

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



eResults - Exploração de Dados Analíticos de Meios Complementares de Diagnóstico e Terapêutica

Pedro Miguel Vilares Jorge Ruas Moreira

Relatório de Projecto

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Maria Teresa Galvão Dias (PhD)

Julho de 2008

eResults - Exploração de Dados Analíticos de Meios Complementares de Diagnóstico e Terapêutica

Pedro Miguel Vilares Jorge Ruas Moreira

Relatório de Projecto

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: José Manuel Magalhães Cruz (PhD)

Arguente: José Carlos Nascimento (MsC)

Vogal: Maria Teresa Galvão Dias (PhD)

15 de Julho de 2008

Resumo

O presente relatório documenta todo o processo de investigação, concepção e desenvolvimento do projecto eResults - Exploração de Dados Analíticos de Meios Complementares de Diagnóstico e Terapêutica, realizado na Companhia Portuguesa de Computadores - Healthcare Solutions (CPCHS).

Tipicamente, uma unidade hospitalar de média/grande dimensão dispõe de um conjunto diversificado de serviços e especialidades clínicas e, no âmbito de cada uma destas, vários meios de análise, podendo estes ser comuns a vários serviços e especialidades. Ao longo do acompanhamento clínico de um doente, quer o episódio tenha uma duração relativamente curta, quer seja um caso crónico, é efectuado um conjunto de exames afim de ser elaborado um diagnóstico correcto. Apesar da divisão existente entre serviços e especialidades, a relevância e utilidade dos resultados produzidos não se restringe a uma área só. Para além deste facto, aquando de cada episódio clínico (visita à unidade hospitalar) importa conhecer o historial clínico do paciente, incluindo exames efectuados anteriormente.

O Projecto foca-se na área de resultados analíticos, a qual corresponde a uma área específica dos Meios Complementares de Diagnóstico e Terapêutica (MCDT), constituídos por um conjunto mais vasto de tipos de resultados. Existe uma série de protocolos clínicos que, quando alimentados com resultados analíticos, definem os aspectos de tratamento do doente. Por este motivo, existe todo o interesse em dotar o pessoal clínico com uma ferramenta de análise mais precisa que potencie o sucesso do acto médico. No domínio dos resultados numéricos, a solução deve ser capaz de gerar gráficos de evolução e de tendência dos vários parâmetros que podem traduzir diagnósticos clínicos.

A solução desenvolvida possui uma arquitectura SOA, recorrendo à implementação de *webservices*. A implementação da solução foi precedida por uma fase de pesquisa sobre tecnologias de suporte à comunicação e ferramentas de apoio ao desenvolvimento. No domínio das tecnologias, foram abordadas ASP.NET Web Services, Web Services Enhancements e Windows Communication Foundation. Para construção dos *webservices* foram testadas as ferramentas Gerador CPCHS e Web Service Software Factory: Modeling Edition. No domínio de acesso a dados foram realizados testes às ferramentas Gerador CPCHS, Repository Factory e LINQ.

Devido à necessidade de apresentações regulares do produto ao cliente, num crescendo do leque de funcionalidades disponíveis, existiu a necessidade de adoptar como referência um modelo de desenvolvimento ágil, mais concretamente o FDD (Feature Driven Development).

À data da elaboração do presente documento, a solução encontra-se já em fase de testes nas instalações do cliente no âmbito de uma auditoria realizada por uma terceira empresa.

Abstract

This report refers to the process of investigation, conception and development of eResults - Exploração de Dados Analíticos de Meios Complementares de Diagnóstico e Terapêutica (Analytical Data Exploration of Complementary Means of Diagnosis and Treatment) that was undertaken at Companhia Portuguesa de Computadores - Healthcare Solutions (CPOCHS).

A healthcare unit, besides its dimension, has a diverse set of services and clinical specialties. For each one, there are different ways of analysing the information that can be common between them. During a patient clinical monitoring, besides its duration in time, there are exams that are made in order to achieve a correct diagnosis. Apart the divisions among services and specialties the importance and utility of the results are not restricted just to one specific area. During a clinical episode it is important to know the patient clinical history, including exams done in the past.

This project focuses on analytic results which is a specific area of Complementary Means of Diagnosis and Treatment, which is composed by a wider set of results. There are different clinical protocols that, based on analytic results, define the patient treatment. Doing so, it is important for those who provide healthcare services to have a specific tool to analyse more deeply and accurately all the available data in order to reach a more successful diagnosis. At the numerical results domain, the tool should be able to draw charts that show the evolution and trends of different parameters that may result in a clinical diagnosis.

The developed solution has a Service Oriented Architecture (SOA) based on web services. The implementation was preceded by a first phase of research on the technologies that may support the communication process. In the domain of technologies the approach was based on ASP.NET Web Services, Web Services Enhancements and Windows Communication Foundation. For the webservices construction process there were tested different tools such as Gerador CPOCHS (a tool developed at CPOCHS) and Web Service Software Factory: Modeling Edition. In order to provide data access, tests were made to Gerador CPOCHS, Repository Factory and LINQ.

Due to the need of periodic presentations to the client, with an increasing set of available features, FDD (an agile software development method) was adopted as reference.

At the moment of this report, the product is being tested and audited at client's facilities by an independent company.

Agradecimentos

Em primeiro lugar, gostaria de agradecer à Companhia Portuguesa de Computadores - Healhtcare Solutions por me ter proporcionado a oportunidade de estar integrado numa empresa dinâmica, onde me foram facultadas todas as condições necessárias ao desenvolvimento do Projecto.

Gostaria também de agradecer a toda a equipa de Investigação & Desenvolvimento Microsoft da CPCHS pelo óptimo ambiente de trabalho, em especial ao Eng. Nuno Ribeiro, o responsável pelo Projecto e que me proporcionou a realização do mesmo e ao Eng. Nuno Mendes, pela orientação e apoio ao longo de todo o processo.

Agradeço profundamente à Professora Maria Teresa Galvão Dias por ter aceite o convite para ser minha orientadora. A sua disponibilidade, apoio e espírito crítico foram fundamentais para ultrapassar as sucessivas barreiras que foram surgindo, mesmo antes do início do Projecto.

A todos os colegas que me acompanharam e ajudaram ao longo da minha passagem pela Faculdade de Engenharia de Universidade do Porto - Daniel Alves, Filipe Garcês, Hélder Silva, Hugo Vara, João Gomes, Joel Campos, Luís Matias e Rui Pinto - um abraço especial.

Um agradecimento especial à minha família, principalmente aos meus pais e irmão pela compreensão e apoio incondicional ao longo de toda a vida.

Obviamente que não podia esquecer todo o apoio e amor demonstrados pela Rita, sem a qual tudo isto não teria o mesmo valor.

Obrigado a todos.

Pedro Ruas Moreira

Conteúdo

1	Introdução	1
1.1	Contexto e Enquadramento	1
1.2	Projecto	2
1.3	Motivação e Objectivos	3
1.4	Estrutura do Documento	4
2	O Papel dos SI/TI na Prestação de Cuidados de Saúde	7
2.1	Linhas de Evolução	7
2.2	Experiências de Utilização	11
2.3	Dificuldades de Implementação	14
2.4	Infra-estruturas Tecnológicas, Tendências Demográficas e Cuidados de Saúde	15
2.5	Conclusões	16
3	Exploração de Dados Analíticos de MCDT	19
3.1	eResults	19
3.2	Resultados Analíticos	20
3.2.1	Resultados Alfanuméricos	22
3.2.2	Microrganismos	22
3.2.3	Gráficos	23
3.3	Filtros Globais	23
3.3.1	Filtragem por Tipo de Documento	24
3.3.2	Filtragem por Serviço	24
4	Revisão Tecnológica	25
4.1	Arquitectura Orientada a Serviços	26
4.2	Tecnologias para a Comunicação	29
4.2.1	ASP.NET Web Services (ASMX)	30
4.2.2	Web Services Enhancements (WSE)	31
4.2.3	Windows Communication Foundation (WCF)	31
4.2.4	Conclusões sobre as Tecnologias para a Comunicação	33
4.3	Ferramentas para a Criação dos <i>Webservices</i>	33
4.3.1	Gerador CPCHS	33
4.3.2	Web Service Software Factory: Modeling Edition	35
4.3.3	Conclusões sobre as Ferramentas para a Criação dos <i>Webservices</i>	39
4.4	Ferramentas para o Acesso a Dados	40
4.4.1	Gerador CPCHS	41

CONTEÚDO

4.4.2	Repository Factory	41
4.4.3	Language-Integrated Query (LINQ)	44
4.4.4	Conclusões sobre as Ferramentas para o Acesso a Dados	46
4.5	Soluções Adoptadas	47
5	Descrição do Processo de Implementação do eResults – EDA de MCDT	49
5.1	Estratégia de Desenvolvimento	49
5.1.1	Características do Processo de Desenvolvimento do eResults	49
5.1.2	Feature Driven Development (FDD)	50
5.1.3	Adaptações realizadas ao Método FDD no âmbito do eResults	52
5.2	Identificação dos Actores	52
5.2.1	Médico	53
5.2.2	Gestor de Sistema	53
5.2.3	Módulo/Aplicação Externa	53
5.3	Casos de Uso	53
5.3.1	Casos de Uso do Módulo Exploração de Dados Analíticos	53
5.3.1.1	Solicitar visualização de uma requisição pelo identifi- cador do documento	53
5.3.1.2	Solicitar visualização de uma requisição pela referência do documento	53
5.3.1.3	Pesquisar requisições segundo critérios de filtragem	53
5.3.1.4	Limpar critérios de pesquisa e dados apresentados	54
5.3.1.5	Ignorar os filtros globais	54
5.3.1.6	Activar os filtros globais	55
5.3.1.7	Ver ficha do doente	55
5.3.1.8	Navegar na árvore de parâmetros	56
5.3.1.9	Seleccionar tipos de exames especiais	56
5.3.1.10	Definir o conjunto de parâmetros visualizados no gráfico de evolução	56
5.3.1.11	Limpar filtros de antibiogramas	56
5.3.1.12	Ver detalhes de um resultado	56
5.3.1.13	Ver detalhes de uma requisição	57
5.3.2	Casos de Uso do Módulo Filtros Globais	57
5.3.2.1	Definir o conjunto de tipos de documento pretendidos	57
5.3.2.2	Definir o conjunto de serviços a excluir	57
5.3.2.3	Guardar as preferências apenas na sessão	57
5.3.2.4	Guardar as preferências de forma persistente	57
5.3.2.5	Repor preferências iniciais	58
5.4	Requisitos da Solução	58
5.4.1	Requisitos Funcionais	58
5.4.2	Requisitos Não-Funcionais	58
5.4.2.1	Requisitos de Qualidade	58
5.4.2.2	Requisitos de Interface Externos	60
5.4.2.3	Requisitos de Ambiente	61
5.4.2.4	Requisitos de Utilizadores	61
5.5	Modelo de Domínio	61
5.5.1	Exploração de Dados Analíticos	61

CONTEÚDO

5.5.2	Filtros Globais	62
5.6	Modelo de Dados	62
5.6.1	Exploração de Dados Analíticos	63
5.6.2	Filtros Globais	66
5.7	Arquitectura Lógica	68
5.8	Arquitectura Física	71
5.9	Módulos dos <i>Webservices</i>	71
5.9.1	Módulo <i>Entities</i>	71
5.9.1.1	Contrato de Dados do Módulo <i>Entities</i>	72
5.9.1.2	Contrato de Serviços do Módulo <i>Entities</i>	72
5.9.2	Módulo <i>Documents</i>	74
5.9.2.1	Contrato de Dados do Módulo <i>Documents</i>	74
5.9.2.2	Contrato de Serviços do Módulo <i>Documents</i>	74
5.9.3	Módulo <i>Activities</i>	75
5.9.3.1	Contrato de Dados do Módulo <i>Activities</i>	75
5.9.3.2	Contrato de Serviços do Módulo <i>Activities</i>	76
5.10	Outros Desenvolvimentos	78
5.10.1	Barras de Pesquisa	79
5.10.2	<i>Provider</i> de Configurações	80
5.10.3	Componente de Selecção do Doente	81
5.11	Interface Gráfica	82
5.11.1	Tabela Global	82
5.11.2	Evolução	82
5.11.3	Antibiogramas	82
5.11.4	Tipos de Documento	83
5.11.5	Serviços	84
5.11.6	Barras de Pesquisa	85
6	Conclusões e Perspectivas de Trabalho Futuro	89
6.1	Satisfação dos Objectivos	89
6.2	Linhas de Orientação para Trabalho Futuro	90
6.2.1	Desempenho das Interrogações	90
6.2.2	Apresentação de Resultados do Tipo Gráfico	90
6.2.3	Evolução de Antibiogramas	91
6.2.4	Barra de Pesquisa Configurável	91
6.3	Conclusões	91
	Referências	96
A	Actores, Pré-Condições, Pós-Condições e Pressupostos dos Casos de Uso	97
A.1	Casos de Uso do Módulo Exploração de Dados Analíticos	97
A.1.1	Solicitar visualização de uma requisição pelo identificador do documento	97
A.1.2	Solicitar visualização de uma requisição pela referência do documento	97
A.1.3	Pesquisar requisições segundo critérios de filtragem	98
A.1.4	Limpar critérios de pesquisa e dados apresentados	98

CONTEÚDO

A.1.5	Ignorar os filtros globais	99
A.1.6	Activar os filtros globais	99
A.1.7	Ver ficha do doente	99
A.1.8	Navegar na árvore de parâmetros	100
A.1.9	Seleccionar tipos de exames especiais	100
A.1.10	Definir o conjunto de parâmetros visualizados no gráfico de evolução	100
A.1.11	Limpar filtros de antibiogramas	101
A.1.12	Ver detalhes de um resultado	101
A.1.13	Ver detalhes de uma requisição	101
A.2	Casos de Uso do Módulo Filtros Globais	102
A.2.1	Definir o conjunto de tipos de documentos pretendidos	102
A.2.2	Definir o conjunto de serviços a excluir	102
A.2.3	Guardar as preferências apenas na sessão	103
A.2.4	Guardar as preferências de forma persistente	103
A.2.5	Repor preferências iniciais	103

Lista de Figuras

3.1	eResults como elemento agregador e distribuidor de MCDT.	20
4.1	Modelo de comunicação entre cliente e servidor.	28
4.2	Arquitectura dos serviços criados com o Service Factory.	36
4.3	Hierarquia da solução gerada com o Service Factory.	38
4.4	Sequência de execução de um serviço.	40
4.5	Exemplo de uma hierarquia possível de uma solução com código gerado pelo Repository Factory.	43
4.6	Abrangência do suporte de fontes de informação por parte do LINQ. . . .	45
4.7	Tecnologias e ferramentas utilizadas no Projecto.	47
5.1	Ciclo de vida de um projecto segundo o modelo FDD.	51
5.2	Diagrama de casos de uso do módulo EDA.	54
5.3	Diagrama de casos de uso do módulo Filtros Globais.	55
5.4	Modelo de domínio do módulo EDA.	62
5.5	Modelo de domínio do módulo Filtros Globais.	63
5.6	Modelo de dados relativo às entidades intervenientes na EDA.	64
5.7	Modelo de dados relativo aos exames e resultados intervenientes na EDA.	65
5.8	Modelo de dados relativo aos filtros globais.	67
5.9	Arquitectura lógica da solução.	68
5.10	Arquitectura física típica da solução.	71
5.11	Diagrama do contrato de dados do módulo <i>Entities</i>	72
5.12	Diagrama do contrato de serviços do módulo <i>Entities</i>	73
5.13	Diagrama do contrato de dados do módulo <i>Documents</i>	75
5.14	Diagrama do contrato de serviços do módulo <i>Documents</i>	76
5.15	Diagrama do contrato de dados do módulo <i>Activities</i>	77
5.16	Diagrama do contrato de serviços do módulo <i>Activities</i>	78
5.17	Integração do componente de selecção do doente no módulo de EDA. . .	81
5.18	Tabela global de apresentação de resultados.	83
5.19	Evolução de resultados numéricos.	84
5.20	Tabela de apresentação de antibiogramas.	85
5.21	Selecção de tipos de documentos a visualizar.	86
5.22	Selecção de serviços a visualizar.	87
5.23	Barras de pesquisa.	87

LISTA DE FIGURAS

Abreviaturas e Símbolos

AJAX	Asynchronous Javascript And XML
ASMX	ASP.NET Web Services
CPCHS	Companhia Portuguesa de Computadores - Healthcare Solutions, S.A.
EDA	Exploração de Dados Analíticos
ES	.NET Enterprise Services
E.U.A.	Estados Unidos da América
FDD	Feature Driven Development
FEUP	Faculdade de Engenharia da Universidade do Porto
HL7	Health Level 7
HTTP	HyperText Transfer Protocol
LINQ	Language-Integrated Query
MCDT	Meios Complementares de Diagnóstico e Terapêutica
MIEIC	Mestrado Integrado em Engenharia Informática e Computação
MSMQ	Microsoft Message Queuing
OOP	Object-Oriented Programming
PDF	Portable Document Format
PME	Pequenas e Médias Empresas
RAM	Random Access Memory
SGBD	Sistema de Gestão de Bases de Dados
SI	Sistemas de Informação
SIH	Sistemas de Informação Hospitalares
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TI	Tecnologias de Informação
UDDI	Universal Discovery Description and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
WCF	Windows Communication Foundation
WSDL	Web Service Definition Language
WSE	Web Services Enhancements
XML	eXtensible Markup Language

ABREVIATURAS

Capítulo 1

Introdução

O presente documento tem como principal objectivo descrever o projecto eResults - uma aplicação para a Exploração de Dados Analíticos (EDA) provenientes de Meios Complementares de Diagnóstico e Terapêutica (MCDT). O Projecto foi realizado no âmbito do Mestrado Integrado em Engenharia Informática e Computação (MIEIC) da Faculdade de Engenharia da Universidade do Porto (FEUP), nas instalações da Companhia Portuguesa de Computadores – Healthcare Solutions, S.A. (CPOCHS).

1.1 Contexto e Enquadramento

É inegável a contribuição positiva das Tecnologias de Informação (TI) para a sociedade mundial onde se enquadra, obviamente, a área da Saúde. O aumento da esperança de vida nas várias sociedades por todo o mundo (de uma forma geral) deve-se aos progressos médicos verificados nos últimos anos mas também ao precioso auxílio das TI. Estes dois factores são indissociáveis pois contribuem mutuamente para a evolução dos mesmos. Um exemplo característico desta contribuição mútua é o facto de novos processos médicos originarem novos tipos de dados que são suportados pelas TI, capacidade de suporte essa, cada vez maior e flexível. Por sua vez, os novos tipos de dados proporcionam novas formas de análise do caso clínico, eventualmente originando uma nova terapêutica.

Dada a sensibilidade do tema, a Saúde, a abrangência de Sistemas de Informação (SI) e TI não deve ser encarada como uma mera contribuição tecnológica pois a vertente social é deveras importante e com um impacto na opinião pública bastante superior. Questões como ética, segurança de dados e disponibilidade dos sistemas assumem um nível superior no que a exigência diz respeito.

Tipicamente, uma unidade hospitalar de média/grande dimensão dispõe de um conjunto diversificado de serviços e especialidades clínicas e, no âmbito de cada uma destas,

vários meios de análise, podendo estes ser comuns a vários serviços e especialidades. Ao longo do acompanhamento clínico de um doente, quer o episódio tenha uma duração relativamente curta, quer seja um caso crónico, é efectuado um conjunto de exames afim de ser elaborado um diagnóstico correcto. Apesar da divisão existente entre serviços e especialidades, a relevância e utilidade dos resultados produzidos não se restringe a uma área só. Para além deste facto, aquando de cada episódio clínico (visita à unidade hospitalar) importa conhecer o historial clínico do paciente, incluindo exames efectuados anteriormente.

1.2 Projecto

O presente Projecto foca-se na área de resultados analíticos, a qual corresponde a uma área específica dos MCDT, constituídos por um conjunto mais vasto de tipos de resultados.

Os MCDT são recursos extremamente importantes no auxílio à tomada de decisão dos médicos. Constituem um vasto conjunto de resultados provenientes de várias áreas e especialidades. Como resultado dos MCDT existem vários tipos de elementos, entre outros, radiografias, ecografias, intervenções cirúrgicas, relatórios e análises de patologia clínica. Os elementos referidos podem ser produzidos em diferentes formatos de ficheiros, nomeadamente, imagens, vídeos, resultados analíticos provenientes de laboratórios e diversos outros tipos de ficheiros.

Como é fácil depreender, o conjunto de MCDT produzido diariamente numa unidade hospitalar assume dimensões consideráveis e cuja gestão eficaz e eficiente é humanamente impossível, quer pela sua diversidade, quer pela sua quantidade. Aliado a este facto, existe a necessidade de partilha dos mesmos em tempo útil. Atendendo ao facto de que a partilha poderá ser necessária em ambiente "multilocal", quer pela distribuição geográfica da unidade hospitalar, quer pela partilha de informação entre unidades, torna-se ainda mais complicado alcançar este objectivo.

Tendo em conta esta realidade, revela-se de extrema importância o desenvolvimento de uma aplicação que colmate estas lacunas. Para alcançar tal objectivo, a aplicação deve ser capaz de, num só local, concentrar toda a informação acerca dos MCDT produzidos para cada doente, para cada serviço e especialidade e até para cada instituição. Deve igualmente ser capaz de disponibilizar a informação pretendida por um determinado utilizador de uma forma organizada que permita a sua fácil identificação e utilização, tornando-a útil. Tendo em conta a importância dos dados, pois estão em jogo vidas humanas, características como a disponibilidade, segurança, fiabilidade e eficiência são de importância fulcral. No âmbito da interacção com a aplicação, é de todo vantajoso que esta proporcione uma interface eficiente, organizada e com uma experiência de utilização agradável.

O projecto eResults foi desenvolvido na Companhia Portuguesa de Computadores – Healthcare Solutions, S.A., uma empresa nacional, especializada em soluções de TI para a área da Saúde [[CPdCHSb](#)].

A CPCHS surgiu no panorama nacional, como empresa autónoma, no início de 2001. Desde o seu nascimento até ao presente registou um forte crescimento. A sua adesão (em Maio de 2007) à rede de PME's Inovadoras, na COTEC, representa o reconhecimento do seu estatuto de empresa de referência em Investigação & Desenvolvimento no domínio das TI no domínio da Saúde [[CPdCHSa](#)].

O projecto eResults, em que a solução de EDA de MCDT se insere, foi desenvolvido tendo em conta a realidade vivida no Centro Hospitalar de Vila Nova de Gaia/Espinho, constituído pelo Hospital Eduardo dos Santos Silva (Unidade I) e pelo Hospital Distrital de Vila Nova de Gaia (Unidade II) [[ARdSdN](#)]. Este centro hospitalar possui inúmeras fontes de elementos e, conseqüentemente, o circuito que estes seguem torna-se difícil de automatizar, sendo muitas vezes, um processo manual ou semi-automático. Tendo em conta toda esta diversidade de fontes, era necessário desenvolver uma solução integradora que conferisse organização e flexibilidade à distribuição e consulta de informação clínica.

Existia já uma solução desenvolvida com esse intuito, no entanto, a interface era muito básica e a interacção com a mesma não constituía uma experiência de utilização agradável. O número de tipos de ficheiro suportados era reduzido e o meio de visualização dos mesmos muito rudimentar. Ao nível dos resultados analíticos não havia qualquer tipo de possibilidade de interacção com os mesmos, ou seja, não era possível explorá-los de uma forma rica. A única forma de visualizar os resultados analíticos era através da consulta do ficheiro PDF¹ que continha os dados de cada requisição. Devido a todas estas lacunas, quer ao nível da interface com o utilizador, quer ao nível de funcionalidades, o desempenho da solução estava aquém das necessidades actuais pelo que foi tomada a decisão do desenvolvimento de uma nova aplicação.

1.3 Motivação e Objectivos

Algumas decisões clínicas da área de cardiologia e nefrologia (hemodiálise) são tomadas tendo em conta os resultados analíticos das análises clínicas. Existe um conjunto de protocolos clínicos que, quando alimentados com resultados analíticos, definem os aspectos de tratamento do doente. As possibilidades de interacções entre parâmetros de análises são várias, pelo que uma ferramenta de análise mais precisa é indispensável para o sucesso do acto médico. Assim, este Projecto deve ser pensado de forma a servir de base à implementação de um sistema de apoio à decisão, embora tal não faça parte do mesmo.

Podemos definir os seguintes objectivos específicos:

¹<http://ncsu.edu/it/access/tutorials/pdf/>, consultado pela última vez em 3 de Julho de 2008.

- A definição de um modelo de dados relativamente estável para o armazém de resultados analíticos, tendo em conta características como o desempenho das interrogações, o facto de poderem existir diferentes aplicações operacionais a produzir resultados, assim como o facto de este ser comum a diferentes instituições.
- O desenvolvimento de todos os serviços necessários para a alimentação da interface gráfica.
- Deve ser preparado o desenvolvimento futuro de um módulo de pedidos.
- A possibilidade de gerar gráficos de evolução dos valores de análises, de gerar gráficos de tendência, isto é, gráficos construídos à custa da composição dos resultados de diferentes análises que, quando vistos como um todo, podem traduzir diagnósticos clínicos (utilizado especialmente em cardiologia) e permitir a filtragem por tipos de análises.
- Considerar a necessidade de comunicação entre a solução e as aplicações externas fornecedoras de dados.

O Projecto insere-se numa solução mais global (o eResults), motivo pelo qual as tarefas realizadas não se restringiram ao círculo da EDA. Considerando a integração um factor fundamental, foi necessário realizar desenvolvimentos não directamente relacionados com o objectivo específico do Projecto e também partilhados por outros módulos, nomeadamente um módulo de filtros globais que afecta não só a EDA analíticos mas também o módulo de pesquisa documental (módulo de pesquisa mais alargada de MCDT que engloba outros tipos para além de resultados analíticos) e as barras de pesquisa (quer no âmbito de documentos, quer no âmbito de resultados analíticos).

Dada a sensibilidade dos resultados, foi imperativo que preocupações relacionadas com segurança e precisão de dados estivessem presentes ao longo de todo o processo de arquitectura e desenvolvimento.

A análise dos modelos conceptual e de dados de outras soluções foi uma preocupação constante, pois são estas que alimentam a solução desenvolvida. Existe ainda a necessidade de aplicações externas poderem apresentar resultados recorrendo ao módulo de EDA, pelo que foi necessário implementar formas flexíveis e fiáveis de permitir a chamada do módulo a partir do exterior.

1.4 Estrutura do Documento

Para além da presente introdução, este documento é constituído por mais cinco capítulos.

O Capítulo 2 apresenta algumas considerações e referências no âmbito do papel dos sistemas e tecnologias de informação na prestação de cuidados de saúde. Apresenta-se

uma perspectiva sociológica e histórica deste tipo de sistemas, de forma a interiorizar qual o caminho futuro, bem como algumas questões chave, tais como experiência de utilização, dificuldades de implementação e infra-estruturas tecnológicas.

O Capítulo 3 é dedicado a uma descrição mais detalhada do problema onde são identificados os objectivos gerais, os requisitos e algumas questões relacionadas com o desenvolvimento do sistema.

O Capítulo 4 é constituído por um conjunto de considerações acerca de tecnologias e ferramentas estudadas, principalmente o conjunto de alternativas existentes sobre a forma de acesso a dados.

O Capítulo 5 é exclusivamente dedicado à descrição detalhada de todo o processo de arquitectura e desenvolvimento da solução nas suas várias vertentes.

Por fim, o Capítulo 6, é dedicado à análise dos resultados obtidos, às perspectivas de trabalho futuro e a outras conclusões.

Introdução

Capítulo 2

O Papel dos SI/TI na Prestação de Cuidados de Saúde

É inquestionável o facto de que o objectivo principal de qualquer actividade médica seja a prestação do melhor serviço possível a todos os pacientes [HS07]. O ideal seria que este objectivo fosse plenamente alcançado, em qualquer parte do mundo. Contudo, devido a vários factores, provenientes das mais variadas áreas (social, económica, etc.), não é possível atingir sempre este objectivo. Neste contexto, a contribuição positiva de qualquer recurso é no mínimo indispensável. Sob esta perspectiva, as TI e os SI são mais um recurso, mais um meio, para atingir o fim.

Neste capítulo pretende-se descrever o papel actual dos SI/TI na prestação de cuidados de saúde. Sempre que possível, procurar-se-á introduzir a realidade portuguesa, pois é uma problemática que está, actualmente, a ser alvo de alguma mediatização devido à implementação do Plano Tecnológico [dCdEdLedPT].

2.1 Linhas de Evolução

Em 1984, Peter Reichertz realizou uma revisão do passado, constatou o presente e apontou algumas orientações para o futuro dos Sistemas de Informação Hospitalares (SIH). Desde essa época, tanto a medicina como a informática sofreram grandes progressos, o que requer uma redefinição dos caminhos traçados assim como daqueles pelos quais os SIH devem seguir. Em 2006 Reinhold Haux realizou uma nova revisão tendo em conta esses mesmos espaços temporais [Hau06].

Nesta perspectiva de espaços temporais, interessa reflectir acerca do passado para que, em conjunto com o conhecimento do presente, se possa definir a melhor estratégia para o futuro, dentro dos limites impostos pela própria evolução.

Haux identificou algumas linhas fundamentais de desenvolvimento segundo as quais os SIH têm evoluído, como por exemplo, a substituição do papel por formatos electrónicos, a evolução das arquitecturas dos SI, a atitude dos pacientes, novos tipos de dados, entre outras. Contudo, é necessário ter em atenção a assimetria da evolução tecnológica nas diferentes regiões do globo. Por este motivo, estas linhas devem ser encaradas como uma "evolução natural" e não como o estado global deste tipo de tecnologias no nosso mundo, de uma forma generalizada.

A evolução do suporte da informação, do papel para o formato digital, é talvez a face mais visível de todo um processo evolutivo. Tal como praticamente todas as evoluções a que assistimos e/ou sofremos, esta tem igualmente vantagens e desvantagens. Como principal desvantagem destaca-se a complexidade inerente a sistemas destas dimensões. No entanto, as vantagens ultrapassam largamente as desvantagens. O conjunto de funcionalidades disponibilizadas pelos SIH na manipulação dos registos de saúde pessoais e do conhecimento médico permite que os profissionais se concentrem no acto médico, reservando para os SI/TI a tarefa de armazenamento e manipulação de dados. O aumento das capacidades de armazenamento e processamento (com tendência para continuar a aumentar) permite a análise de mais casos, que tem como consequência a extracção de mais e melhor informação que poderá servir de base para a área da investigação. É importante realçar que esta evolução não se deve apenas à vertente tecnológica, mas também à evolução dos procedimentos terapêuticos.

Importa referir que vivemos uma fase de transição e, portanto, é comum assistirmos a uma coexistência (por vezes precária) dos dois "sistemas". A coexistência acarreta alguns "senões", tais como mais custos ou as dificuldades inerentes à manutenção dos "sistemas" em paralelo. Actualmente, ainda pode ser encarado como um mal necessário, na medida em que, a validade legal de documentos digitais ainda não se encontra amplamente disseminada e a sua aceitação não é generalizada.

