



Plataforma de Gestão de Conteúdos Clínicos

JOÃO PAULO CASTRO CARNEIRO

Junho de 2021

Plataforma de Gestão de Conteúdos Clínicos

João Paulo Castro Carneiro

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: Emanuel Silva

Supervisora: Joana Barros

Dedicatória

«Aos meus pais, irmão e namorada pelos sacrifícios feitos ao longo deste percurso.»

Resumo

O setor da saúde, que se encontra em metamorfose constante, rege-se pela integridade da informação, uma vez que, a falta de rigor relativamente ao tratamento dos dados, pode colocar vidas em risco. Um dos exemplos concretos da volatilidade do setor da saúde é o surgimento da pandemia COVID-19, que trouxe novos desafios para este setor. Através do desenvolvimento do software clínico ALERT®, a empresa ALERT pretende dar resposta às contínuas alterações e imprevistos, de forma imediata, íntegra e rigorosa. Neste âmbito, a ALERT motiva-se, não só pelas necessidades dos clientes, como também pela procura de resposta às necessidades do mercado em que se insere.

Atualmente, a equipa de conteúdos da ALERT, constituída maioritariamente por profissionais formados na área da saúde, enfrenta algumas dificuldades, no que diz respeito aos processos de gestão de conteúdos do ALERT®, uma vez que a equipa identifica estes processos como complexos, demorados e sujeitos a erros, tanto ao nível da criação/alteração de conteúdos, como ao nível da visualização dos mesmos.

Neste contexto, o principal objetivo deste projeto, assenta na simplificação da gestão de conteúdos, isto é, na criação, alteração e visualização dos conteúdos, através do desenvolvimento de funcionalidades que auxiliem o cumprimento do objetivo nuclear. Assim, pretende-se que estas funcionalidades sejam implementadas numa plataforma intuitiva e segura, a ser desenvolvida por outro membro da ALERT, que permita manter a rastreabilidade de todas as alterações referentes aos conteúdos. Neste sentido, refletiu-se sobre aspetos cruciais para o desenvolvimento da solução proposta, nomeadamente, a comparação entre a base de dados relacional e não relacional e a comparação entre a implementação da lógica de negócio diretamente na base de dados e na camada de aplicação.

Com o intuito de validar a solução desenvolvida, elaborou-se uma avaliação, que consiste no desenvolvimento de testes de desempenho relativamente ao tempo de resposta das funcionalidades e de um questionário de satisfação, com a finalidade de aferir se, de facto, a solução proposta e desenvolvida resolve os problemas identificados e, cumulativamente, cumpre adequadamente os objetivos inicialmente definidos.

Palavras-chave: Software clínico, Gestão de dados, Desenvolvimento na base de dados

Abstract

The health sector, which is in constant metamorphosis, rules itself by the integrity of the information since the lack of rigor in the data treatment, can put lives at risk. One of the more specific examples of the volatility of the health sector, is the emergence of the COVID-19 pandemic, which brought new challenges to this sector. Through the development of a clinical software named ALERT[®], the company ALERT intends to respond to continuous changes and unforeseen circumstances, in an immediate, incorruptible, and rigorous manner. In this context, ALERT is motivated, not only by the needs of its customers, but also by the search for an answer to the needs of the market in which it operates.

Currently, ALERT's content team, made up mostly of professionals trained in the health field, faces some difficulties regarding ALERT[®] content management processes, since the team identifies these processes as complex, time-consuming, and unreliable, both in terms of content creation/alteration and of visualization.

In this context, the main objective of this project is based on the simplification of content management, that is, on the creation, alteration, and visualization of the contents, through the development of functionalities that assist the fulfillment of the main objective. Thus, it is intended that these functionalities are implemented in an intuitive and secure platform, to be developed by another member of ALERT, which allows to maintain the traceability of all changes related to the contents. In this sense, there was a reflection on crucial aspects for the development of the proposed solution, namely, the comparison between the relational and non-relational database and the comparison between the implementation of the business logic directly in the database and in the application layer.

To validate the developed solution, an evaluation was elaborated, which consists in the development of performance tests regarding the response time of the functionalities and a satisfaction questionnaire, with the purpose of concluding if, in fact, the proposed and developed solution solves the identified problems and, cumulatively, adequately fulfills the initially defined objectives.

Keywords: Clinical software, Data management, Database development

Agradecimentos

Agradeço ao engenheiro, Emanuel Silva por ter aceitado o convite para orientação neste projeto, bem como a disponibilidade total para esclarecimento de dúvidas, conselhos e sugestões para concretização desta etapa.

À empresa ALERT pelo acolhimento e oportunidade e, ainda, à equipa de conteúdos, mais concretamente à orientadora do projeto, Joana Barros, e ao elemento Humberto Cardoso, por todo o apoio, disponibilidade e atenção.

À instituição ISEP e a todos os docentes com quem interagi durante a Licenciatura em Engenharia Informática e no Mestrado em Engenharia Informática, por contribuírem para a minha formação enquanto pessoa e engenheiro informático.

Aos meus pais e irmão por toda a ajuda e sacrifícios ao longo deste percurso, tanto a nível pessoal como académico. Todos eles, e cada um à sua maneira, contribuíram positivamente para o alcance dos meus objetivos, ao longo do percurso académico. Finalmente, mas não menos importante, à minha namorada, Inês Fonseca que sempre acreditou nas minhas capacidades e me tem apoiado durante os últimos cinco anos.

Índice

1 Introdução	1
1.1 Enquadramento	1
1.1.1 A ALERT e o ALERT®.....	1
1.2 Problema.....	2
1.3 Objetivos.....	3
1.4 Motivação.....	4
1.5 Contribuições.....	4
1.6 Metodologia	5
1.7 Estrutura do documento	5
2 Contexto	7
2.1 Categorias de conteúdos no ALERT®.....	7
2.2 Processos de gestão de conteúdos	9
2.2.1 Processo de criação/alteração de conteúdos	9
2.2.2 Processo de visualização de conteúdos	11
2.3 Descrição do Problema.....	11
2.4 Content Server	13
3 Estado da Arte	14
3.1 Software Clínico	14
3.1.1 Glintt	15
3.1.2 F3M	17
3.1.3 MedicineOne.....	18
3.1.4 Conclusão da análise do software clínico.....	20
3.2 Soluções de Gestão de Dados	20
3.2.1 Microsoft Access	20
3.2.2 LibreOffice Base	21
3.3 Base de dados	22
3.3.1 Base de Dados Relacional.....	22
3.3.2 Base de Dados Não Relacional	26
3.3.3 Comparação entre as abordagens de base de dados	30
3.4 Abordagens Arquiteturais.....	31
3.4.1 Implementação da lógica de negócio na base de dados	31
3.4.2 Implementação da lógica de negócio na camada de aplicação.....	33
3.4.3 Comparação das abordagens arquiteturas	35
4 Tecnologias	37
4.1 Base de Dados Oracle	37
4.2 SQL e Oracle SQL	39

4.3	PL/SQL	42
4.4	PL/SQL Developer	45
5	Análise de Valor	46
5.1	Identificação da oportunidade	46
5.2	Análise da Oportunidade	47
5.2.1	Enquadramento Estratégico da Oportunidade	47
5.2.2	Avaliação do Segmento do Mercado.....	49
5.3	Proposta de valor.....	50
5.4	QFD.....	54
5.5	Método de Análise Hierárquica	55
6	Engenharia de Requisitos	69
6.1	Partes Interessadas.....	69
6.2	Atores do Sistema	70
6.3	Requisitos Funcionais	70
6.3.1	UC-01: Gestão de conteúdos do tipo queixa	71
6.3.2	UC-28: Alterar estado de conteúdos	74
6.4	Requisitos Não Funcionais	75
7	Design	77
7.1	Alternativas de Design	77
7.2	Visão de Implantação	79
7.3	Modelo Relacional	80
7.4	Base de Dados.....	82
8	Implementação	85
8.1	Metodologia	85
8.2	Estrutura do Código	86
8.3	Caso de Uso	86
8.3.1	Criação de um conteúdo do tipo exame.....	87
8.3.2	Visualização dos conteúdos do tipo exame	93
8.4	Configuração de Conteúdo	95
8.5	Histórico de Alterações.....	102
9	Experimentação e Avaliação	104
9.1	Indicadores de Avaliação	104
9.2	Hipóteses	104
9.3	Metodologia de Avaliação	105
9.4	Avaliação de Resultados.....	105

9.4.1	Testes de Desempenho.....	106
9.4.2	Inquérito de Satisfação	109
10	Conclusões	116
10.1	Visão Geral	116
10.2	Objetivos Alcançados	117
10.3	Trabalho Futuro.....	118
10.4	Apreciação Final	118
	Referências.....	119
Anexo A.	Testes de tempo de resposta - criação de conteúdos	128
Anexo B.	Testes de tempo de resposta - atualização de conteúdos	135
Anexo C.	Testes de tempo de resposta - atualização do estado de conteúdos.....	142
Anexo D.	Testes de tempo de resposta - obtenção dos dados de conteúdo	149
Anexo E.	Inquérito de Satisfação	155

Lista de Figuras

Figura 1 – Processo atual relativo à criação/alteração de conteúdo (imagem retirada da documentação interna da equipa de conteúdos).....	9
Figura 2 – Componentes do modelo relacional (Melton & Simon, 1993)	23
Figura 3 - Exemplo do armazenamento em bases de dados orientadas a documentos. Retirado de (Microsoft Azure, 2018).	27
Figura 4 – Exemplo do armazenamento em base de dados orientadas a colunas. Retirado de (Microsoft Azure, 2018).	28
Figura 5 – Arquitetura de 3 camadas (<i>three-tier architecture</i>).....	33
Figura 6 – Esquema de base de dados numa base de dados Oracle. Retirado de (Kyte, et al., s.d.).....	38
Figura 7 - Aplicação a invocar um <i>procedure</i> armazenado numa base de dados Oracle. Retirado de (Oracle, s.d. c).....	43
Figura 8 – Estrutura de um bloco PL/SQL. Retirado de (Feuerstein & Pribyl, 2014).	44
Figura 9 – Exemplo de bloco PL/SQL anónimo. Retirado de (Feuerstein & Pribyl, 2014).	44
Figura 10 – Exemplo de bloco PL/SQL com nome. Retirado de (Feuerstein & Pribyl, 2014).	45
Figura 11 – <i>Value Proposition Canvas</i> (Liu, et al., 2020).....	51
Figura 12 – Casa da qualidade (QFD)	55
Figura 13 – Árvore hierárquica de decisão para problema do projeto.....	58
Figura 14 – Árvore hierárquica com as prioridades definidas	67
Figura 15 - Diagrama de casos de uso.....	70
Figura 16 - Diagrama de casos de uso (continuação 1)	71
Figura 17 - Diagrama de casos de uso (continuação 2)	71
Figura 18 – Diagrama de sequência do sistema do UC01 (criar queixa)	73
Figura 19 – Diagrama de sequência do sistema do UC01 (alterar queixa).....	74
Figura 20 – Diagrama de sequência do sistema do UC01 (visualizar queixas)	74
Figura 21 – Diagrama de sequência do sistema do UC25.....	75
Figura 22 - Diagrama de componentes do sistema com a lógica de negócio implementada na base de dados	78
Figura 23 - Diagrama de componentes do sistema com a lógica de negócio implementada na camada de aplicação.....	79
Figura 24 - Diagrama de implantação	80
Figura 25 – Coluna do código de tradução do conteúdo exame	81
Figura 26 – Modelo relacional referente ao conteúdo exame	82
Figura 27 - Vista da estrutura lógica da base de dados	83
Figura 28 – Modelo de domínio do conteúdo do tipo exame	86
Figura 29 - Diagrama de sequência do caso de uso de criação de um conteúdo do tipo exame	88
Figura 30 - Diagrama de sequência do caso de visualização dos conteúdos do tipo exame	93
Figura 31 – Lista de produtos ALERT® para hospitais. Retirado de (ALERT Life Sciences Computing, s.d. d).	96

Figura 32 - Lista de produtos ALERT® para cuidados primários. Retirado de (ALERT Life Sciences Computing, s.d. d).	96
Figura 33 – Lista de produtos ALERT® para clínicas privadas. Retirado de (ALERT Life Sciences Computing, s.d. d).	97
Figura 34 - Lista de outros produtos ALERT®. Retirado de (ALERT Life Sciences Computing, s.d. d).	97
Figura 35 - Diagrama de sequência referente à configuração do contexto mercado num conteúdo do tipo exame	100
Figura 36 – Modelo relacional referente ao histórico das alterações dos conteúdos	102
Figura 37 – Alguns dos testes para avaliar o tempo de resposta das funcionalidades de criação de conteúdos	106
Figura 38 – Alguns dos testes para avaliar o tempo de resposta das funcionalidades de atualização de conteúdos	107
Figura 39 - Alguns dos testes para avaliar o tempo de resposta das funcionalidades de atualização do estado de conteúdos	108
Figura 40 - Alguns dos testes para avaliar o tempo de resposta das funcionalidades de obtenção dos dados de conteúdos	109
Figura 41 – Resultados da primeira questão do inquérito de satisfação	110
Figura 42 – Resultados da segunda questão do inquérito de satisfação	110
Figura 43 - Resultados da terceira questão do inquérito de satisfação	111
Figura 44 - Resultados da quarta questão do inquérito de satisfação	111
Figura 45 - Resultados da quinta questão do inquérito de satisfação	112
Figura 46 - Resultados da sexta questão do inquérito de satisfação	112
Figura 47 - Resultados da sétima questão do inquérito de satisfação	113
Figura 48 - Resultados da oitava questão do inquérito de satisfação	113
Figura 49 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos	130
Figura 50 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 1)	131
Figura 51 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 2)	132
Figura 52 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 3)	133
Figura 53 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 4)	134
Figura 54 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos	137
Figura 55 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 1)	138
Figura 56 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 2)	139
Figura 57 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 3)	140

Figura 58 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 4)	141
Figura 59 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos.....	144
Figura 60 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 1).....	145
Figura 61 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 2).....	146
Figura 62 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 3).....	147
Figura 63 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 4).....	148
Figura 64 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos	150
Figura 65 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 1)	151
Figura 66 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 2)	152
Figura 67 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 3)	153
Figura 68 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 4)	154
Figura 69 - Inquérito de satisfação.....	156
Figura 70 - Inquérito de satisfação (continuação - 1)	157
Figura 71 - Inquérito de satisfação (continuação - 2)	158

Lista de Tabelas

Tabela 1 - Categorias de conteúdo	8
Tabela 2 – Base de dados relacional VS Base de dados não relacional	30
Tabela 3 – Implementação da lógica de negócio na base de dados VS lógica de negócio na camada de aplicação	35
Tabela 4 – Análise SWOT.....	47
Tabela 5 – Escala fundamental – Níveis de importância de comparações (Saaty, 2002).....	56
Tabela 6 – Valores de IR para matrizes quadradas de ordem n (Al-Harbi, 2001).....	57
Tabela 7 – Matriz de comparação dos critérios	59
Tabela 8 – Matriz de comparação de critérios normalizada (1º Passo)	59
Tabela 9 – Matriz de comparação de critérios normalizada.....	59
Tabela 10 – Vetor de prioridades da matriz de comparação dos critérios.....	60
Tabela 11 – Matriz de comparação de alternativas para o critério C1.....	61
Tabela 12 - Matriz de comparação de alternativas para o critério C1 normalizada (1º Passo) .	61
Tabela 13 – Matriz de comparação de alternativas para o critério C1 normalizada.....	61
Tabela 14 - Vetor de prioridades da matriz de comparação de alternativas para o critério C1	62
Tabela 15 - Matriz de comparação de alternativas para o critério C2	62
Tabela 16 - Matriz de comparação de alternativas para o critério C2 normalizada (1º Passo) .	63
Tabela 17 - Matriz de comparação de alternativas para o critério C2 normalizada	63
Tabela 18 - Vetor de prioridades da matriz de comparação de alternativas para o critério C2	63
Tabela 19 - Matriz de comparação de alternativas para o critério C3	64
Tabela 20 - Matriz de comparação de alternativas para o critério C3 normalizada (1º Passo) .	64
Tabela 21 - Matriz de comparação de alternativas para o critério C3 normalizada	64
Tabela 22 - Vetor de prioridades da matriz de comparação de alternativas para o critério C3	65
Tabela 23 - Matriz de comparação de alternativas para o critério C4	65
Tabela 24 - Matriz de comparação de alternativas para o critério C4 normalizada (1º Passo) .	66
Tabela 25 - Matriz de comparação de alternativas para o critério C4 normalizada	66
Tabela 26 - Vetor de prioridades da matriz de comparação de alternativas para o critério C4	66
Tabela 27 – Diferentes estados de um conteúdo	72
Tabela 28 – Requisitos não funcionais de acordo com modelo FURPS+	75
Tabela 29 – Metodologia adotadas para cada indicador.....	105
Tabela 30 – As respostas do inquérito de satisfação e a respetiva pontuação numérica.....	109
Tabela 31 – Número de respostas para cada tipo de resposta possível	114
Tabela 32 – Objetivos do projeto e respetivo grau de implementação	117
Tabela 33 – Casos de uso e respetivo grau de implementação.....	117

Lista de Extratos de Código

Extrato Código 1 – Especificação do <i>procedure create_content</i>	89
Extrato Código 2 – Invocação do <i>procedure exam_validation</i>	90
Extrato Código 3 – Invocação do <i>procedure get_content_type_id_by_internal_name_or_content_id</i>	91
Extrato Código 4 – Invocação dos <i>procedures create_exam_body_structure_rel e create_exam_type_rel</i>	91
Extrato Código 5 – Inserção dos dados relativos a um exame na tabela <i>exam</i>	92
Extrato Código 6 – Invocação do <i>procedure register_translation</i>	92
Extrato Código 7 – Especificação da <i>function get_content</i>	94
Extrato Código 8 – Implementação da lógica para obtenção da lista de exames a retornar.....	94
Extrato Código 9 – Especificação do <i>procedure set_context</i>	99
Extrato Código 10 – Especificação do <i>procedure set_context_status</i>	99
Extrato Código 11 – Especificação da <i>function get_context</i>	100
Extrato Código 12 – Implementação do <i>procedure set_context</i>	101
Extrato Código 13 – Especificação dos <i>procedures set_content_history e set_status_history</i>	103
Extrato Código 14 - Teste de normalidade <i>Shapiro-Wilk</i>	114
Extrato Código 15 – Teste não paramétrico de <i>Wilcoxon</i>	115
Extrato Código 16 – Teste de tempo de resposta da funcionalidade de criação de conteúdos do tipo categoria de queixa	129
Extrato Código 17 - Teste de tempo de resposta da funcionalidade de atualização de conteúdos do tipo categoria de queixa	136
Extrato Código 18 - Teste de tempo de resposta da funcionalidade de atualização do estado de conteúdos do tipo categoria de queixa	143
Extrato Código 19 - Teste de tempo de resposta da funcionalidade de obtenção de dados de conteúdos do tipo categoria de queixa	149

Acrónimos e Símbolos

Lista de Acrónimos

ACID	<i>Atomicity, Consistency, Isolation, Durability</i>
AHP	<i>Analytic Hierarchy Process</i>
ALERT	<i>ALERT Life Sciences Computing, S.A</i>
ALERT®	Software clínico desenvolvido e comercializado pela ALERT
ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
BASE	<i>Basically Available, Soft state, Eventual consistency</i>
DBMS	<i>Database Management System</i>
DDL	<i>Data Definition Language</i>
DML	<i>Data Manipulation Language</i>
EHR	<i>Electronic Health Record</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IASFA/ADM	Instituto de Ação Social das Forças Armadas/Assistência na Doença aos Militares
ICNP	<i>International Classification for Nursing Practice</i>
IDE	<i>Integrated Development Environment</i>
ISEP	Instituto Superior de Engenharia do Porto
JSON	<i>JavaScript Object Notation</i>
MCDT	Meios complementares de diagnóstico e terapêutica
NoSQL	<i>Not only SQL</i>
ODBC	<i>Open Database Connectivity</i>
ORDBMS	<i>Object-Relational Database Management System</i>
PDF	<i>Portable Document Format</i>

PL/SQL	<i>Procedural Language/Structured Query Language</i>
QEF	<i>Quantitative Evaluation Framework</i>
QFD	<i>Quality function deployment</i>
RDBMS	<i>Relational Database Management System</i>
SaaS	<i>Software as a Service</i>
SAD-GNR	Saúde e Assistência na Doença da Guarda Nacional Republicana
SAD-PSP	Saúde e Assistência na Doença da Polícia de Segurança Pública
SNS	Serviço Nacional de Saúde
SQL	<i>Structured Query Language</i>
TMDEI	Unidade curricular de Tese/Dissertação/Estágio
VBA	<i>Visual Basic for Applications</i>

Lista de Símbolos

IC	Índice de consistência
IR	Índice aleatório
RC	Razão de consistência

1 Introdução

Este capítulo descreve o enquadramento do trabalho, identifica o problema, os objetivos definidos a atingir, a motivação para a realização da dissertação e contribuições da mesma, a metodologia adotada no projeto, e, por fim, a estrutura do documento.

1.1 Enquadramento

Este documento resulta de um projeto de estágio realizado na empresa ALERT Life Sciences Computing (ALERT), no âmbito da unidade curricular do segundo ano, Tese/Dissertação/Estágio (TMDEI) do Mestrado em Engenharia Informática, na área de especialização de Engenharia de Software, do Instituto Superior de Engenharia do Porto (ISEP).

O trabalho desenvolvido tem como foco a gestão dos conteúdos utilizados no ALERT® e procura atingir uma solução que permita facilitar a gestão dos mesmos por parte da equipa de conteúdos, e, também, reduzir o tempo necessário para a sua disponibilização para os clientes.

Neste sentido, na secção seguinte, elabora-se uma breve apresentação da ALERT e do seu produto, o ALERT®.

1.1.1 A ALERT e o ALERT®¹

Fundada no ano de 1999, pelo médico pós-doutorado Doutor Jorge Guimarães, a ALERT dedica-se ao desenvolvimento de uma solução integrada para a informatização completa dos cuidados de saúde em hospitais e centros de saúde, atuando, inclusive, a nível nacional (ALERT Life Sciences Computing, s.d. a). A missão da ALERT é “Melhorar a saúde e prolongar a vida, alcançar

¹ No decorrer do presente documento, “ALERT” (sem o símbolo de marca registada) refere-se à empresa ALERT Life Sciences Computing, S.A., enquanto que “ALERT®” refere-se ao produto (software).

rentabilidade para benefício da sociedade e inspirar outros para a excelência através do nosso exemplo.” (ALERT Life Sciences Computing, s.d. b).

O ALERT® é uma solução clínica internacional, adotado em diversas geografias no âmbito de projetos individuais de hospitais ou iniciativas de carácter nacional ou regional. A título de exemplo, alguns dos projetos foram implementados em Portugal, Espanha, França, Itália, Reino Unido, Suíça, Estados Unidos da América (EUA), Brasil, México, Chile, Angola, Moçambique, Kuwait e Arábia Saudita (ALERT Life Sciences Computing, s.d. b).

As soluções ALERT®, aptas para *web* e nuvem, são variantes de diferentes configurações de um só produto, com uma única linha de desenvolvimento que se diferencia pela possibilidade de configuração local em cada mercado. Estas particularidades permitem que a ALERT se foque num único produto universal, em contínua mudança, que agrega contribuições dos diferentes mercados de todo o mundo (ALERT Life Sciences Computing, s.d. c).

A ALERT pode implementar, em cada mercado, uma instalação local ou um modelo transacional, o qual disponibiliza um *Software as a Service* (SaaS²), que contempla uma interligação entre diferentes prestadores de cuidados de saúde e respetivos pacientes. Adicionalmente, possui a capacidade de monitorizar, de forma remota, e de prestar suporte reativo e preventivo, 24 horas por dia e 365 dias por ano, aos seus clientes através da equipa de Operações (ALERT Life Sciences Computing, s.d. c).

1.2 Problema

O ALERT® encontra-se inserido num mercado extremamente exigente, o da saúde. No contexto deste setor, existem fatores cruciais, como é o caso dos registos clínicos eletrónicos de um paciente, que desempenham um papel fundamental no que diz respeito à prestação de cuidados de saúde. Adicionalmente, existe a necessidade de manter os conteúdos, clínicos e não clínicos, atualizados, devido à constante mutação do panorama da saúde. Neste âmbito, a ALERT motiva-se, não só pelas necessidades dos clientes, como também, pela procura de resposta às necessidades do mercado em que se insere.

