Sheelagh Carpendale da Universidade de Cagliari intitulado *Healthcare Quality and Information Flow during Shift Change*, onde demonstra as fases iniciais do desenvolvimento de sistema que apoia a partilha de informação entre enfermeiros, aquando a troca

de turnos [80]. Embora não abordem tecnologias e sistemas a implementar, fazem um levantamento de requisitos do que é necessário para realizar tal sistema [80].

Existem também aplicações relacionadas com a formação de profissionais de saúde utilizando sistemas ubíquos como em [81]. Ou então relacionados com a mobilidade de pacientes idosos nomeadamente cadeiras de rodas que percebem o ambiente onde estão inseridos e têm capacidade de se desviar de obstáculos pelo caminho.

2.8 APLICAÇÕES MÓVEIS

Uma aplicação móvel é uma *software application* desenhada para "correr" em dispositivos móveis tais como *smartphones* e *tablets*. Todos os dispositivos apresentam aplicações nativas e ainda a possibilidade de instalar outras utilizando os mercados de aplicações. Nos últimos 10 anos assistiu-se a um crescimento exponencial no mercado de aplicações móveis. Até ao de Junho do ano de 2016 existiam em todas as plataformas perto de 6 milhões de aplicações disponíveis para *download* [52].

De acordo com Anthony L. Wasserman, no trabalho *Software Engineering Issues for Mobile Application Development* concluiu [82]:

1. A maior parte das aplicações são de pequena dimensão em termos de linhas de código e complexidade;

2. Muitas das aplicações existentes utilizam o formato de aplicação web para obter informação;

3. Os programadores utilizam guias para melhores práticas, mas não utilizam processos formais de desenvolvimento;

4. Não existem métricas para o nível de desenvolvimento de aplicações.

As primeiras aplicações eram bastante simples como calendários ou agendas pessoais, mas as exigências por parte dos utilizadores levaram à criação de aplicativos cada vez mais complexos dificultando e criando desafios aos responsáveis pela criação destas.

Embora desenvolver aplicações móveis seja em tudo semelhante a desenvolver aplicações para computadores ou servidores, existem diferentes significativas que devem ser abordadas [83]:

- Interação com diferentes aplicações de diferentes origens;

- Interação com sensores como acelerómetro, toque no ecrã entre outros;

- Utilização de serviços com recurso à *Internet*;

- O programador deve desenvolver aplicações que corram em diferentes versões da mesma plataforma para ter o máximo possível de mercado;

- Com ligação permanente à *Internet* a segurança com informações pessoais é um ponto vital;

- O aspeto visual de qualquer aplicação e sua usabilidade motiva a utilização ou não por parte do utilizador;

- Embora emuladores desempenhem um ótimo papel, o teste de aplicações com recursos a *Internet* móvel é ainda complicado;

- O consumo de recursos assume importância quando inseridos em dispositivos onde leveza e dimensões são importantes.

Algumas das dificuldades de processamento podem ser ultrapassadas por uma nova tecnologia chamada *Mobile Cloud Computing*. Segundo Hoang T. Dinh e seus colaboradores, MCC é *"... at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MC to not just smartphone users but a much broader range of mobile subscribers"*. [83]. Assim sendo o processamento de informação deixa de ser algo desenvolvido em sistemas físicos passando para *clouds* com capacidade de abstração bastante superior.

Atualmente são desenvolvidas aplicações para áreas bastante diversas como biomédica, instalação e utilização de painéis foto-voltaicos ou até cálculos matemáticos com elevada complexidade [84]. Pode também ser enquadrada esta secção, como uma consequência óbvia da revolução industrial 4.0. De acordo com D. Mourtzis, no campo da indústria existiu um grande número de sistemas móveis a apresentar e permitir operações com suporte em *Multi Agent Systems (MAS)* [84]. Existem ainda relatos de aplicações vocacionadas específicamente para o cliente, com informações sobre encomendas ou até interação com unidade de produção diretamente [84].

Embora muito mais pode-se ser dito em relação a aplicações móveis, tal será efetuado nos casos de estudo apresentados nos próximos capítulos. É importante reter a importância atual do mercado de aplicações móveis e o quão difícil é desenvolver uma aplicação de sucesso. Existem diversas condicionantes que podem ditar o insucesso da mesma como usabilidade e aspeto visual. A área da saúde não é exceção, existindo nos últimos anos um aumento exponencial na quantidade de aplicações desenvolvidas para esta área.

2.8.1 *Aplicações Móveis Área da Saúde*

É obvio que o potencial dos dispositivos móveis para transformar a indústria da saúde é tremendo. Existem trabalhos na área da coleta de dados e processamento dos mesmos, na área da Telemedicina e ainda ajuda à prática clinica [85].

Segundo Arabind Kailas e seus colaboradores, existem atualmente registadas em lojas autorizadas pelas plataformas cerca de 7000 aplicações relacionadas com saúde [86].

O conceito de *mobile health* ou mHealth é definido de acordo com Ali Kemal Yetisen no trabalho *"The regulation of mobile medical applications"* como a utilização de comunicação via *wireless/GPRS* com vista a suportar e melhorar a eficiência da saúde pública e prática clínica [87]. As aplicações servem então dois propósitos distintos, servir como acessórios para dispositivos médicos devidamente regulados ou então são o *software* que transforma tal plataforma num dispositivo devidamente regulado [87].

Existem três categorias onde o *mHealth* pode desempenhar um papel interessante. Primeiro na promoção da saúde e medicina preventiva utilizando aplicações informativas que chamem o interesse do utilizador. Segundo servindo como dispositivos de diagnóstico portáteis. E por último, como ferramenta portátil de apoio à decisão, por parte dos médicos e enfermeiros, servindo também como ponto de recolha de informações preciosas [87]. Existem também relatos de implementações na gestão de bases de dados clinicas e treino de pessoal médico [87].

A principal característica deste tipo de aplicações reside talvez no facto de conseguir recolher informação em tempo-real e transmitir essa informação para sistemas de informação devidamente autorizados, utilizando *smartphones* cada vez mais sofisticados, capazes de utilizar sensores biométricos e câmera fotográfica para recolha de dados cada vez mais fiáveis [86].

Os maiores desafios quando se desenvolve aplicações móveis é a questão da autonomia na geração de alertas, tanto para o próprio utilizador como para autoridades em casos de emergência [86]. A título de exemplo a aplicação *eCAALYX Mobile Application* desenvolvida pela *eCAALYX* e co-financiada pela União Europeia, apresenta uma implementação onde é construído um sistema de monitorização destinado a pessoas idosas que apresentem doenças crónicas [85]. A aplicação móvel funciona como um intermediário entre a recolha de dados e a aplicação web onde os profissionais de saúde fazem monitorização assim como reagem a situações consideradas anormais.

Numa abordagem diferente, em *"IEEE 802-15-4 Wireless Mobile Application for Healthcare System"*, Chiew-Lian Yau e Wan-Young Chung desenvolveram um sistema que comunica utilizando rede *wireless IEEE 802.15.4* [88]. Objetivo é reduzir o tempo resposta em caso de doença, eliminar deslocações desnecessárias ao hospital, eliminar registo em papel e reduzir erro médico ao mínimo. Na figura 5 é possível observar os componentes e estrutura do sis-

tema desenvolvido. Desde a coleta de dados no paciente (esquerda da figura 5) que utiliza o telemóvel como meio para enviar os dados para unidades centrais onde posteriormente é feito o tratamento e apresentação de informação (direita da figura 5).

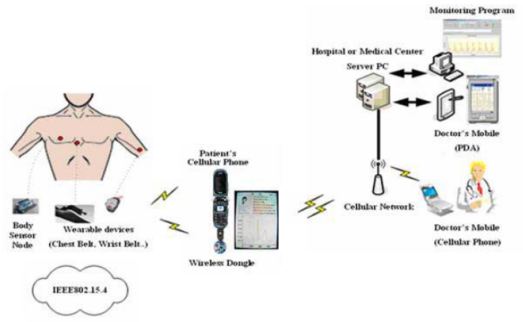


Figura 5.: Arquitetura do sistema IEEE802.15.4 (adaptado de [88]).

Num trabalho proposto por Ching-Ju Chiu e colegas, intitulado *"The attitudes, impact, and learning needs of older adults usinf apps on touchscreen mobile devices: Results from a pilot study"*, é apresentada um estudo dos efeitos da utilização de dispositivos móveis em pessoas de idade avançada [89]. No final dos exercícios estes apresentavam menores índices de stress do que aqueles que são considerados normais.

Desde o estudo comportamental até aquisição de dados de ECG, existe então uma grande diversidade de trabalhos nesta área. Por outro lado, a maior parte das aplicações relacionadas com unidades de saúde incidem sobre informações básicas como marcação de apontamentos, localização da unidade de saúde e contactos. Embora sejam implementações consideradas simples aumentam a comodidade e conforto dos utilizadores.

Em todas as aplicações existe sempre a probabilidade de erro, neste caso pode conduzir a um mau diagnóstico ou informações erradas a serem transmitidas. Nos Estados Unidos da América a *United States Food and Drug Association (FDA)* é responsável pela supervisão de aplicações relacionadas com área clinica, assinalando aquelas que podem conduzir a erros [87]. Em 2013 a FDA publicou o documento *"Mobile Medical Applications Guidance for Industry and Food and Drug Adminstration Staff"* onde entre outras indicações, marca posição entre aplicação não regulada e uma aplicação móvel para área da saúde [87]. Assim sendo uma aplicação só é ilegível para regulamentação se transformar o dispositivo móvel num dispositivo médico. A abordagem da FDA para a regulamentação envolve diretrizes,regras e proteção para académicos, engenheiros de *software*, clínicos e pacientes [87].

Ao longo desta secção foram abordados aspetos relacionados com aplicações móveis desde a sua conceção até questões que podem influenciar a sua utilização. Alguns pontos serão abordados junto do caso de estudo onde é implementada uma solução de *mHealth*.

2.9 CONCLUSÃO

O presente capítulo enquadrou o trabalho desenvolvido ao longo do projeto de acordo com aquilo que a comunidade cientifica tem desenvolvido ao longo dos últimos anos. Foram abordados temas bastante díspares que nem sempre apresentam relação direta com a saúde. Quando se pretende desenvolver aplicações relacionadas com *Sistemas de Informação Hospitalar (SIH)* existe naturalmente uma forte ligação a sistemas de informação, uma vez que estão na base de qualquer implementação. Se no início não se pensava em comunicação entre sistemas de informação com introdução das TIC tudo mudou, sendo o principal ponto de ordem nas implementações que se faz atualmente. A interoperabilidade ocupa assim um papel preponderante ao longo do capítulo. Através da literatura conclui-se que dois dos aspetos mais importantes relacionados com interoperabilidade são a versão 2 para troca de mensagens em formato HL7 e os códigos ICD-9.

Existem no entanto informações algo díspares sobre a implementação de sistemas de processamento de mensagens HL7 uma vez que cada unidade de saúde deve procurar a estratégia e o sistema mais adequado às suas necessidades. Pela dificuldade de meios existentes em muitas unidades de saúde a plataforma AIDA surge como uma alternativa viável para difundir e agregar dados que posteriormente processados conduzem a informações importantes.

Os sistemas de monitorização são uma consequência positiva da implementação de diretrizes da indústria 4.0 ou nova revolução industrial. Nem sempre nas unidades de saúde, esse conceito é aplicado à performance de agentes ou maquinas. A quase totalidade dos trabalhos encontrados estão relacionados com monitorização de pacientes em unidades de saúde ou através da Telemedicina.

Como referido ao longo do capítulo os sistemas de informação não são destinados apenas aos clínicos, também existem aplicações eHealth destinadas ao bem estar e comodidade do paciente, nomeadamente na categoria *mHealth*. Como um dos pontos importantes deste último é a presença em todo o instante na vida do paciente falou-se de *ubiquitous and pervasive computing* tentando encontrar pontes com área médica. Por fim também relacionado com *mHealth* abordou-se a temática das aplicações móveis fazendo o estado de arte da aplicação destas no contexto da saúde.

Assim sendo ao longo do capítulo foi enquadrado todo o trabalho desenvolvido na presente dissertação em concordância com estado atual da literatura.

3

METODOLOGIAS DE INVESTIGAÇÃO E FERRAMENTAS DE DESENVOLVIMENTO

As questões levantadas no capítulo 1, assim como possíveis soluções encontradas necessitam sempre de metodologias que ajudem a passar do papel para algo sólido e concreto.

Cada caso de estudo apresenta uma metodologia de desenvolvimento própria no entanto, optou-se por manter este capítulo de forma a evidenciar aqueles materiais que são de alguma forma transversais a todos os casos de estudo assim como metodologia que serve de apoio a toda a implementação.

3.1 INTRODUÇÃO

O presente capítulo pretende apresentar as ferramentas que de forma direta ou indireta influenciam todas as soluções encontradas para as questões levantadas no inicio do desenvolvimento deste projeto.

Na primeira secção está contemplado o método de resolução de problemas adotado para diversas soluções. Segue-se a linguagem Python, as ferramentas Oracle indispensáveis num qualquer projeto que envolva processamento de informação. Em seguida uma breve alusão aos *Web Services*, nomeadamente do tipo *Restful* utilizando a linguagem *Hypertext Preprocessor (PHP)*. Nas duas últimas secções do presente capítulo são apresentadas as ferramentas e *frameworks* utilizadas no desenvolvimento de aplicações web nomeadamente (*Angularjs e Javascript*) e por último as *frameworks* utilizadas como base das aplicações móveis (*Android SDK*).

3.2 DESIGN RESEARCH

A metodologia de investigação *Design Research*, é considerada fundamental para a concessão e desenvolvimento de produtos, serviços e sistemas capazes de satisfazer as necessidades existentes à partida [90]. Fazendo o balanço entre os objetivos desta dissertação

e aspetos teóricos esta metodologia é aquela que melhor se enquadra em detrimento de outras como *action research*.

O processo de *Design Research* é uma sequência de atividades que resultam num produto ou serviço que até então não existia. Embora exista modelo considerado geral (ver figura 6) não é consensual visto que cada cenário apresenta especificidades muito próprias que devem ser consideras [91]. O modelo geral apresenta as seguintes etapas :

Consciencialização das Necessidades: perceber qual é de facto o problema e qual ou quais as possíveis consequências diretas ou indiretas (formalização do problema e definição de fronteiras);

Sugestão de Desenvolvimento: com base nos problemas levantados anteriormente, devem ser procuradas diversas soluções e de entre essas fazer uma seleção das melhores para serem de facto implementadas;

Desenvolvimento: construir a ou as possíveis soluções para o problema levantado inicialmente. Neste fase diferentes caminhos podem ser seguidos, desde elaboração de protótipos, representações gráficas, algoritmos desde que no final exista um artefacto funcional para apresentar. Embora no final possa ser desenvolvido de facto um produto, o objetivo da metodologia é mais abrangente. Gerar conhecimento que possa ser utilizado na resolução de problemas futuros, melhorar sistemas existentes e provocar novos problemas a ser resolvidos;

Avaliação: é o processo pelo qual existe uma criteriosa averiguação do comportamento do artefacto funcional no ambiente para qual foi projetado. Testes "in loco" que necessitam de ser definidos antes para saber qual ou quais os resultados que devem ser esperados;

Conclusão: pode ser o culminar do ciclo ou se a avaliação não correr como previsto começar de um novo com um problema que pode ser o mesmo ou o resultado da etapa anterior.

Entre a lista anterior e a figura 6, existem algumas discrepâncias. Nomeadamente a figura apresenta uma etapa intermédia entre sugestões de desenvolvimento e desenvolvimento propriamente dito. A **seleção da abordagem** é ponto importante que pode em alguns casos está incluído na sugestão de desenvolvimento.

De acordo com Vijay Vaishnavi, no trabalho *"Design Science Research in Information Systems"*, na última década, surgiu uma adaptação do conceito de *Design Research* ao campo dos *Sistemas de Informação (SI)*, denominado *Design Science Research* [93]. O autor atesta que a principal diferença entre ambas reside, *"The distinction frequently expressed is that DR is research "into" or "about" design whereas DSR is primarily research "using design as a research method" or technique."* [93]. Ou seja, motivado pelo crescente peso da pesquisa cientifica, o

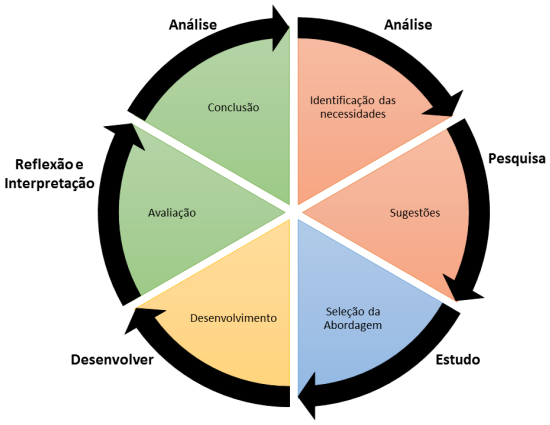


Figura 6.: Diversas etapas do processo *Design Research* (adaptado de [92]).

foco deixou de ser na pesquisa em prol do design ou da solução, passando esta a fazer ela própria parte da etapa de pesquisa. Relativamente ao ciclo propriamente dito, sofre ligeiras alterações nomeadamente a possibilidade de dar inicio a um novo ciclo sem concluir o anterior, numa abordagem *"learning through building"* [93].

Ao longo da apresentação dos casos de estudo é feita a referencia qual é a etapa do ciclo *Design Research* correspondente, por exemplo no capítulo [**?**], a secção da metodologia começa com a apresentação dos possíveis problemas existentes que carecem de solução, contemplando assim a fase de **consciencialização das necessidades**. Nesse caso de estudo, a principal necessidade era a implementação de uma interface HL7, que sustenta-se a interoperabilidade sintática entre aplicações. Posteriormente é demonstrado o processo de escolha das soluções a implementar (**Sugestões de Desenvolvimento**) com vista à resolução dessa necessidade. A secção relativa aos materiais utilizados e novamente a metodologia contribuem para a fase três do ciclo (**desenvolvimento**). Por último, as fases de **avaliação** e **conclusão** ficam a cargo do capítulo 25, onde são abordadas diversas provas de conceito realizadas e resultados e discussão, respetivamente.

3.3 Linguagem Programação Python v.3.4.3

A linguagem de programação Python nasceu no final da década de 80, sendo a primeira implementação datada de 1989 pelas mãos de Guido van Rossum no Instituto Nacional de Pesquisa da Matemática e Ciências Computacionais na Holanda [94]. Deriva de diversas linguagens como ABC, C, C++, Algol-68, entre outras linguagens de script. Pertence então ao grupo das linguagens ditas de alto nível, bastante interativa e interpretativa naturalmente orientada a objetos [95]. Utiliza naturalmente termos na língua inglesa como os demais exemplos supra mencionados, embora prime pela utilização de menos construções sintáticas [96].

No ano de 2000 foi lançada a versão 2 da linguagem, incluindo assim a capacidade de realizar ciclos, especificações adicionais para controlo de memória e ainda suporte para *UNICODE* [94]. No ano 2008 foi lançada a versão 3.0 com alterações evidentes ao nível dos ciclos (*for* por exemplo) e introdução de parêntesis em alguns casos, devido ao facto de *Unicode* ser agora uma verdadeira *string* [97] . Mais alterações são o suporte correto de SSL que até então não acontecia ou uma *asyncio framework* passou a fazer parte do código base da linguagem [98]. Estes são apenas alguns exemplos dos melhoramentos que ajudam a linguagem a ser considerada por muitos como a melhor para uma rápida e fácil produção de conteúdos.

As principais características desta linguagem podem ser compiladas na seguinte lista [96, 95, 99]:

FÁCIL APRENDIZAGEM: pela sua estrutura simples, com poucas *Keywords* a sintaxe é muito simples de interiorizar;

FÁCIL LEITURA E MANUTENÇÃO: código é mais claro que as demais linguagens , em grande parte devido à indentação. O código base é bastante simples, o que permite uma rápida manutenção e ainda instalação de *plugins*;

MODO INTERATIVO: este modo permite um fácil *debug* de *scripts*;

PORTABILIDADE: consegue correr numa grande variedade de plataformas mantendo uma *interface* semelhante;

INTERFACE COM BASE DE DADOS: a linguagem apresenta conectores para as mais utilizadas bases de dados como por exemplo *Oracle e MySql*;

GUI PROGRAMMING a linguagem suporta aplicações GUI criadas e transportadas para utilizações no formato *system call*.

A integração com linguagens como C ou Java permite um leque muito mais abrangente de soluções ao programador tornando o Python muito apetecível para todo o tipo de soluções. Após duas décadas de evolução a linguagem atingiu um patamar de estabilidade, deixando os seus utilizadores de serem especializados, sendo agora utilizada em desenvolvimento Web e ainda por um patamar de programadores mais relutantes como engenheiros ou cientistas.

Python tem sido nós últimos tempos aplicada à *Network Programming*, daí que seja fundamental abordar dois conceitos. O primeiro relativo ao protocolo que deve ser implementado em qualquer sistema com base na *Internet*. E o aproveitamento de código já existente, na forma de livrarias externas, que podem ser incorporadas no código base da aplicação [98]. Por vezes a solução pretendida já foi desenvolvida em forma de *plugin* que funcionam como uma extensão ao código fonte original.

3.4 BASES DE DADOS

Uma base de dados é nada mais nada menos do que um conjunto de informação que se relaciona. Por exemplo uma lista telefónica é uma base dados mas após breve análise problemas óbvios podem ser detetados. Encontrar uma pessoa é bastante difícil, encontrar pessoas que vivam numa determinada é possível mas muito complicado visto que indexação é feita por nome. Pelo facto de ser impressa é pouco suscetível a mudanças no ambiente em redor. Estas desvantagens são aplicáveis às unidades de saúde que ainda utilizem formato papel.

Algumas das primeiras aplicações desenvolvidas foram então destinadas a criar *database systems*, capazes de armazenar dados em formato digital e posteriormente retorna-los de forma rápida e eficaz. Embora as primeiras soluções sejam uma melhoria em relação ao tradicional papel estão a anos luz do que novas implementações conseguem fazer [100, 101].

Nas primeiras implementações a informação era armazenada e "representada" aos utilizadores de diversas formas. Utilizando *hierarchical database system* com base em estruturas em forma de árvore, *network database system* onde existem *links* entre dados e por fim aquele que hoje em dia é mais utilizador *relational model*.

A base de dados utilizando modelo relacional foi primeiro introduzida pelo *Dr. Edgar F. Codd* numa publicação financiada pela IBM com o nome "System R4 Relational" corria o ano de 1970 [100, 101]. Basicamente o modelo propõem que os dados sejam representados com conjuntos de tabelas ligados através de dados redundantes que representam assim elos de ligação ("relações") entre tabelas bi-dimensionais (constituídas por linhas e colunas).

As colunas em diferentes tabelas relacionam-se através de chaves (*keys*) estabelecidas previamente [101]. Uma chave ou *primary key*, identifica inequivocamente uma linha sendo a base de dados responsável por manter a sua integridade [101]. Existem então alguns termos que vale a pena registar, entidade é algo que desperta interesse ao utilizador da base de dados por exemplo um paciente. Uma coluna é uma parte da informação armazenada na tabela que em conjunto com outras forma uma linha descrevendo assim uma entidade ou uma acção por essa realizada. A tabela é então conjunto de acções que pode ser permanente (persistente) ou não, se for resultado de uma QUERY em *Structured Query Language (SQL)* por exemplo. Chave estrangeira é o nome atribuido a um atributo que tem valor de chave primária noutra tabela do modelo relacional, permitindo diretamente a relação entre tabelas [100, 101].

As tabelas e seus constituintes fazem parte de estruturas mais abrangentes, *Data Marts* (DM) e *Data Warehouses* (DW). Este último refere-se a um armazém de dados desenhado e implementado com vista a guardar dados sobre as diversas atividades de uma organização, com objetivo de disponibilizar a informação para uma rápida consulta e ainda possibilitar a utilização de ferramentas de *Business Intelligence (BI)* [102]. Geralmente estas estruturas

são criadas como resultado de processos de *Extract, Load, Transform (ETL)* que atua sobre múltiplas fontes de informação dentro da organização. Por seu turno, os DM são estruturas com uma extensão bastante mais reduzida, embora apresentem as mesmas propriedades do caso anterior, geralmente apresentam informações sobre uma atividade ou uma secção no seio de uma organização [102]. Embora os DM sejam parte de um DW, podem criar ilhas de informação se a sua relação com os demais não for corretamente projetada.

Na mesma altura em que apresentou o modelo relacional, Codd apresentou uma linguagem chamada *DSL/Alpha* para manipular dados em tabelas relacionadas [101]. Após alguns refinamentos e mudanças de nome por parte da IBM, a linguagem SQL foi formalmente apresentada [101]. Está dividida em diversas partes distintas, *SQL schema statements* e *SQL data statements* utilizados na criação e manipulação de estruturas numa base de dados [101]. Um exemplo do primeiro caso pode ser visto em 3.4 adaptado de [101].

```
CREATE TABLE corporation
(corp_id SMALLINT,
name VARCHAR(30),
CONSTRAINT pk_corporation PRIMARY KEY (corp_id) );
```

Como é possível observar (exemplo anterior), a linguagem é bastante simples e muito intuitiva sem a linguagem não *nonprocedural*, ou seja não existe declaração de variáveis ou ciclos por exemplo. Para selecionar informação de uma base de dados uma estrutura fixa deve ser utilizada, começando sempre pela palavra *"select"* [101]. A operação de selecionar dados dá pelo nome de *querie*, sendo o resultado chamado de *result set* apresentado na forma de tabela que pode ser exportada e gravada em outras tabelas.

Com SQL não é possível escrever aplicações completas como acontece com Python ou Java. Para tal é necessário integrar a linguagem noutra linguagem como caso do *Procedural Language-Structured Query Language (PLSQL)*. *"Procedural Language extensions to the Structured Query Language"*foi desenvolvida pela Oracle em meados dos anos 90 para aumentar a capacidade da linguagem SQL [103]. É uma linguagem com elevada estrutura repleta de *keywords* que são bastante explicativos daquilo que realmente fazem, por outro lado segue um pouco as directrizes de Java quanto à portabilidade.

O facto de não necessitar de intermediários para implementar SQL como sucede com outras linguagens (Java ou Python), torna PLSQL muito atrativa. Java necessita de JDBC, PHP do OCI e Python de cx_Oracle[103]. Neste caso o comando insert ou delete pode ser inserido no meio de um procedimento ou trigger sem qualquer tipo de cuidados.

A primeira base de dados que irá ser abordada é a *Oracle Database* seguindo-se a *MySql* sendo que ambas fazem parte da Oracle Corporation desde 2008.

3.4.1 *Oracle D.B.*

Base de dados *Oracle* ou *Oracle RDBMS* é sistema de gestão de base de dados disponibilizado pela *Oracle Corporation* que desde a sua génese em 1977 por Larry Ellison, Ed Oates e Bob Miner utiliza o modelo relacional marcando então a diferença para as demais soluções, tem alcançado um sucesso tremendo em todo mundo [100]

Basicamente uma base de dados Oracle é uma coleção de informação processada como uma unidade, com objectivo de guardar e quando oportuno devolver informação sobre essa unidade. Existe então um servidor que rejeita acessos sem permissão, responde afirmativamente em caso contrário e ainda assim consiga recuperar em caso de erro ou falha [104]. Cada instância Oracle utiliza um SGA *(System Global Area)*, por forma a desenrolar a partilha de memória permitindo assim guardar dados e controlar toda a informação [104]. Para tal toma partido de um ou vários *data files*, que são ficheiros físicos, onde está alocada toda a informação da base dados como tabelas e indices. Os *tablespaces* são então conhecidos como o aglomerar de *data files*.

Quando existe necessidade de aceder a informação, determinado ficheiro é carregado, permanecendo posteriormente em *cache* por um determinado espaço temporal. É assim reduzido ao mínimo o número de acessos a disco a informação é armazenada apenas uma vez no final de determinada operação.

A Oracle disponibiliza também um IDE, possibilitando a execução de tarefas e acesso a informações de modo gráfico e com poucos cliques [105]. Com o objetivo minimizar o tempo de trabalho necessário e ainda maximizar o retorno proveniente do investimento despendido. A principal *feature* do SQL Developer mais utilizada ao longo deste projeto foi o editor com capacidade para executar *scripts* e *queries* na linguagem PLSQL. Em conjunto com Java e SQL, foram os três grandes pilares da programação existentes no *SQL Developer* [103].

Como referido anteriormente é necessário então acrescentar as linguagens algo que permita fazer a ponte entre estas e SQL e consequentemente as base de dados Oracle. Deve então ser instalado no caso do Python o "cx_Oracle5.0.2" [106] e um *Oracle Instant Client*, é possível então, com as credenciais certas e uma conexão à Internet aceder e manipular informação presente numa base de dados. Esta ligação será esmiuçada no capítulo 4. Da mesma forma que caso anterior também existe uma ponte entre linguagem Php e as base de dados oracle, desta feita é necessário Apache configurado na máquina de forma a conseguir utilizar OCI8. É necessário também a existência de um *Oracle Instant Client*.

3.4.2 *Mysql*

MySql é um sistema para gestão de base de dados (SGBD), dos mais utilizados em todo o mundo que utiliza linguagem SQL [107]. Criado por David Axmark, Allan Larsson e Michael Widenius na década de 80. Atualmente é a *database management system* mais utilizada em todo o mundo que à semelhança da secção anterior, utiliza o modelo relacional sendo de igual forma criado um novo diretório novo na estrutura de ficheiros que a suporta a quando da criação de uma base de dados nova.

Grande parte do sucesso desta *framework* deve-se à simples incorporação com linguagem PHP assim como a fácil integração e instalação do servidor [101]. Empresas de renome mundial utilizam Mysql como casos da NASA, Wikipedia, Motorola em aplicação vitais, incorporando atualmente Unicode, Full Text Indexes, GIS, OLAP e muitos outros [101].

Outros pontos fortes desta linguagem são [108]:

- Portabilidade;
- Compatibilidade com diversas linguagens como .NET, Python, Perl, PHP, ASP;
- Bom desempenho e estabilidade;
- Gratuito;
- Suporta Triggers, Cursores, Procedimentos e Funções;
- Apresenta interface gráfica como MySql Workbench;
- Capacidade de correr em sistemas de *32-bits* de RAM, correndo assim em quase todos os computadores existentes atualmente;
- Aplicações Web onde existem muitas leituras e poucas inserções. A rapidez do MySql é ideal para este cenário;
- Capacidade das tabelas crescer para suportar um grande volume de entradas;
- Estabilidade até com novas *features* que foram adicionadas recentemente.

Da mesma forma que bases de dados Oracle utilizam o SQL Developer o MySql utiliza o MySql WorkBench, implementando quase todos os *standards* do SQL [108] possibilitando assim *queries* e desempenho de outras tarefas de forma gráfica.

3.5 WEB SERVICES

Web Services são aplicações cliente-servidor que comunicam através da *World Wide Web's (WWW)* Hypertext Transfer Protocol (HTTP) [109]. De acordo com o documento WSA *(Web*

Service Archicheture), publicado pelo *World Wide Web Consortium*, *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an* Extensible Markup Language (XML) *serialization in conjunction with other Web-related standards."* . Existe assim uma relação muito próxima entre a criação de web services e a integração/comunicação de aplicações que se localizam em diferentes sistemas.

Essencialmente um *web service* faz com que recursos provinientes da aplicação de *software* estejam disponíveis *online* de um forma normalizada. Tipicamente um *web service* tem três componentes fundamentais [110]:

1. Um provedor de serviço, responsável pela criação do *Web Service* e registo do mesmo nas entidades competentes;

2. Um responsável por manter os registos dos serviços existentes;

3. Aqueles que executam o pedido de serviços neste caso o cliente.

Pela citação acima referida podem ser retirados mais alguns aspetos, a existência de interface (*Web Services Description Language*) que descreve o *web Services* de acordo com recomendações W3C. E os ditos clientes da lista podem interagir com essa interface através de *SOAP messages*. Embora o "encapsulamento" de dados possa ser realizado por esta, existem alternativas nomeadamente o REST (*Representational Transfer Protocol*) que serão abordados mais adiante.

Hoje em dia a maior parte serviços disponibilizados são invocados através de um *uniform resource locator*. As informações do pedido podem estar contidas neste ou utilizando uma estrutura XML para enviar parâmetros [109].

Service Oriented Archicheture (SOA) é uma abordagem aos sistemas distribuídos onde as funcionalidades das aplicações devem estar disponíveis na forma de serviços acessíveis por *request-reply*. Existem assim algumas propriedades relevantes [111, 112]:

VISTA LÓGICA : O serviço é abstrato, representado programas reais, bases de dados, processos, etc;

ORIENTADA PARA MENSAGENS : o *core* de todo o processo reside no envio e recepção de mensagens entre aquele que faz o pedido e aquele que lhe fornece a resposta. As linguagens de programação em que estes estão implementados não interessam neste ponto.

GRANULARIDADE : troca de pequeno número de mensagens com elevada complexidade;

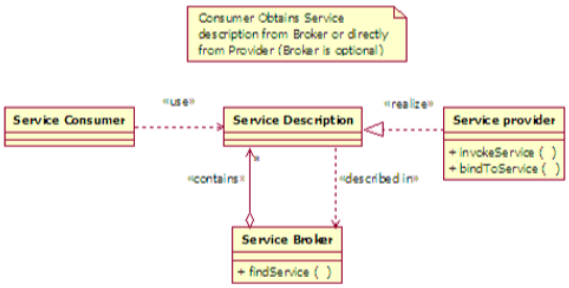


Figura 7.: Modelo conceptual da arquitetura SOA (adaptado de [112])

NETWORK : utilizados essencialmente utilizando a rede embora possam acontecer outros cenários;

INDEPENDENTE DE PLATAFORMA : não estão presa a uma só plataforma porque no fundo aquilo que interessa é a mensagem.

Na figura 7 é então possível observar uma representação desta arquitetura e dos seu componentes principais. As ligações ai observadas são então *web services* [112]. Neste tipo de arquitetura do sistema não fica dependente de unidade ditas centrais, existindo ampla comunicação com sistemas fisicamente divididos mas que partilham conteúdos conduzindo a uma eficiente implementação de serviços [113].

Com a evolução da web na última década surgiram naturalmente soluções mais adaptadas a este campo. Os *Web Services* do tipo *RESTful* são exemplos disso mesmo [114]. *Representational State Transfer (REST)* define assim um conjunto de princípios arquitetônicos focados nos recursos dos sistemas, nomeadamente como esses podem ser acessados por *HTTP* [114]. Atualmente um dos tipos mais utilizados, que no início teve algumas dificuldades em se impor quando apresentado no ano 2000 por Roy Fielding [114]. Com o crescimento da Web as comunicações através de TCP/IP na porta 80 aumentaram sendo agora aceites por quase todas as *firewalls*. A grande diferença deste para os demais tipos de Web Services reside assim na correta e completa utilização do protocolo HTTP.

A utilização explicita do método HTTP é feita em concordância com o protocolo RFC 2616. Dizendo assim, por exemplo, que um método "GET" é invocado por determinado cliente quando este pretende receber um determinado recurso para posterior utilização de informação [115].

Alguns dos pontos que tornam este tipo de *web services* rápidos, leves ao nível de processamento e simples são [116, 115]:

- Identificação do recurso a utilizar utilizando *URI*, fornecendo assim, um endereço global;

- Interface Uniforme, onde apenas são permitidas 4 operações básicas, *GET, PUT, POST, DELETE*. O primeiro termo retorna o presente valor do recurso. O segundo tipo cria um novo recurso dotando este de propriedades definidas pelo utilizador.O terceiro tipo geralmente altera o estado ou alguma propriedade do recurso existente. Por último, o *DELETE* remove determinado recurso de forma definitiva

- Cada recurso pode ser acedido utilizando diferentes formas, *JavaScript Object Notation (JSON)*, XML, *Hyper Text Markup Language (HTML)*, etc.;

- Cada interação com um recurso é *stateless*, depositando do lado do cliente algum trabalho faz com que a resposta seja mais rápida [115].

Uma das discussões mais acesas ultimamente, prende-se com a utilização de *Web Services RESTful* ou *SOAP*. Este último tipo surgiu primeiro que o descrito no capítulo anterior pela mão da Microsoft. A versão 1.2 corrigiu muitos dos erros e pontos fracos da *framework*, tornando-a mais competitiva e fácil de implementar. Embora a carga adicional de processamento ainda se faça sentir apresenta também outras vantagens. Nesta versão a sigla SOAP deixa de reperesentar *Statless Object Access Protocol* passando a poder ser representada por uma de duas abordagens [111]:

- Service Oriented Architecture Protocol, neste caso a mensagem SOAP representa o pedido ou o conteúdo da resposta;

- Simple Object Access Protocol, neste caso a a mensagem pode indicar uma evocação num objeto remoto, sendo que deve ser enviado também a serialização da lista de argumentos.

A base do SOAP assenta em XML de três formas distintas, o envelope onde é definido o conteúdo da mensagem e o método de processamento, conjunto de regras de codificação e o futuro layout das respostas. Assim sendo o envelope é enviado por exemplo por HTTP/HTTPS, existe no destino uma *Remote Procedure Call* e envelope retorna então o resultado final num XML formatado de acordo com *layout* definido [117]. Todos os serviços disponibilizados são então apresentados com recurso a *Web Services Description Language*(WSDL) [118].

O facto de SOAP poder utilizar qualquer meio (SMTP, Java Messaging Service) para enviar o envelope representa uma vantagem em relação ao RESTFUL onde apenas método HTTP/HTTPS pode ser implementado [111]. No entanto perde tempo quando é preciso analisar a resposta motivado em parte pela estrutura XML [117].

A melhor forma é apresentar situações onde se pode implementar uma abordagem e outra. Casos onde RESTful é uma boa alternativa [118]:

- Limitação de recursos e de largura de banda, assim sendo qualquer *browser* consegue fazer retornar a resposta. Por outro lado pode ser utilizado AJAX para retornar XML se desejado;
- Operações sem estado, nomeadamente operações CRUD *(create, read, update, delete)*;
- Situações onde é necessário *cache*, uma vez que a natureza da operação *stateless* permite isso.

Casos onde o SOAP é boa alternativa [117, 118]:

- *Asynchronous calls*, implementado por exemplo com WSRM;
- Necessidade de concordância em relação aos formatos, o XML obriga assim à existência de rigidez quanto a isso;
- Operações onde é necessário guardar estado, com contexto ou informações de segurança;
- Utilizado em grandes empresas com serviços bastante complexos.

Como observado, as duas soluções têm pontos fortes e fracos, cabe ao utilizador escolher, podendo sempre adaptar a sua solução de modo a enquadrar com as suas exigências. O facto de exigir uma curva de aprendizagem acentuada e o *debug* ser bastante complicado, por vezes tornam pouco atrativo o SOAP para utilizadores iniciantes.

3.5.1 *PHP*

Hypertext Preprocessor (PHP) foi apresentada originalmente em meados de 1990, sendo atualmente uma das linguagens mais proeminentes para o desenvolvimento de aplicações Web [119]. Um ponto forte da linguagem é o facto de poder ser embebida dentro da página HTML, assim sendo comandos complexos da linguagem *markup*, podem ser substituídos por simples comandos em Php [119]. Embora o Php tenha a sua génese em aplicações do lado do servidor pode ser utilizado em muitos outros casos, como por exemplo na criação de aplicações para *desktop* ou então para correr comandos da linha de comandos da máquina onde corre [119].

De entre uma lista de 10 vantagens da utilização do PHP para aplicações web deve-se referir, a facilidade de implementação, a capacidade de produzir conteúdos (Facebook, Twitter ou Wikipedia utilizam php), ligação às principais bases de dados (Mysql, Oracle) utilizando conectores, suporta todos os servidores web (Apache Microsoft IIS) e ainda apresenta uma segurança superior a relação a demais linguagens [120] .

Neste projeto a linguagem PHP foi utilizada para implementar um *RESTful*. Assim sendo é necessário implementar um conector à base de dados oracle, geralmente *OCI8*. Este

serviço tem de estar contido numa pasta do apache (por defeito ou configurada para tal) que servirá de servidor web. No caso de máquinas windows pode ser utilizado Xampp um emulador de ambiente Apache [121].

Como visto anteriormente o tipo RESTful necessita de configuração especial para serviços URI para tal, o apache tem de ser configurado para aceitar tais pedidos e ainda tem de ser criado um ficheiro com nome *.htaccess*, sendo este o responsável por redirecionar pedidos realizados na pagina web (utilizando url) para a o web service (ficheiros reais) [121].

O próximo passo é a criação de uma classe RESTfull que serve de base ao serviço. É aqui também que se definem alguns métodos a ser utilizados pelos *service handlers*. Por exemplo métodos para conter diferentes estados do pedido HTTP (200 representa "OK", 202 representa "Accepted") [121].

Por último é então criada a classe que vai processar pedidos, é aqui que no caso deste projeto o *OCI8* é necessário criando assim uma ponte entre aplicação e a base de dados.

3.6 Frameworks e Linguagens de Desenvolvimento Web

Uma *web framework* ou *web application framework* é pensada para o desenvolvimento de aplicações web incluindo *web services*, *web resources* e *web APIs*. A sua função prende-se essencialmente com retirar alguma carga de trabalho e simplificar as atividades a ser desenvolvidas, nomeadamente a relação pedido/resposta quando se executa um pedido no formato HTTP [122].

Nesta secção é assim abordada uma *framework* que tem ganho bastantes adeptos nos últimos anos e ainda a principal linguagem sobre qual esta assenta.

3.6.1 *Angulhar JS*

Angularjs é uma *superheroic framework* em *JavaScript* gratuita atualmente mantida pela *Google*, que faz sucesso no campo das *single-page applications* [123]. Data do ano 2010, atualmente encontra-se na segunda versão, mas aquela que ainda é mais utilizada é a 1.3 visto apresentar mais informação online e maior estabilidade que as demais [124].

Desde o lançamento do HTML em 1990 por Tim Berners-Lee sucederam-se quase em catadupa os desenvolvimentos nesta área. Nos últimos anos o desenvolvimento do JavaScript, HTML e CSS intensificou-se proporcionado novos níveis de experiencia ao utilizador. *Angularjs* está inserido nessa nova vaga de librarias e *frameworks* capazes de produzir aplicações web mais apelativas e flexíveis com maior facilidade [125].

Permite o desenvolvimento de grande aplicações sem grande esforço apenas sabendo HTML e *JavaScript*. A estrutura da aplicação fica a cargo do *Angularjs* que assenta tanto no principio MVVM , ou seja, *model-view-view-model* como em MVC *(Model-View-Controller)*.

Este último modelo pode ainda ser adaptado para um mais recente MVW *(Model-View-Whatever)* [123].O mais importante é que existe uma clara separação entre componentes promovendo a modularidade, flexibilidade e operações de teste [125].

Abordando agora um pouco do MVC, visto ser a base dos demais, este define três papeis principais [123, 124]:

- O modelo, grande motor de toda a aplicação,ou seja, os dados por detrás de aplicação com provenientes de servidores;
- A vista representa a interface com o utilizador. Deve ser dinâmica e com base no estado atual do modelo;
- O controlador é aquele que desenvolve toda a *business logic* e a camada de apresentação. É também responsável pela partilha de dados entre modelo.

Pela pequena lista é possível concluir que cada componente está focado apenas num tipo de função muito específico. Na figura 8 é possível observar o modelo simples pelo qual este esquema se rege.

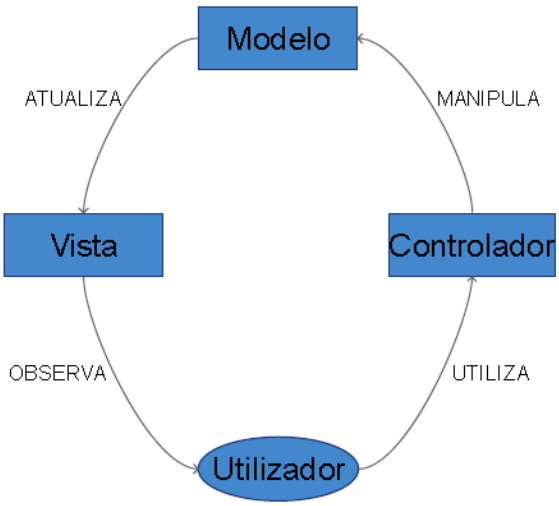


Figura 8.: Representação típica modelo MVC

Data Binding é o nome atribuído ao processo de sincronização entre o modelo e componentes da vista, quando o primeiro é alterado o segundo modifica-se de imediato, o mesmo acontecendo ao em sentido oposto [126]. Embora na figura 8 a seta que faz a ligação entre modelo e vista seja unidirecional, o fluxo de dados é bidirecional. Enquanto em sistemas clássicos a partilha de informação acontecia apenas num único sentido, não refletindo mudanças posteriores ao primeiro carregamento. O programador teria então de escrever

código que realiza-se a sincronização num espaço temporal. O caso do Angularjs é diferente, primeiro o template (HTML) é compilado, existindo uma natural extensão ao HTML normal, no *browser* produzindo assim uma vista. Qualquer mudança nesta é imediatamente refletida no modelo e vice-versa. O controlador neste caso não sabe que existe uma vista (independente), facilitando em muito o processo de teste, porque não existe *DOM/browser dependency* [126]. A partilha de informação utiliza então um objeto partilhado de seu nome *scope*. A acção deste pode ser equiparada ao de uma "cola" que junta o modelo com a vista permitindo assim a partilha eficaz de dados.