Para acompanhar a evolução no domínio das comunicações, as próprias arquitecturas dos SI tiveram de evoluir de locais para globais. Se no início assistíamos a SI específicos para determinadas actividades ou locais (tal como um laboratório), com as suas desvantagens naturais, como por exemplo, dificuldades de integração e coerência dados, hoje é perfeitamente realista a implementação de um sistema que abranja toda uma unidade hospitalar ou um conjunto das mesmas. Podemos ir ainda mais além, pois é viável idealizar e implementar um sistema a nível regional ou mesmo nacional. Esta nova abrangência permite uma gestão mais eficaz dos recursos e informação, o que pode conduzir a uma redução dos custos inerentes. Contudo, existe uma vantagem ainda mais nobre, a capacidade de centrar os cuidados médicos no paciente (que também pode ser visto como um consumidor) e não na entidade prestadora dos cuidados de saúde ou nos seus profissionais.

A própria utilização dos dados que "circulam" nos SIH sofreu uma evolução quanto à sua abrangência. Assistimos a uma evolução da utilização dos mesmos para fins exclu-

sivamente de cuidados de saúde do paciente e administrativos para a possibilidade que existe actualmente de usar este recurso para investigação e planeamento de cuidados de saúde. Poderá ainda ser um precioso auxiliar na monitorização e pesquisa de epidemias.

Hoje em dia, os pacientes não se limitam a uma atitude passiva de consulta dos profissionais de saúde mas também assumem uma atitude pró-activa, procurando informação. Tendo em conta esta nova atitude dos pacientes, surge um novo grupo de utilizadores dos SIH. A discrepância de conhecimentos e os seus objectivos criam mais um desafio no desenho e construção de SIH. Num mesmo sistema, eventualmente com "diversas caras", este tem de ser capaz de ir ao encontro das capacidades e objectivos deste novo grupo. Contudo é impossível centrar a análise única e exclusivamente neste grupo. Para que o sistema alcance o sucesso é imperativo que este possua um nível de polivalência relativamente elevado.

Inevitavelmente, a implementação de sistemas desta envergadura está condicionada por questões técnicas, porém, a constante evolução tecnológica permitiu (e continua a permitir cada vez mais) dedicar a atenção a questões de foro estratégico, quer na vertente administrativa, quer na vertente clínica. Para tal evolução contribuem factores, como por exemplo, problemas organizacionais, questões sociais e gestão da mudança. Actualmente, no nosso país, é possível identificar claramente esta tendência. Os hospitais de gestão privada são uma das faces da introdução de conceitos como planos de gestão estratégica da informação ou alinhamento estratégico ou ainda processos de negócio. Esta nova focalização pode ainda ser vista como uma "estratificação" da gestão já utilizada em outras áreas económicas, as vertentes estratégica, táctica e operacional. Este desafio assume ainda mais importância se for encarado a nível regional ou nacional, e não só ao nível da unidade de saúde. Sob esta perspectiva "empresarial" das unidades de saúde, é inquestionável o auxílio dos SI/TI. Poderão ser aplicados conceitos de *business intelligence* na vertente administrativa, proporcionando redução de custos provenientes de, por exemplo, optimização de recursos e melhor controlo de fluxos financeiros. Na vertente clínica, de forma análoga, é possível aplicar técnicas de *data mining*, podendo ser obtidas tendências epidémicas (ao nível da propagação geográfica, intensidade do surto, etc.) ou dados preciosos para o auxílio na investigação clínica. Contudo, na vertente clínica, a utilização de SI/TI é mais crítica, na medida em que é no período de atendimento que o factor tempo é mais determinante.

A evolução médica conduziu ao aparecimento de novos tipos de dados, dados esses que passaram a ser suportados pelos SI/TI (graças à evolução tecnológica). O caminho seguido por esta linha de evolução prolongar-se-á para o futuro pois os tipos de dados a serem considerados aumentam de forma contínua.

Um dos campos mais beneficiados com a introdução de novas tecnologias é o da monitorização (por exemplo, de sinais vitais). Aliado ao óbvio impacto clínico, pois permite a obtenção de dados contínuos sem intervenção manual e não apenas de pon-

tos discretos no tempo, existe a vertente social. A vertente social é essencial para o bem-estar de qualquer ser humano e, desta forma, sai beneficiada na medida em que é perfeitamente possível realizar as actividades do dia-a-dia, praticamente sem restrições, enquanto são monitorizados dados clínicos, eventualmente com o envio em tempo real. A alteração do meio de aquisição de dados pode permitir uma reorganização dos cuidados de saúde e as próprias infra-estruturas das unidades de saúde. Os próprios meios de cuidados e tratamentos podem ser adaptados a uma sociedade mais envelhecida, pois é essa a tendência das sociedades dos denominados países desenvolvidos. Isto assume particular importância se tivermos em conta que é nas idades mais avançadas que a mobilidade é menor e, consequentemente, as deslocações toenam-se mais penosas e dispendiosas. Está fora de qualquer dúvida o facto de que o estilo de vida mudou radicalmente desde há 20 anos. Contudo, temos de considerar que não só o estilo de vida dos pacientes se alterou, mas também as próprias unidades de saúde e, consequentemente, as condições de trabalho dos seus profissionais. Um exemplo cabal é a própria disseminação de um conjunto enorme de tecnologias de apoio ao diagnóstico e tratamento.

A evolução nos domínios anteriormente referidos, por certo continuará e, eventualmente, a um ritmo ainda mais acelerado, no entanto as áreas que se referem de seguida assumem particular destaque devido ao seu potencial desenvolvimento.

Cada vez mais verificar-se-á a necessidade de partilha de estratégias ao nível institucional. A vertente estratégica é um componente de importância vital para o processo contínuo de manter e melhorar os SIH, tendo como objectivo final a melhoria dos cuidados de saúde. Por este motivo, as estratégias definidas ao nível institucional devem estar perfeitamente integradas com estratégias regionais, nacionais e internacionais. Contudo, esta partilha levanta um sério problema que nos acompanha no quotidiano, a privacidade. Deve existir um perfeito "balanceamento" entre a preservação da privacidade e os recursos informáticos que nos acompanham por toda a parte.

De forma a suportar todos estes novos desafios, é imprescindível explorar e evoluir para novos estilos de arquitectura, no que concerne aos SIH. As arquitecturas tradicionais, centradas na instituição, provavelmente não serão adequadas para implementar SIH de âmbito inter-institucional ou regional [Cot07]. Logo à partida, é possível apontar alguns desafios que se colocam neste domínio. A fiabilidade dos serviços, tornando-os menos redundantes, mais funcionais, imunes a falhas de informação quer ao nível local, regional ou mesmo global. A mudança na arquitectura dos SIH, tornando-se mais aberta permite, ou pelo menos facilita, o alcançar de um outro objectivo, SIH centrados no paciente. Este factor assume particular importância se tivermos em conta a já referida tendência de envelhecimento da nossa sociedade.

Um outro aspecto a ter em conta é a própria preparação e formação dos profissionais de saúde na utilização de sistemas informáticos no seu quotidiano. O nível actual de informatização e automatização de tarefas numa unidade hospitalar é incomparável ao

que se vivia há apenas umas décadas atrás. É fundamental que um profissional de saúde esteja altamente preparado para lidar com SI/TI no seu dia-a-dia. A formação nesta área é fundamental, talvez mais fundamental que alguns temas abordados ao longo de uma formação na área médica. Por este motivo, é de todo o interesse incluir no programa de formação médica esta temática, de forma a proporcionar aos futuros profissionais de saúde conhecimentos que lhes serão preciosos para o exercício da actividade médica. Existe também uma outra vertente no plano da formação que se refere a formação extraordinária para facultar aos profissionais de saúde já em exercício a mesma oportunidade. Contudo, em qualquer das vertentes acima referidas, a formação não pode estagnar neste ponto, até pela efemeridade do conhecimento tecnológico, pois este evolui a um ritmo alucinante.

Nos próximos tempos intensificar-se-á a necessidade de investigação na área da "informática da saúde" e biomédica. Os SIH estão numa fase de rápido desenvolvimento com questões pendentes em vários domínios. Tendo em conta a evolução da nossa sociedade, estes novos sistemas têm todo o potencial (e algumas provas dadas) para assumir uma posição de destaque na reformulação dos sistemas de saúde. O desenvolvimento e investigação poderá verificar-se segundo grandes eixos, nomeadamente, arquitecturas de SIH inter-institucionais adequadas, métodos para gestão estratégica, modelação e avaliação de SIH, métodos para análise de dados clínicos ("data mining clínico") baseados nas novas arquitecturas dos sistemas de informação e registos electrónicos de pacientes, considerando a variedade de tipos de dados a incluir.

Tal como muitas outras vertentes do nosso quotidiano, os cuidados de saúde têm de se adaptar à própria sociedade, sociedade essa cada vez mais global e, por conseguinte, com outros objectivos. A dimensão e o raio de acção da nossa sociedade exigem que os SIH sejam cada vez mais abrangentes, com capacidade de manter coerência de dados e cada vez mais centrados no paciente. "Vinte anos após a perspectiva de Reichertz, podemos redefinir que os objectivos dos SIH são contribuir para cuidados de saúde de alta qualidade e eficientes, agora para pacientes e consumidores e para a investigação médica" [Hau06]. A capacidade crescente do processamento de vários tipos de dados inerente à evolução tecnológica pode reservar-nos inúmeras possibilidades de melhoria e até mesmo abrir-nos novas portas.

2.2 Experiências de Utilização

Independentemente do domínio de uma solução, a experiência de utilização é uma característica determinante do sucesso ou insucesso da mesma. Na área da Saúde esta situação tende a tornar-se mais crítica dada a sua sensibilidade. Comparativamente a outras áreas, a evolução verificada é mais lenta, contudo esta realidade não se deve a questões de ordem tecnológica mas sim de interface entre os utilizadores e o sistema. O valor de um sistema mede-se pela sua capacidade de informar correctamente o utilizador (permitindo

a melhoria de desempenho) e não pelo nível de automatização (redução ou eliminação da intervenção humana) conseguido. Por este motivo, alguns sistemas altamente avançados do ponto de vista tecnológico acabam por ser abandonados devido ao desconforto que os seus utilizadores sentem ao utilizá-lo. A relação entre inovação tecnológica e utilidade não é directamente proporcional.

Atendendo à evolução dos SIH, é possível constatar que o seu leque de utilizadores-chave tende a aumentar pois, inicialmente, restringiam-se a pequenos departamentos ou serviços onde o conjunto de utilizadores era bastante restrito e específico. Hoje em dia é possível considerar vários actores, quer na vertente administrativa quer na vertente clínica.

De uma forma geral, o pessoal médico revela-se muito céptico relativamente aos sistemas de informação e ao seu real contributo para um eficiente acto médico [MCS⁺07]. Um dos principais receios é que o acto médico se torne demasiado dependente da disponibilidade e acesso aos SI. Do ponto de vista médico, é altamente frustrante a impossibilidade de acção ou, pelo menos, o condicionamento da mesma devido à indisponibilidade de um sistema. O armazenamento de informação clínica apresenta duas características diametralmente opostas no que concerne aos seus efeitos. Se, por um lado, o facto de a informação estar disponível electronicamente proporciona um meio expedito de a obter, por outro lado, a elevada dependência deste meio de armazenamento pode impossibilitar o pessoal médico de aceder a informação clínica vital para uma acção rápida e eficiente.

Também as capacidades técnicas dos sistemas são uma causa de preocupação para o pessoal médico. Características do software e do hardware são alvo das suas dúvidas, como por exemplo, a acessibilidade e o desenho do sistema. Se tivermos em conta o factor tempo numa situação de emergência, este assume um papel preponderante. Existindo a necessidade de acesso a informação nestes casos, o sistema deve ser capaz de proporcionar um meio extremamente rápido e explícito de o fazer. Esta diversidade de âmbitos de utilização e rapidez necessária na disponibilização de informação requer dos sistemas uma grande flexibilidade e desempenho. De uma forma geral, o pessoal médico, possui bastantes reservas quanto às capacidades dos sistemas neste domínio.

Uma outra situação recorrentemente abordada é a possibilidade de introdução de novos erros de informação devido ao uso de SI/TI no acto médico. É inquestionável que a utilização de tais sistemas elimina um conjunto de erros normalmente cometidos, uns com mais repercussões que os outros, contudo, é legítimo questionar a possibilidade de introdução de novos tipos de erros. Os sistemas são desenvolvidos por humanos, portanto, existe a possibilidade de estes conterem erros ou de não preverem determinadas situações que fazem com que se tornem instáveis. Outra fonte de possíveis erros é a própria experiência de utilização do pessoal médico pois é perfeitamente normal que utilizadores com menos experiência originem mais erros decorrentes da utilização do sistema.

Independentemente da qualidade final de um sistema, a sua utilização exige que seja dispendido algum tempo no decorrer da interacção. Esta situação provoca problemas de

aceitação por parte do pessoal médico na medida em está em causa a eficiência do seu trabalho. Idealmente, o período de aprendizagem de interacção com um sistema seria nulo, no entanto, a necessidade da sua realização verifica-se regularmente. Não são raras as vezes em que planos de formação no âmbito de SI/TI não são incluídos na distribuição de serviço sendo, muitas vezes, realizados em tempo extraordinário. Esta situação origina algum desconforto laboral, sendo o sistema o principal visado, resultando em alguma relutância na sua utilização.

As gerações mais recentes de pessoal médico já possuem um nível de experiência de utilização de SI/TI bastante mais elevado que as gerações precedentes. Os estudantes de medicina já vêm os SI/TI como um elemento que, inevitavelmente, integrará a sua prática clínica. Paralelamente, a relutância do pessoal médico relativamente aos SI/TI tenderá a diminuir na medida em que o contacto com estes sistemas aumenta.

A tendência de evolução para sistemas centrados no paciente conduz inevitavelmente a uma maior interacção entre este tipo de utilizadores e o sistema [DAS⁺08]. Estes novos actores colocam desafios interessantes no desenho dos sistemas pois a multiplicidade de interfaces necessária atinge um patamar mais elevado. A questão não é apenas tornar o paciente o conceito-base da lógica de negócio mas também dotar o sistema da capacidade de adaptar os conteúdos. No caso particular de um cidadão sem quaisquer conhecimentos médicos, caso a informação seja apresentada de uma forma simples e esclarecedora, poderá favorecer a tomada de decisões no que diz respeito à sua saúde. Dotar o cidadão comum de ferramentas para orientar a sua própria saúde, de uma forma fundamentada, pode ser o primeiro passo para uma redução dos custos associados. Esta situação verifica-se pois os recursos necessários para uma estratégia preventiva são bastante menores que os dispendidos em processos terapêuticos. A informação apresentada não necessita de ser detalhada pois, muitas vezes, seria impossível a sua interpretação ou, num cenário ainda pior, mal interpretada (o que poderia causar a tomada de decisões erradas que pusessem em risco a vida do utilizador).

Nos últimos tempos têm surgido cada vez mais serviços de gestão de informação clínica pessoal. Não considerando eventuais interesses comerciais, o conceito subjacente a este tipo de serviços apresenta-se como um primeiro passo para uma maior responsabilização do cidadão pela sua própria saúde. A hipótese de um utilizador controlar a evolução de determinados parâmetros como o peso, tensão arterial ou o nível de glicemia, eventualmente com a disponibilização de valores de referência, pode tornar-se um excelente meio preventivo. Se existir uma integração entre estes sistemas e os SIH, a possibilidade de troca de informação beneficia não só os cidadãos mas também o próprio sistema de Saúde (por exemplo, a diminuição de custos associados aos cuidados de saúde e a oferta de sistemas mais eficazes e úteis). A disponibilização de informação dos SIH para os sistemas pessoais permite que o utilizador, por exemplo, tenha acesso aos resultados de exames que realize, ou até a orientação médica em diagnósticos de menor importância.

Existem também benefícios decorrentes do fluxo de informação no sentido inverso. A identificação antecipada de eventuais patologias é apenas um dos benefícios directos deste tipo de integração.

A interacção com os sistemas não deve ser encarada numa perspectiva meramente aplicacional mas também ao nível de equipamentos. É cada vez mais comum a realização de exames sem a presença física do paciente nas instalações hospitalares, como por exemplo a monitorização cardíaca ao longo de um dia ou ainda a monitorização de dados vitais de um paciente com uma patologia mais duradoura. É essencial que os equipamentos em causa proporcionem uma interface simples e eficiente independentemente dos conhecimentos técnicos do paciente.

2.3 Dificuldades de Implementação

As dificuldades de implementação podem ser consideradas em duas perspectivas, a decisão de adopção de um SIH ou numa fase posterior, a implementação propriamente dita na unidade hospitalar.

Na primeira perspectiva, surgem como principais barreiras questões de ordem económica e técnica. Estudos realizados nos E.U.A. demonstram que a lacuna ao nível do suporte financeiro para a adopção de SI/TI é a principal barreira. Seguem-se factores como a percepção de que os produtores de software não fornecem soluções adequadas às suas necessidades, a inexistência de um plano estratégico para a implementação da aplicação e a dificuldade em recrutar pessoal habilitado no domínio das TI [And07].

O retorno do investimento neste tipo de sistemas é bastante incerto pois apresentam elevados custos iniciais e custos regulares de manutenção. Os benefícios financeiros decorrentes da implementação de SIH são incertos, motivo pelo qual a aquisição deste tipo de software, sem o apoio financeiro externo, representa um risco elevado que muitas unidades hospitalares não estão dispostas a correr.

Uma segunda barreira identificada é precisamente a complexidade elevada que alguns destes sistemas apresentam. A aposta em sistemas demasiadamente complexos acaba por se traduzir em pouca utilidade para o cliente, pois os benefícios que o sistema proporciona não compensam o tempo e esforço exigidos para a aprendizagem da nova ferramenta.

Indissociável de toda esta questão é a privacidade dos pacientes. O cepticismo quanto à segurança dos dados pessoais que constam nos SIH constitui ainda uma importante barreira à sua adopção. Se tomarmos em conta que existe uma tendência para a implementação de redes sem fios e SIH *Web-based*, as dúvidas acentuam-se pois a possibilidade de acesso ao mesmo por pessoas não autorizadas é maior. Obviamente que estas desconfianças são manipuladas pelo desconhecimento técnico, contudo é inegável que é uma característica fundamental dos SIH o nível de segurança em todo o tratamento de dados.

No momento de implementação de um SI, independentemente da área a que se destina, é comum encontrarem-se algumas dificuldades inerentes a processos deste género [vSR⁺07]. Verifica-se que, obviamente, existem desafios técnicos mas os mais críticos são mesmo de natureza organizacional. A dimensão da organização bem como a sua capacidade de adaptação e o nível de cultura da mudança existente assumem-se como factores preponderantes no sucesso do projecto.

A teoria de implementação de TI nos cuidados de saúde é muito recente e limitada por várias razões, nomeadamente a complexidade dos sistemas de saúde, o ritmo da evolução tecnológica e o número reduzido de estudos realizados. Apesar desta realidade, é possível identificar algumas características que podem facilitar e catalisar todo o complexo processo:

- a escolha do sistema a implementar deve ter em conta as necessidades de todos os visados de forma a que o processo de implementação seja uma tarefa de equipa;
- a decisão de implementação depende da gestão de topo mas é vital que todos contribuam positivamente;
- o sistema escolhido deve apresentar um nível de maturidade razoável pois, potencialmente, proporcionará mais benefícios aos seus utilizadores;
- o sentimento de utilidade para com o sistema promove a aceitação do mesmo, caso contrário a sua implementação será muito difícil, ou até mesmo impossível;
- a decisão de implementação de um sistema deve ser considerada com bastante cuidado e após estudado o seu impacto, no entanto, tomada a decisão de avançar, a implementação deve ser bastante rápida e directa;
- o sistema deve ser bastante fácil de modificar e adaptar pois este tipo de sistemas não pode estar sujeito a longos períodos de indisponibilidade;
- a interface proporcionada deve ser intuitiva, requerendo pouco ou nenhuma formação para que o utilizador possa interagir eficientemente com o sistema.

A especificidade da área em causa exige a realização de mais estudos de forma a perceber as características únicas dos SIH num processo de implementação.

2.4 Infra-estruturas Tecnológicas, Tendências Demográficas e Cuidados de Saúde

As populações dos países denominados desenvolvidos, onde a concepção e implementação de SIH é significativa, de um modo geral, atravessam um processo de envelhecimento,

quer devido ao decréscimo da taxa de natalidade, quer devido ao aumento da esperança média de vida. É na fase mais avançada da idade que a necessidade de cuidados continuados de saúde atinge uma maior frequência, contudo, é também nesta fase que a mobilidade decresce e a relação necessidade/facilidade de acesso a cuidados de saúde se torna mais crítica. Conceitos como telemedicina surgem como uma solução para este tipo de situações, possibilitando o acompanhamento médico à distância, prescindindo da deslocação do paciente até à unidade hospitalar. Este tipo de acompanhamento pode ser ainda mais importante na medida em que a monitorização continuada de sinais vitais pode permitir a antevisão da necessidade de uma intervenção que implique a presença física no mesmo local de médico e paciente.

A implementação de sistemas como a telemedicina não se prende apenas com uma questão de aceitação social mas também com a disponibilidade de infra-estruturas tecnológicas de suporte.

Tomando como exemplo o caso de Portugal, verifica-se que as regiões interiores do país sofrem um processo de desertificação pois a população mais jovem migra para os grandes centros urbanos localizados no litoral do país. Uma consequência previsível desta distribuição populacional é que a zona litoral usufrua de mais unidades hospitalares assim como de mais recursos tecnológicos e mais avançados de forma a servir uma maior fatia da população. O reverso da medalha verifica-se no interior do país onde a população é tendencialmente mais envelhecida e, portanto, a necessidade de cuidados de saúde regulares é potencialmente maior. Neste contexto, a telemedicina era uma possível solução para minimizar a menor disponibilidade de centros hospitalares que, quando existem, normalmente possuem uma capacidade de resposta a casos mais específicos menor. Contudo, a implementação de tais sistemas encontra desde logo dificuldades como a própria disponibilidade de infra-estruturas tecnológicas.

2.5 Conclusões

Independentemente de todas as capacidades de um SIH, existe uma característica bastante importante nos cuidados médicos que jamais um sistema poderá disponibilizar: a relação médico/paciente. As gerações mais recentes possuem uma predisposição para o uso de tecnologias incomparavelmente maior que as gerações precedentes. No entanto, o contacto humano é insubstituível. Nesta perspectiva, os SIH podem e devem evoluir no sentido de proporcionar informação também vocacionada para os pacientes mas o seu foco deverá ser o apoio ao acto médico e não a sua substituição, ou seja, facilitar em vez de automatizar completamente.

A tendência para a construção de sistemas com um maior nível de distribuição permite antever a possibilidade de disponibilização de um conjunto de funcionalidades orientadas ao paciente. As funcionalidades em questão, beneficiarão em muito as unida-

des hospitalares na medida em que existem mais pontos de contacto entre instituição e população-alvo. Para que tal objectivo seja alcançado com sucesso, é necessário investir numa rede de infra-estruturas tecnológicas uniformemente distribuídas, proporcionando a toda a população o mesmo nível de suporte social.

Atendendo à realidade portuguesa, em que os cuidados de saúde são baseados no sector público, a viabilidade de troca de informação clínica entre instituições é maior. No caso dos E.U.A., em que os cuidados de saúde são realizados, essencialmente, pelo sector privado, esta partilha torna-se mais difícil devido a políticas organizacionais e heterogeneidade de sistemas.

Os casos de estudo sobre implementação de SIH, bem como sobre as suas experiências de utilização, são escassos. Verifica-se a necessidade de investir nesta área não só pela sua importância social mas também pelo seu potencial económico. É importante que sejam comparados diferentes cenários de implementação e de utilização para que possam ser identificados pontos comuns e pontos críticos.

Apesar de tudo o que possa ser investigado ou desenvolvido, importa salientar que o objectivo final é o paciente e não a tecnologia. De nada vale um SIH altamente desenvolvido, repleto de inovação, se isso não se traduzir num benefício, directo ou indirecto, para o paciente.

Capítulo 3

Exploração de Dados Analíticos de MCDT

Neste capítulo são descritas em mais pormenor as necessidades que originaram o desenvolvimento do eResults, em especial o seu módulo de Exploração de Dados Analíticos de MCDT e quais as características pretendidas de forma a satisfazer essas mesmas necessidades. Para além deste módulo, são também identificadas as características de um outro módulo (de apoio) do eResults no domínio da filtragem de resultados.

3.1 eResults

Actualmente, uma fatia considerável de serviços clínicos relacionados com os Meios Complementares de Diagnóstico e Terapêutica (com as suas várias especialidades) possui soluções informáticas para a disponibilização de resultados por si produzidos. Contudo, estes são uma mais-valia caso possam ser consultados em tempo útil e pelos intervenientes correctos. Por este motivo, existe todo o interesse em proporcionar um meio rápido, fiável e seguro através do qual a informação possa ser consultada de forma não só eficaz mas também eficiente, substituindo os processos de divulgação manuais ou semi-automáticos que por vezes se verificam. De forma a conferir coerência na consulta, o eResults tem como objectivo ser uma solução agregadora e distribuidora de diferentes tipos de resultados, proporcionando uma boa experiência de utilização onde a rapidez e coerência de consulta são atributos essenciais. A figura 3.1 da página 20 ilustra o objectivo principal do eResults, ser um elemento agregador e distribuidor de informação clínica.

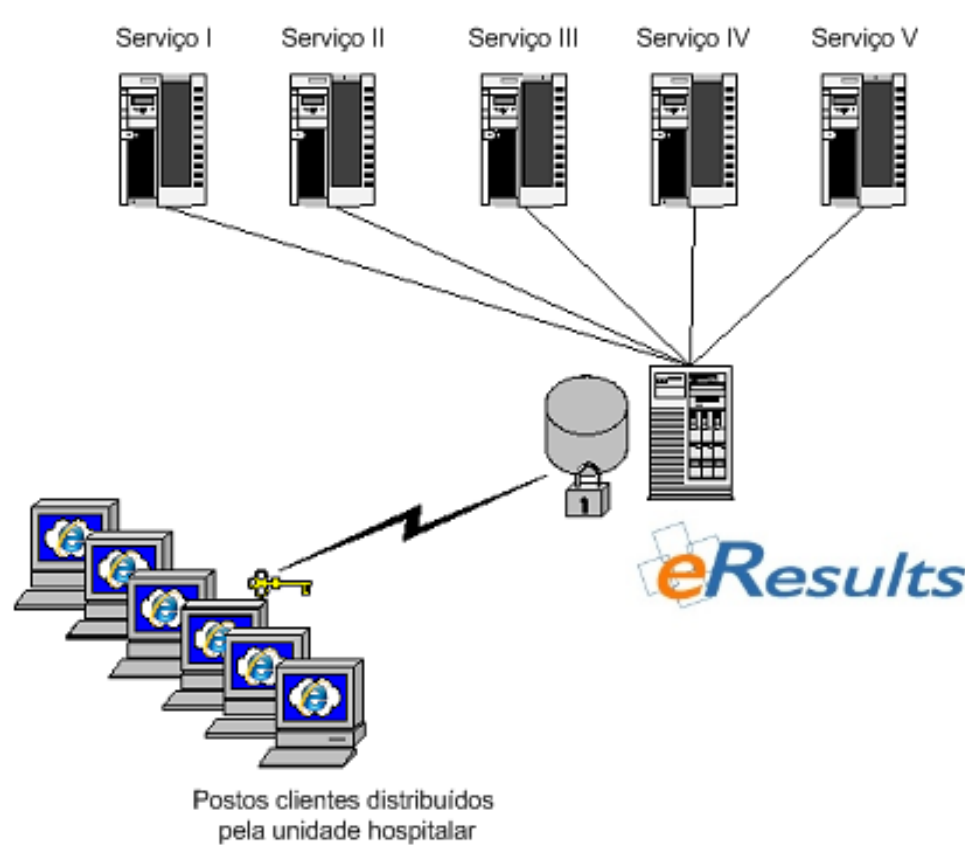


Figura 3.1: eResults como elemento agregador e distribuidor de MCDT.

3.2 Resultados Analíticos

Os resultados analíticos são um grupo específico de MCDT cuja complexidade exige uma atenção redobrada. Os dados visualizados através deste módulo são provenientes de laboratórios existentes numa unidade hospitalar, como por exemplo, Patologia Clínica ou Imunohematologia onde a diversidade de parâmetros de análise e suas hierarquias é bastante elevada (actualmente, na ordem dos 1100 registos).

A acrescentar ao elevado número de dados, cada parâmetro pertence a um dado tipo, nomeadamente, alfanumérico, microrganismo ou gráfico, sendo que cada um deles exige um tratamento diferente ao nível da sua visualização e funcionalidades disponíveis.

Os parâmetros possuem uma hierarquia intrínseca motivo pelo qual é conveniente clarificar alguns conceitos subjacentes. O elemento hierárquico base é uma análise simples. Esta designação advém do facto de ser o elemento mais básico pois é um parâmetro que tem associado um resultado concreto e não possui análises filho. O elemento hierárquico superior é uma análise complexa que representa um conjunto de análises simples, portanto, tem sempre análises filho, podendo ter ou não resultado do tipo gráfico (analisado mais à frente). Existem ainda dois níveis hierárquicos superiores destinados exclusiva-

mente à apresentação, o grupo e o subgrupo. Um subgrupo representa um conjunto de análises simples e análises complexas e pertence sempre a um grupo. Um grupo representa uma qualquer combinação de análises simples, complexas e subgrupos. Existe ainda um outro conceito, o perfil, característico de aplicações de inserção de pedidos de resultados que representa qualquer combinação de simples e complexas. Com este conceito pretende-se proporcionar um meio de adaptação do pedido às necessidades do utilizador/unidade hospitalar sendo especialmente útil aquando da existência de protocolos para determinados casos clínicos. Por exemplo, perante uma suspeita de patologia X devem ser realizadas as análises simples A, B, C e D e as análises complexas M e N, todas pertencentes ao perfil X. Caso não existisse o conceito de perfil, a introdução do pedido poderia ser um processo moroso e propenso ao erro. Desta forma basta introduzir o pedido do perfil X. Do ponto de vista do eResults, interessa conhecer este tipo de relação hierárquica pois a sua apresentação pode ser necessária.

Tendo em conta toda esta diversidade, pretende-se tornar a consulta de resultados analíticos uma experiência de utilização que se pautar pela coerência e uniformização de apresentação dos mesmos, nunca esquecendo que o objectivo final é a sua utilidade num processo de diagnóstico eficiente. No domínio da apresentação dos resultados importa referir que existe um conceito-chave que é o de requisição. Uma requisição não é mais do que o pedido de realização de um conjunto de análises. Esta definição simplista esconde todo um conjunto de características que tornam este tipo de MCDT tão rico e ao mesmo tempo tão complexo e sensível. Desde logo, a requisição tem a si associada um conjunto de informações-base de extrema importância para a correcta gestão do seu ciclo de vida, que vai desde qual o serviço/especialidade que a requereu e qual o serviço/especialidade que a executou, até às datas de validação de cada um dos parâmetros. Informações como o tipo de produto (sangue, urina, fezes, etc.) no qual o parâmetro foi avaliado, a data do episódio associado (visita à unidade hospitalar), identificação externa da aplicação fornecedora, valores de referência e unidades, são alguns exemplos de informação complementar que importa acompanhar o resultado propriamente dito, quer por uma questão de gestão interna da unidade hospitalar, quer para a determinação de um diagnóstico correcto.

Atendendo à quantidade de informação produzida no âmbito dos resultados analíticos, é essencial a disponibilização de meios de pesquisa que permitam ao utilizador seleccionar a informação a visualizar, através de critérios de pesquisa e, eventualmente, aplicar algum tipo de filtragem. Os critérios de pesquisa devem contemplar características do paciente, do episódio e dos documentos aos quais as análises pertencem.