Atualmente, a equipa de conteúdos da ALERT, enfrenta algumas dificuldades no que diz respeito aos processos de gestão de conteúdos, uma vez que se trata de processos que a equipa identifica como complexos, demorados e sujeito a erros, tanto ao nível da criação/alteração de conteúdos, como ao nível da visualização dos mesmos.

No que diz respeito ao processo da criação/alteração de conteúdos destacam-se os seguintes problemas:

- Necessidade de conhecimentos da linguagem SQL, uma linguagem de programação que se encontra fora do âmbito de formação dos membros da equipa de conteúdos;

² SaaS – Forma de distribuição de software em que os fornecedores de software são totalmente responsáveis pela gestão do software e disponibilizam o mesmo para os seus clientes através da internet (Liu, et al., 2020).

- Necessidade de conhecimento profundo do modelo de dados para fazer a gestão dos conteúdos, por parte da equipa de conteúdos;
- Envolvimento de mais do que uma equipa no processo de criação/alteração, para além da própria equipa de conteúdos.

No que concerne o processo de visualização de conteúdos, também se salientam alguns obstáculos, nomeadamente:

- Necessidade de aceder diretamente à base de dados para visualizar os conteúdos existentes;
- Necessidade de desenvolver e executar *queries* SQL, o que, mais uma vez, requer conhecimento da linguagem de programação SQL.

Esta problemática, abordada em pormenor na secção 2.3, afeta não só a própria equipa de conteúdos, como também o cliente do ALERT®, uma vez que atrasa a disponibilização de novos conteúdos e atualizações de conteúdos existentes.

1.3 Objetivos

O principal objetivo deste projeto assenta na simplificação da gestão de conteúdos, isto é, na criação, alteração e visualização dos conteúdos, através do desenvolvimento de funcionalidades que auxiliem o cumprimento do objetivo. Pretende-se que estas funcionalidades sejam implementadas numa plataforma intuitiva e segura, desenvolvida pela ALERT, e que permita manter a rastreabilidade de todas as alterações referentes aos conteúdos.

Para que o objetivo referido anteriormente seja atingido, foram estabelecidas as seguintes tarefas:

- Integrar na equipa de conteúdos;
- Analisar a estrutura dos diversos tipos de conteúdos e configurações existentes nos produtos ALERT®;
- Analisar as ferramentas já em uso na empresa, em particular para gestão de conteúdos.

No âmbito do objetivo principal, a simplificação da gestão dos conteúdos, identificaram-se quatro objetivos específicos:

- Reformular/Atualizar o modelo de dados existente;
- Desenvolver funcionalidades para uma nova plataforma que permita:
 - Listar todos os conteúdos existentes, para cada tipo de conteúdo, e respetivas configurações;
 - Efetuar alterações a cada um dos conteúdos e configurações, salvaguardando a coerência dos dados;

- Manter um registo automático de todas as alterações efetuadas.
- Documentar todo o desenvolvimento efetuado, bem como as regras implementadas;
- Testar as funcionalidades desenvolvidas.

No decorrer do presente documento, o conjunto de funcionalidades a desenvolver para uma nova plataforma que permita a gestão de conteúdos, é identificado como o *back-end*³ da mesma.

1.4 Motivação

A escolha deste projeto deve-se ao interesse do autor pela área da saúde, uma vez que se trata de um setor de extrema importância para a sociedade, cada vez mais dependente da utilização de software clínico.

De facto, o panorama atual de contexto pandémico sensibilizou o autor para a contínua e crescente necessidade de software clínico, não só para a garantia do bem-estar da sociedade, como também para a eficácia e eficiência do trabalho dos profissionais de saúde, nas unidades hospitalares.

Adicionalmente, sendo o irmão do autor enfermeiro, pode acrescentar-se que este lidou de perto com as dificuldades dos profissionais de saúde, e que se tornou mais consciente do valor do software clínico na atualidade.

1.5 Contribuições

O projeto descrito neste documento provém da identificação de necessidades de melhoria na gestão dos conteúdos do ALERT®, que, causam dificuldades para os membros da equipa de conteúdos, no momento da gestão dos mesmos, e, adicionalmente, podem atrasar a sua disponibilização no ALERT®.

Com o desenvolvimento deste projeto, espera-se concretizar duas grandes contribuições:

- Uniformizar e melhorar o modelo de dados existente, reformulando-o, para refletir as necessidades de alterações identificadas pela equipa de conteúdos no modelo atual;
- Agilizar o trabalho da equipa de conteúdos, no que diz respeito à gestão de conteúdos, recorrendo ao desenvolvimento de funcionalidades que se pretende que sejam invocadas por uma plataforma com interface gráfica, desenvolvida pela ALERT. Adicionalmente, espera-se que seja possível disponibilizar novos conteúdos, e atualizações a conteúdos já existentes, para os clientes do ALERT® de forma mais breve.

³ *Back-end* corresponde à parte de um programa que um utilizador não vê.

1.6 Metodologia

No decorrer do desenvolvimento da solução, referente ao presente documento, adotou-se uma metodologia ágil, mais concretamente a metodologia *Kanban*. A utilização desta metodologia assenta na utilização de um quadro (*board*) *Kanban*, que consiste numa “[...] ferramenta ágil de gestão de projetos projetada para ajudar a visualizar o trabalho, limitar o trabalho a decorrer, e maximizar a eficiência (ou fluxo).”⁴.

Segundo o site da Atlassian, a metodologia *Kanban* “[...] depende da total transparência de trabalho e comunicação de capacidade em tempo real, assim, o quadro *Kanban* deve ser visto como a única fonte de verdade para o trabalho da equipa.”⁵.

Neste sentido, no decorrer do estágio foi utilizado a ferramenta Jira Software, que permite a utilização da metodologia *Kanban*. Assim, no início do estágio, foram definidos quais os conteúdos a trabalhar e priorizar, no âmbito deste projeto, sendo estes, posteriormente, transformados em *issues* que foram adicionados ao *backlog*.

1.7 Estrutura do documento

Este documento está dividido em dez capítulos: (1) introdução, (2) contexto, (3) estado da arte, (4) tecnologias, (5) análise de valor, (6) engenharia de requisitos, (7) design, (8) implementação, (9) experimentação e avaliação e (10) conclusões. Cada capítulo está subdividido em várias secções, sendo que algumas contêm subsecções.

No capítulo 1 – Introdução – descreve-se o enquadramento do trabalho, identifica-se o problema, os objetivos definidos a atingir, a motivação para a realização da dissertação e as respetivas contribuições, e, por fim, apresenta-se a metodologia adotada.

No capítulo 2 – Contexto – descrevem-se as diversas categorias de conteúdos utilizadas no ALERT®, os processos aplicados na gestão de conteúdos, apresenta-se uma descrição detalhada do problema, e, ainda, se aborda o *Content Server*.

No capítulo 3 – Estado da Arte – elabora-se uma análise de mercado, no que diz respeito a soluções existentes e pertinentes, tanto ao nível das soluções clínicas, como ao nível das soluções no contexto do presente projeto. Adicionalmente, efetua-se uma análise comparativa de diferentes abordagens que podem ser empregues, no que diz respeito à componente de persistência de informação e ao desenvolvimento do *back-end* do *Content Server*.

⁴ Tradução livre do autor. No original “A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow).” (Atlassian, s.d a).

⁵ Tradução livre do autor. No original “The kanban methodology relies upon full transparency of work and real-time communication of capacity, therefore the kanban board should be seen as the single source of truth for the team’s work.” (Atlassian, s.d. b).

No capítulo 4 – Tecnologias – apresentam-se as diferentes tecnologias relevantes para o desenvolvimento da solução deste projeto.

No capítulo 5 – Análise de Valor – elabora-se uma análise à oportunidade, é identificada a proposta de valor do projeto, é apresentado o *Quality function deployment* (QFD), e, por fim, é aplicado o método de análise hierárquica, *Analytic Hierarchy Process* (AHP).

No capítulo 6 – Engenharia de Requisitos – identificam-se as partes interessadas e os atores deste projeto, os requisitos funcionais e os requisitos não funcionais.

No capítulo 7 – Design – apresentam-se as alternativas de design, identificando a alternativa que melhor se adequa ao projeto, a vista de implantação do sistema, o modelo relacional parcial da solução, e, ainda, a estrutura da base de dados.

No capítulo 8 – Implementação – aborda-se, inicialmente, a metodologia adotada no desenvolvimento da solução, de seguida é apresentada a estrutura do código, e, por fim, apresentam-se detalhes relativos à implementação de determinados casos de uso do projeto.

No capítulo 9 – Experimentação e Avaliação - apresentam-se os indicadores adotados para a avaliação do projeto, de seguida elaboram-se as hipóteses, e, por fim, apresentam-se as metodologias de avaliação escolhidas para a avaliação do projeto, bem como os respetivos resultados.

No capítulo 10 – Conclusões – apresenta-se uma breve visão geral do trabalho desenvolvido no âmbito do projeto de estágio na ALERT, os objetivos alcançados, é descrito o trabalho a desenvolver no futuro, e, por fim, é elaborada uma apreciação final sobre o projeto.

2 Contexto

Este capítulo apresenta a contextualização dos aspetos considerados cruciais do projeto, tendo em vista um esclarecimento adequado do leitor. Assim, são apresentadas as (1) categorias de conteúdos utilizadas no ALERT®, os (2) processos aplicados na gestão de conteúdos, uma (3) descrição detalhada do problema e, por fim, a (4) plataforma de gestão de conteúdos denominada de *Content Server*.

2.1 Categorias de conteúdos no ALERT®

Como referido na secção 1.1.1, o ALERT® é uma ferramenta operacional de fácil utilização, que tira partido de tecnologias *touchscreen*. Esta ferramenta contém vários perfis de utilizador para cada profissional de saúde e está orientada para o ambiente de trabalho, permitindo o registo de informação e o acesso à mesma em tempo real. Para além de interligar a atividade de todos os profissionais, através de *workflows* e de conteúdos adequados a cada situação clínica, o ALERT® apoia também o profissional de saúde no processo de decisão clínica.

Deste modo, por conteúdos entende-se toda a informação relevante para os vários perfis de utilizador, entre outros, o histórico clínico de um paciente.

Atualmente existem mais de 100 tipos de conteúdos no ALERT®, sendo que estes se podem inserir numa das cinco categorias de conteúdos – documentação, outros perfis, conteúdo não clínico, enfermagem e ordens - como descrito na Tabela 1.

Tabela 1 - Categorias de conteúdo

Categorias de conteúdo	Exemplos
Documentação	<ul style="list-style-type: none"> • Histórico clínico do paciente; • Diagnósticos; • Alergias; • Modelos de opções de toque; • Complicações na gravidez;
Outros perfis	<ul style="list-style-type: none"> • Psicólogo; • Assistente social; • Nutricionista; • Fisioterapeuta;
Conteúdo não clínico	<ul style="list-style-type: none"> • Religiões; • Transportes; • Relações familiares; • Planos de saúde; • Tipos de documentos;
Enfermagem	<ul style="list-style-type: none"> • <i>International Classification for Nursing Practice (ICNP⁶);</i> • Sinais vitais; • Vacinas;
Ordens	<ul style="list-style-type: none"> • Medicação; • Procedimentos cirúrgicos; • Exames; • Testes de laboratório;

Cada conteúdo do ALERT® tem as suas próprias regras de negócio, validações, dependências com outros conteúdos e configurações. Um conteúdo pode ser configurado de forma a definir que o mesmo esteja apenas disponível para um determinado software, uma determinada versão, um determinado mercado e serviço clínico, entre outros.

Atualmente, os conteúdos podem estar associados a vários mercados, o que se reflete na existência de traduções para cada um dos idiomas do mercado. Assim, existem traduções dos conteúdos para os seguintes idiomas:

- Árabe (Arábia Saudita);
- Chinês (Taiwan);
- Espanhol (Chile);
- Espanhol (Espanha);
- Espanhol (México);
- Francês (França);
- Francês (Suíça);

⁶ ICNP – “A *International Classification for Nursing Practice (ICNP)* fornece um conjunto acordado de termos que podem ser usados para registrar as observações e intervenções de enfermeiros no mundo inteiro. [...]” (Anon., s.d.) (tradução livre do autor).

- Holandês;
- Inglês (Arábia Saudita);
- Inglês (Estados Unidos da América);
- Inglês (Reino Unido);
- Italiano;
- Português (Angola);
- Português (Brasil);
- Português (Moçambique);
- Português (Portugal).

Embora o mesmo idioma seja utilizado em vários países, como é o caso do espanhol, francês, inglês e português, para cada país existem traduções próprias, uma vez que a nomenclatura de determinados conceitos difere de país para país.

2.2 Processos de gestão de conteúdos

A equipa de conteúdos é responsável pela gestão do ciclo de vida dos conteúdos, isto envolve os seguintes processos: criação/alteração de conteúdos e visualização dos conteúdos. Nas secções 2.2.1 e 2.2.2 descrevem-se, detalhadamente, os processos de criação/alteração de conteúdos e o processo de visualização de conteúdos, respetivamente.

2.2.1 Processo de criação/alteração de conteúdos

Na Figura 1 é representado o processo aplicado à criação/alteração de conteúdos, que se encontra em uso na ALERT há mais de 10 anos. Atualmente, a criação/alteração de conteúdo envolve o esforço de vários intervenientes, entre os quais: equipa de conteúdos, cliente/equipa de um projeto, equipa de configuração e, ainda, a equipa de versionamento.

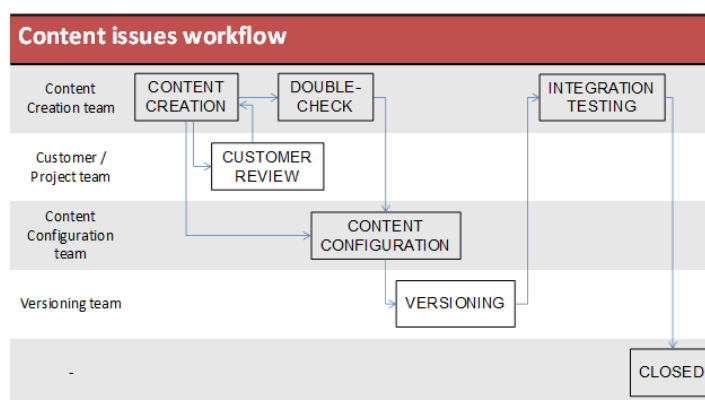


Figura 1 – Processo atual relativo à criação/alteração de conteúdo (imagem retirada da documentação interna da equipa de conteúdos)

A criação/alteração de conteúdos para o ALERT® pode resultar de pedidos de diferentes origens, entre os quais:

- A pedido de um cliente: estes pedidos são identificados pela equipa de Suporte e Operação;
- A pedido interno da equipa de conteúdos: pedidos que surgem da necessidade de criar e otimizar conteúdos *default* do ALERT®;
- A pedido da equipa de *Demos*: pedidos que surgem da necessidade de criação de conteúdos específicos para *demos* de potenciais clientes;
- A pedido da equipa de desenvolvimento: pedidos que surgem da necessidade de adição de conteúdos para novas funcionalidades.

Para a criação/alteração de conteúdos é necessário que os membros da equipa de conteúdos tenham conhecimento em *Structured Query Language (SQL)*⁷. Para além do conhecimento em SQL, os membros da equipa têm de utilizar as ferramentas: Excel e *PL/SQL Developer*⁸.

A equipa de conteúdos criou, ao longo do tempo, vários ficheiros Excel, um para cada um dos tipos de conteúdos do ALERT®. Estes ficheiros Excel são configurados de forma a que um membro da equipa possa preencher um conjunto de campos que definam o conteúdo para a sua criação/alteração. Com base na configuração dos ficheiros Excel e com recurso a fórmulas Excel, é possível obter *scripts* SQL, que representam a criação/alteração de um conteúdo.

Relativamente ao processo de criação/alteração de conteúdos, primeiramente, um membro da equipa de conteúdos inicia o processo, através do preenchimento do ficheiro Excel correspondente ao tipo de conteúdo que pretende trabalhar.

Em seguida, o conteúdo é verificado por outro membro da equipa, com o objetivo de assegurar que o conteúdo criado/alterado é íntegro e corresponde ao pretendido.

Existem situações em que após a verificação do conteúdo criado/alterado, ocorre, ainda, uma validação por parte do cliente/equipa do projeto, com a finalidade de verificar se existe alguma incoerência do conteúdo em relação ao desejado.

Seguidamente, o membro da equipa de conteúdos faz as alterações necessárias na base de dados de desenvolvimento, utilizando o *script* obtido na fase inicial, que é enviado para o versionamento.

Terminada esta fase, inicia-se o processo de versionamento, realizado pela equipa de versionamento, que executa os *scripts* das alterações nas bases de dados de ambiente de controlo.

⁷ “SQL, é uma linguagem de dados usada para aceder a bases de dados relacionais [...]” (Melton, et al., 1998) (tradução livre do autor).

⁸ *PL/SQL Developer* (Automations, s.d.) - “O *PL/SQL Developer* é um *Integrated Development Environment (IDE)* orientado especificamente para o desenvolvimento de unidades de programa armazenadas para bases de dados Oracle.” (tradução livre do autor).

No ambiente de controlo, a equipa de conteúdos realiza testes de integração para verificar se tudo ocorreu como previsto. Caso se verifique algum erro ou incoerência, relativamente ao esperado, a equipa de conteúdos deve corrigi-los através dos novos *scripts* que solucionem o problema identificado. Por fim, os *scripts* devem ser enviados para versionamento novamente.

2.2.2 Processo de visualização de conteúdos

Relativamente ao processo de visualização de conteúdos, os membros da equipa de conteúdos necessitam do *PL/SQL Developer*. Através desta ferramenta, é efetuada a conexão ao *schema* da base de dados, onde os conteúdos se encontram armazenados, e são desenvolvidas e executadas as *SQL queries*. Por vezes as *SQL queries* podem ser complexas, tendo em conta as diferentes configurações que um conteúdo pode apresentar, e, também, tendo em conta a possível necessidade de obter informação que se encontra dispersa por diferentes tabelas de diferentes *schemas*.

2.3 Descrição do Problema

O desenvolvimento desta dissertação prende-se com o facto de existirem vários problemas identificados ao longo do tempo por parte da equipa de conteúdos, no que diz respeito à gestão dos conteúdos do ALERT®. Deste modo, o principal problema está relacionado com a complexidade do processo de gestão de conteúdos, tanto ao nível da criação/alteração de conteúdos, como ao nível da visualização dos mesmos.

A equipa de conteúdos é constituída, maioritariamente, por membros com formação na área da saúde com pouca, ou nenhuma, formação no âmbito de SQL. Tendo em conta que, atualmente, o processo de criação/alteração de conteúdos é complexo para a equipa de conteúdos, considerou-se a simplificação deste processo um fator de importância prioritária.

De facto, o processo é complexo, na medida em que, exige conhecimentos em SQL, uma linguagem de programação pouco acessível, já que se encontra fora do contexto profissional dos membros da equipa de conteúdos. Consequentemente, verifica-se a necessidade de os membros da equipa de conteúdos terem conhecimento profundo do modelo de dados para fazer a gestão dos conteúdos, o que, perante a existência de múltiplos tipos de conteúdos, pode tornar-se difícil para a equipa.

A complexidade do processo de criação/alteração também se verifica na secção 2.2.1, onde se pode constatar que o trabalho da equipa de conteúdos está dependente do trabalho de outras equipas. O envolvimento de várias equipas no processo de criação/alteração de conteúdos atrasa a disponibilização do mesmo, uma vez que é necessário esperar pelo lançamento de uma nova versão do ALERT®, pelo que os conteúdos, muitas vezes de carácter urgente, não ficam imediatamente disponíveis para os clientes.

Um exemplo concreto desta problemática verifica-se no processo de criação/alteração de conteúdos, descrito na secção 2.2.1, onde, inicialmente, são criados/alterados os conteúdos, e obtidos os respetivos scripts da criação/alteração, pela equipa de conteúdos. No entanto, a disponibilização dos conteúdos nas bases de dados é efetuada mais tarde, por outra equipa, o que, tal como referido anteriormente, atrasa a disponibilização dos conteúdos para os clientes do ALERT®.

Assim, a equipa de conteúdos pretende agilizar o processo de criação/alteração de conteúdos, através da redução da intervenção de outras equipas, assumindo, conseqüentemente, o total controlo no que diz respeito aos conteúdos.

Tendo em conta a, já referida, intervenção de várias equipas, surge outro problema relacionado com a complexidade dos processos de criação/alteração de conteúdos, desta vez, no âmbito da rastreabilidade. A rastreabilidade é um processo fulcral para a equipa de conteúdos, na medida em que, permite saber, a qualquer momento, o porquê, por quem e quando se deu uma alteração em determinado conteúdo, e o que foi alterado.

Neste momento, a rastreabilidade da criação/alteração de objetos de base dados, tais como, tabelas, índices e manipulação de dados das tabelas – adicionar e atualizar entradas nas tabelas - é da responsabilidade da equipa de versionamento, e não da equipa de conteúdos. Mais uma vez, o recurso a outra equipa durante o processo de criação/alteração de conteúdos, pode atrasar a disponibilização dos mesmos.

Assim, a equipa de conteúdos pretende, novamente, agilizar o processo de criação/alteração de conteúdos, ao retirar responsabilidade à equipa de versionamento, no que diz respeito à rastreabilidade da manipulação de dados das tabelas, para que a disponibilização dos conteúdos se torne mais rápida e, adicionalmente, tornar o rastreamento mais simples, uma vez que, segundo a equipa, atualmente, perceber a evolução de um conteúdo é considerado complexo.

No que diz respeito ao processo de visualização de conteúdos, descrito na secção 2.2.2, destacam-se, também, alguns problemas.

Atualmente, os membros da equipa de conteúdos necessitam da ferramenta *PL/SQL Developer* para aceder diretamente à base de dados, e, adicionalmente, de desenvolver e executar *queries* SQL. Esta necessidade torna o processo de visualização complexo, para a equipa de conteúdos, na medida em que gera inconveniência.

De facto, o desenvolvimento destas *queries* pode revelar-se complexo e inconveniente para a equipa, uma vez que, entre outros, exige, mais uma vez, um conhecimento profundo do modelo de dados de um determinado conteúdo. Desta forma, o facto de os membros terem pouca, ou nenhuma, formação em SQL, dificulta, desta vez, o processo de visualização de conteúdos, uma vez que o torna complexo para os membros da equipa de conteúdos. Adicionalmente, a elaboração das *queries*, que são apresentadas de uma forma extensa e pouco prática, está sujeita a erros que podem gerar resultados indesejados, o que pode traduzir-se em tempo despendido.

2.4 Content Server

Como referido na secção 1.3, este projeto tem como principal objetivo a simplificação da gestão dos conteúdos do ALERT®. Assim, pretende atingir-se esta simplificação através da criação de uma plataforma que permita aos membros da equipa de conteúdos fazer a gestão dos conteúdos de uma forma simples e prática.

Neste sentido, no presente projeto, pretende-se que sejam desenvolvidas funcionalidades que deem forma ao *back-end*. O *back-end*, por sua vez, tem como finalidade ser consumido por uma plataforma de gestão de conteúdos, com interface gráfica, cujo objetivo é simplificar o trabalho dos membros da equipa de conteúdos, no que diz respeito à gestão dos conteúdos.

Embora a componente gráfica deste projeto esteja fora do âmbito desta dissertação, considera-se importante dar a conhecer as decisões tomadas pela equipa de conteúdos, no que diz respeito à interface gráfica do projeto.

Neste âmbito, relativamente à interface gráfica, o Excel foi definido pela equipa de conteúdos como a plataforma para a gestão dos conteúdos do ALERT®. Assim, a equipa de conteúdos pensou na criação de um *add-in*⁹ para o Excel, ao qual deu o nome de *Content Server*, de forma a permitir que o Excel suporte funcionalidades que habilitem a gestão de conteúdos. De facto, o *Content Server* permite que os membros da equipa de conteúdos façam a gestão dos conteúdos, inteiramente a partir do Excel, evitando o uso de outras ferramentas. Pretende-se que o *Content Server*, que se encontra a ser desenvolvido por outro membro da equipa, invoque as funcionalidades implementadas no *back-end*.