Angularjs apresenta alguns objectivos/beneficios sobre quais deve ser feita uma breve referencia [126, 125]:

- O teste é tão importante como a aplicação normal;
- Utilizar o menor número de linhas de código possível;
- Quase todo o código escrito, acontece no controlador, embora nada tenha a ver com questões ligadas à estrutura MVC. Isso fica a cargo do *Angularjs*;
- Tentar implementar uma abstração entre o lado do cliente e do servidor sendo o processo de implementação realizado em paralelo;
- Existir sempre um acompanhamento em todo o processo de codificação;
- A integração com CSS e BootStrap é uma realidade sem necessidade de codificação em *JavaScript* adicional;
- Integração simples com livrarias provenientes de outras fontes como *jQuery* e *BootStrap*.

Um ponto importante é então a injeção de dependências resultando na passagem de serviços do lado do servidor para o lado do cliente, como é o caso dos controladores reduzindo bastante o consumo de recursos na parte do servidor [126].

Algumas das directivas mais importantes desta *framework* que permitem realizar a extensão ao html são por exemplo:

- *ng-app*, faz do elemento html onde se insere a raiz da aplicação;
- *ng-model*, permite uma ligação bidirecional direta entre vista e scope da aplicação.
- *ng-controller*, especifica um controlador *JavaScript* para determinado html;
- *ng-view*, utilizado para definir rotas internas utiliza JSON para tal.

JavaScript é uma linguagem interpretativa, orientada a objetos embora de uma forma muito rudimentar que apresenta baixo consumo de recursos [127]. Utilizada geralmente a pensar em *newtwork-centric* applications, ou seja inserida em páginas web.

A linguagem foi primeiramente apresentada como *LiveScript* pela empresa criadora *Netscape* corria o ano de 1995 [127]. Ao nível da sintaxe apresenta semelhanças com C, C++ e Java utilizando construtores como if ou while. Uma diferença é o facto de nesta linguagem não ser necessário declarar qual o tipo de uma determinada variável [127]. Os objetos nesta linguagem pouco ou nada têm a ver com aqueles vistos em Java, assemelhando-se mais ao Perl onde é implementado um *associative array*.

Sendo muito utilizada dentro das páginas html leva a páginas web bastante dinâmicas e com baixo consumo de recursos uma vez que é puramente interpretativa não sendo compilada como C ou Java [127].

Muito mais poderia ser abordado neste capítulo sobre Angularjs, mas apenas se realizou uma contextualização teórica sobre as principais características desta *framework*.

3.7 DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

A importância das aplicações móveis no mundo atual é inolvidável sendo milhares as pessoas que fazem do seu trabalho desenvolver aplicações para sistema operativo *Android*. Existem assim diversas ferramentas que possibilitam e facilitam a programação como por exemplo *Ionic* ou *Xamarim*. Optou-se no entanto por utilizar as ferramentas fornecidas pela *Google*, nomeadamente o seu *Android Software Development Kit (SDK)*. Ao longo da secção serão abordadas as principais características deste software e quais as ferramentas de programado que mais se enquadram com o presente projeto.

3.7.1 *Android*

Android é um sistema operativo com núcleo em *Linux* lançado no ano de 2008 e atualmente gerido pela *Google* [128]. Embora o seu foco principal seja *smartphones* e *tablets*, pode também ser encontrado em televisões, relógios e até carros. Embora a licença seja aberta a maior parte dos dispositivos apresenta um híbrido entre livre disponibilizado pela Google e o privado geralmente adicionado pela companhias que fabricam o dispositivo.

Programar para dispositivos móveis é completamente diferente de programar para um computador, motivado em grande parte por restrições bastante especificas deste tipo de dispositivos. Vários factores devem então ser tidos em consideração [128]:

- **Recursos Limitados**, embora bastante poderosos os dispositivos móveis continuam a ser limitados quando comparados com um computador. Nomeadamente memória RAM, processador e capacidade da bateria;

- ***Mobile Mashups***, ou seja, a incorporação de aplicações e APIS já implementadas noutras situações;

- ***Interchangeable Applications***, a utilização de *Intents* é independente de uma determinada implementação em especifico.

O dispositivo não é um ambiente fechado, como qualquer sistema operativo destinado a utilizadores, permite a estender as suas funcionalidades utilizando aplicações. Estas podem ser desenvolvidas em Java em conjunto com o sistema de desenvolvimento do *software* Android(SDK). *Software Development Kit* disponibiliza ferramentas que permitem a programação de aplicações ou soluções para um determinado *software*.

Todas as aplicações Android desenvolvidas desta forma apresentam 4 componentes básicos na sua arquitetura [128]:

- **Atividades**, são porções de código que podem ser chamadas quando necessário e correm num janela temporal desempenhando determinada tarefa. A maior parte do código implementado será neste formato;

- **Serviços** porções de código que correm nos bastidores desempenhando funções de interação com bases de dados por exemplo;

- *Broadcast and Intent Receivers* respondem a eventos recebidos na aplicação com origem no próprio *Sistema Operativo (SO)*, numa outra aplicação ou por exemplo um aviso Push;

- **Provedores de Conteúdo**, criados para partilhar dados entre atividades ou serviços.

Android apresenta grande capacidade de reconhecer que dispositivos móveis têm capacidades limitadas, assim sendo ajustam o seu ciclo de vida para melhorar e apresentar o melhor conteúdo possível ao utilizador. Como visto na figura 9 onde são definidas as diversas etapas e eventos pelo quais uma aplicação passa, este é completamente diferente daquele definido para uma aplicação de *desktop*. As abas ou vistas de uma aplicação são então as atividades, quando um aplicação é iniciada é invocada a função *onCreate()*, onde as disposições gráficas são criadas. Só após a função *onStart()* ser "chamada" é que os pedidos a web services e outras funções são executadas. Um ponto marcante na figura 9 é a pergunta *In Foreground*, esta define o ciclo de vida de uma atividade. Se estiver em utilização (*Yes*), segue o ciclo normal, caso contrário, passa diretamente para o ponto *onStop()*, podendo voltar a ser utilizada e o código volta ao estado *onStart()*, ou então é *Killed* em definitivo, voltando ao ponto inicial. Embora os IDE e *framewoks* façam este código de forma automática, é importante ter em conta este ciclo quando existem pedidos a web services, porque estes podem ditar atrasos em todo o processamento.

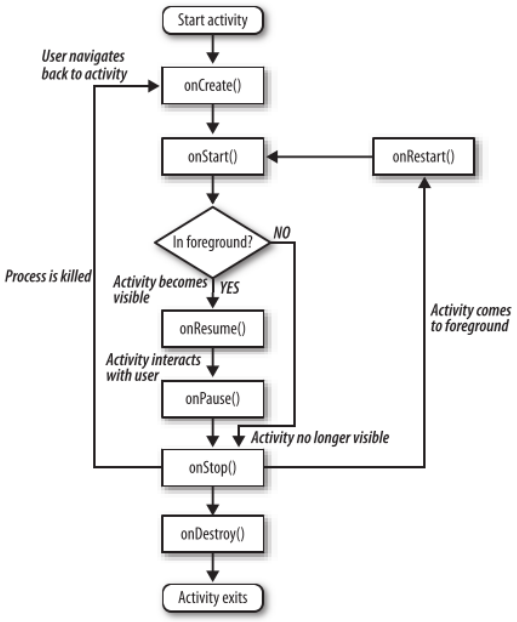


Figura 9.: Ciclo de vida de uma aplicação android (adaptado de [128]).

3.7.2 *Java*

A linguagem Java é uma linguagem orientada para objetos com base em classes, desenvolvida por James Gosling, colaborador da Sun Microsystems no ano de 1995 [129]. O principio por detrás da linguagem é *"write once run anywhere"*. Ou seja o código apenas deve ser compilado uma vez sendo que depois pode ser utilizado em qualquer máquina. Java é atualmente na versão SE 8, é uma das linguagens de programação mais utilizadas principalmente em aplicações cliente-servidor contando em todo o mundo com 9 milhões de programadores a utilizar a linguagem.

A linguagem foi criada com 5 princípios básicos [129]:

- Deve ser simples, orientada a objetos e familiar;
- Deve ser robusta e segura;
- Deve ser independente de arquitetura e portátil;
- Deve ser executada com baixos recursos de processamento;
- Deve ser interpretativa, dinâmica e com recurso a *threads*.

Como dito anteriormente, é necessário uma plataforma para conseguir compilar um programa, no contexto do presente capítulo é o SO android. Embora seja uma distribuição não

oficial uma vez que utiliza características de Java 6 e algumas de Java 8 numa libraria diferente do habitual (Apache Harmony). Apresenta uma *virtual machine* diferente que recorre também a um *bytecode* diferente (*Dalvik bytecode*) [129]. Por ser neste caso orientado para dispositivos móveis, focou-se então no baixo consumo de memória.

3.8 CONCLUSÃO

Ao longo do presente capítulo foram abordadas as ferramentas transversais a todos casos de estudo, que vão ser apresentados nos próximos capítulos. Embora sejam ferramentas e linguagens distintas permite representar o vasto leque de recursos disponíveis quando se pretende implementar soluções relativas à partilha de informação.

Sem dúvida que os web services no acesso a informação em base de dados assumem elevada importância dai a explicação um pouco mais exaustiva e a comparação entre diversos tipos.

O Angularjs é exemplo de uma *framework* recente que trouxe novas ideias ao mundo do desenvolvimento de soluções web e que simplifica em muito o trabalho do programador. Nomeadamente a utilização de ferramentas que possibilitem a reutilização de código em diversas plataformas. A *framework* é exemplo disso mesmo. Utiliza o Angularjs apenas com diversas alterações nas *tags* HTML, possibilitando a escrita de apenas um código para Android e Web.

Aspetos mais técnicos de cada secção não foram abordados por dois motivos, não ser esse o objetivo do presente projeto ou então vão ser referidos em maior detalhe no caso de estudo em que forem utilizados.

4

CASO DE ESTUDO - SISTEMA DE ENVIO E RECEÇÃO DE MENSAGENS EM FORMATO HL7

Num momento em que as autoridades um pouco por todo o mundo estão a fomentar a interoperabilidade entre sistemas de informação é vital que no caso das unidades de saúde existam sistemas de envio e receção de mensagens em formato *Health Level Seven International (HL7)* capazes de estar à altura dos acontecimentos.

Embora não seja uma necessidade identificada no Centro Hospitalar do Porto uma vez que já existe um sistema responsável pelas funcionalidades descritas anteriormente, é importante ter sistemas alternativos com capacidade de competir com aquele implementado.

Ao longo do presente capitulo é então apresentado o trabalho desenvolvido na criação de um sistema de envio e receção de mensagens em formato HL7. Primeiro é feita uma pequena introdução e contextualização com o tema, seguindo-se uma abordagem dos materiais e da metodologia implementada. Nas duas últimas secções são apresentados resultados e conclusões.

4.1 INTRODUÇÃO E ENQUADRAMENTO NO TEMA

A evolução das *Tecnologias de Informação e Comunicação (TIC)* conduziu também a um rápido desenvolvimento no campo dos sistemas de informação hospitalares. Cada vez mais, registos eletrónicos de pacientes, resultados provenientes do laboratório, dados financeiros e outros são transferidos utilizando redes de computadores. Embora os avanços em *Health Information Systems (HIS)* se processem de forma quase exponencial, existem ainda alguns problemas relativos à integração de sistemas informáticos heterogéneos.

Nos últimos anos muito tem sido feito um esforço para fomentar a interoperabilidade sintática entre sistemas que por natureza não comunicam utilizando um formato semelhante. Existem no entanto dificuldades/desafios para a comunicação de informação a nível de unidades de saúde, como por exemplo [42]:

- Acesso a informação clínica em múltiplos sistemas de informação;
- Pouco investimento por parte das instituições de saúde;

- A criação de interfaces feitas à medida de uma unidade de saúde ou secção, aumenta a dificuldade da difusão de informação;
- Interfaces bastante dispares entre sistemas dificulta o estabelecimento de protocolos de comunicação;
- Com o aumento do número de sistemas, o número de interfaces aumenta fazendo com que a dificuldade de conexão seja superior.

Quando sistemas não conseguem por si só comunicar, deve ser colocado entre ambos um "interprete" viabilizando assim a comunicação, fazendo uma analogia com o caso humano. Neste cenário, as unidades de saúde optam pela utilização de *standards* de comunicação propostos pela *HL7 International*. Estes foram abordados com maior detalhe no capítulo 3, devendo ser relembrado que esta organização é responsável pelo lançamento e a atualização diversas diretrizes na tentativa de fomentar a interoperabilidade *(Conceptual Standards, Document Standards, Application Standards, Messaging Standards)* [130, 131]. De entre as opções existentes aquelas que melhor desempenham o papel de "interprete" são *message standards*, nomeadamente a versão 2 para a troca de mensagens em contexto clínico [42]. A organização não produz *software*, daí que todo o trabalho de desenvolvimento fica a cargo de empresas especializadas ou dos responsáveis informáticos nas unidades de saúde, seguindo as *guidelines* publicadas. As implementações são a "face" do *Sistemas de Informação (SI)* na rede que constitui os *Sistemas de Informação em Saúde (SIS)*, denominadas de interfaces, apresentam a responsabilidade de enviar e receber dados e ainda perceber o meio onde estão inseridas.

Representando esta, uma área de elevado interesse económico, existem diversas alternativas no mercado. No documento de 2012 intitulado *"HL7 Interface Technology Survey Results"* divulgado pela Core Health Technologies, é apresentada uma visão geral do mercado de interfaces HL7 [132]. As mais utilizadas são a Cloverleaf da *Lawson (Quovadx)*, a Epic Bridges da *Epic* e o Mirth Connect da *Mirth Corporation* [132]. Nessa mesma lista surge ainda uma distribuição que nos últimos anos tem ganho destaque, Iguana da empresa InterfaceWare. Para além de apresentar uma interface mais apelativa que as demais, surge com novas ideias relativas a *Business Intelligence (BI)* sobre troca de mensagens, controlo de agentes à distância e escrita de mensagens utilizando a plataforma Web disponibilizada. No mesmo documento é perceptivel que quando os responsáveis de uma organização optam por uma distribuição, não a pretendem substituir num futuro proximo. Só em casos de extrema necessidade, prova disso são os 70% de respostas negativas quando questionado se nos próximos 5 anos pretendiam alterar a sua interface [132].

Aplicações com a função d interface são um ingrediente essencial para alcançar a interoperabilidade. Exemplos de aplicações que necessitam dessas implementações são o *Processo Clínico Eletrónico (PCE)*, o HIS, o *Rede Informática da Saúde (RIS)*, o *Laboratory Information Sys-*

tems (LIS), a *Picture Archiving and Communication System (PACS)*, a triagem e urgencias, o departamento financeiro e muitos outros que constituem uma unidade de saúde. A *Corepoint HEALTH* propõem no documento *"What is Your Healthcare Interface Method"* duas formas distintas de realizar a implementação e ainda quais as principais características que devem apresentar [133]:

1. ***Point-to-point:*** as comunicações processam-se de forma independente;
2. ***Interface Engine:*** situado entre as aplicações para ajudar a implementação e monitorização.

Para facilitar a comunicação entre sistemas que utilizam HL7, a interface deve incluir:

- Um terminal de envio para enviar mensagens;
- Um terminal de receção para receber mensagens;
- Um método para enviar a mensagem entre dois terminais;
- Um método para pesquisar mensagens;
- Um método para registar o fluxo de mensagens.

O modelo *point-to-point* (ver figura 10) utiliza especificações variadas consoante os sistemas que se pretendem conectar numa relação *one-to-one*. Cada caso deve ser averiguado de forma individualizada, por exemplo, o modelo utilizado para conectar a urgência com o LIS pode ser diferente daquele que conecta o HIS com o LIS uma vez que o primeiro exige um maior número de mensagens logo deve ter maior capacidade *multiprocessing*. De acordo com entidades que utilizam este modelo, a principal vantagem da implementação é a tal especificidade com que é desenvolvido uma vez que sistemas são configurados de prepósito para o efeito. As principais desvantagens são os custos da implementação em larga escala, impossibilidade de monitorizar o estado de todas as ligações de forma continuada e o facto de a complexidade de implementação/manutenção aumentar proporcionalmente com o aumento do número de interfaces [131, 133].

O modelo *interface engine* (ver figura 11) por seu turno adapta a mensagem de forma a que possa ser aceite pela aplicação de destino sem que seja necessário predefinir isso a quando da implementação da interface. É utilizado o conceito de *one-to-many*, onde os módulos de *import* e *export* estão ligados a um motor de interface para que este as processe e monitorize. De acordo com as entidades que utilizam este modelo, as principais vantagens são a flexibilidade para adaptar a diferentes formatos HL7 assim como diferentes especificações por parte das aplicações, o baixo custo de implementação de uma interface uma vez que é agrupado o trabalho de *import* e *export* e por fim, o aumento da capacidade de monitorização. Como principal desvantagem está a maior sobrecarga da unidade computacional responsável pelo motor de interface [131, 133].

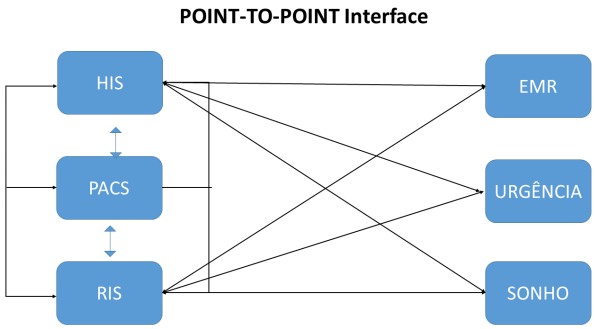


Figura 10.: Exemplo do tipo de implementação *point-to-point* (adaptado de [133])

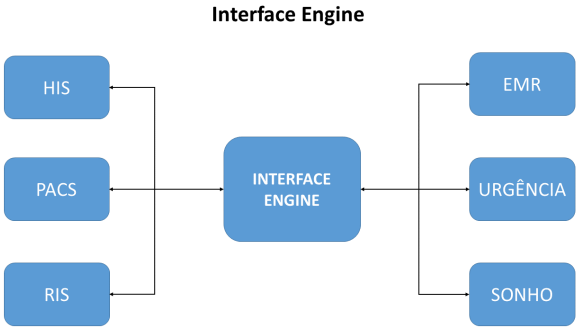


Figura 11.: Exemplo do tipo de implementação *interface engine* (adaptado de [133]).

Existem então diferenças significativas entre as duas abordagens propostas, facilmente percetíveis pela observação das figuras 10 e 11 respetivamente. Existe no entanto uma diferença significativa, enquanto a abordagem *point-to-point* só permite o envio de mensagens utilizando um formato definido pela aplicação de destino, o segundo caso tem capacidade de fazer envio de diversas especificações. Para isso, as plataformas de envio e receção são pensadas e implementadas com foco na interoperabilidade e nos potenciais formatos a serem trocados. São posteriormente conectados a um *"engine"* que faz todo o mapeamento, *routing* e monitorização do sistema. Qualquer que seja a escolha para interface, esta deve apresentar propriedades como modularidade e capacidade de adaptação, disponibilidade, escalonamento.

Embora estes sistemas abordados anteriormente sejam capazes de transportar mensagens em diversos formatos (HL7, Xml, etc.), o propósito deste trabalho reside em mensagem V.2 das diretrizes HL7. Como referido por Matheus Coppetti Silveira no trabalho Sistemas Distribuídos Hospitalares, "*HL7* não é um *software* ou ferramenta, e sim, um conjunto de regras estabelecidas para o envio de mensagens através de blocos de texto que podem ser transferidos entre os mais diversos sistemas" [134].

De acordo com *"Hl7 Bridge Requirements Configuration, Installation and Configuration Guide"*, proposto pela *Scientific Technologies Corporation* a comunicação eletrónica de mensagens é em

tudo igual à comunicação entre humanos a quando da marcação de um encontro. Deve ser combinado o local (método de transporte), a linguagem (formato da mensagem) e vocabulário (conteúdo da mensagem) [135].

Focando atenções ao nível do transporte existe o protocolo *Minimal Lower Layer Protocol (MLLP)*, com longo historial na comunidade HL7 [136]. Este protocolo é uma sessão *Organização Internacional para a Normalização (OSI)* minimalista, utilizado apenas um ambiente de network recorrendo a aspetos básicos dos protocolos de network(TCP/IP) [136]. O objetivo principal é criar uma camada entre HL7 e a rede, tentando causar a menor sobrecarga possível [136]. Uma das limitações deste protocolo prende-se com o baixo suporte para codificação (*character encoding*) motivado pela utilização de *single-byte values*. A razão desta codificação, é a tentativa de não entrar em conlfito com aqueles caracteres que fazem marcação dentro da mensagem, sendo o conteúdo da mensagem envolvido por caracteres especiais, como por exemplo [135, 136]:

$< SB >$: indica o início do bloco de mensagem;

$CORPODAMENSAGEM$: apresenta um número variável de *bytes*. Pode assim conter *single-byte*;

$< CR >$: *Carriage Return* indica o final de determinado segmento;

$< EB >$: Assinala o fim do bloco de mensagem.

De todos os protocolos de rede, talvez o mais conhecido seja *Transmission Control Protocol/ Internet Protocol*(TCP/IP) que é o nome atribuído ao conjunto de protocolos de rede utilizados na construção da *Internet*, também conhecido como DOD *(dee-oh-dee)* ou *Arpanet* [137]. Definido no ano de 1974, TCP assenta nos princípios da transmissão de pacotes do protocolo *Internet Protocol (IP)* possibilitando as aplicações de comunicarem utilizando *streams* contínuos de dados [138]. Salvo raras exceções (perda de conexão ou bloqueio devido a problemas de rede), este protocolo garante que o *stream* de dados chega intacto ao seu destino. Não existindo perda de pacotes, alteração de ordem ou duplicação. Por seu turno o protocolo *User Datagram Protocol* (UDP) é um simples protocolo da camada de transporte que faz encapsulamento num pacote IPV4 ou IPV6 enviado para seu destino, a diferença em relação a TCP é o facto de não garantir que este, vá chegar ao seu destino. Neste tipo de ligação, a conexão cliente-servidor é mais breve que a anterior necessitando assim de menores recursos de *internet* [98].

Também relacionado com aspeto de redes, surge o conceito de *socket*, este funciona como ponto de origem ou destino de uma rede de computadores ou programas. Hoje em dia, utiliza o protocolo IP sendo assim a maior parte são *internet sockets* [138]. *Socket API* é *application programming interface (API)* disponibilizado pelo sistema operativo permitindo assim o controlo de *network sockets*. *Socket Address* é a combinação de IP e número de porta, identificando inequivocamente aquele destino dentro daquela rede [138].

Atualmente alguns dos trabalhos estão relacionados com o envio de mensagens com elevado conteúdo utilizando as diretrizes HL7. O trabalho de King Sung Um e seus colaboradores, *"Development of an HL7 interface engine, based on tree structure and streaming algorithm, for large-size messages which include image data"*, faz uso de uma das diretivas da organização *No limitation of message length*, para em primeiro lugar, provar que isso nem sempre é verdade e segundo implementar um sistema com capacidade para tal [139]. Outros trabalhos estão voltados para a validação de mensagens HL7, como o trabalho apresentado por Ean-Wen Huang intitulado *Design and Implementation of a web-based HL7 message generation and validation system*. Nesta implementação foi desenvolvida uma aplicação Web que faz a validação de mensagens em formato HL7, com a integração deste sistema numa interface torna possível a validação de mensagens em tempo real [42].

A geração e o envio de mensagens em formato HL7 pode ser realizada em diversas linguagens de programação utilizando uma variedade de paradigmas e abordagens. José Machado e seus colaboradores, em *Modeling Medical Ethics through Intelligent Agents*, propõem uma abordagem através de agentes inteligentes para diversos aspetos da atividade hospitalar [140]. Por seu turno, Luciana Cardoso em *Healthcare Interoperability through Intelligent Agent Technology* apresenta uma visão mais focada em sistemas de agentes inteligentes (MAS) dedicados por inteiro à interoperabilidade [33].

Um agente inteligente pode ser visto como um artefacto computacional que apresenta propriedades como autonomia, pro atividade, reatividade e capacidades de interação social. De acordo com Luciana Cardoso e seus colaboradores, em *The Next Generation of Interoperability Agents in Healthcare*, *" They should be defined as troubleshooting entities of autonomous computing capable of operating effectively in a dynamic environment. They are often used in environments where they interact and cooperate in order to achieve a global goal. An intelligent agent is endowed with autonomous behavior, which allows it to decide and to execute its own actions, without direct human intervention . The pro-activity allows one to plan and to perform tasks designed to achieve the proposed objectives. The social skills allow the interaction and cooperation of one agent with the others, which belong to the same multi-agent system. Finally, the reactivity enables the agent to be aware of the environment where it is inserted, and consequently, it reacts to stimuli captured by the sensors"* [35]. Segundo a citação acima apresentada, um agente inteligente tem capacidade de tomar decisões de forma autónoma, não necessitando de intervenção humana. A pro atividade permite alcançar objetivos delineados pelos sistemas e as capacidades sociais permitem a interação com outros agentes de forma competitiva ou cooperativa para assim alcançar objetivos.

Um sistema multi-agente é visto então como um aglomerado de agentes que herda as propriedades supra mencionadas. A agência FIPA (*Foundation for Intelligent Physical Agents*) emergiu como o *standard* para a programação orientada a agentes. De entre todas as diretrizes propostas por esta organização, destaque para uma delas destinada a normalizar a

comunicação entre agentes assegurando assim a interoperabilidade(*Foundation for Intelligent Physical Agents-Agent Communication Language (FIPA-ACL)*) [141].

A ideia da programação orientada a agentes foi proposta por Shoham como um novo paradigma de programação que combina noções mentais para programação autônoma de agentes [142]. Esta corrente de programação apresentou evolução significativa nas ultimas duas décadas mas ainda não atingiu o seu verdadeiro potencial [143].

O controlo sobre sistemas multi agente foi nos últimos anos facilitado por avanços em comunicação e computação [144]. Um aspeto importante na implementação de algoritmos distribuídos prende-se com a comunicação e controlo sobre agentes [144]. Assim sendo é imperativo para além de um sistema de agentes exista um sistema de controlo.

Como é possível observar pela contextualização, ao longo do presente caso de estudo, será implementado uma interface para o envio e a receção de mensagens em formato HL7. Como apresentado no capítulo 2 as diretrizes HL7 surgem como um dos principais impulsionadores da interoperabilidade sintática, possibilitando assim a partilha de informação. Embora o Centro Hospitalar do Porto apresente um sistema interface para partilha de mensagens, este utiliza o método *point-to-point*. Desta forma o presente caso pretende inserir uma alternativa viável capaz de reduzir custos monetários e de recursos, ao mesmo tempo que aumenta capacidade de resposta.

4.2 MATERIAIS

Ao longo da presente secção serão abordados os materiais utilizados no desenvolvimento do presente caso de estudo. Alguns dos materiais já foram abordados com maior detalhe no capítulo 3 uma vez que são transversais a todos os casos de estudo.

Os materiais utilizados para o desenvolvimento deste caso de estudo são:

- **Design Research** como metodologia de investigação e desenvolvimento;
- **Python versão 3.4** como linguagem de programação;
- **HL7 V.2** como *guideline* ao envio e receção de mensagens clínicas.

A linguagem Python é passível de instalar *plugins* que criam a extensão ao código fonte original. Para desenvolver esta aplicação recorreu-se a bibliotecas externas como por exemplo *python-hl7* [145]. Um livrária simples para fazer *parse* de mensagens em formato HL7 para objetos em Python como por exemplo listas. A livrária *psutil (process and system utilities)* é multi plataforma que permite obter informações sobre processos que estão a decorrer e ainda sobre o estado do sistemas como por exemplo CPU, memória ou rede. É uma ajuda vital para determinar o estado de um determinado computador ou unidade computacional visto que pode correr em diversas plataformas (*Windows, Linux (Solaris, Ubunto), OSX*) [146].

Existem outras livrárias que já fazem parte do código fonte da linguagem Python, como por exemplo threading, *os*, *platform*, *sys*, *os* e mais importante *socket*. Este último, permite acesso ao *BSD socket interface*. Disponibilizado em todos os sistemas *Unix, Windows, MAC OS* permite uma interação direta com objeto *socket* permitindo assim efetuar chamadas ao sistema [147].

4.3 METODOLOGIA

Como abordado no início da secção anterior, foi utilizada a metodologia *Design Research*, nesse sentido tudo começa com a definição dos requisitos/problemas para este caso em particular:

- Interface para Sistemas de Informação que utilize exclusivamente os padrões Hl7, na versão 2. para envio de mensagens clínicas;
- Alternativa ao sistema *point-to-point* atualmente implementado na unidade de saúde;
- Utilização de base de dados Oracle;
- Utilização da linguagem de programação Python.

Naturalmente, a implementação de um sistema de envio e receção de mensagens pode ser bastante simples, a linguagem Python apresenta o modulo *socket* que permite trabalhar em alto nível sobre a rede. Basta criar um script que desempenhe papel de servidor e outro papel de cliente e enviar uma mensagem. Naturalmente num caso complexo destinado a enviar milhares de mensagens diariamente, entre diferentes pontos de uma unidade hospitalar, tal implementação está fora de questão.

A linguagem Python teve nos últimos anos um aumento de utilizadores principalmente na área da engenharia, embora seja mais lenta no processamento, apresenta maior facilidade de aprendizagem, desenvolvimento de código e manutenção de *scripts* já implementadas. Por outro lado da lista de *frameworks* com funções semelhantes aquela que se pretende implementar, nenhuma foi desenvolvida utilizando a linguagem Python. A *Python Network Programming* teve nas últimas versões, algumas remodelações que simplificaram algumas funções tornando a vida ainda mais simples ao programador.

A escolha entre o tipo de interface a implementar tem de ser bastante ponderada visto que determina o sucesso ou insucesso da implementação. Se por um lado desenvolver grande número de interfaces personalizadas é complicado, por outro lado, pensar nos diversos cenários à partida também apresenta complexidade. Pensou-se então utilizar agentes inteligentes que fazem a comunicação entre aplicações mas que reportam para uma unidade central. Cada agente pode ainda enviar mensagens para diversos destinos em simultâneo ou receber de diferentes origens também em simultâneo. Optou-se assim

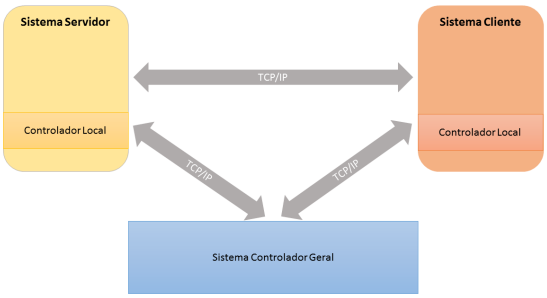


Figura 12.: Esquema geral dos sistema multi agente desenvolvido

por uma adaptação do conceito *interface engine* integrando um MAS que conduz a uma modularização do conceito inicial simplificando a implementação inicial.

Como abordado anteriormente existem pontos importantes quando se implementa um MAS e a interface que devem ser apresentados:

- Protocolo de comunicação e cooperação entre agentes;
- Arquitetura do sistema de forma a recuperar de erros;
- Comunicação com bases de dados;
- Controlo à distância de agentes inteligentes.

Na implementação dos pontos da lista anterior Python desempenha um papel fundamental, tanto ao nível de *multiprocessing* como as diversas funções relacionadas com comunicação e redes. Na figura 12 é possível observar o esquema geral do trabalho desenvolvido utilizando *scripts* desta linguagem. Como proposto pelo esquema de motores de interface, cada agente deve então ter capacidade de enviar mensagens para diversos destinos se o papel for inverso várias origens devem ser contempladas.

A melhor forma de demonstrar a metodologia de implementação é separando em secções os diversos componentes.

4.3.1 *Sistema Servidor*

Este tipo de agente tem como principal missão receber mensagens em formato HL7 e reencaminhar para a sua pasta de destino, apresentando um conjunto de propriedades como autonomia, disponibilidade e capacidade de aguentar um número considerável de mensagens. Assim sendo as tarefas/propriedades deste tipo de agentes são melhor demonstradas na forma de lista:

- Cada agente inteligente é implementado numa máquina com acesso à *Internet* ou *Intranet*, na forma de *Python script*;

- Na configuração inicial de um agente é obrigatório um nome que é inserido no script no momento da instalação;
- Através do nome do agente, é possível aceder ao ficheiro *info.xml* para recolher informações como:
  1. Nome do programa origem da conexão;
  2. Nome do programa destino da conexão;
  3. Nome do agente de origem;
  4. Porta em que o *socket* recebe conexões;
  5. Porta em que o controlador local está a receber conexões.
- Cada agente servidor apresenta um IP, geralmente o da máquina onde estão inseridos;
- O número de processo é o identificador local representando de forma inequívoca o agente;
- O identificador único do agente é uma *string* constituída pelo nome do agente mais tipo (S neste caso) mais o IP;
- Cada agente cria um *socket* utilizando as diretrizes TCP/IP com o IP da máquina onde está inserido e a porta que lhe foi atribuída na configuração inicial;
- O agente ativa a função *listen* do *socket* e aguarda por conexões por parte de clientes. Quando esta acontece é criada uma *thread* para proceder à receção de determinada mensagem;
- Cada programa destino tem uma pasta criada pelo agente onde são depositadas as mensagens recebidas, após confirmação da validade ao nível dos padrões HL7;
- Um agente servidor mantém estreita comunicação com o controlador local que lhe indica as suas conexões através de comunicação via *socket* dentro da mesma máquina;
- Existem funções Python para diversas partes da mensagem, ou seja primeiro segmento, meio da mensagem e último segmento;
- A análise do primeiro segmento é vital, devem ser retiradas informações como ID, Origem, Destino e ainda o tipo de ACK. Se necessitar de tal deve ser criada uma mensagem em formato hl7 que comunique de volta ao cliente o resultado final da operação de envio;
- o agente coloca na pasta de destino a mensagem utilizando um ID com a terminologia *.hl7*. Cada segmento da mensagem deve ser gravado com a respetiva codificação de $< CR >$ indicando assim o final de um segmento;

- Em todas as etapas existe a possibilidade de erro ou perda da ligação com o cliente foram então criadas funções Python que fazem o aviso para despoletar os mecanismos de recuperação do sistema;
- O agente não estabelece conexão direta com a base de dados, essa função está a cargo do controlador geral;
- Agente servidor deve ser parado manualmente.

A utilização de métodos de *multiprocessing* prende-se com o aumento da capacidade de resposta a novos clientes que possam surgir. Este agente vai assim de encontro à topologia de motores de interface onde um agente pode receber mensagens para vários destinos sem criação direta de uma interface.

4.3.2 *Sistema Cliente*

Este tipo de agente tem como principal missão enviar mensagens para diferentes destinos mantendo diferentes origens sobre observação. Deve ainda apresentar propriedades como a autonomia, a disponibilidade e a capacidade de enviar diversas mensagens para diversos destinos. Assim sendo, as tarefas/propriedades deste tipo de agentes são melhor demonstradas na forma de lista:

- Cada agente inteligente é implementado numa máquina com acesso à *Internet* ou *Intranet*, na forma de *Python script*;
- Quando da sua configuração um nome do agente deve ser estabelecido no script de instalação;
- Através do nome do agente, é possível aceder ao ficheiro *info.xml* para recolher informações como:
  1. Nome do programa ou programas de origem da conexão;
  2. Nome do programa ou programas de destino da conexão;
  3. Nome do agente de destino;
  4. Porta em que o *socket* do agente servidor recebe conexões;
  5. IP em que o *socket* do agente servidor recebe conexões;
  6. Porta em que o controlador local está a receber conexões.
- Utilizando o nome do programa de origem são criadas pastas onde este deve colocar as mensagens em formato .hl7;
- Cada agente servidor apresenta um IP geralmente o da máquina onde esta inserido;

- O identificador único do agente cliente é constituído pelo nome do agente, mais o tipo (c neste caso);
- Um agente cliente utiliza a lista recolhida do seu ficheiro de configuração com vista a criar uma *multiprocessing pool* com o intuito de vigiar as pastas onde os programas de origem colocam as mensagens procedendo de imediato ao envio das mesmas;
- É criada uma nova *thread* para fazer o envio de cada mensagem;
- Quando existe uma mensagem para ser enviada, o ficheiro é aberto utilizando a função Python *.read()* e fazendo o *split()* do conteúdo, ficando assim com diversos segmentos da mensagem em formato HL7;
- Utilizando o segmento MSH (primeiro da mensagem), são extraídas informações como:
  - Tipo de mensagem;
  - ID da mensagem;
  - Tipo de ACK;
- O processo de envio de uma mensagem só termina quando é recebido o ACK da mensagem;
- Existe uma estreita comunicação entre cliente e controlador local para a troca de mensagens e avisos;
- Quando é perdida a conexão com determinado servidor, o agente servidor comunica com o controlador local para desencadear o processo de recuperação, sendo a mensagem em questão enviada de novo partindo do principio;
- O agente não estabelece conexão com a base de dados apenas o controlador geral efetua tal interação;
- Agente servidor deve ser parado manualmente.

Por motivos de aumento da capacidade de resposta e diminuição do tempo total até conclusão de um pilha de mensagens é introduzido então o conceito de *multiprocessing*. Observando as propriedades e tarefas apresentadas anteriormente, é possível observar as diretrizes básicas do motor de interfaces.

4.3.3 *Controlador Local*

Este tipo de agente tem como missão controlar e reunir informações de performance de agentes a operar numa determinada máquina. A ligação e comunicação constante com o

controlador geral efetua-se mais uma vez utilizando *socket* implementados em Python. As propriedades/tarefas deste tipo de agentes são melhor demonstradas na forma de lista:

- Agente é implementado na forma de script em Python;
- Quando da criação deste agente são lançados 4 processos distintos com diferentes funções:

  N_COMMUNICATIONS: comunicações normais com agentes. Recebe indicações de inícios de sessão , mensagens enviadas e recebidas e ainda ACK emitidos. Devem no final comunicar ao controlador central todas as ações desenvolvidas e conteúdo da mesma;

  S_PURPOSES: recolha de dados estatísticos dos seus agentes utilizando a extensão *psutil*. Todos estes dados são comunicados ao controlador geral em ciclos de dois minutos. Recolhe também informações sobre quantidades de mensagens em lista de espera para envio recorrendo à função *listdir*;

  E_ORDERS: executar ordens do controlador geral como reiniciar agentes ou alterar portas por exemplo. Quando existe alteração de alguma propriedade do agente o ficheiro xml deste deve ser alterado e depois criado o processo correspondente com novas diretrizes, todos os envolvidos no processo devem ser então alterados não ocorrendo assim erros. No caso de encerrar ou lançar um agente, são utilizadas as funções *os.kill()* ou *subprocess.Popen()* passando nome como atributo;

  EMERGENCY: para receber contactos de agentes em caso de erros, posteriormente transmite informações ao controlador geral esperando pela resposta, passando geralmente pelo *reboot* do processo;

- Todos os processos do pontos anteriores, apresentam *socket* numa determinada porta atribuída na configuração inicial.

Este processo serve essencialmente para aliviar a carga relativa à comunicação com o controlador geral. Por outro lado esta solução utiliza apenas comunicação via *socket* não sendo necessário recorrer a ligações *SSH* ou algo dentro desse esquema que poderia acarretar problemas com segurança a interna das máquinas.

4.3.4 *Controlador Geral*

Este agente desempenha papel de controlo e alimentação de bases de dados utilizadas posteriormente para alimentar a plataforma de monitorização. Um ponto importante de sistemas MAS prende-se com o controlo exercido utilizando uma ferramenta web desenvolvida para o efeito. Algumas das propriedades/tarefas deste agente podem ser agrupadas em formato de lista:

- Um agente deste tipo deve ser único controlando assim todas as operações;
- Quando é criado este agente, são lançados quatro processos:

  N_INTERACTION: este processo é destinado registar mensagens enviadas e recebidas com sucesso. Existe a interação com base de dados utilizando a extensão *cx_Oracle*.

  W_ORDERS: este processo mantém registo do estado atual dos agentes. Em ciclos de tempo pré estabelecidos faz QUERY à base de dados com intuição de saber se utilizador pediu mudança do estado do agente através da plataforma de monitorização. Caso seja alterado estado este processo tem de descobrir qual controlador deste agente e enviar mensagem ao processo que executa ordens;

  PROBLEMS: reúne informações quando existem erros de forma a despoletar o *reboot* dos agentes;

  PERFORMANCE: reúne informações de *performance* e faz o *insert* na base de dados;

  CONTROLLERS: mantém informações sobre controladores locais. Quando estes se iniciam enviam mensagem para este processo, este responde com lista de agentes sobre a sua "jurisdição".

Este processo apresenta certamente uma sobrecarga grande de comunicações e interações com a base de dados. Estas interações geralmente são foco de alguns erros nomeadamente má formatação de dados ou *timeout* de ligação.

4.3.5 *Comunicação entre Agentes*

Como abordado anteriormente, a comunicação entre agentes é importante. Existem dois pontos que devem ser abordados, o formato como mensagem é enviada através do *socket* e o conteúdo da mensagem enviada para que outro agente possa perceber (interoperabilidade semântica).

A comunicação utilizando *sockets* utiliza *strings* para enviar informações. Uma estrutura desta complexidade necessita de estruturas mais complexas como listas ou dicionários para armazenar dados. Recorreu-se à estrutura *JavaScript Object Notation (JSON)* que representa uma forma leve e fácil de *data-interchange* recorrendo à *syntax JavaScript* mas em formato de texto, ao exemplo do *Extensible Markup Language (XML)* [148]. O facto de ser *"self-describing"* e de fácil percepção motiva a sua grande utilização.

Para existir interoperabilidade semântica, é utilizada uma estrutura de lista da linguagem Python. Uma lista é um conjunto ordenado de valores representado por um índice que identifica inequivocamente um elemento dessa estrutura.

Para efeitos de compreensão entre agentes, a posição o de cada lista apresenta o tipo de mensagem que esta ser trocado, assim sendo:

CREATION é enviada pelos agentes cliente e servidor para o controlador local a quando do seu inicio;

LOGIN é enviado pelo cliente para o servidor iniciando assim a troca de mensagem entre ambos. O conteúdo exato desta mensagem enviada e é então:
" $[LOGIN, agent_name, origin_program, unique_id, pid_sub_process, time_of_server_connection, destination_program]$";

MESSAGE , é sobre este desígnio que se desenrola o envio do conteúdo da mensagem. Cada segmento de uma mensagem é então um pacote de dados enviado. Em todas as interações as posições são as mesmas, ou seja , o ID da mensagem na posição 4, o tipo de mensagem na 3, o tipo de ACK na 7 e o segmento na posição 1. Quando a mensagem na apresenta mais segmentos a posição 1 da lista apresenta o conteúdo "EOF" indicando assim o final do mesmo;

ACK , representa mensagem enviada pelo servidor para cliente contendo o *acknowledgement* de mensagem chegou ao destino e está em conformidade ou pelo cliente para o controlador local indicando que processo de troca da mensagem foi concluído;

PROBLEMS enviada pelos agentes indicando um erro, nesta mensagem segue a identificação dos intervenientes.

O controlador do sistema recebe mensagens dos diferentes tipos abordados anteriormente, mas necessita de recorrer a diferentes tipos para comunicar com diversos controladores locais como por exemplo:

SYSTEMKILL , utilizado para terminar um processo;

CPORT alterar a porta de um cliente;

SPORT alterar a porta de um servidor;

REBOOT reiniciar um agente.

