

Pedro Tiago Magalhães Gomes

Master Patient Index



Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Julho de 2009

Pedro Tiago Magalhães Gomes

Master Patient Index

*Relatório de Estágio submetido à Faculdade de Ciências da
Universidade do Porto como parte dos requisitos para a obtenção do grau de
Mestre em Engenharia de Redes e Sistemas Informáticos*

Orientador: Eng^o António Cardoso Martins (Siemens)

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Julho de 2009

Agradecimentos

Agradeço ao meu orientador, o Eng. António Cardoso Martins e ao Eng. Daniel Ramos por todo o apoio, orientação, paciência e tempo que despenderam comigo, tal como pela confiança que em mim depositaram para levar este projecto a bom porto. Nunca esquecendo o resto da equipa da qual fiz parte, e que me acolheu muito bem.

Um agradecimento especial a todos os que comigo estagiaram, pois foi juntos que começamos esta aventura e juntos ultrapassamos todas as adversidades com que nos deparamos. Obrigado pelos imensos debates sobre o que usar, como usar e o porquê de o fazer.

Por último, não posso deixar de agradecer à minha família, namorada e amigos que sempre me apoiaram, e que estiveram presentes nos momentos mais difíceis.

Abstract

Nowadays there is a great diversity of information systems within all healthcare providers. Each one with different specifications and capabilities as well as proprietary communication methods, thus hardly allowing scalability. This heterogenic set of characteristics is an impediment to achieve interoperability between systems which, indirectly, affects the patients well being. It is common that, upon watching each database of each one of these information systems, we notice different entries referencing the same person; entries with insufficient or wrong data, due to errors or miscomprehension upon patient data insertion; and outdated data. These problems bring duplicity, incoherence, lack of actualization and dispersion on patient data.

It is with the purpose of minimizing these problems that the Master Patient Index concept is needed. A Master Patient Index proposes a centralized repository, to index all patient entries within a pre-determined set of information systems. This repository is composed by a set of demographic data, sufficient to unmistakably identify a person, and a list of identifiers that unequivocally identify the several entries that a patient possesses within each information system. This solution allows synchronism between all intervenients minimizing incoherence, lack of actualization, inexistent data, and diminishing entry duplications.

The Master Patient Index is an asset not only to the patients but also to the medical staff and healthcare providers.

Resumo

Nos actuais estabelecimentos de saúde existe uma grande diversidade de sistemas de informação. Cada um com especificidades e capacidades diferentes, métodos de comunicação proprietários, e dificilmente permitem escalabilidade. Este conjunto de características dificulta a interoperabilidade de todos estes sistemas, na procura do bem do paciente. É vulgar que, ao observarmos todas as bases de dados de cada um destes sistemas de informação, nos deparemos com registos diferentes que referenciam a mesma pessoa; registos com dados insuficientes; registos com dados errados, devido a erros ou incompreensão aquando da inserção dos dados do paciente; e registos com dados desactualizados. Estes problemas provocam duplicidade, incoerência, desactualização e dispersão nos dados do paciente.

É com o intuito de minimizar estes problemas que o conceito de um Master Patient Index é necessário. Um Master Patient Index propõe um repositório centralizado, que indexa todos os registos dos pacientes de um determinado conjunto de sistemas de informação. Repositório que é constituído por um conjunto de dados demográficos, suficientes para identificar inequivocamente uma pessoa, e uma lista de identificadores, que identificam os vários registos que o paciente possui nos repositórios de cada sistema de informação. Esta solução permite sincronismo entre todos os intervenientes, minimizando a incoerência, a desactualização, a inexistência de dados, e uma diminuição na duplicação dos registos.

O Master Patient Index é uma mais valia para os pacientes, para o corpo médico e para os prestadores de cuidados de saúde.

Conteúdo

Abstract	4
Resumo	5
Lista de Figuras	11
Listagens	12
1 Introdução	13
1.1 Siemens	13
1.1.1 Marcos Recentes em Portugal	15
1.2 O Projecto	16
1.2.1 Master Patient Index	17
1.2.2 Visão	18
1.3 Estrutura do Relatório	20
2 Estado da Arte	22
2.1 Health Level 7	22
2.1.1 HL7 V2	22
2.1.2 HL7 V3	23
2.2 Integrating the Healthcare Enterprise	24

2.2.1	Patient Identifier Cross-reference	24
2.2.2	Patient Demographics Query	26
2.3	Sistemas Semelhantes	26
2.3.1	Tayside Patient Index	27
2.3.2	VistA / Austin - US Department of Veterans Affairs	28
2.3.3	Composite Application Plataform Suite - SUN	29
3	Projecto Master Patient Index	30
3.1	Casos de Uso	30
3.1.1	Consumidor e funcionalidades associadas	30
3.1.2	Administrador e funcionalidades associadas	31
3.1.3	Enterprise Master Patient Index e funcionalidades associadas . .	32
3.1.4	Master File Manager e funcionalidades associadas	33
3.2	Cenários	34
3.2.1	Pessoa desconhecida em todos os sistemas	34
3.2.2	Pessoa conhecida apenas pelo MPI	36
3.2.3	Existir um EMPI na rede	37
3.3	Repositório	38
3.4	Arquitectura Geral	39
3.4.1	HL7 Sender/Receiver	40
3.4.2	Reference Informational Model	40
3.4.3	Matching Engine	40
3.4.4	Object-Relational Mapping	40
3.4.5	Web Server	41
3.4.6	Subscribers Manager	41
3.4.7	Master Files Client	41

3.4.8	Log	41
4	Implementação	44
4.1	Ferramentas e tecnologias usadas no projecto	44
4.1.1	Java	44
4.1.2	HL7 Application Programming Interface	45
4.1.3	Java Sig	45
4.1.4	Hibernate	45
4.1.5	ZK	46
4.2	Interoperabilidade	46
4.2.1	Adição de uma pessoa	47
4.2.2	Actualização de uma pessoa	48
4.2.3	Pesquisar por pessoas	48
4.2.4	Notificar os consumidores	49
4.2.5	Sincronismo dos Master Files	50
4.2.6	Servidor	50
4.3	Hibernate	52
4.4	Algoritmos de correlação	55
4.5	Interface gráfico	58
5	Conclusão	63
5.1	Objectivos Cumpridos	63
5.2	Dificuldades Sentidas	65
5.3	Trabalho Futuro	66
5.4	Conclusões finais	66
A	Acrónimos	68

B Casos de Uso	69
C Atributos que o MPI suporta	70
D Mensagens HL7	71
D.1 ADT^A01 - Notificação de admissão/visita	71
D.2 ADT^A04 - Registrar um Paciente	72
D.3 ADT^A05 - Pré-admissão de um Paciente	73
D.4 ADT^A28 - Adicionar informação de um Paciente/Pessoa	74
D.5 ACK - Acknowledgement	75
D.6 QBP^Q21 - Obter dados demográficos de uma Pessoa	75
D.7 RSP^K21 - Resposta a uma QBP^Q21	75
D.8 QBP^Q22 - Pesquisar por Pessoas	76
D.9 RSP^K22 - Resposta a uma QBP^Q22	76
D.10 MFN^M14 - Notificação para um <i>Master File</i>	76
D.11 MFK^M14 - Acknowledgement para a MFN^M14	77
E Segmentos das mensagens HL7	78
E.1 MSH - Cabeçalho de uma mensagem	78
E.2 PID - Informação sobre o Paciente	79
E.3 PD1 - Informação extra sobre o Paciente	80
E.4 NK1 - Informação sobre os parentes	81
E.5 AL1 - Alergias	82
E.6 DB1 - Incapacidades	82
E.7 QPD - Definição dos parâmetros para pesquisa	82
Referências	83

Lista de Figuras

1.1	Falta de sincronismo e coerência entre os vários sistemas.	16
1.2	MPI implementado numa unidade médica.	18
1.3	Exemplo de uma rede com vários Enterprise Master Patient Index . . .	20
2.1	Diagrama de fluxo do PIX (adaptado do IHE ITI Technical Framework)	25
2.2	Diagrama de fluxo do PDQ (adaptado do IHE ITI Technical Framework)	26
3.1	Interacção do consumidor com o MPI	31
3.2	Interacção do administrador com o MPI	32
3.3	Interacção do EMPI com o MPI	33
3.4	Interacção do MFM com o MPI	34
3.5	Diagrama de Sequência para o caso de uma pessoa ser desconhecida a todos os sistemas (Adaptado da ideologia do IHE e do HL7)	35
3.6	Diagrama de Sequência para o caso de uma pessoa ser conhecida pelo MPI e não pelo estabelecimento (Adaptado da ideologia do IHE e do HL7)	36
3.7	Adaptado do HL7 RIM Patient Registries Patient Administration (PRPA DM000000UV)	38
3.8	Excerto do Modelo da Base de Dados	42
3.9	Arquitectura de alto nível	43
4.1	Análise dos vários métodos de correlação	56

4.2	Passos necessários a um algoritmo de correlação eficiente	57
4.3	Ecrã inicial do interface gráfico	60
4.4	Exemplo do resultado dos algoritmos de correlação	61
4.5	Opção para adicionar dados através de um ficheiro csv	62

Listagens

4.1	Exemplo de um servidor (adaptado do HAPI)	50
4.2	Exemplo de uma conexão ao servidor (adaptado do HAPI)	51
4.3	Classe Patient.Consumer	52
4.4	Ficheiro XML responsável por mapear a classe Consumer	53
4.5	Manuseamento da classe Consumer	54
4.6	Exemplo da criação de uma pesquisa usando o SOUNDEX	58

Capítulo 1

Introdução

O projecto de estágio foi elaborado nas instalações e em cooperação com a Siemens Healthcare Portugal. Começaremos pela apresentação da empresa e, posteriormente, o projecto desenvolvido.

1.1 Siemens

Com 500 centros de produção em 50 países e presença em 190 países a Siemens está representada em todo o mundo. Em Portugal, a Siemens S.A. dispõe de duas unidades fabris, centro de investigação desenvolvimento de software (Lisboa e Porto) e presença em todo o país, através dos seus parceiros e das suas instalações. Siemens, S.A. A empresa está desde 2008 organizada em três grandes sectores de actividade: Industry, Energy e Healthcare [1].

O **Sector Industry** dispõe de soluções para a indústria nas vertentes de produção, transporte e edifícios, segmentando-se em cinco áreas: Industry Automation and Drive Technologies, Building Technologies, Industry Solutions, Mobility e OSRAM.

O **Sector Energy** disponibiliza produtos e soluções para a geração, transmissão e distribuição de energia eléctrica, segmentando-se em seis áreas: Fossil Power Generation, Renewable Energy, Oil gas, Energy Service, Power Transmission e Power Distribution.

O **Sector Healthcare** oferece um conjunto de produtos inovadores e soluções integradas bem como serviços e consultadoria na área da saúde, segmentando-se em três

áreas: Imaging IT, Workflow Solutions e Diagnostics.

A área Imaging IT disponibiliza sistemas de imagem para diagnóstico precoce e intervenção, bem como para prevenção efectiva, nomeadamente Sistemas de ressonância magnética (MR), Sistemas de tomografia axial computadorizada (CT), Sistemas de radiografia, Sistemas angiográficos digitais, Sistemas de tomografia por emissão de positrões (PET/CT) e tomografia por emissão de fotão único (SPECT e SPECT/CT), Unidades de ecografia, entre outros. Todos os sistemas estão interligados por tecnologias de informação de elevada performance possibilitando uma optimização dos processos a nível dos prestadores de cuidados de saúde (sistemas de gestão hospitalar como o **Soarian**[®], sistemas de processamento de imagem como o **Syngo**[®] e tecnologias knowledge-based como auxiliares de diagnóstico).

A área Workflow Solutions disponibiliza soluções globais para especialidades como a cardiologia, a oncologia e a neurologia. Esta área fornece ainda soluções, por exemplo, para saúde da mulher (mamografia), a urologia, a cirurgia e a audiolgia, englobando igualmente a vertente de consultadoria e soluções globais (soluções globais para prestadores de cuidados de saúde). Simultaneamente, a área de Workflow Solutions engloba a prestação de serviços pós-venda e gestão de clientes.

A área Diagnostics encerra a vertente de diagnóstico in-vitro, incluindo imunodiagnóstico e análise molecular. As soluções da área vão desde os aplicativos point-of-care até à automatização de grandes laboratórios.

Desta forma, o Sector Healthcare é hoje a primeira empresa a nível mundial a disponibilizar um portefólio integrado de tecnologia que permite responder a todas as fases do ciclo de cuidados de saúde.

A **Siemens IT Solutions and Services**, um dos líderes em oferta de serviços na área das Tecnologias de Informação (TI), funciona como unidade de negócio transversal.

Em Portugal, o Sector Healthcare da Siemens S.A. é um dos líderes de mercado no ramo dos cuidados de saúde, reconhecido pelas suas competências e força de inovação em diagnóstico e tecnologias terapêuticas, assim como engenharia de conhecimento, incluindo tecnologias de informação e integração de sistemas.

Nos últimos anos, o Sector Healthcare da Siemens SA tem promovido uma estratégia de contacto e parceria com a Comunidade Académica e Científica em Portugal, no sentido da criação de uma rede de conhecimento e parcerias estratégicas que potenciem a inovação, a investigação e o desenvolvimento (IDI) na área da Saúde. Actualmente o Sector Healthcare conta com um Grupo de IDI com mais de 15 elementos, de-

envolvendo investigação em áreas estratégicas como Sistemas de informação para a Saúde, Imagem Computacional, Análise automática de Imagem Médica, Modelação e ferramentas de suporte à decisão e Avaliação Tecnológica Estratégica, que resultou já no registo de uma patente e submissão de duas outras, bem como na publicação de mais de dez artigos científicos.

1.1.1 Marcos Recentes em Portugal

- Serviços de Patologia Mamária no Hospital de São João, no Porto, no Hospital da Luz, em Lisboa, e na Clínica Dr. João Carlos Costa, em Viana do Castelo - as primeiras unidades com total enfoque no paciente, englobando todas as valências tecnológicas necessárias para todo o processo clínico;
- Hospital da Luz, em Lisboa - primeira unidade hospitalar, em Portugal, a integrar o sistema de informação clínica SOARIAN[®], tornando-se assim numa das mais modernas infra-estruturas de saúde da Europa;
- Clínica Quadrantes, em Lisboa - tecnologia de diagnóstico in vitro e sistemas de tecnologias de informação, que juntamente com um PET/CT complementaram as tecnologias de diagnóstico in vivo Siemens já existentes nesta unidade clínica;
- Universidade de Coimbra - ressonância magnética de 3 Tesla exclusivamente destinada à investigação em neurociência. Esta unidade será utilizada ao abrigo da rede de cooperação científica Brain Imaging Network Grid, que agrupa as Universidades de Coimbra, Aveiro, Porto e Minho;
- Algumas publicações do Grupo de IDI:
 - Registo de patente DE 10 2007 053 393, System zur automatisierten Erstellung medizinischer Reports;
 - F. Soares, P. Andruszkiewicz, M. Freire, P. Cruz e M. Pereira, Self-Similarity Analysis Applied to 2D Breast Cancer Imaging, HPC-Bio 07 - First International Workshop on High Performance Computing Applied to Medical Data and Bioinformatics, Riviera, France (2007);
 - J. Martins, C. Granja, A. Mendes e P. Cruz, Gestão do fluxo de trabalho em diagnóstico por imagem: escalonamento baseado em simulação, Informática de Saúde – Boas práticas e novas perspectivas, edições Universidade Fernando Pessoa, Porto (2007);

- F. Soares, M. Freire, M. Pereira, F. Janela, J. Seabra, Towards the Detection of Microcalcifications on Mammograms Through Multifractal Detrended Fluctuation Analysis, 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada (2009).

1.2 O Projecto

Um estabelecimento de saúde é, na sua generalidade, constituído por inúmeros sistemas de informação distintos e independentes. Cada um possui funcionalidades, capacidades e especificidades diferentes, desde a informação contida na base de dados até aos métodos usados na comunicação com outros sistemas, pois são pensados e construídos com objectivos muito dispares, mas muito precisos. Diferenças que tornam a comunicação, o sincronismo e a coerência dos dados entre estes sistemas, um objectivo árduo de alcançar.

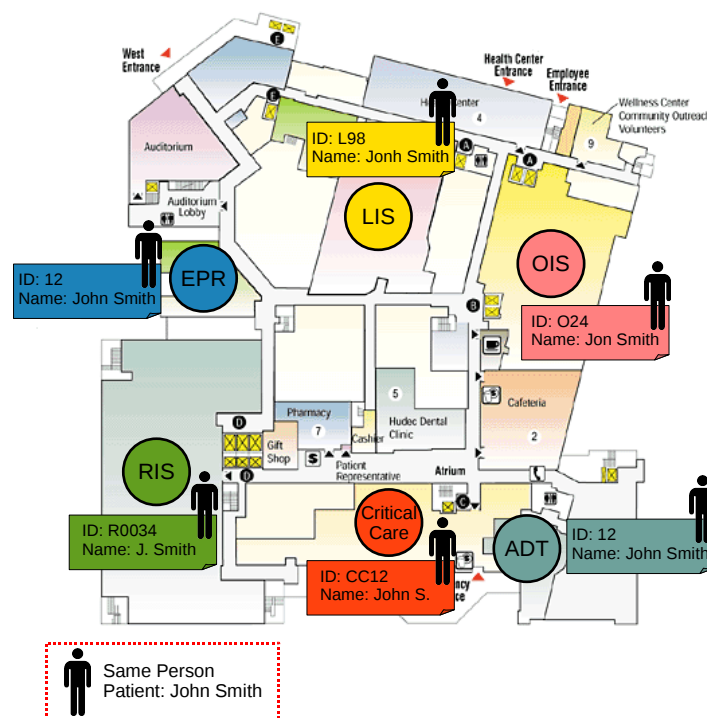


Figura 1.1: Falta de sincronismo e coerência entre os vários sistemas.