A utilização do Excel como interface gráfica é uma restrição imposta pela equipa de conteúdos para este projeto. Tal fica a dever-se ao facto de a equipa identificar o Excel como uma ferramenta com a qual já estão familiarizados, visto que a utilizam nas suas tarefas diárias, pelo que pretendem continuar a utilizá-la. Os membros da equipa consideram, ainda, que a interface gráfica do Excel é *user friendly* e, conseqüentemente, de fácil aprendizagem, o que facilita a execução das suas tarefas. Adicionalmente, o Excel fornece funcionalidades que tornam o tratamento de dados flexível, entre as quais, a possibilidade de encontrar registos duplicados, ordenar e/ou filtrar registos. Por fim, a equipa caracteriza o Excel como uma ferramenta altamente personalizável, o que permite que cada um dos membros da equipa o customize de acordo com as suas preferências, facilitando o seu trabalho.

⁹ Segundo o *website* da Microsoft um "*Add-in* fornece instruções e recursos opcionais para o Microsoft Excel." [tradução livre do autor de (Microsoft, s.d.)].

3 Estado da Arte

No desenvolvimento de um projeto é necessário, inicialmente, realizar um estudo do conhecimento já existente, relevante para o mesmo. Desta forma, no âmbito deste projeto, no presente capítulo, é apresentado o resultado do estudo realizado sobre o estado da arte, através de uma análise do mercado, no que diz respeito a soluções existentes e pertinentes, tanto ao nível das soluções de software clínico, como ao nível das soluções no contexto do presente projeto. Seguidamente, é apresentada uma análise comparativa relativamente às diferentes alternativas que podem ser empregues, no que diz respeito à componente de persistência de informação (base de dados), e é apresentada a escolha mais pertinente. Adicionalmente, é feita uma análise de diferentes abordagens arquiteturas, que podem ser adotadas para o desenvolvimento do *back-end* do *Contente Server*, em seguida, é apresentada uma comparação entre as mesmas e, finalmente, é indicada a escolha da abordagem que a equipa de conteúdos considera mais adequada.

3.1 Software Clínico

O aparecimento e implementação do software clínico constitui um avanço muito importante no que diz respeito ao panorama da saúde. O software clínico permite a informatização de todos os dados que outrora eram armazenados e processados sob a forma de papel. De facto, a utilização do software clínico contribui de forma fulcral para uma melhor prestação de cuidados de saúde aos pacientes.

Dentro do mercado de soluções de software clínico, existem soluções que são mais abrangentes, tendo em conta vários perfis de profissionais de saúde - médico, enfermeiro, psicólogo, dentista, entre outros - ou então soluções focadas em áreas clínicas mais específicas.

O ALERT® é um software clínico que abrange diversos perfis de profissionais de saúde e oferece soluções para várias áreas clínicas. Em Portugal existem outras empresas que, tal como a ALERT,

se focam no desenvolvimento de softwares clínicos e têm como finalidade auxiliar os profissionais de saúde nas suas tarefas diárias. Para o desenvolvimento do presente projeto, opta-se por fazer uma análise do mercado com base em soluções desenvolvidas por empresas portuguesas. Assim, de entre as várias empresas portuguesas existentes, são apresentadas e abordadas apenas três – (1) Glintt, (2) F3M e, por fim, (3) Medicineone – por se considerar serem as mais representativas neste setor e que, pelas suas características, mais se aproximam do contexto deste trabalho.

3.1.1 Glintt

A Glintt é uma empresa com mais de 20 anos de experiência em consultoria e serviços tecnológicos que tem como objetivo “[...] transformar as organizações e melhorar a vida das pessoas.” (Glintt, s.d. a). Esta empresa é responsável pelo desenvolvimento de várias soluções para hospitais, clínicas e farmácias, sendo que estas soluções se encontram implementadas tanto a nível nacional como internacional, nomeadamente, em Espanha, no Brasil e em Angola.

Uma dessas soluções é a Globalcare, uma solução integrada de gestão hospitalar e clínica (Glintt, s.d. b). O Globalcare contempla diversas áreas, entre as quais (Glintt, s.d. c):

1. Sistema de gestão hospitalar;
2. Clínica;
3. Meios complementares de diagnóstico e terapêutica (MCDT);
4. Farmacologia;
5. Interoperabilidade;
6. Paciente.

A primeira área, sistema de gestão hospitalar, inclui três módulos distintos – (1) gestão dos pacientes, (2) faturação e (3) honorário dos profissionais de saúde (Glintt, s.d. d).

Relativamente ao primeiro módulo, a gestão dos pacientes, o Globalcare implementa diversas funcionalidades que permitem a gestão do paciente desde o primeiro contacto com o mesmo, entre as quais se destacam as seguintes:

- Registo e monitorização do processo e fluxo do paciente;
- Acompanhamento do paciente em ambulatório¹⁰;
- Controlo de listas de espera de consultas, exames e cirurgias;
- Controlo do processo de validação e verificação de seguros;
- Controlo da entrada de sinistrados.

¹⁰ Situação clínica de um paciente em que não está acamado ou internado (S.A, 2021).

No que diz respeito ao segundo módulo, faturação, trata-se de uma “[...] ferramenta de gestão que apoia as unidades na adaptação à constante evolução do negócio da saúde.” (Glantt, s.d. d), que permite o processamento de documentos de carácter financeiro e de tesouraria.

Por fim, no que concerne ao terceiro módulo, honorário dos profissionais de saúde, este consiste numa ferramenta que permite o cálculo do valor de pagamento dos honorários dos profissionais de saúde (Glantt, s.d. d).

A segunda área do Globalcare, a área clínica, inclui oito módulos distintos – (1) *Electronic Health Record* (EHR), (2) *beside tracker*, (3) *clinical pathways*, (4) *eWhiteboard*, (5) dietética, (6) triagem, (7) pedidos de exames e, por fim, (8) *bedside nurse suite* (Glantt, s.d. e).

O módulo (1) EHR permite a gestão de toda a informação de carácter clínico do paciente, de forma a possibilitar um atendimento da parte do profissional de saúde mais centrado no paciente (Glantt, s.d. e). Relativamente ao módulo (2) *beside tracker*, este garante o rigor no que diz respeito aos pedidos de componentes, colheita de amostras e administração das respetivas unidades (Glantt, s.d. e). O módulo (3) *clinical pathways* é munido de sistemas de apoio à decisão e *guidelines* de atuação para cada caso e consiste num sistema de apoio aos profissionais de saúde que permite identificar pacientes em risco através da análise dos dados clínicos, sendo que estes começam a ser registados no processo de Triagem de Manchester¹¹ (Glantt, s.d. e). O módulo (4) *eWhiteboard* permite o acesso em tempo real à informação clínica do paciente, o que facilita o acesso e transmissão de informação, reduz o tempo de passagem de turno e auxilia a tomada de decisão dos profissionais de saúde (Glantt, s.d. e). O módulo (5) dietética “[...] permite responder às necessidades de gestão do circuito dos produtos alimentares dentro de uma unidade de saúde, garantindo uma eficiente comunicação entre serviços clínicos e o serviço de alimentação.” (Glantt, s.d. e). O módulo (6) triagem, auxilia a tomada de decisão no que diz respeito à atribuição da prioridade clínica com base no modelo de Triagem de Manchester (Glantt, s.d. e). O módulo (7) pedido de exames consiste num “[...] conjunto de funcionalidades que permitem agilizar e gerir todo o processo de pedidos de exames desde a criação de uma requisição até à disponibilização dos respetivos relatórios e resultados.” (Glantt, s.d. e). Por fim, o módulo (8) *bedside nurse suite* “[...] permite aos enfermeiros aceder, consultar e registar em tempo real toda a informação clínica do paciente.” (Glantt, s.d. e), evitando assim “[...] registos duplicados, perdas de informação e otimiza o tempo dos enfermeiros nas suas práticas clínicas.” (Glantt, s.d. e).

A terceira área do Globalcare, meios complementares de diagnóstico e terapêutica, inclui quatro módulos – (1) workflow dos MCDT, (2) SIBAS mobile, (3) anatomia patológica e, por fim, (4) nefrologia (Glantt, s.d. f).

O módulo (1) workflow dos MCDT, consiste numa solução que garante uma gestão centralizada do fluxo de exames, que permite o acompanhamento de todo o processo de exames (Glantt, s.d. f). O módulo (2) SIBAS mobile, consiste num serviço que “[...] permite ao dador de sangue o

¹¹ Triagem de Manchester consiste num sistema que “[...] utiliza um protocolo clínico que permite classificar a gravidade da situação de cada doente que recorre aos Serviço de Urgência.” (Hospital de Cascais, s.d.).

contacto direto, simples e intuitivo com o serviço de sangue.” (Glintt, s.d. f). O módulo (3) anatomia patológica, permite auxiliar a gestão deste tipo de serviços, incluindo “[...] os processos especificidades inerentes ao diagnóstico anatomopatológico ¹² nas áreas de histologia¹³, citologia¹⁴ e autópsia.” (Glintt, s.d. f). Finalmente, o módulo (4) nefrologia, tem como principal finalidade garantir “[...] o máximo conforto e bem-estar aos pacientes em hemodiálise e diálise peritoneal [...]” (Glintt, s.d. f).

A quarta área do Globalcare, farmacologia, apresenta funcionalidades que permitem garantir o “[...] rigor e segurança em todas as fases do circuito do medicamento.” através de gestão e rastreabilidade no decorrer das diversas fases do circuito do medicamento (Glintt, s.d. g).

A quinta área do Globalcare, interoperabilidade, consiste numa plataforma que permite estabelecer a interligação entre os diferentes sistemas e, conseqüentemente, a partilha de informação entre os mesmos (Glintt, s.d. h).

A sexta e última área, paciente, inclui dois módulos – (1) aplicação do paciente e (2) portal do paciente (Glintt, s.d. i).

A aplicação do paciente consiste numa aplicação para dispositivos móveis e o portal do paciente consiste num módulo que permite o acesso via portal web. Não obstante, são módulos bastante semelhantes, na medida em que, ambos permitem aos pacientes desempenhar diversas ações, entre as quais (Glintt, s.d. i):

- Marcar/remarcar/cancelar consultas;
- Consultar o histórico de visitas;
- Consultar informação de uma consulta;
- Aceder a resultados de exames.

3.1.2 F3M

A F3M Information Systems S.A, sediada em Braga e fundada em julho de 1987, é uma empresa especializada em Tecnologias da Informação e Comunicação (F3M, s.d a). A F3M dedica-se à conceção, produção e implementação de software para diversos mercados, nomeadamente, os mercados da economia social, óticas, indústria têxtil e unidades de saúde.

A F3M disponibiliza, através de licenciamento e SaaS, uma solução integrada para a gestão de unidades de saúde de diferentes dimensões e tipologias tais como unidades hospitalares, policlínicas, clínicas de fisioterapia, unidades de cuidados continuados, unidades de internamento, residências sénior, entre outras (F3M, s.d b). De acordo com a empresa F3M, a solução integrada para a gestão de unidades de saúde, permite “Gerir as diferentes áreas com

¹² Diagnósticos que resultam da análise de material humano colhido ao paciente (Hospital da Luz, s.d.).

¹³ O estudo dos tecidos biológicos (Faculdade de Ciências da Universidade de Lisboa, s.d.).

¹⁴ É o “[...] exame que avalia as características das células presentes numa determinada amostra.” (Unilabs, 2020).

um foco permanente no utente, garantindo a eficiência operacional e estratégica [...]” (F3M, s.d b), os quais são objetivos nucleares de qualquer Unidade de Saúde.

Neste âmbito, o software para a gestão de unidades de saúde da F3M “[...] dispõe de diversas soluções que se complementam entre si e que lhe permitem tirar o melhor partido da tecnologia.” (F3M, s.d. c), nomeadamente:

- Clínica – solução que habilita a gestão administrativa de uma clínica, permite a faturação a utentes e cálculo de honorários, contempla agendas, e permite, ainda, a gestão global do fluxo de trabalho (F3M, s.d. c);
- Internamento – solução que permite a gestão da ocupação de quartos e camas bem como, desenhar e configurar as áreas de internamento (F3M, s.d. d);
- Internamento Lar – solução que permite a gestão da ocupação de quartos e camas numa residência sénior, e, adicionalmente, desenhar e configurar as áreas de internamento (F3M, s.d. e);
- Unidades de cuidados continuados – solução que permite desenhar e configurar as áreas de cuidados continuados cumprindo as instruções definidas pela Rede Nacional de Cuidados Continuados Integrados (F3M, s.d. f);
- Prescrição eletrónica – solução que permite, com rigor, efetuar a prescrição eletrónica de medicamentos, meios complementares de diagnóstico e atestados médicos para carta de condução (F3M, s.d. g);
- Gestão hospitalar – solução que tem em consideração e promove o trabalho em equipa e a articulação entre os diferentes serviços (F3M, s.d. h);
- Processo clínico e social – solução que permite agregar, por áreas e com foco no utente, toda a informação clínica relacionada com os utentes, e, em simultâneo, centraliza os locais de registo dos profissionais de saúde (F3M, s.d. i);
- Bloco operatório – solução que agrega toda a informação e processos de trabalho no que concerne as fases pré-operatória, intraoperatória e pós-operatória (F3M, s.d. j);
- Farmácia hospitalar – solução desenhada para promover a simplificação da gestão de *stocks* da farmácia hospitalar (F3M, s.d. k);
- Processo integrado do utente – solução que permite desmaterializar e agregar toda a informação relacionada com o processo integrado do utente, e, paralelamente, permite simplificar e minimizar os tempos de registos de cada profissional (F3M, s.d. l);
- MpDS – solução para dispositivos móveis que permite agilizar o processo de captação de imagens clínicas de alta qualidade para diferentes contextos clínicos (F3M, s.d. m).

3.1.3 MedicineOne

A MedicineOne é uma empresa que se dedica ao desenvolvimento de software para o mercado da saúde, mais concretamente, soluções de gestão clínica. De origem portuguesa e presente em 4 continentes, dos vários clientes da MedicineOne, podem destacar-se grandes nomes nacionais, como o SNS, grupo Luz Saúde e o grupo CUF, e internacionais, nomeadamente, o Governo do Brasil e o Governo de Cabo Verde (MedicineOne, s.d. a).

A solução clínica desenvolvida pela MedicineOne denomina-se de M1 e é considerada pela empresa como “Uma única solução para as áreas clínica, administrativa e financeira.”. Neste sentido, a solução M1 divide-se em quatro segmentos – M1 Hospitals, M1 Clinics, M1 Occupational Health e M1Small Praticce (MedicineOne, s.d. b).

No que concerne o segmento M1 Hospitals, o M1 apresenta-se como uma solução de gestão presente em hospitais privados de grandes grupos de saúde privada a nível nacional. Assim, o M1 contempla ferramentas que permitem não só ter conhecimento da atividade dos profissionais do hospital, como também dos seus clientes (MedicineOne, s.d. b).

O segundo, e mais antigo, segmento M1 Clinics contribui para a otimização dos processos clínicos e dos processos administrativos, com o objetivo de reduzir custos e tempo de atividade e, ainda, auxiliar a tomada de decisões por parte dos profissionais de saúde (MedicineOne, s.d. b).

No que diz respeito ao terceiro segmento, M1 Occupational Health, este atua na área da saúde ocupacional, através de mecanismos adequados que agilizam a gestão de vários processos deste espectro da saúde, entre os quais, o exame médico e a produção do relatório anual de atividades, sem negligenciar a segurança, higiene e outras atividades relevantes (MedicineOne, s.d. b).

Por fim, o M1Small Praticce, atua ao nível dos pequenos consultórios, através do desenvolvimento de ferramentas nucleares que auxiliam a gestão autónoma e eficiente destas unidades de saúde e qualquer consultório (MedicineOne, s.d. b).

Além da divisão em segmentos, o M1 subdivide-se também em 12 módulos sobre os quais atua, nomeadamente (MedicineOne, s.d. b):

- Consulta;
- Enfermagem;
- Administrativo;
- Bloco Operatório;
- Internamento;
- Urgência;
- Saúde Ocupacional;
- Farmácia e Stocks;
- Fisioterapia;
- Alimentação;
- Gestão;
- Configuração.

3.1.4 Conclusão da análise do software clínico

Das soluções analisadas verificou-se que abordam a gestão de diversas áreas da saúde. No entanto, devido à falta de documentação, não foi possível analisar de que forma é abordada a temática da gestão de conteúdos clínicos.

3.2 Soluções de Gestão de Dados

Nesta secção, são apresentadas duas soluções consideradas relevantes para a gestão dos dados armazenados numa base de dados – (1) Microsoft Access e (2) LibreOffice Base – sendo que ambas apresentam funcionalidades adicionais para além daquelas que permitem a gestão de dados.

3.2.1 Microsoft Access

Segundo o livro *Access 2019 Bible* (Alexander & Kusleika, 2018), o Microsoft Access é uma aplicação de gestão de base de dados relacional que apresenta interface gráfica, desenvolvida pela Microsoft.

Os mesmos autores (Alexander & Kusleika, 2018), referem que a base de dados do Microsoft Access apresenta seis tipos de objetos, os quais são:

- *Table* – objeto que armazena os dados;
- *Query* – objeto que permite obter dados armazenados nas tabelas da base de dados;
- *Form* – objeto que permite adicionar e exibir registos da base de dados forma personalizável;
- *Report* – objeto que permite obter os dados armazenados sob o formato de ficheiro *Portable Document Format (PDF)*;
- *Macro* – objeto que permite automatizar tarefas sem a necessidade de programação;
- *Module* – objeto que permite automatizar tarefas através de instruções escritas na linguagem *Visual Basic for Applications (VBA)*.

Para além da criação de novas bases de dados, o mesmo livro (Alexander & Kusleika, 2018) refere que o Microsoft Access contempla a possibilidade de trabalhar com dados de diferentes fontes externas. Neste âmbito, o Microsoft Access apresenta três diferentes formas de trabalhar com dados de fontes externas, nomeadamente (Alexander & Kusleika, 2018):

- *Linking* (vincular);
- *Importing* (importar);
- *Exporting* (exportar).

No que diz respeito ao *linking* (vincular), este “[...] cria uma conexão com uma tabela de uma outra base de dados do Access ou vincula-se com dados de outros formatos.”¹⁵.

Relativamente ao *importing* (importar), o Microsoft Access importa “[...] uma cópia dos dados de uma fonte de dados, outra base de dados Access, ou ficheiro de outra aplicação de base de dados numa tabela Access.”¹⁶.

Por fim, no âmbito do *exporting* (exportar), o Microsoft Access permite exportar “[...] cópias dos dados de uma tabela do Access para um ficheiro de texto, outra base de dado Access ou ficheiro de outra aplicação.”¹⁷.

O Microsoft Access apresenta ferramentas que permitem trabalhar com dados de diferentes fontes. Entre as várias fontes, no livro *Access 2019 Bible* (Alexander & Kusleika, 2018), são nomeadas:

- Ficheiros de texto;
- Ficheiros Excel;
- Bases de dados ODBC (SQL Server, Sybase Server, Oracle Server, entre outras);
- Tabelas, listas e documentos no formato HTML;
- Ficheiros XML;
- Tabelas do Outlook;
- Ficheiros Microsoft Exchange;
- Listas SharePoint;
- Base de dados Azure;
- Ficheiros Word;
- Ficheiros Rich Text Format (RTF).

3.2.2 LibreOffice Base

De acordo com o *website* do LibreOffice, o LibreOffice Base é uma aplicação de base de dados pensada para ser utilizada pelos mais vastos perfis de utilizadores, incluindo uso empresarial, pertencente ao conjunto de aplicações LibreOffice (LibreOffice, s.d.).

O LibreOffice Base apresenta quatro objetos principais, utilizados para várias tarefas características de uma aplicação de base de dados, nomeadamente (Borges, 2014):

- Tabela – objeto que garante a persistência dos dados;
- Consulta – objeto que permite exibir os dados de uma mais tabelas;

¹⁵ Tradução livre do autor. No original “[...] creates a connection to a table in another Access database or links to the data from a different format.” (Alexander & Kusleika, 2018).

¹⁶ Tradução livre do autor. No original “[...] Importing copies data from a data source, another Access database, or another application’s database file into an Access table.” (Alexander & Kusleika, 2018).

¹⁷ Tradução livre do autor. No original “Exporting copies data from an Access table into a text file, another Access database, or another application’s file.”. (Alexander & Kusleika, 2018).

- Formulário – objeto que habilita a inserção de dados da forma pretendida nas tabelas desejadas e ao mesmo tempo, de uma forma mais *user friendly*;
- Relatório – objeto que permite apresentar os dados das tabelas de uma forma mais *user friendly*.

Esta aplicação fornece “[...] controladores nativos para alguns dos mais utilizados motores de base de dados [...]” (LibreOffice, s.d.), entre os quais:

- MySQL/MariaDB;
- Adabas D;
- MS Access;
- PostgreSQL.

Adicionalmente, o LibreOffice Base apresenta “[...] suporte interno aos controladores padrão JDBC e ODBC [...]” possibilitando, assim, a integração com qualquer outra base de dados.

3.3 Base de dados

No contexto deste projeto, é necessária uma base de dados para que os dados relativos aos conteúdos do ALERT® possam ser armazenados. Neste âmbito, estudaram-se duas alternativas que foram consideradas relevantes para o presente projeto. Contudo, antes de abordar as duas alternativas é necessário compreender o conceito de base de dados. Assim, de acordo com o *website* Oracle (Oracle, s.d. b), uma base de dados corresponde a uma coleção de informação estruturada que é armazenada eletronicamente num computador. O mesmo *website* acrescenta que a mesma é, normalmente, controlada por um sistema de gestão de base de dados [*Database Management System (DBMS)*] que permite definir, manipular, armazenar e obter dados de uma base de dados (IBM, 2014b).

Relativamente às diferentes abordagens que podem ser adotadas, no que diz respeito à persistência de dados, são apresentadas duas abordagens identificadas como relevantes – base de dados relacional (3.3.1) e base de dados não relacional (3.3.2) – e, por fim, feita uma comparação entre ambas, de forma a determinar qual representa a melhor abordagem no contexto deste projeto.

3.3.1 Base de Dados Relacional

No ano de 1970, Edgar F. Codd propôs o modelo relacional, modelo para sistemas de base de dados, que, na época, surgiu como alternativa aos modelos existentes – modelo hierárquico e modelo de rede (Codd, 1983).

No início dos sistemas de base de dados, cada aplicação armazenava os dados utilizando a sua própria estrutura única. Contudo, estas estruturas apresentavam vários problemas, entre os quais, a difícil manutenção e otimização, no que diz respeito ao desempenho da aplicação (Oracle, s.d. a).

O modelo relacional disponibiliza uma forma padrão de representar e consultar informação armazenada numa base de dados, que pode ser utilizado por qualquer aplicação, com o objetivo de resolver os problemas anteriormente mencionados (Oracle, s.d. a).

Segundo o livro *Understanding the New SQL: A Complete Guide* (Melton & Simon, 1993), e tal como se pode verificar na Figura 2, são identificados três componentes relativamente ao modelo relacional, os quais são:

- Relação (*relation*): é a unidade básica de dados, corresponde a uma tabela em termos de terminologia da linguagem SQL;
- Registo (*tuple/record*): registo de uma tabela, corresponde a uma linha em termos de terminologia da linguagem SQL;
- Atributo (*attribute*): é uma propriedade que define uma relação, corresponde a uma coluna em termos de terminologia da linguagem SQL.

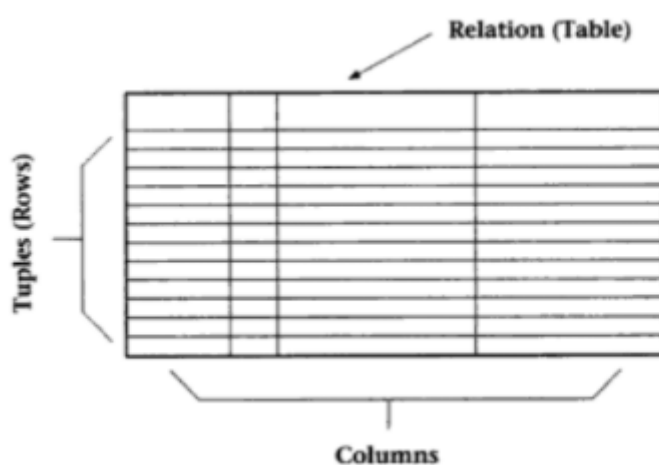


Figura 2 – Componentes do modelo relacional (Melton & Simon, 1993)

Do modelo relacional surgiu a base de dados relacional. Uma base de dados relacional “[...] é um tipo de base de dados que armazena e fornece acesso a pontos de dados relacionados entre si.”¹⁸. Uma base de dados relacional estrutura os dados em tabelas, as quais podem estar relacionadas entre si, através de dados que possam ter em comum, facto que permite melhor compreender a forma como os dados se relacionam entre si (IBM, 2020a).