De forma a proporcionar de imediato uma visão global da informação clínica, pretende-se que o ponto de partida de visualização seja uma organização dos vários resultados, segundo parâmetro e requisição, na forma tabular. A apresentação dos parâmetros deve reflectir a hierarquia existente entre os mesmos assim como proporcionar um meio de

com ela interagir. Com esta perspectiva mais conservadora pretende-se facultar ao pessoal médico uma disposição de resultados com a qual estão mais confortáveis mas, simultaneamente, identificar claramente as várias opções mais específicas possíveis para cada tipo de dados apresentado.

Não só os resultados analíticos mas todos os MCDT, dada a sua natureza, constituem dados que exigem um nível de segurança redobrado devido à sua confidencialidade. Este aspecto não foi directamente abordado pois é da responsabilidade do módulo de segurança do eResults questões como autenticação, controlo de sessões ou controlo de acesso aos vários módulos.

Dada a especificidade dos vários tipos de resultados analíticos segue-se uma descrição mais detalhada de cada um deles bem como as suas necessidades específicas.

3.2.1 Resultados Alfanuméricos

Tal como a designação indica, neste grupo enquadram-se todos os resultados apresentados na forma numérica (1, 2, 10, 0.5, etc.) ou frases (por exemplo, "Bacilos gram negativo", "Cocos gram positivo", "inferior a 20", etc.). Existem parâmetros para os quais não é possível aferir um valor concreto, motivo pelo qual a avaliação do mesmo resulta numa frase elucidativa da situação. Este tipo de dados não carece de nenhuma alternativa à sua visão global pois o próprio resultado encerra em si toda a descrição possível.

Já no caso dos resultados numéricos, além da sua apresentação na sua forma mais convencional, existe todo o interesse em verificar a evolução do parâmetro em causa, a posição relativa de dois ou mais parâmetros ou até avaliar uma possível correlação de parâmetros ao longo do tempo. Para tal, verifica-se a necessidade de criar uma interface cujo elemento central é um gráfico que traduza a evolução dos parâmetros do tipo numérico registados para um conjunto de requisições. Um gráfico, do ponto de vista técnico, não é mais do que uma forma diferente de ver a mesma informação, no entanto, do ponto de vista clínico, poderá fazer a diferença pois a verificação de tendências e interações pode, eventualmente, ser imediata, algo que é bastante mais difícil quando se analisam apenas valores, mesmo que devidamente organizados.

3.2.2 Microrganismos

Os resultados do tipo microrganismo têm como primeira instância a identificação de quais os microrganismos presentes no produto em causa. No entanto, este tipo de resultados tem ainda associado outros conceitos, os conceitos antibiograma e antibiótico.

Um antibiótico é uma substância que tem capacidade de interagir com microrganismos. Os antibióticos interferem com estes, eliminando-os ou inibindo o seu metabolismo e/ou a sua reprodução, permitindo ao sistema imunológico combatê-los com maior

eficácia. Um antibiograma não é mais do que avaliação da sensibilidade de um microrganismo a um conjunto de antibióticos.

Como é possível depreender, estamos já num segundo nível de especificidade dos dados, pelo que, de forma a manter a tabela inicial simples, verifica-se a necessidade do desenvolvimento de mais uma interface sendo esta vocacionada para antibiogramas. Para esta interface pretende-se, novamente, que a apresentação seja na forma tabular onde os resultados estão organizados segundo requisição/microrganismo e antibiótico. Dado o conjunto restrito de resultados que uma avaliação deste género pode originar (resistente, indeterminado e sensível) existe todo o interesse em proporcionar um meio rápido de identificação do resultado (por exemplo, através de um código de cores).

A interface de partida (tabela global) deve identificar claramente a existência deste nível de especificidade para determinado resultado e proporcionar um meio expedito de aceder à interface correspondente.

3.2.3 Gráficos

Alguns aparelhos que efectuem as avaliações dos parâmetros produzem representações gráficas dos resultados, por exemplo, gráficos de dispersão, sob a forma de uma imagem. Este tipo de resultados pode corresponder a uma análise simples, sendo a imagem o resultado em si, ou pode corresponder a uma complexa. Neste último caso, o aparelho produz a representação gráfica com o objectivo de proporcionar um meio de verificar alguma relação entre os vários parâmetros simples que constituem a complexa. Nesta perspectiva, a imagem pode facultar a mesma informação que a visualização do gráfico de evolução poderia proporcionar, no entanto a forma de apresentação pode variar e fornecer informação adicional. Por este motivo, é interessante proporcionar um meio de visualização específico para este tipo de resultados. Importa salientar que dada a raridade deste tipo de resultados, a implementação desta interface específica poderá acontecer numa fase posterior. Contudo, é necessário dotar a aplicação da capacidade de suporte a mais este tipo.

Tal como se verifica no caso dos microrganismos, também este nível de especificidade deve ser identificado claramente na tabela global bem como a existência de um meio expedito de aceder à interface deste tipo de dados.

3.3 Filtros Globais

Considerando a diversidade de serviços existentes numa unidade hospitalar e as características muito próprias de algumas especialidades, assim como o seu nível de interacção,

torna-se evidente que, no quotidiano, nem todos os resultados analíticos interessam a todos os elementos que constituem o pessoal médico. Pelo mesmo motivo, verifica-se uma situação semelhante no domínio dos tipos de documentos.

Surge a necessidade de desenvolver uma interface e um mecanismo persistente para controlar a informação visualizada, mecanismo esse que pode ser extrapolado para outros módulos do eResults, nomeadamente, Pesquisa Documental e Linha de Histórico (pretende mostrar o mesmo tipo de informação que a Pesquisa Documental mas numa perspectiva cronológica). Contudo, deve ser contemplada a hipótese de, pontualmente, efectuar pesquisas não afectadas pelas definições dos filtros globais.

Existem dois tipos de filtros a aplicar que são descritos de seguida, por tipo de documento e por serviço.

3.3.1 Filtragem por Tipo de Documento

Numa unidade hospitalar existe um conjunto diverso de tipos de documentos produzidos por várias aplicações. Obviamente que existem tipos que não estão no domínio de um determinado utilizador ou simplesmente são pouco relevantes para a sua actividade médica e, por conseguinte, perfeitamente dispensáveis nas pesquisas regulares. Esta vertente de filtragem por tipo de documento é mais adaptada a outros módulos do eResults visto que, no caso dos resultados analíticos, o conjunto do tipo de documentos em que estes se enquadram é relativamente restrito. Existe uma hierarquia entre local de origem, aplicação de origem e tipo de documento que deve ser explícita na apresentação dos filtros. Uma unidade hospitalar pode ser composta por instalações dispersas daí o conceito de local. Em cada um desses locais podem existir diferentes aplicações instaladas pelo que também é necessário identificar claramente a aplicação de origem. Cada aplicação pode produzir diferentes tipos de documento sendo que existe também uma hierarquia entre estes, assumindo o papel de tipo de documento agregador ou tipo de documento filho.

3.3.2 Filtragem por Serviço

Para uma dada requisição, um serviço pode assumir duas facetas, serviço requisitante ou serviço executante. Por este motivo, a filtragem por serviço é, na verdade, dividida em filtragem por serviço requisitante e filtragem por serviço executante. A respectiva interface deve permitir definir quais os serviços que devem ser incluídos nas pesquisas posteriormente efectuadas. Atendendo ao número de serviços existentes, deve também ser disponibilizado um meio de pesquisa de serviços de forma a agilizar o processo de escolha.

Capítulo 4

Revisão Tecnológica

Este capítulo tem como objectivo descrever todo o processo de pesquisa e opções tomadas no âmbito das alternativas tecnológicas e ferramentas de apoio ao desenvolvimento disponíveis no mercado.

Ao nível da arquitectura da solução, era imperativo que a arquitectura implementada fosse orientada a serviços (SOA – Service Oriented Architecture¹). Esta necessidade surge de um conjunto de factores. Em primeiro lugar, este tipo de arquitectura possui várias características bastante interessantes, características essas que vão de encontro a alguns requisitos do projecto eResults. Em segundo lugar, é o tipo de arquitectura adoptada nos vários projectos em que o grupo de trabalho responsável pelo eResults se vê envolvido, logo, havia todo o interesse em manter a tendência. A adopção desta arquitectura foi conseguida através da implementação de *webservices*. Atendendo ao facto de que estes dois conceitos são pedras basulares de toda a arquitectura e processo de implementação do eResults, estes serão sucintamente descritos nos pontos que se seguem.

A fase inicial do Projecto foi dedicada a realizar pesquisas e testes a ferramentas bastante recentes cuja documentação, por vezes, era no mínimo escassa. Numa primeira fase, o principal alvo de atenção foi a ferramenta “Web Service Software Factory: Modeling Edition” (também conhecida como **Service Factory**), bem como algumas tecnologias às quais recorre, e a sua comparação com uma aplicação desenvolvida internamente (designada **Gerador CPCHS**). A fase seguinte consistiu na análise e escolha da ferramenta de auxílio à obtenção dos dados a partir do armazém de dados. As ferramentas analisadas foram o **Gerador CPCHS**, **Repository Factory** e **Linq**. Na fase final do período de investigação estudaram-se os componentes da Infragistics² que poderiam auxiliar na

¹http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html, consultado pela última vez em 27 de Junho de 2008.

²<http://www.infragistics.com/>, consultado pela última vez em 26 de Junho de 2008.

produção da interface gráfica. De seguida descrevem-se em mais pormenor cada uma das tecnologias e ferramentas acabadas de enunciar.

Apesar de existir um vasto conjunto de alternativas às opções tomadas estas estavam, de certa forma, condicionadas por dois motivos. O primeiro é a parceria existente entre a CPCHS e a Microsoft o que conduz a uma clara preferência por tecnologias e ferramentas oriundas da mesma. Como factor de reforço, o desenvolvimento do eResults é apoiado pela Microsoft. O segundo motivo prendia-se directamente com uma decisão do cliente, qual o Sistema de Gestão de Base de Dados (SGBD) a utilizar, sendo que as opções eram Oracle ou SQL Server. Esta decisão influenciou a escolha da ferramenta auxiliar para o acesso a dados.

No que diz respeito à plataforma de desenvolvimento, tendo em conta as opções para as várias ferramentas e tecnologias, a escolha recaiu sobre Microsoft Visual Studio residindo a dúvida entre a versão 2005 e 2008. Esta pequena dúvida permanecia devido à incompatibilidade entre o Repository Factory e a versão 2008. Quanto à ferramenta para interacção com o SGBD não foi realizada qualquer pesquisa, tendo sido seguida a prática corrente da equipa de desenvolvimento, a utilização do SQL Navigator 5.1 XPert Edition³. Situação idêntica verificou-se na ferramenta utilizada para os testes unitários, NUnit⁴.

4.1 Arquitectura Orientada a Serviços

Uma arquitectura orientada a serviços (Service Oriented Architecture - SOA), segundo [MLM⁺06] é um paradigma de organização e utilização de soluções distribuídas e sob o controlo de diferentes domínios. Não é uma solução para o domínio do problema em causa mas uma forma de organização e distribuição de funcionalidades detidas pela própria aplicação ou controladas por outras.

De uma forma geral, as pessoas e organizações (entidades) criam soluções para resolver os obstáculos com que se deparam no decurso do seu negócio e também suportar as soluções existentes. É natural que as necessidades de uma entidade possam ser colmatadas pelas ofertas de uma outra entidade, no caso da computação distribuída, as necessidades de um agente de um computador podem ser resolvidas por um agente residente num outro computador.

A relação oferta/procura, ou seja, a correlação entre necessidades e soluções não tem de ser necessariamente um para um. A granularidade da correlação vai desde o fundamental até ao complexo, onde a satisfação de uma necessidade pode requerer o acesso a mais do que uma solução enquanto uma única solução pode satisfazer mais do que uma necessidade.

³<http://www.quest.com/sql-navigator/>, consultado pela última vez em 26 de Junho de 2008.

⁴<http://www.nunit.org>, consultado pela última vez em 26 de Junho de 2008.

No que respeita a este “jogo” de oferta e procura, o paradigma SOA consiste numa poderosa *framework*, cujo objectivo é fazer a correspondência entre necessidades e soluções, proporcionando o endereçamento das soluções para as respectivas necessidades.

O paradigma SOA assenta em três conceitos-chave, visibilidade, interacção e efeito.

A visibilidade refere-se à capacidade de as entidades com necessidades (consumidoras) e as entidades que disponibilizam soluções (fornecedoras) se reconhecerem mutuamente. Tipicamente, isto torna-se possível através da disponibilização de descrições para aspectos tais como funções, requisitos técnicos, políticas de acesso ou resposta. As descrições devem obedecer a uma forma padrão que permita a correcta compreensão, quer a nível sintáctico, quer a nível semântico.

O conceito de interacção refere-se à utilização de uma determinada solução. Regra geral, a interacção consiste numa série de trocas de informação e invocação de acções, recorrendo à troca de mensagens. Existem várias facetas na interacção, contudo, todas elas se verificam num contexto de execução particular (o conjunto de elementos técnicos e de negócio que formam um caminho desde as entidades consumidoras até às entidades fornecedoras). O contexto de execução possibilita que fornecedores e consumidores dos serviços interajam e decidam quais as políticas de segurança e contratos que deverão existir.

Essencialmente, uma interacção é um “acto” e não um “objecto”, tendo como resultado um efeito (ou um conjunto/série de efeitos). Existe uma distinção entre acções públicas e privadas, sendo que as últimas são desconhecidas pelas outras partes. Por outro lado, as acções públicas resultam em alterações do estado partilhado pelas entidades envolvidas na interacção (e eventualmente por outras).

Um pressuposto deste tipo de arquitectura é a transparência de localização, ou seja, as aplicações procuram os serviços num directório e ligam-se aos mesmos durante a sua execução.

Para que os clientes possam encontrar os fornecedores dos serviços pretendidos é necessário existir um local de consulta, um mapa. De forma a proporcionar esta base, os fornecedores de serviços registam-se na UDDI (Universal Discovery Description and Integration)⁵. Após localizar o fornecedor do serviço, o cliente inicia a comunicação com o mesmo através da troca de mensagens de forma a alcançar o objectivo da interacção. A figura 4.1 da página 28 ilustra este modelo de comunicação.

Importa realçar que apesar da arquitectura SOA ser geralmente implementada com recurso a *webservices*, os serviços podem ser disponibilizados através de outras estratégias

⁵http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec, consultado pela última vez em 27 de Junho de 2008.

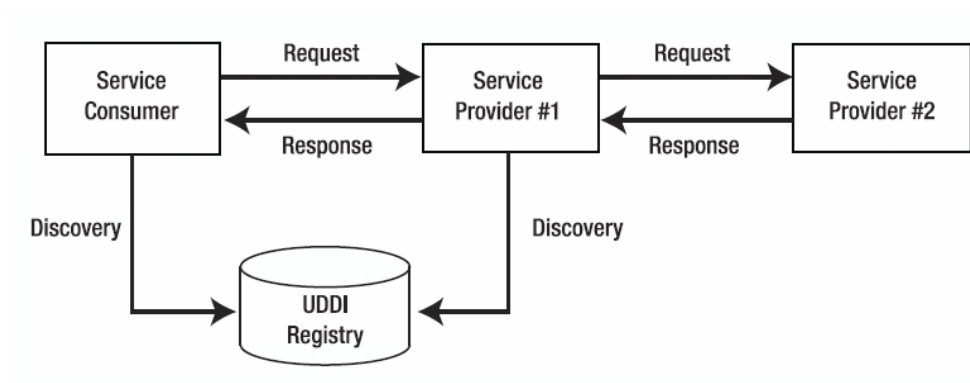


Figura 4.1: Modelo de comunicação entre cliente e servidor.

de implementação, como por exemplo, RPC⁶, DCOM⁷, CORBA⁸, EJB⁹ ou Servlets¹⁰.

De um ponto de vista mais prático, interessa identificar quais as vantagens que tornam a arquitectura SOA uma arquitectura que se prevê amplamente utilizada no futuro.

Numa perspectiva mais economicista, talvez a vantagem mais importante seja um melhor retorno do investimento. A criação de uma camada de serviços robusta conduz directamente ao aumento do retorno do investimento efectuado no desenvolvimento do software. Criando uma camada de serviços independente, esta poderá ter um ciclo de vida superior às aplicações que consomem os serviços.

Na vertente técnica destacam-se várias vantagens, nomeadamente, a mobilidade do código, a definição dos papéis dos programadores, o paralelismo de desenvolvimento, gestão de erros mais eficaz e consequente qualidade global da aplicação, suporte a diferentes tipos de clientes e reutilização de serviços [Ste].

Tendo em conta que a transparência é uma característica da arquitectura SOA, a mobilidade de código é perfeitamente exequível. A procura e endereçamento dinâmico demonstram que não é necessário que o serviço se encontre sempre no mesmo local, o sucesso da interacção não depende disso. Consequentemente, a organização tem a flexibilidade para mover serviços para diferentes máquinas ou mesmo para um fornecedor externo.

Está subjacente à arquitectura SOA que as aplicações possuam diferentes camadas, sendo que cada uma delas exige dos programadores diferentes competências. Deixa de ser fundamental que as organizações recorram a recursos humanos altamente experientes, numa perspectiva global da aplicação, e possam atingir os seus objectivos com recursos humanos menos experientes e focados numa área mais específica. Pelo mesmo motivo,

⁶<http://www.cs.cf.ac.uk/Dave/C/node33.html>, consultado pela última vez em 27 de Junho de 2008.

⁷<http://msdn.microsoft.com/en-us/library/ms809340.aspx>, consultado pela última vez em 27 de Junho de 2008.

⁸<http://www.cse.wustl.edu/~schmidt/corba.html>, consultado pela última vez em 27 de Junho de 2008.

⁹<http://java.sun.com/products/ejb/>, consultado pela última vez em 27 de Junho de 2008.

¹⁰<http://java.sun.com/products/servlet/>, consultado pela última vez em 27 de Junho de 2008.

o paralelismo de desenvolvimento é perfeitamente possível a partir do momento em que estão definidas as interfaces.

Independentemente da metodologia de desenvolvimento, a concepção e implementação de uma aplicação com sucesso requer a realização de um conjunto de testes. Com uma arquitectura orientada a serviços o processo de teste e correcção de anomalias é amplamente facilitado. No processo de desenvolvimento, é possível aplicar testes unitários com maior facilidade na medida em que existe uma interface definida. O isolamento em camadas permite identificar claramente a origem do problema, nos *webservices* ou na camada applicacional. Ao ser facilitada a execução de testes verifica-se a tendência da existência de menos erros, tendo como consequência um melhor nível global de qualidade.

O suporte a clientes de vários tipos é uma característica intrínseca ao próprio paradigma da arquitectura SOA. Conhecida a interface do serviço, qualquer tipo de cliente pode comunicar com qualquer tipo de servidor. Da mesma forma, a reutilização de serviços é bastante fácil de alcançar, existindo independência de versões de compiladores, plataformas e outras incompatibilidades que tornam, por exemplo, a reutilização de código bastante difícil.

Do ponto de vista de características da solução verificam-se ainda várias vantagens, tais como, mais segurança, melhor manutenção e maior escalabilidade e disponibilidade.

Tipicamente, numa arquitectura monolítica, questões de autenticação e segurança em geral são controladas no *front-end*, sendo descuradas tais verificações, por exemplo, ao nível do acesso à base de dados. A implementação de *webservices* implica que estes possuam os seus próprios mecanismos de segurança, conferindo às aplicações autenticação multi-nível, ao nível do cliente e ao nível do serviço. A localização e correcção de erros são processos que podem ser desempenhados mais facilmente na medida em que a lógica de negócio se concentra na camada do serviço.

A necessidade de transparência de localização dá origem a duas vantagens neste domínio. A escalabilidade é potenciada e o nível de disponibilidade da solução é elevado. Visto que é possível executar várias instâncias do mesmo serviço em máquinas diferentes, caso uma máquina falhe, é possível redireccionar o pedido para outro serviço, sem qualquer prejuízo para a camada cliente.

Em suma, uma arquitectura SOA, através da partilha de *webservices*, permite o desenvolvimento de aplicações compostas tendo como consequência uma maior disponibilidade de funcionalidades.

4.2 Tecnologias para a Comunicação

Tendo em conta a opção por uma arquitectura do tipo SOA com implementação de *webservices*, impunha-se investigar as principais tecnologias sobre as quais se realizaria

a comunicação entre a aplicação cliente e a aplicação servidor. No âmbito das tecnologias para a comunicação foram identificadas duas alternativas, a utilização conjunta de ASP.NET Web Services e Web Services Enhancements ou o recurso a Windows Communication Foundation. As três tecnologias consideradas são descritas, de uma forma sucinta, em seguida.

4.2.1 ASP.NET Web Services (ASMX)

O **ASP.NET Web Services**¹¹ não é mais que a implementação de *webservices* recorrendo à *framework* .NET. Por este motivo, interessa definir genericamente o que é um *webservice* e quais as particularidades da implementação através da *framework* .NET.

Um *webservice* é uma plataforma padrão para a construção de aplicações distribuídas com capacidade de interoperabilidade. Permite a interacção de duas máquinas, uma no papel de cliente e outra no papel de servidor (num determinado momento da comunicação, pois os papéis não são fixos), usando para tal uma rede.

Os *webservices* possuem algumas características que elevam a sua flexibilidade e utilidade. São independentes dos sistemas operativos, o que confere uma grande flexibilidade, e podem ser utilizados não só em *intranets* mas também através da Internet [Pod03].

Um *webservice* pode ser visto como uma “caixa negra” com a qual se pode interagir não sendo necessário conhecer os detalhes da sua implementação. Contudo, para que a interacção se realize com sucesso, é necessário conhecer a sua interface. Esta é descrita pelo WSDL (Web Service Definition Language)¹², um formato ao alcance da compreensão de uma máquina. A interacção com o *webservice* é realizada através de mensagens SOAP (Simple Object Access Protocol)¹³ construídas segundo a descrição do serviço *Web* fornecida, normalmente, através do protocolo HTTP (Hypertext Transfer Protocol)¹⁴ e com serialização por XML (eXtensible Markup Language)¹⁵, em conjunto com outros padrões definidos para a *Web*. Na descrição da interface é especificado o formato das mensagens, o tipo de dados, protocolos de transporte e formatos de serialização que devem ser utilizados na comunicação entre o cliente e o fornecedor do serviço. De igual forma, é especificada uma ou mais localizações onde o *webservice* está disponível.

O conceito primário subjacente a um *webservice*, a comunicação entre dois sistemas através de mensagens, não é algo de inovador pois já outras tecnologias o fazem. A grande barreira ultrapassada é o facto de tornar o processo comunicacional através da Internet independente do tipo de servidor e cliente que estão a ser executados em cada extremo. Assim, surge como um protocolo comum, simples e robusto [CPdCHS07].

¹¹<http://msdn.microsoft.com/en-us/library/ms972326.aspx>, consultado pela última vez em 26 de Junho de 2008.

¹²<http://www.w3.org/TR/wsdl>, consultado pela última vez em 24 de Junho de 2008.

¹³<http://www.w3.org/TR/soap/>, consultado pela última vez em 24 de Junho de 2008.

¹⁴<http://www.ietf.org/rfc/rfc2616.txt>, consultado pela última vez em 24 de Junho de 2008.

¹⁵<http://www.w3.org/XML/>, consultado pela última vez em 24 de Junho de 2008.

Utilizar ASP.NET para criar *webservices* na *framework* tem algumas vantagens mas uma das mais importantes é o facto de o programador poder dedicar-se exclusivamente a questões relacionadas com a lógica de negócio, delegando na *framework* as tarefas de mais baixo nível, construção e consumo dos *webservices*, tais como a geração da WSDL e dos *proxies*, a serialização dos dados em XML, o transporte dos dados via HTTP e a reconversão dos dados a partir do XML. Uma outra característica importante é o facto de o leque de linguagens de programação ser considerável. Os ficheiros construídos com o recurso a ASP.NET têm a extensão "asmx" e são estes os pontos de acesso (*endpoint*) ao serviço criado. Os ficheiros "asmx" são automaticamente compilados pelo ASP.NET quando é efectuado um pedido ao serviço, sendo retornado um XML que contém a descrição do serviço.

4.2.2 Web Services Enhancements (WSE)

O WSE¹⁶ é um complemento à *framework* .NET da Microsoft. Fornece um conjunto de classes que implementam especificações adicionais aos *webservices* no domínio da segurança, endereçamento, fiabilidade das mensagens e no envio de anexos. Mais concretamente, o WSE fornece extensões ao protocolo SOAP permitindo, por exemplo, segurança ponta a ponta ao nível da mensagem, endereçamento com base no conteúdo e políticas de acesso, o que possibilita o desenvolvimento de *webservices* mais seguros.

A garantia de integridade, confidencialidade e autenticação das comunicações (através de mensagens SOAP) verifica-se a vários níveis. A integridade da mensagem é obtida através de assinaturas digitais XML garantindo que partes da mensagem não tenham sido alteradas após receber a "assinatura" do remetente. A confidencialidade da mensagem é baseada na especificação de criptografia XML e garante que partes correspondentes da mensagem só possam ser compreendidas pelos destinatários desejados.

Existe ainda a flexibilidade ao nível da fase em que essas capacidades são implementadas, o que pode acontecer no próprio código-fonte ou através de um ficheiro de políticas de acesso.

4.2.3 Windows Communication Foundation (WCF)

O Windows Communication Foundation¹⁷, também conhecido como WCF, é uma plataforma de programação e sistema de execução desenvolvida pela Microsoft para a construção, configuração e implementação de serviços distribuídos, dotando as aplicações de capacidades de intercomunicação. Foi introduzida em Dezembro de 2006 como parte integrante da .NET Framework e dedica-se a todo o processo de comunicações. Pelo

¹⁶<http://msdn.microsoft.com/en-us/library/ms977323.aspx>, consultado pela última vez em 26 de Junho de 2008.

¹⁷<http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>, consultado pela última vez em 26 de Junho de 2008.

facto de se incluir no pacote da .NET Framework, coloca os programadores numa posição mais confortável na medida em que o leque de linguagens de programação suportado é o mesmo que a *framework*.

O WCF não é uma tecnologia "concorrente" das tecnologias anteriormente descritas, pois resulta da junção das duas e ainda de **.NET Enterprise Services (ES)**¹⁸, **.NET Remoting**¹⁹ e **Microsoft Message Queuing (MSMQ)**²⁰. A tecnologia WCF apresenta-se como a junção do melhor de cada um dos mundos, parecendo a solução mais viável e completa. Cada uma das tecnologias anteriores foi, e ainda é, usada desde há cerca de uma década para o desenvolvimento, com sucesso, de aplicações distribuídas em plataformas Windows. No entanto, a forma de utilizar cada uma delas é, por vezes, incompatível com as restantes pelo que o programador podia ser obrigado a possuir um conhecimento mais profundo acerca de pormenores de mais baixo nível, o que acarreta a necessidade de um prazo mais alargado de desenvolvimento. Com esta nova solução, passam a estar disponíveis funcionalidades que outrora eram incompatíveis ou de difícil coexistência. Cada um dos padrões referidos anteriormente foi também desenvolvido pela Microsoft em cooperação com parceiros ao longo de alguns anos, o que garante interoperabilidade com um vasto conjunto de plataformas, ambientes de execução e linguagens de programação [Vas].

O WCF recorre a mensagens do tipo SOAP para a comunicação entre dois processos. Consequentemente, torna as aplicações baseadas em WCF capazes de comunicar com qualquer outra aplicação que recorra igualmente a mensagens SOAP para comunicar. No caso de a comunicação se realizar entre duas aplicações que fazem uso de WCF, as mensagens SOAP são codificadas num formato binário otimizado. Caso a comunicação ocorra com uma aplicação que não implementa WCF, é usada uma codificação baseada em XML. Em qualquer das vertentes, as codificações respeitam o formato das estruturas de dados SOAP.

Um serviço WCF é composto por três componentes essenciais: uma classe onde é implementado o serviço, um ambiente *host* que aloja o serviço e um ou mais *endpoints* (por onde se realizam todas as comunicações com os clientes). Cabe aos *endpoints* fornecer informação aos clientes sobre quais os métodos aos quais dá acesso e a forma como o cliente terá de comunicar com o serviço [Corf]. Dada a complexidade dos dados com que a aplicação iria lidar era previsível que fosse necessário recorrer a actualizações parciais da página com recurso, por exemplo, a AJAX²¹. Uma das vantagens da tecnologia WCF é suportar ASP.NET AJAX [Cora].

¹⁸<http://msdn.microsoft.com/en-us/magazine/cc301491.aspx>, consultado pela última vez em 21 de Junho de 2008.

¹⁹[http://msdn.microsoft.com/en-us/library/kwdt6w2k\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/kwdt6w2k(VS.71).aspx), consultado pela última vez em 21 de Junho de 2008.

²⁰<http://www.microsoft.com/windowsserver2003/technologies/msmq/default.mspix>, consultado pela última vez em 21 de Junho de 2008.

²¹<http://www.w3schools.com/Ajax/Default.Asp>, consultado pela última vez em 21 de Junho de 2008.

4.2.4 Conclusões sobre as Tecnologias para a Comunicação

Tendo em conta todas as vantagens enunciadas ao longo da descrição do WCF e o facto de esta já incluir as duas tecnologias às quais o Gerador CPCHS recorre, a escolha recaiu na utilização de WCF, na medida em que o conjunto de requisitos seria mais facilmente satisfeito (devido à sua abrangência), requerendo menos tempo. Para além da abrangência da tecnologia, o seu desempenho é, regra geral, superior às tecnologias anteriormente referidas [Gup].

4.3 Ferramentas para a Criação dos *Webservices*

O processo de desenvolvimento de um *webservice* implementado de raiz obriga a que o programador dispense bastante tempo para tal tarefa. Se tivermos em conta que uma aplicação como o eResults necessita de mais do que um *webservice* e que cada um desses *webservices* poderá disponibilizar mais do que uma operação, é possível concluir que o tempo de desenvolvimento é bastante elevado e, consequentemente, os custos associados. Se se atender ainda ao facto da possibilidade de introdução de erros de forma involuntária (intrínseca à condição humana), é também possível concluir que a sua correcção e manutenção de alterações inflaciona ainda mais todos os custos. Por estes motivos, existe todo o interesse em recorrer a formas automáticas de geração de código. Os geradores de código permitem que exista uma coerência de implementação entre os vários serviços, livre de erros e de mais fácil manutenção. Desta forma, o programador pode focar a sua atenção em detalhes da lógica de negócio ao invés de a dividir com questões de implementação de mais baixo nível.

No âmbito das ferramentas para a criação dos *webservices* foram identificadas duas alternativas, o **Gerador CPCHS** ou a ferramenta **Web Service Software Factory: Modeling Edition**. Segue-se uma apresentação sucinta de cada uma das ferramentas assim como a identificação dos seus pontos positivos e negativos.

4.3.1 Gerador CPCHS

Tal como a designação atribuída o indica, o Gerador CPCHS foi desenvolvido internamente e com o intuito de automatizar a produção de código-fonte, vocacionado para uma arquitectura do tipo SOA. Esta aplicação é capaz de gerar (com o auxílio da ferramenta CodeSmith²²) todo o código inicial necessário ao desenvolvimento de um serviço, desde os meios de acesso ao armazém de dados (quer Oracle, quer SQL Server) até ao *proxy*. É ainda possível indicar a geração de classes "descendentes" que permitem alterar ou adicionar pormenores às classes principais, o que confere uma capacidade de personalização

²²<http://www.codesmithtools.com/>, consultado pela última vez em 26 de Junho de 2008.

assinalável e proporciona uma margem de manobra bastante confortável no que concerne a liberdade e adaptação ao modelo de negócio.