São estes problemas, principalmente a falta de sincronismo e coerência, que afectam

indirectamente os pacientes, pois originam uma possível deficiência no serviço prestado pelos profissionais de saúde. A duplicação, a inconsistência e a incoerência presente nas heterogêneas bases de dados de cada um destes sistemas de informação, é suficiente para que uma pessoa possua mais do que um registo na mesma base de dados, e vários registos em todo o estabelecimento (ver figura 1.1). Dado que cada um destes registos contém dados pessoais eventualmente diferentes, e informação médica espalhada, torna-se difícil obter um registo único, por pessoa, que possa correlacionar todos os seus dados.

Surge então a necessidade de indexar todos os registos dos pacientes, de todos os sistemas de informação, num só repositório capaz de fornecer uma referência única para cada pessoa. É com este objectivo que nasce o conceito de um Master Patient Index (MPI).

1.2.1 Master Patient Index

Em sistemas médicos, o MPI é um repositório que referencia todos os pacientes conhecidos por um departamento médico, clínica, hospital ou organização de saúde[4]. Este repositório de dados contém, apenas, identificação demográfica de todos os pacientes de forma unívoca, permitindo assim ter um sistema centralizado de identificadores de pacientes, capaz de distribuir essa informação a todos os sistemas associados.

O MPI possui em cada registo do seu repositório: dados demográficos; dados de primeiros socorros, que consistem em valores e observações essenciais para a vida da pessoa, tais como as alergias; e uma lista de identificadores por paciente. É essa lista, composta por um identificador e a referência ao sistema ou entidade que o atribuiu, que torna possível o sincronismo dos registos entre os sistemas associados (ver figura 1.2). Assim todos os estabelecimentos sabem onde podem obter informações sobre um paciente seu e ter garantia de que os seus dados estão tão actualizados quanto o registo mais recente e actualizado na rede do MPI.

As características que diferenciam este sistema de um normal repositório são:

- A sua capacidade de correlacionar dados. O MPI, através de algoritmos de correlação, consegue atribuir uma probabilidade de dois registos referenciam a mesma pessoa, aproximando-se da ideologia de cada pessoa ter um, e apenas um, registo no repositório.

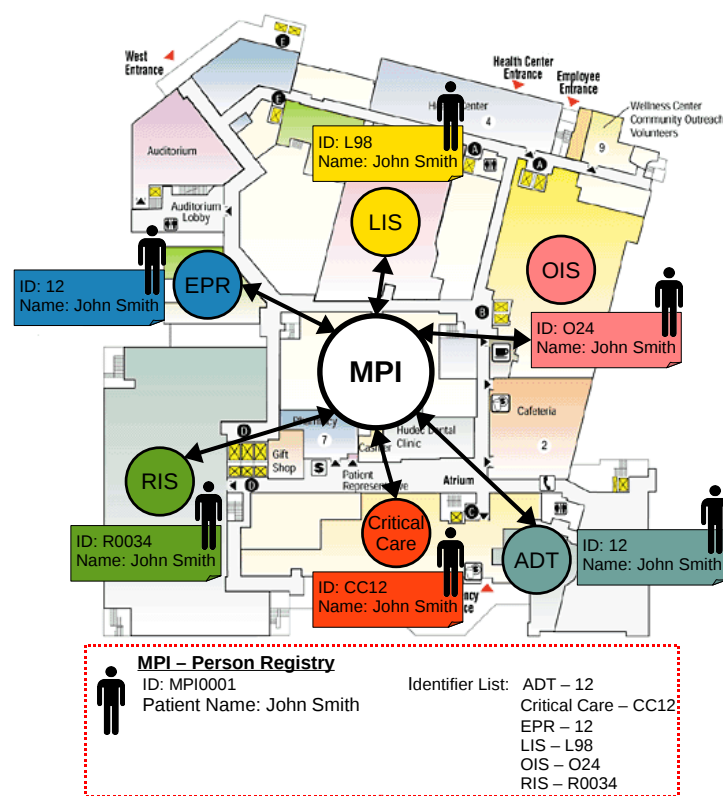


Figura 1.2: MPI implementado numa unidade médica.

- Executar todas as suas acções em tempo real e sem necessidade de intervenção humana.

Este sistema traz muitas vantagens para os profissionais de saúde. O sincronismo faz com que todos os sistemas interligados possuam a mesma referência para a mesma pessoa, o que possibilita que, através dos seus identificadores, se conheça onde é que a pessoa possui a sua informação clínica, e assim facilitar e melhorar o serviço prestado pelos prestadores de cuidados médicos. Estas vantagens fazem do MPI um sistema muito importante no futuro dos sistemas de informação médica.

É de ressaltar que o MPI não substitui os sistemas de informação locais dos estabelecimentos de saúde. É um repositório de dados demográficos usado para sincronismo, identificação e unicidade de pessoas, não contém informação médica.

1.2.2 Visão

Este projecto visa preencher uma lacuna há muito existente nos sistemas de informação médica. Conseguir com que todos os sistemas de um estabelecimento de saúde estejam sincronizados em relação aos seus pacientes, trará uma maior valia na avaliação clínica de uma pessoa, mesmo que seja o primeiro contacto do paciente com o profissional de saúde.

Um sistema MPI, para conseguir fornecer todas estas vantagens tem de implementar as seguintes funcionalidades base:

- capacidade de adicionar pessoas;
- capacidade de alterar os dados de pessoas existentes;
- capacidade de adicionar identificadores a uma pessoa existente;
- reconhecer igualdades nos registos;
- unir dois registos num só;
- responder a pesquisas colocadas ao repositório com uma lista de possíveis resultados;
- sincronizar os sistemas associados;
- responder a todas as acções em tempo real e sem auxílio humano.

Além de ambicionar marcar um passo importante na evolução dos sistemas de informação clínica, este projecto também visa o mercado. Quando tentamos lançar um produto no mercado, temos de ter em atenção os clientes, e, considerando o mercado actual, o MPI não pode deixar de lado algumas importantes características:

- ser autónomo, para que possa facilmente ser implementado em qualquer estabelecimento de saúde;
- conseguir adaptar-se aos sistemas já existentes, e não o contrário;
- usar todos os padrões definidos e aceites pela comunidade de forma a conseguir comunicar com todas as entidades;
- cumprir todas as suas funcionalidades base, pois nenhum sistema consegue vencer se estiver incompleto.

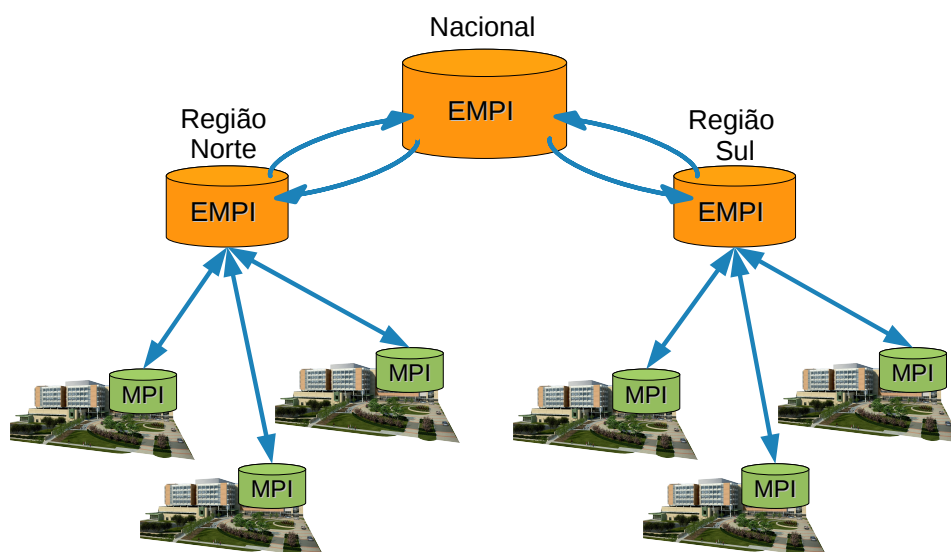


Figura 1.3: Exemplo de uma rede com vários Enterprise Master Patient Index

Um MPI é geralmente visto como um repositório local a um estabelecimento de saúde, mas também pode ser o responsável por sincronizar os sistemas de informação de vários estabelecimentos. Este conceito introduz o Enterprise Master Patient Index (EMPI), em que temos uma hierarquia de MPIs (ver figura 1.3). As características e funções de um EMPI em nada diferem das de um MPI, apenas se encontra num nível hierarquicamente superior e possui um repositório potencialmente de dimensão superior. O EMPI tem como objectivo final, alargar-se a um repositório nacional onde seria possível idealizar um "Bilhete de Identificação de Saúde" para cada pessoa, que seria válido em qualquer unidade de saúde do país como forma única de identificação médica.

1.3 Estrutura do Relatório

O presente relatório encontra-se dividido em 5 capítulos, a saber:

O primeiro capítulo - "Introdução" (página 13), apresenta a empresa onde decorreu o estágio; o problema a resolver; a motivação e visão para o projecto;

O segundo capítulo - "Estado da arte" (página 22), apresenta o estudo que levamos a cabo sobre o projecto, mostrando toda a informação que achamos relevante para o mesmo;

O terceiro capítulo - "Projecto Master Patient Index" (página 30), apresenta a ideologia

definida para o projecto;

O quarto capítulo - "Implementação" (página 44), apresenta a implementação dessa ideologia;

O quinto capítulo - "Conclusão" (página 63), apresenta uma análise crítica ao trabalho efectuado perante o projecto e algumas melhorias que poderão ser implementadas no futuro.

Capítulo 2

Estado da Arte

Neste capítulo vamos mostrar o estudo efectuado para compreender o que é um MPI. Estudo que recaiu sobre os padrões, ideologias, implementações, tecnologias e sistemas semelhantes relacionados com a área.

2.1 Health Level 7

O Health Level 7 (HL7) é uma organização que desenvolve e define padrões para facilitar a comunicação entre sistemas e aplicações ligadas à área da saúde [15]. Através de uma compilação de mensagens bem fundamentadas, proporciona uma estrutura que abrange quase toda a informação necessária nos vários papéis das funções logísticas, administrativas e clínicas de um prestador de saúde.

Antes do HL7, todos os sistemas e aplicações eram desenhados conforme as necessidades encontradas no local de instalação. Esta situação trazia problemas de custos, pois, para que uma aplicação pudesse ser reutilizada, eram necessárias grandes alterações no seu núcleo. Foi para combater esta tendência que alguns membros da comunidade médica formaram esta organização.

2.1.1 HL7 V2

O HL7 V2 foi a primeira versão deste padrão. Desenvolvida por especialistas em aplicações clínicas como forma de ultrapassar o problema da comunicação entre as mesmas, esta versão focou-se na realidade com que os seus criadores se deparavam

diariamente. Como objectivo, pretendiam abranger 80% de toda a estrutura usada pelos estabelecimentos médicos e criar um padrão que toda a comunidade abraçasse, por ser simples de usar e eficaz nas suas funções [21].

A estrutura das mensagens desta versão baseia-se numa simples *string*. As mensagens são constituídas por vários segmentos, geralmente separados por linhas e cada campo destes segmentos é separado pelo caracter ”|”. Todos os segmentos, bem como a sua constituição, estão bem definidos e documentados.

Pode-se afirmar que as primeiras publicações do V2 (V2.1 e V2.2) eram vagas e não documentadas, quando comparadas com publicações mais recentes. Mas é com estas publicações que os primeiros passos foram dados e em 1998, já com a publicação da versão V2.3, o V2 assume uma posição importante no mercado das aplicações clínicas. Mantendo sempre compatibilidade com as suas publicações antecessoras, o V2 continuou a evoluir e encontra-se actualmente na publicação V2.6, com a publicação V2.7 a ser desenvolvida, e a V2.8 a ser pensada e discutida. Nunca foi o objectivo desta versão abranger a 100% toda a infraestrutura clínica e médica e, por isso, existia algum espaço para evoluir. Por esse motivo, um novo grupo de pessoas começou a olhar para o problema e decidiu optar por uma abordagem diferente, nascendo o HL7 V3.

2.1.2 HL7 V3

O HL7 V3, desenvolvido por especialistas em informática médica, tem como objectivo melhorar o conseguido pelo V2 e eliminar as lacunas por este deixadas. Para tal, este grupo pretende criar uma versão com as seguintes características:

Internacionalização - o V3 deve ser global a fim de ser usado em todo o mundo, mas suportar as necessidades locais a cada país.

Modelo de dados consistente - o modelo de dados usado tem de ser bem definido para que seja consistente em todas as aplicações.

Padrão preciso - O V3 pretende criar um padrão que contenha toda a informação necessária, tornando-o menos vago, logo menos flexível.

Novo padrão - O V3 nunca vai ser compatível com o V2. Para manter rectro compatibilidade, o V3 iria ficar com algumas lacunas do V2, e os seus criadores

acreditam que para o V3 poder mudar radicalmente, precisa de ser criado sem se apoiar em nenhum padrão existente.

No final de 2005 a comunidade HL7 publicou o V3, provando que a iniciativa do grupo produziu resultados perante os seus objectivos. Ainda longe de completo, o V3 já abrangia 90% da estrutura médica. Com um modelo de dados explícito, definições claras e mais situações reais, o V3 promete facilitar ainda mais o trabalho a quem desenvolve aplicações clínicas e médicas. Esta versão ainda se encontra em evolução e desenvolvimento mas já apresenta excelentes condições de ser usada em grande escala. Apesar de ser já muito madura, são poucas as aplicações que a usam, em grande parte, devido à complexidade que representa migrar uma aplicação de V2 para V3.

2.2 Integrating the Healthcare Enterprise

O Integrating the Healthcare Enterprise (IHE) é uma iniciativa de vários profissionais de saúde e da própria indústria, com o intuito de melhorar a forma como os sistemas de informação médica partilham informações [17], promovendo o uso de padrões pre-definidos, como HL7. Sistemas desenvolvidos de acordo com o IHE têm a vantagem de comunicarem eficientemente entre eles, serem simples de implementar e ajudarem os estabelecimentos de saúde a usar a sua informação de forma mais eficaz.

Deste grupo surgiram dois perfis de integração como possível solução para implementar um MPI: o Patient Identifier Cross-reference e o Patient Demographics Query.

2.2.1 Patient Identifier Cross-reference

O Patient Identifier Cross-reference Integration Profile (PIX) tem o objectivo de servir, essencialmente, organizações que prestem serviços médicos com uma grande quantidade de instalações. Este perfil suporta a correlação entre os identificadores dos pacientes de vários domínios, garantindo sincronismo e um fácil acesso a todos os pacientes.

O PIX define três actores base: a fonte; o consumidor; e o administrador, respectivamente representados na figura 2.1 como Patient Identify Source, Patient Identifier Cross-reference Consumer e Patient Identifier Cross-reference Manager.

A fonte é a responsável por alimentar o administrador com todos os seus registos

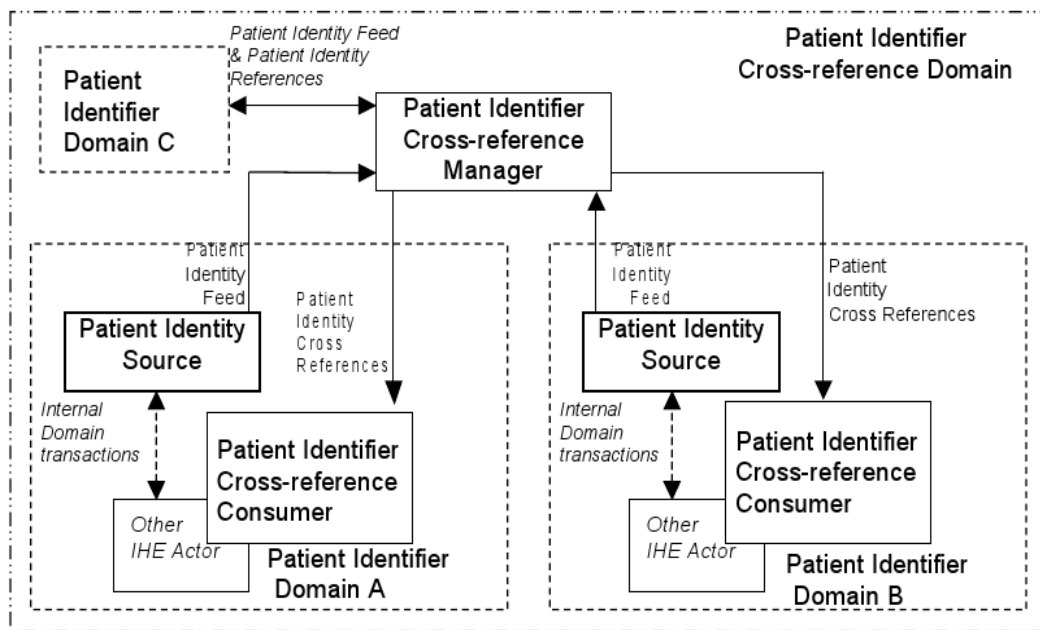


Figura 2.1: Diagrama de fluxo do PIX (adaptado do IHE ITI Technical Framework)

e informar as alterações efectuadas nos mesmos. O consumidor pode questionar o administrador por pacientes que não conheça e deve receber notificações caso os dados de um paciente seu sejam alterados. O administrador é a ponte entre as várias fontes e os vários consumidores da rede. Este recebe os pedidos dos consumidores e procura nas suas fontes respostas aos pedidos, também propaga aos seus consumidores as alterações efectuadas nas fontes.

Na figura 2.1 podemos observar que, na generalidade, cada domínio é constituído por uma fonte e um consumidor e que o administrador sincroniza e correlaciona os vários domínios. Na visão do PIX um MPI enquadra-se nas funções do administrador.

2.2.2 Patient Demographics Query

O Patient Demographics Query Integration Profile (PDQ) define como se deve proceder a uma pesquisa de pacientes, através de um conjunto de dados demográficos.

O PDQ define dois actores base: a fonte e o consumidor, respectivamente representados na figura 2.2 como Patient Demographics Supplier e Patient Demographics Consumer.