Segundo a metodologia apresentada anteriormente, aspetos relacionados com a autonomia, o controlo de agentes e a comunicação ficam compridos. Um esquema geral das comunicações e processos adjacentes pode então ser apresentado na forma de um esquema presente na figura 13. Analisando esta, observa-se as interações pré-mensagem e pós-mensagem. Fundamentalmente existe uma interação com a base de dados quando o controlador local do cliente e servidor confirmam que a mensagem foi recebida em conformidade. Para além destas interações, existe também ainda a transmissão de dados relativos à *performance* dos diversos agentes que se encontram ativos. A mensagem é transmitida partindo dos controladores locais utilizando a classe *StatisticPurposes*, utilizando para tal um

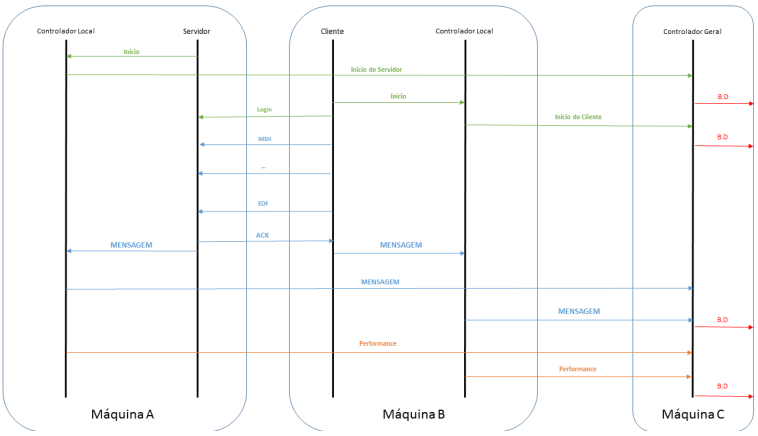


Figura 13.: Esquema geral das interações entre agentes do tipo HL7

dicionário, onde as chaves utilizadas são os identificadores únicos de cada agente. Como valores é utilizada um lista contendo nas diversas posições:

1. Bytes enviados;
2. Bytes recebidos;
3. Erros de envio;
4. Erros de receção;
5. Percentagem de CPU;
6. Percentagem de menória;
7. Percentagem CPU consumido para utilizador;
8. Percentagem CPU consumido pelo próprio CPU;
9. Estado.

A definição de um IP da máquina e porta em que o *socket* vai efetuar as suas operações é essencial para o bom desempenho do sistema. A criação deste sistema abre a necessidade de existir um bom sistema de monitorização capaz de controlar agentes e recolher informações que possibilitem uma apresentação do estado atual e o desempenho do sistema.

4.4 RESULTADOS E DISCUSSÃO

A implementação de um sistema multi agente acarreta um meticuloso trabalho de preparação e criação de estruturas de base. Toda a metodologia apresentada na secção anterior foi implementada utilizando a linguagem Python. O principal resultado a registar deste caso de

estudo foi a criação de uma plataforma que permite a correção de problemas da interoperabilidade sintática.

Os resultados experimentais alcançados foram:

1. *Scripts* em Python que representavam várias facetas propostas inicialmente;
2. Motores de interface que possibilitem assim o envio e receção de mensagens HL7;
3. MAS capaz de corresponder às necessidades de um sistema com características do ponto enunciado anteriormente;
4. Interface com plataforma de monitorização capaz de controlar todos os agentes envolvidos;
5. Sistema com capacidade de recuperar do erro correspondendo assim a especificidade de implementar um sistema *always available*;
6. Sistema multiagente com autonomia, capacidade de comunicação e devidamente controlado.

Para além de resultados experimentais, outros merecem ser abordados em detalhe:

1. Qual ou quais são as diretrizes *HL7 International* aplicadas à troca de mensagens na versão 2;
2. Qual é o segmento de uma mensagem HL7 que mais informações fornece ao agente relativa à mensagem que está a ser trocada;
3. Como pode ser aplicado um sistema HL7 para ligar sistemas de informação em unidades de saúde;
4. Quais são as características dos protocolos TCP/IP que mais favorecem este tipo de sistemas;
5. Qual ou quais são as principais funcionalidades de um motor de interface para mensagens HL7;
6. Qual ou quais são as características do Python que favorecem a implementação deste tipo de sistemas.

Relativamente ao primeiro ponto da lista anterior, nem sempre é fácil abordar uma implementação uma vez que cada situação representa um caso muito particular motivado por questões de meios económicos e físicos (rede e maquinas disponíveis). Quando as diretrizes abordam a questão do MLLP são um pouco latas não indicando especificamente qual o caminho exato que deve ser seguido.

A implementação alcançada segue então as diretrizes MLLP para utilizar os protocolos TCP/IP em conjunto com as *guidelines* propostas no desenvolvimento de um sistema de motores de interface. Inicialmente, este tipo de sistemas acarreta mais custos ao nível da criação da estrutura e delineamento da forma como os agentes devem comunicar visto estar a ser desenvolvido um sistema geral que tecnicamente processa um maior número de mensagens que o tipo *point-to-point*. Embora comunicar utilizando datagramas em formato TCP seja mais pesado, garante assim a sua chegada ao destino e um fator importante sem alteração da ordem dos mesmos. Mantendo a ordem dos pacotes garante-se assim que a mensagem HL7 chega segundo a ordem correta.

Ao desenvolver um sistema autónomo a intervenção humana torna-se dispensável. O sistema MAS desenvolvido apenas precisa que sejam devidamente configuradas as diversas portas relativas aos *sockets* e os nomes dos diversos agentes. A configuração de cada agente deve ser realizada no momento da implementação, caso a caso.

O sistema poderia existir e funcionar perfeitamente sem existir o *parse* da mensagem HL7 proporcionando assim menor tempo de processamento da mesma. Esta abordagem cairia por terra quando existisse um erro, uma vez que seria quase impossível saber quando este ocorreu e como seria possível recuperar. O segmento MSH como abordado anteriormente representa o segmento mais importante para a recolha de informações. As mais importantes são o tipo de mensagem, o tipo de ACK que mensagem pede e o ID da mensagem. Se a mensagem necessitar de ACK o sistema só termina o processamento a quando do recebimento da mesma, caso tal não seja necessário ou tal só ocorrer em caso de erro o sistema pode quase de imediato prosseguir para o próximo caso.

A implementação de uma interface deste tipo não é direta existindo um grande número de testes antes da sua implementação em ambiente de produção. Atualmente encontra-se em testes laboratoriais com vista a melhor a fase de controlo sobre agentes.

4.5 CONCLUSÃO

Cada vez mais as unidades de saúde utilizam sistemas de informação que necessitam de partilhar dados capazes de gerar informação. Um entrave a esta nova vaga de implementações prende-se com a criação de sistemas de informação de forma individualizada para setores específicos das unidades de saúde. Estes sistemas não necessitavam de partilhar informação com os demais levando a ilhas de informação e por vezes a repetição e contradição de informação. Com introdução de apoios por parte de diversos quadrantes para a fomentação da interoperabilidade surgiram soluções interessantes nesta área de investigação. A interoperabilidade sintática assume papel preponderante nesta nova vaga de sistemas, estando posteriormente, na base da interoperabilidade semântica.

A troca de mensagens utilizando as normas HL7 na versão 2 para troca de mensagens, são assim um dos meios mais utilizados a nível mundial para a partilha de informação em ambiente clínico, sendo imperial a existência de soluções e interfaces consistentes que possibilitem tal partilha.

Nos últimos anos a linguagem Python tem aumentado a sua popularidade entre comunidade cientifica, principalmente na área da engenharia. A linguagem confirmou os seus créditos, facilitando a implementação de sistemas *multiprocessing*.

A autonomia, a comunicação e o controlo são as mais importantes de entre um role bastante extenso. A autonomia é alcançada utilizando métodos como *psutil*, *socket*, *threading*, *time* em conjunto com uma base sólida de comportamento dos sistemas. Não existe muita informação sobre a estrutura de agentes HL7 talvez motivado, pelas diferenças de implementação existentes. Cada unidade de saúde representa um caso especifico motivado por razões económicas e de recursos.

A comunicação entre agentes compreende que os agentes sejam capazes de descodificar a mensagem. Assim sendo optou-se pela utilização do formato JSON codificando assim estruturas mais complexas que simples *strings*. Essas estruturas são listas de objetos, como por exemplo *strings*, dicionários ou até outras listas.

O MAS implementado foi direcionado para a criação de um motor de interfaces HL7. Os primeiros sistemas ligavam diretamente um HIS de origem com outro de destino motivando uma abundância de sistemas considerável. Neste sistema quando é necessário levar a cabo algum tipo de manutenção a complexidade é enorme. Utilizando um motor de interface, um único consegue enviar mensagens de diversos programas de origem, reduzindo os custos de manutenção à *posteriori*.

Embora o sistema ainda não esteja em ambiente de produção, está em fase de testes laboratoriais principalmente na componente de controlo de agentes utilizando uma plataforma de monitorização.

Conclui-se então que a importância de mensagens hl7 é inequívoca. O sistema desenvolvido fornece ao Centro Hospitalar do Porto um MAS capaz de ser controlado através de uma plataforma Web capaz, com capacidade de de recolher informações que fomentem o *Bussiness Intelligence*. A implementação deste caso de estudo cria a necessidade de um sistema de monitorização e controlo de agentes desenvolvidos neste caso de estudo, assim como de corresponder às necessidades dos agentes já implementados no *Centro Hospitalar do Porto (CHP)*.

De acordo com método *Design Research*, as necessidades levantadas inicialmente estão cumpridas, falta apenas o *proof-of-principles* (ver capítulo **??**). Embora não volte a entrar no ciclo de desenvolvimento, este caso de estudo permite o input para um novo caso de estudo. As etapas relativas à escolha de melhor solução e implementação, foram sendo respondidas tanto na metodologia como nos materiais.

Futuramente a actualização dos agentes pode ser efetuada via *DLL* por exemplo, assim como o seu método de comunicação melhorado.

Por último, como demonstrado no capítulo do estado de arte, na secção sobre a nova revolução industrial ou indústria 4.0, os agente implementados, através da utilização de controladores locais, transmitem o seu estado aos supervisores, criando um sistemas ágil e capaz de recuperar de erros autonomamente. Ao comunicar quantidades de processador a serem gastas, pacotes de rede enviados por máquina está a disponibilizar informações vitais sobre desempenho, contribuindo assim para a criação de *Cyber Physical Systems*.

5

CASO DE ESTUDO - SISTEMA DE MONITORIZAÇÃO DE AGENTES AIDA E HL7

Este capítulo destina-se a apresentar um caso de estudo que envolve a apresentação de informação ao utilizador no formato de sistema de monitorização. Existem no Centro Hospitalar do Porto duas necessidades que devem ser suprimidas, em primeiro lugar, é necessário um sistema que permita visualizar diversos aspetos dos agentes que fazem o envio e a receção das mensagens em formato HL7, em segundo, o sistema AIDA envia mensagens de aviso via SMS para utilizadores credenciados, é então importante conceber um sistema que permita o acesso a esses avisos utilizando uma aplicação web. A somar aos pontos anteriores, do capítulo anterior ficaram em aberto duas questões, o acesso ao conteúdo das mensagens a serem trocadas (*message query*) e ainda o controlo e monitorização dos agentes criados.

Assim sendo, ao longo do presente capítulo é apresentada a solução para as necessidades supra mencionadas, começando por uma introdução e o enquadramento no tema, seguindo-se a metodologia adoptada, posteriormente os materiais utilizados, os resultados e discussão e para terminar a conclusão do trabalho com apresentação de algum trabalho a desenvolver no futuro.

5.1 INTRODUÇÃO E ENQUADRAMENTO NO TEMA

Monitorizar significa supervisionar atividades em progresso assegurando assim que estas se encontram de acordo com os padrões estabelecidos na projeção inicial [149]. Pode portanto ser adaptada esta definição ao mundo informático utilizando para tal monitorização os meios eletrônicos à disposição. Naturalmente, a apresentação de informação é feita nos tempos que correm com recursos a aplicações Web acessíveis através de um navegador de páginas web [149]. No início eram utilizadas simples páginas *Hyper Text Markup Language (HTML)* com uma interface bastante básica, mas atualmente, desenvolver uma aplicação *Web* é muito mais do que criar uma página na referida linguagem. Sugiram nos últimos anos *frameworks*, que simplificam o desenvolvimento de aplicativos, devendo para tal ser escolhida aquela que mais se adequa às necessidades e requisitos existentes. É importante ter

em consideração o aspeto visual da aplicação, que em determinados casos resulta na não adoção desta. O acesso aos dados no denominado back-end da aplicação é naturalmente um ponto crítico devendo ser consideradas todas as possibilidades. Estes dados com vista a alimentar a página de dados fidedignos (evitando informação repetida ou contraditória), são processados de forma a "'limpar" os mesmos de dados incorretos.

Como referido anteriormente os conceitos provenientes da nova revolução industrial servem como fonte de ideias que podem ser aplicadas no sector da saúde. A implementação de sensores físicos ou *scritpts* com o objetivo de recolher dados sobre o meio envolvente é sem dúvida um caso a ter em consideração. Estes dados são posteriormente enviados, via Web Service para unidades centrais de processamento que os transformam em informação sobre o estado de máquinas e processos. Existem outros trabalhos interessantes, Scott W. Doebling apresenta *"Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review"* onde aplica os conceitos supra mencionados na deteção, localização e na caracterização de danos estruturais utilizando para tal, aspetos relacionados com a resposta à vibração [150]. No dois casos, a informação gerada é apresentada ao utilizador no formato sistema de monitorização ou plataforma de *reporting*, geralmente contendo módulos de *bi*, e estruturas básicas de apresentação de informação (tabelas, listas), que ajudam na tomada de decisão o mais informada possível.

Segundo a patente *"Health Monitoring System"*(US6282441 B1) do ano de 1995, um sistema de monitorização em saúde faz o registo do estado de saúde do paciente e compila de forma cronológica todo o histórico deste [151]. Utiliza para desempenhar tais tarefas um sistema multi sensor que recolhe informações e comunica estas de forma a existir um processamento posterior [151]. Uma observação importante presente na patente é que os dados recolhidos não são diretamente relacionados com a condição médica do paciente, ao invés fornece informação necessária para retirar padrões característicos de pacientes saudáveis e daqueles que apresentam patologia [151]. A aquisição de dados envolve a seleção do tipo de sensores e qual ou quais os locais onde estes devem ser inseridos, assim como o meio de transmissão e armazenamento dos dados recolhidos. Existem um número significativo de trabalhos relacionados com a aquisição de dados, nomeadamente utilizando fibra ótica como no trabalho de Honglei Guo e seus colaboradores intitulado *"Fiber Optic Sensors for Structural Health Monitoring of Air Plataforms"* [152].

Embora exista alguma informação relativa aos pontos referidos no capítulo anterior, o mesmo não se sucede quanto à a recolha e apresentação de informação relativa à *performance* de agentes inteligentes. Luciana Cardoso e seus colaboradores, em *The Next Generation of Interoperability Agents in Healthcare*, demonstra a implementação de agentes capazes do envio e receção de mensagens *Health Level Seven International (HL7)* que posteriormente

são avaliados ao nível da *performance* utilizando um sistema de monitorização bastante rudimentar [35].

Antes de partir para implementação de um sistema de monitorização é necessário ganhar *knowledge* sobre os processos em questão. Focando atenções no caso da troca de mensagens em contexto hospitalar, a companhia de desenvolvimento de *software ExtraHop Network Inc.*, apresentou em 2014 a *ExtraHop Plataform* [153]. Marcando a diferença em relação às demais, com uma plataforma interativa fluída e apelativa ao invés das demais, com funções bastante limitadas. A plataforma utiliza dados provenientes de *sniffers* instalados na network ao em contraponto aos tradicionais registos de log por parte de agentes ou controladores [153]. A plataforma processa os dados com o objetivo de operadores detetarem fontes de possíveis erros o mais rápido possível. Algumas das funcionalidades da *ExtraHop Platform* são [153]:

- Contabilidade do total de mensagens;
- Erros descriminados por aplicação, e ainda a possibilidade de vizualizar o conteúdo da mensagem;
- Mensagens duplicadas;
- Tempos de integração e ACKS;

Estas são apenas algumas funções de uma ferramenta moderna de monitorização de atividade HL7.

Segundo Mike Stockemer na publicação *"Monitoring and Alerting for HL7 Interfaces"*, atualmente a monitorização de interfaces é feita de um modo rudimentar [154]. Daí que o autor defenda que sistemas como o ExtraHoop e aquele ao qual este caso de estudo se propõem sejam fundamentais para uma unidade de saúde. O autor aponta também alguns pontos que considera fulcrais na monitorização de interfaces HL7:

- Mensagens entregues e confirmadas;
- ACKS vs. NACK;
- Mensagens sem resposta;
- Estado de conexões;
- Tempo de início;
- Tempo de fim;
- Erros no processamento de mensagens;
- Conteúdo de mensagem invalido;

- Conexões a base de dados;
- Informações sobre máquinas onde estão inseridas interfaces.

Com vista a gerar dados que respondam aos pontos da lista anterior, tem de existir a implementação prévia de um sistema de *data warehouse/business intelligence* contendo dados processados, uma vez que estes são recolhidos em condições bastante dispares. Bill Inmon define *data warehouse* como, *"subject- oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions"* [155]. A referência ao *time-variant* é deveras importante, num contexto como o do presente caso de estudo, onde a informação varia todos os dias e onde um sistemas capaz de reconhecer isso é deveras importante.

Utilizadores de sistemas DW/BI recorrem a soluções otimizadas para elevada *performance* nas *queries* uma vez devem ser capazes de processar milhares e milhares de dados de forma a fornecer a resposta correta. Uma questão que engrandece ainda mais estes sistemas, está relacionado com o histórico de dados que estes utilizadores quase sempre necessitam [156]. Os objetivos principais do DW/BI são [156]:

CRIAR INFORMAÇÃO ACESSÍVEL: tornar a informação percetível e intuitiva para todos os utilizadores e não apenas para utilizadores especialistas. As ferramentas de *Business Intelligence (BI)* utilizadas devem ser simples e fáceis de utilizar. O tempo de espera pelos resultados devem ser mínimos;

GERAR INFORMAÇÃO CONSISTENTE: utilizando dados credíveis, montados cuidadosamente a partir de diferentes fontes, trabalhados posteriormente para gerar informação que só deve ser apresentada quando estiver de facto devidamente finalizada;

CAPACIDADE DE ADAPTAÇÃO À MUDANÇA uma vez que o ambiente onde está inserido está em constante mudança deve ser capaz de se adaptar;

GERAR INFORMAÇÃO A TEMPO E HORAS: a conversão de dados em informação deve ser realizada quase ao segundo, proporcionando ao utilizador a obtenção de informações vitais para a tomada de decisão.

Desde a primeira edição *The Data Warehouse Toolkit* em 1996, Kimball definiu uma lista completa de técnicas para modelação de dados de forma dimensional [156]. Apresentando ainda quatro decisões que devem ser tomadas durante o *design* de um modelo dimensional [156, 157]:

1. Selecionar um processo de negócio;
2. Identificar a granularidade;
3. Identificar as dimensões;

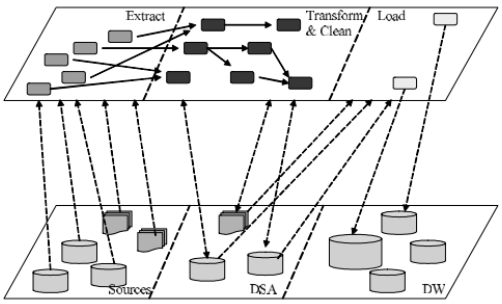


Figura 14.: Ambiente dos processos ETL (adaptado de [158]).

4. Identificar os factos.

Os pontos da lista anterior só podem ser corretamente respondidos se existir um profundo conhecimento das intenções da organização, da forma como os dados são obtidos e qual o objetivo principal do sistema a ser implementado. Compreender os dados envolve saber quais aqueles que são necessários para corresponder a determinada necessidade, envolvendo dimensões e factos para tal. Por outro lado, a granularidade vai identificar em parte o estado em que estes dados estão. A utilização de dimensões visa enquadrar e aumentar a interoperabilidade semântica dos dados obtidos. Deste último ponto pode ser exemplo, enquadrar o estado de um agente numa categoria de 3 erros possíveis (sem erro, probabilidade média de erro, erro iminente).

Com vista a facilitar e manusear o processo de *data warehouse* utilizam-se ferramentas *Extraction-Transformation-Loading*. De entre as funcionalidades destas ferramentas destacam-se [158]:

1. Identificação de informação relevante na fonte;
2. Extração da informação selecionada anteriormente;
3. Processamento da informação proveniente de diversas fontes para um formato comum;
4. Limpeza do *data set* em conformidade com regras pré-estabelecidas;
5. Disseminação dos dados para *data warehouse* e/ou *data marts*.

Na figura 14 é descrita de forma abstrata os processos de *Extract, Load, Transform (ETL)*. Na parte inferior estão colocados os *data stores*, sendo aqueles à esquerda na figura, os fornecedores de dados originais (bases de dados e ficheiros). Após extração (parte superior esquerda), os dados são processados e limpos (eliminar repetições e contradições) numa área chamada *Data Staging Area (DSA)*. No final deste processo são então carregados para o *data warehouse*(canto superior direito) [158].

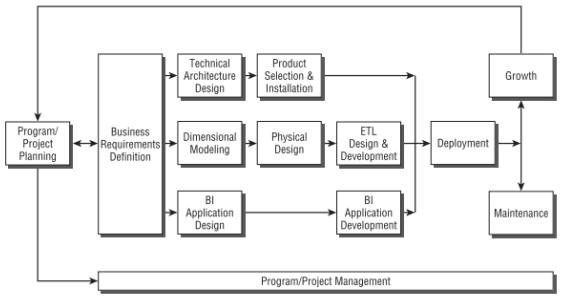


Figura 15.: Diagrama do Ciclo de Kimball (adaptado de [156]).

A primeira fase (extração), apresenta elevada complexidade visto lidar com dados bastante heterogéneos, provenientes de diferentes fontes proporcionando assim dados para a próxima etapa. O processamento de dados é a parte central do processo de ETL. Neste ponto, divergências semânticas e sintáticas são superadas utilizando componentes especializados como [156]:

- Utilização de *mapping technology*;
- Técnicas de *data scrubbing and cleaning* para normalização de dados;
- Utilização de componentes de algebra relacional para agregar e combinar *data sets*.

Consolidar dados numa única fonte física de dados provou ser a abordagem mais eficiente para apresentar informação de forma rápida e com elevada disponibilidade que proporcione acesso a informação relevante [157]. Posto isto, a informação armazenada pode ser utilizada em sistemas que apresentem a informação ao utilizador final.

Kimball DW/BI Lifecycle Overview cobre toda as atividades que ocorrem durante o processo de implementação de um projeto desta natureza, servindo de guia para a iniciação na implementação de *data warehouse e business intelligence* [156]. Este ciclo demonstrado na figura 15 foi proposto em 1998 utilizando o nome *Business Dimensional Lifecycle*. A figura demonstra a sequência de tarefas, dependências e concorrência adjacentes a este processo. Embora as janelas sejam de dimensão semelhante o consumo de recursos é muito superior em alguns pontos do que noutros.

O ciclo de Kimball começa com *program/projet planning*, onde são recolhidos os requisitos iniciais da empresa/instituição para o projeto *Data Warehouse (DW)* e BI [159]. Segue-se *business requirements definition* existindo uma relação bidirecional entre o primeiro ponto e este, sendo o seu entendimento absolutamente crucial. A parte superior da figura, aborda questões do foro mais técnico como a arquitetura do sistema e a integração de diferentes tecnologias [159]. A definição de uma arquitetura influencia todo o restante processo indicando qual ou quais as ferramentas a ser utilizadas. O segmento do meio da figura provém

diretamente dos requisitos de negócio com foco nos dados [159]. Os requisitos são primeiramente traduzidos para um modelo dimensional e posteriormente transformados em estruturas físicas. O culminar é então o desenho e a implementação do processo de ETL, o processo que mais recursos exige em todo o esquema [159]. Por último mas não menos importante, está a projeção e a implementação de aplicações de BI . Sem este componente que envolve a apresentação de informação, a maior parte dos requisitos definidos inicialmente não serão cumpridos [159].

Abordando agora questões relativas à apresentação de informação, o modelo *Model/View/Controller (MVC)* é uma forma bastante útil de projetar *software* interativo separando a interface dos dados [160]. As propriedades e características deste tipo de sistemas foram abordadas no capítulo refestadoarte.

Uma aplicação web é um sistema informático projetado para a utilização através de um *browser* com acesso à Internet, que utiliza linguagens como HTML JavaScript e CSS. Esta aplicação pode ser executada a partir de um servidor *Hypertext Transfer Protocol (HTTP)* (*Web Host*) ou localmente [160]. A diferença entre este tipo e uma simples página web, reside no facto de neste cenário ser necessário o processamento do lado servidor antes de enviar a página para o destinatário. Construir uma aplicação web é um processo complexo que consome bastante tempo e recursos humanos na percepção do domínio envolvente (objetos, comportamentos ou regras) e na concepção de uma interface sólida com uma sistema de navegação simples e intuitivo [161]. Outro aspeto que aumenta a complexidade das aplicações web prende-se com a existência de diferentes perfis de utilizador, apresentando assim acesso a diferentes componentes. A evolução da aplicação deve ainda assim proporcionar uma implementação modular que facilmente possa ser alterada e remodelada sem ser necessário alterar toda a aplicação.

As *Single-page-applications (SPAs)* distinguem-se das demais pela capacidade de redesenhar parte da interface sem necessitar de refazer o pedido HTTP para apresentar um conteúdo diferente de HTML [162]. Para conseguir tal feito existe uma separação entre dados e interface utilizando o um modelo (estrutura MVC abordada) [162].

A aplicação da *framework*, em soluções ligadas às unidades de saúde, pode ser observada no trabalho, *"Improving Healthcare Team Collaboration in Hospital Transfers through Cloud-Based Mobile Systems"* da autoria de Andres Neyem e seus colaboradores, onde aborda a questão da monitorização de trocas de pacientes entre unidades hospitalares e as causas destas nos custos e na qualidade de prestação de serviços [163]. Quanto à implementação de SPA em questões relacionadas com saúde, Lewis E. Kazis apresenta *"The Veterans SF-36 Health Status Questionnaire: Development and Application in the Veterans Health Administration"*, pretende implementar um sistema de controlo de pacientes com base em questionários com intenção de poupar custos de tratamento [164]. Embora não seja precisamente um sistema

de monitorização que utilize sensores ou outro sistema avançado permite obter algumas noções sobre a apresentação de informação em aplicações de uma única página.

Ao longo do presente caso de estudo são então apresentadas as etapas e o impacto final de um sistema de monitorização de agentes inteligentes. Este caso de estudo está assim enquadrado no âmbito da disseminação de informação e subjacente interoperabilidade e ainda na utilização de sistemas que possibilitem a tomada de decisão utilizando informação precisa e assertiva. Analisando a presente secção primeiro demonstrou-se as ligações da indústria 4.0 e outras ideologias semelhantes na coleta de dados. Explicou-se de forma detalhada o método de Kimball para recolha e processamento dos dados recolhidos e no final abordou-se de forma simplista uma vez que já havia sido abordada em capítulos anteriores a utilização de *Single-page Application* como parte dos modelos MVC.

5.2 MATERIAIS

Ao longo da presente secção serão abordados os materiais utilizados no desenvolvimento do presente caso de estudo. O motivo da escolha deste e não outros com funções similares pode ser observado na secção relativa à metodologia. Os materiais utilizados foram:

- Ao nível do desenvolvimento do processo de ETL optou-se pela pela linguagem Python na versão 3.4 com o *plugins* cx_Oracle;
- Base de dados são Oracle apenas com criação de utilizadores definidos para este propósito;
- Plataforma web desenvolvida em Angularjs versão 1.3.4 com *plugins*:
  - angular-bootstrap;
  - angular-nvd3;
  - angular-route;
  - angular-sanitize;
  - angular-toastr;
  - bootstrap;
  - jquery;
  - satellizer, nvd3;
  - Chart.js;
  - oclazyload.
- Servidor RESTFUL que serve como back-end da aplicação desenvolvido na linguagem Php;

- Utilização do método *Design Research* para realização do caso de estudo;
- Utilização do método do Ralph Kimball para o processo de ETL.

5.3 METEDOLOGIA

Ao longo da presente secção serão abordados diferentes aspetos relacionados com a metodologia e as etapas que levaram à implementação final desta ferramenta na unidade de saúde. A melhor forma de demonstrar o trabalho efetuado é dividindo em sub secções, afetos a cada componente do sistema.

Seguindo a metologia *Design Research* o primeiro ponto começa com a análise detalhada das necessidades que atualmente existem no Centro Hospitalar do Porto relativas à partilha e apresentação de informação. Sendo assim, pode ser apresentadas na seguinte lista:

- Meio para monitorizar agentes AIDA e HL7;
- Meio para controlar agentes HL7 desenvolvidos no capitulo anterior;
- Disponibilização de plataforma de BI relativa ao trabalho de agentes;
- Aplicação que utilize as novas *framework* existentes no mercado nomeadamente Angularjs.

Um sistema de controlo e monitorização de agentes não precisa forçosamente de ser aplicação Web, principalmente no início deste tipo de sistemas, o controlo era feito através de um script correndo na linha de comandos. Essa é uma opção que embora rudimentar deva ser considerada, pela sua leveza e facilidade de execução, consegue responder aos dois primeiro itens da lista anterior. Existe outra possibilidade, desenvolver uma aplicação utilizando a linguagem Java em conjunto com *Jswing* para desenvolver a interface mais interativa e funcional que consegue dar resposta aos três primeiros itens da lista. Como um dos requisitos da unidade de saúde era utilizar o navegador de Internet estas soluções devem ser postas de parte.

Atualmente desenvolver aplicações de Internet exige mais do que apenas trabalhar em HTML. Nós últimos tempos com a ascensão das linguagens PHP e JavaScript novas alternativas surgiram neste campo que merecem consideração. As linguagens supra mencionadas começaram por ser utilizadas em conjunto com HTML conseguindo criar assim páginas mais interativas e dinâmicas. Em parte esta evolução deve-se à utilização de Bootstrap e outras *scripts* idênticas como é o caso de CSS. Embora para utilizadores iniciantes esta abordagem seja mais simples, posteriormente na manutenção ou renovação de algo relacionado com o código fonte, revela uma dificuldade acentuada. Surgiram então as *frameworks* que simplificam a vida do programador, aquelas que ultimamente se têm destacado das

demais são o Django com base na linguagem de programação Python e o AngularJS fortemente ligado ao JavaScript. Embora ambas apresentem a base no conceito MVC, por dificuldades de compatibilidade com alguns sistemas de segurança da unidade de saúde, em implementações anteriores optou-se assim pelo Angularjs.

Comparar *frameworks*, livrarias ou extensões que utilizem *JavaScript* hoje em dia é um processo simples, motivado pela grande quantidade de informação disponivel *online*. Num artigo de opinião de Jeremy Likness intitulado *"10 Reason Web Developers Should Learn AngularJS"* do ano de 2014, o autor apresenta o seu ponto de vista enquanto programador de quais são os fatores que tornam esta *framework* atrativa. *Angularjs* simplifica a ligação entre a interface (vista) e o local onde os dados de facto estão armazenados (modelo), utilizando por exemplo a diretiva *ng-model* não sendo necessário declarar explicitamente as propriedades ou dependências. Embora existam alternativas que abordam de forma mais correta, implicam a passagem de dados para um objeto para depois ser utilizado no modelo [165]. O facto de lidar com dependências como quase nenhuma *framework* simplifica o teste das aplicações e a criação de *Single Page Applications* [165]. No primeiro ponto a aplicação pode ser divida em módulos que embora dependentes podem ser iniciados em separado aumentando assim as capacidades de teste. No segundo ponto utilizando o carregamento dinâmico permite obter aplicações quase nativas partindo de aplicações com base na web.

O JavaScript executa pedidos HTTP disponibilizando os dados que constituem o modelo, estes pedidos podem ser executados diretamente no controlador ou utilizando *services ou factories* aumentando assim a reutilização do código. Este pedidos são realizados a um servidor em que constitui assim o *beck-end* de toda aplicação, neste caso do tipo RESTFUL implementado na linguagem PHP. Os dados são assim selecionados por meio de *queries* e enviados para os controladores no formato *JavaScript Object Notation (JSON)* , dando posteriormente lugar ao modelo.

Como abordado anteriormente, é necessário um servidor capaz de responder aos pedidos do navegador para apresentar o conteúdo da aplicação e ainda um servidor do tipo *RESTFUL* (ver capítulo 2). Relativamente ao primeiro caso a questão pode ser resolvida utilizando Apache2, disponibilizado em computadores UNIX de origem ou emulado em *Windows* utilizando XAMPP.

Apache HTTP Server foi utilizado na implementação no Centro Hospitalar do Porto uma vez que as máquinas em produção nessa unidades utilizam a distribuição Ubunto do sistema Linux. Este servidor web é atualmente o mais utilizado em todo o mundo baseado no servidor *NCSA HTTPd* datado do ano de 1995. Algumas das propriedades mais importantes deste servidor são:

- Carregar modelos dinâmicos;
- Processamento de *Multiple Request Processing Modes (MPMs)* incluindo *Event-base/Async*;

- Processar ficheiros estáticos ou indexados;
- *Reverser Proxy*;
- Protocolo HTTP;
- Reescrita de URL.

O caso de *XAMPP* é um pouco diferente, é uma plataforma de distribuição livre multi-plataforma. Consiste basicamente num servidor Apache HTTP, uma base de dados e interpretadores para a linguagem Php e Perl. O nome deriva de *Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)*. Enquanto no caso anterior tudo deve ser implementado em separado, neste caso tudo esta junto numa única ferramenta.

O servidor *restuful* como abordado em 2, foi desenvolvido em PHP. A escolha da linguagem prende-se com a fácil implementação e interpretação utilizando Apache2. A linguagem PHP utiliza conetor o especifico que permite efetuar *queries* as base de dados. O OCI8 *Oracle Call Interface* consiste num conjunto de API na linguagem C que proporciona uma interface para as bases de dados Oracle.

Desde o início da secção foram descritas as primeiras duas etapas do *Design Research*, com o levantamento de requisitos e posterior análise de soluções para encontrar aquela que mais favorece o Centro Hospitalar do Porto. Até o final da secção é apresentada em detalhe a metodologia utilizada no processo de ETL e na criação da aplicação web.

5.3.1 *Processo de ETL*

Os dados desempenham papel preponderante sendo necessário implementar um processo de ETL para depuração destes. A linguagem utilizada para tal foi o Python em parte motivado pela grande facilidade de implementação e uma vez que nas máquinas já existiam todos os conectores necessários (*cx_Oracle*) não era necessário trabalho adicional. Mas também pelo facto de ser utilizado um simples ficheiro que pode correr em qualquer máquina desde que existam as condições necessárias. Existem trabalhos que utilizam a linguagem Python como por exemplo *"pygrametl: A Powerful Programming Framework for Extract–Transform–Load Programmers"* de Christian Thomsen e Torben Bach Pederson que proporciona ferramentas que permitem o processo de ETL que justificam e testemunham as capacidades da linguagem para casos deste tipo [166].

Como abordado na contextualização teórica, o ciclo de Ralph Kimball (figura 15) é dos mais utilizados no campo do DW/BI sendo esses conceitos aplicáveis ao presente caso de estudo, formando assim uma das principais metodologias a ser implementadas. O processo de ETL envolve assim a extração de dados relativos a **agentes hl7**, **agentes aida**, **mensagens hl7** e **avisos**. Na seguinte lista, vão ser apresentadas as principais etapas assim como uma descrição de como foi adaptado ao seguinte caso de estudo:

PROGRAMAÇÃO E PLANEAMENTO DO PROJETO: como abordado anteriormente o processo de DW/BI tem como objetivo alimentar um sistema de monitorização utilizando para tal dados provenientes de uma base de dados Oracle. A monitorização HL7 compreende alguns pontos quase obrigatórios que foram apresentados no enquadramento do presente caso de estudo. Daí que o objetivo neste caso seja angariar o máximo possível desses pontos para posterior apresentação ao utilizador. O objetivo relativo a agentes *Agéncia para a Integração, Difusão e Arquivo de Informação Médica (AIDA)* passa pela deteção de possíveis erros e gerar avisos para o utilizador e ainda pela averiguação do estado do agente (ativo/inativo);

REQUISITOS DE NEGÓCIO: a aplicação compreende três módulos, respetivamente, agentes HL7, mensagens trocadas e agentes AIDA. No primeiro existem duas situações distintas, o controlo de agentes inteligentes desenvolvidos no caso de estudo anterior e ainda agentes que atualmente estão implementados no *Centro Hospitalar do Porto (CHP)*. No segundo ponto as mensagens enviadas e recebidas por agentes HL7 devem ser apresentadas com especial destaque para aquelas que o ACK tenha sido respondido com erro. O caso do módulo AIDA compreende a criação de avisos sobre o estado dos mesmos. Por imposição da base de dados de origem, apenas são armazenadas mensagens relativas aos últimos 3 dias.

LINHA DA TECNOLOGIA: a base de dados utilizada é do tipo Oracle. A linguagem escolhida neste cenário para o processo é Python, sendo essencial a existência de conectores (*cx_Oracle*) específicos para este tipo de linguagem. O *data warehouse* final também é do tipo Oracle não existindo assim necessidade de utilizar outro tipo diferente de conetor. A máquina de destino apresenta um *Sistema Operativo (SO)* Linux distribuição Ubunto daí que o Python seja uma boa escolha visto este já estar implementado de raiz;

LINHA DA DADOS: a maior parte dos dados existentes está em formato *varchar* ou *number*. Esta linha compreende 3 componentes, o modelo dimensional, o desenho físico e por fim o processo de ETL e seu desenvolvimento. No primeiro ponto deve ser implementado uma forma de tanto nos agentes HL7 como AIDA reconhecer que algo está errado com determinado agente. A melhor forma de o fazer é utilizando uma tabela com 3 dimensões (0 indica que o agente não tem problemas, 1 pode apresentar problema e por fim 2 representa um agente em falha). Cada agente tem uma coluna que indica a data da última mensagem, deve então ser enquadrado dentro de um espaço temporal a diferença entre a data atual e aquela encontrada neste campo. Existe então outra tabela que indica quais os processos que estão ativos durante a noite permitindo assim ao programa saber se esse espaço temporal elevado se deve ao processo apenas correr de dia ou de noite. Ao nível do desenho físico, este compreende dois *scripts* em

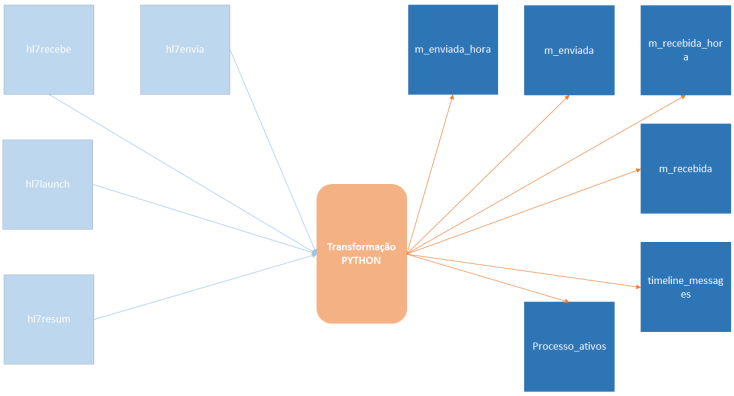


Figura 16.: Esquema geral do processo ETL modulo HL7.

Python que são lançados e encerrados de forma manual. O processo de ETL vai ser abordado em detalhe no decorrer desta sub-secção;

LINHA DA BI : este projeto, aplica este componente ligado a um número de mensagens trocadas por agentes no próprio dia e nos últimos dias. Esta informação é importante para quem trabalha no setor de informação indicando quais são os agentes que mais sobrecarga apresentam diariamente.

Separando agora a implementação física do processo de ETL entre agentes AIDA e HL7, o lado esquerdo da figura 16 apresenta as tabelas inicias onde é realizado o *extract* de dados relativo a este último. Sendo a parte direita da mesma ocupada pelas tabelas que constituem o *data warehouse* de destino que sustenta a plataforma de monitorização.

Começando pelo lado esquerdo da figura 16, é necessário abordar quais os campos mais importantes e qual a chave primária de cada tabela, assim sendo optou-se pela elaboração da seguinte lista:

HL7RECEBE: contém as mensagens recebidas por todos os agentes no últimos 3 dias. Cada linha é identificada pelo id da mensagem sendo as colunas mais importantes a origem, o destino, o tempo de receção e integração assim como se a mensagem foi integrada ou não. As colunas relativas ao erro do tipo hl7 e episodio são também campos importantes permitindo o último aceder ao conteúdo da mensagem;

HL7ENVIA: contém as mensagens enviadas por todos os agentes no últimos 3 dias. Cada linha é novamente identificada pelo id da mensagem sendo as colunas mais importantes a origem, o destino, o tempo de envio e receção do ack da respetiva mensagem assim como o campo do erro hl7 e novamente o episodio para aceder a determinada mensagem;

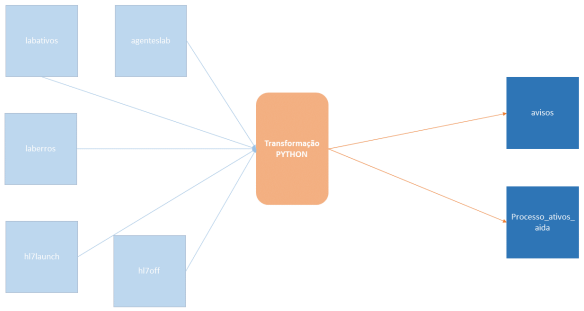


Figura 17.: Esquema geral do processo ETL modulo AIDA.

HL7LAUNCH: cada tuplo representa um agente implementado e devidamente configurado. A chave primária é formada por quatro campos, nome, origem, destino e tipo. Sendo este último servidor ou cliente. As colunas de interesse são o estado (Ligado ou Desligado) e ainda o tempo desde a última mensagem. De referir que alguns agentes só estão ativos durante o período noturno sendo que esta situação deve ser devidamente salvaguardada;

HL7RESUM: cada tuplo representa informações de *performance* de determinado agente implementado e devidamente configurado. A chave primária é formada por quatro campos, nome, origem, destino e tipo. Sendo este último servidor ou cliente. As colunas de interesse são o estado (Ligado ou Desligado) e ainda o tempo desde a última mensagem. São também úteis os campos como total de mensagens em lista de espera, número de *packages* enviados e recebidos, total de memória consumida pelo agente, total de erros de envio e receção, total de espaço de memória ocupado.

Uma questão importante no que as mensagens diz respeito, é o intervalo de tempo em que este ocorre, uma vez que os dados são armazenados apenas num espaço temporal de 3 dias. Assim sendo estipulou-se que uma vez por dia, depois da meia noite, o processo separa as mensagens do dia em questão e dos três anteriores. Durante o resto do dia o processo de ETL corre de 60 em 60 minutos apenas utilizando mensagens da última hora. Como o período da noite apresenta menor movimentação de mensagens HL7 é altura ideal para realizar este processo reduzindo assim o consumo de recursos computacionais durante o período diurno.

No caso dos agentes AIDA, o processo de ETL foca-se em agentes do laboratório e ainda agentes HL7. O objetivo neste caso é gerar avisos ao utilizador e ainda registar quais os processos que de facto estão ativos em determinado momento.

No caso de agentes laboratoriais, primeiro devem ser selecionados da tabela agentes_lab aqueles que estiverem em determinados servidores, sendo a chave primária o código que cada agente apresenta. Posteriormente para cada agente da lista anterior deve ser obtido

o tempo em atividade através da diferença entre data atual e data1 da tabela labactivos. Se a barreira dos 30 minutos for ultrapassada e não existir relatório de erro é porque alguma situação anormal aconteceu e gera aviso, caso contrário apenas indica que o agente terminou.

Estes avisos são então inseridos na tabela avisos, utilizando o tempo, o motivo que apresenta, um titulo e o conteúdo, qual o agente ou servidor a que se aplica e ainda a gravidade do erro segundo parâmetros definidos na definição inicial do projeto.

Todos os agentes são inseridos na forma de tuplo na tabela *processos_ativos_aida*. Neste caso a chave primária é o código definido anteriormente. Obviamente o estado e probabilidade de erro figuram também como campos importantes de cada tuplo.

No caso dos agentes diretamente ligados ao AIDA, estes devem ser monitorizados continuamente e sem interrupções. Assim sendo o script executa as etapas supra mencionadas em intervalos temporais de 5 minutos permitindo assim uma granularidade pequena para detetar erros ou situações anormais.