Toda a geração de código é realizada numa aproximação tabela a tabela, ou seja, cada classe gerada, que representa uma entidade de negócio e é o meio de tradução de registos para objectos, tem como base uma tabela específica da base de dados. Por este motivo, a geração dos métodos de acesso e interacção (selecção, inserção, actualização e eliminação) com a base de dados estão limitados à entidade (e respectiva tabela) a que se referem. Embora seja possível acrescentar a definição de variáveis de entrada nos métodos e também propriedades de classes que não correspondam a atributos das tabelas, a sua gestão tem de ser implementada posteriormente.

Fornecendo um exemplo mais concreto, a tabela "doente" possui três atributos: "nome", "idade" e "pais", em que este último é uma chave estrangeira para a tabela "pais" que possui dois atributos, "identificador" e "nome". A entidade "doente" possui propriedades, "nome" e "idade" (pois "pais" não foi seleccionado). No entanto, pretende-se que esta entidade possua mais uma propriedade, "nomePais". No momento da definição através do gerador é possível incluir a nova propriedade mas, após gerar todo o código pretendido, é necessário alterar a classe de personalização da respectiva entidade para que a propriedade adicionada seja preenchida aquando da transformação dos dados provenientes do armazém de dados para instâncias das entidades. Este tipo de operações torna o desenvolvimento mais moroso e propício a erros.

Para a definição dos serviços e *proxies*, o Gerador CPCHS, recorre a *webservices* ASP.NET (ASMX) existindo a possibilidade de recorrer também à tecnologia WSE que, tal como indicado anteriormente, proporciona mais garantias ao nível de compatibilidade e segurança.

Existem duas aproximações possíveis para a definição da interface do serviço: recorre-se às classes geradas a partir do modelo de dados ou definem-se novas entidades exclusivamente para a interface.

Na primeira aproximação, na eventualidade de o resultado final pretendido para o serviço e de o modelo de dados do armazém de dados diferirem razoavelmente, a manutenção de alterações torna-se penosa, para além do facto da interface do serviço estar, de certa forma, dependente do modelo de dados. No entanto, o código que teria de ser desenvolvido manualmente é bastante menor (dependendo obviamente da complexidade do modelo de dados).

Na segunda aproximação, o esforço e tempo exigidos ao programador é maior (pois é necessário desenvolver mais código, incluindo a tradução de uma entidade para outra). Por outro lado, é eliminada a dependência entre interface do serviço e modelo de dados.

O código produzido pode ser enquadrado em cinco grupos conceptuais. O primeiro corresponde ao *script* SQL (Structured Query Language)²³ onde são definidos os vários

²³<http://www.w3schools.com/sql/default.asp>, consultado pela última vez em 29 de Junho de 2008.

procedimentos identificados para cada tabela. Um segundo grupo que corresponde às regras das entidades de negócio (onde são, por exemplo, criadas as instâncias das entidades a partir dos dados do armazém de dados, ou identificado o nome do pacote a aceder). Um terceiro grupo corresponde às entidades geradas. O quarto grupo engloba a classe de implementação do serviço e o quinto corresponde ao *proxy* de acesso.

4.3.2 Web Service Software Factory: Modeling Edition

O Web Service Software Factory: Modeling Edition²⁴ (habitualmente designado por Service Factory) é uma colecção integrada de recursos (ferramentas, padrões, código-fonte, etc.), desenhado de forma a auxiliar a modelação e construção de *webservices* WCF ou ASMX procurando tornar este processo rápido, fácil, consistente e eficiente, tendo como resultado final uma arquitectura que segue as boas práticas e padrões de desenho definidos. Os recursos consistem em tópicos de arquitectura e padrões na forma de *wizards* e na geração de código sob a forma de ferramentas integradas no Visual Studio 2008 [dGvL]. A figura 4.2 da página 36 ilustra a arquitectura dos serviços criados com o Service Factory.

A construção de um *webservice* recorrendo ao Service Factory implica a definição de três modelos, nomeadamente, o modelo do contrato do serviço, o modelo do contrato de dados e o modelo de "alojamento"(*host*). Esta abordagem por modelos tem como uma das suas vantagens a persistência das definições para além do momento da primeira geração de código-fonte. Em projectos de média ou grande dimensão (como é o caso do eResults), é praticamente impossível desenvolver o projecto sem uma única reformulação ou a realização de pequenas correcções e as alterações necessárias implicam uma nova geração do código. Caso não exista a persistência da definição, à qual é possível realizar ajustes, o processo de manutenção assume elevados níveis de consumo de tempo e demais recursos.

Apresentam-se em seguida cada um dos modelos e suas responsabilidades.

Modelo do Contrato de Dados

No modelo do contrato de dados, são definidos todos os tipos não primitivos utilizados no serviço, tipos de dados esses que podem ser constituídos por uma mescla de outros dados definidos no modelo e dados primitivos (*strings*, valores numéricos, etc.). Para além dos tipos básicos podem ainda ser definidas colecções, enumerações e tipos de contratos de falhas. Os contratos de falhas podem ser devolvidos quando ocorre algum erro na execução do serviço. Também este tipo de dados pode ser constituído por outros tipos, possibilitando a devolução de dados mesmo quando ocorre uma falha. Na definição do modelo podemos indicar qual a tecnologia de implementação, ASMX ou WCF.

²⁴<http://msdn.microsoft.com/en-us/library/cc487895.aspx>, consultado pela última vez em 26 de Junho de 2008.

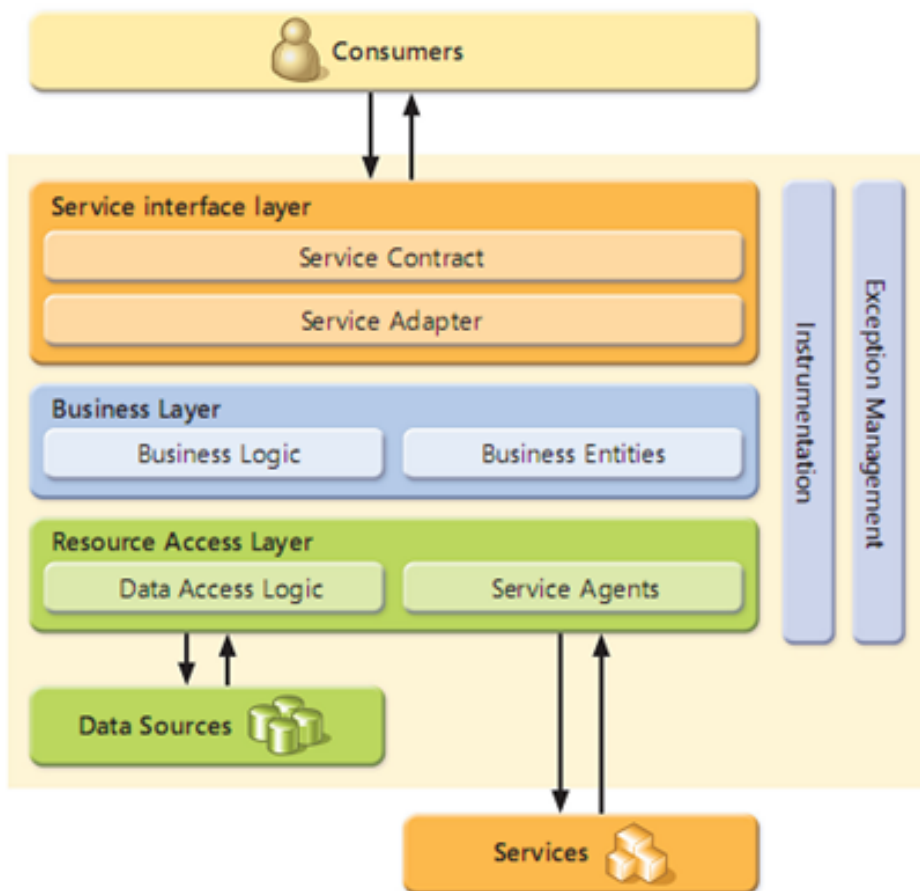


Figura 4.2: Arquitectura dos serviços criados com o Service Factory.

Modelo do Contrato de Serviços

No modelo do contrato de serviços são definidos os serviços e seus contratos, operações e mensagens.

Cada operação de um serviço pode ter, no máximo, dois contratos de mensagens associados, uma para o pedido e outra para a resposta, sendo que cada mensagem é constituída por tipos primitivos e/ou tipos definidos no modelo do contrato de dados. Tal como acontece no modelo do contrato de dados, podemos definir qual a tecnologia de implementação, ASMX ou WCF.

Modelo de "Alojamento"(Host)

No modelo de "alojamento" são definidos os *endpoints* dos serviços e os *proxies* dos clientes. Este modelo é diferente dos anteriores, dado que não é definido com base num diagrama, sendo no entanto perfeitamente funcional e explícito. Para iniciar a definição do modelo é necessário criar uma referência para uma aplicação cliente e outra para uma

aplicação *host*, indicando um projecto de implementação para cada um (já incluídos na estrutura da solução anteriormente gerada). No caso do *host*, é necessário criar uma nova referência para serviço e apontá-la para o respectivo serviço do modelo do contrato de serviços. Para cada referência de serviço, podem ser criados vários *endpoints*. No caso da aplicação cliente, podem ser criados vários *proxies*, sendo que cada um deles está afecto a um *endpoint* previamente criado.

Após a definição dos três modelos, estão reunidas as condições para gerar o serviço. No entanto, existem ainda algumas características que interessa salientar.

A primeira prende-se directamente com a manutenção do código. Para que cada alteração ao diagrama dos dados tenha efeito, é necessário gerar de novo o respectivo código e portanto a perda de qualquer código contido nas respectivas classes. De forma a contornar esta situação, as classes geradas são parciais, podendo parte da implementação estar declarada noutro ficheiro criado pelo utilizador e que não será substituído aquando da geração de código. Em resumo, o processo de gestão de alterações é tornado bastante rápido e eficiente.

Uma segunda característica refere-se à geração automática de tradutores entre entidades de negócio provenientes do acesso a dados e entidades do contrato de dados. Esta geração é apoiada por um *wizard* que possibilita, inclusivamente, o mapeamento propriedade a propriedade. Esta característica confere uma independência entre o modelo do armazém de dados e a interface do serviço bastante interessante.

A terceira característica verifica-se ao nível das tecnologias suportadas. De raiz, o Service Factory suporta as tecnologias ASMX e WCF (opção que é tomada aquando da definição dos modelos de serviços e de dados, como tecnologia de implementação) mas é fácil desenvolver uma extensão ao Service Factory para que este suporte uma tecnologia própria (característica que não foi alvo de investigação).

Um dos pontos fortes desta ferramenta é a estruturação de código que proporciona. A hierarquia de pastas e projectos criada facilita bastante o processo de criação e manutenção do código. A localização de cada projecto e a identificação da sua função na solução global são bastante claras.

Descreve-se seguidamente a hierarquia gerada, sendo esta descrição direccionada para o nível dos directórios "Source" e "Tests" (como é possível verificar na figura 4.3 da página 38).

No primeiro, o directório "Source", existe uma divisão clara em três camadas, algo semelhante ao modelo em três camadas [SCD] mas em que a camada de apresentação é, neste caso, a interface do serviço e não uma interface gráfica. No que diz respeito à lógica de negócio, esta divide-se em entidades de negócio e a própria lógica de negócio. O acesso a recursos inclui um único projecto, dedicado ao acesso a dados. O directório dedicado à interface do serviço, apesar de conter mais projectos, não é mais complexo, sendo a facilidade de compreensão da estrutura igualmente elevada. Todas as classes

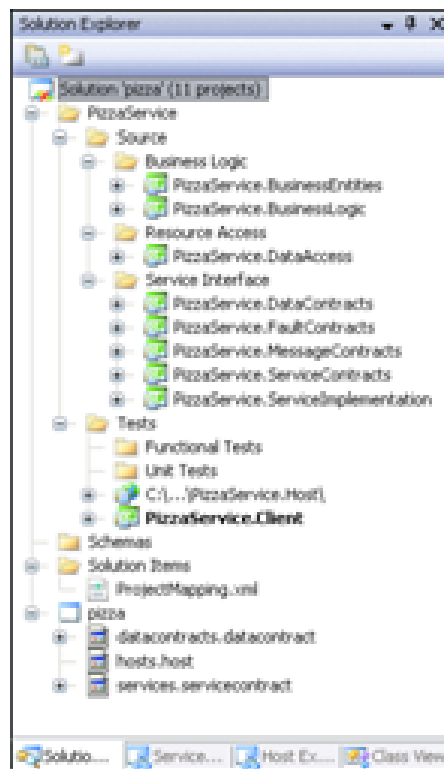


Figura 4.3: Hierarquia da solução gerada com o Service Factory.

correspondentes a tipos de dados e tipos de falhas definidos no contrato de dados são geradas nos projectos "DataContracts" e "FaultContracts" respectivamente. O código gerado a partir do contrato de serviço enquadra-se nos restantes três projectos. No "MessageContracts" são geradas todas as classes correspondentes a mensagens das operações dos serviços, quer de entrada, quer de saída. No "ServiceContracts" incluem-se todas as classes de interface do serviço, mas a implementação das respectivas operações está incluída no projecto "ServiceImplementation".

O directório "Tests" inclui apenas dois projectos "Client" e "Host". O primeiro corresponde a um projecto onde podem ser realizados vários testes ao funcionamento do serviço (por exemplo, testes unitários). No segundo, são criados os ficheiros que permitem o "alojamento" dos meios de acesso aos *webservices*.

Como característica não necessariamente negativa mas, no mínimo, menos positiva, é o facto de não existirem modelos para o acesso a dados e segurança do serviço integrados na ferramenta, ao contrário do que acontecia numa primeira versão. No entanto, existem agora como módulos independentes (o modelo de acesso a dados será analisado na secção respectiva).

De forma a clarificar a divisão de tarefas pelas várias camadas e a sua interacção, apresenta-se em seguida um exemplo de execução que pode também ser acompanhado na

figura 4.4 da página 40.

1. o *host* recebe um pedido;
2. o *host* chama a implementação do serviço;
3. a implementação do serviço chama uma função estática da lógica de negócio para obter entidades de negócio;
4. a lógica de negócio cria as entidades de negócio;
5. a lógica de negócio obtém os dados da fonte de dados (por exemplo, uma base de dados) e com os mesmos preenche as entidades de negócio;
6. a lógica de negócio devolve as entidades de negócio à implementação do serviço;
7. a implementação do serviço cria objectos do contrato de dados;
8. a implementação do serviço requisita aos tradutores a tradução das entidades de negócio para os objectos do contrato de dados;
9. a implementação do serviço cria uma nova mensagem de resposta;
10. a implementação do serviço aloca os objectos do contrato de dados às propriedades respectivas da resposta;
11. a implementação do serviço retorna a resposta ao *host*;
12. o *host* devolve a mensagem de resposta.

4.3.3 Conclusões sobre as Ferramentas para a Criação dos *Webservices*

Comparando directamente as funcionalidades de cada uma das ferramentas, a única vantagem do Gerador CPCHS relativamente ao Service Factory é o facto de já se encontrar integrada a geração do meio de acesso a dados. Contudo, esta lacuna do Service Factory é facilmente colmatada com o recurso a uma das ferramentas analisadas no ponto seguinte, nomeadamente, Repository Factory, LINQ ou até o próprio Gerador CPCHS.

Nas várias experiências realizadas com o Gerador CPCHS foram detectadas algumas falhas de execução que impediram a conclusão da tarefa com sucesso, verificando-se não ser uma aplicação com um nível de fiabilidade muito bom.

O Service Factory apresenta-se como uma solução mais robusta e flexível, onde a conformidade com boas práticas de arquitectura e programação são inerentes à própria ferramenta e de fácil implementação. A metodologia de desenvolvimento adoptada é bastante propícia à necessidade de alterações da aplicação nos vários ciclos percorridos.

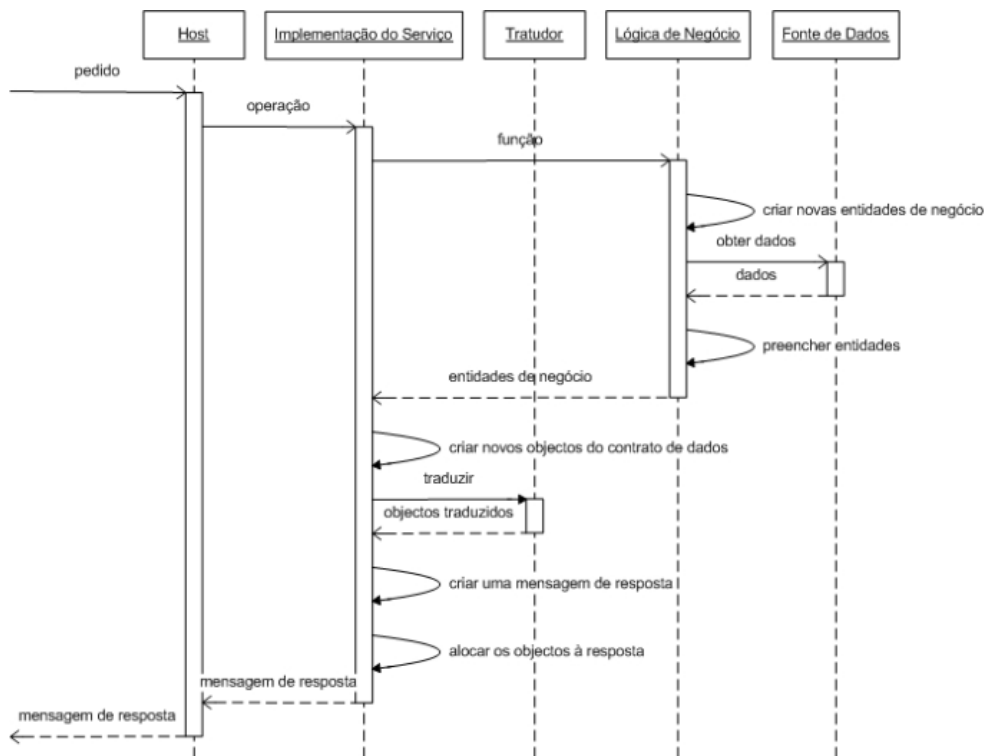


Figura 4.4: Sequência de execução de um serviço.

Nesta perspectiva, a facilidade de manutenção do código e da própria arquitectura assume uma preponderância elevada.

Também numa perspectiva a médio/longo prazo o Service Factory assume uma posição de destaque relativamente ao Gerador CPCHS. O primeiro é uma ferramenta recente com uma grande margem de progressão, com um apoio ao desenvolvimento incomparavelmente maior e com grande flexibilidade à introdução de expansões. Por outro lado, o Gerador CPCHS é uma aplicação com algumas deficiências e com um público-alvo muito restrito. As tecnologias às quais recorre tendem a ser substituídas pelo que a sua margem de progressão é bastante reduzida.

Por todos os motivos referidos a opção recaiu sobre o Service Factory.

4.4 Ferramentas para o Acesso a Dados

Tipicamente, um *webservice* necessita de obter dados que posteriormente são trabalhados e devolvidos ao cliente. A obtenção dos dados pode ser realizada através da chamada a outros serviços, acesso a armazéns de dados ou outras fontes. De forma análoga ao já referido para as ferramentas de criação de *webservices*, existe todo o interesse em alhear o programador de detalhes de implementação de mais baixo nível. O ideal é que seja possível o programador concentrar-se no domínio e na lógica de negócio com o mínimo

de “ruído” proveniente da representação de mais baixo nível da informação bem como da infra-estrutura de suporte. Idealmente, as aplicações interagiriam com o armazenamento persistente da informação no domínio do problema, completamente separado do esquema lógico do armazenamento de baixo nível.

As ferramentas analisadas neste domínio foram o **Gerador CPCHS**, **Repository Factory** e **LINQ**. Segue-se uma descrição sucinta de cada uma das ferramentas, identificação das suas vantagens e desvantagens e também o conceito base do seu funcionamento.

4.4.1 Gerador CPCHS

Tal como já referido anteriormente, a partir do esquema de uma base de dados, o Gerador CPCHS é capaz de gerar todo o código necessário para aceder, interagir e traduzir a informação proveniente do mesmo.

Neste caso não há muito mais a acrescentar à descrição efectuada no ponto 4.3.1 que não seja a identificação de quais as potencialidades desta ferramenta que seriam utilizadas. No caso específico de geração de um meio de acesso a dados é óbvia a inutilidade da geração das classes correspondentes à implementação do serviço e ao *proxy*, contudo, as classes correspondentes às entidades de negócio, regras das entidades de negócio e *scripts* dos pacotes e procedimentos são indispensáveis.

Relativamente a outras alternativas, esta opção tem a vantagem de ser flexível a personalizações ao código gerado assim como adaptado tanto ao SGBD Oracle como ao SQL Server. Do ponto de vista de manutenção de código-fonte, apresenta uma grande debilidade na medida em que uma actualização do modelo de dados implica uma actualização das respectivas entidades e regras de negócio. Este factor é tanto mais crítico quanto maior for o nível de personalização de código pois o volume de linhas de código poderá ser maior ou até a sua própria complexidade.

4.4.2 Repository Factory

O Repository Factory é uma ferramenta que automatiza o processo de criação do código que permite a interacção entre uma aplicação e um ou mais armazéns de dados, portanto, um gerador de código. Numa primeira versão, era parte integrante do Web Service Software Factory, no entanto, pelo facto de este tipo de geração de código não ser necessário exclusivamente no domínio dos serviços, assumiu uma nova faceta como um projecto independente [Cord].

O código gerado pode ser dividido em três grupos, as classes que representam as entidades de negócio (mapeadas directamente das tabelas da base de dados, numa aproximação tabela a tabela), as classes de repositório (que permitem a leitura e escrita das classes de negócio) e os procedimentos. Com o recurso a esta ferramenta, que apresenta uma curva de aprendizagem muito rápida, é possível diminuir o tempo e o esforço necessários

à criação de um modelo de domínio bem como da sua manutenção. O objectivo não é ser uma solução de mapeamento objecto-relacional mas simplesmente um gerador de código bastante leve que automatiza a produção de grande parte do código necessário à construção de um modelo de domínio baseado em objectos, em que estes são persistentes, ou seja, são preservadas as suas características e relações.

No processo de geração de código existe um conceito que interessa tornar bem claro, o conceito de responsabilidade. Existem três responsabilidades a atribuir a projectos, são elas as entidades de negócio, acesso a dados e *host*. Tipicamente, cada uma das responsabilidades é atribuída a um projecto diferente, no entanto é perfeitamente viável que um projecto assuma simultaneamente as responsabilidades das entidades de negócio e de acesso a dados. As entidades de negócio geradas a partir das tabelas são geradas para o projecto com a respectiva responsabilidade. Todas as classes de repositório (uma para cada tabela), bem como as classes com as respectivas operações, são geradas para o projecto com a responsabilidade de acesso a dados. O projecto *host* é o consumidor das funcionalidades disponibilizadas. Atendendo ao exemplo que consta na figura 4.5 da página 43, o primeiro projecto, "PowerCapacity.Bll", assumiu a responsabilidade das entidades de negócio, o segundo projecto, "PowerCapacity.Dal", assumiu a responsabilidade do acesso a dados e, por fim, o projecto "PowerCapacity.UI" consumirá os restantes projectos pois assume a responsabilidade de *host*.

À semelhança do Service Factory, também esta ferramenta possui um conjunto de características que conduz a uma boa estruturação do código e consequentemente de mais fácil manutenção. Entre as características mais importantes encontram-se o facto de o código ser gerado para diferentes pares projectos/pastas (segundo as responsabilidades atribuídas aos projectos) e existir persistência de configurações (guardadas em ficheiros de configuração), o que diminui o tempo dispendido em cada actualização necessária ao longo do processo de desenvolvimento. Na versão analisada, para cada um dos repositórios, é gerada uma interface tornando possível existirem múltiplas implementações de um repositório. Todo o processo de configuração é realizado através de uma interface gráfica bastante intuitiva tornando pouco penosa toda esta fase bastante crucial. As capacidades de "percepção do problema" foram melhoradas (relativamente à versão anterior) na medida em que existe um mapeamento automático entre propriedades das entidades e parâmetros dos procedimentos que, obviamente, podem ser alterados.

Uma outra característica importante é o facto de a sua base de construção ser ao nível do armazém de dados, a base de muitas aplicações. Toda a lógica de negócio, entidades e camada de acesso a dados são construídas à sua volta.

Por muito boa e completa que uma ferramenta deste género seja, é impossível abranger todas as possibilidades e necessidades que o desenvolvimento exige. Neste domínio, o Repository Factory apresenta uma característica bastante interessante. Pelo facto de re-

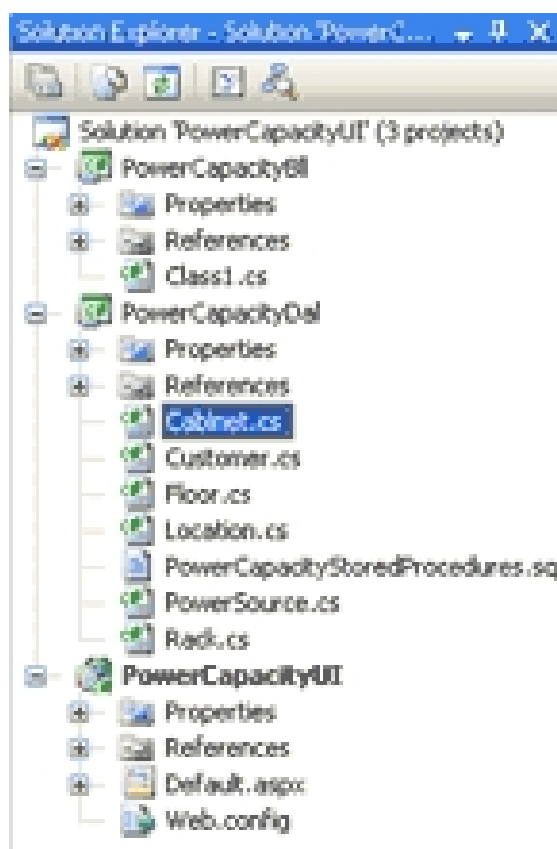


Figura 4.5: Exemplo de uma hierarquia possível de uma solução com código gerado pelo Repository Factory.

correr à Enterprise Library²⁵, é ainda possível implementar chamadas "personalizadas" à base de dados em situações cujo código gerado não seja suficiente [Bro].

Apesar de todas estas características interessantes existem dois pontos negativos associados a esta ferramenta. O primeiro é o facto de estar apenas disponível para o IDE (Integrated Development Environment) Visual Studio 2005, entrando em conflito com a adopção da última versão do Service Factory (para Visual Studio 2008). Após alguma pesquisa em discussões de comunidades virtuais foi possível obter alguns comentários sobre o processo, sendo possível concluir que não possuía a estabilidade exigida para um projecto desta dimensão. O segundo prende-se com o facto de o único SGBD suportado ser SQL Server. Uma das questões pendentes no início do Projecto era qual o SGBD a utilizar (já referido anteriormente), SQL Server ou Oracle. No caso de o cliente optar pelo segundo, inviabilizava a opção pelo Repository Factory. A compatibilidade com o SGBD Oracle era um dos desenvolvimentos em curso identificado pela equipa responsável pelo projecto, existindo indicações de que seguia a bom ritmo [Core].

²⁵<http://www.codeplex.com/entlib>, consultado pela última vez em 24 de Junho de 2008.

4.4.3 Language-Integrated Query (LINQ)

Hoje em dia, o recurso a programação orientada a objectos (OOP – Object-Oriented Programming) é prática comum, quer em ambiente académico, quer em ambiente empresarial, onde conceitos como classe, objecto e método são amplamente aceites. Atendendo às actuais e futuras gerações de tecnologias de programação, torna-se evidente que o próximo grande desafio neste domínio é reduzir a complexidade de aceder e integrar informação que não é, de forma nativa, definida usando tecnologia orientada a objectos, sendo que as fontes mais comuns são bases de dados relacionais e XML [Corc].

O projecto LINQ (Language-Integrated Query) pretende ser a ponte entre estes dois mundos ainda pouco integrados e ainda mais. Em vez de se adicionarem às linguagens de programação capacidades específicas para dados relacionais e XML, o LINQ tem uma abordagem mais geral, onde o objectivo é proporcionar capacidades de consulta de todas as fontes de informação (incluindo informação em memória). O processo de obtenção de dados torna-se transparente para o utilizador na medida em que, independentemente da fonte de dados, são sempre retornados objectos.

A designação do projecto indica mais uma característica bastante interessante, a sua integração com a linguagem de programação primária (por exemplo, Visual C#, Visual Basic). Este facto traz bastantes vantagens, tais como, verificação de sintaxe na compilação ou reconhecimento pelo IntelliSense de propriedades provenientes da fonte de dados, que antes estavam apenas disponíveis no código imperativo. Na figura 4.6 da página 45 é apresentado um diagrama ilustrativo da abrangência do suporte de fontes de informação por parte do LINQ.

Para definir a consulta a executar, o LINQ possui um conjunto de operadores que são aplicáveis a todas as fontes de informação passíveis de serem enumeradas. De forma a expandir ainda mais o poder do LINQ, é possível definir novos operadores apropriados a domínio do problema ou até mesmo, substituir os operadores padrão por operadores personalizados que proporcionem uma melhor adaptação ao problema. Desde que cumpram as convenções do LINQ, qualquer operador definido pelo utilizador usufrui do mesmo tipo de integração com a linguagem de programação primária e suporte da ferramenta de desenvolvimento que qualquer operador padrão.

Dado que os dados são retornados sob a forma de objectos, caso a fonte de informação não suporte este conceito, existe a necessidade de mapear os dados para o domínio dos objectos. Nestes casos, em vez de os operadores da *query* serem executados pelo motor de processamento do LINQ, existe um *provider* específico que implementa um motor próprio ou traduz para um formato diferente de modo a que possa ser executado no suporte da fonte de informação (como é o caso das interrogações SQL).

O *provider* para SQL é denominado LINQ to SQL. Atendendo especificamente a este caso (embora também se verifique noutros casos), é realizado um mapeamento entre ta-



Figura 4.6: Abrangência do suporte de fontes de informação por parte do LINQ.

belas e domínio dos objectos, um Mapeamento Objecto-Relacional²⁶. Esta integração (estando os operadores integrados na tecnologia ADO.NET²⁷) proporciona uma boa conversão em tipos enquanto é mantido o poder expressivo do modelo relacional e o desempenho da avaliação da *query* directamente no armazenamento original.

A forma de mapear a base de dados em objectos é extraordinariamente simples, pois basta indicar a ligação à base de dados e arrastar a tabela pretendida para uma interface gráfica própria da ferramenta, LINQ to SQL Designer Surface. É intrínseco a uma fonte de dados do tipo relacional existirem relações entre os elementos. Aquando da inclusão de uma tabela na interface de mapeamento, as relações entre os elementos são automaticamente detectadas e reflectidas nas estruturas das respectivas classes geradas. Após o processo de mapeamento, os efeitos são imediatos pois as classes correspondentes às tabelas ficam imediatamente disponíveis.

Nesta vertente de acesso a dados não existe geração explícita de código-fonte ou pelo menos o utilizador não tem de conhecer os seus pormenores para poder usufruir das suas funcionalidades [Corb]. A maior abstracção relativamente aos pormenores de armazenamento e a sua adaptação à linguagem de programação utilizada permitem um maior foco no domínio do problema.

²⁶<http://www.agiledata.org/essays/mappingObjects.html>, consultado pela última vez em 29 de Junho de 2008.