A fonte recebe pesquisas de um consumidor e, dependendo dos valores de filtragem presentes, devolve um conjunto de registos como resposta. O consumidor cria uma

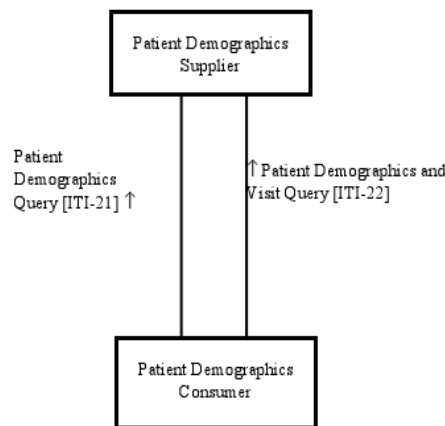


Figura 2.2: Diagrama de fluxo do PDQ (adaptado do IHE ITI Technical Framework)

pesquisa com um, ou vários, valores de filtragem e envia à fonte, ficando à espera de uma resposta.

Este perfil não contém de forma implícita a noção de um MPI na sua implementação, mas ajuda a compreender como é que este se deve comportar quando interage com uma fonte. Na visão do IHE, unindo ao fluxo do PIX a complexidade das pesquisas do PDQ obtemos um MPI mais capaz [9] [10].

2.3 Sistemas Semelhantes

No mercado existem algumas empresas que afirmam possuir uma solução MPI, mas não existem muitos estabelecimentos de saúde onde operem. Nesta secção vamos mostrar, de todos os sistemas encontrados, os que nos forneceram maior quantidade de informação e esclarecimento sobre a implementação e funções de um MPI.

2.3.1 Tayside Patient Index

Em 1978, na antiga região de Tayside, Escócia, a decisão de reunir todos os registos médicos da população local num único sistema de informação, fez desta região, uma das primeiras entidades a desenvolver sobre o conceito de um MPI [2].

Através de um processo humano e moroso de recolha e correlação dos registos médicos de todos os seus estabelecimentos de saúde, conseguiram obter, com uma certeza elevada, um registo por cada habitante. Cada registo continha os dados demográficos da

pessoa e uma lista de identificadores, que consistia nos estabelecimentos que possuíam dados sobre a mesma e a sua identificação nesses estabelecimentos. Posteriormente esses dados foram introduzidos numa base de dados informática, para que se pudesse consultar e actualizar, sempre que necessário.

A Tayside MPI permitia aos seus utilizadores, através de um terminal remoto: introduzir, actualizar e eliminar um registo; unir dois registos e corrigir uma união; actualizar a lista de identificadores e pesquisar por pacientes através de vários dados.

Este sistema não era uma base de dados médicos, mas sim de dados demográficos. A Tayside MPI não substituíu a base de dados dos estabelecimentos de saúde, apenas fornecia os dados demográficos o mais actualizados possível e mostrava onde se podia encontrar mais informação sobre o paciente.

Na altura, as vantagens apontadas para a criação deste sistema foram:

económicas: os estabelecimentos de saúde não precisavam de alterar os seus sistemas de informação locais para usufruir deste sistema.

integridade: usando a mesma fonte, todos poderiam ter uma base de dados sem redundância e incoerências.

sincronização: *triggers* asseguravam que todos os sistemas ficavam conhecedores das alterações na base de dados.

localização de registos: a lista de identificadores tornou simples o processo de um estabelecimento saber onde podia obter informação médica sobre um paciente.

informação populacional: se o MPI for abrangente o suficiente, pode fornecer informação populacional útil.

O facto de estas vantagens, hoje em dia, ainda serem importantes, fez com que o sistema tenha prevalecido e evoluído até ao nosso tempo. Agora, como National Health Service - MPI (NHS), estende-se por todo o Reino Unido, sendo mantido pelo Health Informatics Center of University of Dundee, pelo Tayside Health Board e pelo próprio NHS.

2.3.2 VistA / Austin - US Department of Veterans Affairs

O US Department of Veterans Affairs (VA) é uma instituição governamental Norte Americana que prima pela excelência no serviço médico. Possui uma cadeia própria de

estabelecimentos de saúde, onde são fornecidos todos os serviços de saúde aos soldados, ex-soldados, veteranos Norte Americanos e suas famílias. Para poder receber e tratar os seus utentes com a maior das eficiências, criou uma infra-estrutura MPI que liga todos os seus estabelecimentos de saúde. Infra-estrutura essa, constituída por dois sistemas, um Master Patient Index / Patient Demographic (MPI/PD) e um Enterprise Master Patient Index (EMPI).

O MPI/PD, intitulado VistA, pode-se encontrar em qualquer estabelecimento dos VA, como raiz do sistema de informação [7]. Todos os outros sistemas acedem ao VistA para obter qualquer dado de um paciente. Podemos então afirmar que o VistA é o responsável por manter um repositório sem duplicações e incoerências para os pacientes que já frequentaram o seu estabelecimento.

O EMPI, intitulado Austin, é responsável por correlacionar todos os sistemas VistA. Assim um utente do grupo VA pode dirigir-se a qualquer estabelecimento VA, que o VistA local pode sempre obter os seus dados através do Austin.

Quando uma pessoa se desloca pela primeira vez ao prestador e é apresentada ao VistA, para obter um identificador, os seus dados demográficos são introduzidos e enviados ao Austin. Através do nome (primeiro e último), número da segurança social, data de nascimento e género, é feita uma pesquisa para verificar se existe algum registo contido na rede VA que possa já referenciar o paciente. Se for encontrado um registo, que o sistema possa afirmar ser pertencente ao paciente, o Austin responde com os dados que possui, adiciona o estabelecimento à lista de identificadores e comunica a todos os estabelecimentos interessados a alteração. Se encontrar vários registos, que possam referenciar o mesmo paciente, o Austin cria uma notificação e envia-a para a administração, para futura revisão humana, criando de seguida um novo registo para o paciente. Em caso de não serem encontrados registos que possam pertencer ao paciente, um novo registo é simplesmente criado.

Os VA possuem muitos contratos com estabelecimentos de saúde que não lhes pertencem, e por isso é importante que sistemas, não proprietários, possam comunicar com o Austin de forma a obter informação sobre os utentes do grupo VA. Para tal ser possível, o Austin está preparado para aceitar mensagens HL7 [6]. As mensagens usadas e aceites pelo Austin precisam de estar na versão 2.x e segundo o padrão definido pelo grupo do HL7.

Outro ponto interessante nos sistemas VA, são os *Master Files* [22] usados por eles. Todos os sistemas possuem tabelas sincronizadas, onde estão os valores predefinidos, pelo grupo para alguns campos, como por exemplo, o género da pessoa. Estas tabelas

servem para que todos os sistemas saibam ao que se refere o valor de um campo, independentemente de onde veio a informação, e, também, para garantir que todos os sistemas usam informação válida nos seus registos.

2.3.3 Composite Application Platform Suite - SUN

A Sun Microsystems afirma ter um MPI baseado no SunTMComposite Application Platform Suite (CAPS) [12] e concordante com o Integrating the Healthcare Enterprise (IHE).

A solução da Sun, o Sun SeeBeyond eView/eIndex, é uma base de dados composta por dados demográficos de pacientes, com capacidade de correlacionar os seus registos em tempo real através de uma lista de identificadores. É das poucas soluções a afirmar ter uma aplicação cliente para gestão humana das acções efectuadas pelo MPI e é implementada usando a *framework* IHE PIX e IHE PDQ [11], já abordadas neste capítulo. Para permitir uma comunicação global com todas os estabelecimentos de saúde, esta solução usa mensagens HL7, versões 2.3, 2.5 e 3.

A SUN é das poucas organizações que diz suportar um MPI a comunicar com a versão 3.

Capítulo 3

Projecto Master Patient Index

Neste capítulo vamos mostrar como idealizamos o nosso projecto. Após obter todos os requisitos, necessidades, e os conhecimentos que envolvem um MPI, é preciso projectar o objectivo final. Para tal, recorreremos a casos de uso; diagramas de sequência; e ao desenho de uma arquitectura geral; que iremos explicar de seguida.

3.1 Casos de Uso

Para compreendermos quais as aplicações, sistemas ou pessoas que vão interagir com o MPI, que género de acções vão poder executar e o que esperam receber do MPI, criamos um diagrama que demonstra os casos de uso de cada actor (apêndice B). Nestes casos de uso detectamos quatro actores: o consumidor; o administrador; o Enterprise Master Patient Index; e o Master File Manager (MFM). De seguida vamos explicar cada um deles e demonstrar quais as acções que cada um pode efectuar.

3.1.1 Consumidor e funcionalidades associadas

O consumidor é uma aplicação que possui capacidades de administração sobre um repositório de pessoas e está interessado em o sincronizar com mais aplicações (ver figura 3.1).

Este actor pode pedir ao MPI para: criar um novo registo; alterar os dados de um registo já existente; ligar dois, ou mais, registos; desfazer ligações entre registos; fundir dois registos num só; e desfazer a fusão de dois registos. O consumidor também pode

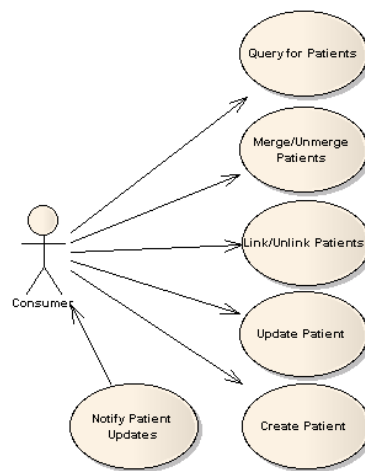


Figura 3.1: Interação do consumidor com o MPI

questionar o MPI, com base num conjunto de dados demográficos, sobre um paciente. Nesta interação, a única acção que é iniciada pelo MPI é a notificação das alterações que ocorrem nos seus registos. Um consumidor pode ser subscritor de dois tipos de notificações: notificação sobre todo o repositório, onde qualquer que seja a modificação sentida no repositório, o consumidor é notificado; ou notificação de domínio, onde apenas modificações a registos que também pertencem ao seu domínio são notificadas.

3.1.2 Administrador e funcionalidades associadas

O administrador é uma pessoa que, através de uma aplicação própria do MPI, consegue ter privilégios de administração sobre o sistema (ver figura 3.2).

Este actor tem acesso ao registo de todas as acções efectuadas pelos consumidores e correspondentes soluções prestadas pelo MPI, para que possa detectar falhas ou más decisões. Com o objectivo de corrigir os eventuais erros, o administrador pode executar todas as acções de um consumidor além de lhe ser permitido eliminar registos.

As restantes acções possíveis a um administrador são: gerir uma lista de consumidores, controlando os seus acessos e privilégios; gerir os catálogos internos do MPI, que passa por aplicar as notificações do MFM; poder configurar os algoritmos de correlação do sistema, por forma a melhorar a sua eficiência; e poder ver potenciais igualdades entre um registo e todo o repositório, para manualmente detectar duplicação de registos. Para eliminar problemas durante a conexão, caso um novo consumidor queira inserir todos os seus pacientes no repositório, o administrador pode optar por inserção através

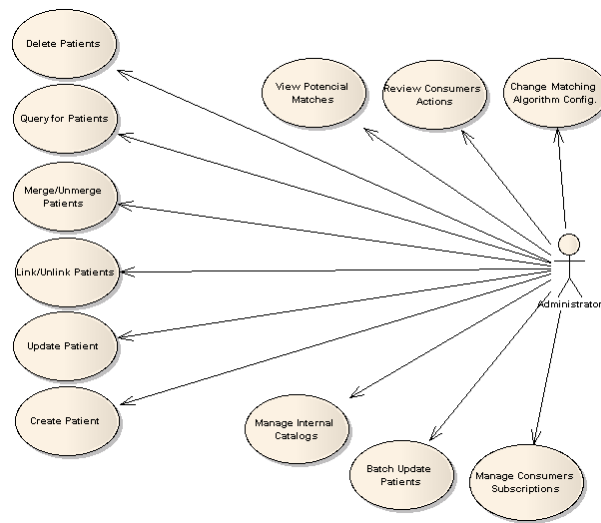


Figura 3.2: Interação do administrador com o MPI

de um *batch file*.

3.1.3 Enterprise Master Patient Index e funcionalidades associadas

O EMPI é um sistema que garante o sincronismo entre vários MPIs. Este actor não é mais do que um sistema MPI hierarquicamente superior, logo, o MPI é um consumidor do EMPI (ver figura 3.3).

Mesmo assim, as acções entre estas duas entidades são ligeiramente diferentes. O MPI pode pesquisar no EMPI por um registo que desconheça, tal como um consumidor, mas as restantes acções são efectuadas de forma diferente. O MPI, após sofrer qualquer modificação no seu repositório, notifica o EMPI de forma a garantir que o EMPI contém, sempre, todos os registos do seu repositório. O EMPI, já só notifica o MPI caso exista alguma modificação em registos que pertencem ao seu domínio. A única acção que um EMPI pode fazer que um consumidor não pode é eliminar um registo.

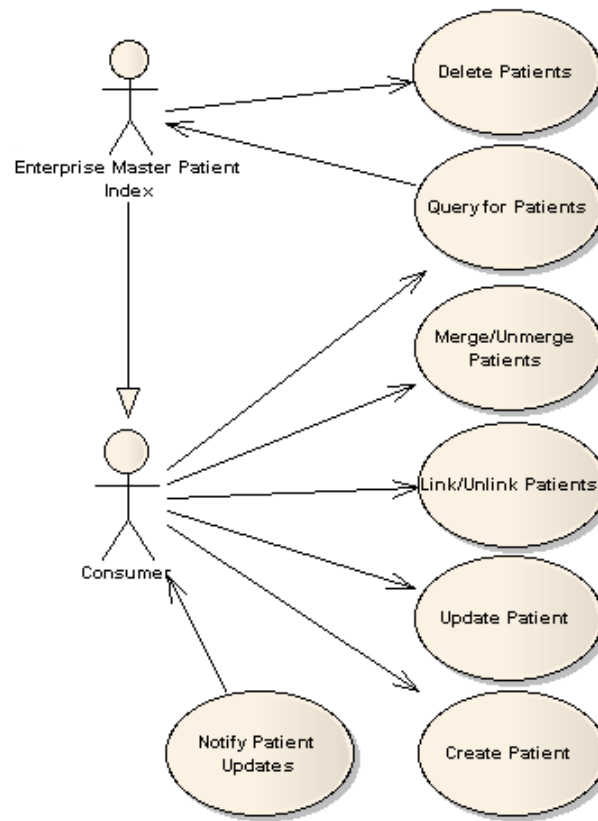


Figura 3.3: Interacção do EMPI com o MPI

3.1.4 Master File Manager e funcionalidades associadas

O MFM é um sistema que garante o sincronismo dos catálogos do MPI com os restantes sistemas e aplicações. Nesta interacção, o MPI é visto como cliente do MFM (ver figura 3.4).

A este actor o MPI pode: pedir a modificação de itens de um catálogo específico; e pedir a subscrição a catálogos do MFM. Qualquer modificação nos catálogos, a que o MPI está subscrito, é notificada pelo MFM.

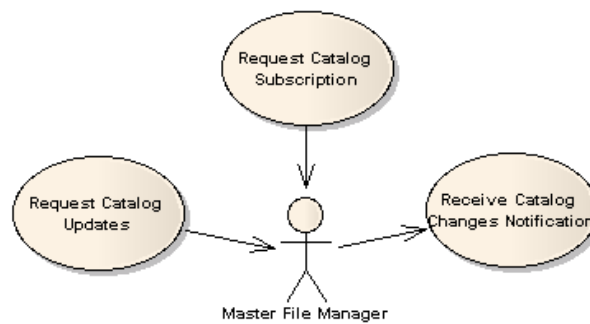


Figura 3.4: Interação do MFM com o MPI

3.2 Cenários

Para melhor compreender o funcionamento de um MPI, criamos alguns cenários que ilustram os casos base entre o MPI e os seus consumidores e que corresponde à interação mais importante do sistema.

3.2.1 Pessoa desconhecida em todos os sistemas

Uma pessoa, chamada Isabel, entra pela primeira vez no estabelecimento de saúde H.C. Porto. Na sua admissão, fornece os seus dados demográficos ao administrativo, que os introduz na aplicação, utilizando um *Patient Administration System* (PAS), onde observa que a pessoa é desconhecida ao sistema. De seguida, com os mesmos dados, anteriormente fornecidos, o administrativo usa a aplicação para questionar o MPI sobre a pessoa em causa. O último passo pode ser evitado se a aplicação estiver preparada para fazer a pesquisa no MPI, em caso de não encontrar resultados localmente.

O MPI internamente determina uma lista de candidatos e retorna-a. Neste caso, como o MPI não conhece a pessoa, pode retornar dois tipos de listas: vazia; ou com candidatos que tenham uma probabilidade alta de ser a pessoa procurada.

Ao receber a lista, o administrativo observa que o MPI também não conhece a pessoa e inicia a criação de um novo registo. Após a criação do registo e atribuição de um identificador local ao mesmo, a aplicação informa o MPI da adição feita localmente e continua com o seu trabalho, que pode ser, ainda, sobre este registo.

Ao receber a informação que um novo registo foi adicionado no estabelecimento de saúde, o MPI também cria um novo registo localmente, com base no recebido, e atribui-

lhe o seu identificador local. Após a criação ter sido efectuada, o MPI notifica todos os consumidores interessados no novo registo. Existem dois tipos de consumidores: do repositório, que são notificados de todas as modificações que acontecem em qualquer registo do repositório; e de domínio, que são notificados caso algum dos registos pertencentes ao seu domínio sofra alterações.

Mais tarde, durante a admissão, o registo pode sofrer modificações. Se for o caso, a aplicação deve informar o MPI para que este possa modificar o seu registo e notificar os consumidores interessados. Esta operação pode acontecer diversas vezes durante o período em que o paciente está admitido.