5.3.2 *Plataforma WEB*

A plataforma web utilizando a *framework* Angularjs deve ser intuitiva aproveitando todas as características e pormenores desta *framework* como *bootstrap e css*. Utilizou-se um *template* disponível online de distribuição livre que serviu como base para todo o trabalho elaborado. *sb-admin-angular* é uma adaptação do famoso *Free Admin Bootstrap Theme SB Admin v2.0* à *framework* Angularjs, servindo de base a toda a implementação. Alterações de fundo foram impostas nomeadamente o desenvolvimento de Web Service, assim como alterações de alguns aspetos relacionados com a interface. Um dos requisitos que esta *framework* apresenta é a utilização de *JavaScript, JQuery e PHP*. Após investigação dos navegadores de Internet presentes na unidade de saúde, conclui-se que não existiria problemas uma vez que no serviço de informação não é utilizado *Internet Explorer* distribuído pela Microsoft Corporation.

Como a aplicação acede à informação relativa a pacientes embora de forma indireta, deve ser capaz de proteger tais dados contra possíveis ataques do exterior. O esquema de utilização da aplicação esta presente na figura 18. Existe na Internet o conceito de *token-based authentication*. Basicamente através do nome de utilizador e *password*, se estes dados estiverem de acordo com o registado na base de dados, a sessão recebe um *token* que é válido por um determinado período de tempo. Assim sendo, sempre que este pretender aceder a um recurso durante esse período de tempo, pode faze-lo sem necessidade de apresentar novamente as credencias. Caso acesso seja rejeitado, o utilizador não é capaz de avançar aartir da página de Login.

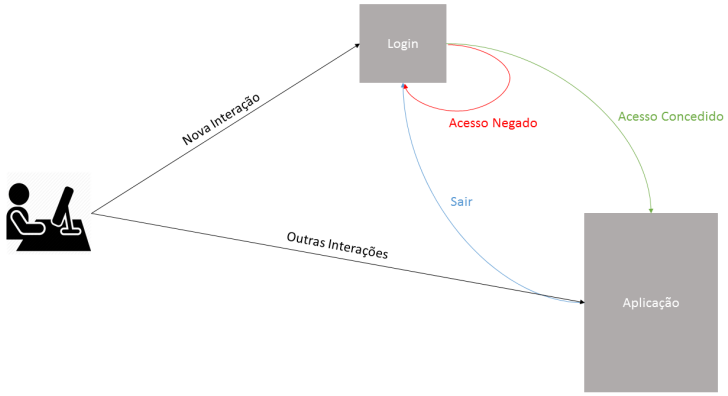


Figura 18.: Esquema de interação com a aplicação desenvolvida para o Centro Hospitalar do Porto.

As credenciais utilizadas para aceder à aplicação são as mesmas que os responsáveis utilizam para aceder à plataforma AIDA. Toda a implementação relacionada com a entrada na aplicação assim como operações relacionadas com o *routing* da aplicação está descriminado no ficheiro *app.js*. Assim sendo, para cada pedido URL este ficheiro sabe exatamente a qual ficheiro HTML se refere e ainda qual o controlador responsável. Nesse momento, o código do controlador é injetado na vista (HTML), permitindo assim um menor consumo de recursos por parte do navegador. Cada controlador tem assim um ficheiro separado que permite uma maior modularidade.

O módulo de BI e indicadores é implementado com recursos a diretivas externas. No modelo é chamada a diretiva onde é indicado o tipo de gráfico pretendido, assim como em que estrutura do modelo estão acessíveis os dados.

5.3.3 *Servidor RESTFUL*

O objetivo de implementar este servidor é aceder a dados em base de dados e realizar outras tarefas que são importantes para a solução encontrada. Em *Hypertext Preprocessor (PHP)*, para implementar um servidor tipo Restful são necessários no mínimo três ficheiros. O primeiro denomina-se o .htaccess para redirecionar pedidos URL, Rest.inc.php para efetiva criação do tipo Rest e ainda o ficheiro contendo o código desenvolvido intitulado api.php. Basicamente, o primeiro ponto da lista anterior utiliza o pedido URL geralmente extenso, e retira apenas a última parte deste, ficando geralmente com uma palavra. Posteriormente dentro da classe principal PHP é pesquisado se existe uma função com o mesmo nome do pedido e se tal for recusado ou inexistente devolve o código 404.

Quando é necessário aceder a informação contida em bases de dados, o servidor utiliza os conectores da linguagem Php denominado OCI8, uma vez que estes estão presentes numa base de dados Oracle.

A resposta para os controladores é processada em formato JSON quando tal acontece. Existem no acesso às bases de dados, elevada possibilidade de existência de erros que seguem as diretrizes daqueles encontrados na Internet.

Este servidor, consegue através da utilização da extensão Auth implementar o processo de permissões de *login* mediante a atribuição do TOKEN e ainda da definição de um espaço temporal para este. Mediante isto definiu-se a validade de 1 dia até o TOKEN expirar pensando ser adequado ao tipo de aplicação, visto se fosse menor e fosse necessário projetar a plataforma sem interação humana durante bastante tempo é incomodativo estar sempre a inserir credenciais.

5.4 RESULTADOS E DISCUSSÃO

Na presente secção serão abordados os principais resultados da implementação dos processos anteriormente descritos assim como uma breve discussão dos mesmos. Devido à longa estenção do capítulo optou-se pela inserção das figuras que demonstrar as diversas facetas da aplicação no anexo B.2

O script apresenta 2 processos que correm de forma paralela com um ligeiro atraso em relação a um terceiro. Este processo tem como função criar tuplos de informações sobre agentes e vai servir de suporte aos processos paralelos, relativos a mensagens. Pelo menor de consumo de recursos, a classe de transformação da *performance* faz parte do primeiro processo mencionado.

O *data warehouse* é então composto pela tabelas descritas na seguinte lista. É também explicado um pouco das transformações por que passam os dados nelas contidos.

M_ENVIADA_HORA: a chave primária é o id da mensagem. Utilizando a origem, o destino e neste caso o tipo C obtém-se o agente responsável. Um ponto importante mais relacionado com parte de BI , é o tempo que decorre desde momento de envio até momento em que chega a confirmação de que foi entregue. A subtração de tempos acontece utilizando a classe *datetime*. A utilização do tempo de envio permite também enquadrar a mensagem nesta tabela indicando que tuplo representa o corrente dia;

M_RECEBIDA_HORA: a chave primária é o id da mensagem. Utilizando a origem, o destino e neste caso o tipo S obtém-se o agente responsável. Neste caso o importante é saber o tempo entre a receção da mensagem e se não existir erro HL7 o tempo que esta foi integrada pelo programa de destino. A subtração de tempos acontece utilizando a classe *datetime*. A utilização do tempo de envio permite também enquadrar a mensagem nesta tabela indicando que tuplo representa o corrente dia;

M_RECEBIDA: o esquema é igual ao anterior mas os espaços temporais são relativos aos 3 últimos dias;

M_ENVIADA: o esquema é igual ao anterior mas os espaços temporais são relativos aos 3 últimos dias;

PROCESSOS_ATIVOS: apresenta informações sobre agentes utilizando como chave primária a origem, o destino, o nome e tipo. Utilizando o tempo desde a última mensagem é possível obter a probabilidade de erro subtraindo essa data ao tempo atual. Depois utilizando intervalo de valores pré definidos obtém-se valor 0 1 ou 2;

TIMELINE_MESSAGES: esta tabela nasceu da necessidade de apresentar ao utilizador as mensagens mais recentes. São apenas retiradas as últimas 5 enviadas e recebidas e inseridas nesta tabela.

A primeira parte do trabalho desenvolvido prende-se com o processo de ETL de dados recolhidos no decorrer da atividade dos agentes AIDA e HL7. Os dados recolhidos até momento, não eram submetidos a qualquer tipo de processamento, com a implementação do processo podem ser posteriormente apresentados ao destinatário. Alguns dos resultados alcançados com a implementação do processo de ETL são:

- Organização de agentes por origem e destino e ordenação por *Internet Protocol (IP)* da máquina onde estão inseridos;
- Mediante a origem e o destino da mensagem obter o agente responsável;
- Obter a diferença temporal entre o envio e o ACK no caso de uma mensagem enviada e diferencial entre data de receção de uma mensagem e sua data de integração;
- Obter os totais diários de mensagens enviadas e recebidas assim como de erros;
- Disponibilização de informação para *timelines* de mensagens;
- Organização de informação por dias de envio ou receção;
- Organização temporal de dados relativos à performance de agentes.

Alguns destes dados poderiam ter sido obtidos no momento em que é feito o pedido ao servidor RESTFUL, mas tal atrasaria o processamento do pedido com *queries* complicadas e envolvendo por vezes milhares de entradas da tabela da base de dados. A organização dos agentes efetuada inicialmente prende-se com o facto de nos registos iniciais alguns agentes terem origens ou destino múltiplos, com a separação destes em diversos tuplos, torna a identificação de destinatários ou recetores de mensagens precisa. A escolha do horário para correr estes processos não pode ser feita de animo leve, foi assim escolhido o horário noturno para percorrer os processos mais pesados. A organização dos últimos 3 dias só é feita uma vez por dia, ficando as mensagens do dia a ser agrupadas de hora a hora. Com este esquema existe menor peso computacional para o SO e maior clareza de processos.

Relativamente ao sistema de monitorização em Angularjs este consiste em 2 módulos, um relativo aos agentes AIDA e outro relativo a agentes HL7. Optou-se por utilizar uma abordagem onde primeiro são apresentados dados gerais e simples, posteriormente observados os detalhes caso seja pedido (figuras 35 e 36). A vista inicial da aplicação pode ser disposta em dois modos diferentes consoante a utilização em computador (tabelas) ou modo projeção para utilização em grandes ecrãs (figuras 27 e 29). Este último utiliza estruturas em forma de quadrilátero, que assumem coloração diferente a probabilidade de erro de determinado agente (verde, amarelo, vermelho). Até ao momento a deteção de possíveis erros em algum agente só era possível pelos efeitos que estes causavam, nomeadamente atrasos em mensagens ou falta de atividade prolongada. Com a utilização de uma cor que desperte atenção do utilizador é possível detetar a fonte do problema e corrigir sem que existam grandes perdas no volume de informação transferida.

A aba representativa dos indicadores foi dividida em mensagens enviadas e recebidas, mais uma vez com pensamento na simplicidade de leitura e interpretação dos dados. O desenvolvimento destes permite ter uma noção de quais os agentes com mais cargas (volume de mensagens transacionadas), utilizando um gráfico de barras é possível descriminar esses pontos por agente e por dia. No caso das mensagens desde as o horas desse dia até ao momento presente descrimina por hora e por agente o volume de mensagens transacionadas (ver figuras 33 e 34). Um nível de tal precisão não existia até momento podendo tal ser obtido através de *queries* que envolviam aceder a bases de dados. Assim sendo as melhorias para o utilizador são tremendas simplificando em muito o processo de tomada de decisão. Uma questão que seria interessante introduzir era um histórico mais alargado em termos temporais permitindo ter noção ao longo de um mês por exemplo, quais os períodos com maior volume de mensagens.

O controlo de agentes inteligentes HL7 (ver figura 31) é uma vertente importante, nomeadamente a opção de desligar e voltar a iniciar à distância. A opção é comunicada ao controlador geral que por sua vez o comunica aos controladores locais que desempenham a determinada opção. Esta opção simplifica em muito a vida aos responsáveis dos sistemas de informação da unidade hospitalar uma vez que se algum erro ocorrer pode ser encerrado o agente à distância sem seja necessário necessário aceder à máquina através de ssh ou outro método de elevada complexidade.

O modulo de agentes AIDA tem uma abordagem um pouco diferente (ver figuras 28 e 32). Primeiro são registados todos os avisos que até agora eram enviados através do sistemas de email do hospital. Com esta implementação é possível aceder a informação extra sobre o agente mal exista algum aviso. Por outro lado a quantidade de agentes implementados é considerável e poderia ser incomodativo receber tantos avisos ao mesmo tempo. Utilizando a plataforma estes estão disponíveis utilizando uma *timeline* interativa.

5.5 CONCLUSÃO

A criação de soluções que se destinem a apresentar informação ao utilizador final são um ponto sensível de qualquer sistema de informação. A plataforma desenvolvida assente em Angularjs e bases de dados Oracle apresenta características inovadoras nomeadamente no controlo de agentes inseridos num sistema MAS. Embora a unidade de saúde já reunisse informações sobre agentes HL7 e AIDA não disponha de uma interface com utilizador na verdadeira escanção da palavra. É vital para a unidade de saúde obter do meio informações constantes e fidedignas sobre dois dos principais provedores de interoperabilidade existentes.

O impacto deste sistema de monitorização na unidade de saúde é tremendo uma vez que permite aos serviços de informação obter informações detalhas e ao minuto o que se passa com agentes AIDA e HL7 e alguns desses agentes sem ser necessário aceder à máquina onde estes estão a operar.

Ao longo do processo foi percetível a complexidade que um sistema de apresentação de informação acarreta, com a criação de interfaces e como devem ser dispostos os dados com vista a gerar informação. Fica também explicito a importância do processo de ETL num processo desta complexidade, sem este, a informação poderia estar a utilizar dados desnecessários, contraditórias ou repetidos.

O Angularjs revelou-se uma escolha acertada, apresenta uma aplicação web com um bom desempenho mesmo quando a quantidade de dados processada é elevada. Com esta ferramenta foi possível implementar uma solução com *design* apelativo e fluido. A criação de servidor RESFUL sustenta todas as interações com a base de dados sendo de fácil implementação e gestão. Este permite também realizar operações de LOGIN e LOGOUT e ainda em caso de necessidade enviar avisos para o telemóvel. Novamente a escolha deste revelou-se acertada já que a simplicidade e versatilidade são mais que evidentes.

Em suma desenvolveu-se um sistema de monitorização com componentes de gestão de um sistema MAS, que também apresenta componente de BI . Os sistemas de informação nas unidades hospitalares são essenciais ao seu funcionamento, se uma aplicação que apresente informação sobre o paciente é importante, um sistema que apresente desempenho dos componentes que permitem tal apresentação não é menos importante. O controlo de agentes revelou-se uma das componentes mais complicadas uma vez que tentar controlar sem utilizar medidas SSH ou DLL por vezes acarreta problemas adicionais ao sistema, estando esta fase ainda num protótipo inicial.

6

CASO DE ESTUDO - FORMULÁRIO DE CODIFICAÇÃO ICD9-CM

A codificação ao nível hospitalar traduz-se na utilização de nomenclaturas e sistemas de codificação adequados para procedimentos, diagnósticos e atos, representando assim o contacto do utente com a unidade de saúde (internamento, hospital de dia, consulta externa ou urgência). Posteriormente, esta normalização pode ser utilizada em *Processo Clínico Eletrónico (PCE)* , ou outros sistemas em qualquer parte do planeta uma vez que estes códigos geralmente são universais.

No que respeita à codificação de diagnósticos e procedimentos realizados em internamento, ambulatório médico ou ambulatório cirúrgico é utilizada a Classificação Internacional de Doenças 9ª Revisão Modificação Clínica (ICD-9-CM).

Ao longo do presente capítulo são assim apresentados os principais aspetos e resultados da implementação de um sistema de codificação *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* de altas hospitalares, integrado num sistema de *Grupo de Diagnósticos Homogêneos (GDH)*, que se destina a utilização por parte de clínicos codificantes do Centro Hospitalar do Porto. De referir por último que todo o trabalho foi desenvolvido em conjunto com Ana Coimbra e Marisa Esteves.

6.1 INTRODUÇÃO E ENQUADRAMENTO NO TEMA

Uma medida capaz de caracterizar a produção hospitalar (quantitativa e qualitativa) é de vital importância, numa economia com fundos cada vez mais restritos para a área da saúde. Em Portugal, essa medição pode ser realizada de duas formas distintas, utilizando Grupos de Diagnósticos Homogéneos ou o *Disease Staging* [167]. Este último mais focado nas características do doente, nomeadamente no risco de morte e na procura de cuidados ao invés do GDH mais ligado à oferta de cuidados de saúde.

A atribuição de um GDH a um episódio só é realizada a partir da informação registada no resumo de alta com informações de diagnóstico principal, diagnósticos secundários, procedimentos cirúrgicos, idade e destino após alta. De acordo com Artigo 3, alínea r da Portaria 567/2006 de 12 de Junho, os diagnósticos e procedimentos no Serviço Nacional de

Saúde, são codificados de acordo com Classificação Internacional de Doenças, 9ª Revisão, Modificações Clínicas (CID-9-MC ou ICD-9-CM) [167]. Na Europa, muitos países utilizam sistemas próprios de GDH embora Portugal ainda utilize sistemas importados dos Estados Unidos da América (*All Patient (AP-DRGs, Centers for Medicare and Medicaid Services (CMS-)DRGs*). Desenvolvido a partir do agrupador original da Medicare, o AP-DRG veio introduzir alterações na obstetrícia, recém-nascidos, transplantes, queimaduras e doentes com VIH sendo considerado o mais indicado para o Sistema Nacional de Saúde [168]. De referir também que Portugal foi pioneiro na implementação modelos GDH para esquemas de pagamentos de seguros no final dos anos 80 [169].

Existem outros tipos de agrupadores GDH, como por exemplo *All Patient Refined DRG (APR-DRG)* agrupando os doentes em 4 graus de severidade e outros 4 de mortalidade [168]. Em paralelo existem alternativas ao sistema de classificação que podem ser aplicadas à classificação e agrupamento de doentes internados em hospitais de agudos. Entre os quais *Disease Staging, Medis Groups ou Acute Physiology and Chronic Health Evaluation (APACHE)* [168].

Como referido no capítulo 2 as classificações ICD são utilizadas a nível hospitalar em GDH e ainda ao nível de sistemas de apoio ao médico [169]. No início o objetivo principal foi definir produtos hospitalares e medir aquilo que estas unidades fazem. Com a expansão por todo o mundo, mas com maior incidência na Europa, o termo GDH tem sido utilizado numa grande variedade de casos. Países como Suécia, Finlândia utilizam para determinar a medida *casemix*, enquanto França ou Alemanha utilizam o sistema como barómetro para atribuição de fundos a uma unidade de saúde [169]. Em Portugal a implementação de sistemas GDH iniciou-se no ano de 1984 com o objetivo original de retirar um *feedback* do que os hospitais são capazes de produzir. No ano de 2010 o propósito foi alterado para alocação de verbas às unidades de saúde [169].

A implementação de sistemas GDH aumenta a transparência identificando produtos cujos diagnósticos estão relacionados e com consumo homogéneo de recursos, constituindo uma metodologia que identifica a complexidade de produtos. Estando assim mais relacionada com a oferta de produtos clínicos do que com a procura.

Cláudia Medeiros Borges no trabalho *"Implementação de um sistema de custeio por actividades nos hospitais do SNS"* apresenta um sistema para determinação de custos em ambiente hospitalar como peça fundamental num mecanismo de contratualização [170]. São apresentadas também as razões da implementação de um Projeto Piloto no ano de 2007 pelo Ministério da Saúde em 5 hospitais do Sistema Nacional de Saúde. A mesma autora do trabalho anterior em conjunto com Nuno Amaro apresentaram também um sistema de compras hospitalares tendo como base o trabalho desenvolvido em GDH no trabalho *"The implementation of a purchasing mechanism for hospital resource allocation in Portugal"* [171].

John E. Wennberg e seus colaboradores, no trabalho *"Will Payment Based on Diagnosis Related Groups Control Hospital Costs?"* concluiram que programas de manutenção de custos baseados em valores fixados antes da admissão do paciente vao assegurar num futuro próximo o *rating* hospitalar para atribuição de verbas [172].

Joseph T. English e seus colaboradores proposeram em *"Diagnoses-Related Groups and General Hospital Psychiatry: The APA study"* uma tentativa de implementação de GDH em hospitais com unidade psiquiatrica para num futuro estabelecer ligações à Medicare. Concluiram no entanto que a falta de dados exatos e concretos neste setor, pode levar a riscos do foro financeiro para as unidades de saúde. Daí que seja imperativo o aumento da interoperabilidade neste setor [173].

A codificação de diagnósticos e procedimentos realizados é então essencial para o dia a dia de uma instituição de saúde. Ao longo do presente trabalho é abordada a implementação de uma aplicação web, com capacidade de apresentar os pdf com relatórios diários de internamentos sendo assim possível codificar os respetivos diagnósticos, as causas externas, a morfologia tumoral e ainda os procedimentos executados durante o internamento. Como demonstrado em 2 existem relações diretas entre causas externas, morfologia tumoral e diagnósticos.

A aplicação de codificação faz parte de um módulo mais abrangente já desenvolvido pelo Centro Hospitalar do Porto. É este modulo que invoca a aplicação desenvolvida utilizando URL que passa parâmetros na *string*. Optou-se então por desenvolver a aplicação utilizando a *framework* Angularjs. Embora este caso de estudo seja semelhante ao anterior na implementação, os destinatários são diferentes, visto que neste caso são os médicos codificantes e no anterior eram profissionais do sistemas de informação. Ao nível da interoperabilidade a possibilidade de um sistema codificar para posterior agrupamento, fomenta a interoperabilidade do tipo semântico entre soluções informáticas.

6.2 MATERIAIS

Ao longo da presente secção são expostos os principais materiais utilizados na implementação deste caso de estudo. A metedologia e o motivo de escolha destas ferramentas pode ser observado na próxima secção juntamente com breve uma descrição de algumas propriedades. Assim sendo os principais materiais foram:

- Angularjs para implementação da *single-page-application* com alguns *plugins* instalados *à posteriori*:
  - angular-router;
  - angular-bootstrap;
  - angular-modal-service.

- Mysql contendo o *data warehouse*;
- Web Service do tipo Restful implementado na linguagem Php.

6.3 METODOLOGIA

Seguindo a metodologia *Design Research* apresentada no capítulo 3, primeiro devem ser recolhidos os requisitos ou situações que carecem de melhorias no *Centro Hospitalar do Porto (CHP)*. Neste caso e como demonstrado até à presente secção, a principal necessidade é uma aplicação web para codificação de resumos de alta que possa ser integrada numa aplicação de Grupo Diagnósticos Homogéneos. As principais necessidades/requisitos recolhidos através de reuniões com pessoal hospitalar são:

- Sistema de codificação ICD-9-CM;
- Aplicação destinada à Web utilizando o esquema SPA.
- Utilização das bases de dados MySql;
- Gravação de dados em formato que possibilite a posterior consulta;
- Metodologia que permite a gravação automática de informação, em casa de acontecerem erros;
- *Single-page Application* com 3 modos de edição/visualização distintos;
- Organização da interface de forma a que seja possível consultar um resumo de alta à medida que é realizada a codificação.

Ao longo da presente secção serão abordados diferentes aspetos relacionados com a solução que melhor permite responder a estes pontos levantados na lista anterior e que culminaram com a implementação dessa ferramenta na unidade de saúde. A melhor forma de demonstrar a metodologia implementada é fazer uma divisão da secção em pequenas sub secções apresentando assim cada componente do sistema. Embora não seja necessário um processo de *Extract, Load, Transform (ETL)* é necessária a utilização de um *data warehouse* com dados sobre códigos e designações ICD-9-CM, pacientes, altas e ainda médicos codificantes.

Assim sendo e seguindo o esquema de *single-page application* a aplicação apresenta características que favorecem à semelhança do caso de estudo anterior a *framework* Angularjs. Faz parte dos requisitos iniciais a utilização da base de dados *Mysql* e a interação com diversos ficheiros no formato pdf vitais ao desenrolar da codificação.

Comparar *frameworks* librarias ou extensões que utilizem *JavaScript* hoje em dia é um processo simples motivado pela grande informação presente *online*. Num artigo de opinião

de Jeremy Likness intitulado *"10 Reasons Web Developers Should Learn AngularJS"* do ano de 2014 o autor apresenta o seu ponto de vista enquanto programador de quais são os factores que tornam esta *framework* atrativa [165]. *Angularjs* simplifica a ligação entre a interface (vista) e o local onde os dados de facto estão armazenados (modelo), utilizando por exemplo a diretiva *ng-model* não sendo necessário declarar explicitamente as propriedades ou dependências. Embora existam alternativas que abordam de forma mais correta, implicam a passagem de dados para um objeto para depois ser utilizado no modelo.

O facto de lidar com dependências como quase nenhuma *framework* simplifica o teste das aplicações e a criação de *Single Page Applications* [165]. No primeiro ponto a aplicação pode ser divida em módulos que embora dependentes podem ser iniciados em separado aumentando assim as capacidades de teste. No segundo ponto utilizando o carregamento dinâmico permite obter aplicações quase nativas partindo de aplicações com base na web.

A percepção de que controladores não funcionavam corretamente fora do mundo *ASP.NET* mas esta *framework* prova o contrário. A reutilização de código foi abordada de diversas formas utilizando *jQuery plugins* ou *KendoUI*, neste caso as *directive* permitindo a criação de elementos e atributos que podem ser utilizados em qualquer ponto da aplicação.

Uma vez que a aplicação corre numa máquina Linux é utilizado o servidor Apache e desenvolvido posteriormente um servidor do tipo *restful* para suportar acessos a base de dados.

Apache HTTP Server foi utilizado na implementação no Centro Hospitalar do Porto uma vez que as máquinas em produção nessa unidades utilizam a distribuição Ubunto do sistema Linux. Este servidor web é atualmente o mais utilizado em todo o mundo baseado no servidor *NCSA HTTPd* datado do ano de 1995. Algumas das propriedades mais importantes deste servidor são:

- Carregar modelos dinâmicos;
- Processamento de *Multiple Request Processing Modes (MPMs)* incluindo *Event-base/Async*;
- Processar ficheiros estáticos ou indexados;
- *Reverser Proxy*
- Protocolo HTTP;
- Reescrita de URL.

O caso de *XAMPP* é um pouco diferente, é uma plataforma de distribuição livre multiplataforma. Consiste basicamente em um servidor Apache *Hypertext Transfer Protocol (HTTP)*, uma base de dados e interpretadores para a linguagem Php e Perl. O nome deriva de *Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)*. Enquanto no caso anterior tudo deve ser implementado em separado, neste caso tudo está junto numa única ferramenta.

O servidor *restuful* como abordado em 2, foi desenvolvido em PHP. A escolha da linguagem prende-se com a fácil implementação e interpretação utilizando Apache2. A linguagem PHP utiliza conetor específico que permite efetuar *queries* as base de dados. Neste cenário o Apache facilita a ligação à base de dados MySql como demonstrado no estado de arte.

Uma codificação ICD-9-CM de um relatório de alta pressupõem a codificação de Diagnósticos, Procedimentos, Morfologia Tumoral e ainda Causas Externas. Estas duas últimas têm forçosamente de estar relacionadas com os diagnósticos podem a relação ser de um para muitos ou um para um. Deve ser definido pelo codificador um diagnóstico principal sendo acrescentados posteriormente outros se tal for necessário. É importante também perceber o contexto em que é utilizado ou modificado o *Presente na Admissão (PNA)*, indicando assim se determinado diagnóstico existe no momento de registo do episódio ou apenas se manifestou à posteriori. Existem aqui diversas situações que devem revistas cuidadosamente pelo codificador existindo para tal formação especializada.

Seguidamente explica-se em separado a metodologia implementada para acesso a bases de dados e para implementar a *single-page application*.

6.3.1 *Modelo Relacional*

O modelo relacional neste contexto envolve tabelas relacionadas com os códigos ICD-9-CM, assim como informações do paciente à qual o resumo de alta se refere. Existe também a necessidade de criar tabelas para dar suporte ao sistema de codificação e como tal não contém informação sobre o paciente ou codificação, a não ser que já tenha sido iniciado o processo de codificação. As tabelas que constituem o modelo relacional são:

CODI.IC9CM_LISTA : tabela onde estão contidos todos os códigos ICD-9-CM relativos a diagnósticos, causas externas, morfologias e procedimentos. A distinção entre diferentes tipos de códigos, faz-se com a propriedade icd_grupo, existindo um caracter diferente para cada cenário. Assim sendo, P identifica os procedimentos, D os diagnósticos, C as causas externas e M as morfologias tumorais. A chave primária neste caso é o icd_cod;

CODI.PROCESSOS_ANTERIORES : esta é uma das tabelas criadas para suportar a aplicação, neste caso, regista todas as versões que são realizadas de determinada codificação. As colunas mais importantes são as que indicam qual o caminho para o ficheiro *JavaScript Object Notation (JSON)* (PATH), onde está toda a codificação elaborada e ainda a coluna contendo a versão em que essa codificação está (VERSAO);

CODI.WLIST_ALTAS nesta tabela é possível obter informações relativas ao paciente, qual o episódio e o identificador único da alta assim como o código de médico que a passou.

A chave primária é a id_alta sendo esta a utilizada nas outras tabelas para identificar inequivocamente a codificação. Um dos pontos necessários para o sistema GDH é o destino após alta, é nesta tabela que essa informação está contida, mais propriamente na coluna des_destino. A coluna local_pdf indica à aplicação o nome do ficheiro PDF onde está o resumo de alta.

CODI.WALTAS_MOVE é nesta tabela que são armazenados a data de começo e termino da codificação, com a coluna respetiva ao id do médico codificante a ser uma das principais colunas nesta tabela. A chave primária é um identificador incremental estando também presente o id de alta como chave estrangeira neste caso;

CODI.PROCESSOS_ANT_LOG apresenta todas a versões de determinada codificação servindo um propósito diferente da tabela processos_anteriores, uma vez que serve de *trigger* a outros processos subsequentes;

CODI.LIST_TRANSF_ALTAS contém informação sobre os diversos serviços onde o paciente esteve internado e ainda quais as datas de entrada e saída em cada um, a chave identificadora neste caso é o número de episódio;

SONHO_MRA.LOCAL_SYS_MEDICOS permite aceder a informações do médico codificante;

SONHO_MRA.LIST_USERS apresenta a lista de utilizadores para efeitos de permissões.

O modelo relacional bastante simples e intuitivo sem qualquer processo de ETL uma vez que tal não é necessário nesta fase de implementação.

6.3.2 *Aplicação Web*

Seguindo o modelo das *single-page application* existe apenas uma página contendo todas as funcionalidades da aplicação. Como abordado no início da secção existem algumas diretrizes que devem ser seguidas, nomeadamente a disposição da página deve estar organizada de tal forma que o PDF com informações do resumo de alta apareça lado a lado com zonas de codificação.

De acordo com indicações da bibliografia e de colaboradores do Centro Hospitalar do Porto foi pedido um sistema que permita codificar 4 pontos que estão presentes no resumo de alta. Em primeiro **Diagnósticos**, seguem-se **Causas Externas**, **Morfologia Tumoral** e por último **Procedimentos**. O primeiro conceito contém as conclusões a que o clínico chegou depois de elaborado o ato médico. O segundo representa atividades em que o indivíduo esteve envolvido que possam ter conduzido a tal diagnóstico. Morfologia tumoral representa assim o perfil histológico das células tumorais aquando da observação ao microscópio. Por último os procedimentos representam todas os tratamentos aplicados durante um episodio a determinado paciente.

Talvez o requisito mais importante para a definição do sistema, sejam as relações que se devem estabelecer entre representações. Ou seja, um diagnóstico pode estar associado a uma causa externa e a uma morfologia tumoral podendo a relação ser de um para muitos.

A melhor solução para implementar estes sistemas é na forma de uma tabela que contenha colunas onde podem ser introduzidos valores que permitam pesquisar códigos ICD-9-CM e ainda estabelecer as relações mencionadas.

Definiu-se em conformidade com diretrizes da unidade hospitalar que a página de codificação pode ser acedida de três formas distintas, um novo registo, um registo que foi gravado mas não terminado e o modo consulta de alguma codificação que foi gravada e enviada previamente.

Quando uma alta é processada é gerado um PDF com o resumo de tudo o que foi realizado ficando este documento guardado numa máquina em produção com o sistema operativo Linux. Essa máquina contém no diretório, pastas com a ID de alta e data da mesma a representarem o nome de pastas que devem ser utilizadas pela aplicação para aceder ao mesmo.

A pesquisa nos diversos campos é bilateral, ou seja, pode ser pesquisado em primeiro lugar a descrição que fornece automaticamente o código respetivo ou inserindo o código e com um simples *enter* obter a descrição.

Para operar nos termos apresentados anteriormente, diversos parâmetros devem ser passados através do URL que os controladores vão interpretar e utilizar, assim sendo é utilizado **id de alta**, **id medico** e por **modo de consulta**. Este último é vital para controlar o modo como os dados são carregados e apresentados ao utilizador. A possíveis opções são:

1. Indica o **início** de um novo processo de codificação ou em caso de erro basta reabrir o navegador utilizando este modo e mantendo o URL;

2. Indica a **recodificação** de um processo. Não altera em nada a informação apresentada ao utilizador ou os campos de inserção. Neste cenário é identificado primeiro qual o caminho e versão para o ficheiro JSON identificado com F de Final, onde estão armazenadas as informações da codificação passada gravada e enviada;

3. Indica a **consulta** de um processo previamente submitido. Neste caso é carregada uma página com as mesmas tabelas apenas os campos estão bloqueados à modificação. Foi também retirado o pdf e janelas *pop-ups* substituídos por listas de ligações.

Para assegurar que nenhuma informação é perdida em caso de erro, deve ser implementado um sistema que grave a informação existente cada vez que é inserida uma nova linha, gravando toda a codificação em formato JSON no mesmo diretório do PDF. Este formato permite a utilização das mesma estruturas JavaScript (dicionários) utilizadas pelos controladores, ou seja, o modelo contém de forma atualizada dicionários que representam as estruturas desenvolvidas na vista. A estrutura JSON é também utilizada nos casos de gravar para

codificar mais tarde (tipo 2) e encerrar e enviar. Neste último caso o nome ficheiro é alterado existindo posteriormente *insert e updates* no modelo relacional (codi.processos_ant_log, codi.processos_anteriores), para informar o sistema que terminou codificação.

A pesquisa dinâmica é outro conceito que merece atenção. Por exemplo quando o utilizador preenche o campo referente à descrição de um diagnóstico na terceira letra é realizada uma QUERY na base de dados que devolver todos diagnósticos (letra D no campo com nome tipo) que contenham essas três letras. Armazenados num dicionário são apresentadas na forma de pesquisa dinâmica ao utilizador que pode selecionar uma opção obtendo assim código.

Um aspeto que exige algum processamento ao nível do navegador são as trocas e remoção de linhas, uma vez que implica uma total redefinição das estruturas criadas (objetos) para armazenar a informação no decorrer do processo de codificação. Fez-se então uso de diversas estratégias da linguagem JavaScript para fazer tais implementações.

O sistema foi assim delineado para facilitar o processo de codificação ajudando o médico responsável no máximo de aspetos possíveis.

6.4 Resultados e Discussão

Das necessidades/requisitos levantados inicialmente, todas foram respondidos com esta solução e submetidos à próxima etapa do *Design Research* onde é feita a avaliação da solução encontrada.

O modelo relacional implementado em MySql já foi previamente demonstrado na secção anterior e foi carregado *à priori*, utilizando dados provenientes do grupo *International Classification Diseases 9 Revision (ICD-9)* para língua portuguesa. Existem no entanto tabelas que só são utilizadas para situações ligadas a versões de codificação assim como o caminho que deve ser utilizado para aceder a essas mesmas versões.

A aplicação desenvolvida apresenta 3 modos de utilização que são passados como argumentos do URL (ver topo figura 19), mediante isso são carregados dois *Hyper Text Markup Language (HTML)*, o primeiro completo (modo 1 e 2) e o segundo, uma versão reduzida e com todos os input bloqueados (modo 3). Para os três modos poderia ter sido utilizado o mesmo HTML visto que se destina a apresentar o resultado da codificação, mas segundo *feedback* dos primeiros testes, optou-se por desenvolver um novo módulo retirando a capacidade de inserir novas linhas ou fazer pesquisas e ainda adicionado um link para abrir o resumo de codificação. Retirou-se também botões de inserção ou alteração de linha visto não serem necessários.

A página do tipo 1 e 2 (codificação e recodificação respetivamente) de desenvolvimento pode ser dividida em 3 componentes distintos:

CABEÇALHO: contendo o logotipo do Centro Hospitalar do Porto, e ainda informações sobre o paciente, médicos de alta e codificante, identificação de alta e episódios, assim como idade do paciente de forma discriminada. Foi aproveitado este espaço para colocar zonas de preenchimento de alguns parâmetros da codificação não sujeitos a codificação, nomeadamente o número de dias de ventilação mecânica (VM), a simultaneidade e o destino pós alta;

ZONA DE CODIFICAÇÃO: conjunto de 4 tabelas e uma zona de texto livre que permitem a codificação. Esta componente será descrita em pormenor adiante;

ZONA DE PDF: com a identificação da alta e data em que esta acontece é possível aceder ao resumo de alta, em formato PDF (ver parte inferior da figura 19). A pedido da instituição, este deve ser bem visível e disponível a qualquer momento.

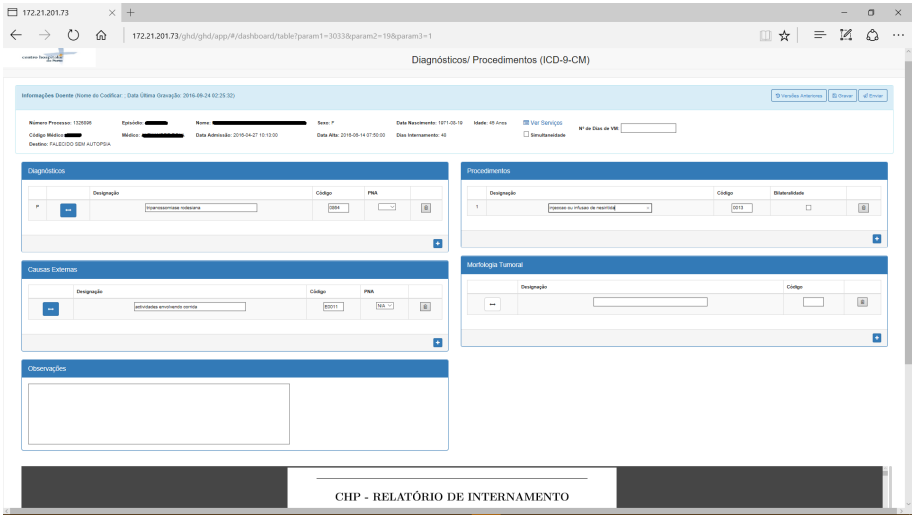


Figura 19.: Aplicação web desenvolvida em modo codificação ou recodificação.

O cabeçalho é transversal a todos os modos de codificação (topo das figuras 19 e 20), não sofrendo alterações de conteúdo nos três modos, o cabeçalho apresenta também informações sobre o médico que prescreveu a alta assim como médico que realizou ou está a realizar a versão atual de codificação. A presença deste último identificador prende-se com a criação de bases para um módulo de *Business Intelligence (BI)* contendo por médico, número de altas mensais, número de codificações mensais e ainda outras informações relevantes para instituição de saúde ou clínicos.

Segundo as diretrizes GDH, devem ser averiguadas algumas situações e respondidas de forma clara (texto, ou *checkbox*) como o destino pós alta, dias ventilação mecânica e bilateralidade que foram colocados no cabeçalho simplificando o design da página. A possibilidade de inserção desses elementos já nesta página possibilita uma maior simplicidade

no desenrolar do processo GDH, caso contrário posteriormente outro *sctipt* informático ou codificador humano teria de responder a essas questões.

Observando em maior detalhe a **zona de codificação** (centro da figura 19), esta apresenta 4 tabelas que se destinam a codificar ou pela inserção do código ou de uma descrição dos diagnósticos, os procedimentos, as causas externas e morfologia tumoral. A ordem pela qual as tabelas se sucedem indica ao codificador que primeiro deve realizar o diagnóstico e só depois as causas e morfologias visto estas estarem diretamente relacionadas com a primeira. Segundo as indicações o primeiro diagnóstico é o principal e só quando este for preenchido é que novas linhas podem ser inseridas diminuindo assim a possibilidade de ocorrência de um erro de codificação.

Na segunda e terceiras tabelas o tipo de pesquisa implementado é semelhante, com diferença de que estas devem ser associadas à tabela principal utilizando as setas bilateriais (ver figura 19). Para tal optou-se pela criação de janelas *pop-up* onde o codificador escolhe de todas as linhas da tabela 1 qual, ou quais mais se adaptam a determinada situação. Para melhor refletir o trabalho desenvolvido é possível clicar no inicio de determinada linha da tabela 1 e saber quais as suas ligações. As ligações ficam registadas em *background* num ficheiro JSON, respondendo assim a alguns dos requisitos impostos inicialmente. A *combo box* presente nestas tabelas representa a situação presente na admissão, já abordada na secção anterior começando com um valor *default* nulo e pode posteriormente ser alterado para as quatro opções possíveis. Este ponto pode ser melhorado futuramente, começando com opção S deixando depois espaço à alteração do mesmo consoante vontade do codificador.

Na última das tabelas de codificação estão os procedimentos, estes não apresentam nenhuma ligação as tabelas anteriores, nem necessidade de inserir o, sendo assim uma implementação bastante mais simples que as anteriores. A pesquisa é igualmente dinâmica e bilateral.

No modo 3 de utilização, afeto à consulta do documento (ver figura 20), a implementação é ainda mais simples uma vez que todos os campos de inserção e outras possibilidades de alteração do documento estão bloqueadas ao utilizador. Foram removidas as janelas *pop-up* e substituídas por listas situadas abaixo do campo da descrição representado assim as ligações daquele código dentro daquele quadro de codificação. Retirou-se também o PDF e substitui-se por um botão que abre PDF caso tal seja necessário para esclarecimento de dúvidas.

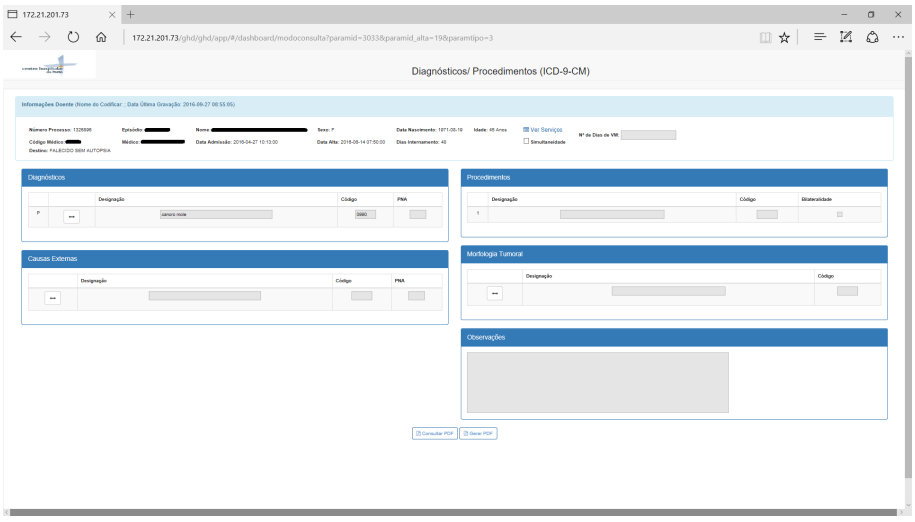


Figura 20.: Aplicação web desenvolvida em modo consulta.

A recuperação do erro foi alcançada implementando um sistema de gravação automática do trabalho elaborado até momento, assim a quando da inserção de uma nova linha faz-se a gravação do documento, se algo de mal ocorrer quando navegador voltar abrir mesmo no modo 1 vai buscar essa informação e a codificação pode continuar. Existem duas formas de terminar o trabalho, encerrar para depois continuar a trabalhar e neste caso é apenas registada uma versão não sendo a final. Caso pretenda dar por terminada a codificação basta selecionar enviar e termina assim o trabalho. Em *background* a aplicação declara a versão como final e faz registo na base de dados.

O impacto de uma aplicação deste tipo para a codificação GDH é tremendo, até então o que existia era um processo completamente *paper based* envolvendo a impressão do resumo de alta para codificação manual. É possível então utilizar um sistema informático que aumenta em muito a velocidade com que ocorre a codificação apresentando maior comodidade para o responsável pelo processo. Se existe maior comodidade e facilidades para os médicos codificantes também deve existir maior produção da parte destes, sendo imperativo utilizar este sistema como barómetro para quantificar o trabalho desenvolvido e apresentando tal informação de forma clara e precisa uma vez que parte do seu vencimento é devida as codificações efetuadas.