¹⁸ Tradução livre do autor. No original “A relational database is a type of database that stores and provides access to data points that are related to one another.” (Oracle, s.d. a).

Uma base de dados relacional pode ser adotada “[...] em qualquer tipo de situações em que exista a necessidade de armazenar dados que apresentem relações entre si e que devam ser geridos de uma forma consistente, segura e baseada em regras.”¹⁹.

Os dados de uma base de dados relacional são organizados em tabelas, as quais, no contexto de uma base de dados relacional, são compostas por:

- linhas - cada linha representa um registo com um identificador único, a chave;
- colunas - cada coluna representa um atributo dos dados da respetiva tabela.

Para que seja possível armazenar, consultar, gerir e obter dados armazenados numa base de dados relacional, é necessário um software que estabeleça uma interface entre os utilizadores, a aplicação e a base de dados. Este software é denominado de sistema de gestão de bases de dados relacionais [*Relational Database Management System (RDBMS)*] (IBM, 2020b). Segundo o *American National Standards Institute (ANSI)*, o SQL é a linguagem padrão utilizada para interagir com bases de dados relacionais geridas por RDBMS (ANSI, 2018).

Atualmente, existem várias soluções no que diz respeito ao mercado de RDBMS. De acordo com o ranking de popularidade disponibilizado pelo *website DB-Engine* (DB-Engines, 2021), nas primeiras dez posições encontram-se os seguintes RDBMS:

1. *Oracle*;
2. *MySQL*;
3. *Microsoft SQL Server*;
4. *PostgreSQL*;
5. *IBM Db2*;
6. *SQLite*;
7. *Microsoft Access*;
8. *MariaDB*;
9. *Teradata*;
10. *Hive*.

As bases de dados relacionais apresentam características que motivam a sua escolha e, simultaneamente, características que as tornam numa opção menos adequada.

Tendo em consideração várias fontes - (Oracle, s.d. a), (IBM, 2020a), (IBM, 2020b) e (Melton & Simon, 1993) - nomeiam-se algumas das características que podem motivar a escolha deste tipo de base de dados, entre as quais se destacam:

¹⁹ Tradução livre do autor. No original “A relational database can be considered for any information need in which data points relate to each other and must be managed in a secure, rules-based, consistent way.” (Oracle, s.d. a).

- Propriedades ACID nas suas transações²⁰, as quais, de acordo com o livro *Transaction Processing: Concepts and Techniques* (Gray & Reuter, 1992), são:
 - Atomicidade (*atomicity*) – as alterações de dados são executadas sobre a forma de uma única operação. Isto significa que ou todas as alterações dos dados são executadas com sucesso ou então nenhuma é;
 - Consistência (*consistency*) – os dados apresentam-se num estado consistente no início e fim da transação;
 - Isolamento (*isolation*) – durante uma transação, todos os objetos dessa transação são tornados invisíveis para as restantes transações que possam estar a ocorrer ao mesmo tempo;
 - Durabilidade (*durability*) – depois da transação terminar com sucesso, as alterações dos dados são persistentes e públicas.
- Possibilidade de utilização de funcionalidades como os *stored procedures*. Estes, consistem em blocos de código, armazenados na base de dados, que, ao serem acedidos, realizam uma determinada tarefa. Isto permite que os desenvolvedores de código, que necessitam de realizar uma tarefa constantemente, não tenham de repetir o respetivo código da tarefa, podendo apenas invocar o *stored procedure*;
- Flexibilidade - advém do facto do SQL, como referido anteriormente, ser considerado a linguagem padrão para interagir com este tipo de base de dados (IBM, 2020a). O SQL incorpora em si uma linguagem, *Data Definition Language* (DDL), que consiste em comandos SQL que são usados para definir o esquema da base de dados (Melton & Simon, 1993), entre os quais:
 - *CREATE* - comando que permite criar diferentes objetos de base de dados, como por exemplo: esquemas, tabelas, vistas, sequências e índices (IBM, 2014);
 - *DROP* - comando que permite eliminar diferentes objetos de base de dados, como por exemplo: esquemas, tabelas, vistas, sequências e índices (IBM, 2014);
 - *ALTER* - comando que permite alterar diferentes objetos de base de dados, como por exemplo: esquemas, tabelas, vistas, sequências e índices (IBM, 2014).
- Desempenho – uma base de dados relacional é “[...] capaz de suportar os serviços centrados em dados mais exigentes do mundo, incluindo recursos modernos como técnicas de *caching* e *in-memory*.”²¹;
- Fiabilidade – as bases de dados relacionais apresentam uma “[...] longa história de suporte bem-sucedido aos maiores governos e empresas do mundo [...]”²²;
- Segurança - as bases de dados relacionais podem ser consideradas seguras na medida em que “[...] ao longo de 30 anos têm protegido os dados mais sensíveis do mundo.”²³.

²⁰ Transação no âmbito de bases de dados pode ser definido como o processo que resulta em qualquer tipo de alteração na base de dados (Gray & Reuter, 1992).

²¹ Tradução livre do autor. No original “*The performance of relational databases has been perfected to support the world’s most demanding data-centric services, including modern capabilities like caching and in-memory techniques.*” (IBM, 2020b).

²² Tradução livre do autor. No original “*With a long history of successfully supporting the world’s largest governments and businesses, the relational database has proven to be trustworthy and reliable.*” (IBM, 2020b).

²³ Tradução livre do autor. No original “*Today’s relational databases perform like security veterans, having matured over the 30 years that they have been protecting the world’s most sensitive data.*” (IBM, 2020b).

Relativamente às características que tornam este tipo de base de dados numa opção menos adequada, destacam-se as seguintes:

- Modelos de dados rígidos – estes modelos requerem uma preocupação adicional, no que diz respeito ao seu desenho, para que seja garantido o desempenho, de forma a suportar as expectativas iniciais relativas a este parâmetro. Adicionalmente, alterações a um esquema deste tipo de base de dados, podem resultar em períodos de inatividade da base de dados (IBM, 2020c);
- Em bases de dados relacionais, os dados têm de ser estruturados para que possam ser armazenados e organizados sob a forma de tabela. No entanto, existem dados que se podem revelar difíceis de converter em tabelas, podendo fazer com que a que a estrutura da base de dados seja desafiante de alcançar, complexa e os dados de difícil tratamento (Leavitt, 2010).

3.3.2 Base de Dados Não Relacional

Bases de dados não relacionais, atualmente também chamadas de NoSQL, existem desde os finais da década de 1960 (Leavitt, 2010). Segundo (Mohamed, et al., 2014), o termo NoSQL foi inicialmente introduzido por Carlo Strozzi, no ano de 1998, como designação de uma base de dados baseada em ficheiros que este estava a desenvolver. Desde aí que o termo NoSQL estava relacionado com bases de dados relacionais que omitiam o uso da linguagem SQL.

O mesmo autor afirma ainda que, mais tarde, em 2009, o termo NoSQL tornou-se rival das bases de dados relacionais, devido às necessidades de grandes empresas como a Amazon e a Google. Estas necessidades surgiram devido a problemas com as bases de dados relacionais, causados pelo elevado aumento de informação armazenada (Leavitt, 2010).

Este tipo de base de dados não organiza os dados através do modelo relacional. De facto, constata-se que este tipo de base de dados não aplica um esquema de dados fixo, uma vez que, “[...] usam modelos de armazenamento que são otimizados para os requisitos específicos dos tipos de dados que estão a ser armazenados.”²⁴.

Segundo o artigo *A middle layer solution to support ACID properties for NoSQL databases* (Lotfy, et al., 2016) as bases de dados não relacionais não apresentam o total suporte das propriedades do ACID. De facto, as bases de dados não relacionais aplicam as propriedades do princípio otimista - *Basically Available, Soft state, Eventual consistency (BASE)*. Assim, as propriedades do princípio BASE são:

- *Basically Available*: garante a disponibilidade dos dados e existe uma resposta a qualquer pedido. Contudo, há a possibilidade de os dados não estarem disponíveis durante um determinado período, no entanto, é necessário ocorrer algo grave para que todos os dados fiquem indisponíveis (Ganesh Chandra, 2015);

²⁴ Tradução livre do autor. No original “[...] non-relational databases use a storage model that is optimized for the specific requirements of the type of data being stored.” (Microsoft Azure, 2018).

- *Soft state*: os dados podem estar inconsistentes e o estado do sistema pode-se alterar ao longo do tempo (Ganesh Chandra, 2015);
- *Eventual consistency*: um sistema distribuído eventualmente tornar-se-á consistente. Nem todos os nós apresentam as mesmas cópias de dados num determinado momento.

Relativamente aos diferentes tipos de bases de dados não relacionais, destacam-se três tipos – (1) base de dados orientada a documentos (*document-based stores*), (2) base de dados orientada a colunas (*column-oriented databases*) e (3) base de dados orientada a chave-valor (*key-value stores*) (Leavitt, 2010). Estes tipos de base de dados resultam de uma diferenciação de cada base de dados relativamente aos respetivos modelos de dados.

Em desenvolvimento de software, normalmente, os dados são representados de forma intuitiva, podendo ser representados como um objeto ou representados através do formato *JavaScript Object Notation* (JSON).

1. No que diz respeito a bases de dados orientadas a documentos, estas armazenam os dados sob a forma de documentos idênticos a objetos do formato JSON. Estes documentos consistem num conjunto de campos do tipo texto (*string*) e respetivos valores (Microsoft Azure, 2018). Como representado na Figura 3, para cada documento existe uma chave única, através da qual se pode obter o respetivo documento e também se verifica que valores dos campos podem ser de diferentes tipos, desde tipos primitivos até a objetos de dados.

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

Figura 3 - Exemplo do armazenamento em bases de dados orientadas a documentos. Retirado de (Microsoft Azure, 2018).

Normalmente, cada documento representa os dados de uma entidade, sendo que estes dados, num contexto de base de dados relacional, poderiam estar dispersos em várias tabelas (Microsoft Azure, 2018). Assim, verifica-se que este tipo de base de dados não relacional apresenta uma grande flexibilidade, no sentido em que pode armazenar diferentes estruturas de dados em documentos. Esta flexibilidade, ao nível da estrutura

de dados, pode revelar-se benéfica, no sentido em que, permite a adaptação a diferentes estruturas de dados que podem surgir de mudanças na lógica de negócio (Microsoft Azure, 2018). Adicionalmente, este tipo de base de dados não relacional “[...] pode ser escalado horizontalmente de forma a conseguir armazenar grandes quantidade de dados.”²⁵.

Segundo o *web site DB-engines*, a base de dados não relacional, do tipo orientada a documentos, mais popular, de acordo com o ranking do *DB-engines*, é a MongoDB (DB-Engines, 2021).

2. Relativamente às bases de dados orientadas a colunas, estas caracterizam-se pela organização dos dados armazenados em linhas e colunas. Assim, o “[...] verdadeiro valor da base de dados orientada a colunas está na sua abordagem desnormalizada para estruturar dados dispersos, que decorre de uma abordagem orientada à coluna para armazenar dados.”²⁶.

Como referido anteriormente, neste tipo de base de dados não relacional, os dados são organizados em linhas e colunas, no entanto, as colunas são divididas em diferentes grupos denominados de famílias de coluna, em que cada uma apresenta um conjunto de colunas logicamente relacionadas (Microsoft Azure, 2018). Na Figura 4, é possível verificar que existem duas famílias de coluna – Identidade e Informação de contacto – em que uma entidade apresenta a mesma chave para cada uma das famílias de coluna. Assim, verifica-se que este tipo de base de dados é flexível, na medida em que, cada linha pode conter um conjunto de colunas que difere das restantes linhas (Microsoft Azure, 2018).

CustomerID	Column Family: Identity	CustomerID	Column Family: Contact Info
001	First name: Mu Bae Last name: Min	001	Phone number: 555-0100 Email: someone@example.com
002	First name: Francisco Last name: Vila Nova Suffix: Jr.	002	Email: vilanova@contoso.com
003	First name: Lena Last name: Adamczyk Title: Dr.	003	Phone number: 555-0120

Figura 4 – Exemplo do armazenamento em base de dados orientadas a colunas. Retirado de (Microsoft Azure, 2018).

Segundo o *web site DB-engines*, a base de dados não relacional, do tipo orientada a colunas mais popular, no que diz respeito ao ranking do *DB-engines*, é a Cassandra (DB-Engines, 2021).

²⁵ Tradução livre do autor. No original “They can horizontally scale-out to accommodate large data volumes.” (MongoDB, s.d. a).

²⁶ Tradução livre do autor. No original “The real power of a column-family database lies in its denormalized approach to structuring sparse data, which stems from the column-oriented approach to storing data.” (Microsoft Azure, 2018).

3. Por fim, no que diz respeito às bases de dados orientadas à chave-valor, estas consistem num “[...] sistema que armazena valores indexados para serem obtidos através de chaves.”²⁷. Este tipo de base de dados permite localizar e obter um determinado valor armazenado, de uma forma rápida e confiável, através da chave associada ao respetivo valor, o que torna este tipo de base de dados adequada para sistemas que tenham como prioridade obter dados constantemente (MongoDB, s.d. b). Este tipo de base de dados pode ser “[...] extremamente escalável, uma vez que este tipo de base de dados pode distribuir facilmente dados entre vários nós em diferentes máquinas.”²⁸

Segundo o *web site DB-engines*, a base de dados não relacional, do tipo orientada a chave-valor mais popular, no que diz respeito ao ranking do *DB-engines*, é a *Redis* (DB-Engines, 2021).

Da mesma forma que as bases de dados relacionais apresentam características que motivam a sua escolha e, simultaneamente, características que as tornam numa opção menos adequada, também as bases de dados não relacionais partilham dessas mesmas particularidades.

Neste sentido, algumas das características que motivam a escolha deste tipo de base de dados, são as seguintes:

- Flexibilidade - as bases de dados não relacionais são flexíveis, na medida em que, não apresentam um modelo de dados estático. Este facto permite armazenar dados com diferentes esquemas, sem ter de fazer alterações ao esquema de dados existente, e, assim, evitar períodos em que os dados se encontram indisponíveis (MongoDB, s.d. c);
- Capacidade de lidar com grande volume de dados em constante mudança, o que se deve ao facto de este tipo de base de dados ser facilmente escalável horizontalmente e, também, à sua flexibilidade (MongoDB, s.d. c);
- Ausência da necessidade de um desenho do modelo de dados muito pensado, na medida em que, é flexível quanto ao esquema dos dados, mostrando-se adequada para situações em que exista uma necessidade de desenvolvimento rápido/ágil (IBM, 2020d);
- Não implementa todos os princípios ACID, o que faz com que processe dados de forma mais rápida do que as bases de dados que implementam ACID (Leavitt, 2010).

Por outro lado, as bases de dados não relacionais possuem características que as podem tornar numa opção menos adequada. Destas características, destacam-se as seguintes:

- Dimensão - as bases de dados não relacionais podem revelar-se de maior dimensão relativamente a bases de dados relacionais, uma vez que, não evitam a duplicação dos dados (MongoDB, s.d. d);

²⁷ Tradução livre do autor. No original “[...] a key-value store is a system that stores values indexed for retrieval by keys.” (Leavitt, 2010).

²⁸ Tradução livre do autor. No original “A single key/value store can be extremely scalable, as the data store can easily distribute data across multiple nodes on separate machines.” (Microsoft Azure, 2018).

- Ausência de suporte de todas as propriedades do princípio ACID – isto pode traduzir-se em inconsistência de dados em determinados períodos (Leavitt, 2010);
- Ausência do uso de linguagem SQL – esta ausência significa que é necessária uma linguagem manual de programação para consulta de dados que, pode revelar-se desafiante, na medida em que, pode ser adequada para tarefas simples, mas complexa para as restantes (Leavitt, 2010).

3.3.3 Comparação entre as abordagens de base de dados

Tendo em consideração as duas abordagens identificadas como relevantes para a base de dados – base de dados relacional (3.3.1) e base de dados não relacional (3.3.2) - é necessário fazer uma análise comparativa de ambas, de forma a determinar qual a que melhor se adapta às características deste projeto.

Como referido anteriormente, o presente projeto está relacionado com a gestão de conteúdos utilizados por um software clínico, ALERT®. Esta é uma área em que a consistência e integridade dos dados são fatores críticos, sendo que erros nos dados podem causar consequências para as vidas dos pacientes. Desta forma, fatores como a consistência e integridade dos dados são fundamentais no momento da escolha do tipo de base de dados.

Na Tabela 2 apresenta-se um sumário comparativo das características dos dois tipos de base de dados considerados no âmbito deste projeto.

Tabela 2 – Base de dados relacional VS Base de dados não relacional

	Base de dados relacional	Base de dados não relacional
Transações	ACID	BASE
Modelo de dados	Rígido	Flexível
Modelo de armazenamento	Tabelas constituídas por colunas e linhas	Documentos, Chave-Valor, entre outros
Conhecimento na empresa	Existente	Não existente

Como referido nas secções 3.3.1 e 3.3.2, as bases de dados relacionais aplicam as propriedades do princípio ACID, garantindo, assim, a consistência e integridade dos dados a todo o momento. As bases de dados não relacionais, por sua vez, não apresentam todas as propriedades do princípio ACID. Este tipo de base de dados apresenta uma consistência eventual, isto é, num mesmo momento, podem existir diferentes cópias dos mesmos dados que, mais tarde, acabam por se tornar consistentes. Assim, tendo em conta a consistência eventual da base de dados não relacional, a escolha mais indicada, no âmbito deste projeto, é a base de dados relacional.

Adicionalmente, verifica-se que, na ALERT, não existe tanto conhecimento de tecnologias relacionadas com bases de dados não relacionais, como existe, no que diz respeito, a bases de dados relacionais. Este facto foi indicado pela equipa de conteúdos como fator condicionante no momento da escolha do tipo de base de dados, na medida em que, é necessário ter em

consideração a futura manutenção da mesma. Desta forma, tendo em conta o nível de conhecimento referente aos dois tipos de base de dados, a escolha recai sobre a base de dados relacional.

Por fim, e principalmente, a ALERT possui licença de base de dados relacional, nomeadamente a Oracle, o que representa um fator restritivo no momento da escolha da base de dados a utilizar, neste caso a base de dados relacional Oracle.

Assim, inevitavelmente, concluiu-se que a opção mais indicada, no contexto deste projeto, é uma base de dados relacional, nomeadamente a Oracle, como sistema de armazenamento dos dados relativos aos conteúdos do ALERT®.

3.4 Abordagens Arquiteturais

Nesta secção é feita uma apresentação de possíveis abordagens arquiteturais consideradas relevantes para a solução deste projeto. Neste âmbito, são analisadas abordagens relativamente à implementação da lógica de negócio, nomeadamente – (1) lógica de negócio implementada na base de dados e (2) lógica de negócio implementada na camada de aplicação.

Neste contexto, antes de prosseguir à análise de ambas as abordagens, é necessário entender em que consiste a lógica de negócio. Assim, de acordo com (Wang & Wang, 2006), a lógica de negócio pode ser vista como a parte do programa que contém a informação que determina como o negócio deve ser gerido. Esta lógica pode ser representada através de regras de negócio, sendo que estas regras podem ser vistas como unidades de conhecimento de gestão do negócio, que permitem garantir a integridade da lógica de um programa na perspetiva do negócio em causa.

3.4.1 Implementação da lógica de negócio na base de dados

Relativamente à primeira abordagem - implementação da lógica de negócio na base de dados - esta consiste na implementação da lógica de negócio diretamente na base de dados. Assim, nesta abordagem, a base de dados é mais do que uma camada de persistência de dados, uma vez que, é, também, um motor de processamento. De facto, a base de dados armazena, para além dos dados, funcionalidades que permitem geri-los e implementar as regras de negócio definidas. Desta forma, através destas funcionalidades, todos os dados armazenados na base de dados apenas podem ser geridos, acedidos e modificados, a partir da lógica de negócio implementada na base de dados (Oracle, s.d. c).

Estas funcionalidades, que tornam uma base de dados mais do que uma camada de persistência de dados, no contexto das bases de dados Oracle (base de dados escolhida para este projeto, como descrito na secção 3.3.3), são armazenadas diretamente na base de dados, sob a forma de um subprograma. A Oracle define um subprograma como um bloco de código escrito recorrendo à linguagem *Procedural Language/Structured Query Language* (PL/SQL) que é

armazenado diretamente na base de dados e pode ser invocado por uma aplicação (Oracle, s.d. c). Estes blocos de código, podem ser de dois tipos – *procedure* ou *function* – os quais são idênticos entre si, sendo que a principal diferença entre ambos é o facto das *functions* retornarem um valor único enquanto as *procedures* não retornam qualquer valor (Oracle, s.d. c).

A implementação da lógica de negócio na base de dados é uma abordagem que apresenta características que motivam a sua escolha e, simultaneamente, características que a tornam numa opção menos adequada.

Relativamente às características que motivam a sua escolha, destacam-se a seguintes:

- Flexibilidade – esta abordagem concede flexibilidade no que diz respeito às tecnologias adotadas para o desenvolvimento das aplicações, uma vez que, as aplicações em si não implementam a lógica de negócio. Esta lógica de negócio encontra-se armazenada diretamente na base de dados, o que possibilita uma maior flexibilidade no momento de migração para diferentes tecnologias (Oracle, s.d. c);
- Desempenho - permite que sejam feitas alterações à lógica de negócio, as quais estão logo disponíveis, uma vez que, a forma compilada de um subprograma se encontra logo disponível na base de dados, evitando a necessidade de compilar em tempo de execução (Oracle, s.d. c);
- Integridade - verifica-se que existem benefícios em termos de integridade e consistência dos dados armazenados. Como referido anteriormente, através da implementação da lógica de negócio, diretamente na base de dados, os dados são geridos – obtidos, inseridos e modificados – através da utilização de subprogramas armazenados na base de dados que implementam a lógica de negócio. Assim, concentrando a lógica de negócio na base de dados, é possível garantir que todas as aplicações que interagem com a mesma gerem os dados de forma correta, isto é, de acordo com as regras de negócio definidas na base de dados. Adicionalmente, esta abordagem permite que as aplicações que interagem com a base de dados se abstraiam das alterações da lógica de negócio (Oracle, s.d. c);
- Segurança – como referido anteriormente, ao implementar a lógica de negócio diretamente numa base de dados apenas se permite que os dados (da base de dados) sejam geridos a partir dos subprogramas armazenados na mesma, negando, assim, o acesso direto às tabelas (Oracle, s.d. c).

Relativamente às características que tornam esta abordagem arquitetural numa opção menos adequada, destaca-se:

- Dependência ao fornecedor da base de dados - tendo toda a lógica de negócio implementada na base de dados, fica-se dependente do fornecedor da mesma. Este facto, pode revelar-se uma razão para não adotar esta abordagem, especialmente, em projetos nos quais se procura independência relativamente ao fornecedor de base de dados.

3.4.2 Implementação da lógica de negócio na camada de aplicação

Relativamente à segunda abordagem - implementação da lógica de negócio na camada de aplicação – esta consiste na implementação da lógica de negócio diretamente numa aplicação. Esta aplicação é, assim, responsável pela implementação da lógica de negócio. Desta forma, pode estruturar-se uma aplicação em diferentes camadas de computação lógica e física (IBM, 2020e).