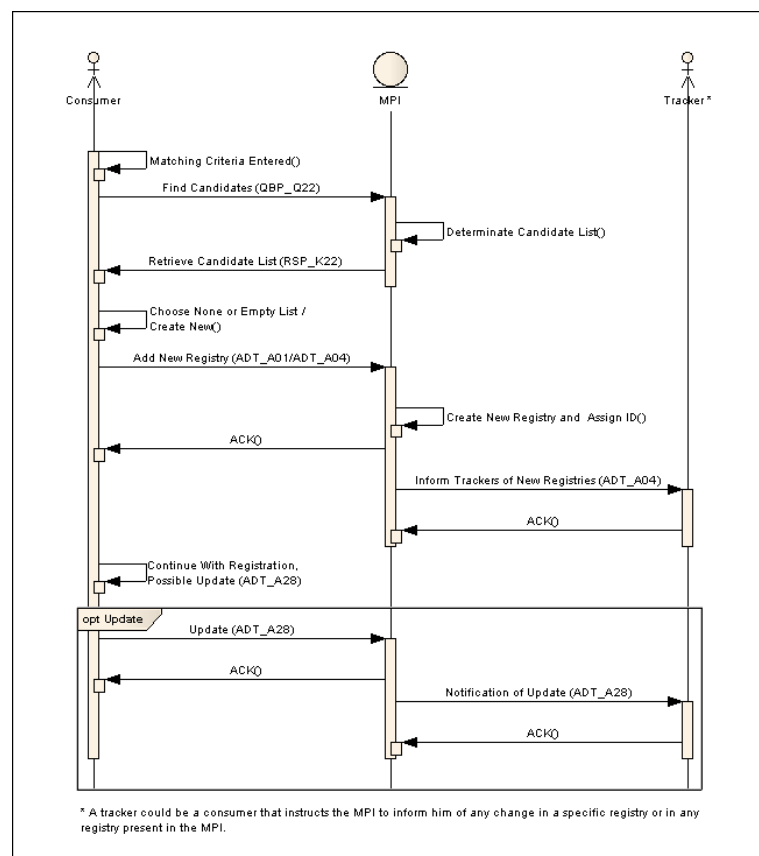


Figura 3.5: Diagrama de Sequência para o caso de uma pessoa ser desconhecida a todos os sistemas (Adaptado da ideologia do IHE e do HL7)

O diagrama representado na figura 3.5 ilustra o cenário apresentado, com o acréscimo de mostrar quais as mensagens HL7 V2 que devem ser trocadas pelos vários actores da peça. Este diagrama demonstra o protocolo de comunicação esperado para estes casos, e segue as ideologias apresentadas tanto pelo IHE como pelo HL7 [9] [8].

3.2.2 Pessoa conhecida apenas pelo MPI

A mesma pessoa, a Isabel, entra agora no estabelecimento de saúde H.C. Gaia. Aquando da sua admissão, fornece os seus dados demográficos, que o administrativo insere na aplicação, utilizando um PAS, e o sistema informa que não conhece a pessoa. Da mesma forma que foi anteriormente explicado, uma pesquisa é feita no MPI.

O MPI internamente determina uma lista de candidatos e retorna-a. Desta vez o MPI já conhece a pessoa, e por isso o seu registo está contido na lista enviada, que pode conter mais registos.

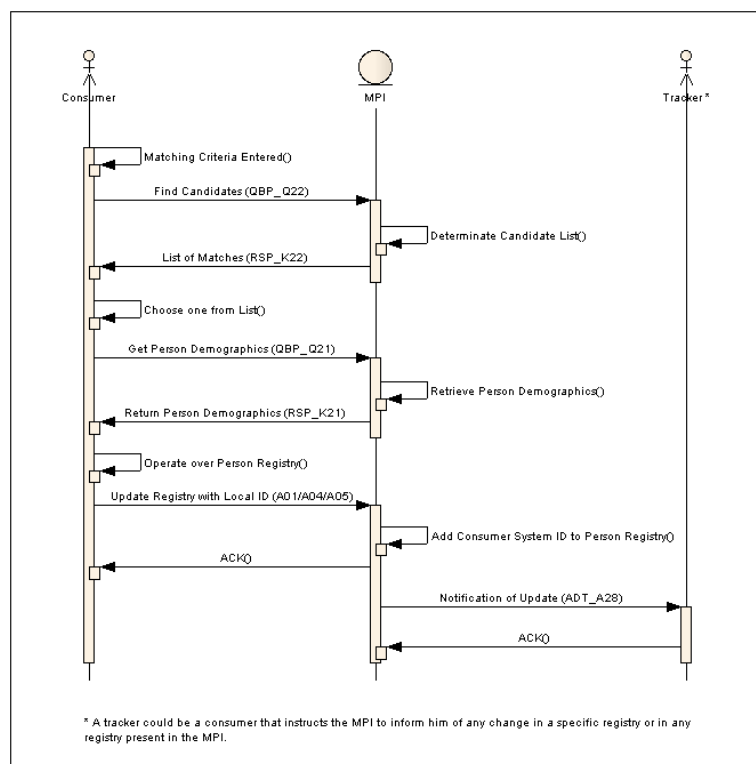


Figura 3.6: Diagrama de Sequência para o caso de uma pessoa ser conhecida pelo MPI e não pelo estabelecimento (Adaptado da ideologia do IHE e do HL7)

Ao receber a lista, o administrativo escolhe a pessoa correcta e volta a pedir ao MPI informação sobre esta pessoa, mas neste caso pede especificamente, todos os dados demográficos de uma determinada pessoa. Este passo pode ser desnecessário se o MPI retornar com a lista de candidatos já todos os dados que possui por pessoa, mas como a lista pode conter vários registos é aconselhável apenas retornar um pequeno conjunto de dados capazes de identificar a pessoa. Após a recepção do registo completo, o administrativo pode continuar com a admissão. É normal neste casos, o

estabelecimento de saúde guardar o registo para si, logo vai criar um identificador local para atribuir ao registo. Este acto deve ser seguido de uma notificação a informar o MPI deste novo identificador. Assim o MPI pode adicionar o novo identificador ao seu registo e notificar todos os consumidores interessados, que neste caso seria, pelo menos, o H.C. Porto.

O diagrama (ver figura 3.6) ilustra o cenário apresentado, com o acréscimo de mostrar quais as mensagens HL7 V2 que devem ser trocadas pelos vários actores da peça. Este diagrama demonstra o protocolo de comunicação esperado para estes casos, e segue as ideologias apresentadas tanto pelo IHE como pelo HL7 [9] [8].

3.2.3 Existir um EMPI na rede

Para o primeiro cenário apresentado, em que a pessoa não é conhecida em nenhum dos sistemas, tendo o MPI acesso a um EMPI, deveria:

1. caso a sua lista de candidatos não possuisse um candidato certo de ser a pessoa procurada, fazer a mesma pesquisa no EMPI e retornar os resultados obtidos;
2. havendo dessa lista uma escolha por parte do consumidor, o MPI seria informado do novo identificador. Nesse momento o MPI cria um novo registo para si e informa o EMPI dos dois novos identificadores, o do MPI e o do consumidor, para aquele registo;
3. se da lista não for escolhido nenhum candidato, o consumidor cria um registo novo e informa o MPI sobre a criação. Por sua vez o MPI faz o mesmo e informa o EMPI do novo registo, mantendo assim o sincronismo.

Para o segundo cenário apresentado, só faz sentido pensar no EMPI aquando da criação de novos identificadores e de qualquer alteração nos registos. Sempre que o MPI modifica algo nos seus registos, deve informar o EMPI, apesar de não ser o caso exacto, o EMPI pode ser visto como um consumidor do repositório, pois é informado de todas as modificações que ocorrem no repositório do MPI.

Após todo o estudo feito à volta do MPI, sabíamos que apenas seria necessário guardar no repositório, dados demográficos e uma lista de identificadores por cada pessoa. Para tal baseamo-nos no modelo de dados que o grupo do HL7 V3 propõe (ver figura 3.7).

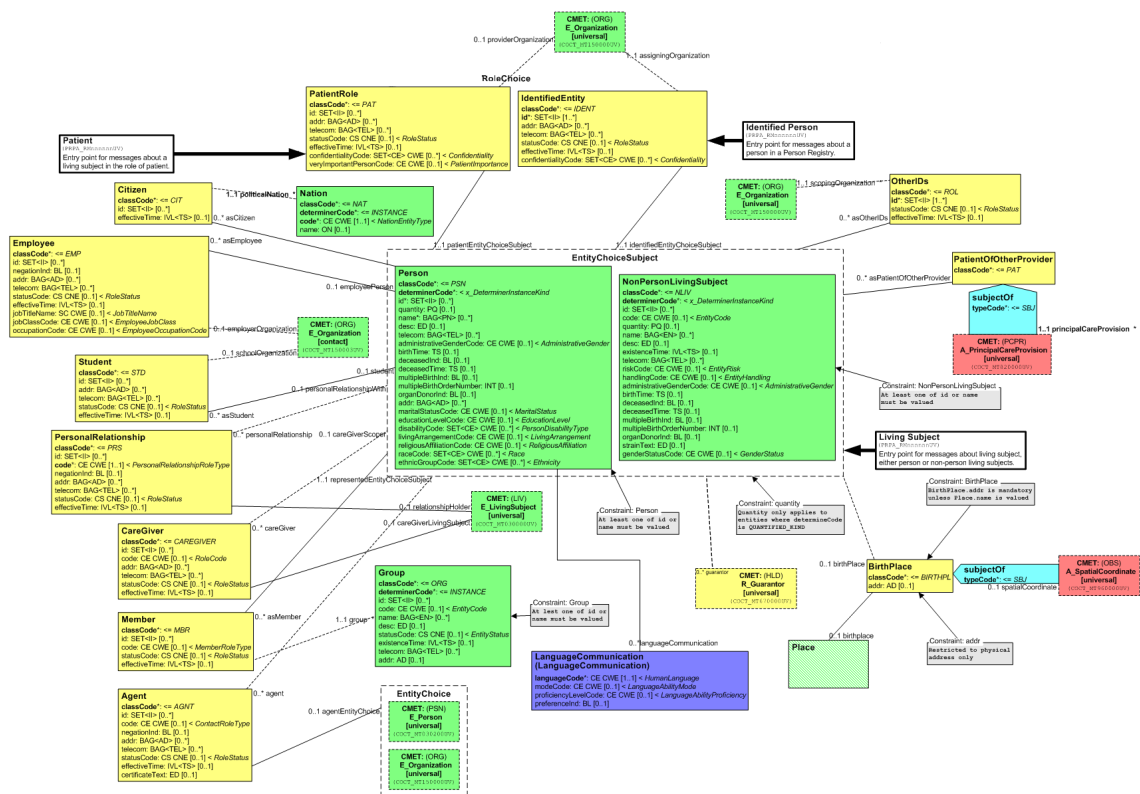


Figura 3.7: Adaptado do HL7 RIM Patient Registries Patient Administration (PRPA DM000000UV)

Este modelo define quais os atributos que devem ser guardados aquando da admissão de uma pessoa num estabelecimento de saúde, por isso conseguimos abranger todos os dados que interessam e que podem identificar uma pessoa. As mensagens criadas pelo grupo do HL7 V3, foram desenhadas com base neste modelo, o que já não aconteceu com as HL7 V2. O mesmo acontece com o tipo de dados, para cada campo, que este modelo propõe. Cada tipo de dados é definido pelo grupo do HL7 V3 e nem todos são compatíveis com as versões V2. Por haver tanta diferença nas duas versões, decidimos criar um repositório que se baseia neste modelo mas sofrendo algumas alterações, quer por motivos de adaptação às duas versões; quer por motivos de em Portugal alguns valores não serem necessários ou serem ilegais segundo a Comissão Nacional de Protecção de Dados (CNPd); ou mesmo porque concluímos que uma complexidade

tão elevada no modelo de dados não era necessária.

Analisando quais os valores necessários para identificar uma pessoa (ver Apêndice C), chegamos a um modelo relacional que nos pareceu suficiente. Na figura 3.8 podemos observar um excerto desse modelo e verificar que é um sub-conjunto do RIM PRPA. Neste excerto excluimos as ligações aos Master Files, pois apenas são usadas para manter sincronismo e coerência entre os valores dos campos.

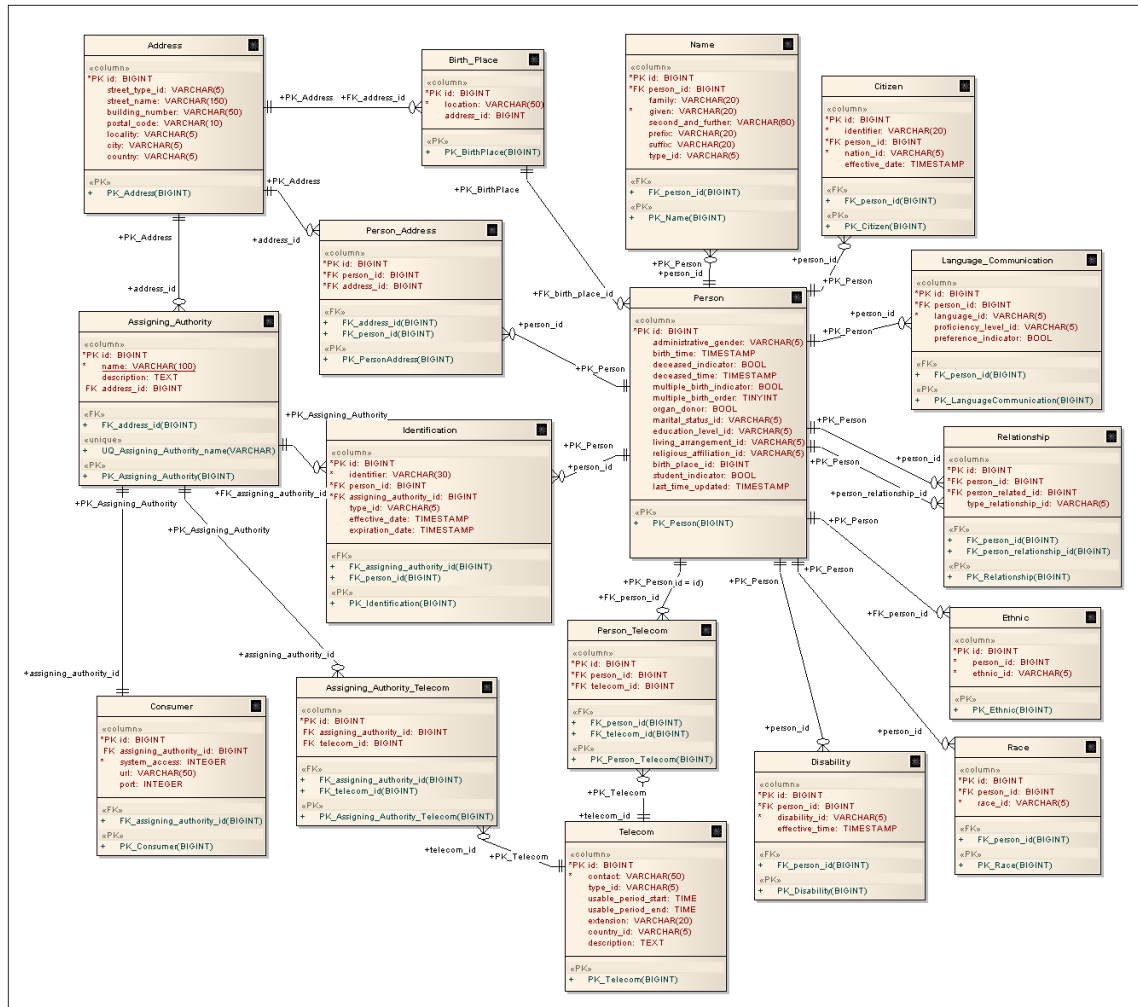


Figura 3.8: Excerto do Modelo da Base de Dados

Durante o estudo efectuado sobre os dados que deviam ser guardados no MPI, verificamos que algumas entidades referiam os dados de primeiros socorros, tais como as alergias e as últimas análises, como valores muito importantes a guardar neste repositório [14]. Também consideramos esses valores importantes, mas como o RIM não define de forma óbvia, onde guardar estes valores nas mensagens da versão V3 e na versão V2 também não existir uma forma simples de o fazer, decidimos deixar estes

valores como trabalho futuro a implementar.

3.4 Arquitectura Geral

A arquitectura geral do projecto foi desenhada tendo em conta o conjunto de informação que todo o estudo que demonstramos anteriormente nos trouxe. Tendo o conhecimento de quais as entidades que precisam de comunicar com o MPI e que operações este deve efectuar internamente, deu-nos uma melhor visão do objectivo final deste projecto. A arquitectura ilustrada na figura 3.9 existe com esse intuito, de definir melhor quais os nossos objectivos e o caminho a seguir ao longo do projecto. Nesta secção vamos inicialmente descrever sucintamente cada bloco da arquitectura, por nós definida, e depois iremos aprofundá-los ao pormenor.

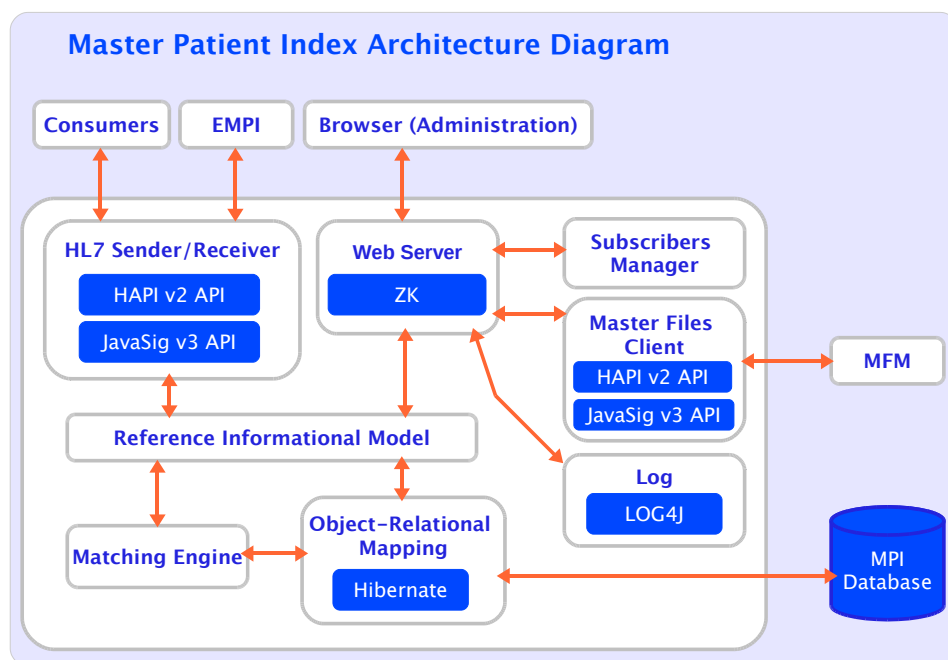


Figura 3.9: Arquitectura de alto nível

3.4.1 HL7 Sender/Receiver

A principal forma de comunicação deste sistema com sistemas externos é através de mensagens HL7, por isso temos um bloco dedicado a enviar e receber estas mensagens. Este bloco tem com funções receber mensagens nas versões HL7 V2 e HL7 V3; con-

seguir obter a informação nelas contida; proporcionar a criação de mensagens, segundo o padrão; e enviá-las aos consumidores.