²⁷[http://msdn.microsoft.com/en-us/library/h43ks021\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/h43ks021(VS.71).aspx), consultado pela última vez em 29 de Junho de 2008.

O principal inconveniente do LINQ é o facto de suportar apenas motores de base de dados proprietários da Microsoft. Existindo a hipótese de o Projecto recorrer a Oracle, foi realizada alguma pesquisa com o intuito de verificar a existência de soluções para esta incompatibilidade. De facto existiam algumas empresas que disponibilizavam já *providers* para que fosse possível utilizar o LINQ sobre o SGBD Oracle, como por exemplo, a Data Direct²⁸. Contudo, por motivos inerentes à própria CPCHS, não estavam reunidas as condições necessárias para a aquisição de tal software. Durante a pesquisa foi também encontrado um projecto denominado DbLinq²⁹ cujo objectivo é o desenvolvimento de *providers* para MySQL, Oracle e PostgreSQL. A ferramenta não foi testada directamente pois na sua descrição era indicado que o *provider* para Oracle tinha sido desenvolvido para a versão OracleXE solicitando *feedback* acerca do seu desempenho para versões normais de Oracle, portanto, não devidamente testada para as mesmas. Para além deste facto, o planeamento do Projecto ao nível da empresa exigia o início do desenvolvimento.

4.4.4 Conclusões sobre as Ferramentas para o Acesso a Dados

De entre as ferramentas analisadas, o LINQ assume uma posição de destaque quer pela sua versatilidade quer pela margem de progressão que apresenta. O facto de proporcionar a detecção de erros em tempo de compilação é um bom suporte ao desenvolvimento de aplicações mais maduras e menos propensas a erros. A facilidade com que é realizado o mapeamento entre o domínio dos objectos e o domínio relacional, é incomparavelmente superior ao processo de geração de código necessário para construir o acesso a dados com o Gerador CPCHS ou com o Repository Factory.

Apesar de utilização mais complicada e um âmbito muito mais restrito, o Repository Factory proporciona uma estruturação de código bastante interessante e um nível de abstracção do modelo da base de dados razoável.

Tal como referido anteriormente, o Gerador CPCHS é uma ferramenta que apresenta algumas deficiências e cuja utilização obriga a seguir algumas particularidades que denunciam a pouca flexibilidade da ferramenta.

Não atendendo ao suporte a SGBD, a escolha tende claramente para o LINQ, seguindo-se o Repository Factory e finalmente o Gerador CPCHS. Contudo, é precisamente esta característica a mais importante no âmbito do Projecto pois está directamente dependente de uma opção do cliente.

O Gerador CPCHS é a única ferramenta cujo suporte ao SGBD Oracle está plenamente testado pois tem sido a ferramenta utilizada em vários projectos da CPCHS. Por outro lado, o suporte a Oracle pelo Repository Factory e pelo LINQ surgem como projectos independentes do próprio desenvolvimento da ferramenta, com um escasso número de testes realizados e pouco maduros. Para um projecto deste nível é fundamental que as

²⁸<http://www.datadirect.com/index.ssp>, consultado pela última vez em 3 de Julho de 2008.

²⁹http://code2code.net/DB_Linq/, consultado pela última vez em 29 de Junho de 2008.

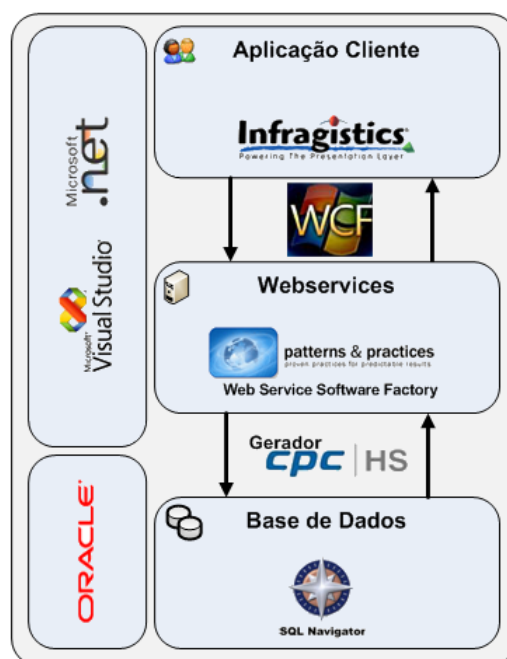


Figura 4.7: Tecnologias e ferramentas utilizadas no Projecto.

ferramentas utilizadas dêem a garantia de estabilidade de forma a serem um bom alicerce da solução. Por estes motivos, e dada a impossibilidade de adquirir um *provider* para Oracle disponibilizado por empresas externas, a escolha acabou por ser o Gerador CPCHS pois o cliente optou pelo SGBD Oracle.

4.5 Soluções Adoptadas

Em suma, foram realizadas várias opções relativamente a tecnologia e ferramentas a utilizar durante o processo de implementação. Como tecnologia de comunicação foi utilizado o **Windows Communication Foundation** sendo a ferramenta **Web Service Software Factory: Modeling Edition** a eleita para construir os *webservices*. A escolha para ferramenta de construção do acesso a dados recaiu sobre o **Gerador CPCHS**. Importa referir que todo o desenvolvimento é baseado na .NET Framework³⁰ da Microsoft e, por decisão do cliente, o SGBD utilizado é o Oracle10gR2³¹.

A figura 4.7 da página 47 ilustra as opções tomadas.

³⁰<http://msdn.microsoft.com/en-us/netframework/default.aspx>, consultado pela última vez em 2 de Julho de 2008.

³¹<http://www.oracle.com/database/index.html>, consultado pela última vez em 2 de Julho de 2008.

Capítulo 5

Descrição do Processo de Implementação do eResults – EDA de MCDT

Neste capítulo são apresentados a especificação e processo de implementação da solução. Na especificação é apresentado, de uma forma genérica, o modelo de desenvolvimento, a identificação dos actores, identificação dos principais casos de uso e quais os seus requisitos funcionais e não funcionais (e respectivos subgrupos). Na descrição da implementação são apresentados os modelos de domínio e de dados, as arquitecturas lógica e física, os vários módulos dos *webservices* (com as respectivas operações e detalhes de implementação), uma secção dedicada à descrição de implementações de menor dimensão e, por fim, a descrição da interface gráfica.

5.1 Estratégia de Desenvolvimento

Todo o desenvolvimento de uma solução consiste num processo evolutivo no qual surgem mais ou menos barreiras, resultando em diferentes níveis de qualidade final da solução. A adopção de um modelo de desenvolvimento é uma boa forma de disciplinar a construção e manutenção de uma solução, contudo não existem fórmulas perfeitas e o ideal é definir uma referência, uma linha orientadora, fazendo pequenas adaptações consoante a natureza e particularidades do projecto.

5.1.1 Características do Processo de Desenvolvimento do eResults

O projecto eResults foi auditado (à data deste relatório ainda decorre o processo) por uma empresa externa, motivo pelo qual existia a necessidade de apresentar um protótipo

funcional em períodos de tempo regulares. Ao longo de todo o processo foram realizadas várias apresentações do produto num crescendo de *keyusers*, ou seja, consoante o avanço do projecto maior era o número de pessoas envolvidas assim como o âmbito das suas responsabilidades. O *feedback* obtido em cada uma das apresentações seria usado como base para melhoramentos a efectuar bem como eventuais novos desenvolvimentos, quer ao nível do modelo de dados quer ao nível da interface gráfica.

No que concerne à equipa de desenvolvimento, verificava-se a impossibilidade de desenvolver todos os módulos em simultâneo devido ao facto de existirem outros projectos que requeriam a atenção dos vários elementos. Por este motivo, o desenvolvimento teria de ser realizado de uma forma assíncrona.

Dada a necessidade de apresentações regulares de um protótipo funcional, o modelo de referência teria de ser iterativo. De entre um conjunto alargado de modelos foram identificados três potenciais referências, modelo em espiral [Boe88], "Feature Driven Development"(FDD) [CLL99] e Scrum [TN86].

De uma forma geral, o modelo FDD é o que se adapta melhor a todo o processo principalmente devido ao seu conceito base, a funcionalidade. O que se pretendia ao longo das várias apresentações era um crescendo do conjunto de funcionalidades estáveis disponíveis (ocorrendo, muito pontualmente, ligeiras adaptações às já existentes). Para além do conceito chave, os outros dois modelos considerados apresentavam algum desfaseamento relativamente ao pretendido. O modelo em espiral tinha como incompatibilidade o facto de, a cada iteração, existir a definição de um novo protótipo o que não era o caso do eResults. Este Projecto, à partida, possuía um conjunto de funcionalidades que iam sendo implementadas para as diferentes apresentações, contudo, a arquitectura estava definida à partida, não existindo a necessidade de novas definições que não fosse de funcionalidades. No que diz respeito ao Scrum, este tinha como inconveniente o facto de, no decorrer de cada *sprint*, ser impossível alterar o respectivo *backlog* (documento de alto nível de todo o projecto). Esta característica colidia com o carácter assíncrono da participação dos vários programadores envolvidos. Caso estivesse a decorrer um *sprint* e fosse detectada uma incompatibilidade existia a necessidade de interromper o mesmo, pois a sua continuidade implicaria a produção de inconsistências funcionais.

5.1.2 Feature Driven Development (FDD)

O FDD é um processo de desenvolvimento de software ágil, iterativo e incremental. Todas as práticas a si inerentes são conduzidas numa perspectiva de disponibilização de funcionalidades com valor para o cliente (*stakeholders*), tendo como objectivo a disponibilização de um produto tangível e funcional com uma periodicidade definida.

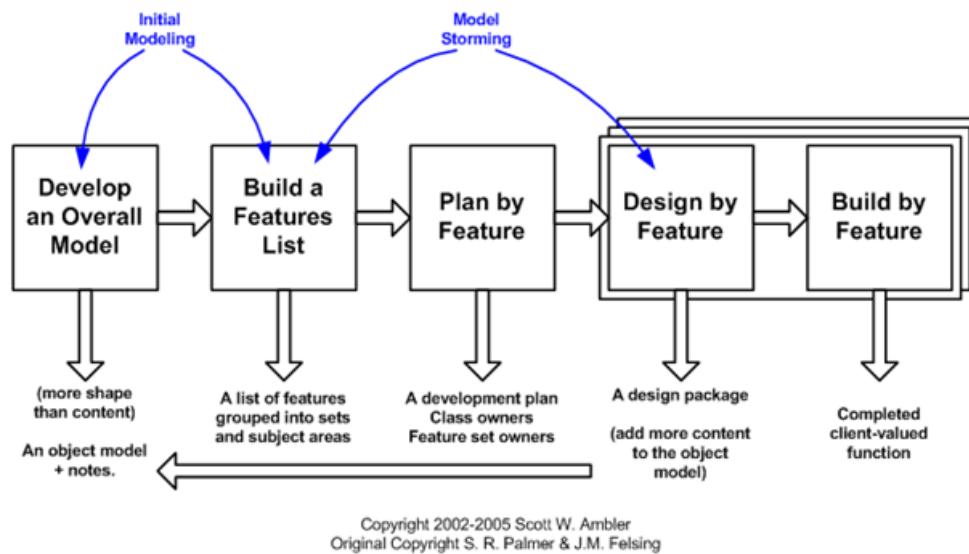


Figura 5.1: Ciclo de vida de um projecto segundo o modelo FDD.

O FDD foi concebido por Jeff DeLuca segundo as necessidades de um projecto para um banco de Singapura em 1997, com a duração de 15 meses e envolvendo 50 pessoas. Após esta experiência o processo foi descrito no livro "Java Modeling In Color with UML", publicado em 1999.

As funcionalidades consideradas e desenvolvidas devem apresentar mais-valias para o cliente, ou seja, representar um benefício tangível para o negócio. Normalmente, uma funcionalidade está associada a um caso de uso pois são estes que representam as necessidades do cliente no âmbito de um produto. Para além de o conceito de mais-valia permitir identificar quais as funcionalidades a desenvolver, através da sua importância, é possível definir uma hierarquia de prioridades de desenvolvimento [Cau].

O processo de elaboração da lista de funcionalidades é deveras importante pois vai orientar todas as fases subsequentes. Por este motivo, é incontornável a importância da participação de todos os envolvidos no projecto, conduzindo a que todos possuam uma perspectiva comum do mesmo.

Sendo um processo de desenvolvimento ágil, tem como objectivo evitar uma longa espera até que alguns resultados sejam visíveis. Cada funcionalidade deverá ser possível de concretizar num período de 1 a 10 dias, contudo, verifica-se que a generalidade das mesmas requer um período de 1 a 3 dias para a sua concretização. Caso exista a previsão da necessidade de um período que ultrapasse os 10 dias, então a funcionalidade deve ser dividida em objectivos menores.

Atendendo ao processo propriamente dito, este é composto por cinco actividades principais que são percorridas de forma iterativa, tal como é possível verificar na figura 5.1 da página 51.

A primeira fase consiste na elaboração de um modelo de alto-nível e um conjunto de notas orientadoras que pode ser revisto ao longo do processo de forma a reflectir o produto em construção. O objectivo principal é identificar e compreender claramente os conceitos fundamentais do domínio do problema em causa.

Percebido o domínio do problema, importa identificar as funcionalidades pretendidas compilando-as numa lista organizada por conjuntos relacionados, de preferência elaborada com a presença de todos os intervenientes-chave.

O passo seguinte consiste no planeamento, mais uma vez orientado à funcionalidade. Nesta fase, é suposto serem atribuídas responsabilidades de desenvolvimento a funcionalidades específicas ou a conjuntos das mesmas.

As duas fases finais incluem tarefas como a definição de um modelo mais detalhado no âmbito de uma funcionalidade ou conjunto das mesmas, o desenvolvimento de código-fonte, testes e integração no pacote já existente do produto [[Amb](#)].

5.1.3 Adaptações realizadas ao Método FDD no âmbito do eResults

Tal como referido anteriormente, este é o modelo de desenvolvimento de software que mais se adapta às características do Projecto e não o modelo seguido rigorosamente. Por este motivo, um dos principais pressupostos do modelo não pode ser cumprido, a envolvimento de todos os intervenientes no momento da definição do modelo global. A dispersão dos vários elementos por outros projectos obrigou a que os artefactos produzidos nas duas primeiras fases sofressem alterações mais profundas que o desejável, em cada iteração. Em vez de sofrerem pequenas correcções de forma a reflectir fielmente o produto desenvolvido, correspondiam, algumas vezes, a uma nova definição (mas nunca a definição de um novo protótipo). Este desfasamento provocou que o produto sofresse duas reformulações, a primeira aquando da disponibilidade da equipa responsável pelas aplicações que geram os dados apresentados pelo eResults e a segunda correspondeu à constituição da equipa responsável pelo desenvolvimento do processo de comunicação entre aplicações externas e o eResults, via HL7 (Health Level 7)¹.

5.2 Identificação dos Actores

Numa primeira fase da implementação da solução foram identificados os grupos de actores que interagirão com o módulo EDA, médico, gestor de sistema e módulo/aplicação externa. Com esta primeira identificação, tornou-se mais fácil definir as funcionalidades comuns e características de cada um dos grupos. Se considerarmos todos os módulos do eResults existe ainda o actor administrativo, contudo, visto não interagir directamente com o módulo de EDA de MCDT, não será alvo de análise.

¹<http://www.hl7.org/>, consultado pela última vez em 2 de Julho de 2008.

5.2.1 Médico

Tal como o próprio nome indica, neste grupo enquadram-se os elementos do pessoal hospitalar com responsabilidade clínica sobre os pacientes, os médicos. Este grupo possui um conjunto alargado de acções possíveis pois a solução está direccionada para este grupo.

5.2.2 Gestor de Sistema

Este tipo de actor poderá aceder às mesmas funcionalidades que o actor médico no âmbito da sua tarefa de monitorização do sistema.

5.2.3 Módulo/Aplicação Externa

Neste grupo enquadram-se todos os módulos do eResults ou aplicações externas que interajam com o módulo de EDA de MCDT.

5.3 Casos de Uso

Para cada um dos grupos de utilizadores foram identificados os casos de uso típicos. Para melhor distinção e compreensão, os casos de uso foram divididos segundo o módulo a que pertencem, Exploração de Dados Analíticos (figura 5.2 da página 54) ou Filtros Globais (figura 5.3 da página 55). No anexo A, a título de complemento, são identificados os actores, pré-condições, pós-condições e pressupostos de cada um dos casos de uso.

5.3.1 Casos de Uso do Módulo Exploração de Dados Analíticos

5.3.1.1 Solicitar visualização de uma requisição pelo identificador do documento

Existe a possibilidade de outros módulos do eResults solicitarem ao módulo de EDA de MCDT a visualização dos dados analíticos de uma determinada requisição indicando para tal o identificador do respectivo documento.

5.3.1.2 Solicitar visualização de uma requisição pela referência do documento

Existe a possibilidade de aplicações externas ao módulo de EDA de MCDT solicitarem a visualização dos dados analíticos de uma determinada requisição indicando para tal a referência do respectivo documento.

5.3.1.3 Pesquisar requisições segundo critérios de filtragem

Tipicamente, a pesquisa de requisições e respectivos resultados é realizada com base num conjunto de critérios referentes ao doente, episódio e documento. Pode ser realizada

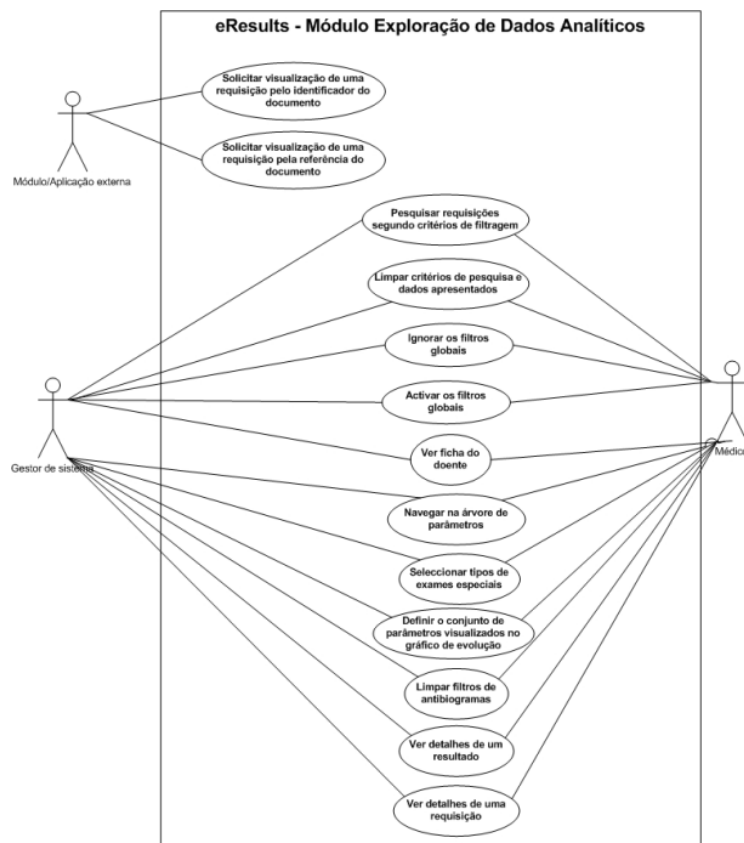


Figura 5.2: Diagrama de casos de uso do módulo EDA.

qualquer combinação de critérios, no entanto tem de estar preenchido pelo menos um critério.

5.3.1.4 Limpar critérios de pesquisa e dados apresentados

Após a visualização dos resultados de um doente, caso se pretenda realizar uma nova pesquisa, é conveniente limpar os critérios de pesquisa de forma a não ficar definido nenhum critério da pesquisa anterior que condicione a actual.

5.3.1.5 Ignorar os filtros globais

Apesar de as configurações dos filtros globais estarem activas na sessão, o utilizador tem a hipótese de as ignorar por um tempo indeterminado durante a mesma sessão. Para tal, indica explicitamente a sua intenção através de um clique no botão "Ignorar filtros".

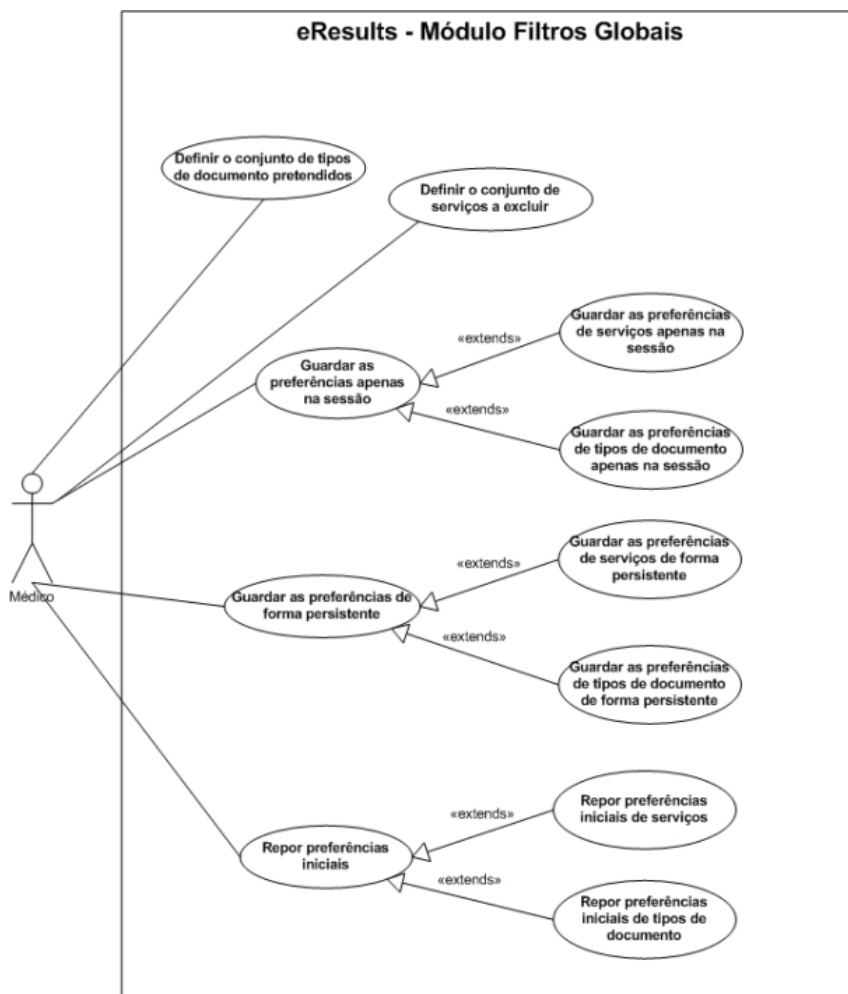


Figura 5.3: Diagrama de casos de uso do módulo Filtros Globais.

5.3.1.6 Activar os filtros globais

Após a indicação para ignorar os filtros globais, o utilizador tem a hipótese de voltar a activar as configurações definidas nos filtros globais. Para tal indica explicitamente a sua intenção através de um clique no botão "Activar filtros".

5.3.1.7 Ver ficha do doente

Para além dos dados acerca do doente considerados como critérios de pesquisa, o utilizador pode visualizar uma ficha mais completa. Para tal deve clicar no botão "Ver ficha".

5.3.1.8 Navegar na árvore de parâmetros

Após a apresentação dos resultados de uma ou mais requisições, e caso a hierarquia possua mais do que um nível hierárquico, o utilizador pode esconder e mostrar os vários níveis que não o primeiro. Desta forma, pode comparar mais facilmente os valores de alguns parâmetros.

5.3.1.9 Seleccionar tipos de exames especiais

Após a apresentação dos resultados de uma ou mais requisições, as células da tabela global correspondentes aos resultados que possuem especificidades (microrganismos e gráficos) são identificados com uma imagem esclarecedora. Caso o utilizador pretenda, poderá navegar até aos detalhes do mesmo através de um clique na respectiva célula, sendo redireccionado para a interface adequada.

5.3.1.10 Definir o conjunto de parâmetros visualizados no gráfico de evolução

No caso específico dos resultados numéricos, existe uma interface própria para a visualização da sua evolução e possíveis correlações entre valores. Do conjunto de parâmetros cujos resultados são numéricos, através de uma árvore hierárquica de parâmetros com comportamento *tri-state* e expansível, o utilizador pode seleccionar quais os parâmetros a incluir no gráfico.

5.3.1.11 Limpar filtros de antibiogramas

Após o clique numa célula da tabela global correspondente a um microrganismo, o utilizador é redireccionado para a interface de antibiogramas onde constam apenas os respectivos registos. No caso de o utilizador pretender voltar a visualizar todos os antibiogramas de todas as requisições poderá fazê-lo através de um clique no botão "Limpar filtros".

5.3.1.12 Ver detalhes de um resultado

Em qualquer uma das interfaces de resultados (tabela global, evolução e micro) o utilizador tem a possibilidade de aceder a informação mais detalhada sobre um determinado resultado (no caso da tabela global e da micro, uma célula da tabela; no caso da evolução, um ponto do gráfico) através de *tooltips*. Para aceder aos detalhes basta posicionar o cursor do rato sobre o resultado pretendido durante um determinado período de tempo.

5.3.1.13 Ver detalhes de uma requisição

Nas interfaces tabela global e micro, o utilizador tem a possibilidade de aceder a informação mais detalhada sobre uma determinada requisição através de *tooltips*. No caso da tabela global, cada coluna corresponde a uma requisição enquanto que, no caso da tabela micro, cada coluna corresponde a um par requisição/microrganismo. Em qualquer um dos casos, para aceder aos detalhes basta posicionar o cursor do rato sobre o cabeçalho da coluna que representa a requisição pretendida durante um determinado período de tempo.

5.3.2 Casos de Uso do Módulo Filtros Globais

5.3.2.1 Definir o conjunto de tipos de documento pretendidos

Os vários locais, aplicações e tipos de documento são apresentados numa árvore com comportamento *tri-state* e expansível. Existem ainda dois botões, um para seleccionar todos os nós da árvore e outro para remover todas as selecções existentes. Com estas três formas de interagir com a árvore o utilizador pode definir os tipos de documento pesquisáveis. Caso o utilizador já possua configurações guardadas anteriormente os respectivos tipos de documento surgem já seleccionados.

5.3.2.2 Definir o conjunto de serviços a excluir

Todos os serviços que constam no sistema são apresentados em caixas de selecção distintas com capacidade de selecção de múltiplas linhas, duas para o papel de serviço requisitante (disponíveis e excluídos) e outras duas para o papel de serviço executante (também disponíveis e excluídos). Em qualquer uma das situações o utilizador tem ao seu dispor quatro botões que lhe permitem efectuar trocas entre serviços disponíveis e serviços excluídos. Caso o utilizador já possua configurações guardadas anteriormente, os serviços excluídos de cada um dos papéis devem surgir na caixa de selecção respectiva.

5.3.2.3 Guardar as preferências apenas na sessão

Após as alterações às preferências (tipos de documento e serviços) o utilizador pretende que as definições actuais afectem apenas a sessão que está a decorrer.

5.3.2.4 Guardar as preferências de forma persistente

Após as alterações às preferências (tipos de documento e serviços) o utilizador pretende que as definições actuais sejam guardadas de forma permanente (para as sessões actual e futuras).

5.3.2.5 Repor preferências iniciais

O utilizador pretende repor as configurações de filtros globais verificadas no início da sessão actual.

5.4 Requisitos da Solução

A qualidade final de uma solução depende de vários factores verificados ao longo do seu processo de desenvolvimento. Uma das primeiras fases é a definição dos requisitos, portanto, uma fase basilar. A eliminação de erros cometidos nesta fase torna-se cada vez mais difícil e dispendiosa à medida que o sistema avança para iterações posteriores do seu ciclo de vida. Atendendo a estes factores, é possível concluir que uma correcta definição de requisitos pode evitar alguns dissabores no futuro. Nesta secção pretende-se identificar quais os requisitos funcionais e não funcionais dos módulos em análise.

5.4.1 Requisitos Funcionais

Os requisitos funcionais dos módulos EDA e Filtros Globais foram já identificados nos casos de uso. Cada caso de uso corresponde a uma funcionalidade que se pretende ver implementada.

5.4.2 Requisitos Não-Funcionais

Os requisitos não-funcionais agregam todos os requisitos da solução que não correspondem a funcionalidades. Segue-se a sua identificação.

5.4.2.1 Requisitos de Qualidade

Atendendo ao facto de que uma falha ou execução incorrecta do sistema pode afectar negativamente vidas humanas, a solução insere-se no grupo de sistemas críticos. Por este motivo, existe todo o interesse em analisar os requisitos de qualidade mais significativos nesta área bem como outros que proporcionem uma melhor experiência de utilização.

Apesar de o foco serem os módulos EDA e Filtros Globais, os requisitos a seguir identificados são igualmente aplicáveis à solução na sua generalidade. A ordem de apresentação não representa qualquer hierarquia de prioridades ou importância.

Acessibilidade: A solução deverá ter todas as suas funcionalidades e suporte em língua Portuguesa. No caso de as funcionalidades estarem descritas por uma imagem, se o utilizador colocar o cursor do rato sobre as ligações dessas mesmas funcionalidades os seus nomes/descrições deverão surgir também em língua Portuguesa.

Coerência de Funcionamento: Os dados apresentados como resultados das pesquisas devem ser coerentes com os vários critérios de pesquisa definidos.

Disponibilidade: No caso particular de SI na área da Saúde, atendendo à gravidade das consequências, a fiabilidade é preferível à disponibilidade. No entanto, um nível de disponibilidade a rondar os 100% é o ideal. O objectivo da solução é auxiliar o pessoal médico na identificação e acompanhamento de patologias, motivo pelo qual a indisponibilidade do sistema pode influenciar o rendimento do pessoal médico, prejudicando indirectamente os pacientes.

Eficiência: O sistema deverá ter um tempo de resposta às acções realizadas pelo utilizador reduzido (2 a 3 segundos). Sempre que seja inevitável um período de espera superior a este limite, a interface deve fornecer informação ao utilizador acerca do progresso da operação. Atendendo ao facto que a informação estará armazenada num repositório para o efeito, também o desempenho deste deve ser tomado em conta.

Experiência de Utilização: Pretende-se que a solução seja desenhada tendo como objectivo permitir uma boa experiência de utilização. A disposição dos dados deve ser estruturada de forma a promover o reconhecimento imediato da informação pretendida. A curva de aprendizagem deve ser o mais baixa possível. Para tal deve ser realizada uma aposta no reconhecimento dos passos a efectuar em detrimento da memorização. Ao longo de todas as interfaces, deve ser fornecida ao utilizador informação acerca do estado das operações ou da forma de as iniciar. Sempre que possível, a interface deve informar quais as operações disponíveis no estado actual de execução. O utilizador deve sentir que controla a aplicação e não o contrário. No caso de ocorrer um erro de utilização, o sistema deve ser capaz de esclarecer o utilizador da forma correcta de o fazer e recuperar da situação de erro.

Fiabilidade: A solução deve ter todas as suas funcionalidades disponíveis em qualquer momento, não sendo consideradas para o caso falhas exteriores à mesma. A robustez da aplicação deverá ser o mais apurada possível, evitando perda ou incoerência de informação. Neste aspecto, a apresentação de dados errados poderá ter consequências mais gravosas do que a própria não apresentação, na medida em que poderá ser realizado um diagnóstico e aplicadas terapêuticas desajustadas, com prejuízo da própria vida do paciente.

Manutenção: A manutenção dos dados inseridos no sistema fica a cargo do cliente, eventualmente com o apoio da CPCHS, caso a situação o justifique. No que respeita a manutenção na vertente técnica, a actualização do sistema para correcções de eventuais

falhas do sistema é da responsabilidade da CPCHS. Exclui-se do apoio prestado questões relativas a hardware.