A arquitetura de três camadas (*three-tier architecture*) é a arquitetura de desenvolvimento de software predominante para aplicações do tipo cliente-servidor (IBM, 2020e). As três camadas desta arquitetura, como é possível verificar na Figura 5, são:

- Camada de apresentação;
- Camada de aplicação;
- Camada de dados;

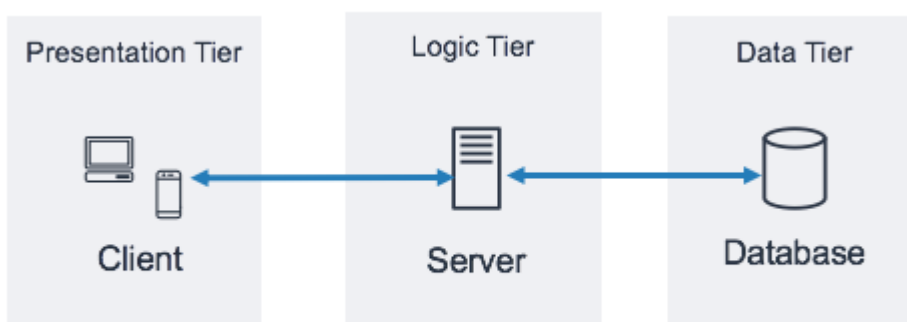


Figura 5 – Arquitetura de 3 camadas (*three-tier architecture*)

A camada de apresentação (*Presentation Tier*) é a camada que define como a aplicação se apresenta ao utilizador, na medida em que, lhe apresenta a interface gráfica (ou de outro tipo), mostra informação ao utilizador e obtém a informação fornecida pelo utilizador. Esta é a camada de nível superior que pode ser executada num navegador de internet (*web browser*), numa aplicação *desktop*²⁹, entre outros (IBM, 2020e). Embora se tenha definido a camada de apresentação, é importante mencionar que, caso esta seja a abordagem escolhida, a implementação desta camada é da responsabilidade de outro membro da equipa, como referido na secção 2.4, não fazendo parte do âmbito deste projeto.

A camada de aplicação (*Logic Tier*), é a camada com a responsabilidade de fazer o processamento da informação de acordo com a lógica de negócio. É nesta camada que é implementada a lógica de negócio, responsável por determinar o comportamento da aplicação face a uma ação de um utilizador. Esta camada é, também, responsável por comunicar com a camada de dados (*Data Tier*) de forma a adicionar, modificar e remover dados. Por fim, pode

²⁹ Uma aplicação desktop consiste num software que pode ser instalado num computador.

acrescentar-se que esta camada também é conhecida como camada de lógica ou camada intermediária (IBM, 2020e).

A camada de dados (*Data Tier*), é a camada onde os dados, anteriormente processados pela camada de aplicação, são armazenados e geridos. Esta camada também pode ser denominada de base de dados ou camada de acesso aos dados (IBM, 2020e).

Na arquitetura de três camadas, recomenda-se que a camada de apresentação e a camada de dados não comuniquem diretamente uma com a outra, sendo que todas as comunicações devem ser feitas através da camada de aplicação (IBM, 2020e).

A implementação da lógica de negócio na camada de aplicação significa a utilização do padrão de arquitetura em camadas. Assim, esta é uma abordagem que apresenta características que motivam a sua escolha e, simultaneamente, características que a tornam numa opção menos adequada.

Relativamente às características que motivam a sua escolha, destacam-se as seguintes:

- Separação lógica e física de funcionalidades – de acordo com o *website* da IBM (IBM, 2020e), a separação lógica e física de uma aplicação em diferentes camadas permite que cada camada possa ser executada num servidor dedicado, de forma que “[...] os serviços de cada camada possam ser customizados e otimizados sem influenciar as outras camadas.”³⁰;
- Escalabilidade – cada camada pode ser escalada de forma independente das restantes camadas (IBM, 2020e);
- Independência do fornecedor da base de dados – com a lógica de negócio implementada na camada de aplicação é alcançada uma independência relativamente ao fornecedor da base de dados, na medida em que, a base de dados apenas armazena os dados (Schussel, 2001) (Kambalyal, 2010);
- Segurança – com a implementação da lógica de negócio na camada de aplicação, a camada de apresentação, como referido anteriormente, não comunica diretamente com a camada de dados. Esta comunicação é feita através da camada de aplicação que funciona como uma *firewall* interna, protegendo, assim, a camada de dados de possíveis ataques maliciosos (IBM, 2020e).

Relativamente às características que tornam esta abordagem arquitetural numa opção menos adequada, destacam-se:

- Desempenho – o desempenho pode ser afetado, negativamente, devido à separação física e/ou lógica da camada de aplicação, onde é implementada a lógica de negócio, e a camada de dados (Kambalyal, 2010);

³⁰ Tradução livre do autor. No original “[...] so the services of each tier can be customized and optimized without impact the other tiers.” (IBM, 2020e).

- Complexidade – a separação em diferentes camadas resulta numa estrutura, eventualmente, mais complexa.

3.4.3 Comparação das abordagens arquiteturais

Tendo em consideração as duas abordagens identificadas como relevantes sobre o local da implementação da lógica de negócio – implementação da lógica de negócio na base de dados (3.4.1) e implementação da lógica de negócio na camada de aplicação (3.4.2) – é necessário fazer uma análise comparativa entre ambas, com a finalidade de determinar qual a que melhor se adapta às características e necessidades deste projeto.

Na Tabela 3 apresenta-se um sumário comparativo das características das abordagens arquiteturais consideradas no âmbito deste projeto.

Tabela 3 – Implementação da lógica de negócio na base de dados VS lógica de negócio na camada de aplicação

	Lógica de negócio implementada na base de dados	Lógica de negócio implementada na camada de aplicação
Disponibilidade de alterações	Imediatamente disponíveis	Disponíveis após compilação da aplicação
Transações entre camadas	Entre 2 camadas	Entre 3 camadas
Segurança dos dados	Sem acesso direto	Acesso fora do âmbito da base de dados
Separação de responsabilidades	Não existente	Existente (<i>Single Responsibility</i>)
Dependência do fornecedor de base de dados	Dependente	Independente

Neste âmbito, a equipa de conteúdos identificou que existe uma necessidade constante de criação e atualização de conteúdos, que pode resultar de diferentes origens (como descrito na secção 2.2.1). Sendo a lógica de negócio implementada diretamente na base de dados, sob a forma de subprogramas, permite-se que sejam feitas alterações à lógica de negócio e que estas estejam logo disponíveis. Estas alterações ficam imediatamente disponíveis, uma vez que, a forma compilada de um subprograma se encontra logo disponível na base de dados, o que evita a necessidade de compilar em tempo de execução. Por sua vez, quando a lógica de negócio é implementada na camada de aplicação, sempre que é feita uma alteração à lógica de negócio, é necessário compilar a aplicação, o que pode resultar em períodos de indisponibilidade. Assim, conclui-se que, sendo a lógica de negócio implementada diretamente na base de dados, podem ser feitas alterações à lógica de negócio, sem ter de esperar por períodos específicos para

compilar e tornar disponíveis as novas alterações, como se verifica quando a lógica de negócio é implementada na camada de aplicação.

Relativamente ao desempenho, ao implementar a lógica de negócio diretamente na base de dados existem menos transações entre camadas, uma vez que, esta lógica se encontra diretamente na base de dados. Ao implementar a lógica de negócio na camada de aplicação, a camada de apresentação e a camada de dados não podem comunicar diretamente uma com a outra, o que faz que todas as comunicações tenham de ser feitas através da camada de aplicação. Assim, é expectável que, perante este cenário, o desempenho seja superior ao implementar a lógica de negócio diretamente na base de dados, do que quando esta é implementada na camada de aplicação.

Relativamente à segurança dos dados, com a implementação da lógica diretamente na base de dados, apenas se permite que os dados da base de dados sejam geridos a partir dos subprogramas armazenados na mesma, negando, assim, o acesso direto às tabelas que armazenam os dados. Com a implementação da lógica de negócio na camada de aplicação, por sua vez, há um acesso aos dados fora do âmbito da base de dados, o que pode ser menos seguro.

Por outro lado, a implementação da lógica de negócio na camada de aplicação promove a separação de responsabilidades e o uso de padrões de software (por exemplo *Single Responsibility*). Isto é, a base de dados é apenas responsável por armazenar e gerir os dados e a camada de aplicação tem a responsabilidade de implementar as regras de negócio, que determinam o comportamento da aplicação, tendo em conta uma ação de um utilizador. Isto é um ponto a favor da implementação da lógica de negócio na camada de aplicação, na medida em que, permite que cada camada possa ser escalada de forma independente das restantes camadas, de acordo com as necessidades identificadas.

Por fim, e principalmente, a equipa de conteúdos desde o início do projeto, definiu como restrição de desenvolvimento a implementação da lógica de negócio diretamente na base de dados. Esta restrição resulta da forte cultura da ALERT, relativamente ao desenvolvimento da lógica de negócio diretamente na base de dados, através da linguagem PL/SQL. Por esta razão, a equipa de conteúdos considerou que este projeto deveria seguir a cultura de implementação da lógica de negócio diretamente na base de dados.

Assim, inevitavelmente, concluiu-se que a opção a ser adotada, no contexto deste projeto, é a implementação da lógica dos conteúdos do ALERT® diretamente na base de dados.

4 Tecnologias

Neste capítulo, são apresentadas as tecnologias relevantes para o desenvolvimento da solução deste projeto – (1) base de dados Oracle, (2) SQL e Oracle SQL, (3) PL/SQL e (4) PL/SQL Developer.

4.1 Base de Dados Oracle

No âmbito deste projeto, como referido na secção 3.3.3, é utilizada uma base de dados Oracle para armazenar toda a informação referente aos conteúdos do ALERT®. Neste projeto, em específico, está a ser utilizada a versão 19c.

A base de dados Oracle, é um RDBMS que implementa vários recursos/características orientadas a objetos, entre os quais (Kyte, et al., s.d.):

- Tipos definidos por um utilizador;
- Herança;
- Polimorfismo;

Um RDBMS que implementa recursos orientados a objetos pode ser chamado de sistema de gestão de base de dados objeto-relacional [*Object-Relational Database Management System (ORDBMS)*]. Assim, a base de dados Oracle consiste num ORDBMS, tendo “[...] passado de um modelo relacional para um modelo objeto-relacional, permitindo assim armazenar modelos de negócio complexos numa base de dados relacional.”³¹.

Uma característica de um RDBMS é a “[...] independência do armazenamento físico de dados da estrutura lógica dos dados.”³². Na base de dados Oracle, um utilizador da base de dados

³¹ Tradução livre do autor. No original “*Oracle Database has extended the relational model to an object-relational model, making it possible to store complex business models in a relational database.*” (Kyte, et al., s.d.).

³² Tradução livre do autor. No original “*One characteristic of an RDBMS is the independence of physical data storage from logical data structures.*” (Kyte, et al., s.d.).

possui apenas um esquema de base de dados (*database schema*), sendo que, este tanto pode ser uma coleção de estruturas lógicas de dados, como uma coleção de objetos de esquema (*schema objects*).

Na Figura 6, é possível verificar que o utilizador com o nome *HR* (*HR User*) possui um esquema com o mesmo nome do utilizador, *HR* (*HR Schema*). O esquema do utilizador *HR* contém objetos de esquema, nomeadamente, tabelas e índices.

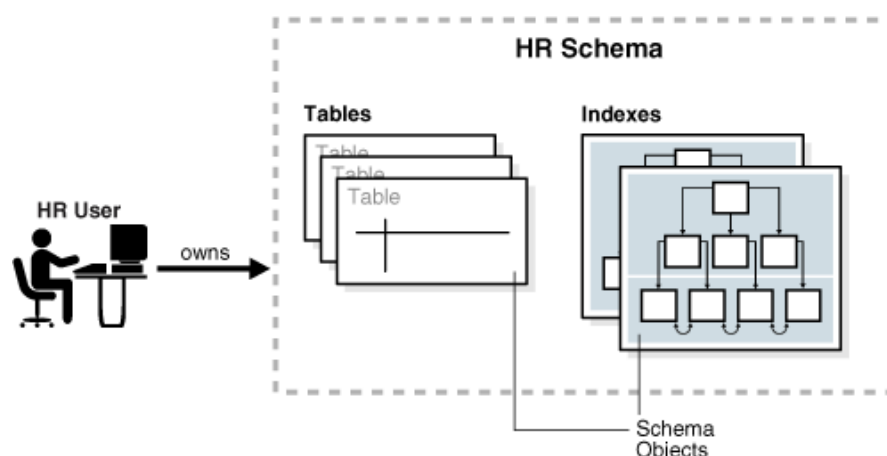


Figura 6 – Esquema de base de dados numa base de dados Oracle. Retirado de (Kyte, et al., s.d.).

Apesar da Figura 6 apenas fazer referência a dois tipos de objetos de esquema – tabelas (*tables*) e índices (*indexes*) - existem muitos mais. De acordo com o *website* da Oracle, (Kyte, et al., s.d.), os principais tipos de objetos de esquema são:

- Tabelas (*table*) - uma tabela é a unidade básica de organização dos dados na base de dados Oracle e descreve uma identidade (Kyte, et al., s.d.);
- Índices (*indexes*) - objeto de esquema que contém uma entrada de uma tabela ou grupo de tabelas e garante acesso rápido e direto às linhas das tabelas;
- Partições (*partitions*) - objeto de esquema que corresponde a uma parte de grandes tabelas e índices;
- Vistas (*views*) – pode-se pensar em vistas como *queries* armazenadas. São apresentações de dados customizáveis;
- Sequências (*sequences*) – é um objeto de esquema para gerar números inteiros, que é criado por um utilizador, mas pode ser partilhado por outros utilizadores;
- Dimensões (*dimensions*) – objeto de esquema que define uma relação do tipo pai-filho entre pares de conjuntos de colunas, usado normalmente para categorizar dados;
- Sinónimos (*synonyms*) – corresponde a um apelido para um objeto de outro esquema;
- Subprogramas e *packages* PL/SQL (*PL/SQL subprograms and packages*) – um subprograma PL/SQL é um bloco de código escrito em PL/SQL. Por sua vez, um *package* PL/SQL agrupa logicamente variáveis e subprogramas relacionados.

A base de dados Oracle é uma base de dados que é pensada para ser utilizada por vários utilizadores em simultâneo. Assim, esta base de dados, tem de ter em consideração a possibilidade da existência de várias transações de diferentes utilizadores, sob o mesmo conjunto de dados (Kyte, et al., s.d.). Desta forma, a base de dados Oracle apresenta as seguintes características (Kyte, et al., s.d.):

- Concorrência dos dados – garante que diferentes utilizadores possam aceder simultaneamente aos dados;
- Consistência dos dados – garante que cada utilizador tem uma visão consistente dos dados. Isto inclui tanto modificações com origem nas transações do utilizador como transações de outros utilizadores.

Relativamente à concorrência dos dados, a base de dados Oracle, implementa mecanismos de bloqueio (*lock*), mecanismos estes “[...] que impedem a ocorrência de interações destrutivas entre diferentes transações que acedem ao mesmo recurso compartilhado.”³³. Este mecanismo “[...] ajuda a assegurar a integridade dos dados enquanto permite o máximo número de acessos concorrentes aos dados.”³⁴.

4.2 SQL e Oracle SQL

Como referido na secção 3.3.1, o SQL é a linguagem padrão utilizada para interagir com bases de dados relacionais geridas por RDBMS. A base de dados Oracle, sendo um RDBMS, define a linguagem declarativa SQL como a interface da base de dados. Assim sendo, todas as operações efetuadas ao nível dos dados da base de dados Oracle são feitas através de instruções SQL (Kyte, et al., s.d.).

Através das instruções SQL é possível realizar as seguintes tarefas (Kyte, et al., s.d.):

- Criar, alterar, substituir e eliminar objetos de base de dados;
- Inserir, atualizar e eliminar linhas de colunas;
- Consultar dados;
- Controlar o acesso a uma base de dados e respetivos objetos;
- Garantir a consistência e integridade dos dados da base de dados.

³³ Tradução livre do autor. No original “A lock is a mechanism that prevents destructive interaction between transactions accessing a shared resource.” (Kyte, et al., s.d.).

³⁴ Tradução livre do autor. No original “Locks help ensure data integrity while allowing maximum concurrent access to data.” (Kyte, et al., s.d.).

A base de dados Oracle, utiliza a sua própria versão de SQL, denominada de Oracle SQL que “[...] inclui muitas extensões para a linguagem SQL padrão ANSI/ISO e as ferramentas e aplicações da base de dados Oracle fornecem instruções adicionais.”³⁵.

As instruções Oracle SQL podem dividir-se em seis categorias, nomeadamente (Kyte, et al., s.d.):

1. Instruções de linguagem de definição de dados [*Data Definition Language (DDL Statements)*];
2. Instruções de linguagem de manipulação de dados [*Data Manipulation Language (DML Statements)*];
3. Instruções de controlo de transações (*Transaction Control Statements*);
4. Instruções de controlo de sessão (*Session Control Statements*);
5. Instruções de controlo de sistema (*System Control Statement*);
6. Instruções SQL incorporadas (*Embedded SQL Statements*).

Relativamente à categoria das instruções DDL, estas permitem “[...] definir, modificar estruturalmente e eliminar objetos de esquema.”³⁶. De acordo com o *website* da Oracle (Kyte, et al., s.d.) destacam-se as seguintes instruções:

- CREATE – permite a criação de objetos de esquema, de estrutura de base de dados, da base de dados e utilizadores da base de dados;
- ALTER – permite a alteração de objetos de esquema, de estrutura de base de dados, da base de dados e utilizadores da base de dados;
- DROP – permite a eliminação de objetos de esquema, de estrutura de base de dados, da base de dados e utilizadores da base de dados;
- TRUNCATE – permite eliminar todos os dados dos objetos de esquema mantendo intacta a estrutura dos mesmos;
- GRANT – permite conceder privilégios e funções (*role*);
- REVOKE – permite revogar privilégios e funções;
- AUDIT – permite habilitar a opção de auditoria da base de dados;
- NOAUDIT – permite desabilitar a opção de auditoria da base de dados;
- COMMENT – permite adicionar um comentário.

No que diz respeito à categoria das instruções DML, estas permitem a “[...] consulta e manipulação dos dados em objetos de esquema existentes.”³⁷. O *website* da Oracle (Kyte, et al., s.d.) destaca as seguintes instruções:

- SELECT – permite a obtenção de dados armazenados em tabelas ou vistas;

³⁵ Tradução livre do autor. No original “*Oracle SQL includes many extensions to the ANSI/ISO standard SQL language, and Oracle Database tools and applications provide additional statements.*” (Kyte, et al., s.d.).

³⁶ Tradução livre do autor. No original “*Data definition language (DDL) statements define, structurally change, and drop schema objects.*” (Kyte, et al., s.d.).

³⁷ Tradução livre do autor. No original “*Data manipulation language (DML) statements query or manipulate data in existing schema objects.*” (Kyte, et al., s.d.).

- INSERT – permite inserir dados sob a forma de linhas numa tabela;
- UPDATE – permite alterar valores de determinadas colunas em linhas existentes numa tabela;
- MERGE – permite atualizar ou inserir linhas em tabelas ou vistas de forma condicional;
- DELETE – permite remover linhas de tabelas ou vistas;
- EXPLAIN PLAN – permite visualizar o plano de execução³⁸ (*execution plan*) de uma instrução SQL;
- LOCK TABLE – permite bloquear temporariamente uma tabela de forma a limitar o acesso à respetiva tabela.

Relativamente à categoria das instruções de controlo de transações, estas “[...] gerem as alterações feitas pelas instruções DML e agrupam instruções DML em transações.”³⁹. O *website* da Oracle (Kyte, et al., s.d.) destaca as seguintes instruções:

- COMMIT – permite tornar as alterações de uma transação permanentes;
- SAVEPOINT – permite definir um ponto para o qual é possível voltar;
- ROLLBACK – permite desfazer as alterações de uma transação desde o início da respetiva transação;
- ROLLBACK TO SAVEPOINT – permite desfazer as alterações de uma transação desde um determinado *savepoint*;
- SET TRANSACTION – permite estabelecer as propriedades de uma transação.

Relativamente à categoria das instruções de controlo de sessão, estas “[...] gerem dinamicamente as propriedades da sessão de um utilizador.”⁴⁰. O *website* da Oracle (Kyte, et al., s.d.) destaca duas instruções relativamente a esta categoria, as quais são:

- ALTER SESSION – permite alterar a sessão atual;
- SET ROLE – permite habilitar e desabilitar funções da sessão atual.

A categoria das instruções de controlo de sistema, apenas conta com uma instrução – ALTER SYSTEM – que permite modificar propriedades da base de dados instanciada.

Por fim, a categoria das instruções SQL incorporadas “[...] incorporam instruções DDL, instruções DML e instruções de controlo de transação num programa escrito através de

³⁸ “O plano de execução consiste na combinação de passos utilizados pela base de dados para a execução de uma instrução SQL” (Kyte, et al., s.d.).

³⁹ Tradução livre do autor. No original “Transaction control statements manage the changes made by DML statements and group DML statements into transactions.” (Kyte, et al., s.d.).

⁴⁰ Tradução livre do autor. No original “Session control statements dynamically manage the properties of a user session.” (Kyte, et al., s.d.).

linguagem procedimental.”⁴¹. O *website* da Oracle (Kyte, et al., s.d.) destaca as seguintes instruções:

- DECLARE, OPEN, CLOSE – permite definir, alocar e lançar um cursor;
- DECALRE DATABASE, CONNECT – permite determinar uma base de dados e conectar à mesma;
- DECLARE STATEMENT – permite atribuir nomes a variáveis;
- DESCRIBE – permite iniciar descritores (*descriptors*);
- WHENEVER – permite determinar como as condições de erro e aviso são tratadas;
- PREPARE, EXECUTE, EXECUTE IMMEDIATE – permite analisar e executar instruções SQL;
- FETCH – permite obter dados de um base de dados.

4.3 PL/SQL

O PL/SQL, no âmbito deste projeto, é utilizado como a linguagem procedimental para a implementação da lógica de negócio diretamente na base de dados Oracle.

O PL/SQL “[...] é uma linguagem de programação que fornece extensões procedimentais à versão SQL do Oracle [...]”⁴². Através da utilização do PL/SQL é possível o controlo do fluxo de um subprograma da base de dados usando variáveis e constantes, manipulação de erros, lógica de controle e lógica condicional (Oracle, s.d. e).

Na Figura 7, é apresentada uma aplicação a invocar um *procedure* que se encontra diretamente armazenado numa base de dados Oracle. A invocação de subprogramas – *procedure* e *function* – armazenados na base de dados Oracle pode ser feita através de diversos clientes de base de dados, entre os quais: Pro*C, JDBC, ODBC e OCI (Oracle, s.d. c).

⁴¹ Tradução livre do autor. No original “*Embedded SQL statements incorporate DDL, DML, and transaction control statements within a procedural language program.*” (Kyte, et al., s.d.).

⁴² Tradução livre do autor. No original “[...] *PL/SQL – a programing language that provides procedural extensions to Oracle’s version of SQL [...]*” (Feuerstein & Pribyl, 2005).

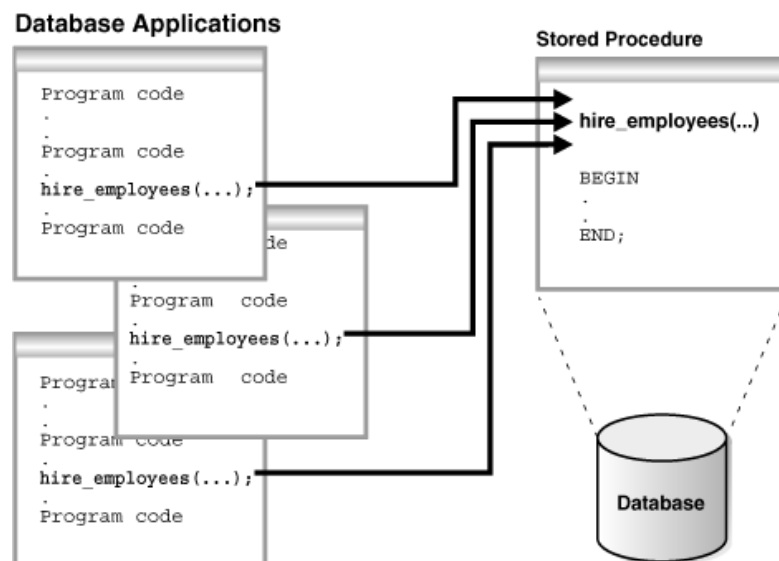


Figura 7 - Aplicação a invocar um *procedure* armazenado numa base de dados Oracle. Retirado de (Oracle, s.d. c).

Na linguagem PL/SQL a unidade mais pequena de código é chamada de bloco. De acordo com o livro *Oracle PL/SQL programming* (Feuerstein & Pribyl, 2014), o bloco na linguagem PL/SQL é constituído por quatro secções, as quais:

- Cabeçalho (*header*) – utilizada apenas em blocos com nome. Esta secção, “[...] determina a forma como um bloco com nome ou programa é invocado.”⁴³. Secção opcional;
- Secção de declaração (*declaration section*) – nesta secção são identificadas as “[...] variáveis, cursores, e sub-blocos que são referenciados nas secções de execução e exceção.”⁴⁴. Secção opcional;
- Secção de execução (*execution section*) – esta secção contém “[...] instruções que o mecanismo de execução PL/SQL executa no tempo de execução.”⁴⁵. Secção obrigatória;
- Secção de exceção (*exception section*) – secção que lida com as exceções. Secção opcional.