3.4.2 Reference Informational Model

Para conseguirmos uma melhor abstracção de tudo o que rodeia o MPI, decidimos adoptar por um modelo de dados interno sempre fixo. Assim, qualquer bloco sabe como se moldar ao MPI e nunca precisamos de alterar o núcleo da aplicação. Mesmo que uma nova tecnologia ou protocolo precise de ser implementado, apenas um novo bloco teria de ser criado tendo por base este modelo de dados, que, como já explicado na secção 3.3 (página 38) tenta ser semelhante ao RIM do HL7 V3.

3.4.3 Matching Engine

Para existir a possibilidade de correlacionar os dados do repositório do MPI com os valores que este recebe, é necessário o uso de algoritmos e técnicas de correlacionamento de dados. Este bloco tem como suas funções exactamente esse papel. Usando variadas formas de identificar igualdades entre um conjunto de valores, este bloco deve proporcionar ao MPI uma das suas características mais importantes, permitindo unicidade nos registos do repositório e fornecendo uma pesquisa mais abrangente, contudo precisa.

3.4.4 Object-Relational Mapping

Este bloco advém apenas da necessidade de este projecto ter de funcionar perante uma grande variedade de tipos de bases de dados. Com este bloco temos uma abstracção em relação ao tipo de base de dados que usamos, o que nos proporciona uma maior flexibilidade no tratamento de dados, inserções e actualizações. Havendo um tipo de base de dados fixa para o projecto, tal como SQL Server, MySQL ou ORACLE, este bloco deixava de ser necessário.

3.4.5 Web Server

Como já foi anteriormente mencionado, o protocolo HL7 é a base de comunicação com o MPI, excepto para o administrador do sistema. Como o administrador precisa de

aceder a mais operações que os restantes intervenientes, e acede na sua maioria, de forma local ao MPI, decidimos criar um acesso mais directo, criando um *web server* para tal. Este bloco vai proporcionar o único acesso gráfico ao sistema, contendo a quantidade de ferramentas de administração necessárias à operabilidade do mesmo.

3.4.6 Subscribers Manager

Este bloco deve proporcionar uma gestão sobre os consumidores do nosso sistema, tal como, o tipo de acessos e privilégios que possuem. Este bloco, que é gerido apenas pelo administrador, permite um acesso de leitura ao MPI para que este possa saber quem pode aceder ao repositório e como notificar os consumidores. Internamente o MPI não modifica a informação contida neste bloco, apenas o administrador tem esse poder.

3.4.7 Master Files Client

Os catálogos que são mantidos pelo MPI devem poder estar sincronizados com os restantes sistemas que o rodeiam. Para tal o MPI possui um cliente do *Master Files Manager* que recebe todas as actualizações que este lhe envia. Esta comunicação também é feita através do protocolo HL7. As alterações que o MFM pede para serem efectuadas devem ser implementadas de imediato, mas o administrador pode reter a alteração até confirmação humana.

3.4.8 Log

Este bloco guarda todas as acções que o MPI executa, para que o administrador possa rever e corrigir se assim for necessário. Este bloco é muito importante para o sistema. Sem ele torna-se difícil descobrir falhas no sistema; falhas nos protocolos de comunicação; ou más decisões por parte do MPI.

Capítulo 4

Implementação

Neste capítulo vamos descrever como implementamos o nosso MPI, concretizando as soluções idealizadas anteriormente. Apresentando, as ferramentas e tecnologias escolhidas para nos auxiliar, e posteriormente a implementação de cada componente, funcionalidade, e característica do MPI.

4.1 Ferramentas e tecnologias usadas no projecto

Nesta secção vamos introduzir as ferramentas e tecnologias que usámos no nosso projecto. Explicar quais as vantagens que nos trazem e o porquê de as termos escolhido.

4.1.1 Java

Java é uma linguagem de programação orientada a objectos, que é compilada para um *bytecode* para ser executado por uma máquina virtual própria [20]. Esta funcionalidade faz com que qualquer aplicação, feita em java, consiga correr sobre qualquer sistema operativo, logo que a máquina virtual de Java esteja instalada. Por ser uma linguagem com uma grande comunidade de utilizadores, pelas características já faladas, e porque verificamos que muitas das soluções e ferramentas de apoio, relacionadas com o projecto, eram feitas nesta linguagem, optamos por usá-la no projecto.

4.1.2 HL7 Application Programming Interface

O HL7 Application Programming Interface (HAPI) é um projecto *open-source* que consiste num *parser* para as versões 2.x do HL7, feito na linguagem Java [13]. Este projecto não está filiado na organização do HL7, mas é referenciado, pelo mesmo, como uma excelente solução.

O motivo pelo qual usámos esta tecnologia recai no facto de ser mais simples, ler e construir as mensagens HL7 V2 que precisamos para o projecto.

4.1.3 Java Sig

O Java Sig é uma API para manusear mensagens HL7 V3 [18]. Esta ferramenta foi inicialmente criada pelo grupo Java Sig, que mudou de nome para RIMBAA (*RIM Based Application Architecture*). Este grupo pertence à organização do HL7 e tem como objectivo criar uma aplicação capaz de usar o RIM, que é a base da versão 3 do HL7.

Esta API encontra-se em desenvolvimento e ainda apresenta alguns problemas, mas é também, a única ferramenta *open-source* no mercado. Na eventualidade de usarmos a versão 3 do HL7, ou qualquer funcionalidade por ela fornecida, esta será a ferramenta usada.

4.1.4 Hibernate

O Hibernate é uma *framework object-relational mapping* (ORM), técnica utilizada para reduzir a dificuldade na programação orientada a objectos utilizando bases de dados relacionais [16].

As tabelas das bases de dados são representadas através de classes de Java e o mapeamento entre os dois é feito por configuração de ficheiros XML. O que permite uma abstracção da linguagem SQL e facilita o manuseamento dos dados, e das relações um-para-muitos e muitos-para-muitos. O Hibernate suporta mapeamento para os tipos de dados comuns a Java.

Permitir persistência através de Plain Old Java Objects (POJOs), é também outra vantajosa funcionalidade do Hibernate. Apenas com a introdução de um construtor, sem argumentos na classe podemos tornar essa classe persistente, o que faz com que se

aperceba de todas as alterações feitas nas outras classes e reaja se assim for necessário.

O Hibernate é compatível com qualquer base de dados, logo que exista um Java Database Connectivity (JDBC), que é uma API escrita em Java e que tem como funcionalidade enviar instruções SQL para as bases de dados relacionais

Por ser *open-source*, feito em Java, e facilitar o trabalho relativamente ao tratamento de dados, esta ferramenta é considerada importante no desenvolvimento do nosso sistema.

4.1.5 ZK

ZK é uma *framework* para aplicações web baseada em Ajax, escrita em Java, que permite a criação de interfaces gráficos sem elevados conhecimentos de JavaScript e apenas o básico conhecimento em programação [19].

Suporta *scripting* embebido com a linguagem Java, o que permite que usemos a mesma linguagem na interface e no *backend*. Possui suporte para Hibernate, pode-se integrar com JSP, JSF e Servlet, e permite ligar os dados introduzidos na aplicação directamente aos POJOs do Hibernate no servidor.

O ZK é um Rich Interface Application (RIA) já consagrado por várias empresas. Por ter alguma estabilidade no mercado mundial, e ser compatível com todas as tecnologias que vamos usar, a nossa escolha recai nesta tecnologia para criar um Graphic User Interface (GUI) apelativo para a nossa aplicação.

4.2 Interoperabilidade

O HL7 é a base fundamental da comunicação do nosso MPI, por isso, ter uma relevância tão marcante no nosso projecto. A nossa primeira visão foi criar um sistema capaz de comunicar com HL7 V3, visto que este é apontado pela comunidade como o futuro do protocolo. A API JavaSig foi o nosso primeiro contacto com esta versão, e devido a inúmeras dificuldades encontradas no seu uso, à dificuldade em compreender o protocolo em si, a estrutura das suas mensagens e o pouco esclarecimento prestado pelos seus criadores, fez-nos optar por colocar a versão 3 como implementação futura.

Após considerarmos o HL7 V3 como trabalho futuro dedicámo-nos à versão 2, que já se encontra bem documentada e é muito usada a nível internacional [8]. Para esta versão

usámos a API do HAPI, que se mostrou de muito simples compreensão. Com esta API podemos aceder a qualquer um dos campos da mensagem, logo que se saiba qual a estrutura da mensagem recebida, sem grande dificuldade. Para criar uma mensagem, o processo é igualmente simples, novamente apenas precisamos de saber qual a estrutura da mensagem e o nome dos campos que queremos preencher. Usando esta API, as mensagens que temos implementadas no nosso projecto são suficientes para garantir as funcionalidades base de um MPI. Para melhor compreensão das mensagens usadas, ver os apêndices D e E.

4.2.1 Adição de uma pessoa

Para adicionar uma pessoa ao repositório, um consumidor pode enviar três mensagens para o MPI. Tanto a ADT_A01 (notificação de admissão de um paciente), a ADT_A04 (registo de um paciente) como a ADT_A05 (pré-admissão de um paciente) são mensagens que o nosso MPI vai assumir como pedido de adição de uma nova pessoa. Destas mensagens, apenas alguns dos segmentos que as constituem possuem dados importantes para o nosso sistema:

- O cabeçalho (MSH) - Possui toda a informação sobre a estrutura da mensagem e sobre quem a enviou. Esta informação é importante para sabermos qual a mensagem que recebemos e se o seu emissor é um consumidor autorizado.
- Dados demográficos (PID) - Possui todos os dados demográficos que se conhecem da pessoa em causa.
- Dados demográficos secundários (PD1) - Possui dados extra sobre a pessoa. Este segmento é opcional na mensagem.
- Parentes (NK1) - Alguns dados demográficos sobre os parentes, e o grau de relação para com o paciente. Este segmento é opcional na mensagem.
- Incapacidades (DB1) - Dados sobre incapacidades ou deficiências da pessoa. Este segmento é opcional na mensagem.
- Alergias (AL1) - Dados sobre alergias da pessoa. Este segmento é opcional na mensagem.

Com estes dados, contidos nestes segmentos, somos capazes de criar um registo completo sobre uma pessoa. Apesar de estas mensagens servirem para pedir a criação de

um novo registo, o MPI pesquisa sempre no seu repositório se não existe pelo menos um registo que possa referenciar a mesma pessoa, se for o caso, não adiciona a pessoa ao repositório mas actualiza os dados no registo encontrado.

Para confirmar a recepção destas mensagens, o MPI envia ao emissor a mensagem HL7 ACK. Esta mensagem não apresenta nenhuma estrutura especial, apenas informa se a operação de recepção e tratamento interno da mensagem enviada ocorreu com sucesso ou não.

4.2.2 Actualização de uma pessoa

Como já mencionamos, é possível actualizar os dados de uma pessoa com mensagens usadas para a adição, mas estas implicam ao MPI usar avançados algoritmos de correlação para encontrar os possíveis registos semelhantes. Para facilitar esse processo, um consumidor deve usar a mensagem ADT_A28 (adicionar informação de um paciente/pessoa). Esta mensagem possui os mesmos segmentos que as anteriores, logo, continuamos a poder enviar os mesmos valores, com a vantagem de esta mensagem trazer um conjunto de identificadores válidos, que identificam o registo que desejamos modificar. Assim o processo de actualização é mais rápido e eficaz. Esta mensagem também deve ser usada para o caso de desejarem adicionar um novo identificador ao registo.

Após a mensagem ADT_A28 ser tratada, tal como nas mensagens usadas para inserir uma nova pessoa, um ACK é enviado.

4.2.3 Pesquisar por pessoas

Para que um consumidor possa pesquisar no repositório do MPI por um paciente, pode enviar duas mensagens: a QBP_Q21 (obter dados demográficos de uma pessoa) para obter toda a informação de uma pessoa em concreto; e a QBP_Q22 (pesquisar por pessoas) para obter os dados demográficos mais relevantes de um conjunto de pessoas, que são filtradas com base em alguns valores.

Ambas as mensagens assentam a sua funcionalidade no segmento que define uma pesquisa, o QPD. Este segmento difere nas duas mensagens: na QBP_Q21, o terceiro campo do segmento contém um identificador, constituído por um valor alfa numérico e que entidade o atribuiu, capaz de identificar um único registo de forma imediata; na

QBP_Q22, o terceiro campo do segmento possui um conjunto de valores pelos quais temos de filtrar a nossa pesquisa. Esses valores são identificados de uma forma peculiar, mas simples de entender. Cada valor é constituído por dois campos, a localização do campo num segmento de uma mensagem HL7 e o valor esperado. Como exemplo, se quisermos filtrar a nossa pesquisa por todas as pessoas com o nome de família “Gomes” usamos “PID.5.1” no campo a pesquisar e “Gomes” no valor esperado. Como numa mensagem HL7 o segmento PID, no campo 5, sub-campo 1 refere-se ao nome de família da pessoa, sabemos qual o campo pelo que devemos filtrar a nossa mensagem. Obviamente que, para esta funcionalidade ser aplicável, ou o projecto está preparado para todos os campos possíveis, ou definimos quais os que aceitamos. No nosso projecto aceitamos como filtros, os campos: PID.5.1 (nome de família); PID.5.2 (primeiro nome); PID.7.1 (data de nascimento); PID.8 (género administrativo).

Para responder a estas mensagens existe apenas uma mensagem de resposta, que serve para as duas mensagens em causa. A RSP_K21 (resposta a ambas as QBP) tem como segmentos, relevantes a conhecer, o QAK (*query acknowledgment*), o PID, o PD1 e o NK1. Os três últimos segmentos já os explicamos e têm a função de transportar os dados sobre as pessoas, o QAK contém dois campos importantes: o segundo campo, que traz o estado da resposta, ou seja, se foi processado com sucesso ou se houve algum erro; e o quarto campo, que informa a quantidade de pessoas que a resposta transporta. As diferenças nesta estrutura para as duas QBP encontram-se no número de pessoas que transporta e na quantidade de informação que leva. Como resposta à QBP_Q21 devemos sempre encontrar apenas referência a uma pessoa e toda a informação existente sobre a mesma, na resposta à QBP_Q22 podemos encontrar zero ou mais referências a registos e devido a esta mensagem poder conter uma quantidade grande de registos, devemos apenas enviar alguma informação demográfica. A quantidade de informação enviada deve ser suficiente para identificar a pessoa, e se depois o emissor quiser informação mais detalhada pode usar uma QBP_Q21 com um dos identificadores enviados para a obter.

4.2.4 Notificar os consumidores

Como já foi dito na secção 3.1.1 (página 30), os consumidores precisam de ser notificados após qualquer modificação sofrida no repositório do MPI, para tal, o MPI usa a mensagem ADT_A28. Esta mensagem, como já foi explicado anteriormente, pode ser populada com todos os dados demográficos de uma pessoa e deve ser sempre considerada como uma actualização a um registo já existente. Se o consumidor apenas

recebe notificações sobre os registos que possui, esta mensagem não apresenta qualquer adversidade, mas se o consumidor é um consumidor de todo o repositório, é possível que receba esta mensagem e não tenha qualquer registo da pessoa em causa. É esperado que os consumidores deste género suportem a capacidade de identificar se a mensagem enviado pelo MPI se refere a uma pessoa já conhecida ou não.

4.2.5 Sincronismo dos Master Files

O MPI é uma aplicação cliente de um Master Files Manager, que também usa o protocolo HL7 para comunicar, logo temos de estar preparados para receber as mensagens por este enviadas. Sabemos que este envia a mensagem MFN_M14 (notificação para um *master file*) para notificar as alterações que são efectuadas nos *master files* em que o MPI está subscrito. Nesta mensagem vem a informação de qual o *master file* a modificar e o conjunto de alterações a efectuar. As alterações possíveis são: actualizar uma entrada; colocar uma entrada como desactivada; voltar a reactivar uma entrada; adicionar uma entrada; e eliminar uma entrada.

A esta mensagem devemos responder com uma MFK_M01 (*acknowledgement* para a MFN_M14) , onde informamos, para cada modificação pedida, se tivemos sucesso ou não.

4.2.6 Servidor

O MPI, para permitir que outros sistemas comuniquem com ele, precisa de ter um processo, que esteja sempre à escuta de novas mensagens. O HAPI fornece uma API que proporciona a criação de um servidor deste género (listagem: 4.1), onde as mensagens recebidas são enviadas para um *handler* que retorna uma possível resposta.

Listagem 4.1: Exemplo de um servidor (adaptado do HAPI)

```
1  /*
2   * Create a server to listen for incoming messages
3   */
4
5  // The port to listen on
6  int port = 8888;
7  // The transport protocol
8  LowerLayerProtocol llp = LowerLayerProtocol.makeLLP();
9  // The message parser
```

```

10 PipeParser parser = new PipeParser();
11 SimpleServer server = new SimpleServer(port, llp, parser);
12
13 /*
14  * The server may have any number of "application" objects
15  * registered to handle messages. We are going to create an
16  * application to listen to ADT^A01 messages.
17  */
18 Application handler = new ExampleReceiverApplication();
19 server.registerApplication("ADT", "A01", handler);
20
21 /*
22  * We are going to register the same application to handle
23  * ADT^A02 messages. Of course, we could just as easily have
24  * specified a different handler.
25  */
26 server.registerApplication("ADT", "A02", handler);
27
28 /*
29  * Another option would be to specify a single application
30  * to handle all messages, like this:
31  *
32  * server.registerApplication("*", "*", handler);
33  */
34
35 // Start the server listening for messages
36 server.start();
37
38 // After the job is done
39 server.stop();

```

No caso das notificações, é o MPI que tem de iniciar a comunicação com as aplicações consumidoras. Neste caso o HAPI também possui, na sua API, uma solução simples de aplicar (listagem: 4.2), onde apenas precisamos de saber a porta e o endereço em que o consumidor deseja ser contactado. Esta conexão fica à espera de uma resposta por parte do receptor.