Para além do aumento da interoperabilidade semântica entre sistemas, o presente caso de estudo foca-se muito na usabilidade e interação com utilizador através de uma interface simples e com informação detalhada sobre passos a dar.

A aplicação encontra-se atualmente em utilização pelos medicos e pessoal codificador no Centro Hospitalar Porto como parte do seu sistema GDH, não existindo até ao momento erros graves ou situações de falha iminente. Existem pequenas alterações que vão sendo efetuadas a pedido da unidade hospitalar no sentido de melhorar a *performance* ou inserir novos componentes.

6.5 CONCLUSÃO

A importância dos GDH em Portugal é cada vez maior, cada vez é mais utilizado como sistema de distribuição de verbas no *Serviço Nacional de Saúde (SNS)*, sistema para avaliação de encomendas da unidade hospitalar e avaliação de médicos codificantes e daqueles que fazem a alta clínica.

Em Portugal, no presente, a totalidade dos Contratos-Programa, as linhas de produção de internamento, ambulatório cirúrgico e parte do médico são financiadas na integra com este sistema de classificação representando no ano de 2009 51% do financiamento do SNS. Fora aspetos económicos, esta aplicação no SNS serve como avaliação e apresentação de produtos clínicos e ainda como forma de medir o trabalho desenvolvido pelo médico codificante.

De acordo com o estado de arte sistemas de informação dedicados a clínicos são de vital importância para as unidades de saúde. Neste caso de estudo foi proposta uma solução que vai de encontro a essas diretrizes embora envolvido numa funcionalidade que não envolve contacto direto com o paciente.

Todos os requisitos/lacunas apurados inicialmente, foram cumpridos com sucesso como pode ser observado na secção anterior. Optou-se por um *design* minimalista para simplificar o trabalho desenvolvido, assim como utilização do Angularjs permite que a página seja bastante interativa e fluida.

O ICD-9-CM revelou-se uma ferramenta de fácil utilização uma vez que toda a informação está presente numa única tabela Mysql utilizando chaves diferentes para cada cenário. A codificação de diagnósticos é ponto vital dos sistemas GDH e sem ela não é possível realizar tal agrupamento. O impacto deste sistema reside na interoperabilidade semântica que é possível alcançar quando utilizados códigos que representam diagnósticos, causas externas, morfologias e procedimentos.

De referir que num futuro próximo serão introduzidos códigos ICD-10-CM e ainda a possibilidade de gerar pdf com codificação. Existem também relações de exclusividade entre tabelas que deveriam ser automáticamente assumidas pelo sistema que deverão ser implementadas também. A inclusão de um módulo de BI está também nos planos com informações relativas ao volume de altas codificadas por cada médico permitindo estes ter uma noção do impacto deste na sua folha salarial uma vez que existem limites mínimos que devem ser superados.

7

CASO DE ESTUDO - APLICAÇÃO MÓVEL AIDA_PACIENTE

As aplicações móveis atingiram nos últimos anos um patamar de utilizadores bastante elevado, motivado em grande parte por avanços tecnológicos significativos, possibilitando o setor público e as empresas privadas de chegarem a faixas mais abrangentes da população. Através da implementação de aplicações da monitorização da saúde e da atividade física o setor da saúde alcançou soluções originais e criativas no que à partilha de informação diz respeito. Neste contexto surgiram aplicações do tipo *appointment reminder*, bastante ligadas ao conceito de *mHealth* que por sua vez faz parte do conceito de *Electronic Health (eHealth)*. De referir que é impossível construir uma aplicação deste tipo sem recurso à interoperabilidade física, sintática e semântica. Assim sendo, este conceito está presente em todos os casos desenvolvidos ao longo do projeto.

Ao longo do presente capitulo é então apresentada a implementação de um sistema *appointment remainder* vocacionado para a partilha de informação do Centro Hospitalar do Porto com os utentes numa abordagem ao *mHealth*.

7.1 INTRODUÇÃO E ENQUADRAMENTO NO TEMA

A quando da sua apresentação em meados de 2003, o *mHealth* estava bastante ligado ao conceito de Telemedicina. Este conceito evoluiu e atualmente abrange qualquer utilização de tecnologia móvel que permita o acesso, a utilização e a interação com normas ou conceitos clínicos. Os dispositivos móveis não conseguem transportar drogas ou realizar exames médicos mas podem transmitir preciosas informações entre plataformas utilizando texto, imagens ou vídeos para tal. O processo de partilha de informação beneficia de redes móveis como *GSM, GPRS, 3G e 4G-LTE*, tecnologias com base em computadores (*Wifi e WiMAX*) e para comunicações a curta distância tira proveito da tecnologia *Bluetooth* [174]. As plataformas em que estas aplicações vão estar inseridas são igualmente diversas, desde sistemas de distribuição livre como Linux ou Android até sistemas de proprietário como iOS e ainda Microsoft Windows [174].

Para conseguir atrair utilizadores num mercado cada vez mais competitivo, existe a necessidade de apresentar soluções inovadoras que sejam capazes de se sobrepor a dificuldades

económicas ou de meios, para acesso à informação. Os cuidados de saúde proporcionados por países desenvolvidos são completamente diferentes daqueles prestados por países em vias de desenvolvimento. Para atingir o mesmo patamar dos primeiros, é necessário explorar ideias e tecnologias inovadoras utilizando os meios disponíveis, viáveis em termos económicos. Com a proliferação de tecnologia móvel nestes países estão então criados os meios para tal disseminação de informação. Por exemplo, em todo o mundo, atualmente existem 5 biliões de pontos de contacto entre sistemas se saúde e pacientes [174].

As potencialidades da *mHealth* têm sido nos últimos anos notabilizadas pelo setor público e privado. Sendo considerado por muitos *decisionmakers* como uma plataforma global que pode revolucionar o setor dos sistemas de saúde [174].

De acordo com a Organização Mundial de Saúde no documento *"mHealth New horizons for health through mobile technologies"* as maiores iniciativas relacionadas com *mHealth* estão relacionadas com *call centers* ou linhas de ajuda telefônica, serviços de emergência pelo telefone, emergências e Telemedicina [175]. Por outro lado no leque daqueles que menos frequentemente são reportados estão os aplicativos de vigilância, de aviso de riscos e os sistemas de apoio à decisão [175].

Na figura 21 é possível observar o descrito anteriormente onde os *call centers* são as iniciativas mais implementadas em todas as fases de desenvolvimento, exceto no continente Africano. Na Europa existe uma grande proximidade entre sistemas de apoio telefónico e os *toll-free emergency*.É importante reter que os dados apresentados são resultado de um questionário enviado aos países membros da *World Health Organization*.

Desenvolver ferramentas nem sempre conduz à sua aceitação e utilização por parte do destinatário final. De acordo com figura 22 países com elevados rendimentos apresentam um maior número de iniciativas quando comparados com aqueles onde tecnicamente os rendimentos são menores.

Outro dos pontos de destaque dos países com elevados rendimentos é o facto de existir implementações relativas ao *appointment reminder* [175]. Este tipo de sistemas utilizados para enviar avisos aos pacientes que têm uma marcação de qualquer tipo na unidade de saúde tem crescido entre estado membros da *Organização Mundial de Saúde (OMS)*. Analisando as figuras 22 e 21 é possível observar que 71% dos países responderam afirmativamente à existência deste tipo de soluções [175]. A lista é encabeçada pelas Americas com 58% seguido pela Europa com 53% dos países a reportarem inciativas. Os custos de um paciente não comparecer a uma marcação são enormes, por exemplo, em países como Reino Unido e Irlanda perdem por ano 790 milhões de libras [11]. A introdução de uma aplicação móvel que relembre o paciente do agendamento ou que este possa utilizar para confirmar a sua presença pode ser uma solução pouco dispendiosa com uma enorme utilidade [175]. Podem também ser introduzidas questões relativas ao *Patient Relationship Manager (PRM)*, conceito que se baseia no *Costumer Relationship Manager* (CRM) para proporcionar aos paci-

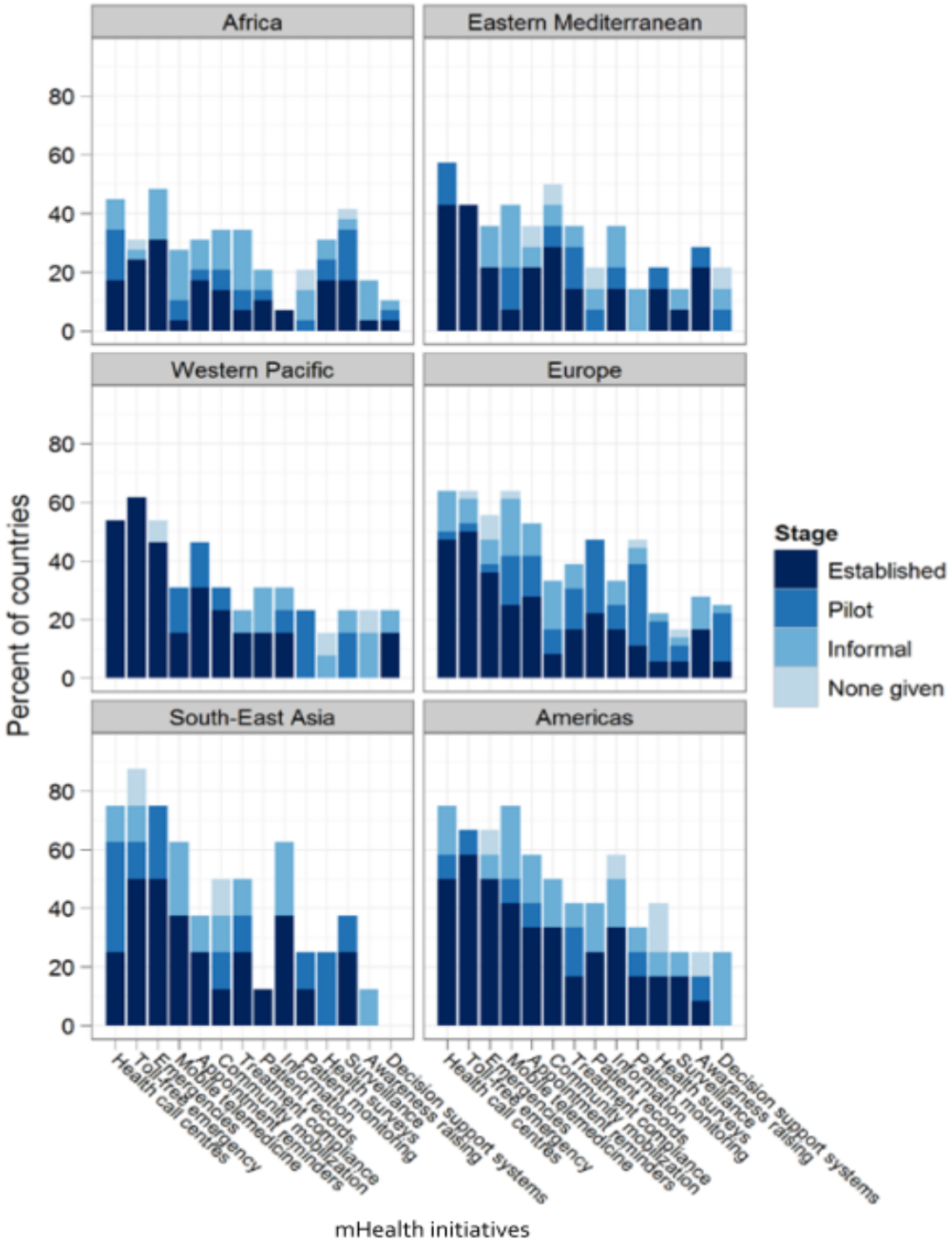


Figura 21.: Iniciativas mHealth (adaptado de [175]).

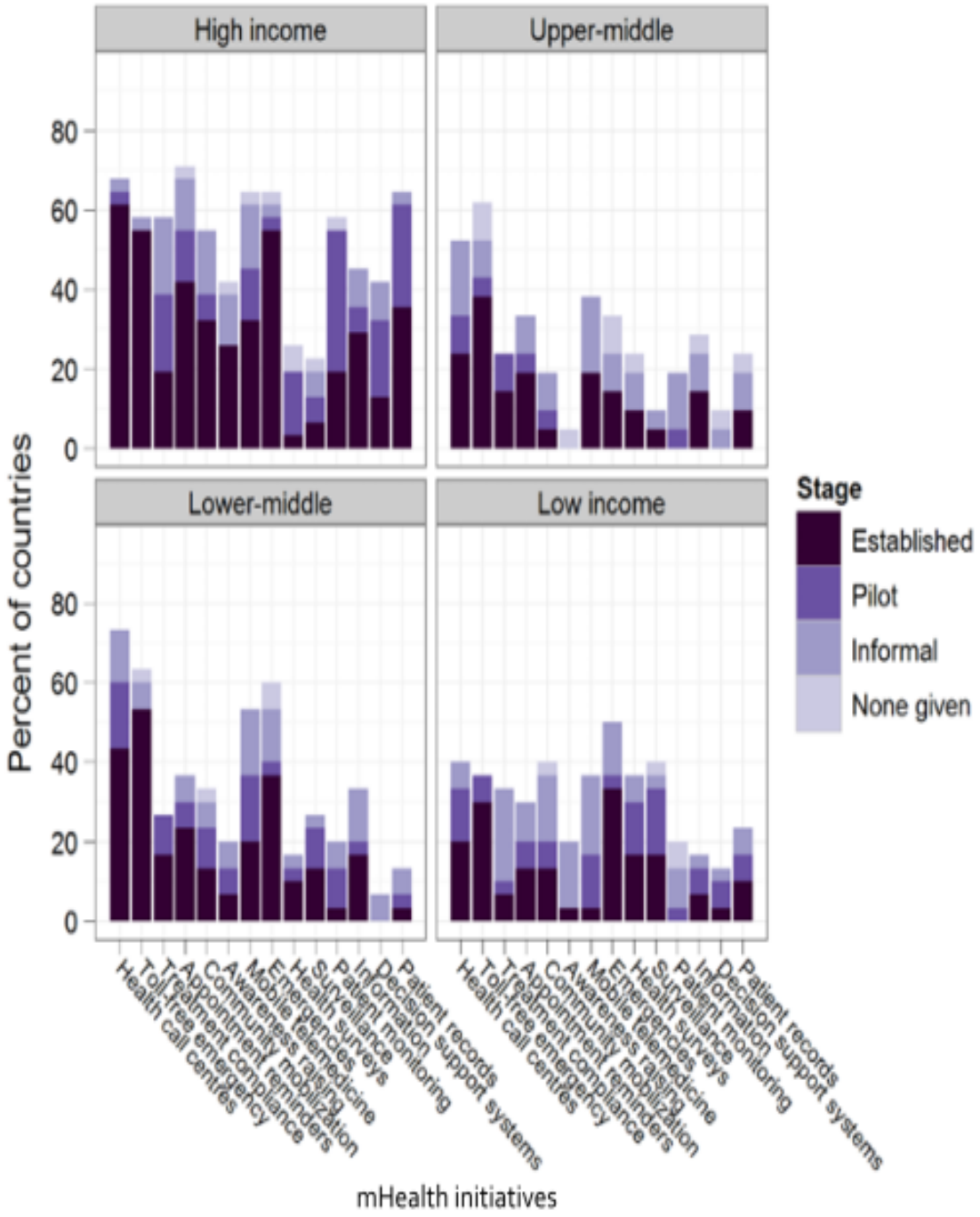


Figura 22.: Adoção de iniciativas mHealth nas mais diversas fase de desenvolvimento (adaptado de [175]).

entes um serviço melhorado, capaz de criar neste lealdade [176]. De acordo com a empresa Infosys no documento sobre *"Patient Relationship Management Solution"*, o PRM ajuda as unidades de saúde a alcançar os seguintes pontos [176]:

- Aumento da satisfação do paciente;
- Integração da informação e aumento da interoperabilidade;
- Aumento da proximidade com pacientes, principalmente doentes crónicos;
- Aumentar a base de clientes em caso das unidades de saúde privadas.

Nos primórdios da implementação deste tipo de sistemas os resultados não eram conclusivos. Em [11, 177, 178] são abordadas as implementações resultados obtidos em países como China, Brasil e Reino Unido. Num estudo realizado em São Paulo no Brasil que envolveu o envio de 7890 SMS utilizando *software* clínico e aplicações web, para pacientes de 4 clinicas privadas levou a uma diminuição significativa nas faltas de comparência [179]. Um dos problemas do envio de SMS é o custo que ele acarreta para a entidade que pretende enviar. Envolve também utilização de uma operadora de rede móvel que o execute.

Com a introdução dos conceitos de *pervasive and ubiquitous computing* novas soluções foram desenvolvidas, nomeadamente novos conceitos de *software* que vão para além dos comuns SMS. Surgiram as *Push Notification*, um mecanismo com base num evento enviado por servidores remotos para o dispositivo móvel. Duas situações se levantam [180]:

- **Acessibilidade:** relacionado com o fazer o *route* do aviso para um dispositivo que pode estar em qualquer ponto do mundo. Por vezes as operadoras móveis utilizam *firewalls* que inviabilizam a seu envio;
- ***Contacto Único:*** relacionado com problemas ao identificar determinado utilizador.

A patente *Method and system of providing enhanced appointment notification service to mobile devices (US 9003328 B1)* [181], propõem um sistema onde o dispositivo móvel inclui uma aplicação configurada de tal forma que é capaz de processar e apresentar uma Push Notification. A aplicação desenvolvida apresenta ainda opções relacionadas com ato médico em questão (adiamento, cancelar, confirmar). Existe então uma interação bidirecional entre unidade de saúde onde aplicação está implementada e o paciente.

Numa outra abordagem(*Design and Implementation of Mobile Push Service- Based Mobile Medication-Hour Notification System*), Jun-Ho Huh e seus colegas propõe um sistema que evita o esquecimento das horas em que deve ser tomada a medicação [182]. Para além do aviso da hora envia também instruções de toma e precauções a ter.

Assim sendo ao longo do presente caso de estudo é demonstrado o impacto do desenvolvimento e implementação de uma solução *appointment reminder*, que serve também como agenda destinada ao paciente. Esta aplicação destina-se à partilha de informação entre

unidade de saúde e o exterior favorecendo assim a comodidade dos utilizadores destas unidades. É necessário ter bastante atenção com a confidencialidade dos dados daí que um sistema de verificação de utilizadores bastante rígido deva ser implementado.

No Centro Hospitalar do Porto já existe um sistema que envia mensagens para paciente a avisar do agendamento. Com a evolução das tecnologias móveis existem alternativas que permitem enviar avisos sem que seja necessário o pagamento de uma taxa por parte da unidade de saúde que merecem ser exploradas.

7.2 MATERIAIS

A presente secção destina-se a apresentar de forma sucinta quais os materiais utilizados ao longo do presente caso de estudo. A completa explicação sobre quais os motivos que conduziram a e tal escolha, será apresentada na secção da metodologia. Os materiais utilizados foram:

- Java Adaptado à programação andorid em conjunto com *Extensible Markup Language (XML)* para programação de interfaces;
- Android Studio;
- Base de dados Mysql;
- Angularjs Para sistema de registo;
- Simples processo de *Extract, Load, Transform (ETL)* utilizando Python. São utilizadas as seguintes expansões á linguagem:
  - cx_Oracle para acesso a base de dados;
  - PYfcm para envio de pedidos *Hypertext Transfer Protocol (HTTP)* para servidor *Firebase Cloud Message*;
  - Mysqlconnector na versão disponibilizada para Python 3.4.

7.3 METODOLOGIA

Seguindo a metodologia *Design Research* primeiro devem ser identificadas quais as carências existentes assim como todos requisitos que envolvem o desenvolvimento deste projeto. Assim sendo a seguinte lista representa o que foi apurado inicialmente:

- Sistema que permita melhorar a comunicação hospital/utente;
- Utilização de diretrizes eHealth que favoreçam partilha de informação;

- Utilização da base de dados MySql;
- Sistema que permita poupar dinheiro à Unidade de Saúde com a falta de comparência aos agendamentos (apointment reminder ou PRM).

Existem diversas formas de relembrar um agendamento hospitalar ou através de um email ou como *Centro Hospitalar do Porto (CHP)* atualmente faz utilizando um SMS. Neste caso o envio envolve gasto de verbas com a operadora a cobrar por cada mensagem enviada. Nos últimos anos surgiram então os dispositivos móveis com capacidade de receber avisos através de um servidor online, muitos deles gratuitos ou a um preço diminuto quando comparado com aquele cobrado pelas operadoras móveis. O campo das *mHealth* é bastante diverso, no que respeita aos tipos de aplicações que podem ser desenvolvidas, como foi abordado no enquadramento do tema. De todos esses tipos aquele que mais se enquadra com os requisitos da unidade de saúde é tipo *appointment remainder* bastante direcionado para o campo do agendamento e avisos de agendamentos implementado com sucesso nos Estados Unidos da América e grande parte do norte da Europa.

Escolhido o tipo de aplicação a desenvolver, o próximo passo e como e para que dispositivos o fazer. Como abordado anteriormente existe uma diversidade de plataformas (IOs, Android, Windows, Linux) que devem ser tidas em conta para desenvolver uma aplicação. De entre as opções apresentadas anteriormente restringiu-se as duas mais significativas no mercado Android da Google e IOS da Apple. As arquiteturas e modelos de negocio que sustentam estes sistemas operativos diferem consideravelmente [183]:

MODELO DE NEGOCIO ANDROID: a Google fornece serviços grátis que no interior contém publicidade, o *Sistema Operativo (SO)* Android é um desses serviços. Ao nível das aplicações, é um mercado aberto onde todos os interessados podem desenvolver e distribuir. Existe uma loja oficial chamada *Google Play*;

MODELO DE NEGOCIO IOS: a Apple tem *integrated model* onde o iOS está integrado no iPhone, ao nível da criação de aplicações, apenas para programadores autorizados (*iOS Developer Program*), sendo as aplicações apenas distribuídas na *App Store*. Existem empresas acreditadas pela Apple que podem desenvolver *software* e utilizar este dentro de portas.

Uma vez que Android apresenta maior facilidade para programação e difusão das aplicações foi este o SO escolhido para destino da aplicação desenvolvida. Atualmente desenvolver aplicações para este SO é mais simples que no começo, existindo diversas alternativas e uma enorme comunidade de ajuda online. O desenvolvimento é então facilitado em grande parte pelo aparecimento de *frameworks* em conjunto com IDE capazes de criar uma estrutura base transversal a todo o desenvolvimento. De acordo com G. Crowd, as melhores plataformas de desenvolvimento de aplicações móveis são escolhidas com base em 5 pontos [184]:

LINGUAGEM DE PROGRAMAÇÃO: à semelhança de IDE estes são específicos de uma linguagem de programação;

***BACK-END VS. FRONT-END*:** dependendo da função do programador, existem plataformas voltadas para o front-end enquanto outras dedicam-se mais ao back-end;

***CROSS-PLATAFORM*:** depois do desenvolvimento a forma como é feita o *deployment* é um ponto importante. Existem algumas distribuições apenas voltadas para aplicações nativas enquanto outras conseguem fazer o *deployment* de diversas versões;

ESCALONAMENTO: mais direcionada para o desenvolvimento de aplicações back-end, quando se pretende a transferência de *software* para uma *cloud* por exemplo. Ou seja, antes de desenvolver o código deve ser analisado se no futuro a transferência de código será necessária;

EMULADORES: é dos pontos mais importantes para desenvolver aplicações. Emuladores fazem a duplicação de funções associadas aos dispositivos móveis comportando-se de forma semelhante a estes.

Comparando o Android Studio, com o Xamarim ou o Visual Studio (Ionic) segundo as diretrizes da lista anterior, a última é excluída pela facto de apenas ser capaz de criar aplicações hibridas. O caso de Xamarim implicava a aprendizagem de um sistema considerado por utilizadores como complexo. De acordo com [184] numa pontuação de o a 5 android studio recebe 4.6 enquanto Xamarim 4.3 numa avaliação realizada com base em utilizadores. Existe também a possibilidade de utilizar uma expansão do IDE Eclipse que dota este da capacidade de desenvolver aplicações Android, mas a ausência de um emulador eficaz levou à não utilização deste.

Assim sendo escolheu-se Android Studio que utiliza todas as características abordadas no capítulo 2. A linguagem Java é então adaptada ao desenvolvimento Android como demonstrado anteriormente não deixando de ser uma ferramenta preciosa e versátil indispensável ao desenvolvimento deste projeto. Android Studio utiliza XML para desenvolvimento de interfaces e estruturas dinâmicas, sendo assim uma ferramenta ao desenvolvimento do trabalho.

O grande foco da aplicação está no envio de notificações para o dispositivo móvel, a melhor solução encontrada foi o Firebase Cloud Message da Google que substitui o Google Cloud Message, é uma API recente que deve ser implementada no código fonte da aplicação permitindo esta receber avisos via Internet. Para tal deve ser enviado via servidor para os recetores disponíveis online da API o destinatário (token) e o conteúdo da mensagem. Este sistema depois encarrega-se de fazer os envios.

É assim necessário criar um servidor que através de um processo de ETL faça a junção das mensagens que se devem enviar naquele dia e posteriormente faça envio de notificações

com conteúdo adaptado ao utilizador em questão. Optou-se assim pela linguagem Python devido à simplicidade e ainda pela existência de um *plugin* (PYFCM) que possibilita o envio de mensagem para Firebase sem necessidade de ser programar a editar o pedido HTTP.

Plataformas de desenvolvimento de aplicações móveis fornecem aos programadores ferramentas vitais para tal tarefa, inclusive a possibilidade de desenvolver aplicativos com capacidade de ser integradas em múltiplas plataformas(aplicações híbridas) [184]. Aplicações nativas são diretamente descarregadas da loja e instaladas e podem correr diretamente em dispositivos móveis. Estas podem também utilizar todas as funções *built-in* do aparelho onde estão instaladas como por exemplo *multi-touch* e APIs nativas [184]. Aplicações híbridas são criadas utilizando *Hyper Text Markup Language (HTML)*, CSS e JavaScript que posteriormente são traduzidas para aplicações nativas. Embora não sejam capazes de utilizar funções específicas dos dispositivos móveis conseguem utilizar a câmera, sistema de geolocalização e notificações [184].

Focando agora as atenções no processo que conduz ao envio de notificações e outras informações ao utilizador deste tipo de aplicações. Tudo começa com a adesão ao serviço na unidade de saúde.

A aplicação de registo a ser implementada na unidade de saúde, é uma *single-page application* utilizando a framework *Angularjs*. Embora o paciente esteja registado no SONHO ou noutra base de dados da unidade de saúde, é necessário que este execute o registo de forma presencial para utilizar a aplicação. À semelhança dos casos de estudo anteriores, a aplicação necessita do servidor apache para responder a pedidos por parte do navegador. A escolha deste servidor prendem-se com o tipo de máquina a qual se destina, Linux, onde este está implementado de raiz. Algumas alterações são necessárias nos ficheiros para permitir que o ficheiro *.htacess* consiga fazer o *routing* interno. A criação de servidor *restful* em *Hypertext Preprocessor (PHP)*, prende-se com a utilização da base de dados MySql e com fato to tipo REST ser aquele mais utilizado para aplicações móveis já que não afeta o tempo de processamento despendido pela aplicação.

Uma característica importante desta aplicação prende-se com a confirmação do email e posteriormente do número de telefone. O esquema definido com a instituição de saúde foi o seguinte:

1. Registo na plataforma;
2. Confirmação do email;
3. Envio de *password* para telemóvel e confirmação do mesmo;
4. Utilização da aplicação.

Após registo para enviar o email, criou-se uma função *send_email* com função que recebe uma *string* gerada automaticamente pelo código JavaScript e envia o email com o *link* para

confirmação. Após a correta validação do email o utilizador recebe uma SMS no telemóvel com a *password* gerada automaticamente para validar assim no seu dispositivo móvel.

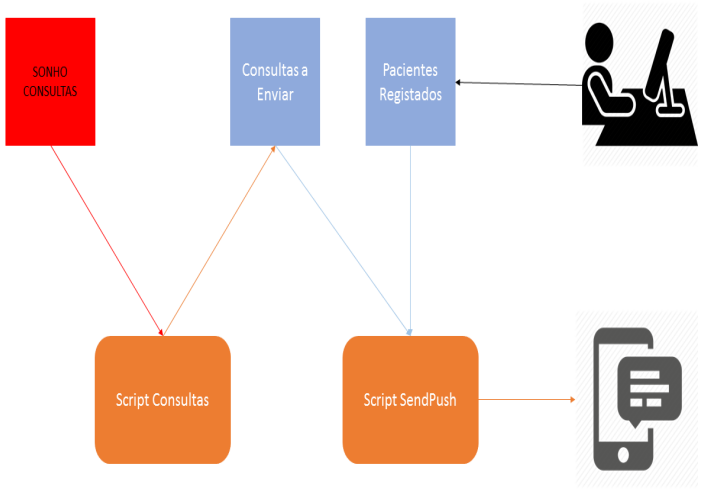


Figura 23.: Esquema implementado para o envio de notificações.

A figura 23 apresenta um *overview* de todo o processo desde ETL até ao utilizador final. Este processo consiste num script executado diariamente que necessita dos respetivos conetores às base de dados para desempenhar o seu papel corretamente. Basicamente é feita *querie* na tabela com informação de todos os agendamentos para os próximos 7 dias e desses são retirados os que estão devidamente registados na plataforma, para carregamento numa outra tabela para posterior processamento. A chave primária destas tabelas é o número telemóvel visto ser um identificador único e inequívoco de determinado paciente, existindo assim a necessidade de rever informações nas tabelas SONHO, uma vez que nem todos os pacientes apresentam informação do contacto de telemóvel registado.

No final do processo descrito anteriormente é lançado o script responsável por enviar notificações. Para realizar tal processo duas tabelas são necessárias, a tabela preenchida com o processo anterior, e a tabela preenchida no momento de registo do paciente na unidade de saúde, como é possível observar no canto superior direita da figura 23. Novamente as chaves primárias são o número de telemóvel. Foi seguido o modelo de desenvolvimento proposto pela Google e *Firebase Cloud Message (FCM)* para servidores com interação direta via pedido HTTP aos servidores da empresa. Quando o script responsável pelo envio de notificações percorre a tabela de avisos a enviar, utiliza o número de telefone como chave primária e *token* para interagir com servidor FCM, enviado um texto fixo com mais ligeiras alterações consoante as informações do paciente. Este pequeno programa desenvolvido na

linguagem Python, utiliza uma extensão ao código base chamada PyFCM, que simplifica o trabalho de programador no pedido HTTP [185].

O *token* assume papel preponderante no bom funcionamento da aplicação, uma vez que sem este não é possível contactar o dispositivo móvel. Para tal é necessário criar uma classe que estenda *FirebaseInstanceIdService*. O *token* altera sempre que uma das seguintes atividades ocorrer, aplicação eliminar *Instance ID*, reinstalada no mesmo dispositivo, nova instalação num dispositivo novo ou dados da aplicação forem apagados. Dentro do código da aplicação a função que faz *retrieve* do novo código é *FirebaseInstanceID.getToken()*. Com vista a manter o *token* sempre atualizado, na tabela de pacientes registados um campo é destinado a este código. Sempre a aplicação se inicia, utilizando um pedido HTTP imediato ao servidor *Restful* o código é atualizado na base de dados.

Ao nível de interfaces e projetos, o *android studio* disponibiliza um conjunto de *templates* que podem ser utilizados para desenvolver aplicações simples. Seguindo a tendência da Google no desenvolvimento das suas próprias aplicações, o modelo escolhido foi *Navigation Drawer Activity* [186]. Basicamente existe uma barra lateral retrátil onde podem ser encontradas todas as opções de navegação existentes na aplicação. Este modelo foi escolhido pela simplicidade e facilidade de utilização e ainda pela vasta existência de documentação online. Existe então uma atividade principal (*MainActivity*) que posteriormente carrega fragmentos, um pouco à semelhança das *single-page applications*. Foram assim criados 5 fragmentos que correspondem às 5 funcionalidades básicas da aplicação móvel. A somar a isto, as atividades de *login* e alteração de dados que devem ser incluídas em atividades em separado.

Para escrever uma aplicação com *Firebase Cloud Messaging Android* foi seguido o modelo proposto pela Google no site oficial [187]. Assim sendo na aplicação deve ser acrescentado API *FirebaseMessaging* utilizando para tal *Gradel*. Importante referir que FCM implica uma versão Android superior a 2.3 [187]. Alguns dos passos envolvidos na implementação são:

1. Criar no centro de projetos da Google um novo projeto para esta aplicação. Após isto, ficam disponíveis na *Firebase Console* diversas chaves que identificam o projeto.

   CÓDIGO DE PROJETO: chppacientes;

   CHAVE DA API: AIzaS...;

   API TOKEN: identifica inequivocamente um dispositivo.

2. Adicionar *Firebase* ao projeto Android, nomeadamente o ficheiro de configuração (google-services.json) e a adição de permissões aos diversos ficheiros de configuração;

3. Criar classes JAVA na forma de serviço que faça a extensão *FirebaseMessagingService*, que permite realizar operações que envolvam a mensagem para além de receber mensagem em *background* [187];

4. Criar classe JAVA que estende o *FirebaseInstanceService* com vista à criação e atualização de *tokens*. Apenas com estas *strings* em forma de código é possível enviar uma notificação para o paciente de forma individualizada;

5. É necessário adicionar ao manifesto a seguinte cláusula "android:minSdkVersion=8", assegurando assim que a aplicação não é instalada num sistema que não consiga receber mensagens de notificação.

A sustentar tanto os pedidos HTTP por parte da plataforma web como da plataforma móvel está o Web Service do tipo Restful. Optou-se pela linguagem PHP por questões relacionadas com a simplicidade da linguagem e de implementação do servidor. Esta linguagem apresenta afinidade natural para bases de dados MySql não sendo necessária a utilização de um conetor específico e por último o facto de a máquina de destino ser Linux com distribuição Ubunto, a utilização do Apache com Php faz todo o sentido sendo apenas necessária uma pequena configuração nos ficheiros.

7.4 RESULTADOS E DISCUSSÃO

O sucesso da implementação de uma aplicação móvel pode ser medida pelo número de utilizadores ou número de *downloads* nas lojas oficiais. A aplicação criada visa chamar a atenção de determinado utente para o facto de num futuro próximo ter uma marcação no hospital. A vantagens para o utilizador prende-se com o aumento da comodidade e da proximidade com sua unidade de saúde ficando com a sensação que esta se preocupa com seu bem estar. Para a unidade de saúde o principal resultado obtido é um sistema que lhe permite evitar em parte faltas de comparência a marcações clínicas e ainda poupar orçamento uma vez que enviando notificações para telemóvel não é necessário fazer avisos através de SMS.

A figura 24 apresenta a aplicação implementada embora ainda num protótipo inicial. A figura contém 4 aspetos que vão ser abordados de seguida. As abas implementadas que constituem a aplicação são (ver telemóvel mais à esquerda na figura 24):

VISTA INICIAL: contendo informações básicas do paciente e ainda o botão que permite ao paciente confirmar a sua presença no agendamento. Este botão desencadeia um conjunto de ações na base de dados semelhantes ao processo que o funcionário desempenha na secretaria (ver telemóvel número 4 da figura 24);

HISTÓRICO DE AGENDAMENTOS: seguindo o formato de lista *(feed de noticias)* apresenta informações como data, hora, especialidade e se de facto se realizou ou não relativamente a agendamentos passados. A lista começa no item mais recente e termina naquele que está mais afastado relativamente à data atual (ver telemóvel número 3 da figura 24);



Figura 24.: Resultado final da aplicação Android.

AGENDAMENTOS FUTUROS: seguindo o formato de lista *(feed de noticias)* apresenta informações como data, hora, especialidade, nome do médico e localização onde decorrerá o agendamento. A informação começa pelos mais próximos até aqueles mais longínquos relativamente ao momento presente(ver telemóvel número 4 da figura 24);

INFORMAÇÕES DA UNIDADE DE SAÚDE: aba bastante simples apenas com informação da localização da unidade e ainda contactos básicos como número de telefone ou email.

Os resultados do desenvolvimento deste caso de estudo do ponto de vista do utilizador são:

- Maior proximidade com a unidade de saúde;
- Maior comodidade na marcação ou alteração de situações hospitalares;
- Organização de informação sobre paciente e familiares;
- Consulta do seu histórico dentro da unidade hospitalar;
- Consulta das próximas marcações dentro da unidade hospitalar.

Para os resultados enumerados na lista anterior em muito contribui o processo de ETL implementado, utilizando o número de telemóvel como identificador único é possível um utilizador que pode não ser paciente receber notificações sobre alguém que de facto o é. Por exemplo, pensando nos idosos e crianças que não possuem dispositivo móvel, algum familiar regista o seu número e assim recebe os avisos. É assim possível uma aplicação com apenas um ID de utilizador ser destinada a diversos utentes.

Sem dúvida que o grande aspeto inovador desta implementação é a notificação ou Push Notification, o exemplo do aviso enviado pode ser observado no telemóvel número 4 da figura 24. Enviado pelo script em Python para servidor FCM, o utilizador recebe no dispositivo móvel o aviso acompanhado de sinal sonoro se este estiver ativo. Geralmente na

barra superior localiza-se um icon, e deslizando a barra é possível observar o texto da notificação. Ao clicar no aviso, faz um *update* na base de dados indicando que utilizador já foi informado do seu agendamento.

A implementação da aplicação não é muito complicada uma vez que o *design* implementado é disponibilizado pela Google sendo de distribuição livre pode ser utilizado em qualquer tipo de aplicação móvel. Claro que para correr perfeitamente é necessário um servidor RESTFUL que responda aos pedidos enviados diretamente pela aplicação. A aplicação de registo (ver figura 38 em anexo) partilha o mesmo servidor anterior com pedidos completamente diferentes, neste caso registo e confirmação de *email* e *password*. A aplicação foi criada utilizando o Angularjs e conta com 2 páginas, uma para inserção de um novo utilizador outra com para aceder a todos os utilizadores da aplicação e data que aconteceu a adesão ao serviço. Primeiro existe a inserção de dados, como número do Serviço Nacional de Saúde, telemóvel e email. Com estes dados facilmente através de *querie* no *Sistema de Informação para a Gestão de Doentes (SONHO)*, é garantida a totalidade da informação. Quando o pedido é enviado, o utilizador deve receber um mail para proceder à confirmação.

A implementação do processo em Python utiliza um script bastante simples sem necessidade para a utilização de *multiprocessing*. O mesmo resultado poderia ser alcançado com uma *querie* diária realizada de forma manual mas as implicações de consumo de recursos põem de parte imediatamente esta opção. Um pormenor importante desta implementação é facto de utilizador o conetor Oracle para extrair dados e MySql para carregar dados no modelo relacional demonstrado.

7.5 CONCLUSÃO

Com a evolução dos dispositivos móveis o mercado das aplicações é hoje uma fonte de rendimentos para aqueles que se dedicam ao seu desenvolvimento. A área da saúde não pode passar ao lado desta nova corrente de desenvolvimento, e *eHealht* apresenta um "ramo" orientado para tal denominado *mHealth*. Existem diversas áreas que servem de mote ao desenvolvimento de aplicações como monitorização de pacientes, motivação e trabalho de memória utilizando aplicações e ainda diversos aspetos das atividades de uma unidade hospitalar.

A falta dos pacientes a uma marcação clínica é uma das situações onde uma unidade de saúde mais perde dinheiro. Desenvolveu-se assim uma aplicação do tipo *appointment reminder* que pretende alertar assim o paciente de que tem marcação de modo a que este não falhe ao mesmo.

O sucesso deste tipo de implementações depende em grande parte do número de utilizadores quando estiver disponível, mas uma vez que era um protótipo para ser submetido junto da direção do Centro Hospitalar do Porto o sucesso passa pela sua aceitação ou não.

O projeto entretanto obteve luz verde seguindo assim para próxima fase de desenvolvimento onde é criado um protótipo definitivo e começando primeiros testes em ambiente real.

O trabalho futuro passa então pela redefinição de alguma da informação presente nas abas iniciais e de informações da unidade hospitalar uma vez que no primeiro caso tentara ser implementado um sistema que utilizando a posição geográfica do dispositivo móvel fazer um *match* com coordenadas do hospital confirmando a sua presença no agendamento. De referir que esta confirmação só é possível em casos de pacientes isentos de taxas moderadoras uma vez que os outros devem efetuar o pagamento na secretaria ou junto do *kiosk*. No segundo cenário as informações devem ser um pouco mais interativas utilizando por exemplo sistema de *e-mail* direto ou efetuar chamadas automaticamente. Existe também a necessidade de adicionar um novo módulo que permite remarcar ou cancelar determinado agendamento daí que seja necessário implementar um sistema que leve ao contacto direto com a unidade de saúde.

Numa abordagem mais técnica será implementado um novo modelo relacional recorrendo ao *cloud computing* e aproveitando as funcionalidades da *framework* Firebase em conjunto com a Google.

A apresentação de informação utilizando o estilo *feed* de noticias faz com que a informação seja apresentada de forma clara e fluída e ainda que o carregamento dos dados não seja efetuada toda de uma vez, mas consoante o *scroll* por parte do utilizador.

O impacto da utilização deste caso de estudo para a unidade de estudo e utilizador são evidentes. No primeiro caso, a poupança com consultas ou outros agendamentos que não comparece utente é evidente e ainda um ganho na proximidade com o utilizador e diminuição de filas de espera para confirmar a presença. No caso do utilizador, este ganha em comodidade, simplicidade de processos (remarcação e cancelamento) e sensação de que a unidade de saúde está próxima e interessada no seu bem estar. Existem alguns aspetos legais que devem considerados, por exemplo, pode ser importante apresentar um resumo do que foi efetuado em agendamentos anteriores sendo necessário apresentar informação que até ai era confidencial.

A organização mundial de saúde dá grande ênfase ao desenvolvimento de aplicações relacionados com *mHealht*, sendo a maior parte das implementações vocacionada com a monitorização de pacientes à distância. Nesta abordagem optou-se pelo tipo *appointment reminder* uma vez ser aquele que mais contribui para a disseminação de informação e ainda fazer a ponte para fora das fronteiras do hospital.

8

PROVA DE CONCEITO

Um projeto na área dos sistemas de informação deve ser submetido a testes antes de estar disponível para utilizadores. Um pouco à semelhança do que acontece no desenvolvimento de *software*, onde existem diversas versões antes daquela considerada oficial para lançamento. Com este teste sabe-se assim se os objetivos delineados no início do projeto foram ou não atingidos e neste caso o que fazer para que tal aconteça.

Com a inserção de uma *Proof-of-Concept (POC)* ou *Proof-of-Principles (POP)*, os custos com produtos incapazes são reduzidos ao mínimo, uma vez que, através da sua realização em fases intermédias do desenvolvimento, obriga à tomada de decisões antecipadamente [188].

Na presente dissertação, o processo de POC realizou-se para cada caso de estudo de forma individualizada, através da análise SWOT assim como um breve estudo de usabilidade para o caso de estudo do capítulo 4.

O presente capítulo, aborda numa primeira fase os pressupostos teóricos que servem de base a estes tipo de testes, seguindo-se de forma individualizada a análise SWOT implementada. Uma vez que no caso do teste de usabilidade ainda não existem resultados suficientes para conclusões, será apenas demonstrado o questionário elaborado ao utilizador. Esta análise pode ser enquadrada na fase de avaliação das soluções desenvolvidas do ciclo *Design Research*, como uma medida para provar que cumpre os requisitos iniciais, tornando essas fraquezas em forças.

8.0.1 *Análise SWOT*

Enquadramento Teórico

A análise SWOT é uma das várias ferramentas que as organizações dispõem para *strategic planning* e POC, garantindo assim que o objetivo delineado inicialmente permanece intacto e todas as medidas tomadas para o atingir, quer positivas ou negativas, estão bem identificadas [189]. O processo envolve assim 4 áreas: forças (*strengths*), fraquezas (*weaknesses*), oportunidades (*opportunities*) e por último ameaças (*threats*) [189]. Quando é realizada uma análise devem ser tidos em conta factores internos e externos que podem fazer cair por

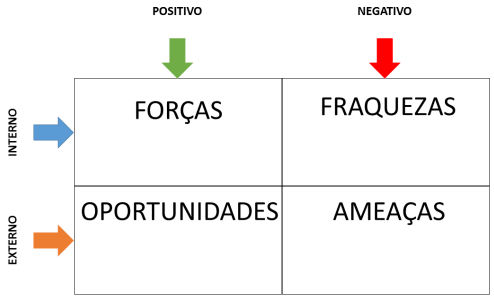


Figura 25.: Esquema geral da análise SWOT (adaptado de [189]).

terra ou ser chave para o sucesso da aplicação. Para utilizadores com pouca experiência, definir os tópicos em cada um dos pontos apresentados é uma tarefa algo complicada.