Na Figura 8 é apresentada a estrutura de um bloco PL/SQL.

⁴³ Tradução livre do autor. No original “The header determines the way the named block or program must be called.” (Feuerstein & Pribyl, 2014).

⁴⁴ Tradução livre do autor. No original “Identifies variables, cursors, and subblocks that are referenced in the execution and exception sections.” (Feuerstein & Pribyl, 2014).

⁴⁵ Tradução livre do autor. No original “Contains statements the PL/SQL runtime engine will execute at runtime.” (Feuerstein & Pribyl, 2014).

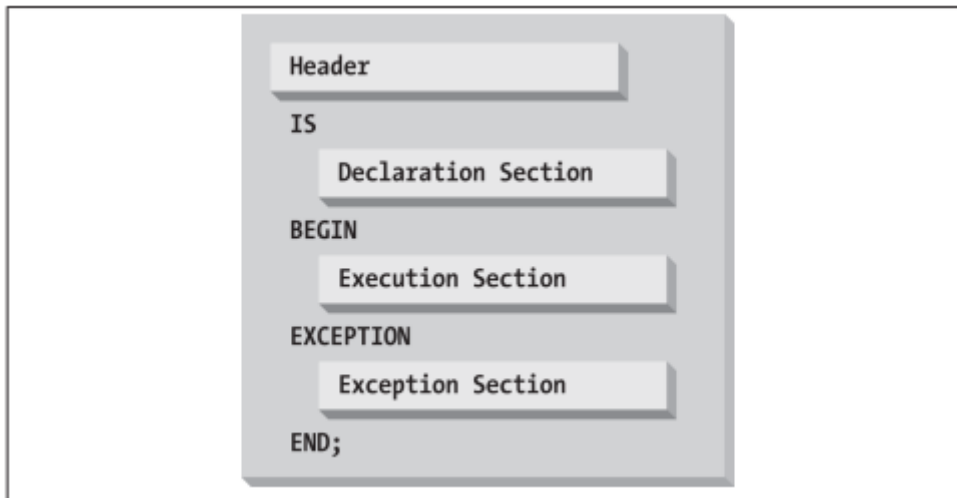


Figura 8 – Estrutura de um bloco PL/SQL. Retirado de (Feuerstein & Pribyl, 2014).

De acordo com o livro *Oracle PL/SQL programming* (Feuerstein & Pribyl, 2014), o PL/SQL contempla a criação de dois tipos de blocos:

- Blocos anónimos (*anonymous blocks*);
- Blocos com nome (*named blocks*).

Blocos anónimos, são blocos de código que não têm nome. Estes blocos tanto podem começar pela secção de declaração, como pela secção de execução, não apresentando a secção de cabeçalho (Feuerstein & Pribyl, 2014). Na Figura 9 é apresentado um bloco anónimo, escrito em PL/SQL, que apenas contempla a secção de execução.

```

BEGIN
  DBMS_OUTPUT.PUT_LINE ('Hello world');
END;

```

• Execution Only

Figura 9 – Exemplo de bloco PL/SQL anónimo. Retirado de (Feuerstein & Pribyl, 2014).

Relativamente aos blocos com nome, estes apresentam, obrigatoriamente, as secções de cabeçalho e execução. Na Figura 10 é apresentado um bloco com nome escrito em PL/SQL, com as quatro secções definidas.

```

PROCEDURE get_happy (ename_in IN VARCHAR2) • Header
IS
  l_hiredate DATE; • Declaration
BEGIN
  l_hiredate := SYSDATE - 2;
  INSERT INTO employee • Execution
    (emp_name, hiredate)
  VALUES (ename_in, l_hiredate);
EXCEPTION
  WHEN DUP_VAL_IN_INDEX • Exception
  THEN
    DBMS_OUTPUT.PUT_LINE
      ('Cannot insert.');
```

Figura 10 – Exemplo de bloco PL/SQL com nome. Retirado de (Feuerstein & Pribyl, 2014).

4.4 PL/SQL Developer

Como referido na secção 3.4.3, decidiu-se, no âmbito deste projeto, implementar a lógica de negócio diretamente na base de dados Oracle. “O PL/SQL Developer é um ambiente de desenvolvimento integrado (IDE) que é pensado especificamente para o desenvolvimento de unidades de programa armazenados em bases de dados Oracle”⁴⁶.

Existem outras ferramentas que permitem o desenvolvimento de código PL/SQL diretamente em bases de dados Oracle, entre as quais:

- Oracle SQL Developer;
- Toad for Oracle;
- dbForge Studio for Oracle.

Apesar de existirem diversos IDE, que permitem o desenvolvimento de código PL/SQL diretamente numa base de dados Oracle, no decorrer deste projeto foi utilizado o PL/SQL Developer. Esta decisão fica a dever-se ao facto de a ALERT possuir licenças para este IDE, que, consequentemente, é utilizado na empresa.

⁴⁶ Tradução livre do autor. No original “PL/SQL Developer is an Integrated Development Environment that is specifically targeted at the development of stored program units for Oracle Databases.” (Automations, s.d.).

5 Análise de Valor

Lawrence D. Miles, no seu livro “*Techniques of Value Analysis and Engineering*”, definiu a análise de valor (Miles, 2015) como “[...] análise de valor é um sistema, um conjunto completo de técnicas, devidamente organizadas, com o único propósito de identificar com eficiência custos desnecessários [...]”.⁴⁷

O processo de análise de valor é usado com a finalidade de obter um produto ou serviço com melhor qualidade e que satisfaça as necessidades de um cliente, através da identificação e remoção de recursos que não agreguem valor real ao produto ou serviço, mas que representem um custo elevado para o processo de desenvolvimento do mesmo.

Neste capítulo é identificada a oportunidade que deu azo a esta dissertação, feita uma análise à oportunidade, identificada a proposta de valor do projeto, apresentado o *Quality function deployment* (QFD), e, por fim, é aplicado o método de análise hierárquica, *Analytic Hierarchy Process* (AHP).

Embora este projeto seja desenvolvido no âmbito do *back-end* do *Content Server* (secção 2.4), nas secções 5.2 e 5.3, é feita uma análise tendo em conta toda a plataforma, o que inclui, também, o *front-end*.

5.1 Identificação da oportunidade

Uma oportunidade, no âmbito da análise de valor, representa a distância entre a situação atual e o futuro imaginado, de forma a identificar vantagens competitivas, resolver determinados problemas ou responder a uma ameaça (Belliveau, et al., 204).

No âmbito do projeto desta dissertação, a equipa de conteúdos da ALERT, identificou um conjunto de problemas. De facto, a equipa de conteúdos da ALERT, enfrenta algumas

⁴⁷ Tradução livre do autor. No original “[...] value analysis is a system, a complete set of techniques, properly arranged, for the sole purpose of efficiently identifying unnecessary cost [...]”.

dificuldades no que diz respeito aos processos de gestão de conteúdos, uma vez que se trata de processos que a equipa identifica como complexos, demorados e sujeito a erros. Esta problemática, abordada detalhadamente na secção 2.3, afeta não só a própria equipa de conteúdos, como também o cliente do ALERT®, uma vez que atrasa a disponibilização de novos conteúdos e atualizações a conteúdos existentes. Estes problemas constituem uma oportunidade para melhorar o processo de gestão de conteúdos. A identificação desta oportunidade surgiu de uma análise qualitativa, sob a forma de entrevista, aos membros da equipa de conteúdos. Pretende-se atingir esta melhoria através do desenvolvimento do *back-end*, a ser consumido pelo *Content Server*, para que este auxilie e facilite o trabalho da equipa de conteúdos, no que diz respeito à gestão de conteúdos.

5.2 Análise da Oportunidade

A oportunidade identificada na secção 5.1 necessita de ser analisada, de modo a garantir que vale a pena investir na mesma. Neste âmbito, na presente secção, é feito o enquadramento estratégico da oportunidade e a avaliação do segmento do mercado.

5.2.1 Enquadramento Estratégico da Oportunidade

A análise SWOT permite fazer uma avaliação das forças, fraquezas, oportunidades e ameaças da oportunidade. Assim, na Tabela 4, através da análise SWOT, é feito o enquadramento estratégico. Esta análise pode dividir-se em dois ambientes distintos:

- Ambiente externo - componente em que são identificadas as oportunidades e ameaças da oportunidade;
- Ambiente interno – componente em que são identificadas as forças e fraquezas da oportunidade.

Neste contexto, as forças correspondem às razões pelas quais a oportunidade agrega valor, as fraquezas representam os pontos negativos relativos à oportunidade, as oportunidades representam os fatores externos que podem representar hipóteses de sucesso e as ameaças são os fatores externos que podem prejudicar o sucesso da oportunidade.

Tabela 4 – Análise SWOT

Fatores Internos	<p style="text-align: center;">Forças</p> <ul style="list-style-type: none"> • Permite a criação, alteração e visualização dos conteúdos de uma forma mais prática, simples e segura; • Reduz a complexidade do processo de gestão de conteúdos; 	<p style="text-align: center;">Fraquezas</p> <ul style="list-style-type: none"> • Custos associados ao desenvolvimento.
-------------------------	---	---

	<ul style="list-style-type: none"> • Elimina a necessidade de conhecimento da linguagem SQL; • Permite a disponibilização dos conteúdos de forma mais rápida para os clientes; 	
Fatores Externos	<p style="text-align: center;">Oportunidades</p> <ul style="list-style-type: none"> • Facilidade de manutenção e ajustamento a novas regras de negócio; • Facilidade de contratação de novos membros para equipa de conteúdos. 	<p style="text-align: center;">Ameaças</p> <ul style="list-style-type: none"> • Corte do orçamento na área da saúde.

Na Tabela 4, são identificadas as forças, fraquezas, oportunidades e ameaças, da oportunidade identificada.

Relativamente às forças, foram identificadas quatro razões favoráveis à implementação da oportunidade – (1) a criação, alteração e visualização dos conteúdos de uma forma mais prática, simples e segura, (2) a redução da complexidade do processo de gestão de conteúdos, descrito nas secções 2.2.1 e 2.2.2, (3) a disponibilização dos conteúdos de forma mais rápida para os clientes e (4) a eliminação da necessidade de conhecimento da linguagem SQL.

A simplificação do processo de gestão de conteúdos, referida no ponto (1), constitui uma força, uma vez que, atualmente, a equipa de conteúdos enfrenta um processo complexo, como descrito nas secções 2.2.1 e 2.2.2, em que necessita de várias ferramentas e tem de aceder diretamente à base de dados para completar as operações desejadas. Para além disso, a oportunidade permite tornar o processo mais seguro e mais resistente a erros, uma vez que, se pretende que o *back-end* seja desenvolvido de forma a garantir que as regras de negócio de cada tipo de conteúdo são respeitadas.

Também a redução da complexidade do processo de gestão de conteúdos, referida no ponto (2), constitui uma força, uma vez que, exclui a ação da equipa de versionamento no que diz respeito à garantia da rastreabilidade dos registos das diferentes tabelas dos conteúdos. Assim, através da oportunidade, pretende-se que a rastreabilidade seja garantida de forma automática pelo *back-end*, o que permite que a equipa de conteúdos não dependa da equipa de versionamento e que a disponibilização dos conteúdos para os clientes do ALERT® seja mais rápida, o que também constitui uma força, apresentada no ponto (3).

Por fim, destaca-se uma outra força, a eliminação da necessidade de conhecimento da linguagem SQL (4). Esta eliminação é conseguida através do *back-end*, que permite à equipa gerir os conteúdos de forma abstraída do SQL.

Relativamente às fraquezas, apenas foi identificado o custo do desenvolvimento da oportunidade. De facto, a implementação do *back-end* implica custos relevantes no que diz respeito à análise dos conteúdos existentes, respetivas configurações e regras de negócio de cada um deles. Adicionalmente, também existem custos no que diz respeito ao desenvolvimento do *back-end* e aos testes do mesmo. Assim, visto que os custos podem ser relevantes e não compensar o valor acrescentado, é necessário tê-los em consideração.

No que diz respeito às oportunidades, foram identificadas a facilidade de manutenção, ajustamento a novas regras de negócio e a facilidade de contratação de novos membros para equipa de conteúdos.

Relativamente à facilidade de manutenção e ajustamento a novas regras do negócio, com esta oportunidade, pretende-se que o *back-end* seja desenhado e implementado com a finalidade de permitir uma fácil manutenção e adaptação a novas regras do negócio, de forma a que a equipa se abstraia de alterações que possam ocorrer devido a fatores externos.

Atualmente a equipa de conteúdos da ALERT, está ligada à área da saúde, contudo, espera-se que os membros da equipa tenham conhecimento em SQL, o que dificulta o processo de contratação. Assim, com a eliminação da necessidade de conhecimento SQL, pretende-se que a contratação de membros para a equipa, se torne mais fácil, uma vez que, os conhecimentos em SQL não fazem parte do contexto profissional de um indivíduo com formação na área da saúde.

Relativamente às ameaças, apenas foi identificada a possibilidade de haver cortes no orçamento da área de saúde. De facto, a existência de cortes orçamentais pode traduzir-se em perda de valor para a oportunidade, visto que, devido a este fator externo, a oportunidade pode deixar de ser vista como uma prioridade de investimento.

5.2.2 Avaliação do Segmento do Mercado

O mercado em que a oportunidade se enquadra, é o da saúde, um setor em constante mudança, em que a integridade da informação é crítica, uma vez que, a falta de rigor e disciplina no tratamento dos dados, pode colocar vidas em risco. Neste contexto, ao desenvolver o software clínico ALERT®, a ALERT tem de ter em consideração as contantes mutações no contexto da área da saúde, para conseguir dar resposta às contínuas alterações, de uma forma íntegra e rigorosa. Um dos exemplos concretos da volatilidade do setor da saúde, é o surgimento da pandemia COVID-19⁴⁸. Com a aparecimento deste vírus, o ALERT® foi forçado a adaptar-se às novas necessidades do mercado, nomeadamente, através da introdução de conteúdos relacionados com o novo vírus, como por exemplo, análises que determinam se um paciente testa negativo ou positivo para o COVID-19.

⁴⁸ COVID-19 é uma doença causada por um novo coronavírus designado por SARS-CoV-2 (World Health Organization: WHO, 2020).

Relativamente à oportunidade em concreto, pode considerar-se que se enquadra num segmento do tipo *business-to-customer*. Neste âmbito, a oportunidade em causa, visa permitir que a gestão de conteúdo, por parte dos membros da equipa de conteúdos, possa ser feita de uma forma mais prática e, ao mesmo tempo, garantir a integridade dos conteúdos.

5.3 Proposta de valor

Pode entender-se por proposta de valor, a forma como uma empresa descreve os benefícios, tais como benefícios funcionais, de um produto desenvolvido pela mesma, que produz valor para um segmento específico de clientes (Bagchi & Tulske, 2000). Complementarmente, segundo (Lindič & Marques da Silva, 2011), uma proposta de valor também “[...] descreve como uma oferta de uma empresa difere das ofertas da concorrência e explica o porquê dos clientes comprarem à empresa.”⁴⁹.

Este projeto, realizado na ALERT, é inserido numa atividade de apoio, nomeadamente de desenvolvimento tecnológico, cuja proposta de valor pode ser determinada através da ferramenta - *Value Proposition Canvas*.

O *Value Proposition Canvas* é uma ferramenta que permite auxiliar na garantia de que um produto se encontra bem enquadrado em relação ao valor para o cliente⁵⁰ e às necessidades definidas (International, B2B, s.d.). O *Value Proposition Canvas* é alcançado através do desenvolvimento de duas componentes (Figura 11) (Osterwalder, et al., 2014):

- Perfil do cliente: componente em que se descreve o entendimento do produto por parte do cliente;
- Mapa de valor: componente em que se detalha a forma como é previsto criar valor para o cliente.

⁴⁹ Tradução livre do autor. No original “A value proposition describes how a company’s offer differs from those of its competitors and explains why customers buy from the company.” (Lindič & Marques da Silva, 2011).

⁵⁰ Valor para o cliente é o valor que o cliente atribui a um produto com base na sua perceção.

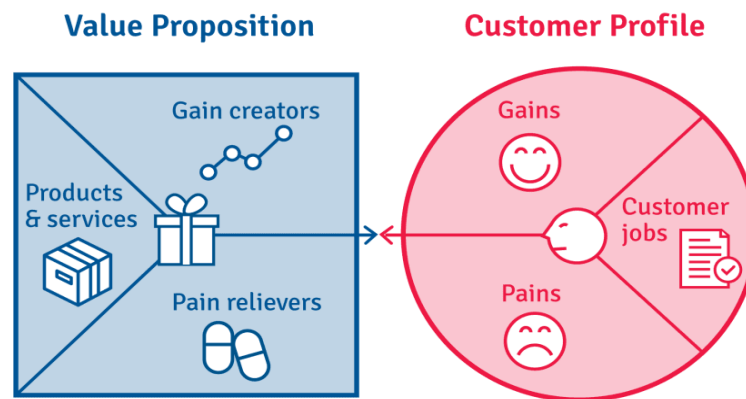


Figura 11 – *Value Proposition Canvas* (Liu, et al., 2020)

No que diz respeito ao perfil do cliente, é necessário definir os benefícios expectáveis (*Gains*), os sacrifícios enfrentados pelos clientes (*Pains*) e, por fim, as tarefas dos clientes (*Customer jobs*).

Os benefícios expectáveis (*Gains*) representam benefícios, resultados ou características que os clientes desejam ou requerem, que o produto apresente. São aspetos que auxiliam os clientes a desempenhar o seu trabalho da melhor forma (Osterwalder, et al., 2014).

Os sacrifícios enfrentados pelos clientes (*Pains*) correspondem a aspetos que dificultam ou impedem que o cliente desempenhe o seu trabalho, o que, conseqüentemente, pode representar riscos (Osterwalder, et al., 2014).

As tarefas dos clientes (*Customer jobs*) podem ser tarefas que estes estão a tentar executar, problemas para os quais procuram solução ou necessidades que se encontram sem resposta (Osterwalder, et al., 2014).

Relativamente ao mapa de valor, é necessário identificar os benefícios proporcionados pelo produto ao cliente (*Gain creators*), os aliviadores de sacrifícios (*Pain relievers*), e, por fim, os produtos e serviços (*Products & services*).

Os benefícios proporcionados pelo produto (*Gain creators*) descrevem como o produto, ou serviço, representa vantagens para os clientes, isto é, determinam como o produto representa benefícios relativamente às necessidades expectáveis pelo cliente (Osterwalder, et al., 2014).

Os aliviadores de sacrifícios (*Pain relievers*) detalham como é que o produto, ou serviço, se propõe a aliviar os sacrifícios identificados, isto é, determinam como é que o produto reduz ou elimina os sacrifícios que dificultam o trabalho ao cliente (Osterwalder, et al., 2014).

Os produtos e serviços (*Products & services*) correspondem ao produto, ou serviço, que se pretende desenvolver.

No âmbito deste projeto, as tarefas dos clientes (*Customer jobs*) estão relacionadas com a gestão de conteúdos. A equipa de conteúdos realiza a gestão de conteúdos do ALERT®, o que inclui, a visualização, atualização e criação de conteúdos.

Atualmente a equipa de conteúdos utiliza uma solução para a gestão de conteúdos que passa pela aplicação de processos, detalhados nas secções 2.2.1 e 2.2.2, descritos pela equipa como bastante complexos. A utilização destes processos traduz-se em vários sacrifícios enfrentados (*Pains*) pela equipa de conteúdos, entre os quais:

- A necessidade de recorrer a várias ferramentas para a gestão de conteúdos;
- A imprescindibilidade do conhecimento e recurso à linguagem SQL;
- A complexidade dos processos, na medida em que, exigem um conhecimento profundo do modelo de dados;
- A possibilidade de ocorrência de erros na visualização, alteração ou criação de conteúdos, devido à criação de SQL *queries* e execução das mesmas na base de dados.

Relativamente aos benefícios expectáveis (*Gains*), espera-se que a solução desenvolvida no âmbito deste projeto permita:

- Abstração do SQL para a equipa de conteúdos;
- Diminuição do tempo despendido na gestão dos conteúdos;
- Integridade dos conteúdos, o que se traduz em menor probabilidade de ocorrência de erros na gestão dos conteúdos

No que diz respeito ao produto (*Products & services*), espera-se o desenvolvimento de uma plataforma que auxilie a equipa de conteúdos na gestão de conteúdos.

Relativamente aos aliviadores de sacrifícios (*Pain relievers*), a solução pretende aliviar os sacrifícios identificados anteriormente. Pretende-se:

- Reduzir o número de ferramentas (secção 2.2) utilizadas pelos membros da equipa de conteúdos através do desenvolvimento do *back-end*. O *back-end* implementa um conjunto de funcionalidades que permite que o membro da equipa de conteúdos apenas necessite de utilizar essas funcionalidades para realizar as suas tarefas relativamente à gestão de conteúdos;
- Eliminar a necessidade de conhecimentos em SQL por parte dos membros da equipa de conteúdos. O *back-end* permite que os membros da equipa de conteúdos possam desempenhar as suas tarefas sem terem conhecimentos da linguagem SQL;
- Simplificar a gestão de conteúdos. É pretendido que as funcionalidades do *back-end* implementem a lógica de negócio de forma a que o membro da equipa de conteúdos apenas tenha de preencher os campos para a criação ou atualização de um determinado conteúdo;

- Reduzir os riscos associados a erros que possam ocorrer durante a criação ou atualização dos conteúdos, através da implementação das regras de negócio nas funcionalidades do *back-end*, de forma a garantir a integridade dos conteúdos.

Relativamente aos benefícios proporcionados pelo produto (*Gain creators*), a solução representa vantagens para o cliente, uma vez que:

- Permite a abstração do SQL. Os membros da equipa de conteúdos deixam de ser obrigados a desenvolver SQL *queries* para a gestão de conteúdos e passam a apenas a executar as funcionalidades disponibilizadas pelo *back-end*;
- Garante a integridade dos conteúdos através de validações realizadas pelo *back-end* nos momentos de criação e alteração de conteúdos;
- Permite reduzir o tempo e esforço despendidos pelos membros da equipa de conteúdos, devido ao facto de estes deixarem de necessitar do conjunto de ferramentas que necessitavam antes (secção 2.2), uma vez que o *back-end* concentra em si todas as funcionalidades que antes se encontravam dispersas pelas diferentes ferramentas.

Tendo em conta o *Value Proposition Canvas*, descrito anteriormente, pode definir-se a proposta de valor através da resposta a cada uma das seguintes questões:

- Qual é o produto?
- Quem é o cliente alvo?
- Que valor acrescenta o produto?
- O que torna o produto único?

No contexto deste projeto, e com base nas questões anteriores, a proposta de valor consiste numa aplicação *back-end*, mais tarde consumida por uma plataforma a ser desenvolvida pela ALERT, que concentra em si um conjunto de funcionalidades que permitem a gestão dos conteúdos do ALERT® para a equipa de conteúdos da ALERT.

Esta aplicação oferece uma solução que vai tornar a gestão dos conteúdos mais prática, simples e segura, na medida em que garante a integridade dos conteúdos. Para além disto, a aplicação diferencia-se pelo facto de permitir aos membros da equipa de conteúdos fazer a gestão de conteúdos de forma abstraída da linguagem SQL.

5.4 QFD

O *Quality Function Deployment* (QFD), é uma técnica que transforma as necessidades dos clientes em requisitos técnicos para um produto. Esta técnica é usada para escolher e priorizar as funcionalidades de um produto que satisfaçam as necessidades dos clientes (Kahraman, et al., 2006). “O QFD tem em consideração a voz do cliente e dessa forma aborda as lacunas entre as expectativas do cliente e a experiência real do produto”⁵¹.

No que diz respeito ao QFD, é necessário desenvolver a casa da qualidade (*House of Quality*). “A casa da qualidade é uma ferramenta de desenho que suporta o processamento de informação e tomada de decisão no processo de desenho de engenharia.”⁵².

No âmbito da casa da qualidade é necessário identificar:

- As necessidades/requisitos do cliente;
- O fator de importância para cada um dos requisitos identificados. Para isto, é feita uma classificação de cada um dos requisitos de acordo com a importância que representam para o cliente;
- Os requisitos técnicos que dão resposta às necessidades do cliente;
- Correlação entre as necessidades do cliente, identificadas com os requisitos técnicos da solução;
- Identificar a relação, ou falta dela, entre os requisitos técnicos identificados, o que se representa no “telhado” da casa da qualidade.