Listagem 4.2: Exemplo de uma conexão ao servidor (adaptado do HAPI)

```

1 /*
2  * Now, create a connection to that server, and send a message
3  */
4
5 // Create a message to send
6 String msg = "MSH|^~\&|MPI|MPI||20090310151441||ADT^A01|12345|P|2.5\r"

```

```

7         + "EVN|||20090310151519|||20090310151543|Hospital\r"
8         + "PID|||19^^^HA~13^^^HP||Gomes^Pedro^Tiago||19861113000000|M
          |||&Rua  Maior&890^^1^2^2023^PT^R|||PT|D|CAT|PT|
          PT\r"
9         + "PV1||";
10
11 Parser p = new GenericParser();
12 Message adt = p.parse(msg);
13
14 // The connection hub connects to listening servers
15 ConnectionHub connectionHub = ConnectionHub.getInstance();
16
17 // A connection object represents a socket attached to an HL7 server
18 Connection connection = connectionHub.attach("localhost", port, new
          PipeParser(), MinLowerLayerProtocol.class);
19
20 // The initiator is used to transmit unsolicited messages
21 Initiator initiator = connection.getInitiator();
22 Message response = initiator.sendAndReceive(adt);
23
24 String responseString = parser.encode(response);
25 System.out.println("Received response:\n" + responseString);
26 /*
27  * MSH|^~\&|||20070218200627.515-0500|ACK|54|P|2.2
28  * MSA|AA|12345
29  */
30
31 // Close the connection and server
32 connection.close();

```

4.3 Hibernate

Usar Hibernate acabou por ser uma necessidade imposta pelo nosso objectivo de desenharmos um sistema que fosse independente do tipo de base de dados usada. Com o uso do Hibernate passamos a ter classes (listagem 4.3) que reflectem as tabelas do nosso repositório e ficheiros XML (listagem 4.4) que fazem a ligação entre as classes e as tabelas respectivas

Listagem 4.3: Classe Patient.Consumer

```

1 package Patient;
2 /**

```

```

3  * Generated Apr 27, 2009 10:13:52 AM by Hibernate Tools 3.2.1.GA
4  * Consumer generated by hbm2java
5  */
6  public class Consumer implements java.io.Serializable {
7
8      private Long id;
9      private AssigningAuthority assigningAuthority;
10     private int systemAccess;
11     private String url;
12     private Integer port;
13
14     public Consumer() {
15     }
16     // Gets and sets ...
17 }

```

Listagem 4.4: Ficheiro XML responsável por mapear a classe Consumer

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
3      3.0//EN" "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd"
4      >
5  <!-- Generated Apr 27, 2009 10:13:55 AM by Hibernate Tools 3.2.1.GA -->
6  <hibernate-mapping>
7      <class catalog="mpi" name="Patient.Consumer" table="Consumer">
8          <id name="id" type="java.lang.Long">
9              <column name="id" />
10             <generator class="identity" />
11         </id>
12         <many-to-one class="Patient.AssigningAuthority" fetch="select" name=
13             "assigningAuthority">
14             <column name="assigning-authority-id" />
15         </many-to-one>
16         <property name="systemAccess" type="int">
17             <column name="system_access" not-null="true" />
18         </property>
19         <property name="url" type="string">
20             <column length="50" name="url" />
21         </property>
22         <property name="port" type="java.lang.Integer">
23             <column name="port" />
24         </property>
25     </class>
26 </hibernate-mapping>

```

Como se pode ver pelos exemplos das listagens 4.3 e 4.4 existe um mapeamento directo entre a classe, o xml e a tabela, o que faz com que a nível de código não seja necessário qualquer conhecimento sobre o repositório ou mesmo qual a linguagem que esta usa.

Para conseguirmos trabalhar com os registos do nosso repositório primeiro precisamos de os obter do próprio repositório. Existem duas formas de o fazer: pesquisando o repositório através de uma *query* feita na linguagem própria do Hibernate, o HQL; ou através do método “load(classe, primary key)”. Existe uma grande diferença entre os dois, enquanto que o HQL é uma linguagem muito semelhante a uma *query* de SQL, o que faz com que tenhamos de criar uma pesquisa elaborada para conseguirmos obter o resultado pretendido, o método “load” permite-nos obter exactamente a entrada de tabela que desejamos logo que seja fornecido a classe que a referencia e a sua chave primária. Cada um tem as suas vantagens e locais de uso específico.

De resto é muito simples de usar o Hibernate e facilita em muito o manuseamento dos dados do nosso repositório (exemplo na listagem 4.5). Os métodos “save” e “update” são os responsáveis por conseguirmos guardar e modificar, respectivamente, os dados.

Listagem 4.5: Manuseamento da classe Consumer

```
1 Session hibernateSession = HibernateUtil.getSessionFactory().  
    getCurrentSession();  
2 hibernateSession.beginTransaction();  
3  
4 AssigningAuthority assigningAuthority = new AssigningAuthority();  
5 assigningAuthority.setName(consumerName);  
6 ...  
7  
8 Consumer consumer = new Consumer(assigningAuthority, systemAccess, url,  
    port);  
9 hibernateSession.save(consumer);  
10 hibernateSession.getTransaction().commit();
```

No nosso projecto o pacote Patient contém todas as classes, e ficheiros XML correspondentes, que mapeiam o repositório do MPI. A salientar neste repositório estão as imensas relações que possuímos entre as várias classes. Estas relações mapeiam as verdadeiras relações muitos-para-um e muitos-para-muitos do repositório. Com o Hibernate, visto que usamos classes, o acesso a estas relações torna-se muito simples, pois estão representadas por listas e apontadores. O Hibernate apesar de ser algo que seria descartável se o projecto utilizasse apenas um tipo de base de dados, veio facilitar em muito o trabalho.

4.4 Algoritmos de correlação

Existem duas situações onde a correlação de dados é muito importante num MPI [5]. A primeira acontece quando se adiciona um novo registo ao repositório. Este registo precisa de ser correlacionado a todos os dados das pessoas já existentes, de forma a garantir que não existe nenhuma similaridade entre eles. Este processo deve ser automático e imediato. A segunda situação acontece quando uma pesquisa é feita ao MPI e é necessário retornar registos do repositório, como resposta. Mesmo usando estes algoritmos, a resposta não deve ser muito demorada e deve ser capaz de retornar resultados exactos, semelhantes e equivalentes. A quantidade de dados existente, o seu tipo e a qualidade dos mesmos vão ser, no final, o que vai determinar a exactidão destes algoritmos, independentemente das suas características.

Existe uma variedade de métodos que podem ser usados para correlacionar dados. A mais simples é a comparação dos dados por igualdade de todo o seu conteúdo, onde uma comparação é feita caracter a caracter, mas este método não nos fornece dados que sejam semelhantes ou equivalentes. Para tal é necessário aplicar métodos que considerem adições, remoções, trocas e transposições de caracteres para considerar se duas palavras podem conter semelhanças. Exemplos de algoritmos que implementam estes métodos são: a distancia de *Hamming*; o algoritmo de *Damerau-Levenshtein*; e o algoritmo *Shift-Or*. A fonética também é um dos métodos usados para encontrar semelhanças entre palavras, transformando cada sílaba num código é possível atribuir semelhanças fonéticas entre duas palavras.

Existem outras formas de correlacionar dados, não só baseadas em algoritmos matemáticos, mas também através de aprendizagem inteligente por parte do MPI, métodos baseados em regras e métodos de controlo de erros. Estes métodos são mais capazes, mas também muito mais difíceis de implementar. Estes métodos precisam sempre de muitas alterações para se conseguirem comportar de forma eficaz, e só com muitos dados é que se conseguem adaptar. É simples um algoritmo deste género deixar passar um caso raro ou único, mas depois de ser detectado é simples de afinar o método para nunca mais falhar. Na imagem 4.1, podemos observar alguns métodos já existentes e até onde se estendem as suas capacidades.

Independentemente de quais os métodos ou algoritmos usados, o ideal é conseguir analisar o problema de vários ângulos e passando por todos os níveis de igualdades possíveis. Estes são os passos que achamos necessários tomar para atingir uma correlação eficaz (ver imagem 4.2):

Comparison of matching techniques.													
	Letter-Related Mismatches					Word- or Field-Related Mismatches							
	Transposition	Omission	Misspelling (few letters)	Typing error	Misread characters	Change of address	Change of name	Nicknames	Sex change	Fraud	Change of zip code	Change of phone number	Incorrect data entry (word)
Damerau-Levenstein	X		X	X	X								
Jaro Weighting	X												
Fellegi-Sunter SOUNDEX/ NYSIIS		X	X	X	X		X						
Neural Network						X	X	X	X		X	X	
Correlation		X	X	X	X								
Rule-based Approach						X	X	X	X		X	X	
Fuzzy Logic Approach						X	X		X		X	X	
Clustering	X	X	X	X	X								
Matched Filter		X	X	X	X								

Figura 4.1: Análise dos vários métodos de correlação

Parse : Obter dos valores recebidos aqueles que são usados para correlação. Na sua generalidade estes dados são: o nome de família; o nome próprio; os restantes nomes, iniciais; identificações governamentais; morada; localidade; cidade; código postal; data de nascimento; género administrativo; e o estado civil.

Cultura : A análise da origem étnica de um nome pode ser fundamental na aplicação de algoritmos fonéticos, por isso se conseguirmos deduzir a origem cultural do nome conseguimos melhorar os nossos resultados.

Equivalências : Muitos nomes têm diminutivos, alcunhas, ou nomes equivalentes a que os associamos. Mesmo as ruas, locais e instituições podem ter mais do que um nome como referência. O uso de uma tabela com essas equivalências, novamente enriquece os nossos resultados. O uso de *master files* pode diminuir estes erros, ou ajudar a detectar equivalências.

Comparação : Comparar os nossos valores com os valores do repositório usando todos os métodos de comparação de palavras mencionados.

Fonética : Comparar os nossos valores com os valores do repositório usando todos os métodos de análise fonética de uma palavra.

Regras : Não devemos avaliar apenas as semelhanças que encontramos entre os dados dos registos, as diferenças podem ajudar a devolver resultados mais precisos.

O exemplo da mudança de morada ou estado civil são um caso vulgar e se detectarmos essas diferenças através de algumas regras, facilmente podemos reduzir o nosso resultado a um registo.

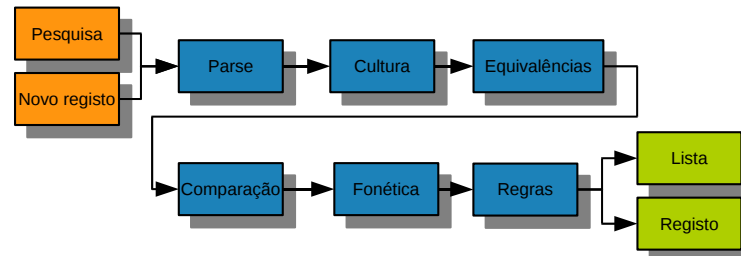


Figura 4.2: Passos necessários a um algoritmo de correlação eficiente

De todo o processo explicado, apenas alguns dos passos estão implementados no nosso projecto. O nosso projecto está preparado para reagir aos dois cenários anteriormente expostos: obter do conjunto de dados recebidos os que são importantes para conseguir correlacionar registos, que no nosso caso são o último nome, o nome próprio, a data de nascimento, o género administrativo e qualquer identificador; usar os *master files* para validar alguns dos campos; e comparar os dados através da sua fonética com o uso do algoritmo SOUNDEX (ver listagem: 4.6), já implementado nas diversas linguagens de base de dados e no Hibernate. Apesar de não termos tudo o que é necessário para possuirmos um algoritmo de correlação completo, conseguimos obter um resultado muito satisfatório no que se trata a encontrar registos semelhantes através dos métodos que implementamos.

Listagem 4.6: Exemplo da criação de uma pesquisa usando o SOUNDEX

```

1 String hql = "select Person.id"
2       + " from Person as Person , Name as Name"
3       + " where Person.id=Name.person";
4
5 hql = hql + " and soundex(Name.given) = soundex('"
6       + pid.getPatientName(0).getGivenName().toString()
7       + " ')" ;
8
9 hql = hql + " and soundex(Name.family) = soundex('"
10      + pid.getPatientName(0).getFamilyName().getSurname().toString()
11      + " ')" ;
12
13 String dateOfBirth = pid.getDateTimeOfBirth().getTime().toString();
14 hql = hql + " and Person.birthTime like '" + dateOfBirth.substring(0, 4)

```