De acordo com Ifediora Christian Osita em *"Organization's stability and productivity: the role of SWOT analysis an acronym for strengthm weakness, opportunities and threat"*, a análise SWOT é *"the most renowned tool for audit and analysis of the overall strategic position of the business and its environment...it is the foundation for evaluating the internal potential and limitations and the probable/likely opportunities and threats from the external environment. It views all positive and negative factors inside and outside the firm that affect the success."* [189]. Pela citação é possível perceber que este tipo de análise representa a fundação do planeamento estratégico de muitas empresas ou projetos, conseguindo perceber quais factores positivos e negativos, dentro ou fora das organizações, bem como possíveis ameaças ao desenrolar de todo o trabalho.

A figura 25 representa assim o esquema geral do processo da análise SWOT. Muitas vezes, esta é organizada no formato de matriz 2×2 com Forças (S) e Fraquezas (W) na primeira linha e Oportunidades (O) e Ameaças (T) na segunda. Ao dispor os elementos desta forma apresenta uma melhor clareza entre aspetos positivos e negativos (coluna 1 ou 2) e aspetos de natureza interna e externa ao projeto (linha 1 ou 2) [189].

Descrevendo agora os componentes da figura 25:

FORÇAS: representando as qualidades que permitem ao projeto alcançar os seus objetivos;

FRAQUEZAS: representando as qualidades que impedem o projeto de alcançar os seus objetivos;

OPORTUNIDADES: diz respeito ao ambiente onde o projeto está inserido. São então abordados aspetos desse ambiente que podem trazer benefícios para o projeto fazendo este evoluir;

AMEAÇAS: diz respeito a condições do ambiente que podem pôr em risco o desenvolvimento do projeto.

Este método é então uma simples e barata ferramenta de avaliação de projetos ou empresas que facilmente pode ser adaptada aos casos de estudo apresentados ao longo da presente dissertação.

Análise SWOT Sistema HL7

Relembrando este caso de estudo destina-se ao envio e receção de mensagens sob os padrões *Health Level Seven International (HL7)*. Foi desenvolvido um motor de interfaces que sustenta agentes inteligentes numa implementação MAS.

Os pontos fortes encontrados foram:

- A utilização do motor de interfaces simplifica a manutenção;
- A linguagem Python traz capacidade de manutenção na atualização de versões;
- Ligação com a plataforma de monitorização;
- Capacidade de controlo por aplicação web;
- Recuperação do erro sem a necessidade de intervenção humana;
- Recolha de informações sobre a performance agentes;
- Utilização da estrutura JSON para comunicação via *socket*;
- Recolha de informações sobre o impacto do agente na máquina onde está inserido.

Como não poderia deixar de ser a aplicação tem pontos menos bons considerados fraquezas. Sendo assim, os pontos fracos a ser considerados são:

- Requer ligação à Internet;
- Requer a interação com base de dados o que pode causar erros;
- Implementação numa nova máquina, requer a perceção de como os sistemas comunicação funcionam (*Internet Protocol (IP)* e porta);
- Exige módulos Python que devem ser adicionados utilizando Internet;
- O módulo da recuperação de erro carece de melhoramentos;
- Definição da porta e IP cria dificuldades na configuração;
- Medição de tempo entre receção e integração no programa carece de ser melhoramentos;
- A comunicação entre agentes é algo complexa.

Seguindo o modelo SWOT, são agora abordadas questões exteriores à aplicação, nomeadamente o ambiente onde esta está implementada. As oportunidades são pontos positivos, que não podem ser controlados. Entre estes destacam-se:

- Necessidade de partilhar informação;
- A implementação de um modelo inovador, utilizando motores de interface, é capaz de modernizar o sistema atual;
- Disponibilização gratuita de todos os materiais necessários ao desenvolvimento e implementação;
- Crescente interesse da comunidade cientifica em agentes MAS;
- Necessidade da existência de interoperabilidade sintática para posterior implementação de outros níveis.

Em contraponto, as possíveis ameaçadas identificadas são:

- Falta de aceitação pelos profissionais dos sistemas informativos;
- Concorrência por parte do sistema já implementados e outros desenvolvidos comercialmente;
- Falta de acesso externo à Internet para possíveis correções à distância ou instalação de novos módulos;
- No caso de estudo foi referido que mais de 70% dos responsáveis não pretendem alterar a sua interface HL7, resulta assim uma ameaça à implementação do trabalho.

Análise SWOT Sistema Monitorização

Este caso de estudo reporta a implementação de uma aplicação web com a função de sistema de monitorização para agentes AIDA e HL7 já instalados no Centro Hospitalar do Porto e ainda desempenha o papel de plataforma de controlo de agentes que haviam resultado do caso de estudo anterior. De realçar que ao mesmo tempo existe em *background* sistema de *Extract, Load, Transform (ETL)* a processar informação recolhida.

Os pontos fortes da aplicação são:

- Geração de informações que facilitam a tomada de decisão;
- Apresenta um módulo desenvolvido especialmente para projeção em monitores com *reload* automático da página;
- Módulo dedicado ao *Business Intelligence (BI)* com informações sobre erros em mensagens e volume de mensagens processadas por agente;

- Acesso à distância de um *click* a informação detalhada sobre cada agente em atividade incluindo a probabilidade de erro;
- Módulo *Agéncia para a Integração, Difusão e Arquivo de Informação Médica (AIDA)* permite receber mensagens de aviso gerados pelo processo de ETL;
- Design simples e atrativo;
- Utilização do sistema de *login* que recorre ao sistema de *TOKEN* garantindo assim a segurança dos dados;
- Implementação de um processo de ETL que simplifica o acesso aos dados por parte da aplicação;
- Histórico de mensagens com erros à distância de um *click*.

As fraquezas identificadas foram:

- O sistema de ETL deve ser simplificado uma vez que por vezes demora a terminar;
- Ausência de um processo de ETL específico para a geração de indicadores;
- O sistema de envio de notificações para telemóvel precisa de acesso à Internet exterior;
- Algumas interfaces demoram algum tempo a carregar;
- O processo de ETL só reporta os últimos três dias o que pode ser pouco em casos onde é necessário maior volume de dados;
- Nem todas as mensagens apresentam aplicação que enviou ou recebeu, aquando do processo de ETL;
- O processo destinado a agentes AIDA está implementado apenas para AGENTESLAB e HL7;
- Poucos indicadores na aba de BI;
- A identificação do motivo causador do erro em caso mensagens ou agentes necessita melhorias.

Os aspetos do ambiente que são uma influência positiva para aplicação são:

- A modernização e desenvolvimento organizacional;
- A expectativa elevada por parte dos profissionais dos sistemas de informação sobre monitorização e controlo de agentes inteligentes;
- A crescente utilização de ferramentas de apoio à decisão;

- Ferramentas de utilização gratuita;
- O *template* Angularjs que permite implementações mais complexas;
- Utilização de fontes de informação já implementadas na unidade de saúde, no caso dos agentes AIDA e HL7.

Por último, os aspetos do ambiente que podem ser influência para a aplicação são:

- A utilização da Internet externa;
- A falta de recursos computacionais das máquinas em produção para sustentar o processo de ETL;
- A má configuração de alguns agentes HL7 que motiva algumas mensagens sem aplicação de envio ou receção;
- A utilização de credenciais de acesso AIDA;
- Concorrência de sistemas comerciais.

Análise SWOT Formulário ICD9

A necessidade de um sistema interativo para levar a cabo a codificação de altas médicas para posterior utilização em sistema *Grupo de Diagnósticos Homogêneos (GDH)* serviu de mote ao desenvolvimento deste caso de estudo.

Os pontos fortes desta aplicação podem ser agrupados na seguinte lista:

- O sistema apresenta uma interface apelativa e de fácil perceção;
- A implementação de diferentes modos de codificação aumenta a simplicidade e a comodidade;
- Gravação automática pode ser vital para recuperar em caso de erro;
- Pesquisa dinâmica e Tooltip em cada caixa para acelerar o processo de codificação;
- Capacidade de visualizar o relatório de alta ao mesmo tempo que é efetuada a codificação;
- Possibilidade de ver PDF com resumo de alta na mesma página onde se desenrola a codificação;
- Informações básicas à distância de um *click*.

Os pontos fracos são assim agrupados de igual forma:

- A pesquisa dinâmica apresenta falhas e demoras;

- Existem algumas situações em que o relatório de alta não carrega devidamente;
- Lentidão ou erros quando uma linha é eliminada;
- Utilização de códigos *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* ao invés dos mais atuais ICD-10-CM;
- Ausência de restrições para associações entre diagnósticos, morfologia e causas externas;
- Falta de opção para exportar a codificação em formato PDF.

Em relação ao ambiente onde a aplicação está inserida, destacam-se as seguintes oportunidades:

- Os códigos ICD-9-CM são um dos sistemas de codificação mais utilizados em todo o mundo;
- Motivação da unidade de saúde para a implementação de um sistema GDH;
- Possibilidade de criar métricas para o desempenho de médicos codificantes;
- Utilização de ferramentas gratuitas;
- Utilização da base dados ICD-9-CM com toda a informação detalhada e organizada.

Já as ameaças ao projeto identificadas são:

- Falta de interesse ou empenho no projeto;
- Concorrência de processos *paper-based*;
- Alguns navegadores não conseguem utilizar todas as funcionalidades da plataforma desenvolvida.

Análise SWOT AIDA_Paciente

Este projeto pretende desenvolver uma aplicação direcionada para os paciente um pouco à semelhança das *appointment remainder*. Existe aviso por sistema Push Notification das consultas, cirurgias e outras marcações médicas. É importante referir também que existem processos Python que correm em *background* para encontrar pacientes com consultas e enviar os avisos;

Os pontos fortes:

- Utilização de interface típica das aplicações Google com um design simples e minimalista;

- Aplicação com capacidade de ser adaptada a qualquer unidade de saúde com ligeiras alterações;
- Processo Python que suporta a aplicação é de simples manutenção;
- A utilização de serviços Google Firebase permite um fácil envio de mensagens;
- Organização da informação em listas para fácil perceção do conteúdo;
- Aplicação desenvolvida na língua Portuguesa;
- Avisos Push garantem proximidade do paciente com a unidade de saúde.

Existem também pontos negativos nesta implementação como em todas as outras. Assim sendo, pode ser organizada uma lista:

- O facto de nesta versão ser utilizado registo presencial pode levar ao desinteresse por parte do utilizador;
- O conteúdo dos avisos Push pouco atrativo;
- Ausencia de um mecanismo para saber se o Push Notification já foi lido e informe o sistema de tal;
- Ausência de serviços de cache da aplicação atrasa a apresentação de dados.

Os fatores ambientais externos que favorecem a aplicação são:

- Crescente interesse em aplicações móveis;
- Dispositivos móveis cada vez mais evoluídos assim como Internet móvel e sistemas de rede Wireless;
- Interesse em aplicações relacionadas com área da saúde;
- Informação necessária está contida no SONHO já existente nas unidades de saúde;
- Utilização de ferramentas Google, gratuitas e com enorme qualidade;

Porém, o ambiente pode também trazer pontos negativos para o projeto como:

- Desinteresse para com a aplicação ou aviso por parte da população;
- Concorrência de aplicações como calendário ou agendas;
- Soluções já presentes nas lojas de aplicações.

8.0.2 *Estudo de Usabilidade*

A implementação de um questionário destina-se a obter um feedback por parte dos utilizadores sobre a usabilidade e as funcionalidades da plataforma web para monitorização de agentes AIDA e HL7. Uma vez que ainda não existem dados para apresentar resultados conclusivos, apenas se demonstra quais as perguntas realizadas assim como as diversas opções de resposta. Ao longo do processo de desenvolvimento do sistema de monitorização foi essencial a ajuda do professor Doutor José Machado na depuração de alguns erros não perspectiveis numa interação menos especializada.

Existem propósitos para cada uma das perguntas elaboradas (ver tabela 1), o tipo de utilizador permite saber quem mais utiliza as páginas e qual dentro de cada tipo é a aba que mais lhe interessa. A aba considerada menos importante pode então ser remodelada e alterada consoante os comentários tecidos pela pessoa que preencheu o questionário. Com as primeiras perguntas é avaliada a capacidade de processamento e apresentação da plataforma, caso as respostas sejam negativas pode ser lançada nova versão com *layout* diferente.

Em suma, o questionário pretende inquirir mais do que pontos fortes, ou pontos onde aplicação apresenta deficiências de alguma espécie para num futuro próximo poder ser corrigidas. Embora existam opiniões diferentes nestes casos, desenvolver um módulo de BI relacionado com a utilização da plataforma seria interessante e inovador.

Questões Presentes no Questionário	
Questão	Respostas Possíveis
Avaliação Global da Plataforma	• Muito Bom • Bom • Mau • Muito Mau
Design e Aspeto Visual	• Muito Bom • Bom • Mau • Muito Mau
Rapidez e Eficiência	• Muito Bom • Bom • Mau • Muito Mau
Ajuda no Trabalho Desenvolvido	• Muito • Média • Má • Muito Má
Aba Mais Importante	Todas as abas presentes no trabalho
Aba Menos Importante	Todas as abas presentes no trabalho
Tipo de Utilizador	• Administrativo • Sistemas de Informação • Médico • Externo
Comentários/Recomendações	Campo para inserção de texto

Tabela 1.: Apresentação das perguntas elaborados no questionário ao utilizador

9

CONCLUSÕES

O projeto desenvolvido no âmbito da presente dissertação, tem por objetivo identificar os pontos relevantes e apresentar novas abordagens que possam ser introduzidas nos diversos sistemas de informação do *Centro Hospitalar do Porto (CHP)*, como forma de aumentar a interoperabilidade existente. A importância da partilha de informação por parte das instituições prestadoras de cuidados de saúde é inquestionável sendo necessário desenvolver soluções que elevem os padrões da saúde em Portugal para novos patamares.

A primeira parte do projeto naturalmente foi dedicada à pesquisa bibliográfica como forma de realizar o enquadramento dos diversos temas de acordo com os trabalhos desenvolvidos a nível nacional e internacional. Foram abordados conceitos de *Sistemas de Informação (SI)* , *Health Information Systems (HIS)*, *Processo Clínico Eletrónico (PCE)* entre outros, como forma de avaliar a situação atual nas instituições de saúde para posterior abordagem da partilha de informação entre estes. Interoperabilidade é o ponto central neste projeto, visto marcar a forma como se procede à troca de informação de forma a que os diversos sistemas consigam perceber aquilo que está a ser partilhado para trabalhar em conjunto. De acordo com a bibliografia é possível definir três camadas de interoperabilidade (**Sintática, Semântica, Processo**) assim como avaliar a interoperabilidade de acordo com o modelo *LCIM*, em seis níveis (o para **inexistente** e 6 para **interoperabilidade conceptual**). O tipo semântico só é alcançado quando os dois sistemas que comunicam são capazes de perceber o conteúdo das mensagens. Para tal, existem as codificações de transporte (**HL7 Standards**) e clínicas (**ICD-9-CM**) que servem de suporte aos casos de estudo dos capítulos 4 e 6. A interoperabilidade semântica hoje em dia representa um ponto que nem sempre é alcançado e que influência em grande parte a difusão de informação. Por outro lado, os níveis que a precedem são em Portugal um bem adquirido, uma vez que todas as instituições já utilizam meios (*networks* internas) para difundir informação. Por outro lado, a interoperabilidade de processo é bastante difícil de alcançar uma vez que, deve envolver toda a organização que requer bases bastantes solidas para gerar nesta confiança.

Naturalmente os conceitos de *Mobile Health (mHealth)* e *Electronic Health (eHealth)* devem ser revisitados. A maior parte dos trabalhos nesta área, são implementações na área da monitorização de pacientes à distância assim com treino para futuros médicos. No caso

de estudo do capítulo 7 foram abordados conceitos ligados a aplicações do tipo *Appointment Reminder*. Existe naturalmente uma propensão natural para desenvolver aplicações vocacionadas para o *Sistema Operativo (SO)* android, visto ser de distribuição livre com o maior número de vendas registadas anualmente. Para além dos *appointment reminders*, ligados ao conceito de *mHealth*, o *Patient Relationship Manager (PRM)* serve também de mote ao desenvolvimento dos caso caso de estudo do capítulo 7. Por último, referir que esta implementação não seria possível sem interoperabilidade sintática e semântica como referido anteriormente.

Dos conceitos abordados, aquele que apresenta menor ligação ao que atualmente está implementado nas unidades de saúde em Portugal é a indústria 4.0. A inserção deste, prende-se com a tentativa de perceber para onde a tecnologia e as correntes de pensamento estão a evoluir, nomeadamente com os *Cyber-Physical Systems* que servem de base aos *medical Cyber-Physical Systems*. Desta forma, o sistema MAS implementado apresenta as capacidades destes sistemas, uma vez que cada agente comunica o seu estado num formato de "rede social". Por outro lado, os conceitos de *pervasive e ubiquitous computing* estão relacionados com perceção do meio e de como inserir estes de forma que sejam quase impercetíveis na recolha e processamento de dados, levando à informação.

Apesar do carácter independente dos capítulos, foram colocadas diversas questões durante o capitulo de introdução que devem ser respondidas ao longo do trabalho. De forma a fazer ponte entre todos os conceitos abordados e tecnologias utilizadas, as questões são respondidas ao longo das próximas sub secções.

9.1 questão 1: qual é a evolução dos sistemas de informação nas unidades de saúde e qual é a importância da interoperabilidade para estes?

O termo informação só pode ser utilizado quando os dados têm a capacidade de chegar ao seu destinatário no momento correto, com a precisão e detalhe necessários, caso contrário, são apenas *strings* ou números sem sentido aparente. Assim sendo é imperativo que os SI para além do armazenamento de dados, tenham a capacidade de os apresentar ao utilizador a qualquer instante.

Os primeiros SI eram *paper-based*, ainda hoje alguns consultórios ou unidades se saúde mais pequenas utilizam este método sendo as desvantagens por demais evidentes. Com o surgimento e posterior desenvolvimento de sistemas computacionais, os registos começaram a ser realizados em formato digital. Durante alguns anos desenvolveram-se ferramentas individualizadas para cada serviço ou setor de uma unidade de saúde, visto ser essa a principal necessidade na altura. As implicações desse desenvolvimento são hoje mais que evidentes uma vez que foram criadas ilhas de informação sem qualquer partilha da mesma.

Principalmente com o desenvolvimento da *Internet* e consequente IoT, novas soluções foram desenvolvidas dedicadas à partilha de informação entre sistemas. As primeiras implementações apenas atingiam níveis de interoperabilidade técnica, disso é exemplo o envio de informação pelo correio eletrónico. O desenvolvimento de sistemas que permitam o envio e receção de mensagens em formato HL7 v2 foi sem dúvida um marco no que à interoperabilidade sintática diz respeito. A título de exemplo nos Estados Unidos da América 90% das instituições utilizam estas diretivas estando por demais enraizadas em diversas instituições espalhadas pelo globo. A terceira versão para troca de mensagens não obteve a mesma aceitação por parte dos profissionais uma vez que não existe ponte possível a versão anterior.

Embora a comunidade científica tenda a avançar para utilização de Cloud Computing ainda em muitas unidades de saúde não se atinge o nível de interoperabilidade semântica. Soluções como *Agéncia para a Integração, Difusão e Arquivo de Informação Médica (AIDA)* são formas de alcançar a partilha de informação sem necessidade de reformular todos os sistemas implementados ao longo dos anos. Atualmente os sistemas informativos estão a começar a utilizar sistemas multi agente com autonomia e capacidade de cooperação, resultando numa capacidade de trabalho bastante superior às demais. Por outro lado, abordagens ao mHealth, possibilitam o acesso a informação utilizando apenas um telemóvel por exemplo.

Respondendo à segunda parte da pergunta referente à importância da interoperabilidade, esta é vital. Na configuração dos SI modernos é essencial que exista interoperabilidade pelo menos do tipo sintática. Por outro lado, soluções verdadeiramente funcionais só são alcançáveis quando existe interoperabilidade do tipo semântico. Com a Internet, soluções como web services ou *sockets* permitem a partilha de informação de forma rápida e quase livre de erros. Porém ainda hoje, nem todos os sistemas se percebem plenamente. Pela importância para o presente projeto, a interoperabilidade é transversal a este, sendo a questão de certa forma respondida ao longo de todo o trabalho.

9.2 Questão 2: Como melhorar a interoperabilidade existente nas unidades de saúde, nomeadamente no CHP?

Como abordado no capítulo 2 , existem diferentes níveis de interoperabilidade segundo diferentes ideologias. Como seria de esperar os níveis mais avançados serão aqueles mais visados por trabalhos científicos. Nas unidades de saúde em Portugal a interoperabilidade semântica ainda apresenta uma ramificação algo limitada (**nível 2 ou interoperabilidade do tipo semântico**). Motivada em parte por anos de desenvolvimento de SI modular onde não era necessária a partilha de dados e consequente perceção dos mesmos. Em Portugal o Sistema Nacional de Saúde, tem proposto diversas diretivas no sentido de aumentar

a interoperabilidade e quantidade de implementações relacionadas com mHealth (Plano Nacional de Saúde 2009-2016).

A melhor forma de fomentar a interoperabilidade entre HIS é continuar a desenvolver soluções criativas, com baixo custo monetário e de processamento, que sejam capazes de ser implementadas a curto prazo nas unidades de saúde. A utilização do *standard* v.2 para troca de mensagens em formato *Health Level Seven International (HL7)* é amplamente utilizada, onde os principais trabalhos desenvolvidos atualmente estão vocacionados para o envio de grandes quantidades de dados como acontece em imagiologia. Existe também o contínuo melhoramento das interfaces já implementadas, este ponto serve assim de mote ao caso de estudo do capítulo 4. A implementação de um sistema MAS, possibilita o aumento da independência e fiabilidade da troca de informação. O capítulo 6 apresenta a implementação de um sistema que utiliza códigos *International Classification Diseases 9 Revision Clinical Modification (ICD-9-CM)* para codificar resumos de alta, com vista à posterior integração num sistema *Grupo de Diagnósticos Homogêneos (GDH)*, fomentando assim a utilização de códigos na normalização dos processos de alta (interoperabilidade semântica).

Uma aplicação/plataforma que muito tem contribuído para aumento da interoperabilidade no Centro Hospitalar do Porto é a AIDA, conforme abordado no capítulo 2, representa uma forma criativa e inovadora de circundar os problemas provocados pelas "'ilhas de informação."

Só através de sistemas construídos de raiz a pensar na interoperabilidade se pode melhorar significativamente esta.

9.3 questão 3: é viável estruturar um mas de forma a utilizar padrões HL7 possibilitando assim a interoperabilidade sintática e qual é a melhor linguagem para o fazer?

A estrutura de um sistema MAS está intimamente ligada ao meio onde está inserida e qual função a função dos agentes a ser implementados. Como demonstrado no capítulo 4, existe naturalmente um agente responsável pelo envio de mensagens e outro que as recebe e encaminha para seu destino final. Estes dois tipos vão apresentar uma estrutura ligeiramente diferente visto o cliente ter de coletar a mensagem num determinado diretório, fazer um processamento para retirar informações relevantes (ID, tipo de ACK) e no final fazer o envio para o agente de destino. Por seu turno este deve receber a mensagem enviada em segmentos, voltar a colocar em formato original, retirar informações e caso seja necessário elaborar mensagem ACK notificando assim o agente de origem.

Os *socket* implementados desempenham um papel vital na comunicação de determinado agente com os demais. O tópico da comunicação é vital num sistema MAS, devendo os agentes acordar no formato e no tipo de estruturas para tal. No caso de estudo do capítulo 4,

optou-se pelo *JavaScript Object Notation (JSON)* para codificação e listas ordenadas para efeitos de perceção de conteúdo. Mais detalhes sobre este tema em especifico podem ser observados na figura 13.

Os agentes devem também apresentar propriedades como capacidade de recuperar do erro, elevada disponibilidade e ainda registar as transações efetuadas e possíveis erros no desenrolar do seu trabalho. Por último, de referir que implementar agentes sem capacidades de multi-processamento é possível mas resultaria num processamento lento quando um volume de mensagens considerável fosse atingido, utilizou-se assim threading e *multi-processing*.

Em suma, é viável implementar um MAS direcionado à troca de mensagens em formato HL7 v2. Este agentes devem ser vocacionados para ir de encontro às propriedades descritas anteriormente, com especial atenção na comunicação entre agentes e na recuperação após o erro. Como não poderia deixar de ser deve existir um profundo conhecimento das diretrizes HL7 dotando assim os agentes do conhecimento necessário.

Em relação à questão da linguagem, existem algumas passiveis de serem utilizadas neste cenário. Naturalmente Java é um candidato a ter em consideração, utilizando a *framework* Jade. Existem no entanto algumas limitações de implementação que só a simplicidade proporcionada pelo Python consegue superar. A rapidez de desenvolvimento e facilidade de manutenção de um script são sem dúvida duas das principais propriedades da linguagem. Por outro lado, existe a necessidade de que dois agentes comuniquem, daí que o Python *networking* é um dos melhores entre as diversas linguagens identificadas para possível utilização. Outro ponto de relevo, prende-se com a facilidade com que novas expansões ao código fonte podem ser realizadas dotando o código de funcionalidades ímpares. Um exemplo disso mesmo é o *HL7* que permite transformar a mensagem HL7 numa lista, sendo os diversos segmentos acedidos através do seu índice posicional. O *psutil* por outro lado consegue obter valores de performance de um programa em específico através do seu *programm id (PID)*. A principal desvantagem identificada na linguagem Python em relação às demais (Java e C++), é a maior lentidão de processamento, motivado em parte pela não declaração inicial dos tipos de variáveis. Esta desvantagem não se revelou um impedimento ao longo do presente trabalho, sendo superada pelas características acima mencionadas.

Em suma o caso de estudo que responde em maior detalhe situa-se no capítulo 4, principalmente na secção de metodologia onde em detalhe é apresenta a estrutura e função de cada agente.

Ao nível da interoperabilidade, os motores de interface HL7 trabalham com interoperabilidade sintática contribuindo posteriormente para a semântica. Ao nível da comunicação entre agentes, é vital um definição à priori das diversas etapas e qual a ordem pela qual toda a comunicação se deve processar.

9.4 questão 4: qual é o papel de um sistema de monitorização da performance hospitalar?

Antes de elaborar uma resposta precisa e concreta é necessário referir que monitorizar a *performance* neste caso refere-se a averiguar o estado atual de agentes AIDA e HL7 já implementados na unidade de saúde, assim como o sistema MAS desenvolvido no capítulo 4. Ao monitorizar estes agentes, é garantida de forma indireta que a interoperabilidade continua a existir entre SI. Embora na literatura não existam muitos trabalhos sobre este tipo de sistemas de monitorização relacionados com a *performance* de sistemas hospitalares é uma temática que pode ser relacionada com a nova revolução industrial (indústria 4.0). O facto de agentes ou *scripts* capazes de recolher informações e transmitir para unidades centrais conduz à crição de um *Cyber-physical System* onde as informações devem ser apresentadas aos utilizadores na forma de *Computerized Maintenance Monitoring System*. O objetivo é a análise do estado atual dos agentes implementados assim como dos processos de partilha de informação utilizando diretrizes v2. HL7. Um sistema de monitorização deve ser capaz de renovar a informação apresentada ao minuto ou até em espaços temporais menores. Para o profissional identificar erros e corrigir o mais rapidamente possível.

De acordo com a literatura, sistemas semelhantes (Iguana, HealthPoint) e artigos de opinião, algumas das informações que devem estar presentes prendem-se com possíveis erros em agentes ou mensagens; erros nas diretrizes HL7; estatística de mensagens enviadas, recebidas e integradas; diferenciais de tempos entre envio e ACK; receção e integração e ainda tempos de atividade de agentes (datas de inicio e final).

As possibilidades e funções de um sistema desta natureza são inúmeras desde o controlo de agentes à distância, agendamentos e bloco de notas, enviar mails ou notificações, receber avisos e apresentar ao utilizador, apresentar indicadores de *Business Intelligence (BI)* ou até servir de calendário. Para esta diversidade e elasticidade uma propriedade é fundamental, a modularização do sistema, sendo assim possível acrescentar novas funcionalidades ou rever outras sem rever ou alterar todo o código.

Neste tipo de sistemas o utilizador é quem dita o que é ou não relevante. O sistema até pode apresentar informação correta mas se for demasiado lento ou visualmente pouco atrativo leva ao desinteresse e à não utilização. A melhor forma de saber se o sistema satisfaz o utilizador é através de um questionário, onde no futuro alterações podem conduzir a novos módulos ou alterações de funções.

Em suma, a importância deste tipo de sistemas depende em muito do conteúdo e do formato da informação apresentada. Só através de medições à posteriori é possível chegar a conclusões definitivas (análise SWOT e questionário), por outro lado, na literatura existem diversos casos de sucesso nesta área. No presente projeto, o caso de estudo que possibilita uma resposta a esta questão é aquele do capítulo 5.

9.5 Questão 5: Quais são as melhores tecnologias que viabilizam o desenvolvimento de um sistema como aquele abordado na questão 4?

Com a evolução da Internet e dos dispositivos móveis, cada vez mais as aplicações web necessitam de um *design* atrativo, fluído e que acima de tudo não necessitem de muitos recursos por parte do navegador de Internet. Desenvolver estas páginas há muito que deixou de ser apenas trabalhar em *Hyper Text Markup Language (HTML)* passando este a ser integrado com JavaScript ou *Hypertext Preprocessor (PHP)*. Estes apresentam alguma complexidade de perceção de código nomeadamente no acesso às bases de dados. Nos últimos anos surgiram *frameworks* que facilitam em muito o trabalho dos programadores.

No mercado existem *frameworks* como o Angularjs ou o Django que servem de suporte ao desenvolvimento deste tipo de aplicações. O resultados alcançados seriam semelhantes quer utilizando esta nova vaga de *frameworks* ou as linguagens apresentadas anteriormente. O Angularjs como apresenta uma abordagem bastante interessante ao conceito de MVC, com uma maior clareza e divisão de código, modularidade e facilidade de implementação é uma opção bastante viável neste sentido. Outro pronto interessante é a existência de uma enorme base de utilizadores online que permite um suporte elevado. A questão do acesso às base de dados num sistema de monitorização é deveras importante, como descrito no caso de estudo do capítulo 6 e nas tecnologias utilizadas, utilizam-se em JavaScript pedidos *Hypertext Transfer Protocol (HTTP)* utilizando url a um servidor RESTFUL, que podem ser de quatro tipos distintos, *(GET, SET,PUT,DELETE)*.

O caso de estudo que melhor responde a esta questão é o capítulo 6, nomeadamente a metodologia onde é explicado o processo de escolha da *framework*.

9.6 Questão 6: Qual é o melhor tipo de *web service* para ser implementado na unidade de saúde servindo como base para a interoperabilidade entre as bases de dados e a aplicação?

Um Web service é um método de comunicação entre duas aplicações ou dispositivos eletrónicos, através da *World Wide Web (WWW)*. Embora seja um sistema complexo, normalmente, existem dois tipos distintos que devem ser considerados, *Simple Object Access Protocol (SOAP)* e *Representational State Transfer (REST)*. Estas duas abordagens, definem o formato e o protocolo de comunicação entre processos geralmente utilizando pedidos HTTP. Por exemplo, o tipo SOAP exige que Web Service publique a sua descrição em UUID (*Universal Unique Identifier*) que posteriormente responde a um pedido com o WSDL (*Web Service Description Language*) descrevendo assim toda a metodologia e formato da mensagem com vista à realização de pedidos ao provedor. Só neste ponto, é que é necessário o SOAP quem em conjunto com *Extensible Markup Language (XML)* prepara e faz o encapsulamento da mensa-

gem a enviar ao provedor de determinado serviço. Este utiliza XML para responder. O tipo REST não exige este formalismo, descreve um conjunto de princípios arquitetónicos sobre quais os dados podem ser transmitidos por uma interface, tal como HTTP. Não utiliza a linguagem XML, optando por JSON uma estrutura da linguagem JavaScript com grande afinidade por navegadores de Internet. REST não apresenta camada adicional de sintaxe como acontece com no caso do SOAP, focando-se em regras para criar *stateless services*. O cliente neste cenário acede à informação utilizando URI, sendo retornada a representação do serviço. Ao aceder através do protocolo HTTP, o URL identifica o recurso e as operações *GET, SET, PUT, DELETE* representa o tipo de execução a desenvolver. A principal vantagem deste tipo é a facilidade de integração com web sites e ao contrário do XML não consumir tantos recursos computacionais, somando ao facto da transparência do código ser bastante superior àquela apresentada pelo tipo SOAP.

Embora o tipo REST,apresente menos estruturas de segurança do que SOAP, ganha em tempo de processamento e consumo de recursos, algo vital em sistemas distribuídos e implementações móveis ou web, dai ser o melhor tipo neste cenário. A diferença entre dois tipos pode ser vista em maior detalhe no capítulo 3.

9.7 questão 7: o que representa um sistema de gdh para as unidades de saúde?

Em Portugal este tipo de sistemas apresenta funcionalidades diversas, desde o financiamento das instituições de saúde a sistemas para a programação de compras hospitalares. No CHP, a implementação deste sistema prende-se com financiamento estatal como de resto acontece com 50% dos casos no Sistema Nacional de Saúde. Através da implementação de um sistema interativo e devidamente estruturado, é possível monitorizar o volume de trabalho dos codificadores, averiguando a quantidade de resumos de alta codificados num período de tempo. A Medicare, uma seguradora de renome mundial, utiliza um sistema GDH para cálculo de indemnizações para pacientes em internamento em hospitais de agudos.

De acordo com a pesquisa, as principais vantagens da utilização de um GDH numa unidade hospitalar são:

- Maior transparência da gestão financeira e financiamento do sistema hospitalar;
- Permitem às entidades pagadoras um melhor controlo dos gastos com reembolsos;
- Auxiliam as entidades pagadoras a preverem quais os vencimentos financeiros dos hospitais;

Os papéis destes "agrupadores" de diagnósticos são vitais nos dias de hoje para o bom funcionamento de uma instituição de saúde. A resposta a esta pergunta pode ser encontrada

em maior detalhe no caso de estudo do capítulo 7 onde é implementado um sistema de codificação para posterior agrupamento.

9.8 Questão 8: Qual é o papel ICD-9-CM no sistema GDH e quais as tecnologias que permitem ao utilizador fazer a codificação?

As diretrizes do ICD-9-CM são o *core* do funcionamento de um sistema GDH, uma vez que possibilitam um agrupamento final sem erros de interpretação em qualquer parte que utilize os mesmo códigos. Favorecem assim a interoperabilidade semântica entre diversos constituintes do mesmo sistema. Sem a utilização de ICD-9-CM era perfeitamente possível realizar o agrupamento, utilizando um agrupamento manual do resumo de altas. Porém, o consumo de tempo e de recursos assim como a possibilidade de erros aumentava consideravelmente nesse cenário. A importância do ICD-9-CM é evidente, na tentativa de contar a história de todo o internamento utilizando códigos apropriados. A codificação não é de todo um processo simples, os responsáveis por essa tarefa são obrigados a comparecer a ações de formação para estarem habilitados para tal. A maior dificuldade surge na ambiguidade presente nos relatórios diários escritos durante o tempo de internamento. Talvez se logo nessa fase fossem utilizadas estas normalizações, o tempo de processamento nesta fase intermédia fosse muito menor.

Ao nível das tecnologias, existe naturalmente um grande leque de possibilidades. Desde uma aplicação para a instalação no computador (Java ou .NET) até aplicações móveis (IOS, android) passando naturalmente pelas páginas web (Angularjs, PHP + HTML, Django), tudo pode ser implementado, desde que cumpra as diretrizes e estrutura necessária à codificação ICD-9-CM. Uma questão a ter em conta, é que existem relações entre códigos, daí que a tecnologia a aplicar deve ser capaz de responder a essa necessidade com *design* atrativo e fluído. Nesta implementação destinada ao CHP foi pedido uma aplicação web de apenas uma página com interação à base de dados para pesquisas dinâmicas e com capacidade de gravação das codificações efetuadas. Optou-se então pela utilização de Angularjs em conjunto com servidor RESTFUL desenvolvido em PHP. A questão pode ser claramente respondida com leitura do capítulo 6.

9.9 Questão 9: Qual é a melhor forma de implementar aplicações *mhealth* e em que cenários favorecem o Centro Hospitalar do Porto?

Como comprovado por estudos da Organização Mundial de Saúde e demonstrados no capítulo 7, o *mHealth* ocupa atualmente um papel de destaque dentro do eHealth motivado em parte, pelos recentes desenvolvimentos em dispositivos móveis, comunicações através de Internet e ainda a capacidade de armazenamento em *clouds*.

Quando abordada a questão da *mHealth*, a maior parte dos trabalhos foca-se em Telemedicina ou treino médico utilizando aplicações pedagógicas. Existem ainda aplicações vocacionadas ao estimulo através de jogos interativos de pacientes com idade avançada. A necessidade do CHP, prende-se com melhorar a comunicação com os seus utentes e ao mesmo tempo tentar reduzir despesas envolvidas no processo.

Unidades de saúde a nível nacional e internacional perdem por ano milhares de euros em consultas onde o paciente não comparece ou por não se lembrar ou por não desmarcar/remarcar. Assim sendo o CHP procura uma solução para este problema e naturalmente uma aplicação móvel com capacidade de receber notificações consegue resolver este problema de forma eficaz. O caso da aplicação desenvolvida enquadra-se na categoria de *appointment reminders* sendo a 3ª mais trabalhada em toda a Europa (ver figura 21).

Quando se pensa em aplicações móveis dois tipos de SO vêm à cabeça, IOS da Apple e Android da Google. Por ser de distribuição livre e de fácil implementação utilizando linguagem JAVA, optou-se pelo Android, sendo também importante o *Firebase Cloud Messaging* para enviar notificações de forma gratuita.

Esta questão está respondida em detalhe no capítulo 7.

9.10 questão 10: qual a melhor forma de apresentar informações em contexto de aplicações móveis e qual o impacto da interoperabilidade neste cenário?

Aplicações que envolvem apresentação de informação são sempre um ponto delicado, um ponto assente é que não se pode agradar a todos os utilizadores. Uma boa pesquisa e um *design* consistente garantem que esta vai ser aceite por uma grande fatia da comunidade. Aproveitando estudos sobre informação presente em páginas web ficou demonstrado que excesso de informação numa página pode causar desinteresse por parte do utilizador e até fazer com que este abandone a aplicação. Utilizou-se então uma interface simples e intuitiva seguindo a tendência atual das aplicações disponibilizadas pela Google. Existe um menu auto-deslizante que aparece e desaparece por vontade do utilizador, estando contidos os links para as diversas abas da aplicação como histórico de agendamentos no hospital, marcações futuras futuros nessa unidade de saúde e ainda informações relativas à unidade de saúde (contactos e localização).

As notificações são assim a melhor forma de relembrar um paciente que vai ter uma consulta ou um exame, evitando assim a falha. Um aviso Push Notification surge na barra de notificações do telefone, quando é selecionado aparecem informação relativa à data, hora e local de realização da consulta.

A interoperabilidade contribui diretamente para a implementação de uma aplicação móvel qualquer que seja o seu tipo. Quer no pedido de dados por parte da aplicação a

servidores centrais, quer na apresentação quer retorna destes. Sem interoperabilidade do tipo semântica não seria possível apresentar uma interface diferente consulte a consulta tinha acontecido ou ainda esteja para acontecer.

9.11 Questão 11: É viável implementar uma aplicação móvel dedicada a pacientes com vista à disseminação de informação e qual o possível impacto desta?

Os impactos económicos e relativos à imagem da unidade de saúde já foram de certa forma abordados nas questões anteriores. Relativamente à disseminação de informação o impacto é tremendo assistindo-se em parte à saída de informação das fronteiras da unidade de saúde para o utente. A questão da disseminação de informação através de fronteiras de países e instituições há muito têm sido abordada existindo opiniões a favor e outras contra tendo em vista a confidencialidade dos dados. No capítulo 7, é possível observar a quantidade de passos e verificações que foram implementadas de forma a garantir que dados confidenciais não são enviados para destinatários não acreditados. Nos últimos anos a Comissão Europeia apresentou diversas diretivas na fomentação da interoperabilidade entre *Sistemas de Informação Hospitalar (SIH)* de forma criar um PCE europeu. Embora na aplicação desenvolvida não sejam utilizados dados relativos a diagnósticos ou procedimentos, existe sempre a noção de privacidade que deve ser respeitada uma vez que são enviados e recebidos dados relativos à agenda pessoal de cada um.

É viável implementar uma aplicação desta natureza desde que os princípios da confidencialidade de dados e partilha de dados entre sistemas sejam salvaguardados. A utilização de servidores e estruturas que disponibilizam para o exterior dados, geralmente confinados à unidade de saúde carece de especial atenção ao nível de segurança. A interoperabilidade semântica neste ponto assume papel preponderante no envio de dados em formato JSON ou XML para posterior processamento, garantindo assim a perfeita perceção daquilo que está a ser partilhado. Mais que a interoperabilidade semântica, a organizacional deveria ser alcançada por forma a toda a organização estar focada neste novo conceito. Por último, ao nível semântico, a aplicação deve reconhecer determinados códigos para posterior apresentação de diferentes *layouts* aos utilizadores. Ao longo do capítulo 7 esta pergunta vai sendo respondida com maior detalhe.

9.12 Questão 12: Qual é o futuro dos SI e o da interoperabilidade?

Os sistemas de informação desempenham um papel preponderante nas unidades hospitalares em todo o mundo. O futuro destes sistemas passa pelo melhoramento da partilha e cruzamento de informações, que até à data, ainda apresentam aspetos negativos em muitas

das unidades de saúde em Portugal e não só. Essas novas características proporcionam bases para a disseminação de informação para além das fronteiras da unidade de saúde ou até do próprio país. Caso disso são as tentativas da União Europeia para a criação do processo clínico eletrónico comum a todos os países membros. No documento que serve de *guideline* à implementação, autores referem os diversos entraves de cariz legal relacionados com a confidencialidade e privacidade dos registos.

Numa abordagem mais técnica, existem atualmente novas formas de armazenar dados nomeadamente a utilização de *clouds*, eliminando em parte problemas de falta de capacidade computacional ou espaço físico.

O futuro dos sistemas de informação passa então pela solidificação das boas práticas já implementadas, nomeadamente a interoperabilidade e pela extração da informação para fora das fronteiras das unidades de saúde.

Ao nível da interoperabilidade, a consolidação das boas práticas já implementadas em muitos locais, assim como o fomentar da interoperabilidade organizacional são sem duvida as principais tarefas a desenvolver. Neste último caso, toda a instituição deve estar envolvida, sendo os projetos pensados desde início com a partilha de informação em mente e não apenas abordar esta questão à *posteriori*.

9.13 questão 13: de que forma é que os casos de estudo apresentados contribuem para o melhoramento da interoperabilidade a nível hospitalar?

Esta questão de certa forma foi sendo respondida ao longo das anteriores, sem dúvida que o caso de estudo 1 (capítulo 5) apresenta uma nova metodologia para a implementação de *standards* de comunicação utilizando um MAS. Estes, funcionam sem ajuda externa, constituindo um impacto considerável no desempenho dos HIS e na otimização dos processos informativos. Em suma, este primeiro caso, mais que os meios para comunicar (*Internet sockets*) implementa uma via para o aumento da interoperabilidade sintática.

Nem todos os sistemas desenvolvidos têm como destinatário final o paciente ou os cuidados deste, neste sentido, desenvolveu-se um talhado para os funcionários e responsáveis dos serviços de informação hospitalar. Denominado de sistema de monitorização, tem como objetivos monitorizar agentes AIDA e HL7 já implementados na unidade de saúde e ainda controlar o MAS desenvolvido anteriormente. Embora não relacionado diretamente com interoperabilidade, ajuda na deteção e rápida resolução de erros e ainda na ajuda à toma de decisão de forma o mais informada possível. Constituindo assim uma ferramenta útil à sustentação da partilha de informação e cooperação entre sistemas.

O impacto do GDH é imenso dentro de uma unidade de saúde sendo o sistema de informação nele baseado utilizado em situações diversas como financiamento, método de

pagamento de indemnizações e até gestão de compras hospitalares. Ao agrupar em 25 códigos possíveis os relatórios de alta, permite criar barómetros essenciais na medição da produção clínica contribuindo em parte para interoperabilidade organizacional. O sistema em si, compreende a utilização dos códigos de representação clínica ICD-9-CM. Estes fomentam a interoperabilidade semântica contribuindo para que os dois sistemas se percebam mutuamente e ainda possibilitando que um relatório de alta seja percetível em qualquer parte de globo.