Para este projeto, em particular, a casa da qualidade está representada na Figura 12.

⁵¹ Tradução livre do autor. No original “*It is driven by the voice of the customer and because of that, it helps service providers to address gaps between specific and holistic components of customer expectations and actual service experience.*” (Cudney & Elrod, 2011).

⁵² Tradução livre do autor. No original “*The House of Quality (HoQ) is a popular design tool that supports information processing and decision making in the engineering design process.*” (Olewnik & Lewis, 2008).

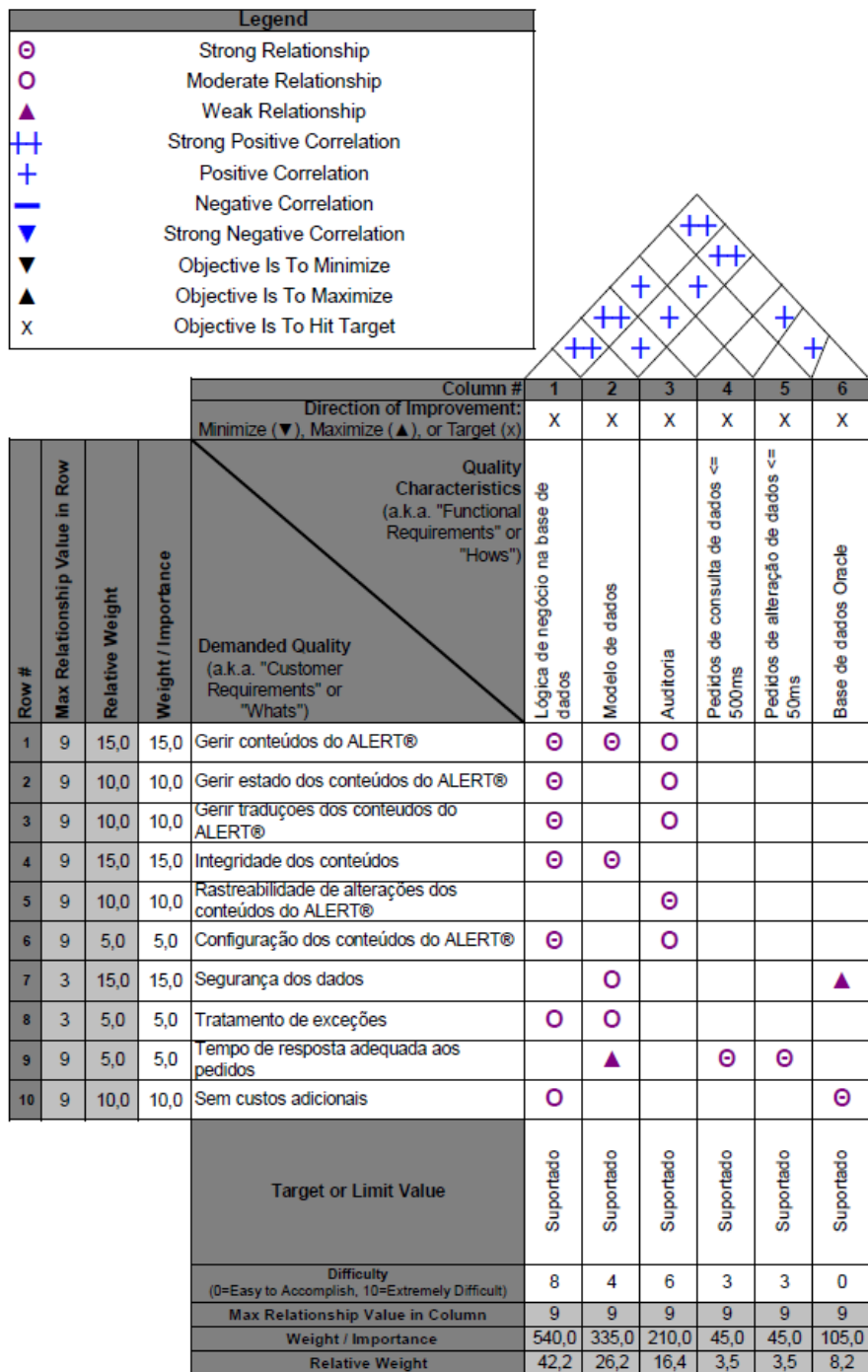


Figura 12 – Casa da qualidade (QFD)

5.5 Método de Análise Hierárquica

Após apresentar a oportunidade, exposta na secção 5.1, é necessário identificar e avaliar as diferentes alternativas que visam solucioná-la. Para avaliar as diferentes alternativas recorre-se ao método de análise hierárquica, *Analytic Hierarchy Process (AHP)*.

O método AHP, criado pelo professor Thomas L. Saaty em 1980, é um método que permite analisar alternativas complexas e tem como finalidade ajudar a tomar decisões através da utilização de fórmulas matemáticas (Saaty, 1988). No método criado pelo Thomas L. Saaty podem identificar-se as seguintes fases (Saaty, 2002):

- Definir o problema;
- Estruturar a divisão hierárquica, identificando os diferentes critérios e alternativas para solucionar o problema. É necessário definir um modelo para que os critérios e alternativas sejam, de facto, relevantes no âmbito do problema identificado;
- Definir matrizes de comparação par a par quadrada. Nesta fase é feita uma comparação entre os diferentes elementos através de “[...] uma escala numérica que indica quantas vezes mais importante um elemento é relativamente a outro elemento no que diz respeito ao critério ou propriedade em relação à qual são comparados.”⁵³. Esta escala, presente na Tabela 5, definida por Thomas L. Saaty, é denominada de escala fundamental e é utilizada na aplicação do método AHP;
- Utilizar as prioridades obtidas das comparações para ponderar as prioridades para cada um dos elementos.

Tabela 5 – Escala fundamental – Níveis de importância de comparações (Saaty, 2002)

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra

⁵³ Tradução livre do autor. No original “[...] a scale of numbers that indicates how many times more important or dominant one element is over another element with respect to the criterion or property with respect to which they are compared.” (Saaty, 2002).

9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2, 4, 6, 8	Valores intermédios	Quando se procura uma condição de compromisso entre duas definições

O AHP é um método em que as prioridades dos critérios são determinadas de uma forma subjetiva, pelo que, é vital avaliar e garantir a consistência das comparações realizadas. Desta forma, é necessário fazer o cálculo da razão de consistência (RC) que “[...] é uma medida usada no método AHP para aumentar a validade dos resultados [...]”⁵⁴. Considera-se que as comparações são confiáveis se o valor do RC calculado for inferior a 0.10, caso contrário, considera-se que os resultados obtidos não apresentam valores consistentes e necessitam de ser revistos e melhorados (Al-Harbi, 2001).

Para calcular o RC é necessário primeiro calcular o índice de consistência (IC), que pode ser determinado através da seguinte fórmula:

$$IC = \frac{\lambda_{max} - n}{n - 1} \quad (1)$$

, onde λ_{max} corresponde ao maior valor próprio da matriz obtido pela multiplicação da matriz comparativa de critérios pela matriz de prioridades relativas e n representa a ordem da matriz (Al-Harbi, 2001).

Depois de obtido o IC, pode-se proceder ao cálculo do RC através da fórmula:

$$RC = \frac{IC}{IR} \quad (2)$$

, em que IR representa o índice aleatório relativo a número de comparações par a par efetuadas, Tabela 6, e IC representa o índice de consistência.

Tabela 6 – Valores de IR para matrizes quadradas de ordem n (Al-Harbi, 2001)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

⁵⁴ Tradução livre do autor. No original “[...] consistency ratio (CR) of the pairwise comparison matrix is a measure used in AHP to improve the validity of the outcome [...]” (Russo & Camanho, 2015).

Com base na Tabela 6, verifica-se que a ordem das matrizes quadradas deve ser igual ou superior a três, o que significa que na aplicação do AHP tem de ser identificados pelo menos três critérios e alternativas.

No âmbito deste projeto, na Figura 13 está representada a árvore hierárquica de decisão. No topo da árvore é definido o problema - desenvolvimento do *back-end* do *Content Server*. Para este problema, foram identificados quatro critérios distintos – custo, segurança, manutenção e tempo de desenvolvimento. Por fim, são apresentadas as três alternativas que podem auxiliar a resolução do problema – implementar o *back-end* na base de dados, implementar o *back-end* na camada de aplicação e recorrer a uma empresa externa (*outsourcing*).

A utilização do método AHP, neste contexto, permite determinar qual a melhor alternativa para solucionar o problema identificado, tendo em conta os diferentes critérios apresentados anteriormente.

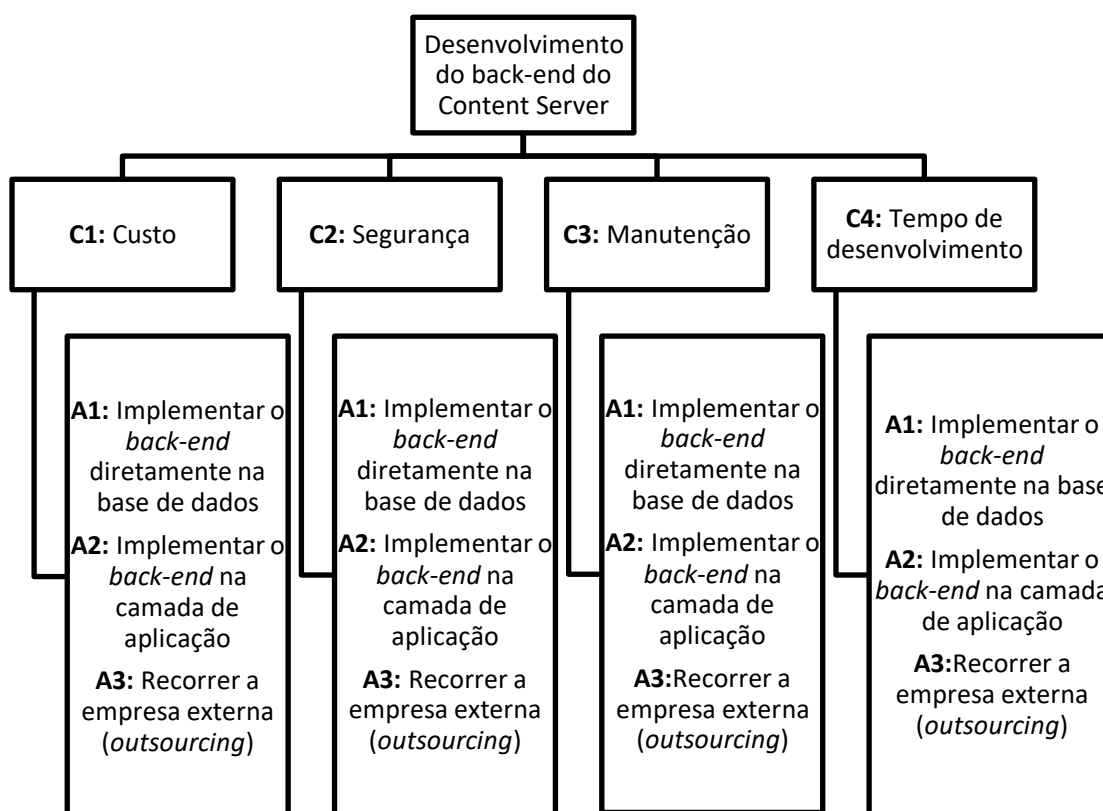


Figura 13 – Árvore hierárquica de decisão para problema do projeto

Após definida a árvore hierárquica de decisão segue-se a fase de comparação dos critérios. Assim, na Tabela 7, é apresentada a matriz de comparação dos critérios, na qual é feita uma comparação da importância entre os critérios.

Tabela 7 – Matriz de comparação dos critérios

	C1	C2	C3	C4
C1	1	1/3	1/2	4
C2	3	1	2	3
C3	2	1/2	1	4
C4	1/4	1/3	1/4	1

Depois de definida a matriz de comparação dos critérios, segue-se a normalização da mesma. Na Tabela 8 é apresentado o primeiro passo da normalização da matriz, em que se calcula a soma de cada uma das colunas.

Tabela 8 – Matriz de comparação de critérios normalizada (1º Passo)

	C1	C2	C3	C4
C1	1	1/3	1/2	4
C2	3	1	2	3
C3	2	1/2	1	4
C4	1/4	1/3	1/4	1
Somatório	6 1/4	2 1/6	3 3/4	12

Após o cálculo do somatório de cada coluna, procede-se à divisão de cada valor da matriz pelo resultado do somatório da respectiva coluna. Assim, a Tabela 9 apresenta a matriz de comparação de critérios normalizada.

Tabela 9 – Matriz de comparação de critérios normalizada

	C1	C2	C3	C4
C1	1/6	1/6	1/7	1/3
C2	1/2	1/2	1/2	1/4
C3	1/3	1/4	1/4	1/3
C4	0,04	1/6	0,07	0,08

Depois da matriz de comparação normalizada, segue-se o cálculo do vetor de prioridades, que permite identificar a ordem de importância de cada critério (Nicola, 2020). O vetor de prioridades, apresentado na Tabela 10, é obtido através da “[...] média aritmética dos valores de cada linha da matriz normalizada [...]” (Nicola, 2020).

Tabela 10 – Vetor de prioridades da matriz de comparação dos critérios

	C1	C2	C3	C4	Prioridades
C1	1/6	1/6	1/7	1/3	0,1951
C2	1/2	1/2	1/2	1/4	0,4312
C3	1/3	1/4	1/4	1/3	0,2877
C4	0,04	1/6	0,07	0,08	0,0860

Com base na Tabela 10, verifica-se que o critério com maior prioridade é o C2 – segurança – com 43,12%, seguindo-se o critério C3 – manutenção – com 28,77%, o critério C1 – custo – com 19,51%, e, por fim, o critério C4 – tempo de desenvolvimento – com 8,6%.

O passo seguinte consiste em avaliar a consistência do vetor de prioridades, apresentado na Tabela 10. Desta forma, é necessário calcular o RC e conseqüentemente o IC. Pode calcular-se o IC através da fórmula (1), sendo necessário calcular o λ_{max} . Para calcular o λ_{max} pode utilizar-se a seguinte fórmula (Nicola, 2020):

$$Ax = \lambda_{max}x \quad (3)$$

, onde A representa a matriz de comparação de critérios, Tabela 7, e x corresponde ao vetor de prioridades, previamente apresentado na Tabela 10. Assim, através da fórmula (3), é possível calcular o valor de λ_{max} :

$$1. \begin{bmatrix} 1 & 1/3 & 1/2 & 4 \\ 3 & 1 & 2 & 3 \\ 2 & 1/2 & 1 & 4 \\ 1/4 & 1/3 & 1/4 & 1 \end{bmatrix} * \begin{bmatrix} 0,1951 \\ 0,4312 \\ 0,2877 \\ 0,0860 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1951 \\ 0,4312 \\ 0,2877 \\ 0,0860 \end{bmatrix}$$

$$2. \begin{bmatrix} 0,8266 \\ 1,8499 \\ 1,2374 \\ 0,3504 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1951 \\ 0,4312 \\ 0,2877 \\ 0,0860 \end{bmatrix}$$

$$3. \lambda_{max} = 4,2258$$

Com λ_{max} calculado, é possível determinar o valor de IC com base na fórmula (1):

$$IC = \frac{4,2258 - 4}{4 - 1} = 0,0753$$

Desta forma, já com o IC calculado, através da fórmula (2) é possível obter o valor de RC. O valor de IR, que advém da Tabela 6, neste caso, é de 0,90, uma vez que o número de critérios corresponde a quatro. Assim, com base na fórmula (2):

$$RC = \frac{0,0753}{0,90} = 0,0836$$

Uma vez que o valor de RC (0,0836) é inferior a 0.1, pode concluir-se que os valores das prioridades são consistentes.

Após calcular o vetor de prioridades dos critérios definidos, segue-se o cálculo do vetor de prioridades de cada critério, tendo em conta a importância relativa de cada uma das alternativas definidas na árvore hierárquica, refazendo os passos anteriores.

Assim, a Tabela 11, a Tabela 12, a Tabela 13 e a Tabela 14 apresentam os diferentes passos para o cálculo do vetor de prioridades do critério C1 – custo – tendo em conta as alternativas.

Tabela 11 – Matriz de comparação de alternativas para o critério C1

C1	A1	A2	A3
A1	1	1/2	1/3
A2	2	1	1/4
A3	3	4	1

Tabela 12 - Matriz de comparação de alternativas para o critério C1 normalizada (1º Passo)

C1	A1	A2	A3
A1	1	1/2	1/3
A2	2	1	1/4
A3	3	4	1
Somatório	6	5 1/2	1 4/7

Tabela 13 – Matriz de comparação de alternativas para o critério C1 normalizada

C1	A1	A2	A3
A1	1/6	0,0909	1/5
A2	1/3	1/5	1/6
A3	1/2	5/7	5/8

Tabela 14 - Vetor de prioridades da matriz de comparação de alternativas para o critério C1

C1	A1	A2	A3	Prioridades
A1	1/6	0,0909	1/5	0,1560
A2	1/3	1/5	1/6	0,2243
A3	1/2	5/7	5/8	0,6196

Segue-se a análise de consistência do vetor de prioridades:

$$1. \begin{bmatrix} 1 & 1/2 & 1/3 \\ 2 & 1 & 1/4 \\ 3 & 4 & 1 \end{bmatrix} * \begin{bmatrix} 0,1560 \\ 0,2243 \\ 0,6196 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1560 \\ 0,2243 \\ 0,6196 \end{bmatrix}$$

$$2. \begin{bmatrix} 0,4747 \\ 0,6913 \\ 1,9851 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1560 \\ 0,2243 \\ 0,6196 \end{bmatrix}$$

$$3. \lambda_{max} = 3,1093$$

Com λ_{max} calculado, é possível determinar o valor de IC com base na fórmula (1):

$$IC = \frac{3,1093 - 3}{3 - 1} = 0,0546$$

Desta forma, já com o IC calculado, através da fórmula (2) é possível obter o valor de RC. O valor de IR, que advém da Tabela 6, neste caso, é de 0,58, uma vez que o número de critérios corresponde a três. Assim, com base na fórmula (2):

$$RC = \frac{0,0546}{0,58} = 0,0942$$

Como o valor de RC (0,0942) é inferior a 0.1, pode concluir-se que os valores das prioridades são consistentes.

A Tabela 15, a Tabela 16, a Tabela 17 e a Tabela 18 apresentam os diferentes passos para o cálculo do vetor de prioridades do critério C2 – segurança – tendo em conta as alternativas.

Tabela 15 - Matriz de comparação de alternativas para o critério C2

C2	A1	A2	A3
A1	1	3	5
A2	1/3	1	4
A3	1/5	1/4	1

Tabela 16 - Matriz de comparação de alternativas para o critério C2 normalizada (1º Passo)

C2	A1	A2	A3
A1	1	3	5
A2	1/3	1	4
A3	1/5	1/4	1
Somatório	1 1/2	4 1/4	10

Tabela 17 - Matriz de comparação de alternativas para o critério C2 normalizada

C2	A1	A2	A3
A1	2/3	5/7	1/2
A2	2/9	1/4	2/5
A3	1/8	0,0588	0,1

Tabela 18 - Vetor de prioridades da matriz de comparação de alternativas para o critério C2

C2	A1	A2	A3	Prioridades
A1	1/6	0,0909	1/5	0,6194
A2	1/3	1/5	1/6	0,2842
A3	1/2	5/7	5/8	0,0964

Segue-se a análise de consistência do vetor de prioridades:

$$1. \begin{bmatrix} 1 & 3 & 5 \\ 1/3 & 1 & 4 \\ 1/5 & 1/4 & 1 \end{bmatrix} * \begin{bmatrix} 0,6194 \\ 0,2842 \\ 0,0964 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,6194 \\ 0,2842 \\ 0,0964 \end{bmatrix}$$

$$2. \begin{bmatrix} 1,9541 \\ 0,8764 \\ 0,2913 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,6194 \\ 0,2842 \\ 0,0964 \end{bmatrix}$$

$$3. \lambda_{max} = 3,0867$$

Com λ_{max} calculado, é possível determinar o valor de IC com base na fórmula (1):

$$IC = \frac{3,0867 - 3}{3 - 1} = 0,0433$$

Desta forma, já com o IC calculado, através da fórmula (2) é possível obter o valor de RC. O valor de IR, que advém da Tabela 6, neste caso, é de 0,58, uma vez que o número de critérios corresponde a três. Assim, com base na fórmula (2):

$$RC = \frac{0,0433}{0,58} = 0,0747$$

Como o valor de RC (0,0747) é inferior a 0.1, pode concluir-se que os valores das prioridades são consistentes.

A Tabela 19, a Tabela 20, a Tabela 21 e a Tabela 22 apresentam os diferentes passos para o cálculo do vetor de prioridades do critério C3 – manutenção – tendo em conta as alternativas.

Tabela 19 - Matriz de comparação de alternativas para o critério C3

C3	A1	A2	A3
A1	1	3	4
A2	1/3	1	3
A3	1/4	1/3	1

Tabela 20 - Matriz de comparação de alternativas para o critério C3 normalizada (1º Passo)

C3	A1	A2	A3
A1	1	3	4
A2	1/3	1	3
A3	1/4	1/3	1
Somatório	1 4/7	4 1/3	8

Tabela 21 - Matriz de comparação de alternativas para o critério C3 normalizada

C3	A1	A2	A3
A1	5/8	2/3	1/2
A2	1/5	1/4	3/8
A3	1/6	0,0769	1/8

Tabela 22 - Vetor de prioridades da matriz de comparação de alternativas para o critério C3

C3	A1	A2	A3	Prioridades
A1	5/8	2/3	1/2	0,6080
A2	1/5	1/4	3/8	0,2721
A3	1/6	0,0769	1/8	0,1199

Segue-se a análise de consistência do vetor de prioridades:

$$1. \begin{bmatrix} 1 & 3 & 4 \\ 1/3 & 1 & 3 \\ 1/4 & 1/3 & 1 \end{bmatrix} * \begin{bmatrix} 0,6080 \\ 0,2721 \\ 0,1199 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,6080 \\ 0,2721 \\ 0,1199 \end{bmatrix}$$

$$2. \begin{bmatrix} 1,9040 \\ 0,8346 \\ 0,3626 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,6080 \\ 0,2721 \\ 0,1199 \end{bmatrix}$$

$$3. \lambda_{max} = 3,0741$$

Com λ_{max} calculado, é possível determinar o valor de IC com base na fórmula (1):

$$IC = \frac{3,0741 - 3}{3 - 1} = 0,0371$$

Desta forma, já com o IC calculado, através da fórmula (2) é possível obter o valor de RC. O valor de IR, que advém da Tabela 6, neste caso, é de 0,58, uma vez que o número de critérios corresponde a três. Assim, com base na fórmula (2):

$$RC = \frac{0,0371}{0,58} = 0,0639$$

Como o valor de RC (0,0639) é inferior a 0.1, pode concluir-se que os valores das prioridades são consistentes.

A Tabela 23, a Tabela 24, a Tabela 25 e a Tabela 26 apresentam os diferentes passos para o cálculo do vetor de prioridades do critério C4 – tempo de desenvolvimento – tendo em conta as alternativas.