```
15 |         + "-" + dateOfBirth.substring(4, 6) + "-" + dateOfBirth.  
16 |           substring(6, 8)  
    |         + "%'";
```

4.5 Interface gráfico

Todo o nosso projecto baseia-se na comunicação entre sistemas médicos, sem qualquer interacção humana directa, usando o protocolo HL7 como forma de comunicação. Mas independentemente da interacção humana não se poder sentir nas acções automáticas do MPI, uma revisão humana a todas a suas acções é necessário. Para tal decidimos criar um interface *web* capaz de proporcionar a um administrador do sistema: registos de todas as acções efectuadas pelo MPI; controlo sobre os acessos e privilégios dos vários consumidores; as acções base de um MPI para poder corrigir alguns erros feitos pelo sistema; e a hipótese de modificar o repositório de forma restrita.

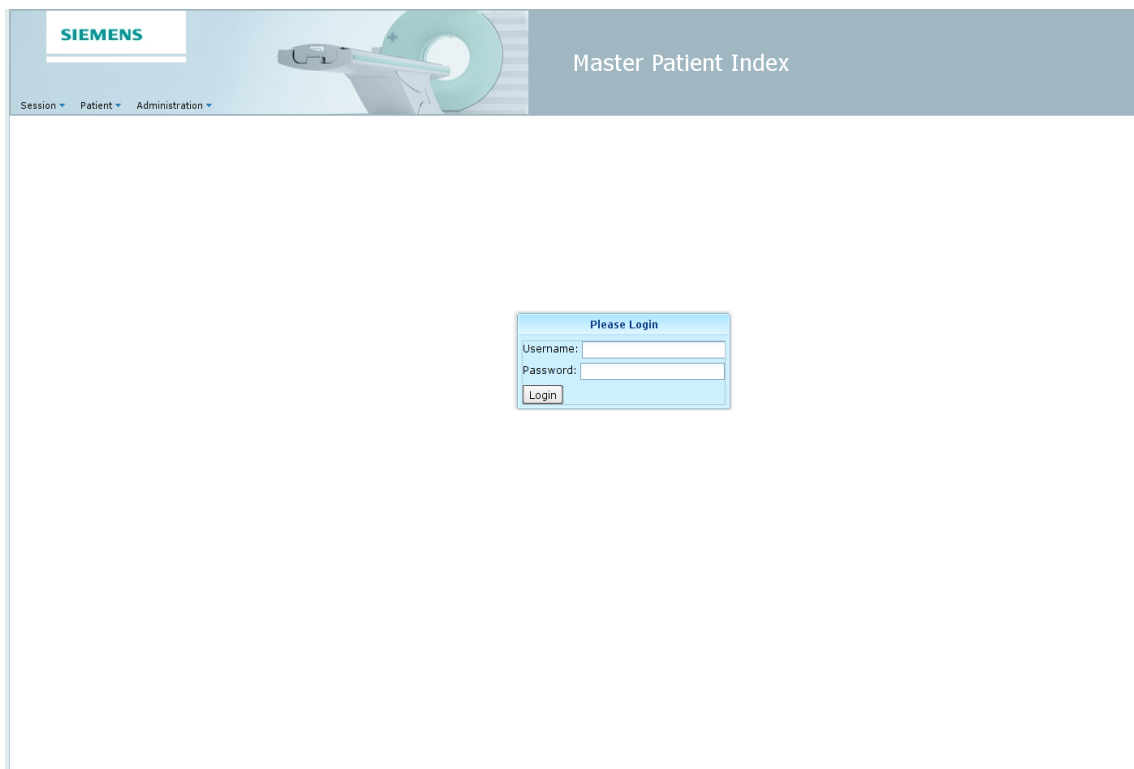


Figura 4.3: Ecrã inicial do interface gráfico

Utilizando o ZK desenvolvemos uma interface simples (ver imagem 4.3), que conseguiu atingir os seguintes objectivos: pesquisar por pessoas, observar o resultado da pesquisa

e poder ver em pormenor os dados de uma pessoa em específico; adicionar uma grande quantidade de pessoas através de ficheiros csv; e adicionar consumidores através de ficheiros csv.

A pesquisa que esta interface fornece ao administrador é muito idêntica aos serviços disponibilizados por HL7. Os campos pelos quais podemos pesquisar uma pessoa são: o nome próprio; o último nome; o género administrativo; um intervalo de tempo que corresponde à data de nascimento da pessoa; a localidade onde vive; qualquer identificador da pessoa; e ainda por entidades que possam ter atribuído um identificador à pessoa. Esta pesquisa também suporta os mesmos algoritmos de correlação que estão implementados para as mensagens, por exemplo, se pesquisarmos por todas as pessoas com o nome próprio “Ana”, podemos observar, pela imagem 4.4 que são retornados registos com o nome “Anne” e “Ani”.



The screenshot shows the Siemens Master Patient Index interface. At the top, there's a Siemens logo and a navigation bar with 'Session', 'Patient', and 'Administration' tabs. Below this is a search bar with fields for 'ID', 'First Name' (containing 'Ana'), 'Family Name', 'Administrative Gender', 'Assigning Authority', 'Date of Birth', and 'Locality'. A 'Search' button is present. The results are displayed in a table with columns: ID, First Name, Second Name, Family Name, Date of Birth, and Administrative Gender. The table lists 25 results, all with 'First Name' as 'ANA'. The 'Second Name' and 'Family Name' vary, including names like 'ANNE MARIE PIERRE BRANCO', 'ANI ELISANGELA MOREIRA COXO', 'ANA PINA PEREIRA', etc. The 'Date of Birth' and 'Administrative Gender' (all 'Feminino') are also listed. At the bottom, there's a pagination bar showing '1 / 25' and a total count of '[1 - 20 / 495]'.

ID	First Name	Second Name	Family Name	Date of Birth	Administrative Gender
16831	ANNE	ANNE MARIE PIERRE BRANCO	BRANCO	1970-07-14 18:00:00	Feminino
9797	ANI	ANI ELISANGELA MOREIRA COXO	COXO	1981-04-03 00:00:00	Feminino
49	ANA	ANA PINA PEREIRA	PEREIRA	1971-10-26 16:00:00	Feminino
103	ANA	ANA RITA RIBEIRO SILVA	SILVA	2000-02-27 10:00:00	Feminino
134	ANA	ANA PAULA RIBEIRO LOURENCO	LOURENCO	1998-12-25 21:00:00	Feminino
161	ANA	ANA MARIA JACINTO CAROCA	MATEUS	1973-04-14 01:00:00	Feminino
165	ANA	ANA CRISTINA GONCALVES MENDES	SILVA	2006-07-10 12:00:00	Feminino
232	ANA	ANA RITA FILIPE RIBEIRO BARATA	BARATA	1983-08-06 06:00:00	Feminino
274	ANA	ANA RITA RODRIGUES VALENTE	VALENTE	2007-04-10 05:00:00	Feminino
275	ANA	ANA SOFIA RAMOS FERREIRA	FERREIRA	1972-01-04 03:00:00	Feminino
302	ANA	ANA RITA BARTOLO VALENTE	VALENTE	1980-08-03 16:00:00	Feminino
304	ANA	ANA PAULA PIRES MARTINS	MARTINS	1997-11-09 08:00:00	Feminino
312	ANA	ANA CATARINA SIMOES MARTINS	RODRIGUES	1972-07-15 01:00:00	Feminino
353	ANA	ANA BELO RAMOS FERNANDES	FERNANDES	1976-08-01 18:00:00	Feminino
359	ANA	ANA JUSTINO LOURENCO	LOURENCO	2004-06-09 12:00:00	Feminino
388	ANA	ANA LUISA DUARTE	DUARTE	1990-04-27 10:00:00	Feminino
398	ANA	ANA TERESA BRITO PENEDO	PENEDO	2008-10-04 14:00:00	Feminino
505	ANA	ANA SOFIA PEREIRA BRANCO LIMA	LIMA	1980-06-17 16:00:00	Feminino
506	ANA	ANA SOFIA RODRIGUES LOURENCO	LOURENCO	1989-05-08 12:00:00	Feminino
520	ANA	ANA CRISTINA MARTINS CALADO	CALADO	1977-05-13 19:00:00	Feminino

Figura 4.4: Exemplo do resultado dos algoritmos de correlação

Os valores que mostramos nos resultados, como na visualização específica de uma pessoa não são definitivos. Os valores por nós mostrados, apesar de estarem contidos na versão final, servem apenas de prova de conceito.

Função única, em todo o projecto, que se encontra implementada neste interface é a possibilidade de adicionar pessoas ou consumidores através de ficheiros csv (ver

imagem 4.5). Estes ficheiros são muito simples de criar e de muito fácil processamento. No caso dos csv relativos as pessoas seguimos um formato simples, apenas como prova de conceito, onde aceitamos: número do BI; nome; género; data de nascimento; morada; localidade (segundo o id dos nossos *master files*); número de telefone; id de um hospital (segundo o id dos nossos consumidor); e o identificador nesse hospital.

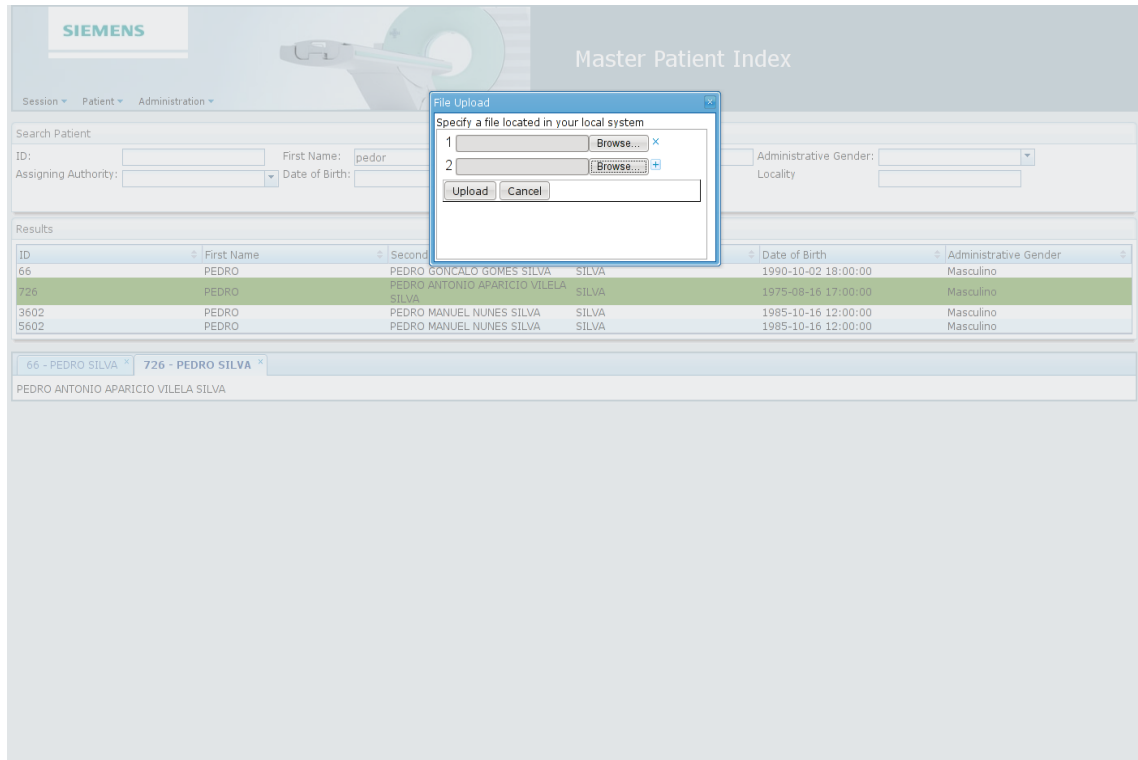


Figura 4.5: Opção para adicionar dados através de um ficheiro csv

Estas operações não superam todos os objectivos inicialmente propostos para o interface do administrador, mas todas as acções possíveis de executar a um registo estão implementadas para as mensagens, logo, apenas precisamos de criar um interface gráfico para este requisito ser atingido, o mesmo acontece com os *master files* e os consumidores. Em relação aos registos das acções efectuadas pelo MPI, estamos a criar, com sucesso, *logs* personalizados. A estes *logs* falta fazer um *parse* e mostrá-los no GUI. Resumindo, apesar de não termos todos os objectivos cumpridos, temos a infraestrutura preparada para os podermos implementar definitivamente.

Capítulo 5

Conclusão

Neste capítulo será feita uma apreciação global dos resultados obtidos e consequente satisfação com os mesmos, no sentido de apresentar uma análise crítica ao trabalho desenvolvido, apresentando, no fim, caminhos para eventual trabalho futuro no seguimento da nossa prestação.

5.1 Objectivos Cumpridos

Observando a lista de objectivos a que nos propusemos no início do projecto, podemos concluir que não conseguimos atingir todos os objectivos pretendidos. De uma forma geral, conseguimos criar um MPI com as seguintes características e funcionalidades:

- adicionar pessoas;
- alterar os dados de pessoas existentes;
- adicionar identificadores a uma pessoa existente;
- reconhecer igualdades nos registos;
- responder às pesquisas efectuadas ao repositório com uma lista de possíveis resultados;
- sincronizar os sistemas associados;
- responder a todas as acções em tempo real e sem auxílio humano;

- usar todos os padrões definidos e aceites pela comunidade.

Não atingimos, na totalidade, os seguintes objectivos:

- unir dois registos num só;
- criar um interface gráfico para o administrador;
- o MPI conseguir adaptar-se aos sistemas já existentes, e não o contrário;

É notório que conseguimos atingir uma grande quantidade dos objectivos iniciais, mas é importante ressaltar que para o projecto final, muitos outros, não definidos inicialmente foram cumpridos, para que estes pudessem ter sucesso. O estudo que levamos a cabo sobre os registos médicos; sobre quais os dados demográficos considerados importantes para identificar uma pessoa; sobre quais os dados médicos, considerados essenciais para a vida de uma pessoa, relevantes para uma primeira abordagem clínica; e sobre os dados que são suportados nas mensagens HL7; permitiram a criação de um repositório capaz de responder a todas as nossas necessidades. Sendo este repositório fulcral para o desenvolvimento do projecto, consideramos o trabalho desenvolvido à sua volta um sucesso e por isso destacámo-lo em parceria com os objectivos inicialmente propostos. O mesmo acontece perante o sucesso da comunicação do nosso sistema com outros sistemas. Onde tentamos, em conjunto com os restantes projectos em desenvolvimento na empresa, comunicar através de mensagens HL7 e obtivemos sucesso na sua troca, com o Master Files Manager e com o Encounter Management [3].

Em relação aos objectivos não atingidos, existem dois elementos que se evidenciam pela falta de implementação e um, a nosso ver, por impossibilidade de atingir. Unir dois registos num só, por motivo de referenciar a mesma pessoa, é um processo moroso que pode ser iniciado através de várias mensagens HL7, tais como a ADT^A18; a ADT^A30; a ADT^A34; a ADT^A35; a ADT^A36; a ADT^A39; a ADT^A40; a ADT^A41; e a ADT^A42 (ver o capítulo 3 de [8] para mais informações sobre todas as mensagens). Apesar de existir um vasta quantidade de mensagens para que o MPI execute esta funcionalidade, a forma como esta acção seria implementada deixou-nos sempre com imensas dúvidas, sendo muito difícil de implementar sem intervenção humana. A criação do interface gráfico, apesar de ter toda a infra-estrutura criada, ficou em falta em relação à parte visual. Neste objectivo a falta de tempo para implementação foi a principal razão do insucesso do objectivo.

Pretender criar um MPI capaz de se adaptar a qualquer sistema já existente, é em Portugal, neste momento, algo impraticável. Apesar do nosso projecto comunicar

com um protocolo aceite internacionalmente os sistemas existentes em Portugal, na sua maioria, não o fazem ou, não usam as mensagens HL7 de forma correcta. Por isso os sistemas teriam de sofrer alterações para conseguirem comunicar com o nosso MPI, quer no uso do protocolo como na sua forma de funcionamento. Isto não implica que o MPI não seja independente, autónomo e facilmente integrável, mas sim que neste momento os sistemas de informação médica não estão preparados para um sistema destes, e por isso consideramos este objectivo como não atingido.

5.2 Dificuldades Sentidas

Este projecto responde a uma necessidade há muito sentida no mundo da saúde e por isso esperávamos que existisse uma comunidade grande à volta deste assunto. Infelizmente o esperado não foi comprovado, a escassez de documentação concreta sobre os objectivos de um MPI obrigou-nos a uma pesquisa muito demorada para conseguirmos obter alguma informação válida. E mesmo assim, apenas no decorrer da fase de implementação, é que conseguimos obter respostas a algumas perguntas que inicialmente colocamos.

Outra das dificuldades sentidas no projecto foi o uso do HL7. Por ter sido a primeira vez que observamos este protocolo, não possuíamos o conhecimento necessário para, em muitos dos casos, discernirmos rapidamente as soluções pretendidas. Mas a maior dificuldade que tivemos com este protocolo foi no uso da versão 3 do HL7, que não conseguimos compreender. É de ressaltar que tentamos contactar os responsáveis pela criação desta versão, no núcleo do grupo do HL7, mas nunca conseguimos estabelecer uma relação de onde conseguíssemos obter muitas respostas.

Um pouco motivado pela falta de tempo que todos os outros problemas nos trouxeram, não tivemos muito tempo para aprofundar os nossos conhecimentos sobre o ZK, o que limitou muito o nosso interface gráfico. Independentemente da nossa experiência com a tecnologia, deparamos-nos com facilidades no seu uso, muito devido à partilha de conhecimentos por parte de colegas de trabalho, mas uma enorme adversidade no *design* da interface.

5.3 Trabalho Futuro

Como trabalho futuro podemos apresentar todos os objectivos não cumpridos e algumas características que gostávamos de ver implementadas, tais como:

- implementar o *merge*, que se resume a unir dois registos do nosso repositório num só;
- desenhar para o administrador uma interface gráfica onde possa facilmente gerir as acções do MPI tal como administrar os acessos ao sistema;
- rever, perante entidades responsáveis, o repositório e garantir que não existe um único dado que nos tenha possivelmente escapado;
- permitir o uso de qualquer implementação da versão 2 do HL7 para comunicar com o MPI, visto apenas aceitarmos a implementação em versão 2.5;
- implementar a versão 3 do HL7;

Estas adições no nosso projecto tornavam-no capaz de ser implementado em qualquer sistema de informação médica, e por isso serem por nós apontadas como sequência ao nosso trabalho. Obviamente não deixando de parte o aperfeiçoamento e os testes a tudo o que foi implementado, de forma exhaustiva.

5.4 Conclusões finais

Apesar de não termos obtido como resultado final, um MPI completamente funcional e capaz de executar todas as suas funções, estamos satisfeitos com o nosso trabalho. Um projecto desta magnitude precisa de muito estudo e análise antes de se proceder a qualquer tipo de implementação, o que num projecto de 7 meses não é possível. Olhando agora para o início do projecto, é obvio que não podíamos fugir a uma fase de estudo tão intensa, mas tinha sido melhor abordagem entrar em contacto com o HL7 e outras entidades mais cedo, para que as nossas decisões fossem tomadas mais rapidamente do que foram. A decisão de usar várias tecnologias que nunca tínhamos usado, foi outro dos motivos que prejudicou o nosso tempo, mas, excluindo o uso do ZK, não achamos ter sido um desperdício o tempo gasto para aprendizagem. No caso do ZK não conseguimos ter uma avaliação tão completa como para com as restantes tecnologias, mas a curva de aprendizagem pareceu-nos muito elevada.

Em conclusão, achamos que estivemos no nível pretendido pelo projecto, prejudicados pela análise dos objectivos que considerávamos possíveis de concluir no tempo que possuíamos. No fim: possuímos um estudo alargado sobre o tema, conhecendo a fundo o que é um MPI e para que serve; e apresentamos uma prova de conceito muito próxima da realidade de uma possível implementação de um MPI.

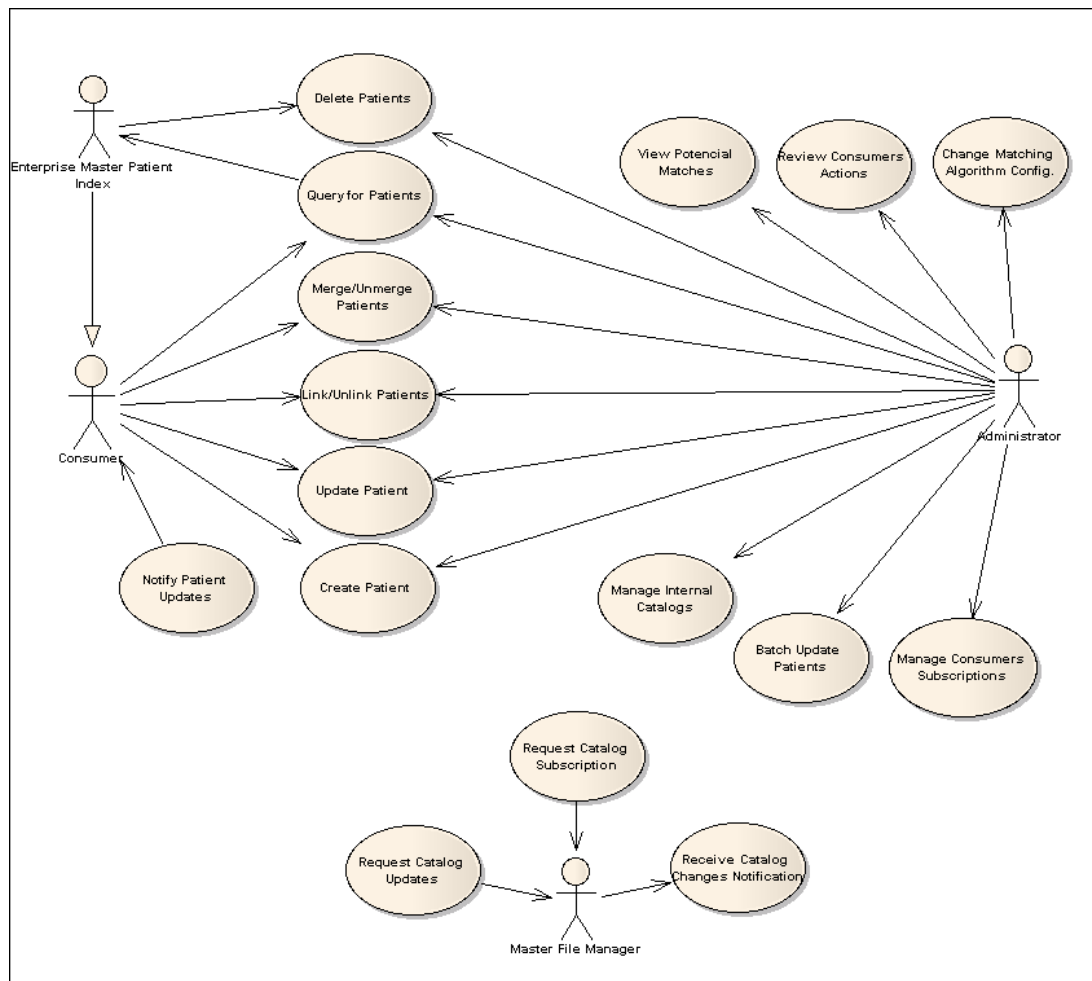
Apêndice A

Acrónimos

ADT	Admission Discharge and Transfer
CIS	Cardiology Information System
EHR	Eletronic Health Record
EM	Encounter Management