Patient Relationship Manager e aplicações *appointment reminder* são uma nova tendência na área do *mHealth* que devem ser exploradas. Com a construção do caso de estudo do capítulo 7 é assim possível ao Centro Hospitalar do Porto fazer chegar informação de qualidade aos seus utentes em qualquer ponto do globo. Neste cenário, a interoperabilidade é posta à prova, ao nível sintático (envio dados formato JSON), ao nível físico (pedido HTTP) e semântica (alteração de *layout* consoante conteúdo).

9.14 TRABALHO FUTURO

Como observado pelas respostas às perguntas de investigação, os objetivos gerais proposto para o projeto foram alcançados. Olhando para o capítulo 25, referente às diversas *Proof-of-Concept (POC)* dos diferentes casos de estudo é possível perceber que todas as soluções apresentam bastantes aspetos positivos. Naturalmente numa análise SWOT, surgem aspetos menos bons que devem ser melhorados e ameaças que devem ser ultrapassadas. Na conclusão dos respetivos casos de estudo, já foram de certa forma apresentados os próximos passos em cada cenário. A seguinte lista pode ser elabora como forma de apresentar o trabalho futuro a desenvolver em cada cenário:

SISTEMA DE ENVIO E RECEÇÃO DE MENSAGENS EM HL7: Aperfeiçoar as acções a desenvolver após o erro e ainda a comunicação entre agentes. Melhorar a metodologia da instalação dos agentes em máquinas novas;

SISTEMA DE MONITORIZAÇÃO AIDA E HL7: implementação de novo web service para acesso ao conteúdo de mensagens HL7. Acesso à rede externa, com vista à conversão da plataforma em aplicação móvel. Melhorar o processo de *Extract, Load, Transform (ETL)* para permitir um BI mais eficaz;

PLATAFORMA DE CODIFICAÇÃO ICD-9-CM: inserção de método que permita gravar *templates* de codificações. Criação de restrições à conjugação de determinados códigos e desenvolvimento de uma aplicação de BI sobre *performance* das codificações e codificadores;

APLICAÇÃO MÓVEL AIDA_PACIENTES: desenvolvimento de web service novo para fornecer informações sobre pacientes assim como novo método de ETL para carregamento de tabelas. Implementação de novas abas na aplicação móvel.

Paralelo a qualquer dissertação está a investigação cientifica, neste sentido ao longo do presente projeto, desenvolveram-se duas publicações diretamente relacionadas com aspetos abordados. Interoperabilidade semântica, numa aplicação sobre SNOMED ct (ver refa) e outra sobre interoperabilidade sintática, abordando conceitos relativos ao HL7(A.1.2).

BIBLIOGRAFIA

[1] M. N. Almunawar and M. Anshari, "Health Information Systems (HIS): Concept and Technology," 2006.

[2] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, vol. 2, p. 3, 2014.

[3] D. E. Detmer, "Information technology for quality health care: a summary of United Kingdom and United States experiences," *Qual. Health Care*, vol. 9, no. 3, pp. 181–189, 2000.

[4] S. V. B. Jardim, "The Electronic Health Record and its Contribution to Healthcare Information Systems Interoperability," *Procedia Technology*, vol. 9, pp. 940–948, 2013.

[5] W. G. of April, "Recommendations for implementing the strategic," Tech. Rep. April, 2013.

[6] F. de Medicina da Universidade do Porto, "Relatório sobre sistemas de informação em saúde." http://im.med.up.pt/si{%}7B{_}{%}7Dsaude/si{%}7B{_}{%}7Dsaude.html{%}7B{#}{%}7Dgen2, 2005, visited 2016-07-18, 2005.

[7] J. Duarte, C. F. Portela, A. Abelha, J. Machado, and M. F. Santos, "Electronic health record in dermatology service," *Communications in Computer and Information Science*, vol. 221 CCIS, no. PART 3, pp. 156–164, 2011.

[8] J. Valacich, C. Schneider, and L. Jessup, "Managing in the digital world," in *Information systems today*, ch. 1, pp. 2–37, Prentice Hall, 2010.

[9] L. S. Gonçalves, "Sistema de informação," 2016.

[10] R. Satair and G. Reynolds, *An Introduction to Information Systems*. Course Technology, 10 ed., 2012.

[11] D. Q. Chen, D. Preston, M. Mocker, and A. Teubner, "INFORMATION S YSTEMS S TRATEGY : RECONCEPTUALIZATION , MEASUREMENT , AND IMPLICATIONS," *MIS Quarterly*, vol. 34, no. 2, pp. 1–8, 2010.

[12] V. Arvidsson, J. Holmstr??m, and K. Lyytinen, "Information systems use as strategy practice: A multi-dimensional view of strategic information system implementation and use," *Journal of Strategic Information Systems*, vol. 23, no. 1, pp. 45–61, 2014.

[13] Y. K. Dwivedi, D. Wastell, S. Laumer, H. Z. Henriksen, M. D. Myers, D. Bunker, A. Elbanna, M. N. Ravishankar, and S. C. Srivastava, "Research on information systems failures and successes: Status update and future directions," *Information Systems Frontiers*, vol. 17, no. 1, pp. 143–157, 2014.

[14] J. S. Valacich and C. Schneider, "Valuing Information Systems Investments," in *Information systems today*, ch. 3, pp. 82–129, Prentice Hall, 2010.

[15] A. Hasman, R. Haux, and A. Albert, "A systematic view on medical informatics," *Computer Methods and Programs in Biomedicine*, vol. 51, no. 3, pp. 131–139, 1996.

[16] R. Haux, "Health information systems - Past, present, future," *International Journal of Medical Informatics*, vol. 75, no. 3-4 SPEC. ISS., pp. 268–281, 2006.

[17] Alto Comissariado para a Saúde, "Operacionalização do Plano Nacional de Saúde - I Políticas Transversais: 4) Tecnologias de Informação e Comunicação," *Plano Nacional de Saúde 2011-2016*, pp. 1–37, 2011.

[18] R. Espanha, "Sistemas de Informação em Saúde e Saúde Online," *Plano Nacional de Saúde - Alto Comissariado da Saude*, pp. 1–12, 2010.

[19] H. K. Andreassen, M. M. Bujnowska-Fedak, C. E. Chronaki, R. C. Dumitru, I. Pudule, S. Santana, H. Voss, and R. Wynn, "European citizens' use of E-health services: a study of seven countries," *BMC Public Health*, vol. 7, p. 53, 2007.

[20] A. A. A. Sandi, "A importância dos Sistemas de Informação em Saúde – Estudo de caso na USF CelaSaúde," pp. 1–84, 2015.

[21] K. Bernstein, M. Bruun-rasmussen, S. Vingtoft, S. Kjær, and C. Nøhr, "Modelling and implementing electronic health records in Denmark," 2005.

[22] T. D. Gunter and N. P. Terry, "The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions," *Journal of Medical Internet Research*, 2005.

[23] R. Marinho, J. Machado, and A. Abelha, "Processo Clínico Electrónico Visual," in *INForum 2010 - II Simpósio de Informática* (L. S. Barbosa and M. P. Correia, eds.), pp. 767–778, 2010.

[24] R. J. C. Correia, "Registos Clínicos Electrónicos," tech. rep., Universidade de Medicina da Universidade do Porto, Porto, 2015.

[25] M. Berg, "Implementing information systems in health care organizations: myths and challenges," *International Journal of Medical Informatics*, vol. 64, pp. 143–156, 2015.

[26] N. Ide and J. Pustejovsky, "What Does Interoperability Mean , Anyway ? Toward an Operational Definition of Interoperability for Language Technology," *Proceedings of the Second International Conference on Global Interoperability for Language Resources.*, 2010.

[27] M. E. Mansoor and R. Majeed, *Achieving Interoperability among Healthcare Organizations*. PhD thesis, Blekinge Institute of Technology, 2010.

[28] G. Vetere and M. Lenzerini, "Models for semantic interoperability in service-oriented architectures," *IBM Systems Journal*, vol. 44, no. 4, pp. 887–903, 2005.

[29] T. Benson, "Principles of Health Interoperability HL7 and SNOMED," in *Healht Informatic Series*, vol. 6 of *Health Informatics*, ch. 2, p. 287, Springer London, 2010.

[30] A. Jin and L. Ahlfors, *Interoperability in Electronic Medical Records*. PhD thesis, Lund University, School of Economics and Management, 2012.

[31] A. Tolk, S. Y. Diallo, and C. D. Turnitsa, "Applying the Levels of Conceptual Interoperability Model in Support of Integratability , Interoperability , and Composability for System-of-Systems Engineering," *Journal of Systemics, Cybernetics and Informatics*, vol. 5, no. 5, pp. 65–74, 2007.

[32] H. Peixoto, M. Santos, A. Abelha, and J. Machado, *Intelligence in Interoperability with AIDA*, pp. 264–273. Springer Berlin Heidelberg, 2012.

[33] L. Cardoso, F. Marins, F. Portela, A. Abelha, and J. Machado, "Healthcare Interoperability through Intelligent Agent Technology," *Procedia Technology*, vol. 16, pp. 1334–1341, 2014.

[34] M. Wooldridge, "Introduction," in *An Introduction to MultiAgent Systems*, ch. 1, pp. 5–10, Wiley, 2nd ed., 2009.

[35] L. Cardoso, F. Marins, F. Portela, M. Santos, A. Abelha, and J. Machado, "The next generation of interoperability agents in healthcare," *International Journal of Environmental Research and Public Health*, vol. 11, no. 5, pp. 5349–5371, 2014.

[36] Health Information and Quality Authority, "Overview of Healthcare Interoperability Standards," tech. rep., Health Information and Quality Authority, 2013.

[37] J. Pathak, K. R. Bailey, C. E. Beebe, S. Bethard, D. S. Carrell, P. J. Chen, D. Dligach, C. M. Endle, L. A. Hart, P. J. Haug, S. M. Huff, V. C. Kaggal, D. Li, H. Liu, K. Marchant, J. Masanz, T. Miller, T. A. Oniki, M. Palmer, K. J. Peterson, S. Rea, G. K. Savova, C. R. Stancl, S. Sohn, H. R. Solbrig, D. B. Suesse, C. Tao, D. P. Taylor, L. Westberg, S. Wu, N. Zhuo, and C. G. Chute, "Normalization and standardization of electronic health

records for high-throughput phenotyping: the SHARPn consortium," *Journal of the American Medical Informatics Association : JAMIA*, vol. 20, no. e2, pp. e341—-e348, 2013.

[38] M. S. Unluturk, "Using HL7 in hospital staff assignments," *Computers in Biology and Medicine*, vol. 45, no. 1, pp. 67–71, 2014.

[39] L. Ahlfors, A. Olerup, and M. Lahtinen, "Interoperability in Electronic Medical Records," no. June, 2012.

[40] D. Bender, "HL7 FHIR : An Agile and RESTful Approach to Healthcare Information Exchange," pp. 1–6, 2013.

[41] D. M. López and B. Blobel, "Architectural approaches for HL7-based health information systems implementation," *Methods of Information in Medicine*, vol. 49, no. 2, pp. 196–204, 2010.

[42] E. W. Huang, S. H. Hsiao, and D. M. Liou, "Design and implementation of a web-based HL7 message generation and validation system," *International Journal of Medical Informatics*, vol. 70, no. 1, pp. 49–58, 2003.

[43] H. L. S. International, "About HL7." http://www.hl7.org/about/index.cfm?ref=nav, 2016, visited 2016-07-30, 2016.

[44] H. L. S. International, "HL7 Mission." http://www.hl7.org/about/index.cfm?ref=nav, 2016, visited 2016-07-30, 2016.

[45] M. Corporation, "The OSI Model's Seven Layers Defined and Functions Explained." https://support.microsoft.com/en-us/kb/103884, 2014, visited 2016-07-31, 2014.

[46] T. Viangteeravat, M. N. Anyanwu, V. R. Nagisetty, E. Kuscu, and M. E. Sakauye, "Clinical data integration of distributed data sources using Health Level Seven (HL7) v3-RIM mapping," *Journal of Clinical Bioinformatics*, vol. 1, no. 1, p. 32, 2011.

[47] D. Shaver, "HL7 101: A Beginner's Guide," 2007.

[48] C. Health, "Comparing HL7 Version 2 to Version 3 Including a History of Version 2," tech. rep., 2008.

[49] M. Corporation, "HL7 Message Structure." https://msdn.microsoft.com/en-us/library/ee409289.aspx, 2013, visited 2016-08-07, 2013.

[50] D. Rajeev, C. J. Staes, R. S. Evans, S. Mottice, R. Rolfs, M. H. Samore, J. Whitney, R. Kurzban, and S. M. Huff, "Development of an electronic public health case report using HL7 v2.5 to meet public health needs," *Journal of the American Medical Informatics Association*, vol. 17, no. 1, pp. 34–41, 2010.

[51] H. L. S. International, "MSH - message header segment." https://www.hl7.org/documentcenter/public{%}7B{_}{%}7Dtemp{%}7B{_}{%}7D5631C143-1C23-BA17-0CEEDD0B42A99B22/wg/conf/HL7MSH.html, 2016, visited 2016-08-29, 2016.

[52] N. C. f. H. Statistics, "Classification of Diseases, Functioning, and Disability," 2015.

[53] J. DiSantostefano, "Official ICD-9-CM Guidelines," *The Journal for Nurse Practitioners*, vol. 1, no. 1, pp. 36–37, 2005.

[54] A. C. do Sistema de Saúde, "Codificação Clínica." http://www.acss.min-saude.pt/DepartamentoseUnidades, 2010, visited 2016-08-01, 2010.

[55] R. H. White, M. Garcia, B. Sadeghi, D. J. Tancredi, P. Zrelak, J. Cuny, P. Sama, H. Gammon, S. Schmaltz, and P. S. Romano, "Evaluation of the predictive value of ICD-9-CM coded administrative data for venous thromboembolism in the United States," *Thrombosis Research*, vol. 126, no. 1, pp. 61–67, 2010.

[56] M. J. A. Abrantes, C. Costa, F. Moura, and B. Soediono, "Avaliação do Desempenho dos Hospitais Públicos (Internamento) em Portugal Continental (2013)," *Revista Portuguesa de Sa{ú}de P{ú}blica*, vol. 4, no. 1, p. 105, 2015.

[57] R. S. Stern and A. M. Epstein, "Institutional Responses to Prospective Payment Based on Diagnosis-Related Groups," *The New England journal of medicine*, vol. 312, no. 10, pp. 621–627, 1985.

[58] C. M. Borges, "Grupos de Diagnósticos Homogéneos (GDH)." http://portalcodgdh.min-saude.pt/index.php/Grupos%7B_%7Dde%7B_%7DDiagn%7B%25%7DC3%7B%25%7DB3sticos%7B_%7DHomog%7B%25%7DC3%7B%25%7DA9neos%7B_%7D(GDH), 2004, visited 2016-08-01, 2011.

[59] R. F. Averill, N. Goldfield, J. S. Hughes, J. Bonazelli, E. C. McCullough, B. a. Steinbeck, R. Mullin, A. M. Tang, J. Muldoon, L. Turner, and J. Gay, "All Patient Refined Diagnosis Related Groups (APR-DRGs): Methodology Overview," *3M Health Information Systems*, pp. 1–81, 2003.

[60] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of Things," *International Journal of Communication Systems*, vol. 25, pp. 1101–1102, 2012.

[61] D. Wee, R. Kelly, J. Cattel, and M. Breunig, "Industry 4.0," tech. rep., Mckinsey Digital, 2015.

[62] M. Hosseini, "What will the future look like under Industry 4 . 0 and digital transformation in the healthcare space ?," 2015.

[63] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[64] J. Bloem, "The Fourth Industrial Revolution Things to Tighten the Link Between it and ot," *VINT research report*, pp. 1–39, 2014.

[65] J. Jimenez, "Three ways the Fourth Industrial Revolution can transform healthcare." https://www.weforum.org/agenda/2016/01/three-ways-the-fourth-industrial-revolution-could-transform-healthcare/, 2016, visited 2016-07-30, 2016.

[66] S. Heng, "Talking point eHealth : Industry 4 . 0 can serve as the model for digital healthcare," *Deutsche Bank Research*, pp. 1–3, 2015.

[67] S. Bernasconi, "The role of medical technology in the 4th Industrial Revolution." http://medtechviews.eu/article/role-medical-technology-4th-industrial-revolutionl, 2016, visited 2016-08-03, 2016.

[68] WHO, "mHealth: New horizons for health through mobile technologies: 2nd Global Survey on eHealth," vol. 3, no. June, 2011.

[69] D. D. Luxton, R. a. McCann, N. E. Bush, M. C. Mishkind, and G. M. Reger, "mHealth for mental health: Integrating smartphone technology in behavioral healthcare.," *Professional Psychology: Research and Practice*, vol. 42, no. 6, pp. 505–512, 2011.

[70] IBM, "Welcome to the Era of Cognitive Health." http://www.ibm.com/watson/health/, 2016, visited 2016-09-17, 2016.

[71] C. S. Europe, "Medical Technology," no. September, p. 2015, 2014.

[72] E. Bartocci, S. Gao, and S. A. Smolka, "Medical Cyber-Physical Systems (Track Introduction)," pp. 353–355, 2014.

[73] E. Bartocci, F. Corradini, M. R. Di Berardini, E. Entcheva, S. A. Smolka, and R. Grosu, "Modeling and simulation of cardiac tissue using hybrid I/O automata," *Theoretical Computer Science*, vol. 410, no. 33-34, pp. 3149–3165, 2009.

[74] E. Bartocci, F. Corradini, E. Entcheva, R. Grosu, and S. a. Smolka, "CellExcite: an efficient simulation environment for excitable cells.," *BMC bioinformatics*, vol. 9 Suppl 2, p. S3, 2008.

[75] W. K. Edwards, "Discovery systems in ubiquitous computing," *IEEE Pervasive Computing*, vol. 5, no. 2, pp. 70–77, 2006.

[76] K. Lyytinen and Y. Yoo, "Issues and Challenges in Ubiquitous Computing," *Communications of the ACM*, vol. 45, no. 12, p. 34, 2002.

[77] M. Weiser, "The computer for the 21st century," *Scientific American (International Edition)*, vol. 265, no. 3, pp. 66–75, 1991.

[78] I. Brown and A. a. Adams, "The ethical challenges of ubiquitous healthcare," *International Review of Information Ethics*, vol. 8, pp. 53–60, 2007.

[79] P. C. Hii and W. Y. Chung, "A comprehensive ubiquitous healthcare solution on an Android??? mobile device," *Sensors*, vol. 11, no. 7, pp. 6799–6815, 2011.

[80] C. Tang and S. Carpendale, "Healthcare Quality and Information Flow during Shift Change," *Quality In Health Care*, 2006.

[81] K. P. Fishkin, S. Consolvo, J. Rode, B. Ross, I. Smith, and K. Souter, "Ubiquitous Computing Support for Skills Assessment in Medical School," *International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, vol. 2004, 2004.

[82] A. I. Wasserman and Fosser, "Software Engineering Issues for Mobile Application Development," *Acm*, pp. 1–4, 2010.

[83] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[84] D. Mourtzis, M. Doukas, F. Psarommatis, C. Giannoulis, and G. Michalos, "A web-based platform for mass customisation and personalisation," *CIRP Journal of Manufacturing Science and Technology*, vol. 7, no. 2, pp. 112–128, 2014.

[85] M. N. K. Boulos, S. Wheeler, C. Tavares, and R. Jones, "How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX.," *Biomedical engineering online*, vol. 10, p. 24, 2011.

[86] A. Kailas, C. C. Chong, and F. Watanabe, "From mobile phones to personal wellness dashboards," *IEEE Pulse*, vol. 1, no. 1, pp. 57–63, 2010.

[87] A. K. Yetisen, J. L. Martinez-Hurtado, F. da Cruz Vasconcellos, M. C. E. Simsekler, M. S. Akram, and C. R. Lowe, "The regulation of mobile medical applications.," *Lab on a chip*, vol. 14, no. 5, pp. 833–840, 2014.

[88] C.-L. Yau and W.-Y. Chung, "IEEE 802.15.4 Wireless Mobile Application for Healthcare System," *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pp. 1433–1438, 2007.

[89] C.-J. Chiu, Y.-H. Hu, D.-C. Lin, F.-Y. Chang, C.-S. Chang, and C.-F. Lai, "The attitudes, impact, and learning needs of older adults using apps on touchscreen mobile devices: Results from a pilot study," *Computers in Human Behavior*, vol. 63, pp. 189–197, 2016.

[90] C. Bereiter, "Design research for sustained innovation," *Cognitive studies, bulletin of the Japanese cognitive* {...}, vol. 9, no. 3, pp. 321–327, 2002.

[91] P. Lee, "Design Research: What is it and Why do it?." http://reboot.org/2012/02/19/design-research-what-is-it-and-why-do-it/, 2012, visited 2016-05-26, 2012.

[92] C. C. f. N. M. T. Learning, "About Design Research at CCNMTL." http://ccnmtl.columbia.edu/dr/about/, 2016, visited 2016-07-27.

[93] V. Vaishnavi and B. Kuechler, "Design Science Research in Information Systems," in *12th Interantional Conference on Design Science Research in Information Systems and Technology*, 2015.

[94] G. V. Rossum, "The Python Language Reference Manual," 2012.

[95] D. Kuhlman, "A Python Book: Beginning Python, Advanced Python, and Python Exercises," pp. 1–227, 2009.

[96] M. Lutz, *Python Pocket Reference: Python in Your Pocket*. 2014.

[97] J. Melorose, R. Perroy, and S. Careas, *Python Cookbook*, vol. 1. 2015.

[98] J. Goerzen, E. Board, S. Anglin, D. Appleman, E. Buckingham, G. Cornell, T. Davis, J. Franklin, J. Gilmore, C. Mills, S. Rycroft, D. Shakeshaft, J. Sumser, K. Watterson, G. Wray, and J. Zukowski, "Foundations of Python Network Programming," *Network*.

[99] J. E. Ingvaldsen, z. ??zg??bek, and J. A. Gulla, *Context-aware user-driven news recommendation*, vol. 1542. 2015.

[100] R. Greenwald, R. Stackowiak, and J. Stern, *Oracle Essentials - Oracle Database 11g*. 2008.

[101] A. Beaulieu, *Learning SQL*. Sebastopol: O'Really Media nc., 2009.

[102] P. Lane, "Data Warehousing Guide 12c Release 1 (12.1)," Tech. Rep. E41670-08, 2013.

[103] S. Feierstein and B. Pribyl, *Oracle PL/SQL Programming*. O'Really Media nc., 5 ed., 2009.

[104] R. Ramakrishnan and J. Gehrke, "Database Management System," pp. 677–679, 2000.

[105] S. D. Oracle, "What is SQL Developer." http://www.oracle.com/technetwork/developer-tools/sql-developer/what-is-sqldev-093866.html, 2014, visited 2016-07-23, 2014.

[106] O. Coporation, "Using Python With Oracle Database." http://www.oracle.com/technetwork/articles/dsl/python-091105.html, 2016, visited 2016-07-23, 2016.

[107] Oracle Corporation, "What is MySql." http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html, 2016, visited 2016-07-23, 2016.

[108] S. Suehring, *MySQL™ Bible*. 202.

[109] E. Newcomer, *Understanding Web Services*. independen ed., 2013.

[110] J. Roy and A. Ramanujan, "Understanding Web Services Policy," *IT professional*, vol. 3, no. 6, pp. 1–28, 2001.

[111] D. Booth, H. Haas, F. MacCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web Services Architecture." http://www.w3.org/TR/ws-arch/{%}7B{#}{%}7Dintroduction, 2004, visited 2016-07-23, 2004.

[112] A. Arsanjani, "Service-oriented modeling and architecture," *IBM developer works*, no. January, pp. 1–15, 2004.

[113] R. Perrey and M. Lycett, "Service-oriented architecture," *2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings.*, pp. 116–119, 2003.

[114] C. Pautasso, "RESTful Web Services: Principles, Patterns, Emerging Technologies," in *Web Services Foundations*, vol. 9781461475, pp. 31–51, Springer New York, 2014.

[115] A. Rodriguez, "Restful web services: The basics," *Online article in IBM DeveloperWorks Technical Library*, no. November, pp. 1–11, 2008.

[116] O. Affliliates, "What are RESTFUL Web Services?." http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html, 2013, visited 2016-07-23, 2013.

[117] Mike Rozlog, "REST e SOAP: Usar um dos dois ou ambos?." https://www.infoq.com/br/articles/rest-soap-when-to-use-each, 2013, visited 2016-07-24, 2013.

[118] F. Belqasmi, J. Singh, S. Bani, and R. H. Glitho, "SOAP vs REST - Multimedia Conferencing Applications," 2012.

[119] R. Lerdorf, K. Tatroe, and P. MacIntyre, *Programming PHP*, vol. 37. 2006.

[120] Sakshay, "10 Advantages of Choosing PHP Web Development," 2013.

[121] Phppot, "PHP RESTful Web Service." http://phppot.com/php/php-restful-web-service/, 2016, visited 2016-07-24, 2016.

[122] J. Knupp, "What is a Web Framework?." https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/, 2013, visited 2016-07-24, 2013.

[123] R. Branas, *AngularJS Essentials*. 2014.

[124] A. Prikaznov, "MVC and MVVM Design Patterns With Angular JS." https://dzone.com/articles/angularjs-tutorial-lesson-1, 2016, visited 2016-07-24, 2016.

[125] S. Seshadri and B. Green, *AngularJS Up {&} Running*. O'Really Media nc., 2014.

[126] G. Company, "Data Binding." https://docs.angularjs.org/guide/databinding, 2016, visited 2016-08-24, 2016.

[127] D. Flanagan, *JavaScript: The Definitive Guide*. No. January 1997, 2011.

[128] R. Rogers, J. Lombardo, Z. Mednieks, and G. Blake Meike, *Android Application Development: Programming with the Google SDK*. 2009.

[129] S. Microsystems, "Design Goals of JAVA." http://www.oracle.com/technetwork/java/intro-141325.html, 1997, visited 2016-07-25, 1997.

[130] C. Health, "Why Do I Need an Interface Engine ?," tech. rep., 2016.

[131] O. Health, "What is a HL7 interface engine?." http://www.hl7.com/interface-engine.html, 2016, visited 2016-02-26, 2016.

[132] C. H. Technologies, "2012 HL7 Interface Technology Survey Results," no. August, 2012.

[133] C. Health, "What is Your Healthcare Interface Method ?," 2009.

[134] M. C. Silveira and E. Arte, "Sistemas Distribuí dos Sistemas Distribuí dos Sistemas Distribuídos," pp. 1–18, 2001.

[135] S. T. Corporation, "HL7 BRIDGE CONFIGURATION Scientific Technologies Corporation," tech. rep., 2007.

[136] R. Spronk and R. Spronk, "Transport Specification : MLLP , Release 1 Edit or Minimal Lower Layer Protocol (MLLP)," pp. 1–4, 2003.

[137] M. Oliver, "Part1: Introduction and Fundamental Protocols." http://www.itprc.com/tcpipfaq/faq-1.htm{%}7B{#}{%}7Dwhat-tcpip, 1999, visited 2016-08-11, 1999.

[138] T. Socolofsky and C. Kale, "A TCP/IP Tutorial," tech. rep., 1991.

[139] K. S. Um, S. K. Yun, H. Cho, and K. K. Il, "Development of an HL7 interface engine, based on tree structure and streaming algorithm, for large-size messages which include image data," *Computer Methods and Programs in Biomedicine*, vol. 80, no. 2, pp. 126–140, 2005.

[140] J. Machado, M. Miranda, A. Abelha, J. Neves, and J. Neves, *Modeling Medical Ethics through Intelligent Agents*, pp. 112–122. Springer Berlin Heidelberg, 2009.

[141] J. Luís, C. De Moraes, W. Lopes De Souza, L. Ferreira Pires, L. T. Cavalini, . Antônio, and F. Do Prado, "UMA ARQUITETURA PARA TROCA DE MENSAGENS NO CUIDADO DE SA{Ú}DE PERVASIVO BASEADA NO USO DE AGENTES INTELIGENTES An Architecture for Message Exchange in Pervasive Healthcare based on the use of Intelligent Agents," vol. 2, no. 4, pp. 145–156, 2013.

[142] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi, "Multi-agent oriented programming with JaCaMo," *Science of Computer Programming*, vol. 78, no. 6, pp. 747–761, 2013.

[143] W. van der Hoek and M. Wooldridge, "Chapter 24 Multi-Agent Systems," *Foundations of Artificial Intelligence*, vol. 3, no. 07, pp. 887–928, 2008.

[144] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[145] J. Paulett, "Python-hl7 - Easy HL7 v2.x Parsing." https://python-hl7.readthedocs.io/en/latest/, 2011, visited 2016-08-09, 2011.

[146] G. Rodola, "Summary." https://github.com/giampaolo/psutil, 2009, visited 2016-08-09, 2009.

[147] P. S. Foundation, "18.1. socket — Low-level networking interface." https://docs.python.org/3.4/library/socket.html, 2016, visited 2016-08-09, 2016.

[148] ECMA-404, "ECMA-404 The JSON Data Interchange Standard." http://www.json.org/, 2016, visited 2016-08-13.

[149] F. p. C. e. Tecnologia, "Significado de Monitorizar," 2016.

[150] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, "Damage Identification and Health Monitoring os Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review," *Los Alamos National Laboratory*, 1996.

[151] S. A. Raymond, G. E. Gordon, and D. B. Singer, "Health Monitoring System." https://www.google.com/patents/US6282441, 1995, visited 2016-07-24, 1995.

[152] H. Guo, G. Xiao, N. Mrad, and J. Yao, "Fiber optic sensors for structural health monitoring of air platforms," *Sensors*, vol. 11, no. 4, pp. 3687–3705, 2011.

[153] E. N. Inc., "Real-Time HL7 Analytics," tech. rep., ExtraHop Networks Inc, Seatle, 2014.

[154] M. Stockemer, "Monitoring and Alerting for HL7 Interfaces," 2007.

[155] Eggebraaten, "A health-care data model based on the HL7 Reference Information Model," vol. 46, no. 1, pp. 5–18, 2007.

[156] R. Kimball and M. Ross, *The Data Warehouse Toolkit*, vol. 53. Indianapolis: John Wiley and Sons, 3 ed., 2013.

[157] A. Albrecht and F. Naumann, "Managing etl processes," *New York*, no. May, pp. 1–7, 2008.

[158] A. Simitsis and P. Vassiliadis, "A Methodology for the Conceptual Modeling of ETL Processes," *CAiSE workshops.*, pp. 305–316, 2003.

[159] M. Ross, "Kimball Lifecycle in a Nutshell." http://www.kimballgroup.com/2009/08/design-tip-115-kimball-lifecycle-in-a-nutshell/, 2009, visited 2016-08-18, 2009.

[160] A. Leff and J. T. Rayfield, "Web-application development using the Model/View/-Controller design pattern," *Proceedings - 5th IEEE International Enterprise Distributed Object Computing Conference*, vol. 2001-Janua, no. January, pp. 118–127, 2001.

[161] D. Schwabe, L. Esmeraldo, G. Rossi, and F. Lyardet, "Engineering Web applications for reuse," *IEEE Multimedia*, vol. 8, no. 1, pp. 20–31, 2001.

[162] M. Takada, "Single Page Apps in Depth," 2012.

[163] A. Neyem, M. J. Carrillo, C. Jerez, G. Valenzuela, J. I. Benedetto, and J. S. Rojas-riethmuller, "Improving Healthcare Team Collaboration in Hospital Transfers through Cloud-based Mobile Systems," vol. 2016, 2016.

[164] L. E. Kazis, "The Veterans SF-36 Health Status Questionnaire : Development and Application in the Veterans Health Administration," *Monitor*, vol. 5, no. 1, 2000.

[165] J. Likness, "10 Reasons Web Developers Should Learn Angularjs." http://www.wintellect.com/devcenter/jlikness/10-reasons-web-developers-should-learn-angularjs, 2014, visited 2016-08-17, 2014.

[166] C. Thomsen and T. B. Pedersen, "pygrametl : A Powerful Programming Framework for Extract – Transform – Load Programmers," 2009.

[167] C. Costa, S. Lopes, and R. Santana, "Diagnosis Related Groups e Disease Staging: importância para a administração hospitalar," *Revista Portuguesa de Saúde Pública*, vol. 7, pp. 7–28, 2008.

[168] A. C. do Sistema de Saúde, "Grupos de Diagnósticos Homogéneos." http://www.acss.min-saude.pt, 2016, visited 2016-08-20, 2016.

[169] R. Busse, A. Geissler, A. Aaviksoo, F. Cots, U. Hakkinen, C. Kobel, C. Mateus, Z. Or, J. O'Reilly, L. Serden, A. Street, S. S. Tan, and W. Quentin, *Diagnosis related groups in Europe: moving towards transparency, efficiency, and quality in hospitals?*, vol. 346. 2013.

[170] C. Borges and R. Ramalho, "Implementação de um sistema de custeio por actividades nos hospitais do SNS," *Revista Portuguesa de Saúde Pública*, pp. 141–160, 2010.

[171] N. Amaro, C. M. Borges, F. Candoso, A. C. Ferreira, and M. d. C. Valente, "The implementation of a purchasing mechanism for hospital resource allocation in Portugal," *24th PCSI Working Conference - Casemix beyond funding - Contributions for health policy - 8-11 October*, no. October, p. 19, 2008.

[172] J. E. Wennberg, K. Mcpherson, and P. Caper, "Will Payment Based on Diagnosis-related Groups Control Hospital Costs?," *The New England Journal of Medicine*, vol. 311, 2012.

[173] T. English, S. S. Sharfstein, B. Astrachan, and I. L. Muszynski, "AMERICAN OF Psychiatry :," no. February, pp. 131–139, 1986.

[174] Christine Zhenwei Qiang, M. Yamamichi, V. Hausman, and D. Altman, "Mobile Applications for the Health Sector," no. December, pp. 1–61, 2011.

[175] W. H. Organization, "mHealth: New Horizons for Health through Mobile Technologies," tech. rep., 2011.

[176] Infosys, "Patient Relationship Management solution," 2011.

[177] Z.-W. Chen, L.-Z. Fang, L.-Y. Chen, and H.-L. Dai, "Comparison of an SMS text messaging and phone reminder to improve attendance at a health promotion center: a randomized controlled trial.," *Journal of Zhejiang University. Science. B*, vol. 9, no. 1, pp. 34–38, 2008.

[178] K. Fairhurst and A. Sheikh, "Texting appointment reminders to repeated non-attenders in primary care: randomised controlled study.," *Quality {&} safety in health care*, vol. 17, no. 5, pp. 373–376, 2008.

[179] T. M. da Costa, P. L. Salomão, A. S. Martha, I. T. Pisa, and D. Sigulem, "The impact of short message service text messages sent as appointment reminders to patients' cell phones at outpatient clinics in S{ã}o Paulo, Brazil," *International Journal of Medical Informatics*, vol. 79, no. 1, pp. 65–70, 2010.

[180] I. Warren, A. Meads, S. Srirama, T. Weerasinghe, and C. Paniagua, "Push notification mechanisms for pervasive smartphone applications," *IEEE Pervasive Computing*, vol. 13, no. 2, pp. 61–71, 2014.

[181] J. K. Boutcher and M. T. Mateer, "Method and system of providing enhanced appointment notification service to mobile devices." https://www.google.com/patents/US9003328, 2014, visited 2016-08-17, 2014.

[182] J.-h. Huh, N. Kim, and K. Seo, "Design and Implementation of Mobile Push Service-Based Mobile Medication-Hour Notification System Medicine Intake Notification System Using Mobile Push Service The application prototype has been implemented as follows : The list on the left is In the sys," vol. 117, pp. 92–96, 2015.

[183] Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and iOS users' differences concerning security and privacy," *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, p. 817, 2013.

[184] G. Crowd, "Best Mobile Development Plataforms." https://www.g2crowd.com/categories/mobile-development-platforms, 2016, visited 2016-08-22, 2016.

[185] E. Adegbite, "PyFCM." https://github.com/olucurious/PyFCM, 2016, visited 2016-07-27, 2015.

[186] Google Company, "Creating a Navigation Drawer." https://developer.android.com/training/implementing-navigation/nav-drawer.html, 2016, visited 2016-08-23, 2016.

[187] Google Company, "Set Up a Firebase Cloud Messaging Client App on Android." https://firebase.google.com/docs/cloud-messaging/android/client, 2016, visited 2016-08-23, 2016.

[188] B. Schmidt, "Prof of Principle Studies," *epilepsy research*, vol. 68, pp. 48–52, 2006.

[189] I. Osita, I. Onyebuchi, and J. Nzekwe, "Organization's stability and productivity: the role of SWOT analysis an acronym for strength, weakness, opportunities and threat," *International Journal of Innovative and Applied Research*, vol. 2, no. (9): 23- 32, pp. 1–12, 2014.

GLOSSÁRIO

B

BACK-END *Das diversas camadas entre hardware e o utilizador, este representa a abstração de todo o trabalho subjacente, significando a apresentação de dados ao utilizador.*. 86, 92, 126

C

CLOUD COMPUTING *Utilização de recursos computacionais, assim como capacidade de processamento em computadores ou servidores, ligados através da Internet.*. 147

E

EXABYTE *Segundo Sistema Internacional um exabyte equivale a* 1.000.000.000.000.000.000 *Bytes.*. 3

F

FRONT-END *Das diversas camadas entre hardware e o utilizador, este representa a camada que antecede a interface responsável por armazenar dados.*. 126

H

HARDWARE *Terminologia aplicada ao aspeto físico de uma implementação tecnológica.*. 12, 28

HEALTHCARE *Campo científico relacionado com a manutenção e restauração da saúde física e mental.*. 19

I

INPUT *No âmbito da informática representa a entrada de dados.*. 12, 83, 113

INTERFACE *Pode expressar a presença de ferramentas para o uso ou movimentação de qualquer sistemas de informações, seja material ou virtual.*. vi, ix, 10, 20, 52, 53, 58, 66–68, 70–73, 75, 81, 82, 85, 91, 93–95, 99, 104, 108, 109, 116, 138, 140, 141, 149, 152, 154

N

NETWORK *Capacidade de estabelecer contacto com algo ou alguém. Em ciências computacionais pode representar um rede de computadores ou servidores a trabalha em conjunto.*. 27, 36, 38, 39, 69, 87

O

OUTPUT *No âmbito da informática representa a saída de dados.*. 12

P

PUSH NOTIFICATION *É o envio de informação por parte de uma aplicação sem que o recetor tenha requisitado tal informação.*. 123, 131, 141, 142, 154

Q

QUERY *Pode ser vista como uma questão que utiliza um modelo formal utilizando linguagens como SQL. A pergunta pode ser do tipo select ou tipo action consoante a reação que despoletar.*. 49, 78, 113

S

SCRIPT *Refere-se a linguagens com capacidade de serem executadas no interior de programas e/ou outras linguagens de programação não estando restringidas ao ambiente onde estão inseridas.*. 47, 72, 74, 75, 77, 93, 99, 101, 128, 131, 132, 149

SOFTWARE *São uma sequência de instruções a serem seguidas e/ou executadas na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.*. 5, 12

T

TELEMEDICINA *Utilização de tecnologias modernas de informação e telecomunicações para processamento de informação relativa a pacientes ou cuidados de saúde.*. 36–38, 42, 44, 119, 120, 154

THREADING *Em computação, representa a quantidade mínima de instruções em forma de código que pode ser executada por um scheduler que por sua vez faz parte de um sistema operativo.*. 72, 149

W

WIRELESS *Redes sem fios representam uma infraestrutura de comunicações que permite a transmissão de dados.*. 15, 142

A

APÊNDICE A

A.1 PUBLICAÇÕES

A.1.1 *An Approach for the Semantic Interoperability*

AUTORES: Júlio Duarte, Magda Amorim, Filipe Miranda

PUBLICAÇÃO: Applying Business Intelligence to Clinical and Healthcare Organizations

ANO: 2016

ESTADO: Publicado

ABSTRACT: The semantic and syntactic interoperability introduces the capability of two machines to communicate and understand each other improving then the quality of Electronic Health Records.

In this work, is presented an independent application of the medical record, using a web service (with protocol TCP/IP) capable of provide human interaction with interfaces in different devices (web, android app., browser).

SNOMED CT is a comprehensive and scientifically validated health care terminology resulting in an organized computer processable collection of medical terms. This can be mapped into other systems of codes like ICD also used in our application. A data base (SNOMED codes and relations) was created capable of answer to all sort of queries from the users using the browser or the mobile application.

The first hospital unit to enjoy this system was the pathological anatomy unit of the CHAA hospital. Here after receiving a "piece" and a task, the responsible performs all kind of procedures with the purpose of performing a report (diagnosis for example). With the implementation of SNOMED CT, to produce reports a physician could search for the name of the "piece" or the code and immediately upload in that patient HER the diagnosis. The usage of this codes leads to report uniformed that could be read and understood around the world. Another important feature of the application is

the incorporation within the AIDA, *Agéncia para a Integração, Difusão e Arquivo de Informação Médica - Processo Clínico Eletrónico (AIDA-PCE)* and *Agéncia para a Integração, Difusão e Arquivo de Informação Médica - Business Intelligence (AIDA-BI)*

Experiments with real user show a successful software implementation judging by the utilization rate and medical personal acceptance. The mobile application should suffer an upgrade allowing the patient usage for example.

A.1.2 *An Online-Processing Critical Patient Monitoring System- An Interoperability Overview*

AUTORES: Rui Rodrigues, Filipe Miranda, Filipe Portela, José Machado , António Abelha, Manuel Filipe Santos

REVISTA: Comsis

ANO: 2016

ESTADO: Submetido

ABSTRACT: With the increasing expansion of Healthcare Information Systems, the healthcare platforms for interoperability and monitoring systems have become vital sources of clinical information. In this context, the INTCare system was developed with the purpose to create new useful knowledge for decision support in real-time. It is a potential area to develop effective systems for the prediction of clinical events, such as organ failure and death in intensive care. The INTCare uses multiple data sources that are collected at the bedside. However, this system fails on the recognition of the patient absence in bed. This problem led to the development of the Patient Localization and Management System (PaLMS), a RFID localization and monitoring system. The paper describes the PaLMS implementation as well as an intelligent Multi-Agent System for the integration of PaLMS into the hospital platform for interoperability (AIDA platform). An usability evaluation was performed in order to assess the level of usability of the existing systems at Centro Hospitalar do Porto, such as the PaLMS, the INTCare and the AIDA platform. Despite the fact that physicians and some nurses feel that they lose too much time with records, the usability of the systems was evaluated with a high level, demonstrating the success of the PaLMS implementation.

B

APÊNDICE B

Pela importância, que representa para os resultados e para a discussão do caso de estudo presente no capítulo 5, optou-se por inserir neste capítulo imagens que representam o resultado final da implementação.

B.1 FUNCIONALIDADES DA APLICAÇÃO WEB

Para uma melhor compreensão dividiu-se por secções os resultados alcançados. Primeiro é necessário apresentar a forma como foi implementado o acesso à página, o *login* (ver figura 26).