Segurança: A segurança deverá ser um pilar da solução, quer pela necessidade de controlo de acessos afim de proteger a informação de acessos indevidos, quer pela precisão dos dados apresentados.

Suporte: Deverá existir suporte relativamente ao produto desenvolvido assim como a possibilidade de os utilizadores reportarem eventuais erros ou dificuldades de utilização.

5.4.2.2 Requisitos de Interface Externos

Nesta secção são apresentados os requisitos de interface com elementos externos que a solução deverá proporcionar.

Interface de comunicações: O ambiente de execução típico da solução será uma *intranet* na unidade hospitalar. Para que o funcionamento seja possível, deve existir uma rede de suporte, ligando todos os possíveis componentes físicos da aplicação, nomeadamente, posto de trabalho, aplicação cliente, servidor aplicacional e servidor de dados. Eventualmente, todos estes papéis podem concentrar-se numa única máquina, contudo, continua a ser necessária a rede de suporte para que as aplicações externas possam enviar os resultados produzidos.

Interface de hardware: A versatilidade da arquitectura da solução permite diferentes tipos de distribuição física da mesma, tendo como implicação directa o hardware necessário. Contudo, do ponto de vista do utilizador final (localizado num posto de trabalho) este necessita apenas de um computador com os respectivos periféricos comuns (monitor, teclado e rato).

Interface de software: No que diz respeito a software, para a execução da aplicação é necessário um *browser*. As versões suportadas são Internet Explorer 6 ou 7. Dada a natureza do componente de pesquisa de doentes, é necessário que a máquina cliente possua o Silverlight 2.0 Beta 2 instalado.

Interface com utilizadores: Para a visualização de resultados e gestão de filtros globais os utilizadores interagirão com a aplicação através de um *browser*. Cada um dos módulos será disponibilizado numa página *Web* diferente, sendo que a navegação é realizada através de um menu comum.

5.4.2.3 Requisitos de Ambiente

Os módulos em questão não exigem bastantes recursos da máquina cliente pelo que uma máquina com 1GB de RAM e um processador com 2GHz possui capacidades suficientes para executar a aplicação. Ao nível da(s) máquina(s) servidor as características exigidas variam consoante o conjunto de aplicações instaladas e qual o volume e picos de acessos. Dada a importância da rede para o bom funcionamento da solução devem ser considerados os tempos de latência da mesma, que se pretendem baixos. Deve ser considerado o volume de tráfego na rede para cada situação particular de execução.

5.4.2.4 Requisitos de Utilizadores

Para interacção com o sistema não são necessários conhecimentos técnicos avançados de informática. A interface deve ser desenhada de forma a ser simples, eficaz e eficiente, pelo que conhecimentos básicos de utilização de TI são suficientes. Já no domínio médico, exigem-se conhecimentos avançados para a correcta interpretação dos resultados.

5.5 Modelo de Domínio

Um dos pontos basilares para uma melhor compreensão do problema e início do desenvolvimento da solução é a identificação das entidades que constituem o domínio do mesmo. O objectivo desta identificação não é compilar um conjunto exaustivo de entidades e suas propriedades mas apenas identificar as entidades principais e perceber as suas relações. Segue-se a descrição dos modelos de domínio de cada um dos módulos em análise.

5.5.1 Exploração de Dados Analíticos

A figura 5.4 da página 62 apresenta as entidades identificadas para o módulo de EDA.

A entidade base de todo o domínio é o paciente pois é a partir desta que todas as outras fazem sentido. Um paciente pode realizar várias visitas a uma unidade hospitalar (episódio), por exemplo, consulta externa, urgência, cirurgia, etc. Em cada uma dessas visitas pode ser necessária a observação do paciente por várias especialidades em diferentes serviços, ou seja, vários detalhes da visita. Em cada um dos serviços/especialidades podem ser requisitados vários MCDT, originando vários documentos, tais como análises patológicas (onde cada requisição corresponde a um documento), radiografias, vídeos de cirurgia, etc. Entre os vários documentos pode ser estabelecida uma hierarquia, dando origem a documentos simples e documentos agregadores. Um documento agregador não é mais do que um agrupador de outros documentos agregadores e/ou simples. Por sua vez, um documento simples possui MCDT associados. No âmbito dos resultados

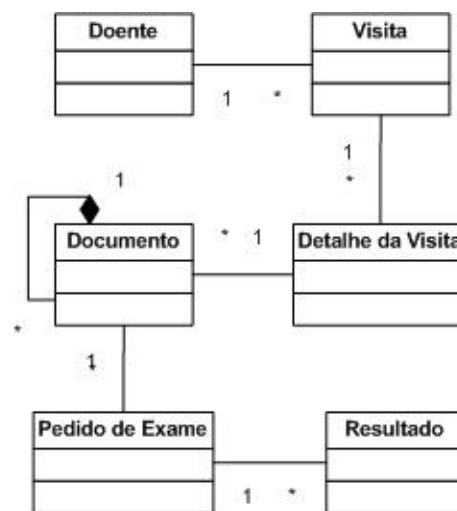


Figura 5.4: Modelo de domínio do módulo EDA.

analíticos, um documento (requisição) pode conter um ou mais pedidos. Por exemplo, numa requisição pode ser pedida a avaliação dos parâmetros "Hemoglobina" e "Ureia", sendo cada um deles um pedido de exame diferente. Após a realização do exame, um pedido origina um ou mais resultados.

5.5.2 Filtros Globais

A figura 5.5 da página 63 apresenta as entidades identificadas para o módulo de Filtros Globais.

Para que o modelo de domínio dos filtros globais faça sentido, a primeira entidade a considerar é a própria instituição. Tipicamente, a solução contemplará apenas uma instituição, no entanto, nada impede que sejam consideradas várias instituições independentes. Uma instituição é composta por um ou mais locais, no caso concreto do Centro Hospitalar de Vila Nova de Gaia (instituição), este é composto por dois locais. Em cada um dos locais podem existir diferentes aplicações instaladas, sendo que cada uma delas pode funcionar em vários locais. Considerando uma aplicação, esta pode produzir vários tipos de documento. De uma forma análoga a situação dos documentos, também os tipos de documento podem possuir uma hierarquia. A entidade serviço não possui nenhuma relação com as restantes identificadas neste caso particular dos filtros globais.

5.6 Modelo de Dados

Após a análise do modelo de domínio, importa apresentar o esquema do armazém de dados de forma a perceber como o primeiro foi mapeado e como se interligará a informação relativa a cada entidade. Dada a dimensão do modelo, serão apresentadas

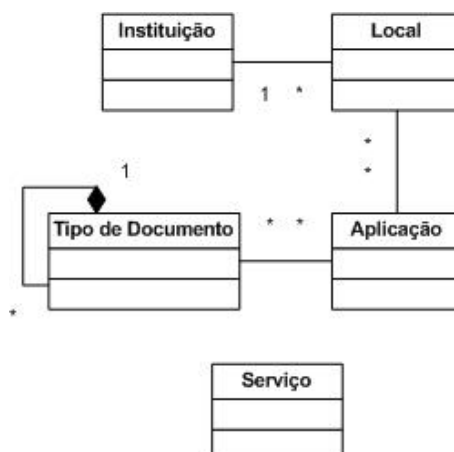


Figura 5.5: Modelo de domínio do módulo Filtros Globais.

apenas duas porções do mesmo, a primeira relativa à EDA e a segunda apresenta apenas as tabelas consultadas no âmbito dos filtros globais.

5.6.1 Exploração de Dados Analíticos

As figuras 5.6 da página 64 (relativa a entidades) e 5.7 da página 65 (relativa a exames e resultados) apresentam o conjunto de tabelas necessárias para o armazenamento da informação relativa a resultados analíticos, desde o doente até aos resultados. Os diagramas são acompanhados por uma descrição muito sucinta de cada tabela.

GR.ENTIDADE

Tabela responsável por guardar informação relativa a cada uma das entidades codificadas no sistema como é o caso de um doente.

GR.DOENTE

Tabela que resulta da particularização de uma entidade para o caso de um doente, ou seja, é responsável por armazenar a informação característica de um doente.

GR.COD_SEXO

Tabela de codificação do sexo do doente.

GR.COD_ESTADO_CIVIL

Tabela de codificação do estado civil do doente.

GR.VISITA

O objectivo desta tabela é guardar todos registos de episódios de um doente.

GR.COD_T_EPISODIO

Tabela de codificação do tipo de episódio de uma visita.

GR.VISITA_DET

Armazena todos os detalhes de uma visita.

GR.VISITA_DET_DOCUMENTO

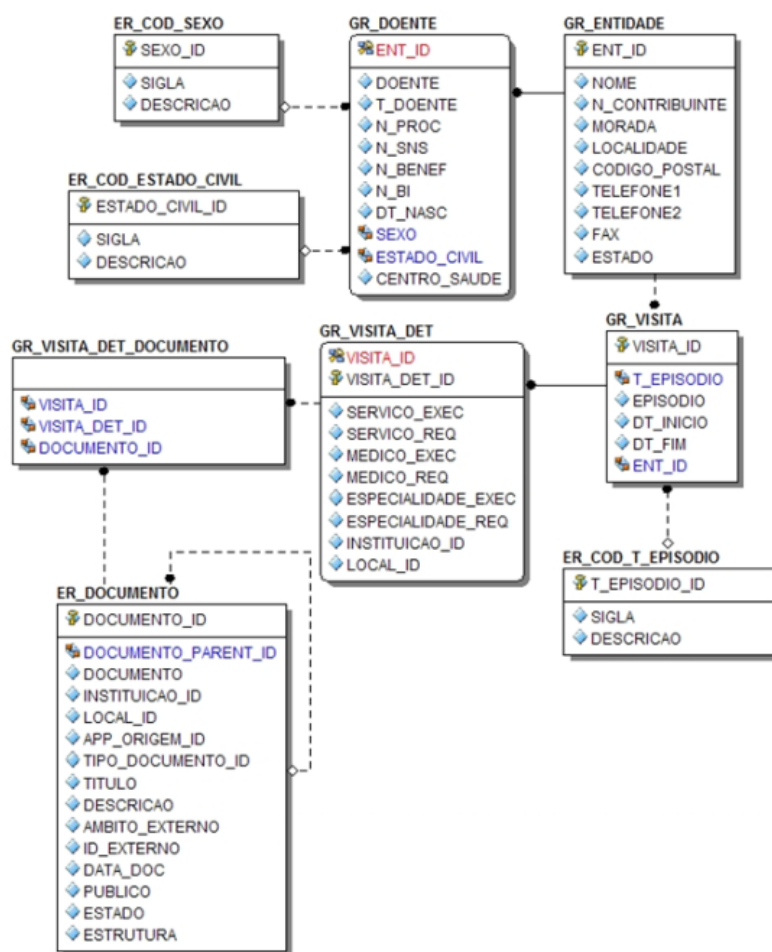


Figura 5.6: Modelo de dados relativo às entidades intervenientes na EDA.

Consiste numa tabela de ligação entre o detalhe de uma visita e os respectivos documentos.

ER.DOCUMENTO

Responsável por guardar informação relativa a um documento, sendo possível existir uma hierarquia dos mesmos mapeada na tabela. No caso dos resultados analíticos, é nesta tabela que é guardada informação sobre uma requisição.

ER.ELEMENTO

Uma requisição é composta por um ou mais pedidos de análises simples, complexas ou perfis. É nesta tabela que os pedidos são registados, sendo realizada a respectiva ligação ao documento e controladas as versões dos resultados.

ER.MCDT

Toda a codificação relativa a um parâmetro, como por exemplo, Hemoglobina ou Ureia, encontra-se nesta tabela. Assume particular importância pois as codificações existentes em aplicações externas que exportem os seus dados para o eResults têm de estar repre-

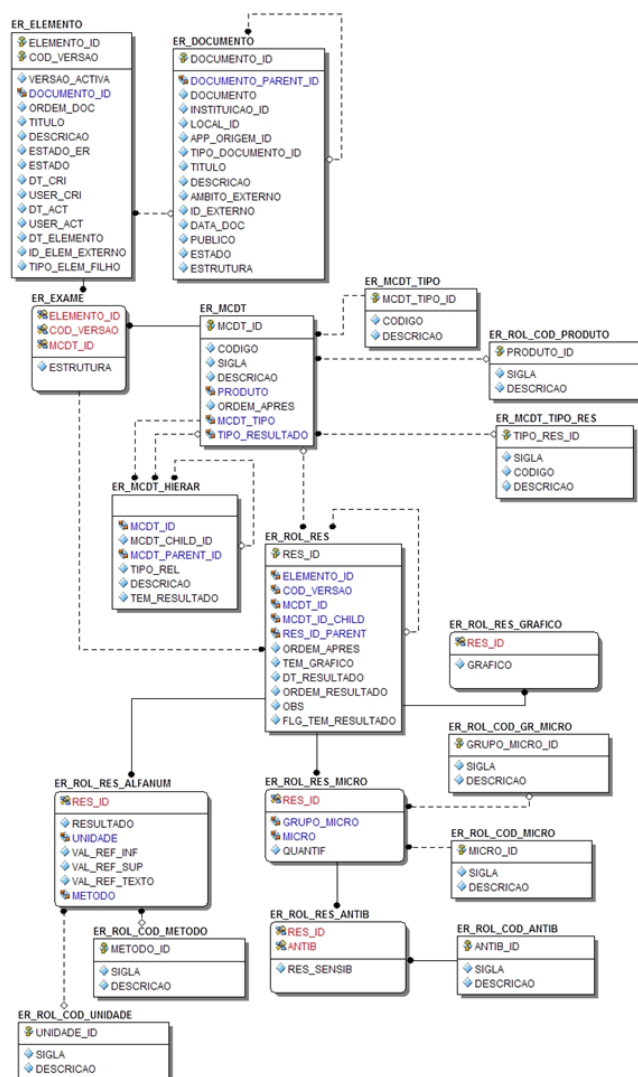


Figura 5.7: Modelo de dados relativo aos exames e resultados intervenientes na EDA.

sentadas nesta tabela.

ER_EXAME

Consiste numa tabela de ligação entre um elemento e a codificação do respectivo exame pedido. O campo "ESTRUTURA" possui a particularidade de guardar código XML que representa a hierarquia superior do exame pedido. Quando é inserido um novo registo nesta tabela é activado um trigger que, consultando a tabela de hierarquia de exames, cria tantos nós quantos os ascendentes do exame pedido.

ER_MCDT_HIERAR

Nesta tabela são estabelecidas todas as relações hierárquicas existentes entre todos os exames codificados (simples, complexas, subgrupos, grupos e perfis).

ER_MCDT_TIPO_RES

Tabela de codificação dos tipos de resultados possíveis (actualmente, alfanuméricos, microrganismos e gráficos).

ER_MCDT_TIPO

Tabela de codificação do tipo de MCDT (actualmente são suportadas apenas análises clínicas, contudo, futuramente poderá ser necessário suportar outros tipos de exame).

ER_ROL_COD_PRODUTO

Tabela de codificação de tipos de produto (sangue, urina, etc.).

ER_ROL_RES

A existência desta tabela é fundamental pois é nela que se define toda a hierarquia de exames pedidos, seus exames filhos e os respectivos resultados.

ER_ROL_RES_ALFANUM

Tabela dedicada a guardar todos os resultados alfanuméricos, consistindo numa "especialização" da tabela ER_ROL_RES.

ER_ROL_COD_UNIDADE

Tabela de codificação de unidades dos resultados alfanuméricos.

ER_ROL_COD_METODO

Tabela de codificação do método utilizado para avaliar o parâmetro cujo resultado é do tipo alfanumérico.

ER_ROL_RES_MICRO

Tabela dedicada a guardar todos os resultados do tipo microrganismo, consistindo numa "especialização" da tabela ER_ROL_RES.

ER_ROL_COD_GR_MICRO

Tabela de codificação dos grupos de microrganismos.

ER_ROL_COD_MICRO

Tabela de codificação dos nomes de microrganismos.

ER_ROL_RES_ANTIB

Tabela dedicada a guardar todos os resultados de antibiogramas associados a um resultado do tipo microrganismo.

ER_ROL_COD_ANTIB

Tabela de codificação dos nomes de antibióticos.

ER_ROL_RES_GRAFICO Tabela dedicada a guardar todos os resultados gráficos, consistindo numa "especialização" da tabela ER_ROL_RES.

5.6.2 Filtros Globais

A figura 5.8 da página 67 apresenta o conjunto de tabelas necessárias para o armazenamento da informação relativa a instituições, locais, aplicações, tipos de documento e

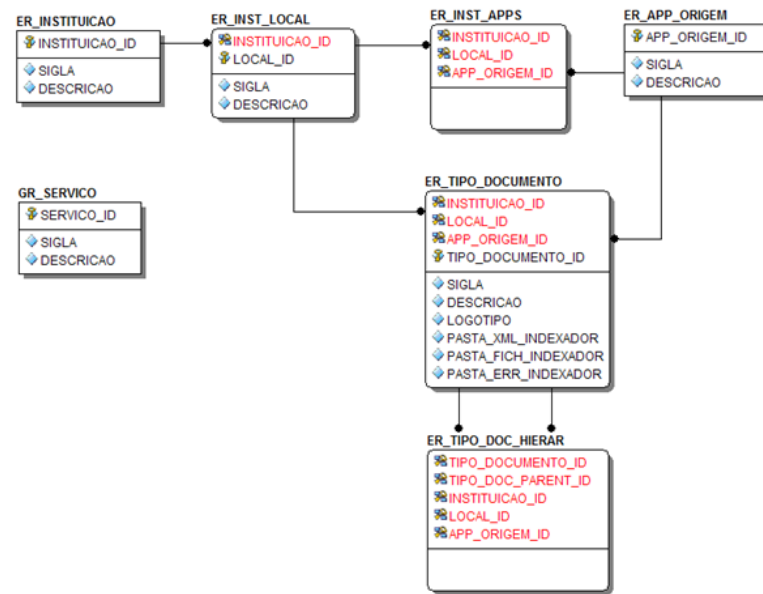


Figura 5.8: Modelo de dados relativo aos filtros globais.

serviços. Todos estes dados são necessários ao módulo de filtros globais. O diagrama é acompanhado por uma descrição muito sucinta de cada tabela.

R.INSTITUICAO

Armazena toda a informação de codificação de instituições representadas no sistema.

ER.INST.LOCAL

Armazena toda a informação de codificação de locais representados no sistema, estabelecendo a ligação com a instituição a que pertence.

ER.APP.ORIGEM

Armazena toda a informação de codificação de aplicações geradoras de informação representadas no sistema.

ER.INST.APPS

Estabelece a ligação entre aplicações de origem e locais. Desta forma é possível identificar quais as aplicações instaladas em cada local.

ER.TIPO.DOCUMENTO

Armazena toda a informação de codificação de tipos de documento gerados pelas várias aplicações.

ER.TIPO.DOC.HIERAR

Nesta tabela são estabelecidas todas as relações hierárquicas existentes entre todos os tipos de documento.

GR.SERVICO

Armazena toda a informação de codificação de serviços representados no sistema.

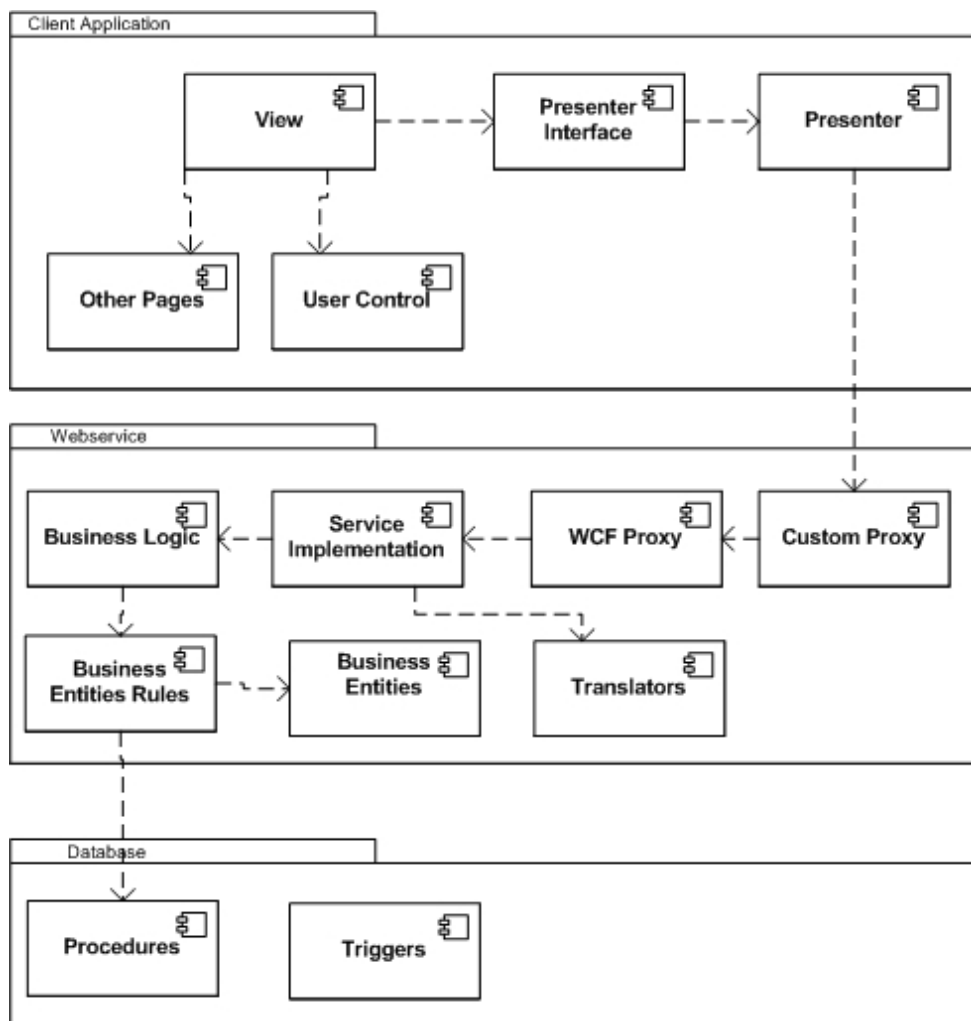


Figura 5.9: Arquitectura lógica da solução.

5.7 Arquitectura Lógica

Como já foi referido anteriormente, a arquitectura da solução desenvolvida é orientada ao serviço, recorrendo à implementação de *webservices*. A enorme flexibilidade proporcionada permite que o desenvolvimento da aplicação consumidora do serviço e da aplicação fornecedora do mesmo seja independente, desde que seja respeitado o formato da informação a trocar.

De uma forma geral, a solução é constituída por três camadas, a aplicação consumidora, responsável pela apresentação dos dados solicitados pelo utilizador, os *webservices*, responsáveis por obter os dados e implementar a lógica de negócio e o armazém de dados. Internamente, também na aplicação, existe um papel bem definido para cada componente, tornando o funcionamento da mesma mais flexível e de fácil actualização, caso existam alterações a efectuar. A figura 5.9 da página 68 apresenta a arquitectura lógica da solução.

O componente identificado como *view* contém apenas o código necessário à inclusão de vários controlos e o respectivo aspecto, como por exemplo, etiquetas, botões, grelhas, etc. Caso exista a necessidade de realizar algumas tarefas na máquina cliente, a inclusão do respectivo código é realizada neste componente. Eventualmente, uma *view* pode ser composta por várias outras *views* e por *user controls*, como é o caso das barras de pesquisa.

O *presenter* é responsável por todas as restantes operações que permitem a visualização da informação solicitada pelo utilizador, nomeadamente, a aquisição de dados, tarefas de inicialização, acções necessárias após a detecção de um determinado evento, etc. O código que constitui o *presenter* podia estar incluído no *code behind* da *view* possibilitando o seu funcionamento. Contudo, esta divisão permite uma estruturação mais clara e independente.

A *view* inclui e trabalha o aspecto dos vários elementos (embora algumas configurações de aspecto sejam realizadas também pelo *presenter*) e o *presenter* é responsável pela recepção dos dados provenientes do(s) *web service(s)* e por preencher os elementos com esses mesmos dados. Para garantir a comunicação entre *presenter* e *view*, é definida uma interface que garante que a *view* disponibiliza os métodos de acessos aos seus elementos e respectivos eventos.

No que diz respeito ao *web service*, a arquitectura implementada é bastante fragmentada, estando as tarefas de cada componente claramente definidas. O componente de entrada no serviço é o *custom proxy*. Este componente assume esta designação pois o código que inclui não é gerado automaticamente. De entre todo o código gerado através do Service Factory, existe um componente também designado de *WCF proxy* e responsável por facultar o acesso ao *web service*.

Esta facilidade de geração de código colocou um problema no âmbito deste Projecto. Dado que o ambiente de execução é multi-empresa, existe a necessidade de que a aplicação tenha a capacidade de carregar um conjunto de configurações a partir de um ficheiro (por exemplo, os dados de acesso ao armazém de dados). O local-chave para proceder à aquisição de configurações é precisamente no *proxy*, contudo, o facto de ser código gerado automaticamente impediria uma fácil manutenção do mesmo pois a cada nova geração perder-se-ia o código acrescentado. A solução encontrada passou por criar o *custom proxy*, sendo este o responsável por obter as configurações de acesso e posteriormente solicitar a operação ao *proxy* gerado. Nesta solução, o *custom proxy* serve de ponte entre a aplicação consumidora e o *proxy* gerado e torna-se possível implementar a aplicação desenvolvida em várias instituições sem efectuar uma única alteração no código-fonte.

Através dos *proxies*, é chamado o componente que possui a implementação do serviço. Este componente tem como tarefas requisitar os dados necessários à satisfação do pedido e realizar a sua tradução (com o auxílio de tradutores), ou seja, receber dados de tipos

definidos nas entidades de negócio (*business entities*) e convertê-los em tipos definidos no contrato de dados (*data contract*).

Neste processo existem outros componentes fundamentais, os tradutores. De forma a manter a clareza na estrutura de desenvolvimento, cada tradutor é responsável pela tradução de um, e apenas um, tipo de dados (por exemplo, a tradução do tipo de dados "Paciente" definido como entidade de negócio e o tipo de dados "Paciente" definido no contrato de dados). A "implementação do serviço" (*service implementation*) assume um papel de coordenador da tradução, delegando em cada tradutor a responsabilidade de traduzir o respectivo tipo de dados.

Um exemplo claro deste papel de coordenador verifica-se quando há a necessidade de traduzir diferentes tipos de dados e, posteriormente, agregar as traduções num único tipo de dados definido no contrato de dados. Eventualmente, a implementação do serviço poderá realizar algumas operações adicionais tais como a construção de um dicionário relativo às informações de um documento, agregação de dados para completar determinada propriedade, etc.

De forma a obter as listas de dados a partir das quais a lógica de negócio trabalhará os dados, esta requisita-os às regras das entidades de negócio (*business entities rules*). É este o componente responsável por comunicar com a base de dados e obter os registos necessários, criando instâncias para os objectos e respectivas listas. O Gerador CPCHS cria duas classes, a primeira contém todos os métodos básicos para acesso à base de dados e a segunda, que estende a primeira, tem como objectivo permitir alterações à forma como se procede ao acesso a dados. Um exemplo que se verificou recorrentemente ao longo do desenvolvimento foi a necessidade de indicar um novo *package* que disponibiliza o procedimento a executar, que não o gerado pelo Gerador CPCHS.

Ao nível da base de dados, a consulta processa-se através da execução de procedimentos (*procedures*). Os procedimentos são igualmente gerados pelo Gerador CPCHS. Aquando da sua geração, a abrangência da pesquisa restringe-se à tabela a que a entidade diz respeito. Muitas vezes, existe a necessidade de introduzir na pesquisa referências a outras tabelas, sendo necessária a criação de um pacote semelhante com os procedimentos personalizados (nesta operação é necessário alterar o pacote de referência nas regras da entidade de negócio).

Cada entidade de negócio (*business entity*) é criada através do Gerador CPCHS, constituída por dois ficheiros, um que contém as definições base, tais como as propriedades da entidade e construtores, e um outro que estende o primeiro onde é possível personalizar a entidade, nomeadamente a inclusão de campos adicionais da entidade que não correspondem a atributos da respectiva tabela ou a chamada a construtores de entidades que representam uma parte constituinte da primeira entidade.

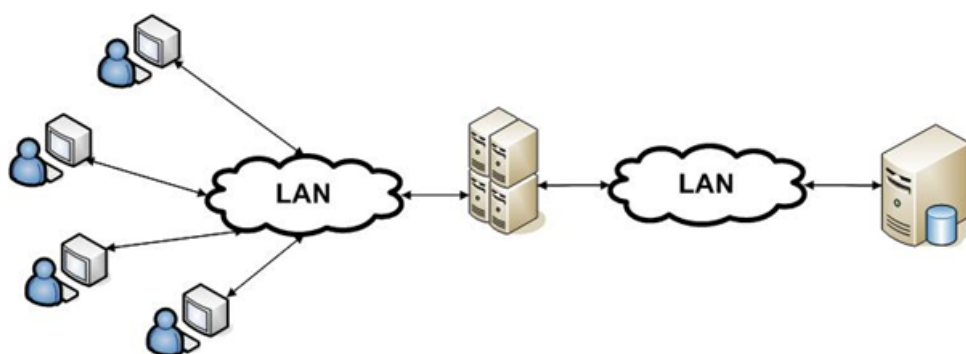


Figura 5.10: Arquitectura física típica da solução.

5.8 Arquitectura Física

Considerando o já referido na secção sobre a arquitectura SOA, existe a possibilidade de que a solução desempenhe as suas funções normalmente dentro de determinados limites no que a configuração física diz respeito. Os vários serviços podem estar localizados em máquinas diferentes e até um mesmo serviço pode estar disponibilizado em mais do que uma máquina. Contudo, a solução foi desenhada, implementada e testada segundo a configuração em que será executada no cliente. Tal como é possível verificar na figura 5.10 da página 71, a solução completa estará distribuída por duas máquinas (o servidor aplicacional e o servidor de dados). O servidor aplicacional concentrará em si tanto os *webservices* como a aplicação cliente. Os vários utilizadores acederão à aplicação recorrendo a terminais ligados ao servidor aplicacional através de uma rede local.

5.9 Módulos dos *Webservices*

Todos os *webservices* necessários ao eResults foram agrupados em três módulos consoante o seu enquadramento no modelo de domínio, entidades, documentos e actividades. Cada módulo é definido com base nos contratos de serviço e de dados. Seguidamente, são descritos detalhadamente cada um dos módulos, incluindo as operações possíveis e tipos de dados associados.

5.9.1 Módulo *Entities*

O módulo denominado "Entities", agrega todos os serviços relacionados com entidades do modelo de domínio, como por exemplo, um doente, uma instituição, um local, uma aplicação, etc.