EMPI	Enterprise Master Patient Index
EPR	Electronic Patient Record
HIS	Hospital Information System
HL7	Health Level 7
IHE	Integrating the Healthcare Enterprise
LIS	Laboratory Information System
MPI	Master Patient Index
MPI/PD	Master Patient Index / Patient Demographic
NHS	National Health Service
OIS	Oncology Information System
PAS	Patient Administration System
PDQ	Patient Demographic Query
PIX	Patient Identifier Cross-reference
RIS	Radiology Information System
VA	Veterans Affairs

Apêndice B

Casos de Uso



Apêndice C

Atributos que o MPI suporta

Atributo	Cardinalidade	Atributo composto por
ID	1	
Identificadores	1..*	Entidade; Identificador; Data de início; Data de fim;
Nome	1..*	Primeiro nome; Nome de família; Nomes restantes; Prefixo; Sufixo; Tipo de Nome;
Morada	0..*	Rua; Número; Código postal; Localidade; Cidade; País; Tipo de residência;
Data de Nascimento	0..1	
Data de Óbito	0..1	
Género administrativo	0..1	
Estado Civil	0..1	
Nascimento múltiplo	0..1	
Contacto	0..*	Contacto; Extensão; País; Tipo de contacto; Hora para contacto
Cidadania	0..*	Identificador; País; Data de início
Local de Nascimento	0..1	Nome do local; Morada
Dador de Órgãos	0..1	
Comunicação	0..1	Linguagem; Proficiência
Incapacidades	0..*	Tipo de incapacidade; Data de início
Relações	0..*	Pessoa; Tipo de relação
Última actualização	1	

Apêndice D

Mensagens HL7

D.1 ADT^A01 - Notificação de admissão/visita

ADT^A01^ADT_A01	Mensagem ADT (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
EVN	Tipo do evento
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ ROL }]	Acção clínica
[{ NK1 }]	Parentes
PV1	Visita do paciente
[PV2]	Visita do paciente - informação extra
[{ ROL }]	Acção da visita
[{ DB1 }]	Informação sobre incapacidades
[{ OBX }]	Observações / Resultados
[{ AL1 }]	Alergias
[{ DG1 }]	Diagnósticos
[DRG]	Grupo dos diagnósticos
[{	– Início do procedimento –
PR1	Procedimento
[{ ROL }]	Acção do procedimento
}]	– Fim do procedimento –
[{ GT1 }]	Tutor ou responsável pelo paciente

[{	–Início das seguradoras
IN1	Seguradora
[IN2]	Seguradora - informação extra
[{ IN3 }]	Seguradora - informação extra
[{ ROL }]	Acção da seguradora
}]	–Fim das seguradoras
[ACC]	Informação do acidente
[UB1]	Factura do procedimento
[UB2]	Factura do procedimento
[PDA]	Autopsia e informação sobre morte

D.2 ADT^A04 - Registar um Paciente

ADT^A04^ADT_A01 Mensagem ADT (Tipo^Evento^Estrutura)

MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
EVN	Tipo do evento
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ ROL }]	Acção clínica
[{ NK1 }]	Parentes
PV1	Visita do paciente
[PV2]	Visita do paciente - informação extra
[{ ROL }]	Acção da visita
[{ DB1 }]	Informação sobre incapacidades
[{ OBX }]	Observações / Resultados
[{ AL1 }]	Alergias
[{ DG1 }]	Diagnósticos
[DRG]	Grupo dos diagnósticos
[{	– Início do procedimento –
PR1	Procedimento
[{ ROL }]	Acção do procedimento
}]	– Fim do procedimento –
[{ GT1 }]	Tutor ou responsável pelo paciente
[{	–Início das seguradoras
IN1	Seguradora

[IN2]	Seguradora - informação extra
[{ IN3 }]	Seguradora - informação extra
[{ ROL }]	Acção da seguradora
}}	–Fim das seguradoras
[ACC]	Informação do acidente
[UB1]	Factura do procedimento
[UB2]	Factura do procedimento
[PDA]	Autopsia e informação sobre morte

D.3 ADT^A05 - Pré-admissão de um Paciente

ADT^A05^ADT_A05 Mensagem ADT (Tipo^Evento^Estrutura)

MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
EVN	Tipo do evento
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ ROL }]	Acção clínica
[{ NK1 }]	Parentes
PV1	Visita do paciente
[PV2]	Visita do paciente - informação extra
[{ ROL }]	Acção da visita
[{ DB1 }]	Informação sobre incapacidades
[{ OBX }]	Observações / Resultados
[{ AL1 }]	Alergias
[{ DG1 }]	Diagnósticos
[DRG]	Grupo dos diagnósticos
[{	– Início do procedimento –
PR1	Procedimento
[{ ROL }]	Acção do procedimento
}}	– Fim do procedimento –
[{ GT1 }]	Tutor ou responsável pelo paciente
[{	–Início das seguradoras
IN1	Seguradora
[IN2]	Seguradora - informação extra
[{ IN3 }]	Seguradora - informação extra
[{ ROL }]	Acção da seguradora

}}	–Fim das seguradoras
[ACC]	Informação do acidente
[UB1]	Factura do procedimento
[UB2]	Factura do procedimento

D.4 ADT^A28 - Adicionar informação de um Paciente/Pessoa

ADT^A28^ADT_A05 Mensagem ADT (Tipo^Evento^Estrutura)

MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
EVN	Tipo do evento
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ ROL }]	Acção clínica
[{ NK1 }]	Parentes
PV1	Visita do paciente
[PV2]	Visita do paciente - informação extra
[{ ROL }]	Acção da visita
[{ DB1 }]	Informação sobre incapacidades
[{ OBX }]	Observações / Resultados
[{ AL1 }]	Alergias
[{ DG1 }]	Diagnósticos
[DRG]	Grupo dos diagnósticos
[{	– Início do procedimento –
PR1	Procedimento
[{ ROL }]	Acção do procedimento
}}	– Fim do procedimento –
[{ GT1 }]	Tutor ou responsável pelo paciente
[{	–Início das seguradoras
IN1	Seguradora
[IN2]	Seguradora - informação extra
[{ IN3 }]	Seguradora - informação extra
[{ ROL }]	Acção da seguradora
}}	–Fim das seguradoras
[ACC]	Informação do acidente

[UB1]	Factura do procedimento
[UB2]	Factura do procedimento

D.5 ACK - Acknowledgement

ACK^Axx^ACK	Típico Acknowledgement (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
MSA	Acknowledgement da mensagem
[ERR]	Erro

D.6 QBP^Q21 - Obter dados demográficos de uma Pessoa

QBP^Q21^QBP_Q21	Pesquisa por parâmetro (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
QPD	Definição dos parâmetros para a pesquisa
RCP	Controlo para a resposta
[DSC]	Apontador para mensagem de continuação

D.7 RSP^K21 - Resposta a uma QBP^Q21

RSP^K21^RSP_K21	Resposta a pesquisa (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
MSA	Acknowledgement da mensagem
[ERR]	Erro
QAK	Acknowledgement da pesquisa
QPD	Definição dos parâmetros para a pesquisa
[– Início da resposta –
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ NK1 }]	Parentes

QRI	Informação sobre o motivo do sucesso
]	– Fim da resposta –
[DSC]	Apontador para mensagem de continuação

D.8 QBP^Q22 - Pesquisar por Pessoas

QBP^Q22^QBP_Q21	Pesquisa por parâmetro (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
QPD	Definição dos parâmetros para a pesquisa
RCP	Controlo para a resposta
[DSC]	Apontador para mensagem de continuação

D.9 RSP^K22 - Resposta a uma QBP^Q22

RSP^K22^RSP_K21	Resposta a pesquisa (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
MSA	Acknowledgement da mensagem
[ERR]	Erro
QAK	Acknowledgement da pesquisa
QPD	Definição dos parâmetros para a pesquisa
[– Início da resposta –
PID	Identificação do paciente
[PD1]	Dados demográficos extra
[{ NK1 }]	Parentes
QRI	Informação sobre o motivo do sucesso
]	– Fim da resposta –
[DSC]	Apontador para mensagem de continuação

D.10 MFN^M14 - Notificação para um *Master File*

MFN^M14^MFN_M14	Definida localmente (Tipo^Evento^Estrutura)
MSH	Cabeçalho da mensagem

[{ SFT }]	Segmento do <i>software</i>
MFI	Identificação do <i>Master File</i>
[– Início das alterações –
MFE	Identificação do item
...	Qualquer segmento HL7 e/ou segmento Z que ajude a identificar o item
]	– Fim das alterações –

D.11 MFK^M14 - Acknowledgement para a MFN^M14

MFK^M14^MFK_M01 Acknowledgement (Tipo^Evento^Estrutura)

MSH	Cabeçalho da mensagem
[{ SFT }]	Segmento do <i>software</i>
MSA	Acknowledgement da mensagem
[ERR]	Erro
MFI	Identificação do <i>Master File</i>
[MFA]	Acknowledgement do <i>Master File</i>

Apêndice E

Segmentos das mensagens HL7

Legenda:

- O - opcional
- R - requerido
- C - por compatibilidade com versões antigas
- SEQ - número de sequência
- TD - tipo de dados
- OPT - opção de uso
- RP - repetitivo

E.1 MSH - Cabeçalho de uma mensagem

SEQ	TD	OPT	RP	Elemento
1	ST	R		Character separador
2	ST	R		Caracteres codificadores
3	HD	O		Aplicação emissora
4	HD	O		Instituição emissora
5	HD	O		Aplicação receptora
6	HD	O		Instituição receptora
7	TS	R		Data da mensagem

8	ST	O		Segurança
9	MSG	R		Tipo da mensagem
10	ST	R		Identificador para controlo
11	PT	R		Identificador do processo
12	VID	R		Versão
13	NM	O		Número de sequência
14	ST	O		Apontador para a continuação da mensagem
15	ID	O		Tipo de Acknowledgment aceitáveis
16	ID	O		Quando se deve enviar Acknowledgment
17	ID	O		Código do país
18	ID	O	S	Tipo de alfabeto
19	CE	O		Língua da mensagem
20	ID	O		Tipo de alfabeto alternativo
21	EI	O	S	Perfil da mensagem

E.2 PID - Informação sobre o Paciente

SEQ	TD	OPT	RP	Elemento
1	SI	O		PID ID
2	CX	B		ID do Paciente
3	CX	R	S	Lista de identificadores do Paciente
4	CX	B	S	Alternativo ID do Paciente
5	XPN	R	S	Nome
6	XPN	O	S	Nome de solteiro
7	TS	O		Data de nascimento
8	IS	O		Género administrativo
9	XPN	B	S	Alcunhas do Paciente
10	CE	O	S	Raça
11	XAD	O	S	Morada
12	IS	B		Código da Freguesia
13	XTN	O	S	Contacto - Casa
14	XTN	O	S	Contacto - Trabalho
15	CE	O		Língua Nativa
16	CE	O		Estado Civil
17	CE	O		Religião
18	CX	O		Número da conta do Paciente

19	ST	B		Número da Segurança Social
20	DLN	B		Número da Carta de Condução
21	CX	O	S	Identificação da Mãe
22	CE	O	S	Etnia
23	ST	O		Local de nascimento
24	ID	O		Indicador de nascimento múltiplo
25	NM	O		Ordem de nascimento
26	CE	O	S	Cidadania
27	CE	O		Estado militar
28	CE	B		Nacionalidade
29	TS	O		Hora e data da morte
30	ID	O		Indicador de morte
31	IS	O		Indicador de identidade desconhecida
32	IS	O	S	Valor dos dados enviados
33	TS	O		Data do última modificação
34	HD	O		Entidade que fez a última modificação

E.3 PD1 - Informação extra sobre o Paciente

SEQ	TD	OPT	RP	Elemento
1	IS	O	S	Dependente
2	IS	O		Como vive
3	XON	O	S	Entidade médica primária
4	XCN	B	S	Médico de família
5	IS	O		Indicador de estudante
6	IS	O		Dificuldades/Deficiências
7	IS	O		Desejo de vida
8	IS	O		Dador de Órgãos
9	ID	O		Factura separada
10	CX	O	S	Paciente duplicado
11	CE	O		Descrição
12	ID	O		Indicador de protecção ao registo
13	DT	O		Data válida para protecção
14	XON	O	S	Local de adoração
15	CE	O	S	Instruções para os profissionais de saúde

E.4 NK1 - Informação sobre os parentes

SEQ	TD	OPT	RP	Elemento
1	SI	R		NK1 ID
2	XPN	O	S	Nome
3	CE	O		Tipo de Relação
4	XAD	O	S	Morada
5	XTN	O	S	Número de telefone
6	XTN	O	S	Número de telefone - Trabalho
7	CE	O		Papel do Parente
8	DT	O		Data de início
9	DT	O		Data do Fim
10	ST	O		Título profissional
11	JCC	O		Emprego
12	CX	O		ID de funcionário
13	XON	O	S	Nome da organização
14	CE	O		Estado civil
15	IS	O		Género administrativo
16	TS	O		Data de nascimento
17	IS	O	S	Dependência
18	IS	O	S	Mobilidade
19	CE	O	S	Cidadania
20	CE	O		Língua Nativa
21	IS	O		Onde vive
22	CE	O		Tipo d privacidade
23	ID	O		Indicador de proteção
24	IS	O		Indicador de estudante
25	CE	O		Religião
26	XPN	O	S	Nome de solteiro
27	CE	O		Nacionalidade
28	CE	O	S	Etnia
29	CE	O	S	Razão para contacto
30	XPN	O	S	Nome do contacto
31	XTN	O	S	Número de telefone do contacto
32	XAD	O	S	Morada do contacto
33	TS	O	S	Identificadores
34	IS	O		Estado profissional

35	CE	O	S	Raça
36	IS	O		Dificuldades/Deficiências
37	ST	O		Número de Segurança Social
38	ST	O		Local de nascimento
39	IS	O		Indicador VIP

E.5 AL1 - Alergias

SEQ	TD	OPT	RP	Elemento
1	SI	R		ID AL1
2	CE	O		Código da alergia
3	CE	R		Descrição
4	CE	O		Grau da alergia
5	ST	O	S	Código da reacção
6	DT	B		Data da primeira observação

E.6 DB1 - Incapacidades

SEQ	TD	OPT	RP	Elemento
1	SI	R		ID DB1
2	IS	O		Identifica que tem a Incapacidade
3	CX	O	S	Identificação da pessoa
4	ID	O		Identificador da incapacidade
5	DT	O		Data de início
6	DT	O		Data do fim
7	DT	O		Data para voltar ao trabalho
8	DT	O		Data do primeiro dia em que se encontra incapaz

E.7 QPD - Definição dos parâmetros para pesquisa

SEQ	TD	OPT	RP	Elemento
1	CE	R		Nome da pesquisa
2	ST	C		Identificação da pesquisa
3-n	Varies			Parâmetros a pesquisar

Referências

- [1] *Apresentação da Empresa*. Siemens S.A., Sector Healthcare, Portugal, 2009.
- [2] J. Angus, L. Cameron, Miss C.F. Finlayson, R.T. Hopkins, K.B Martin, and J.H. Scott. *The Tayside Master Patient Index*. Outubro 1978.
- [3] João Aradas. Encounter management. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2009.
- [4] D. Baksi. Integrating mpi and deduplication engines: A software architecture roadmap. *Int. J. Med. Inform*, 2008. doi:10.1016/j.ijmedinf.2008.06.002.
- [5] Glenn B. Bell and Anil Sethi. Matching records in a national medical patient index. *Communications of the ACM*, 44(9), September 2001.
- [6] Department of Veterans Affairs. *Master Patient Index/Patient Demographics (MPI/PD), HL7 Interface Specifications*, 2.21 edition, Abril 1999. Revised, Dezembro 2007.
- [7] Department of Veterans Affairs. *Master Patient Index/Patient Demographics (MPI/PD) Vista, Programmer Manual*, 1.0 edition, Abril 1999. Revised, Junho 2008.
- [8] Health Level Seven. *HL7 Messaging Standard Version 2.5*, 2003.
- [9] IHE. The it infrastructure (iti) technical framework. Technical report, IHE, Dezembro 2008. Volume 1 - Integration Profiles.
- [10] IHE. The it infrastructure (iti) technical framework. Technical report, IHE, Dezembro 2008. Volume 2 - Transactions.
- [11] IHE. The it infrastructure technical framework - patient identifier cross-reference hl7 v3 (pixv3) and patient demographic query hl7 v3 (pdqv3). Technical report, IHE, 2008.

- [12] Sun Microsystems. The compliant master patient index based on sun java composite application platform suite (caps). Sun Microsystems Presentacion.
- [13] University Health Network. Hapi website.
<http://hl7api.sourceforge.net/>.
- [14] House of Commons London: The Stationery Office Limited. The electronic patient record. Technical report, House of Commons London, september 2007. Sixth Report of Session 2006–07 Volume I.
- [15] Health Level 7 [Online].
Disponível em: URL:www.hl7.org.
- [16] Hibernate [Online].
Disponível em: URL:<http://www.hibernate.org>.
- [17] IHE [Online].
Disponível em: URL:<http://www.ihe.net/>.
- [18] Java Sig [Online].
Disponível em: URL:<http://aurora.regenstrief.org/javasig>.
- [19] Potix Corporation [Online].
Disponível em: URL:<http://www.zkoss.org/>.
- [20] Sun Microsystems [Online].
Disponível em: URL:<http://www.java.com>.
- [21] Dave Shaver. The hl7 evolution. NeoTool, 2007.
- [22] Nuno Miguel Silva. Master file management. Master's thesis, Faculdade de Ciências da Universidade do Porto, 2009.