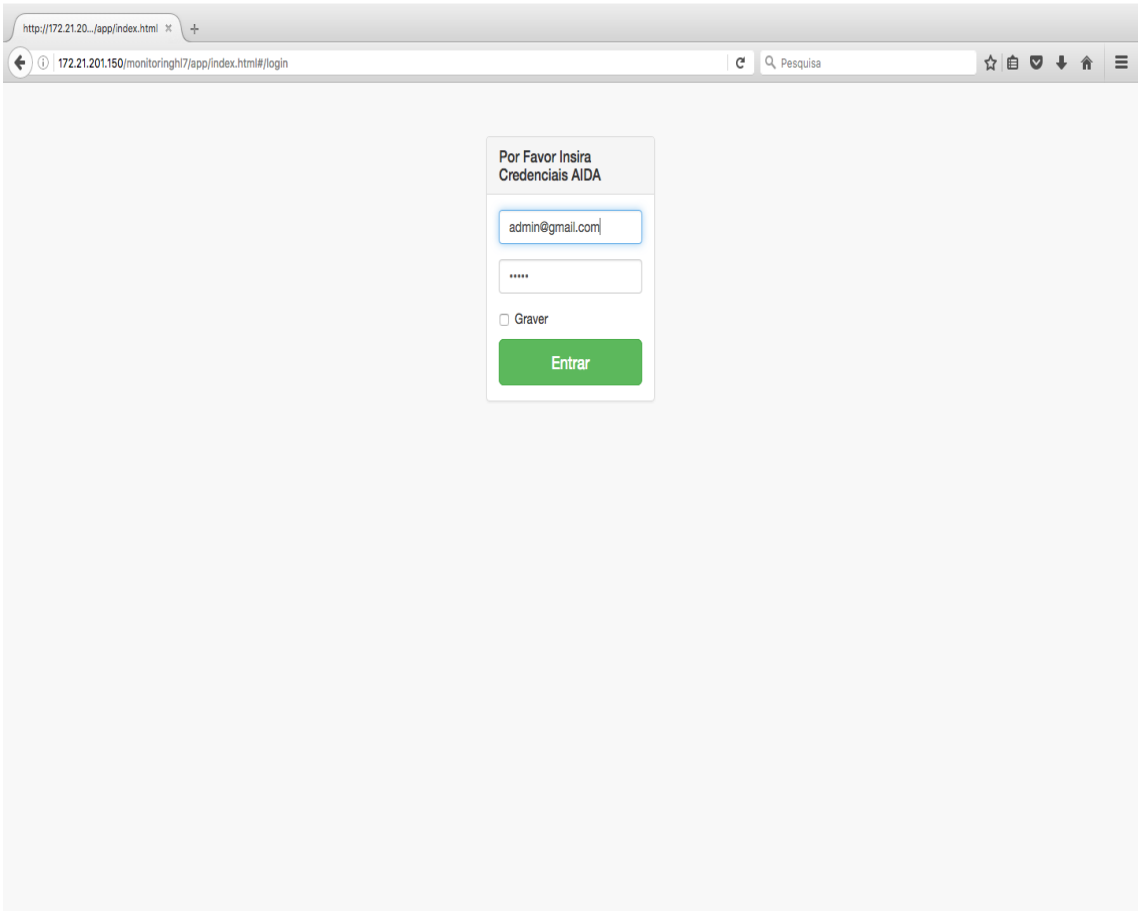


Figura 26.: Resultado final da aplicação web - *Login*.

B.1.1 *Vista Inicial*

Após o *login*, a página que aparece é vista inicial dos agentes HL7.

HL7

A vista inicial referente aos agentes HL7 já implementados no CHP.

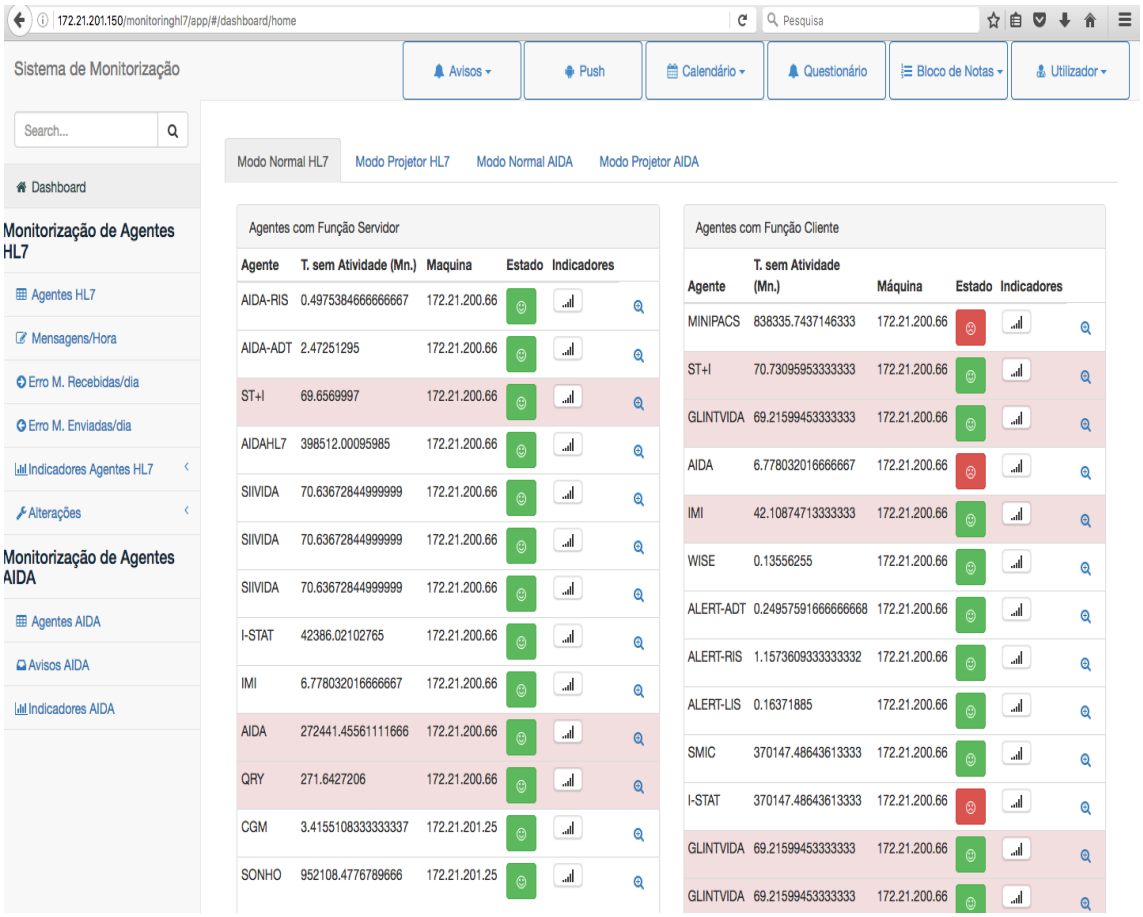


Figura 27.: Resultado final da aplicação web - Vista Inicial HL7.

AIDA

A vista inicial referente aos agentes HL7 já implementados no CHP.

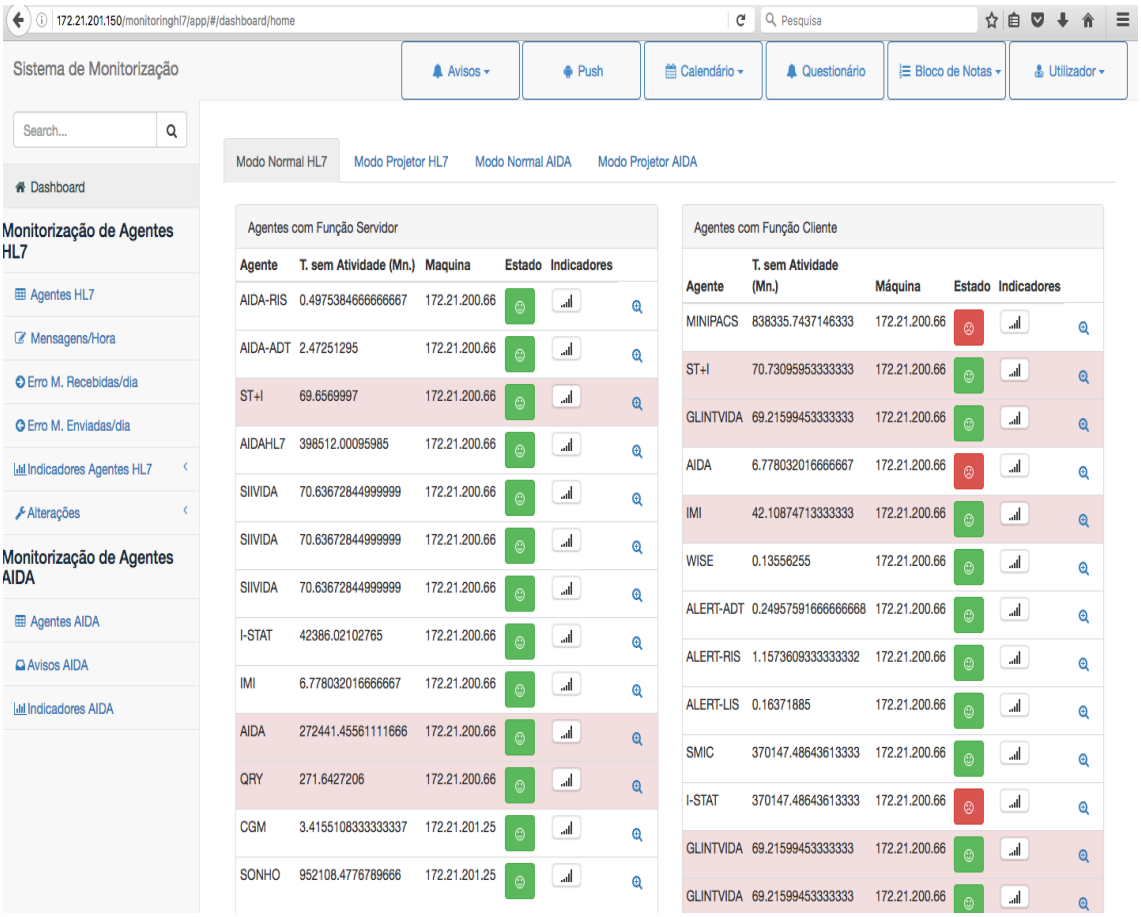


Figura 28.: Resultado final da aplicação web - Vista Inicial HL7.

Moso Projeção

A vista inicial referente aos agentes HL7 e AIDA já implementados no CHP, em modo projeção com *reload* automático.

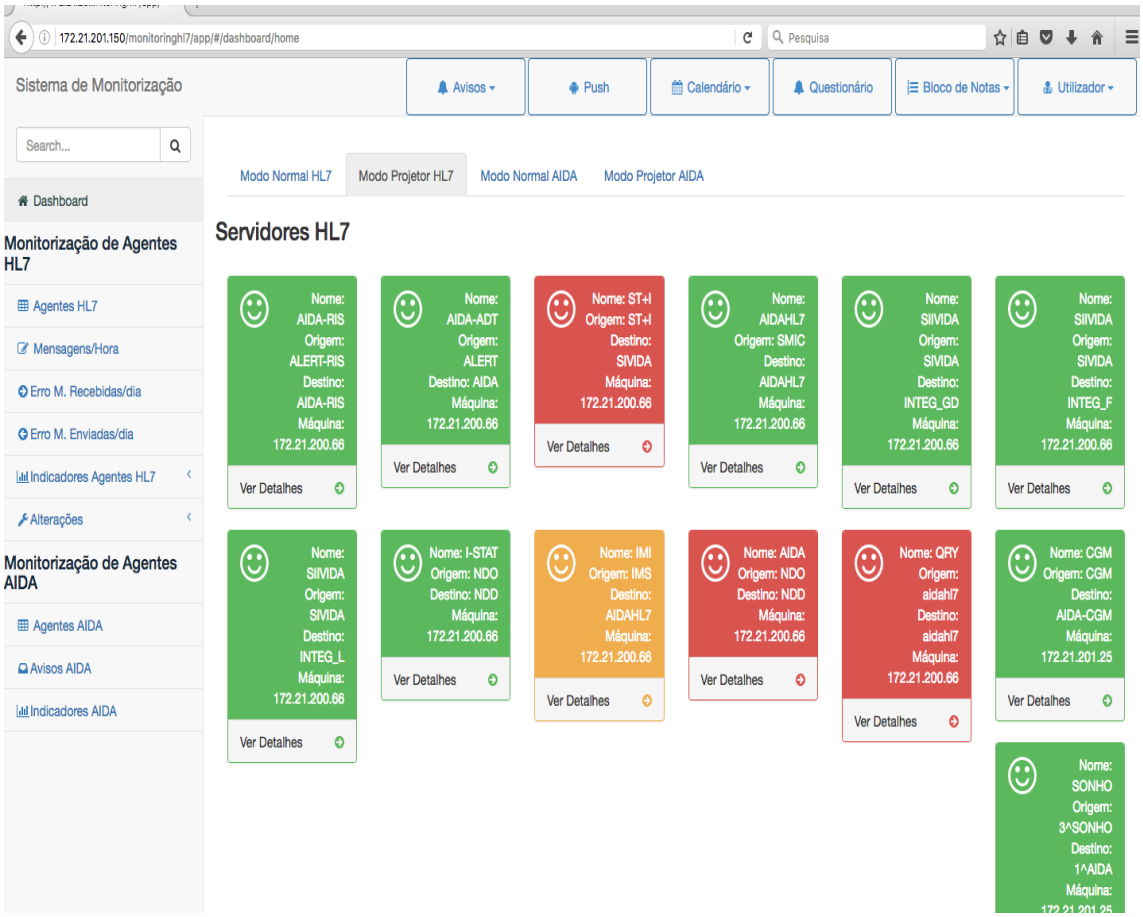


Figura 29.: Resultado final da aplicação web - Vista Inicial em Modo Projeção.

B.1.2 *Agentes HL7*

Na barra lateral é possível escolher a opção para ver apenas agentes HL7. Posto isto, pode ser escolhido um agente para ver detalhes do seu funcionamento.

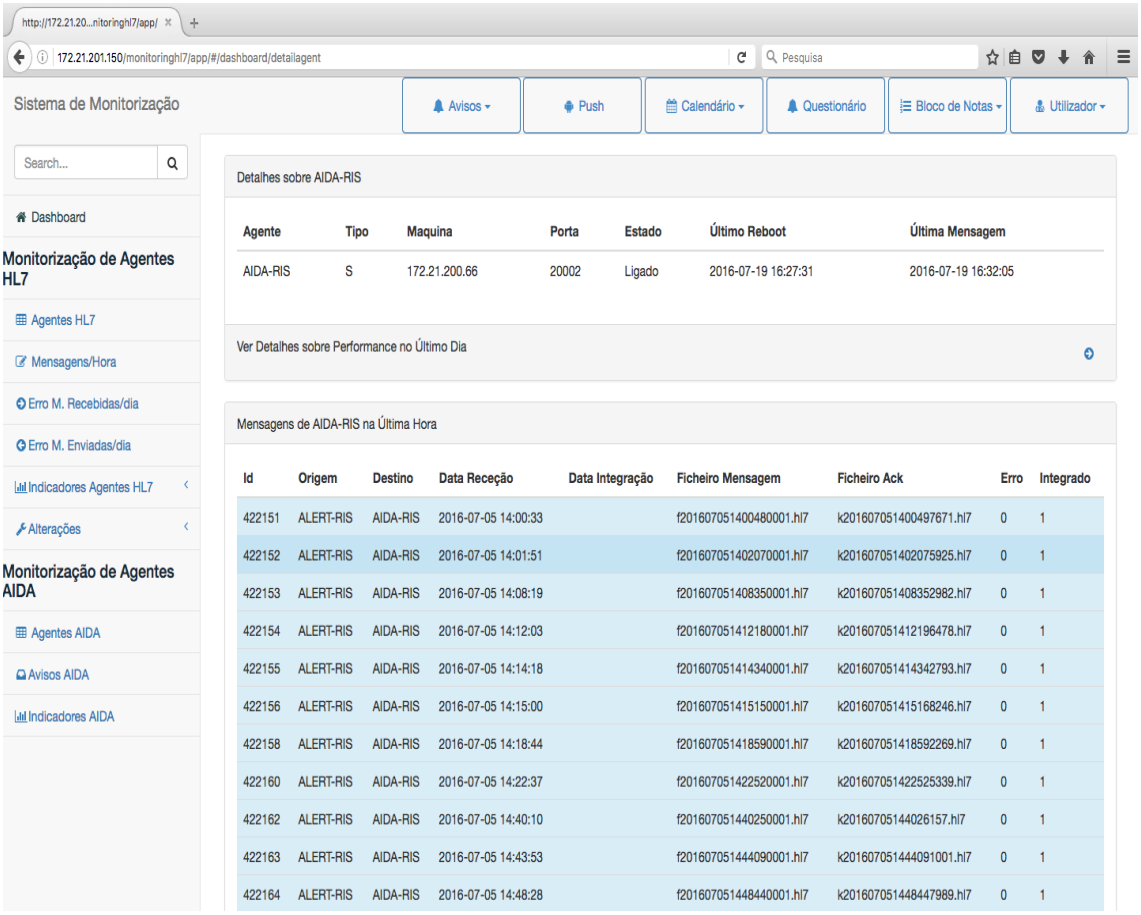


Figura 30.: Resultado final da aplicação web - *Zoom* nas funcionalidades de determinado agente.

B.1.3 *Controlo de Agentes*

Na aba respectiva à alteração de agentes é possível, iniciar ou para um agente pertencente ao MAS, assim como alterar definições.

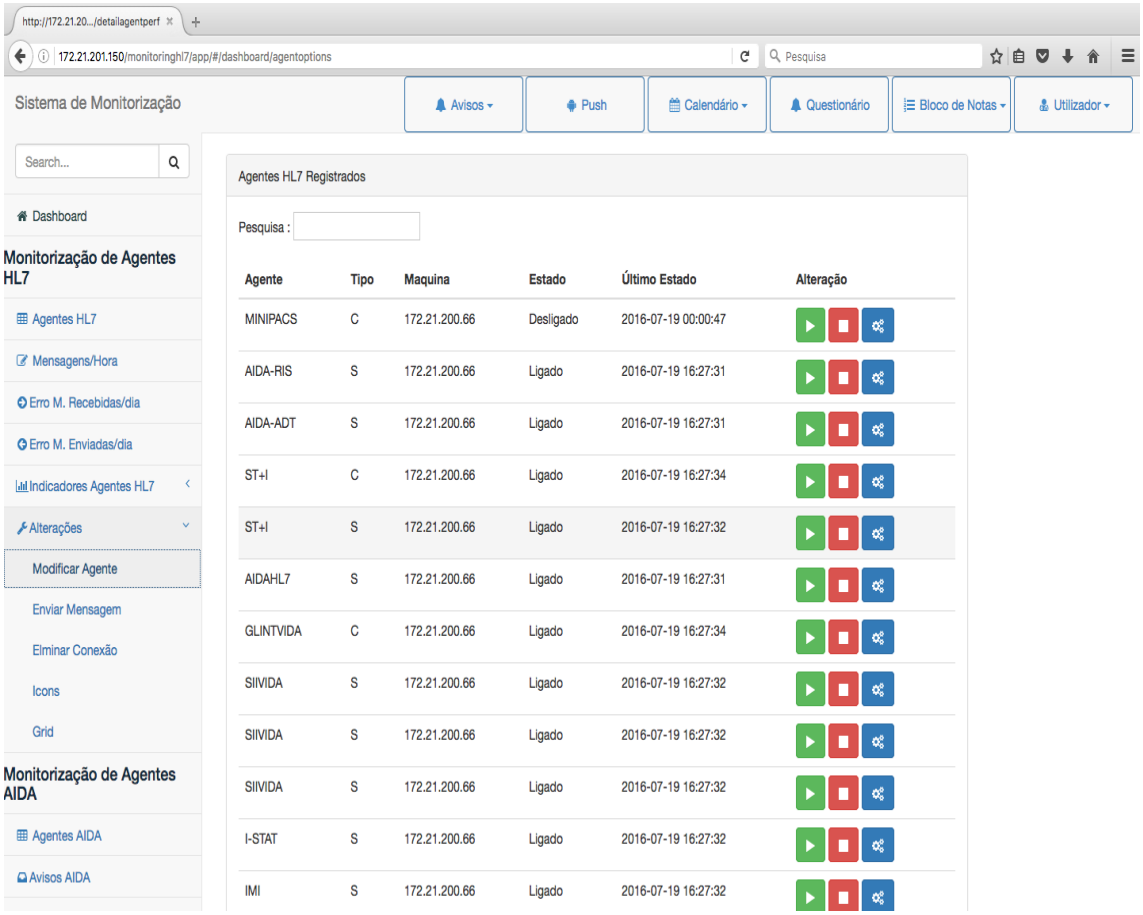


Figura 31.: Resultado final da aplicação web - Controlo de agentes pertencentes ao MAS.

B.1.4 *Agentes AIDA*

Na barra lateral é possível escolher a opção para ver agentes AIDA assim como, avisos sobre estes.

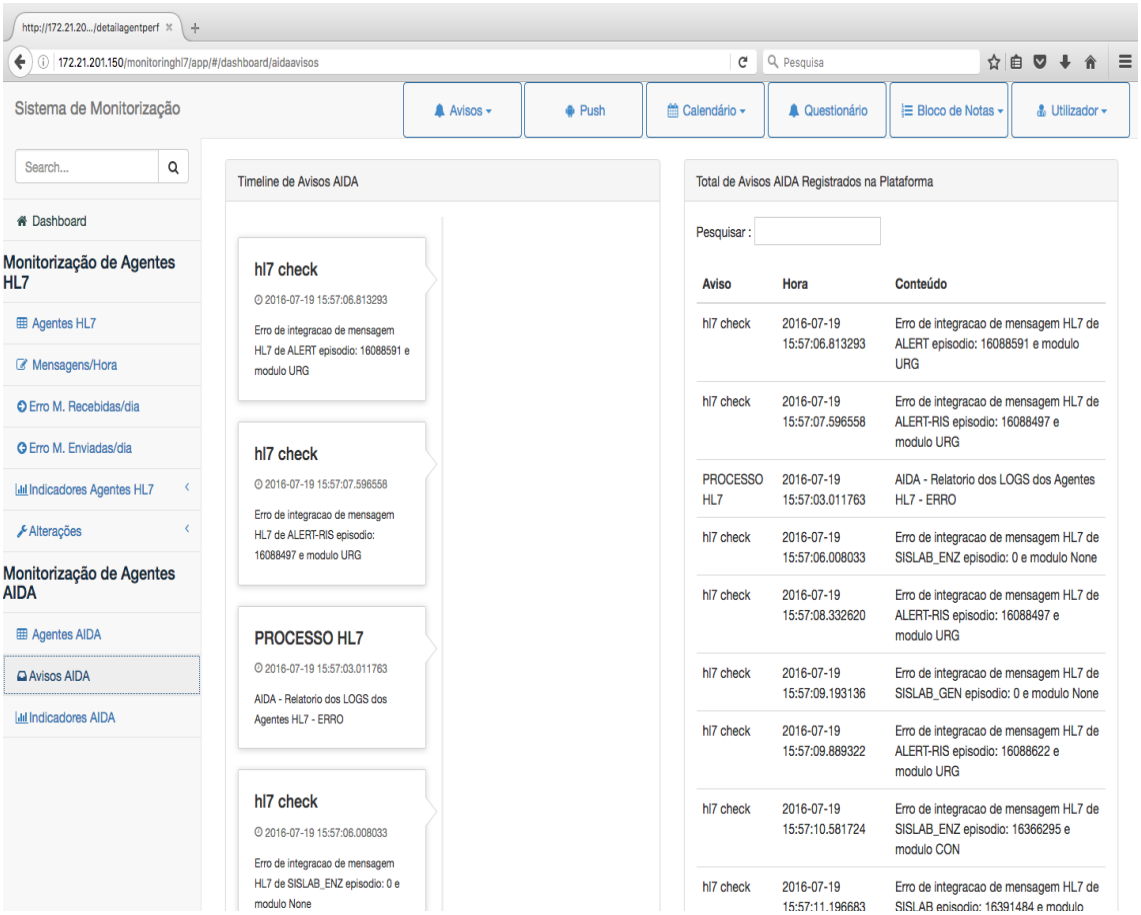


Figura 32.: Resultado final da aplicação web - Avisos sobre agentes AIDA.

B.1.5 *Indicadores*

Na aba de indicadores sobre mensagens é possível escolher entre último dia ou dois dias anteriores. Em cada caso é ainda possível a escolha por agentes servidor ou cliente.

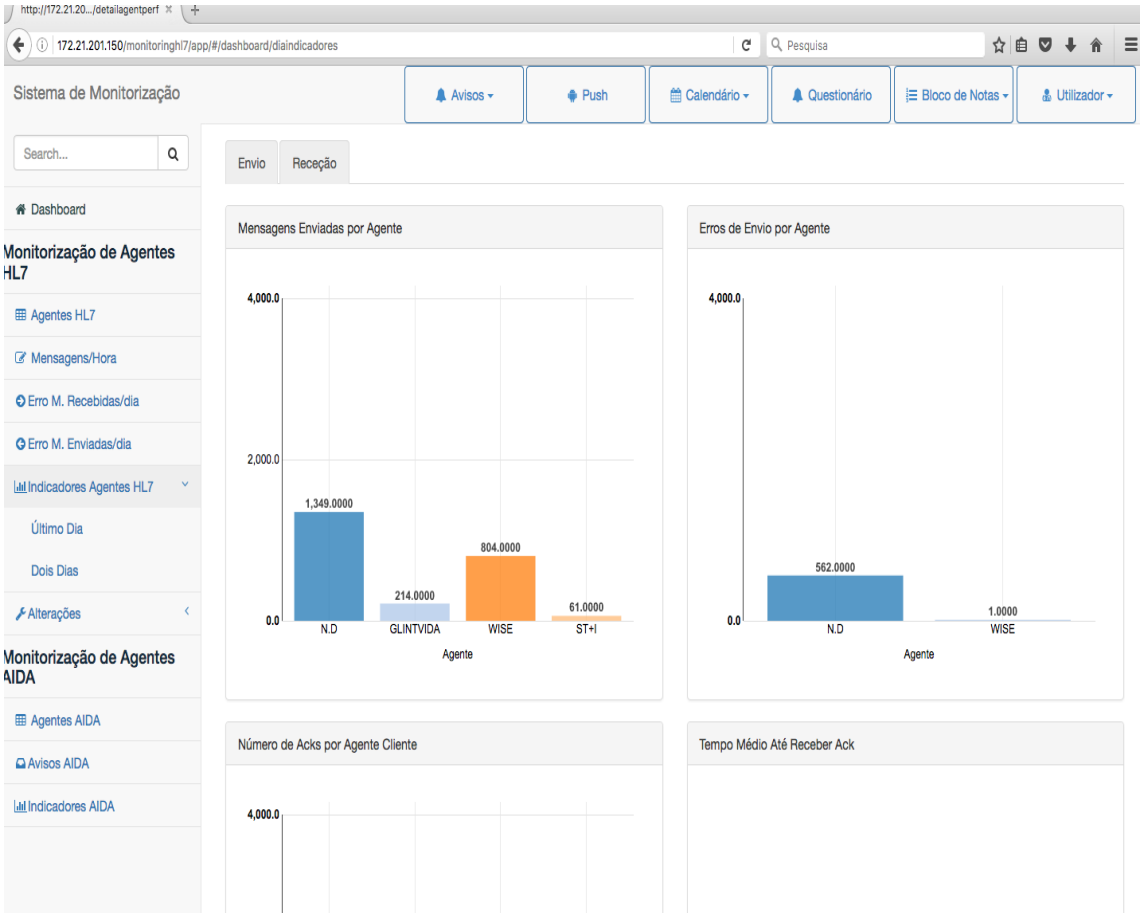


Figura 33.: Resultado final da aplicação web - Indicadores espaço temporal de um dia.

Observando para os dias anteriores.

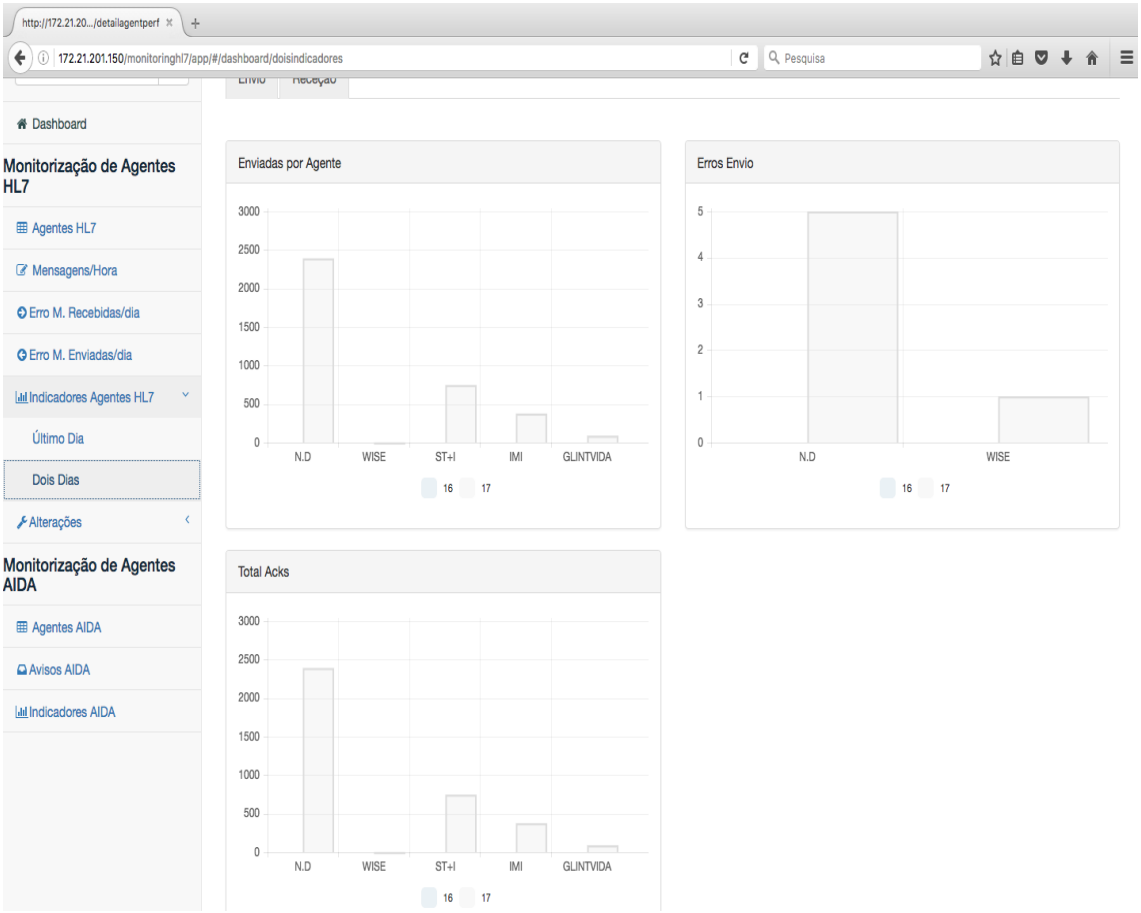


Figura 34.: Resultado final da aplicação web - Indicadores espaço temporal 2 a 3 dias.

B.1.6 *Mensagens*

A troca de mensagens é um ponto vital da monitorização, é importante analisar as que obtiveram ACK positivo bem como, as que por algum motivo apresentam erros.

Normal

A vista quando as mensagens estão em conformidade, são marcadas com a cor verde.

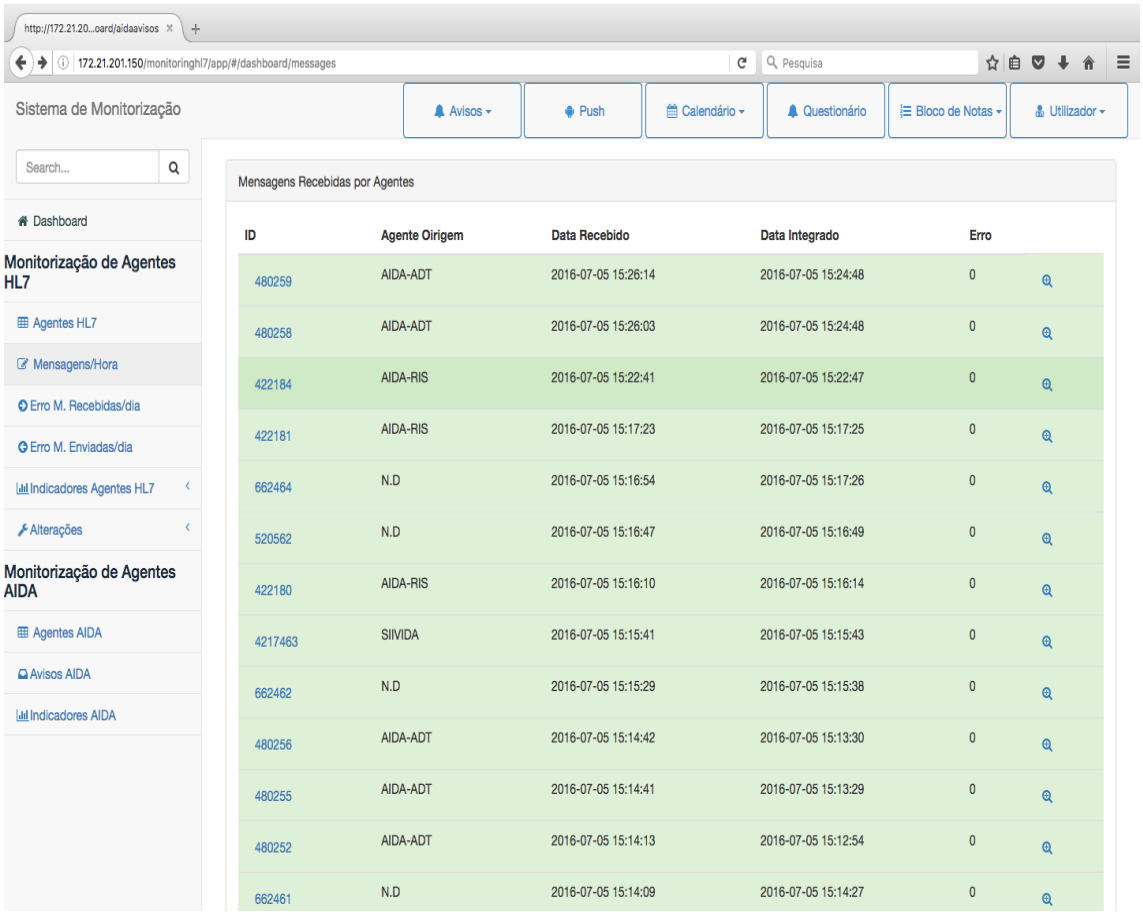


Figura 35.: Resultado final da aplicação web - Mensagens com ACK positivo.

Erros

É interessante analisar em separado as mensagens que apresentam erros HL7.

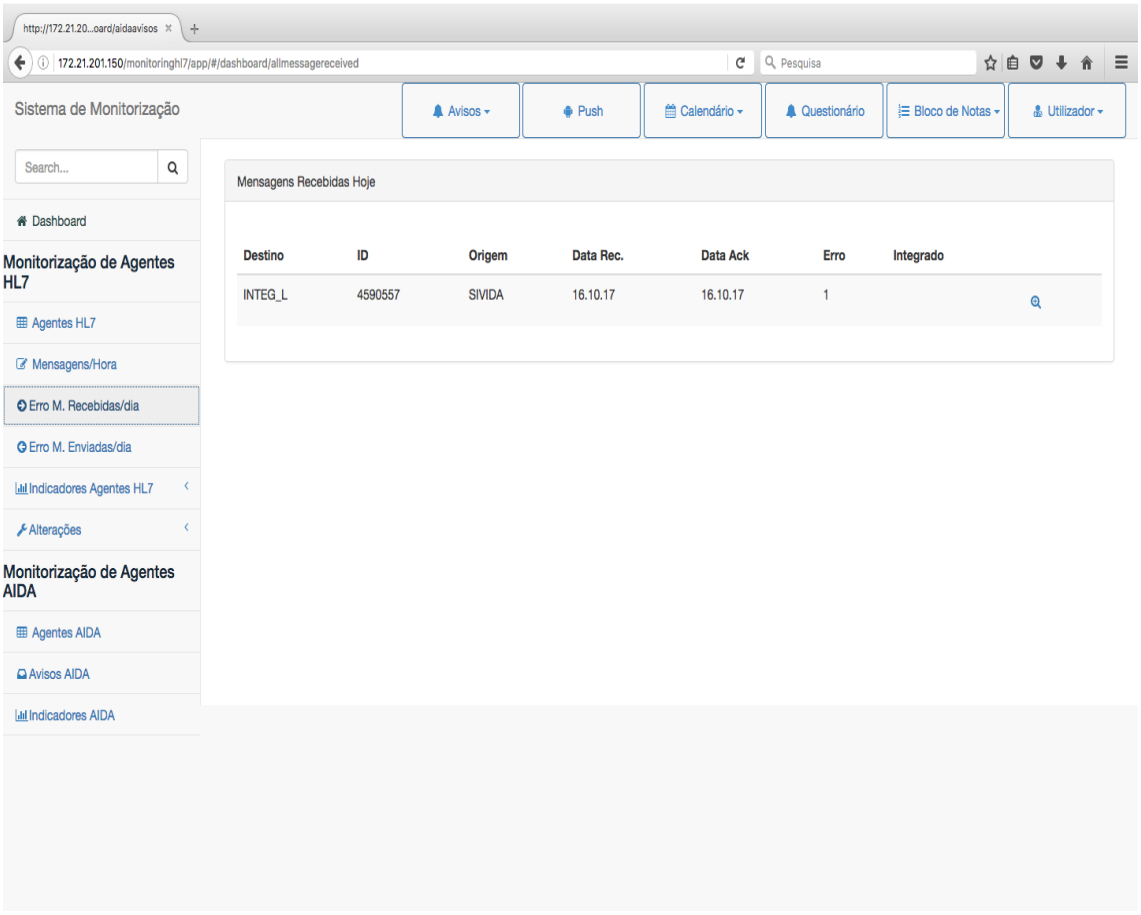


Figura 36.: Resultado final da aplicação web - Mensagens com erro.

B.1.7 *Extras*

A aplicação mais que um sistema de monitorização, pretende ser um aplicativo onde podem ser enviados avisos, consulta de data e hora e ainda agendamento de situações utilizando o bloco de notas.

Bloco Notas

Um bloco notas é apenas um exemplo de uma funcionalidade que pode facilitar a vida aos utilizadores.

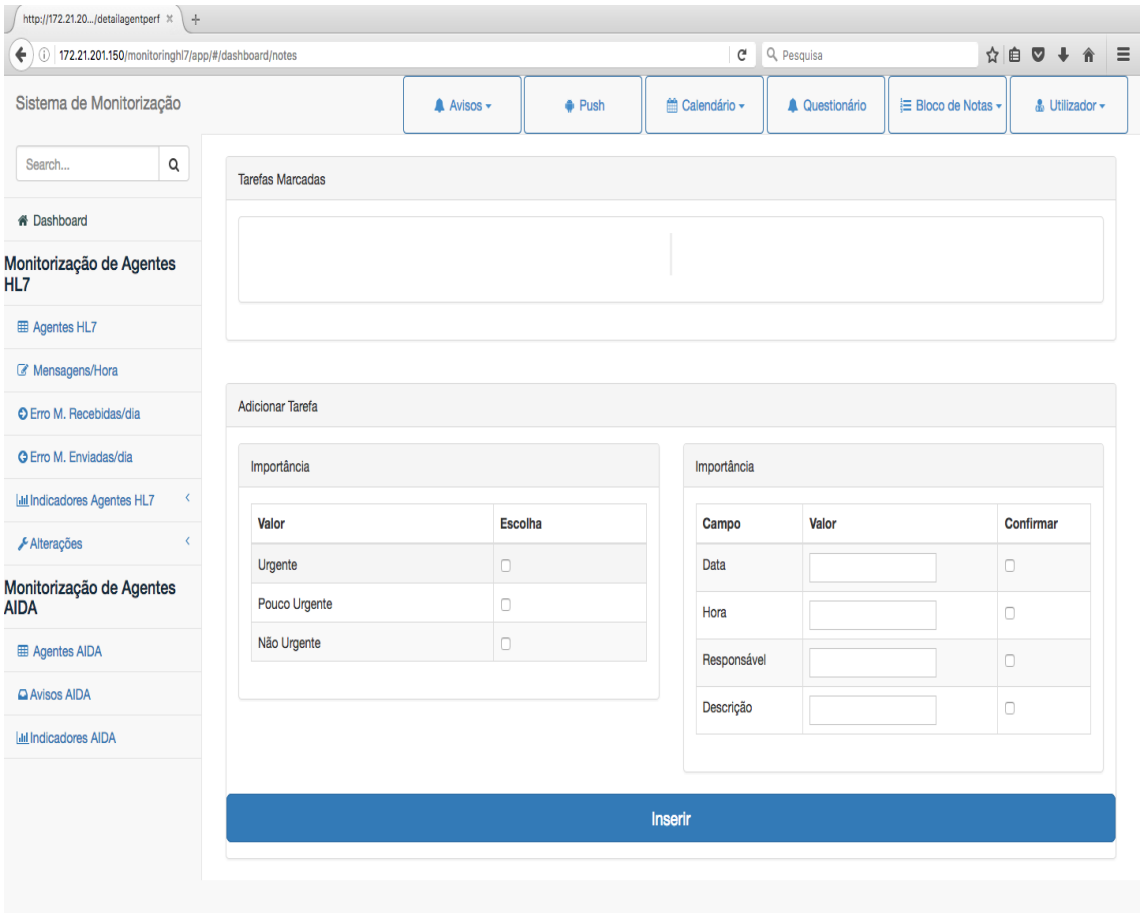


Figura 37.: Resultado final da aplicação web - Bloco de Notas.

B.2 APLICAÇÃO DE REGISTO NO CENTRO HOSPITALAR DO PORTO

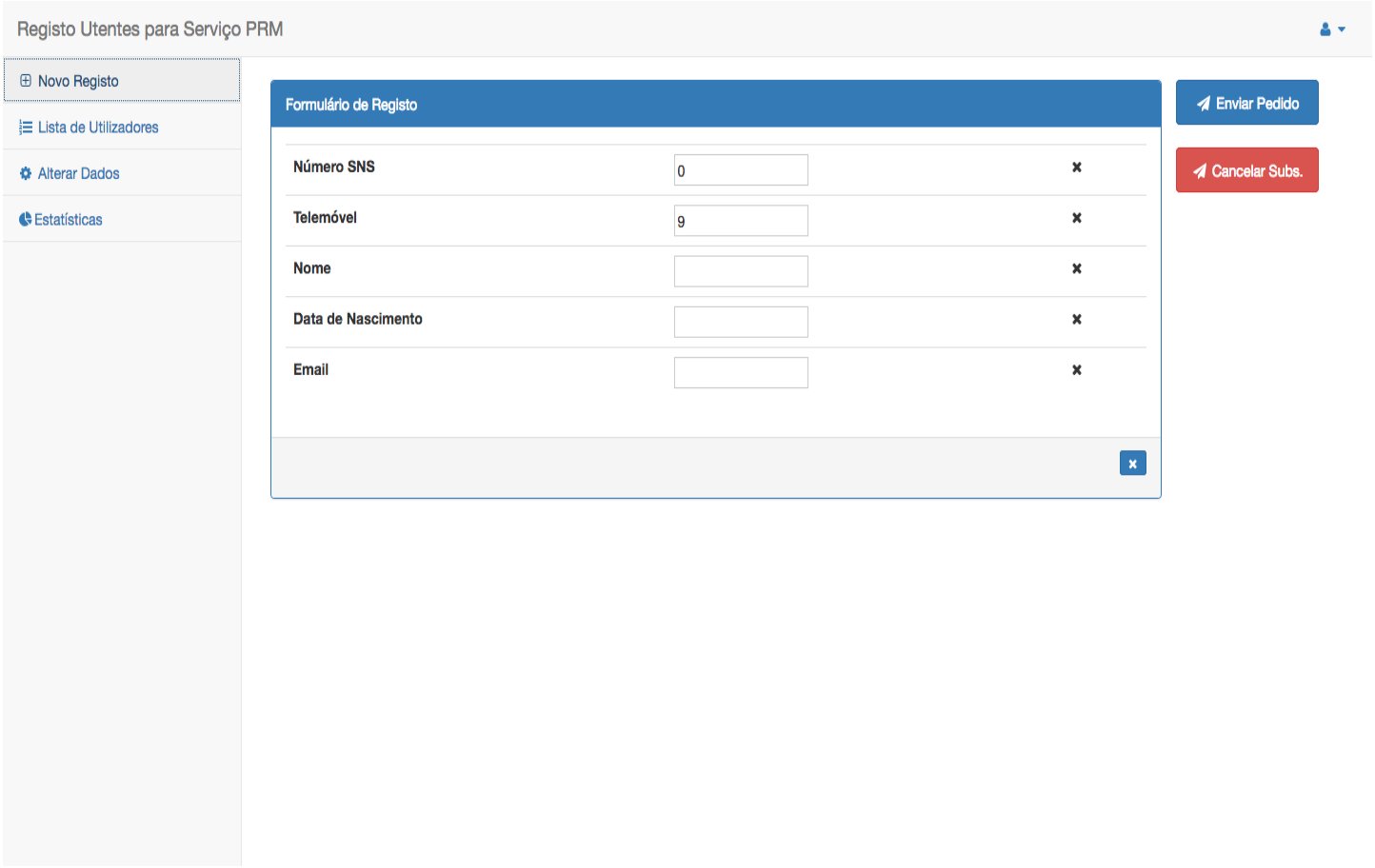


Figura 38.: Resultado final da aplicação web - Registo de Clientes no CHP.