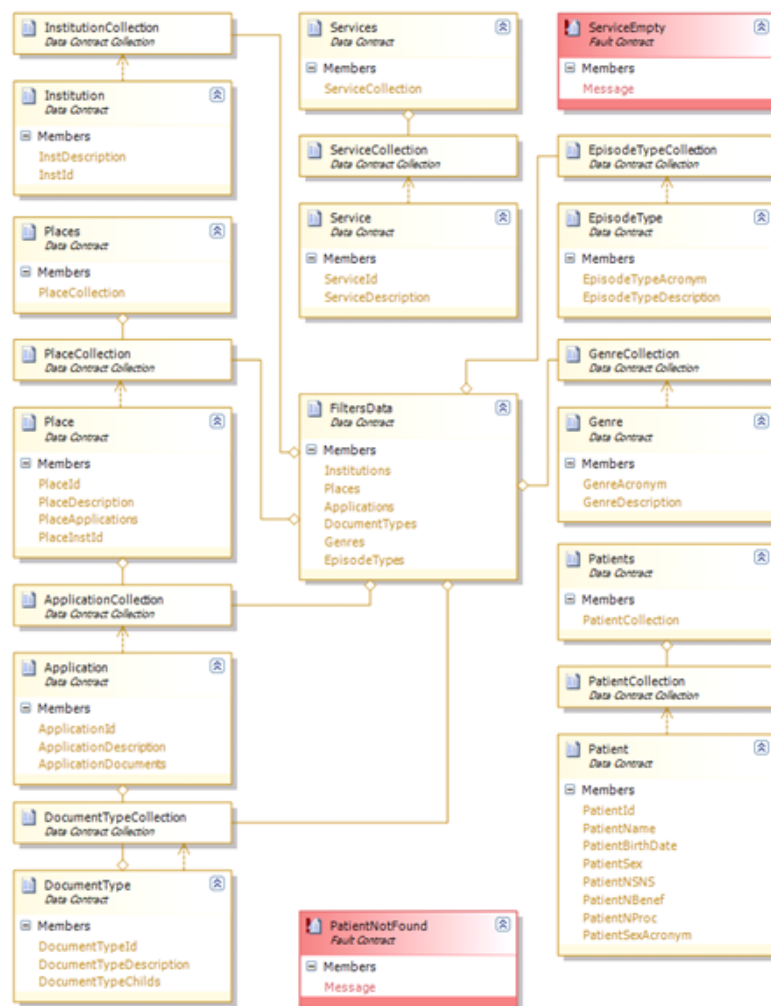


Figura 5.11: Diagrama do contrato de dados do módulo *Entities*.

5.9.1.1 Contrato de Dados do Módulo *Entities*

O módulo "Entities" possui um conjunto de dados definidos para várias entidades do domínio do problema, tal como é possível verificar na figura 5.11 da página 72. As entidades são instituição, local, aplicação, tipo de documento, serviço, tipo de episódio, sexo, paciente e ainda um tipo composto que agrega todos os tipos que constam como critério na barra de pesquisa. São ainda definidos dois tipos de faltas a serem usados caso ocorra uma exceção na respectiva operação, quando um paciente ou um serviço não foram encontrados.

5.9.1.2 Contrato de Serviços do Módulo *Entities*

A figura 5.12 da página 73 apresenta o diagrama do contrato de serviços do módulo "Entities". As caixas identificadas a lilás são as operações possíveis e as caixas a amarelo

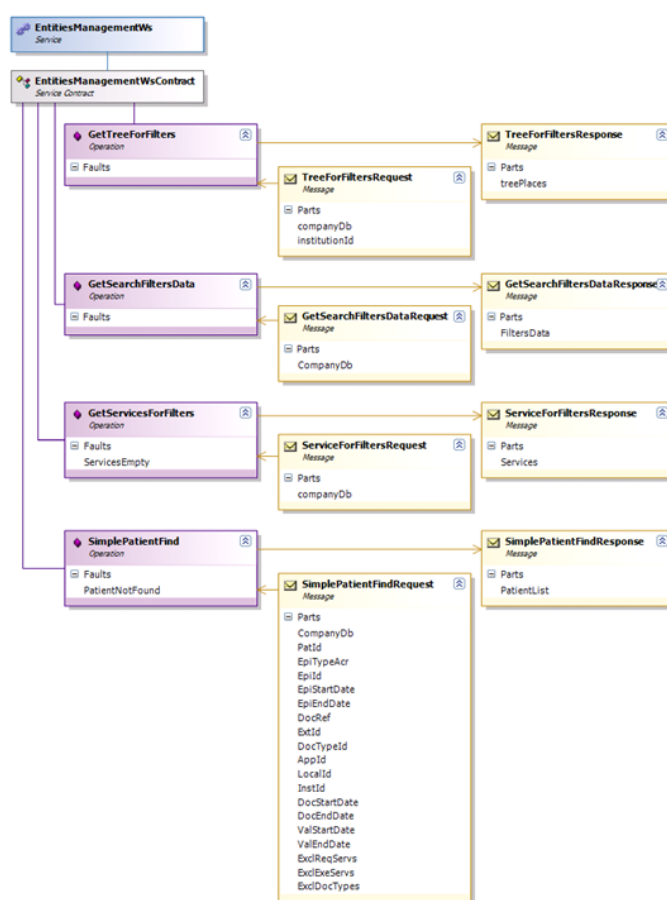


Figura 5.12: Diagrama do contrato de serviços do módulo *Entities*.

representam as mensagens correspondentes a pedidos e respostas.

GetTreeForFilters: Esta operação tem como objectivo devolver uma hierarquia de locais, aplicações e tipos de documento para uma dada instituição (cujo identificador é indicado no pedido). Estes dados destinam-se à construção da árvore de tipos de documento dos filtros globais. O tipo devolvido é "Places" e contém toda a hierarquia subsequente até "DocumentType". Após a obtenção dos dados, a lógica de negócio é responsável por criar os vários níveis hierárquicos colocando em cada um deles os tipos de dados correspondentes.

GetServicesForFilters: Esta operação não faz mais do que devolver uma lista com todos os serviços codificados no armazém de dados. É a partir desta lista que são preenchidas as caixas de selecção de serviços dos filtros globais.

GetSearchFiltersData: Para o preenchimento de todas as opções dos vários critérios de pesquisa (na barra de pesquisas) é necessário obter todas as entidades codificadas no armazém de dados. Esta operação devolve um objecto do tipo "FiltersData" que não é mais do que um agregador dos vários tipos que representam as entidades.

GetSimplePatientFind: Analisando a quantidade de propriedades do pedido poder-se-ia deduzir que esta é talvez a operação que envolve um processamento mais exigente ao nível da lógica de negócio. No entanto, é uma das mais simples pois os dados devolvidos resumem-se a uma lista de pacientes. É ao nível do respectivo procedimento da base de dados que se realiza o maior processamento, pois o código SQL da consulta é construído dinamicamente.

5.9.2 Módulo *Documents*

No segundo módulo, denominado "Documents", estão definidos todos os serviços de gestão de documentos e elementos.

5.9.2.1 Contrato de Dados do Módulo *Documents*

O módulo "Documents" possui apenas os tipos de dados relativos a uma entidade, o documento. É também definido um tipo de dados de falta que pode ser usado no caso de um determinado documento não ter sido encontrado. A figura 5.13 da página 75 ilustra a realidade descrita.

Neste diagrama interessa salientar a propriedade "DocInfo" do tipo de dados "PatientDocument" que é um dicionário utilizado para devolver informação adicional de um documento, como por exemplo, o episódio a que se refere. Por este motivo, é devolvida mais informação do que a que o diagrama apresenta. Esta solução confere mais flexibilidade no momento de efectuar eventuais alterações aos detalhes a apresentar.

5.9.2.2 Contrato de Serviços do Módulo *Documents*

A figura 5.14 da página 76 apresenta o diagrama do contrato de serviços do módulo "Documents".

GetPatientDocuments: De uma forma análoga à operação "GetSimplePatientFind", esta operação é responsável pela devolução da lista de todos os documentos associados a um determinado paciente que obedecem aos critérios de pesquisa que constam do pedido. O respectivo procedimento da base de dados também constrói dinamicamente o código SQL da consulta.

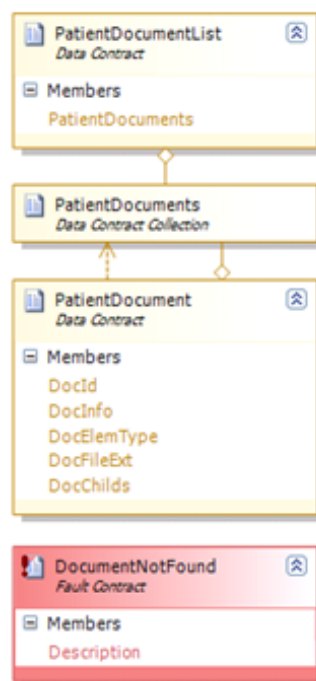


Figura 5.13: Diagrama do contrato de dados do módulo *Documents*.

5.9.3 Módulo *Activities*

Por fim, o módulo "Activities" concentra em si todos os serviços relativos a actividades de um paciente na unidade hospitalar, nomeadamente, uma visita (episódio) ou a realização de um exame. Este módulo é o mais importante do ponto de vista da EDA de MCDT.

5.9.3.1 Contrato de Dados do Módulo *Activities*

O módulo "Activities" possui a definição de todos os tipos de dados relativos a uma ou mais requisições, o que inclui a definição de um exame, um resultado numérico, um resultado microrganismo, um resultado do tipo gráfico e um resultado do tipo antibiograma. Existe também a definição de um tipo de falta com o objectivo de ser utilizado quando não são encontrados exames. A figura 5.15 da página 77 apresenta o diagrama de dados do módulo "Activities".

O modelo definido é muito semelhante ao modelo de dados com a excepção do facto de a hierarquia de exames e respectivos resultados já estar inerente à própria estrutura dos respectivos objectos (situação trabalhada na lógica de negócio).

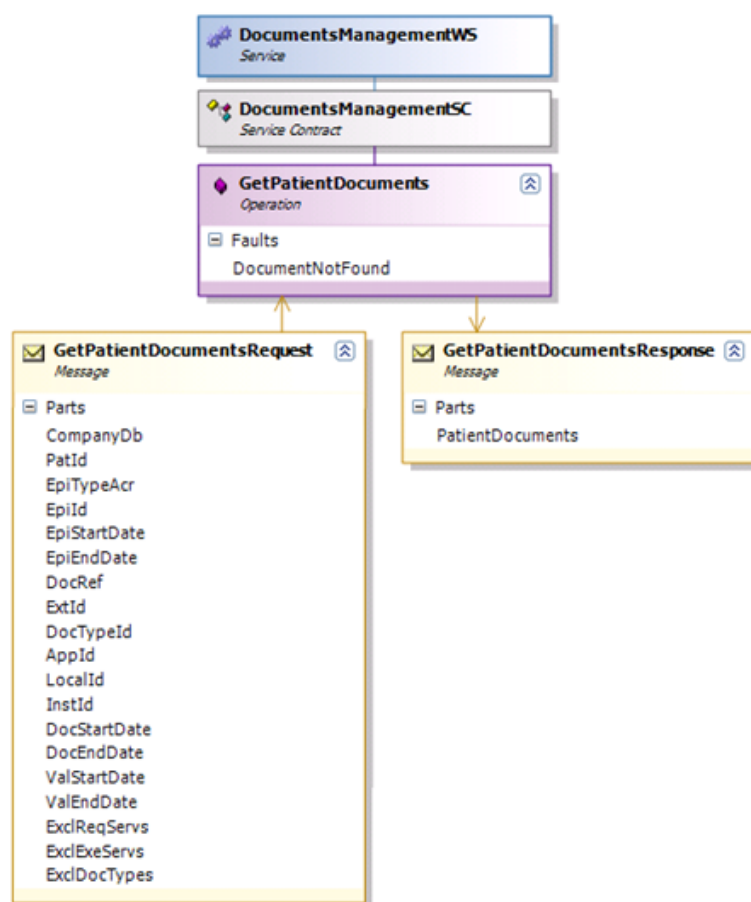


Figura 5.14: Diagrama do contrato de serviços do módulo *Documents*.

5.9.3.2 Contrato de Serviços do Módulo *Activities*

A figura 5.16 da página 78 apresenta o diagrama do contrato de serviços do módulo “Activities”.

GetExamsByDocumentId: O objectivo desta operação é devolver toda a hierarquia de exames e respectivos resultados para uma determinada requisição, dado o identificador do documento associado. O papel da lógica de negócio assume particular importância nesta operação devido ao nível exigido no que diz respeito à manipulação dos dados recebidos. Primeiramente, a lógica de negócio tem de obter cinco tipos de dados diferentes, os exames requisitados, a respectiva hierarquia (em árvore), os resultados alfanuméricos, microrganismos e antibiogramas. O passo seguinte consiste em criar a hierarquia de exames para cada um dos exames pedidos e posteriormente associar os respectivos resultados a cada exame. Para efeitos de visualização, é necessário associar a cada exame pedido os respectivos grupos e subgrupos, caso existam. Para tal, é construído um conjunto de

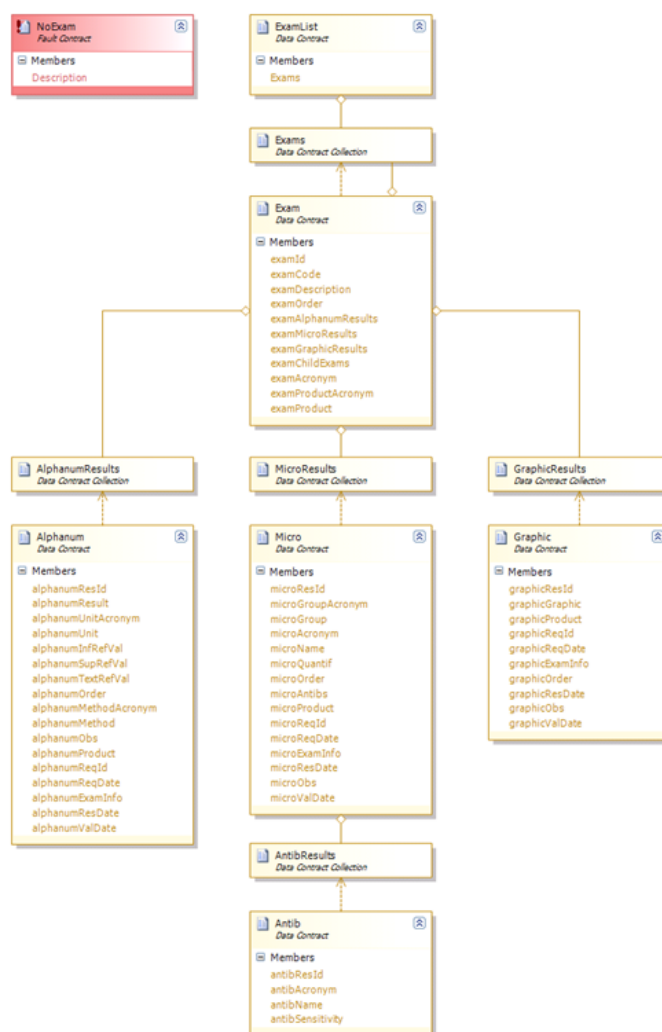


Figura 5.15: Diagrama do contrato de dados do módulo *Activities*.

objectos a partir de código XML que representa os níveis hierárquicos superiores e adicionados a cada um dos exames pedidos (estes deixam de serem as raízes das árvores). De forma a devolver apenas uma hierarquia, todas as hierarquias iniciais são comparadas nível a nível de forma a garantir que em cada um não existem exames repetidos. Neste processo de junção, caso se verifique a repetição de um exame é necessário garantir que todos os resultados são incluídos no nó. Após a comparação de um determinado nível é realizada a ordenação dos exames segundo um parâmetro existente para o efeito. Com esta ordenação é garantida a visualização dos resultados segundo a ordem pretendida pelo cliente. Também a ordenação é realizada nível a nível.

GetExamsByDocumentRef: Esta operação realiza exactamente os mesmos passos que a descrita no ponto anterior, sendo a única diferença que o parâmetro de pesquisa

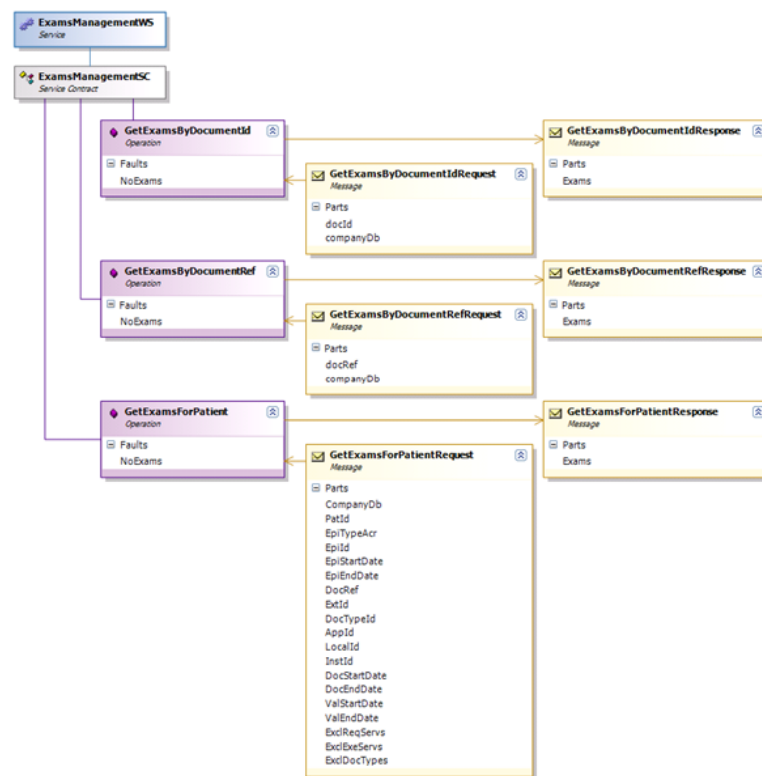


Figura 5.16: Diagrama do contrato de serviços do módulo *Activities*.

é a referência do documento associado e não o seu identificador.

GetExamsForPatient: Também esta operação realiza os passos descritos no ponto anterior, diferenciando-se ao nível dos parâmetros de pesquisa. No caso das duas operações anteriores, os exames devolvidos correspondiam apenas a uma requisição dado que uma requisição é um documento. No caso de a pesquisa ser realizada por paciente (segundo os critérios de pesquisa) existe a possibilidade de os dados devolvidos corresponderem a mais do que uma requisição. Os procedimentos da base de dados que devolvem os cinco tipos de dados são responsáveis por construir dinamicamente o código SQL da consulta.

5.10 Outros Desenvolvimentos

O foco do desenvolvimento ao longo do Projecto foi o módulo "Activities", na camada de *webservices*, e os módulos EDA e Filtros Globais, na aplicação cliente. Tendo em conta que os módulos se inserem numa solução mais global, era natural existirem desenvolvimentos que não beneficiassem exclusivamente os módulos em causa. O inverso

também poderia acontecer, ou seja, desenvolvimentos externos proporcionarem funcionalidades extra aos módulos em análise. Na primeira situação encontram-se os desenvolvimentos da barra de pesquisa e do *provider* de configurações. Na segunda situação enquadra-se o componente de pesquisa de doentes. De forma a clarificar quais os benefícios proporcionados e recebidos, segue-se uma descrição sucinta acerca de cada uma das funcionalidades.

5.10.1 Barras de Pesquisa

Para que sejam visualizados os resultados pretendidos era imprescindível dotar a solução de capacidades de pesquisa. Em qualquer funcionalidade do eResults relacionada com documentos e resultados existe um pressuposto fundamental no que respeita a pesquisa, esta deve ser orientada a um único doente. Este pressuposto advém do propósito do eResults, a possibilidade de tomar decisões fundamentadas aquando da elaboração de um diagnóstico para um paciente específico. Nesta perspectiva, não existe qualquer vantagem em proporcionar meios de comparar resultados de diferentes pacientes.

Os vários critérios de pesquisa podem ser enquadrados em quatro diferentes grupos segundo a entidade a que se referem. Assim, podemos considerar critérios relativos a pacientes, visitas (episódios), documentos e resultados.

Do ponto de vista de experiência de utilização, existe todo o interesse em proporcionar ao utilizador meios de alcançar o seu objectivo com o mínimo de passos e onde a informação pretendida seja o elemento principal da interface. Atendendo a estas características, os critérios de pesquisa foram divididos em dois grupos, primário e secundário, segundo a frequência com que são utilizados. Os elementos considerados no grupo primário constituem a barra de pesquisa básica, enquanto que os restantes constituem a barra de pesquisa avançada.

A barra de pesquisa básica é composta única e exclusivamente por critérios relativos ao paciente:

- Nome
- Número de doente
- Número de processo
- Número do Serviço Nacional de Saúde

A barra de pesquisa avançada é composta pelos critérios:

- Data de nascimento
- Sexo
- Tipo de episódio

- Número do episódio
- Identificador externo
- Tipo de documento
- Número do documento
- Aplicação de origem
- Tipo de data (que permite seleccionar entre data do documento, data do episódio e data de validação dos resultados)
- Data de início
- Data de fim

A barra de pesquisa básica está sempre visível enquanto que a barra de pesquisa avançada está incluída num painel deslizante. Com esta interface o espaço ocupado pela barra de pesquisa avançada é consideravelmente diminuído pois todos os campos estão escondidos. Caso o utilizador pretenda aceder aos mesmos poderá fazê-lo com apenas um clique. Esta conjugação está de acordo com as características de experiência de utilização anteriormente referidas.

De forma a facilitar o processo de implementação das barras de pesquisa em outras páginas, o *user control* de cada uma das barras foi incluído num outro *user control* denominado "GlobalSearchBar". Este componente permite que a página que o inclui defina quais as barras visíveis através de dois parâmetros de configuração. É também responsável pelo correcto preenchimento das opções dos vários critérios assim como disponibilizar meios de acesso a cada critério e lançar os eventos dos botões da barra de pesquisa básica.

Este componente foi reutilizado no módulo de pesquisa documental.

5.10.2 Provider de Configurações

Para que seja possível guardar de forma persistente as preferências de um utilizador relativamente a filtros de tipos de documento e serviços foi necessário implementar um meio de gerir as mesmas. A solução encontrada passou pelo desenvolvimento de um *provider* que gere o armazenamento de configurações por utilizador. Para cada par utilizador/configuração é guardado código XML que contém o valor da configuração. As operações sobre configurações são apenas duas e permitem o total controlo das mesmas. Uma primeira operação permite obter todas as configurações definidas para um determinado utilizador. A segunda operação permite guardar (caso não exista nenhum

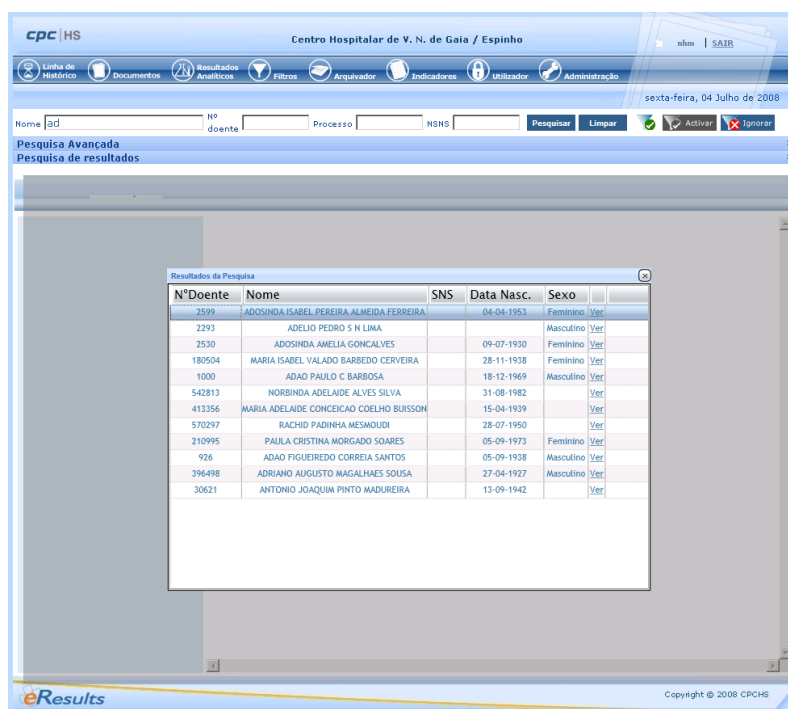


Figura 5.17: Integração do componente de selecção do doente no módulo de EDA.

par utilizador/configuração) ou actualizar as preferências do utilizador. Este desenvolvimento foi integrado no módulo de autenticação da CPCHS e inclusive já utilizado por outras soluções.

5.10.3 Componente de Selecção do Doente

Após o preenchimento dos critérios de pesquisa pretendidos e o clique no botão que inicia a pesquisa, a solução deve permitir ao utilizador seleccionar o paciente desejado, caso existam dois ou mais pacientes que cumpram os requisitos. No caso de ser apenas um, a solução deve imediatamente apresentar os dados pretendidos, dependendo do contexto. Na primeira situação descrita, a solução apresenta uma lista de pacientes com alguns dados relativos ao mesmo de forma a serem perfeitamente identificados. Este componente foi desenvolvido (em Silverlight) para o módulo de pesquisa documental e posteriormente integrado no módulo de EDA. Esta opção deveu-se ao facto de manter a coerência da interface, na medida em que não fazia sentido apresentar duas formas diferentes de realizar a mesma tarefa. Desta forma, pretende-se promover o reconhecimento em detrimento da memorização, no que à interface diz respeito. Na figura 5.17 da página 81 é possível observar a integração do componente de selecção do doente no módulo de EDA.

5.11 Interface Gráfica

O objectivo desta secção não é descrever exhaustivamente as várias interfaces desenvolvidas ao longo do Projecto, apenas resumir os seus componentes e funcionalidades e tornar um pouco mais concreta toda a descrição que foi sendo realizada. À data da entrega do presente documento, decorre ainda o processo de auditoria ao projecto eResults (realizado por uma empresa externa à CPCHS e ao cliente), motivo pelo qual as interfaces apresentadas não podem ser consideradas finais. Contudo, os componentes base não sofrerão grandes alterações ao nível das suas funcionalidades, sendo viável a realização de alguns ajustes no que diz respeito a posicionamento ou aspecto. Em cada interface é indicada a figura correspondente.

5.11.1 Tabela Global

A interface de entrada no módulo de EDA foi designada de Tabela Global pelo facto de apresentar uma visão geral de todos os resultados existentes, numa perspectiva mais convencional. A figura 5.18 da página 83 permite acompanhar a descrição da interface. A identificação da requisição [1], bem como os parâmetros [2], unidades [3] e valores de referência [4] são colunas bloqueadas, pelo que o deslocamento, quer vertical, quer horizontal, se reflecte apenas nos resultados permitindo, a qualquer momento, enquadrar correctamente qualquer valor visualizado. É possível mostrar e esconder os resultados de determinados parâmetros segundo a hierarquia dos mesmos [5]. A interface fornece informação acerca das células com as quais é possível interagir, ou seja, aquelas que correspondem a resultados com detalhes (no exemplo trata-se de um microrganismo) [6]. Quando o cursor é colocado sobre a identificação de uma requisição, é apresentada mais informação sobre a mesma [7]. De uma forma análoga, poderá ser visualizada informação mais detalhada sobre os resultados.

5.11.2 Evolução

No caso dos resultados numéricos, é possível verificar a evolução dos parâmetros [1] escolhidos através da respectiva hierarquia [2]. Colocando o cursor do rato sobre um ponto específico (que corresponde à realização de um exame) é apresentada informação complementar[3]. A figura 5.19 da página 84 permite identificar os pontos descritos.

5.11.3 Antibiogramas

Os resultados do tipo microrganismo possuem antibiogramas associados. Para que a tabela não se torne demasiado preenchida com a apresentação de mais esta especificidade, foi criada uma interface própria que pode ser acedida pelo respectivo resultado na tabela global ou através da selecção do respectivo separador. A identificação da requisição [1],

Centro Hospitalar de V. N. de Gaia / Espinho

sexta-feira, 04 Julho de 2008

Nome: ADAO PAULO C BARBOSA Nº doente: 1000 Processo: NSNS

Pesquisa Avançada
Pesquisa de resultados

Tabela Evolução Micro

Parâmetro	Unid.	Val. ref.	03-07-2008 SISLAB1 SANGUE	02-07-2008 521577 SANGUE
Titulação Antic.				
Teste Confirmatório Ac HIV 1/2		-		
Antig.HBs (Confirmatório)		-		
☒ Hemograma				
Hemoglobina	g/dl	-	1.2	
Basófilos	%	-	2.7	
Eosinófilos	%	-	3.5	
Eritrócitos	milhões/mm	-	9.7	1.2
Neutrófilos	%	-	3.6	6.5
Volume globular	l/l	-	5.7	5.6
Linfócitos	%	-	7.5	1.9
Leucócitos	milhares/mm	-	5.7	6.7
Monócitos	%	-	4.2	1.0
☒ TAXA DE PROTROMBINA				
Taxa de Protrombina	%	>70-		65.3
INR		-		1.23
Agregação plaquetária com agonista		-		D5
Agregação plaquetária induzida pela ristocetina		-		92
Microorganismo		-		Excol

Pop-up Information:
Tipo de episódio: Internamentos
Episódio: 123456
Início do episódio:
Fim do episódio:
Serviço requerente:
Especialidade requerente:
Serviço executante: Cardiologia
Especialidade executante:
Id. externo: SISLAB1
Última validação: 03-07-2008

Figura 5.18: Tabela global de apresentação de resultados.

bem como os antibióticos [2] são colunas bloqueadas, pelo que o deslocamento, quer vertical, quer horizontal da tabela, se reflecte apenas nos resultados permitindo, a qualquer momento, enquadrar correctamente qualquer valor visualizado. Quando o cursor é colocado sobre a identificação de uma requisição, é apresentada mais informação sobre a mesma. De uma forma análoga, poderá ser visualizada informação mais detalhada sobre os resultados[3]. No caso de a interface ser acedida através da selecção de um resultado específico na tabela global, os resultados são filtrados pelo(s) respectivo(s) microrganismo(s). Se o utilizador pretender voltar a visualizar todos os antibiogramas, poderá desactivar a filtragem através de um clique no botão respectivo [4]. A figura 5.20 da página 85 permite identificar os pontos descritos.

5.11.4 Tipos de Documento

A interface de selecção de tipos de documento a visualizar é bastante simples. É constituída pela árvore hierárquica de locais, aplicações e tipos de documento [1], pelos botões de gestão de configurações (gravar de forma persistente [2], repor valores iniciais

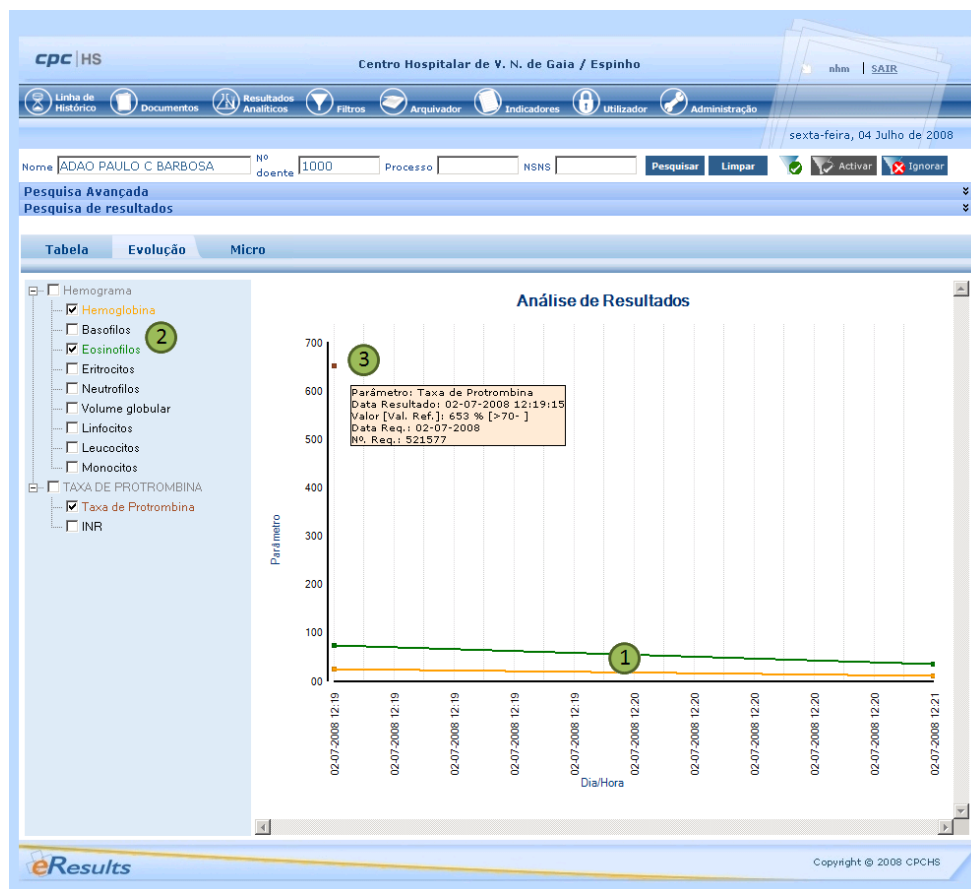


Figura 5.19: Evolução de resultados numéricos.

[3] e gravar para a sessão [4]) e pelos botões de selecção de nós (seleccionar todos [5] e cancelar todas as selecções [6]). A figura 5.21 da página 86 permite identificar os pontos descritos.

5.11.5 Serviços

A interface de selecção de serviços disponibiliza quatro grupos de serviços que permitem definir claramente as opções pretendidas (serviços requisitantes disponíveis [1], serviços requisitantes excluídos [2], serviços executantes disponíveis [3] e serviços executantes excluídos [4]). Dado o volume de serviços existentes, foram disponibilizadas caixas de pesquisa com suporte a *wildcards* [5]. Para definir qual o grupo a que um serviço pertence o utilizador pode recorrer aos botões de movimentação (selecção para a direita [6], selecção para a esquerda [7], todos para a direita [8] e todos para a esquerda [9]). Nesta interface existem também os botões de gestão de configurações (gravar de forma persistente [10], repor valores iniciais [11] e gravar para a sessão [12]). A figura 5.22 da página 87 permite identificar os pontos descritos.

Antibiótico	Resultado	Detalhes
ANTIB2	Resistente	Tipo de episódio: Internamentos Episódio: 123456 Início do episódio: Fim do episódio: Serviço requisitante: Especialidade requisitante: Serviço executante: Cardiologia Especialidade executante: Id. externo: SISLAB1 Última validação: 03-07-2008
ANTIB1	Sensível	
ANTIB2	Indeterminado	

Figura 5.20: Tabela de apresentação de antibiogramas.

5.11.6 Barras de Pesquisa

O componente de barras de pesquisa está dividido em duas zonas, pesquisa básica [1] e pesquisa avançada [2]. É constituído, também, pelos botões de pesquisa[3], limpar campos [4], activar filtros globais [6] ignorar filtros globais [7] e pelo indicador do estado dos filtros globais [5]. A figura 5.23 da página 87 permite identificar os pontos descritos.

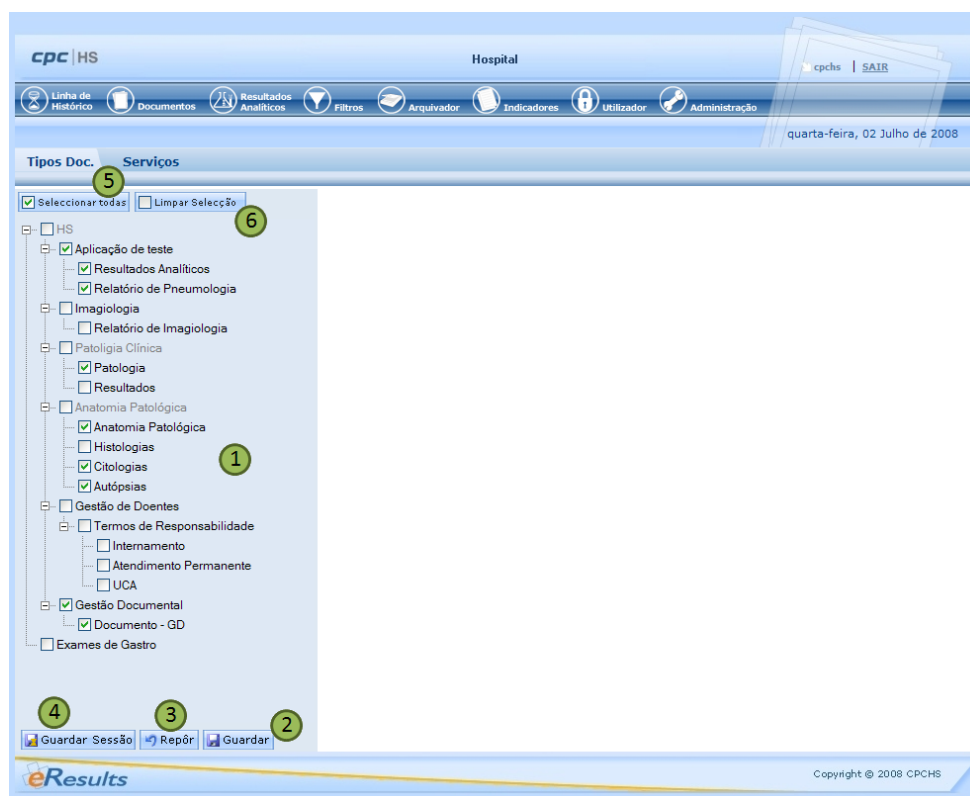


Figura 5.21: Selecção de tipos de documentos a visualizar.

Descrição do Processo de Implementação do eResults – EDA de MCDT

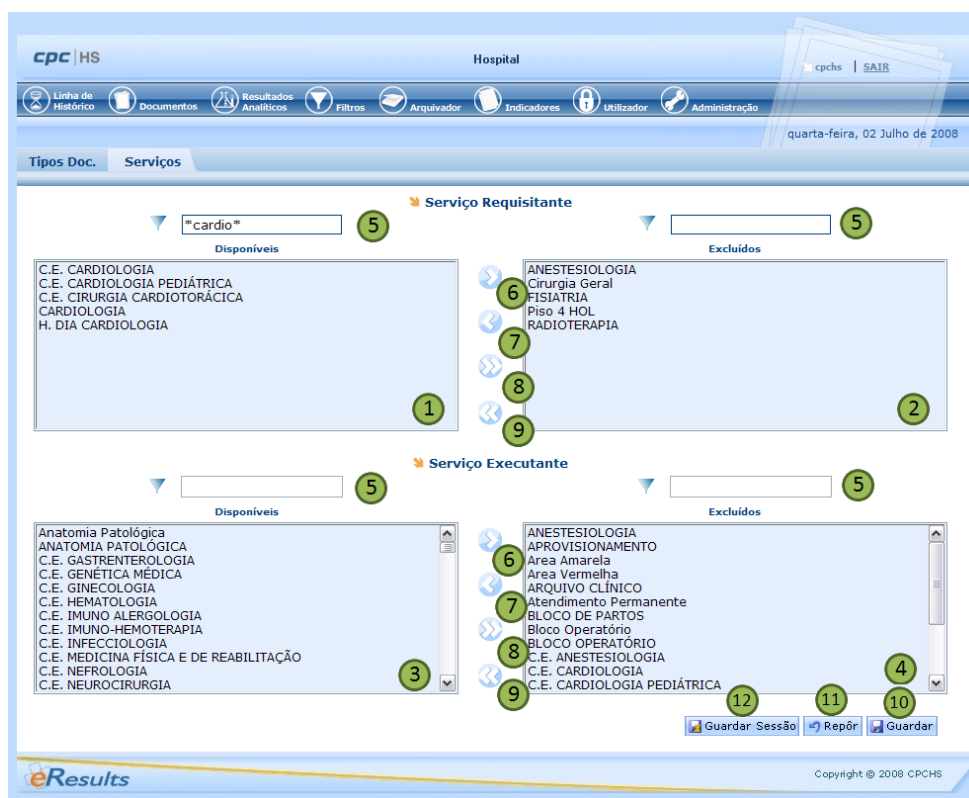


Figura 5.22: Selecção de serviços a visualizar.

Figura 5.23: Barras de pesquisa.

Capítulo 6

Conclusões e Perspectivas de Trabalho Futuro

Neste capítulo constam as principais conclusões e resultados obtidos no decorrer do Projecto. É também realizada a identificação de algumas questões que merecem mais atenção no futuro.

6.1 Satisfação dos Objectivos

Os objectivos definidos no início do Projecto foram atingidos com sucesso. Os módulos de EDA e Filtros Globais foram desenvolvidos nos prazos estabelecidos, apresentando as funcionalidades pretendidas. Foram ainda realizados outros desenvolvimentos, como o *provider* de configurações e a barra de pesquisa, igualmente com sucesso.

O modelo de dados de suporte a toda a solução encontra-se estável, indo de encontro a todas as necessidades da solução. Desde o início do Projecto, o modelo sofreu algumas reformulações de média dimensão com especial ênfase no suporte aos resultados analíticos. O resultado final obtido permite concluir que o modelo se encontra suficientemente genérico de forma a suportar várias aplicações externas fornecedoras de dados. De igual forma, a integração de um módulo de pedidos de exames está prevista, não implicando qualquer alteração às tabelas já definidas.

Foram desenvolvidos todos os *webservices* que alimentam a interface gráfica, incluindo o módulo de pesquisa documental. Durante o seu desenvolvimento foram realizadas algumas avaliações de desempenho o que conduziu a uma solução que, em condições de não congestionamento da rede, possui um tempo de resposta na ordem de 2 ou 3 segundos.

Um dos objectivos mais importantes definidos no início do Projecto era o desenvolvimento de uma interface que permitisse a geração de gráficos de evolução de parâmetros e sua tendência. A interface foi implementada com sucesso e permite que o utilizador possa determinar quais os parâmetros a comparar, conferindo a capacidade de verificar eventuais correlações entre dois ou mais parâmetros.

No domínio dos filtros globais, o utilizador tem a capacidade de definir e gerir as suas preferências através de uma interface que se considera simples e eficiente. O desenvolvimento do *provider* de suporte foi realizado tendo em mente a sua eventual utilização por outras soluções da empresa, situação que se veio a verificar.

6.2 Linhas de Orientação para Trabalho Futuro

Devido a restrições de tempo e prioridades de implementação, algumas questões não foram exploradas de forma a converter todo o seu potencial em valor acrescentado. Segue-se a identificação de alguns pontos que, devidamente explorados, podem conferir à solução ainda mais funcionalidades e incrementar o seu desempenho.

6.2.1 Desempenho das Interrogações

Uma das questões tidas em conta desde o início do Projecto foi o desempenho global da solução. Para além da exigência de absoluta precisão dos dados, o tempo de resposta é uma característica bastante importante no domínio da Saúde. Tal como referido anteriormente, foi avaliado o desempenho da solução, contudo, uma análise mais profunda das interrogações ao armazém de dados e eventual optimização pode diminuir ainda mais o tempo de resposta aos vários pedidos.

6.2.2 Apresentação de Resultados do Tipo Gráfico

Um dos tipos de resultados suportados que exige um tratamento especial devido à sua especificidade é precisamente o tipo gráfico. A sua existência foi considerada ao longo de todo o desenvolvimento, incluindo a definição da camada de serviços, no entanto, devido a prioridades de desenvolvimento, não foi implementada a interface de visualização. Para a resolução desta questão foram identificadas duas soluções, o desenvolvimento de raiz de uma interface ou a inclusão de um componente Silverlight (desenvolvido no âmbito do módulo de pesquisa documental) que possui já capacidades como rotação e zoom de imagens. Em primeira análise, a segunda solução afigura-se como a mais aconselhável atendendo a vários factores. O primeiro é o bom desempenho do componente e a liberdade que permite ao utilizador. Uma segunda vantagem prende-se com a própria interface gráfica que, desta forma, manteria a coerência de aspecto e funcionalidade, não obrigando o utilizador a memorizar ou a reconhecer dois modos diferentes de realizar a

mesma operação, a manipulação de imagens. O terceiro motivo prende-se com a própria integração de componentes Silverlight no módulo EDA, já testada com a inclusão do componente de selecção de pacientes, e que conduz a bons resultados.

6.2.3 Evolução de Antibiógramas

Apesar de não ser um requisito definido no início do Projecto, ao longo do mesmo, a evolução de antibiógramas foi identificada como sendo uma funcionalidade com um forte potencial. Da mesma forma que é possível verificar a evolução e tendência de resultados analíticos, também no caso dos antibiógramas poderão obter-se conclusões clínicas adicionais a partir da observação da evolução da resistência de um microrganismo a determinado antibiótico.

6.2.4 Barra de Pesquisa Configurável

Tal como referido na Introdução do documento, o Projecto foi realizado atendendo à realidade vivida no Centro Hospitalar de Vila Nova de Gaia/Espinho, contudo, ao longo de todo o desenvolvimento foi tida em conta adaptabilidade da solução a outras realidades. Neste domínio, as barras de pesquisa podem sofrer um melhoramento ao nível da configuração dos critérios de pesquisa disponíveis. A situação ideal seria a possibilidade de definir as preferências num ficheiro de configuração.

6.3 Conclusões

A fase inicial do Projecto consistiu em pesquisa e testes a um conjunto de tecnologias e ferramentas com potencial envolvimento no desenvolvimento da solução. O objectivo era reunir um conjunto tecnológico capaz de suportar eficientemente o desenvolvimento e manutenção da solução. Analisando agora todo o processo de desenvolvimento e resultados finais obtidos, é possível concluir que o conjunto de opções tomadas (atendendo às condicionantes descritas) verificou-se o mais correcto.

Os vários testes aos quais a solução foi sujeita e os resultados das apresentações ao cliente, fazem crer que a solução cumpre os requisitos definidos e apresenta um bom nível de estabilidade. À data da elaboração do presente documento, a solução encontra-se já em fase de testes nas instalações do cliente no âmbito de uma auditoria realizada por uma terceira empresa.

Conclusões e Perspectivas de Trabalho Futuro

Referências

- [Amb] Scott W. Ambler. Feature driven development (fdd) and agile modeling. Agile Modeling, disponível em <http://www.agilemodeling.com/essays/fdd.htm>, acessado pela última vez em 5 de Julho de 2008.
- [And07] James G. Anderson. Social, ethical and legal barriers to e-health. *International Journal of Medical Informatics* 76, page 480–483, 2007.
- [ARdSdN] I.P. Administração Regional de Saúde do Norte. Ars norte - centro hospitalar de vila nova de gaia. disponível em http://www.arsnorte.min-saude.pt/h_vn_gaia.htm, acessado pela última vez em 3 de Julho de 2008.
- [Boe88] Barry W. Boehm. A spiral model of software development and enhancement. *IEEE* 21, pages 61–72, 1988.
- [Bro] Peter Bromberg. Repository factory out. Peter Bromberg’s Un-Blog, disponível em <http://petesbloggerama.blogspot.com/2007/10/repository-factory-out.html>, acessado pela última vez em 5 de Julho de 2008.
- [Cau] Grant Cause. Delivering real business value using fdd. Methods And Tools, disponível em <http://www.methodsandtools.com/archive/archive.php?id=19>, acessado pela última vez em 5 de Julho de 2008.
- [CLL99] Peter Coad, E. Lefebvre, and Jeff De Luca. *Java Modeling in Color With UML: Enterprise Components and Process*. Prentice Hall International, 1999.
- [Cora] Microsoft Corporation. Ajax integration and json support. MSDN, disponível em <http://msdn.microsoft.com/en-us/library/bb412173.aspx>, acessado pela última vez em 21 de Junho de 2008.
- [Corb] Microsoft Corporation. Comparing linq and its contemporaries. .NET Framework Developer Center, disponível em <http://msdn.microsoft.com/en-us/library/aa479863.aspx>, acessado pela última vez em 5 de Julho de 2008.
- [Corc] Microsoft Corporation. Linq: .net language-integrated query. .NET Framework Developer Center, disponível em [93](http://msdn.</p></div><div data-bbox=)

REFERÊNCIAS

- [microsoft.com/pt-pt/library/bb308959\(en-us\).aspx](http://microsoft.com/pt-pt/library/bb308959(en-us).aspx),
acedido pela última vez em 5 de Julho de 2008.
- [Cord] Microsoft Corporation. patterns and practices: Repository factory. Codeplex, disponível em <http://codeplex.com/RepositoryFactory>, acedido pela última vez em 5 de Julho de 2008.
- [Core] Microsoft Corporation. patterns and practices: Repository factory. Codeplex, disponível em <http://www.codeplex.com/RepositoryFactory/WorkItem/View.aspx?WorkItemId=5084>, acedido pela última vez em 5 de Julho de 2008.
- [Corf] Microsoft Corporation. Windows communication foundation architecture overview. MSDN, disponível em <http://msdn.microsoft.com/en-us/library/aa480210.aspx>, acedido pela última vez em 21 de Junho de 2008.
- [Cot07] Carole M. Cotter. Making the case for a clinical information system: The chief information officer view. *Journal of Critical Care* 22, page 56–65, 2007.
- [CPdCHSa] S.A. Companhia Portuguesa de Computadores Healthcare Solutions. Historial da cpchs. disponível em http://www.cpchs.pt/eportal/v10/PT/asp/CPCHS/CPCHS_Historial/index.aspx, acedido pela última vez em 3 de Julho de 2008.
- [CPdCHSb] S.A. Companhia Portuguesa de Computadores Healthcare Solutions. Missão da cpchs. disponível em http://www.cpchs.pt/eportal/v10/PT/asp/CPCHS/CPCHS_Missao/index.aspx, acedido pela última vez em 3 de Julho de 2008.
- [CPdCHS07] S.A. Companhia Portuguesa de Computadores Healthcare Solutions. Fornecimento de solução para distribuição de resultados electrónicos de medt's - documento de especificação para parceiro. Technical report, Companhia Portuguesa de Computadores - Healthcare Solutions, S.A., Dezembro 2007.
- [DAS⁺08] George Demiris, Lawrence B. Afrin, Stuart Speedie, Karen L. Courtney, Manu Sondhi, Vivian Vimarlund, Christian Lovis, William Goossen, and Cecil Lynch. Patient-centered applications: Use of information technology to promote disease management and wellness. a white paper by the amia knowledge in motion working group. *Journal of the American Medical Informatics Association* 15, page 8–13, 2008.
- [dCdEdLedPT] Gabinete do Coordenador da Estratégia de Lisboa e do Plano Tecnológico. Plano tecnológico - portugal a inovar... disponível em <http://www.planotecnologico.pt/>, acedido pela última vez em 3 de Julho de 2008.

REFERÊNCIAS

- [dGvL] Gerardo de Geest and Gerben van Loon. Service station - web service software factory modeling edition. MSDN Magazine, disponível em <http://msdn.microsoft.com/en-us/magazine/cc164250.aspx>, acessado pela última vez em 3 de Julho de 2008.
- [Gup] Saurabh Gupta. A performance comparison of windows communication foundation (wcf) with existing distributed communication technologies. MSDN, disponível em <http://msdn.microsoft.com/en-us/library/aa480210.aspx>, acessado pela última vez em 21 de Junho de 2008.
- [Hau06] Reinhold Haux. Health information systems - past, present, future. *International Journal of Medical Informatics* 75, page 268—281, 2006.
- [HS07] Bradford W. Hesse and Ben Shneiderman. ehealth research from the user's perspective. *American Journal of Preventive Medicine*, page S97–S103, 2007.
- [MCS⁺07] Ann Scheck McAlearneya, Deena J. Chisolmb, Sharon Schweikharta, Mitchell A. Medowd, and Kelly Kelleherb. The story behind the story: Physician skepticism about relying on clinical information technologies to reduce medical errors. *International Journal of Medical Informatics* 76, page 836–842, 2007.
- [MLM⁺06] C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter Brown, and Rebekah Metz. Reference model for service oriented architecture. Technical report, OASIS, Fevereiro 2006.
- [Pod03] Saumendra Poddar. Introduction to .net web services, Junho 2003. CodeProject, disponível em <http://www.codeproject.com/KB/IP/intro2websvc.aspx>, acessado pela última vez em 3 de Julho de 2008.
- [SCD] Darleen Sadoski and Santiago Comella-Dorda. Three tier software architectures - software technology roadmap. Carnegie Mellon Software Engineering Institute, disponível em <http://www.sei.cmu.edu/str/descriptions/threetier.html>, acessado pela última vez em 22 de Junho de 2008.
- [Ste] Michael Stevens. The benefits of a service-oriented architecture. developer.com, disponível em <http://www.developer.com/services/article.php/1041191>, acessado pela última vez em 5 de Julho de 2008.
- [TN86] Hirotaka Takeuchi and Ikujiro Nonaka. The new new product development game. *Harvard Business Review*, pages 61–72, 1986.
- [Vas] Clemens Vasters. Introduction to building windows communication foundation services. MSDN, disponível em <http://msdn.microsoft.com/en-us/library/aa480190.aspx>, acessado pela última vez em 22 de Junho de 2008.

REFERÊNCIAS

- [vSR⁺07] John Øvretveit, Tim Scott, Thomas G. Rundall, Stephen M. Shortell, and Mats Brommels. Implementation of electronic medical records in hospitals: two case studies. *Health Policy* 84, page 181–190, 2007.

Anexo A

Actores, Pré-Condições, Pós-Condições e Pressupostos dos Casos de Uso

A execução de cada caso de uso possui algumas pré-condições, pós-condições e pressupostos e tem um ou mais grupos de utilizadores associados. É à identificação de cada um desses parâmetros que este anexo é dedicado.

A.1 Casos de Uso do Módulo Exploração de Dados Analíticos

A.1.1 Solicitar visualização de uma requisição pelo identificador do documento

Actores

Módulo/Aplicação Externa

Pré-condições

Existência do par chave/identificador no URL da página.

Pós-condições

É apresentada toda a interface de resultados preenchida em conformidade com os resultados correspondentes à requisição e preenchidos os critérios de pesquisa com os dados do paciente. Caso não exista, não ocorre qualquer preenchimento de dados e é apresentada ao utilizador uma mensagem informativa da inexistência da referida requisição.

Pressupostos

Existe uma requisição com o identificador definido, com uma estrutura hierárquica e resultados correctamente codificados no armazém de dados.

A.1.2 Solicitar visualização de uma requisição pela referência do documento

Actores

Módulo/Aplicação Externa

Pré-condições

Existência do par chave/referência no URL da página.

Pós-condições

É apresentada toda a interface de resultados preenchida em conformidade com os resultados correspondentes à requisição e preenchidos os critérios de pesquisa com os dados do paciente. Caso não exista, não ocorre qualquer preenchimento de dados e é apresentada ao utilizador uma mensagem informativa da inexistência da referida requisição.

Pressupostos

Existe uma requisição com a referência indicada, com uma estrutura hierárquica e resultados correctamente codificados no armazém de dados.

A.1.3 Pesquisar requisições segundo critérios de filtragem

Actores

Médico; Gestor de Sistema

Pré-condições

A barra de pesquisa está visível e activa na página, incluindo o botão "Pesquisar".

Pós-condições

Numa primeira fase são apresentados os pacientes que obedecem aos requisitos especificados. Após a selecção do paciente pretendido, são apresentadas todas as requisições do mesmo que, mais uma vez, obedecem aos critérios definidos. Os critérios de pesquisa são preenchidos com os dados do paciente em causa.

Pressupostos

Existe pelo menos uma requisição de um doente em que ambas as entidades obedecem aos critérios definidos. As requisições em causa possuem uma estrutura hierárquica e resultados correctamente codificados no armazém de dados.

A.1.4 Limpar critérios de pesquisa e dados apresentados

Actores

Médico; Gestor de Sistema

Pré-condições

A barra de pesquisa está visível e activa na página, incluindo o botão "Limpar". São visíveis os dados de pelo menos uma requisição.

Pós-condições

É apresentada ao utilizador uma mensagem interrogativa acerca da validação da acção. Caso o utilizador responda afirmativamente, todos os campos de pesquisa são apagados bem como os dados presentes na interface de resultados. Em caso de resposta negativa, permanecem todos os dados.

Pressupostos

O estado actual do sistema consiste na visualização dos resultados de uma ou mais requisições. Os critérios de pesquisa estão preenchidos com os dados relativos ao paciente em causa.

A.1.5 Ignorar os filtros globais

Actores

Médico; Gestor de Sistema

Pré-condições

A barra de pesquisa está visível e activa na página, incluindo o botão "Ignorar filtros".

Pós-condições

Caso existam, as requisições com características excluídas nos filtros globais são incluídas nos resultados das pesquisas caso sejam conformes com os restantes critérios. A imagem identificativa do estado de activação dos filtros globais indica que os filtros não estão activos.

Pressupostos

Os filtros globais estão activos. Existe pelo menos uma requisição de um doente em que ambas as entidades obedecem aos critérios definidos na pesquisa embora não sejam conformes com os critérios definidos nos filtros globais.

A.1.6 Activar os filtros globais

Actores

Médico; Gestor de Sistema

Pré-condições

A barra de pesquisa está visível e activa na página, incluindo o botão "Activar filtros".

Pós-condições

As pesquisas efectuadas voltam a ser condicionadas pelas configurações definidas nos filtros globais. A imagem identificativa do estado de activação dos filtros globais indica que os filtros estão activos.

Pressupostos

Existe pelo menos uma requisição de um doente em que ambas as entidades obedecem aos critérios definidos na pesquisa e não são conformes com os critérios definidos nos filtros globais.

A.1.7 Ver ficha do doente

Actores

Médico; Gestor de Sistema

Pré-condições

Existe um doente que obedece a todos os critérios de pesquisa definidos e os dados das suas requisições, que também obedecem aos critérios, estão definidos. A barra de pesquisa está visível e activa na página, incluindo o botão "Activar filtros".

Pós-condições

É apresentada uma nova janela que representa a ficha do paciente. Os dados apresentados devem corresponder ao paciente em causa.

Pressupostos

Existem dados adicionais relativos ao paciente em causa.

A.1.8 Navegar na árvore de parâmetros

Actores

Médico; Gestor de Sistema

Pré-condições

Estão visíveis resultados em que os níveis hierárquicos superiores apresentam botões de esconder e mostrar.

Pós-condições

Os níveis hierárquicos mais baixos do nível clicado são mostrados ou escondidos consoante o seu estado anterior.

Pressupostos

Os resultados apresentados possuem mais do que um nível hierárquico.

A.1.9 Seleccionar tipos de exames especiais

Actores

Médico; Gestor de Sistema

Pré-condições

Estão visíveis resultados de uma ou mais requisições e os que possuem especificidades estão claramente identificados.

Pós-condições

O utilizador é redireccionado para uma nova interface que apresenta os detalhes do resultado em causa.

Pressupostos

As requisições em causa apresentam resultados que possuem especificidades.

A.1.10 Definir o conjunto de parâmetros visualizados no gráfico de evolução

Actores

Médico; Gestor de Sistema

Pré-condições

Existe uma árvore de parâmetros.

Pós-condições

A árvore apresenta um comportamento *tri-state* e é expansível. Cada nó responde correctamente aos sucessivos cliques que são efectuados na respectiva *checkbox* numa sequência de selecção/remover selecção. As alterações aos nós seleccionados na árvore repercutem-se no gráfico gerado.

Pressupostos

De entre todas as requisições visíveis existe pelo menos um parâmetro com um resultado numérico.

A.1.11 Limpar filtros de antibiogramas

Actores

Médico; Gestor de Sistema

Pré-condições

O utilizador encontra-se na interface de antibiogramas e estão visíveis apenas os registos relativos aos microrganismos de uma célula. O botão "Limpar filtros" está disponível.

Pós-condições

Passaram a estar visíveis todos os antibiogramas de todas as requisições.

Pressupostos

Inicialmente não estão visíveis todos os antibiogramas de todas as requisições.

A.1.12 Ver detalhes de um resultado

Actores

Médico; Gestor de Sistema

Pré-condições

Em qualquer uma das interfaces deve existir, pelo menos, uma requisição com, no mínimo, um resultado.

Pós-condições

Após o período de tempo referido, a *tooltip* está visível e com os dados relativos ao(s) resultado(s) em causa.

Pressupostos

Existe informação adicional sobre o(s) resultado(s) em causa.

A.1.13 Ver detalhes de uma requisição

Actores

Médico; Gestor de Sistema

Pré-condições

Em qualquer uma das interfaces deve existir, pelo menos, uma requisição com, mínimo, um resultado.

Pós-condições

Após o período de tempo referido, a *tooltip* está visível e com os dados relativos à(s) requisição(ões) em causa.

Pressupostos

Existe informação adicional sobre a(s) requisição(ões) em causa.

A.2 Casos de Uso do Módulo Filtros Globais

A.2.1 Definir o conjunto de tipos de documentos pretendidos

Actores

Médico

Pré-condições

Existe uma árvore de locais, aplicações e tipos de documento. Se eventualmente existirem configurações prévias, os respectivos nós estão seleccionados. Estão também disponíveis os botões "Seleccionar todos" e "Limpar todos".

Pós-condições

A árvore apresenta um comportamento *tri-state* e é expansível. Todos os nós ficam seleccionados após um clique no botão "Seleccionar todos" e o inverso acontece após um clique no botão "Limpar todos". Cada nó responde correctamente aos sucessivos cliques que são efectuados na respectiva *checkbox* numa sequência de selecção/remover selecção.

Pressupostos

Existe um conjunto de locais, aplicações e tipos de documento codificado no armazém de dados e devidamente relacionado.

A.2.2 Definir o conjunto de serviços a excluir

Actores

Médico

Pré-condições

As quatro caixas de selecção são apresentadas correctamente e permitem selecção múltipla. Se eventualmente existirem configurações prévias, os respectivos serviços encontram-se nas caixas de serviços excluídos.

Pós-condições

Os serviços nas várias caixas respondem correctamente às acções de cada um dos oito botões de movimentação (quatro para serviços requisitantes e quatro para serviços executantes). As acções dos botões são movimentação dos serviços seleccionados do grupo de disponíveis para o grupo de excluídos (e vice-versa) e movimentação de todos os serviços do grupo de disponíveis para o grupo de excluídos (e vice-versa).

Pressupostos

Existe um conjunto de serviços codificado no armazém de dados.

A.2.3 Guardar as preferências apenas na sessão

Actores

Médico

Pré-condições

O botão "Guardar para a sessão" está visível e as preferências estão actualizadas.

Pós-condições

O utilizador recebe uma mensagem informativa do sucesso da operação. Caso tenha sido bem sucedida, as novas configurações ficam activas, caso contrário, permanecem as configurações anteriores.

Pressupostos

Não foi identificado nenhum pressuposto.

A.2.4 Guardar as preferências de forma persistente

Actores

Médico

Pré-condições

O botão "Guardar" está visível e as preferências estão actualizadas.

Pós-condições

O utilizador recebe uma mensagem informativa do sucesso da operação. Caso tenha sido bem sucedida as novas configurações ficam activas, caso contrário, permanecem as configurações anteriores.

Pressupostos

Não foi identificado nenhum pressuposto.

A.2.5 Repor preferências iniciais

Actores

Médico

Pré-condições

O botão "Repor" está visível.

Pós-condições

O utilizador recebe uma mensagem interrogativa acerca da confirmação da acção. Caso o utilizador responda afirmativamente as preferências actuais são perdidas e repostas as que se encontram guardadas de forma persistente. Em caso de resposta negativa, permanecem as configurações actuais.

Pressupostos

Não foi identificado nenhum pressuposto.