Tabela 23 - Matriz de comparação de alternativas para o critério C4

C4	A1	A2	A3
A1	1	2	1/5
A2	1/2	1	1/5

A3	5	5	1
-----------	---	---	---

Tabela 24 - Matriz de comparação de alternativas para o critério C4 normalizada (1º Passo)

C4	A1	A2	A3
A1	1	2	1/5
A2	1/2	1	1/5
A3	5	5	1
Somatório	6 1/2	8	1 2/5

Tabela 25 - Matriz de comparação de alternativas para o critério C4 normalizada

C4	A1	A2	A3
A1	1/6	1/4	1/7
A2	0,0769	1/8	1/7
A3	7/9	5/8	5/7

Tabela 26 - Vetor de prioridades da matriz de comparação de alternativas para o critério C4

C4	A1	A2	A3	Prioridades
A1	1/6	1/4	1/7	0,1822
A2	0,0769	1/8	1/7	0,1149
A3	7/9	5/8	5/7	0,7028

Segue-se a análise de consistência do vetor de prioridades:

$$1. \begin{bmatrix} 1 & 2 & 1/5 \\ 1/2 & 1 & 1/5 \\ 5 & 5 & 1 \end{bmatrix} * \begin{bmatrix} 0,1822 \\ 0,1149 \\ 0,7028 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1822 \\ 0,1149 \\ 0,7028 \end{bmatrix}$$

$$2. \begin{bmatrix} 0,5527 \\ 0,3466 \\ 2,1886 \end{bmatrix} \cong \lambda_{max} \begin{bmatrix} 0,1822 \\ 0,1149 \\ 0,7028 \end{bmatrix}$$

$$3. \lambda_{max} = 3,0542$$

Com λ_{max} calculado, é possível determinar o valor de IC com base na fórmula (1):

$$IC = \frac{3,0542 - 3}{3 - 1} = 0,0271$$

Desta forma, já com o IC calculado, através da fórmula (2) é possível obter o valor de RC. O valor de IR, que advém da Tabela 6, neste caso, é de 0,58, uma vez que o número de critérios corresponde a três. Assim, com base na fórmula (2):

$$RC = \frac{0,0271}{0,58} = 0,0467$$

Como o valor de RC (0,0467) é inferior a 0.1, pode concluir-se que os valores das prioridades são consistentes.

Na Figura 14 é apresentada a árvore hierárquica já com o peso dos critérios e com os vetores de prioridades obtidos para cada uma das alternativas, relativamente a cada um dos critérios definidos.

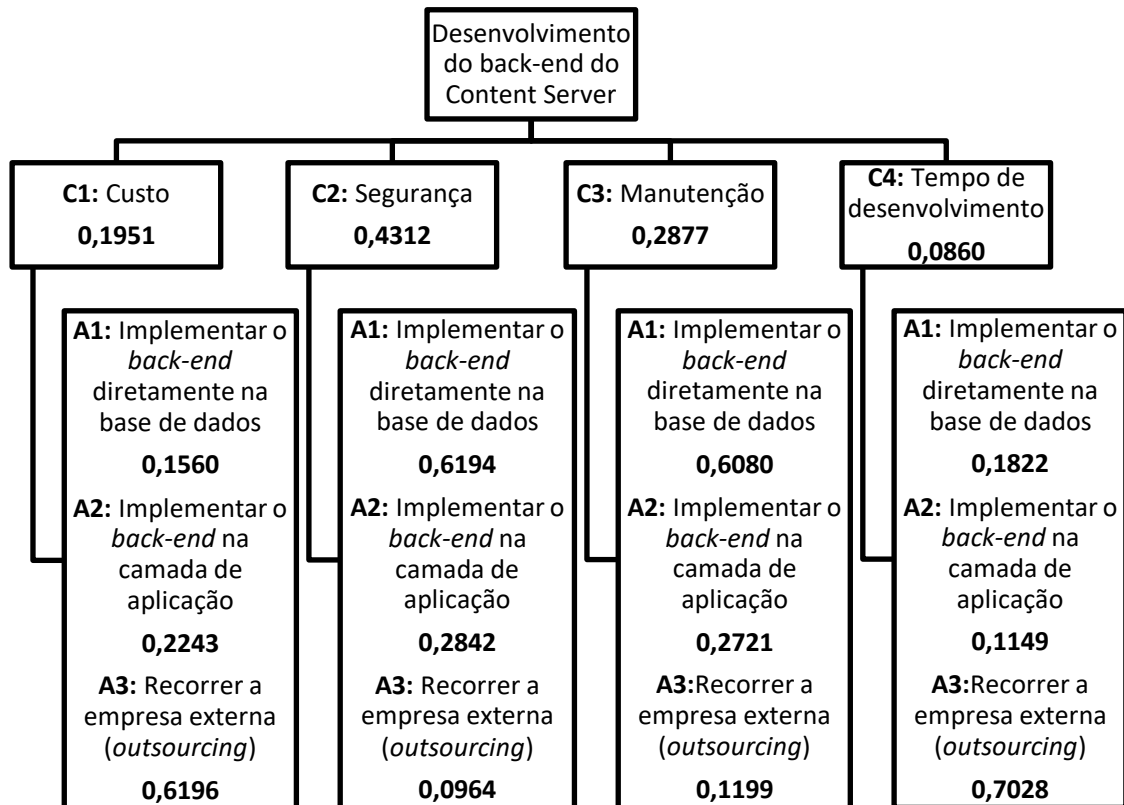


Figura 14 – Árvore hierárquica com as prioridades definidas

Assim, chega-se à última etapa do AHP, que consiste em obter a prioridade composta para as alternativas (Nicola, 2020), através da multiplicação da matriz das prioridades calculadas para cada critério pelo peso de cada critério:

$$\begin{bmatrix} 0,1560 & 0,6194 & 0,6080 & 0,1822 \\ 0,2243 & 0,2842 & 0,2721 & 0,1149 \\ 0,6196 & 0,0964 & 0,1199 & 0,7028 \end{bmatrix} * \begin{bmatrix} 0,1951 \\ 0,4312 \\ 0,2877 \\ 0,0860 \end{bmatrix} = \begin{bmatrix} 0,4881 \\ 0,2545 \\ 0,2574 \end{bmatrix}$$

Em função dos critérios definidos e das respectivas importâncias definidas, verifica-se que a solução mais adequada para o problema definido é a alternativa A1 – implementar o *back-end* diretamente na base de dados – que obteve um resultado de 48,81%. Relativamente às restantes alternativas, a A2 – implementar o *back-end* na camada de aplicação – obteve um valor de 25,45% e a A3 – recorrer a empresa externa (*outsourcing*) – obteve um valor de 25,75%.

6 Engenharia de Requisitos

A engenharia de requisitos, também conhecida como análise de requisitos, consiste no processo de definir as necessidades de um cliente relativamente a um projeto (Visual Paradigm, s.d.).

Recorreu-se à técnica de entrevista, como forma de levantamento dos requisitos juntamente com a equipa de conteúdos.

Nas seguintes subsecções são apresentadas as partes interessadas e os atores deste projeto, os requisitos funcionais e os requisitos não funcionais.

6.1 Partes Interessadas

As partes interessadas (*stakeholders*) de um projeto, são as pessoas ou grupos de pessoas que são afetadas, direta ou indiretamente pelo mesmo.

No contexto desta dissertação, as partes interessadas identificadas são:

- Equipa de conteúdos;
- Clientes do ALERT®.

Tal como referido na secção 1.5, uma das partes interessadas deste projeto, é a equipa de conteúdos, que, sendo responsável pela gestão de conteúdos, beneficia deste projeto que tem como finalidade contribuir para o desenvolvimento de uma plataforma que permita a gestão dos conteúdos do ALERT® de uma forma intuitiva e segura.

Outra parte interessada, são os clientes do ALERT®, na medida em que, este projeto contribui para a criação de conteúdos de forma mais rápida e íntegra, uma vez que o projeto pretende acelerar a disponibilização de conteúdos no ALERT® e diminuir as falhas na sua configuração.

6.2 Atores do Sistema

De acordo com o artigo *Unified Modeling Language User Guide* (Booch, et al., 1999), o ator de um sistema corresponde a uma pessoa/sistema que interage com um sistema por algum motivo.

No âmbito deste projeto existe apenas um ator, que, como referido na secção 1.5, são os membros da equipa de conteúdos, pelo que a solução deste projeto visa apenas ser utilizada pelos mesmos.

6.3 Requisitos Funcionais

Os requisitos funcionais correspondem às funcionalidades de que os utilizadores de um sistema usufruem (Booch, et al., 1999).

Na Figura 15, Figura 16 e na Figura 17 são apresentados os requisitos funcionais sob a forma de diagramas de casos de uso. De forma a simplificar a apresentação dos requisitos funcionais, quando é referido o termo “Gestão de conteúdos”, nos requisitos número 1 ao número 26, pretende referir-se a um conjunto de ações que englobam a criação, alteração e visualização de conteúdos.

O diagrama de casos de uso foi separado em três figuras com o objetivo de facilitar a legibilidade.

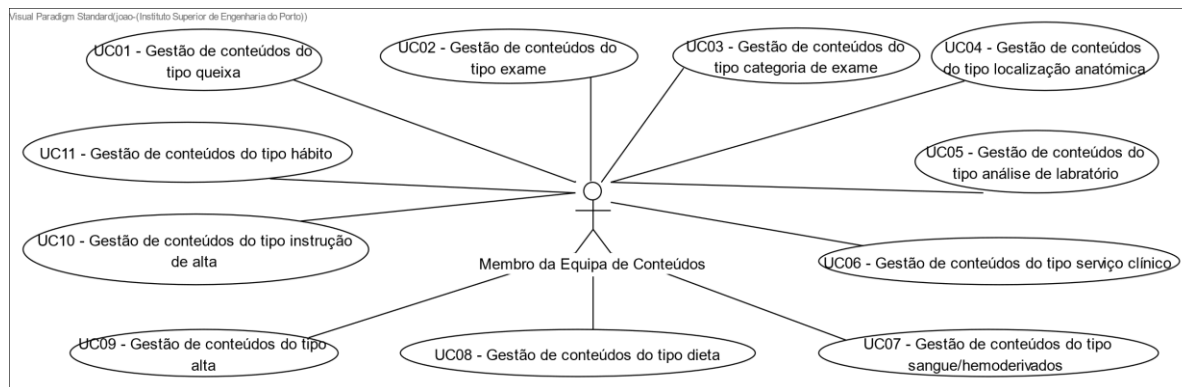


Figura 15 - Diagrama de casos de uso

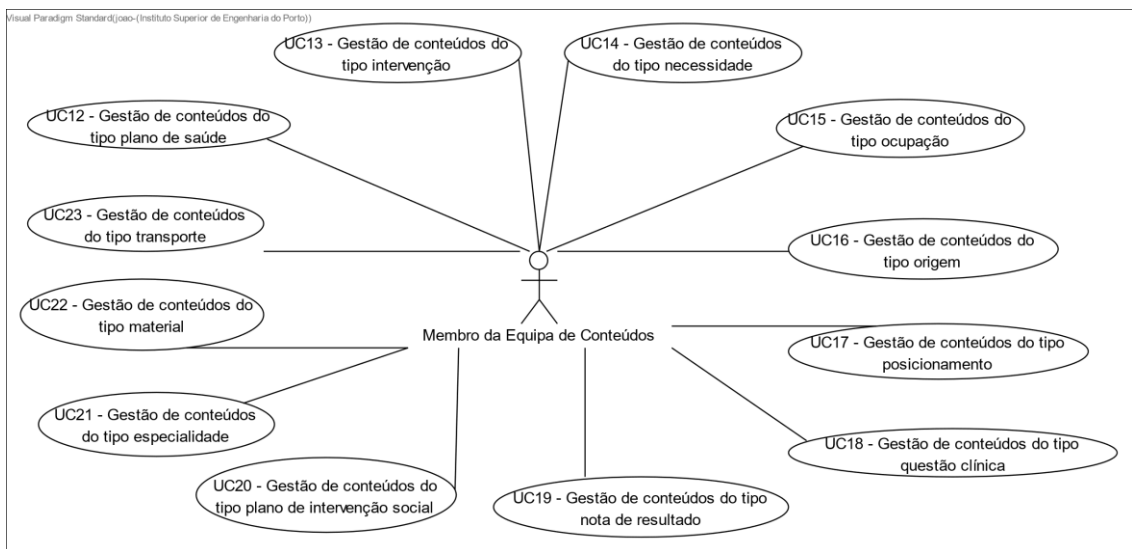


Figura 16 - Diagrama de casos de uso (continuação 1)

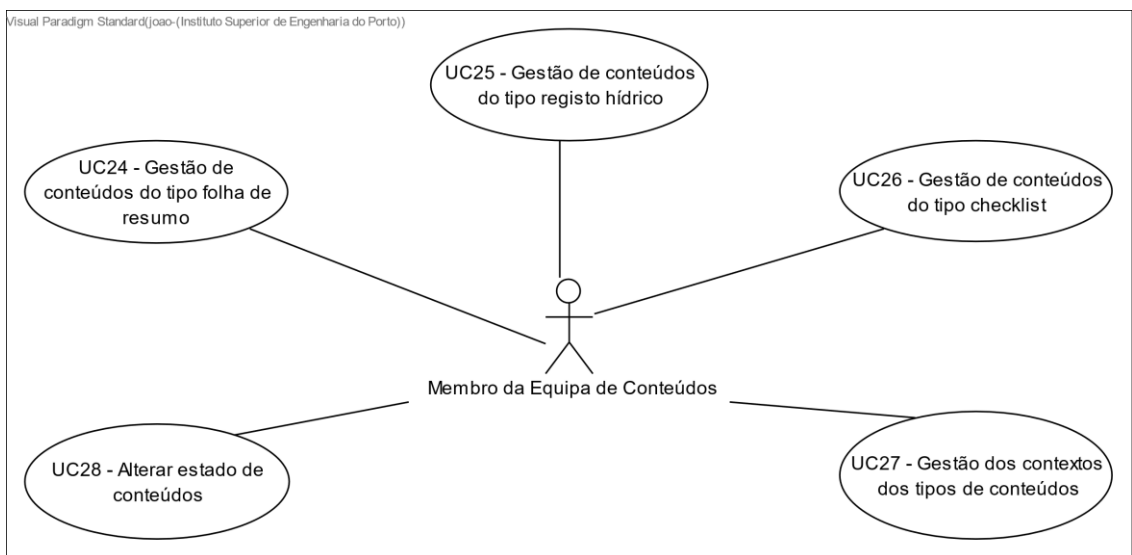


Figura 17 - Diagrama de casos de uso (continuação 2)

Os requisitos funcionais do número 1 ao número 26 possuem vastas similaridades, na medida que as interações com o sistema são idênticas, pelo que, a principal diferença entre eles reside na lógica de negócio. Por esta razão, apenas é apresentada uma descrição mais detalhada para um requisito - UC-01: Gestão de conteúdos do tipo queixa. Adicionalmente, é apresentado o requisito UC-28: Alterar estado de conteúdos.

6.3.1 UC-01: Gestão de conteúdos do tipo queixa

No contexto do ALERT®, conteúdos do tipo queixa, correspondem às diferentes razões que levam um paciente a uma consulta, urgência ou internamento.

Como referido anteriormente na secção 6.3, quando se refere à gestão de um conteúdo, entende-se a criação, alteração e visualização do mesmo.

Neste âmbito, relativamente à criação, um membro da equipa de conteúdos deve ser capaz de criar um conteúdo do tipo queixa. Desta forma, deve preencher informação que é específica deste conteúdo e, adicionalmente, informação que é comum a todos os tipos de conteúdo, nomeadamente:

- Identificador interno;
- Identificador do conteúdo;
- Identificador do estado do conteúdo.

Para cada conteúdo é necessário definir um identificador interno, que é um texto (*string*) único. Este identificador interno, é utilizado pela equipa de conteúdos para conseguir identificar um conteúdo de uma forma mais simples e legível.

Cada conteúdo tem também um identificador de conteúdo. Este identificador é utilizado com o objetivo de identificar, inequivocamente, cada conteúdo.

Por fim, é necessário atribuir um identificador do estado do conteúdo. Um conteúdo pode ser criado com diferentes estados, conforme apresentado na Tabela 27.

Tabela 27 – Diferentes estados de um conteúdo

Identificador do estado de um conteúdo	Descrição do estado
1	Rascunho
2	Pronto para revisão
3	Aprovado
4	Inativo

Na Figura 18 está representado o diagrama de sequência do sistema, relativo à criação de uma queixa. Inicialmente, o membro da equipa de conteúdos escolhe o tipo de conteúdo queixa e, o sistema apresenta a lista de todas as queixas existentes. Em seguida, o membro da equipa de conteúdos inicia a ação de criar uma nova queixa, introduz os dados da mesma e o sistema valida a informação introduzida de forma a garantir que as regras de negócio são respeitadas. Se a informação for válida, a nova queixa é tornada permanente e é feito um registo de forma a identificar a alteração feita (quem fez a alteração, o que alterou, quando alterou e a razão que levou à alteração), caso contrário, uma mensagem de erro é apresentada ao membro da equipa de conteúdos.

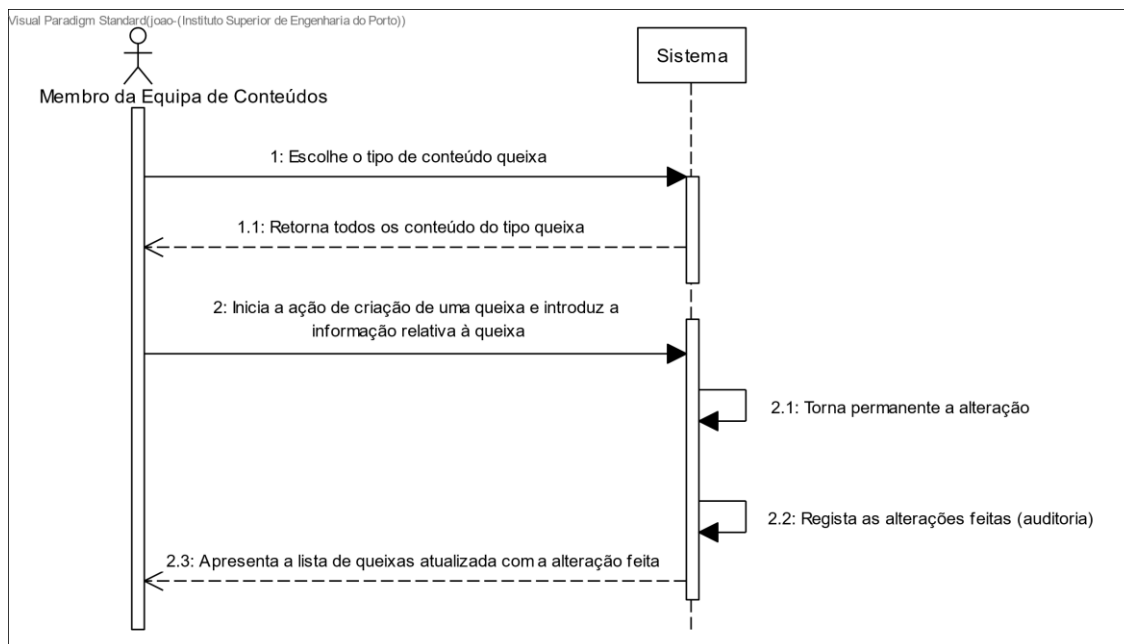


Figura 18 – Diagrama de sequência do sistema do UC01 (criar queixa)

No que diz respeito à alteração de um conteúdo do tipo queixa, um membro da equipa de conteúdos deve ser capaz de alterar um conteúdo existente do tipo queixa. Na Figura 19 é apresentado o diagrama de sequência do sistema, relativo à alteração de uma queixa, em que o membro da equipa de conteúdos escolhe o tipo de conteúdo queixa e o sistema apresenta todos os conteúdos desse tipo. De seguida, o membro da equipa de conteúdos procede à alteração do(s) atributo(s) de uma determinada queixa e o sistema valida as alterações efetuadas, de forma a garantir que as regras de negócio são respeitadas. Se as alterações forem válidas, são tornadas permanentes e é feito um registo de forma a identificar a alteração feita (quem fez a alteração, o que alterou, quando alterou e a razão que levou à alteração), caso contrário, uma mensagem de erro é apresentada ao membro da equipa de conteúdos e as alterações efetuadas não são tornadas permanentes.

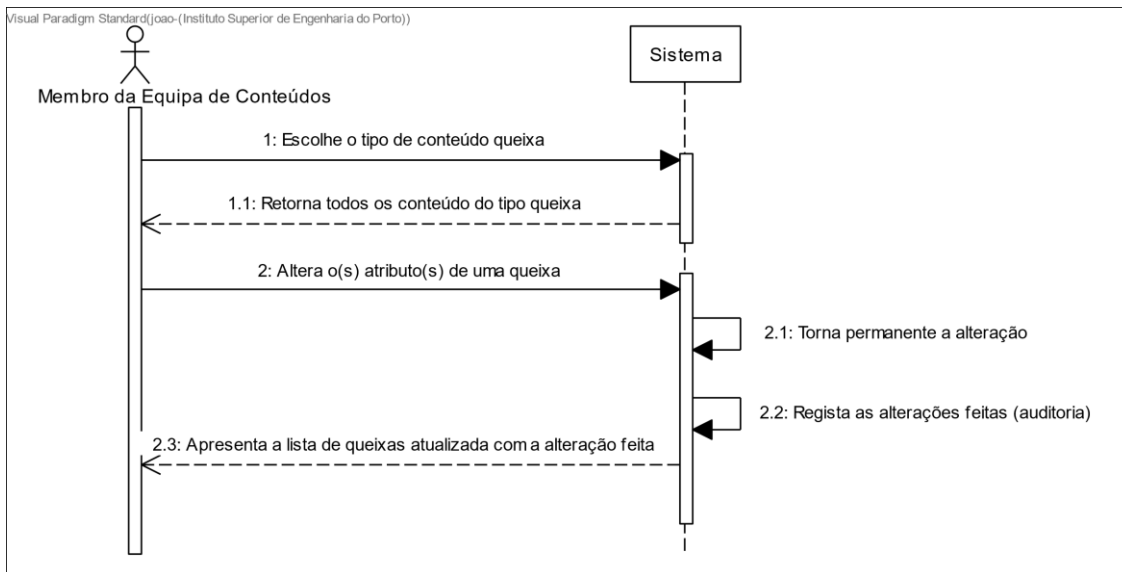


Figura 19 – Diagrama de sequência do sistema do UC01 (alterar queixa)

Por fim, um membro da equipa de conteúdos deve ser capaz de visualizar todos os conteúdos existentes do tipo queixa. Na Figura 20 é apresentado o diagrama de sequência do sistema, relativo à visualização de queixas, em que inicialmente o membro da equipa de conteúdos escolhe o tipo de conteúdos queixa e, em seguida, o sistema retorna todos os conteúdos existentes deste tipo.

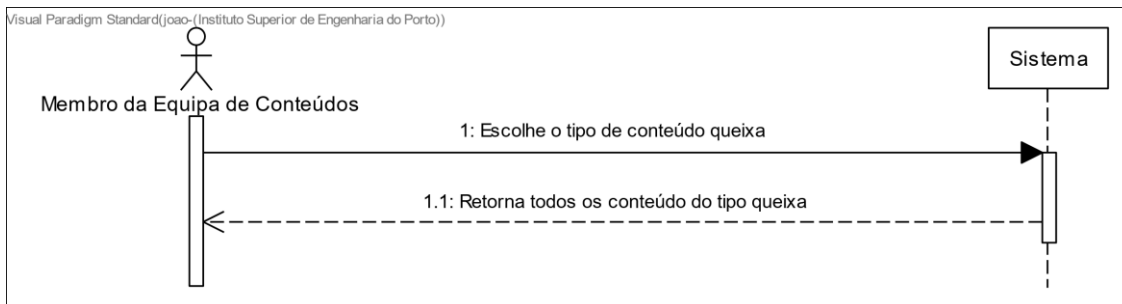


Figura 20 – Diagrama de sequência do sistema do UC01 (visualizar queixas)

6.3.2 UC-28: Alterar estado de conteúdos

Um membro da equipa de conteúdos deve ser capaz de alterar o estado de um determinado tipo de conteúdo. Como representado na Tabela 27, um conteúdo pode apresentar diferentes estados – rascunho, pronto para revisão, aprovado ou inativo.

Através destes estados, é possível gerir o ciclo de vida de um determinado conteúdo. Normalmente, quando um conteúdo é criado, é-lhe atribuído o estado de rascunho. Em seguida, passa para o estado de pronto para revisão, em que um elemento da equipa de conteúdos, diferente do elemento que criou o conteúdo, faz uma revisão do conteúdo criado, de forma a garantir que este se encontra correto e de acordo com as necessidades. Se o conteúdo estiver

corretamente criado, então passa para o estado de aprovado. Mais tarde, se necessário, um conteúdo pode deixar de ser necessário, passando então para o estado de inativo.

Tal como representado na Figura 21, referente ao caso de uso de alterar o estado de um conteúdo, inicialmente, o membro da equipa de conteúdos deve escolher o tipo de conteúdo de que deseja alterar o estado. De seguida, o sistema apresenta todos os conteúdos existentes do tipo de conteúdo escolhido e o membro da equipa de conteúdos altera o estado de um conteúdo. Por fim, o sistema torna permanente a alteração realizada e efetua o registo da mesma (quem fez a alteração, o que alterou, quando alterou e a razão que levou à alteração).

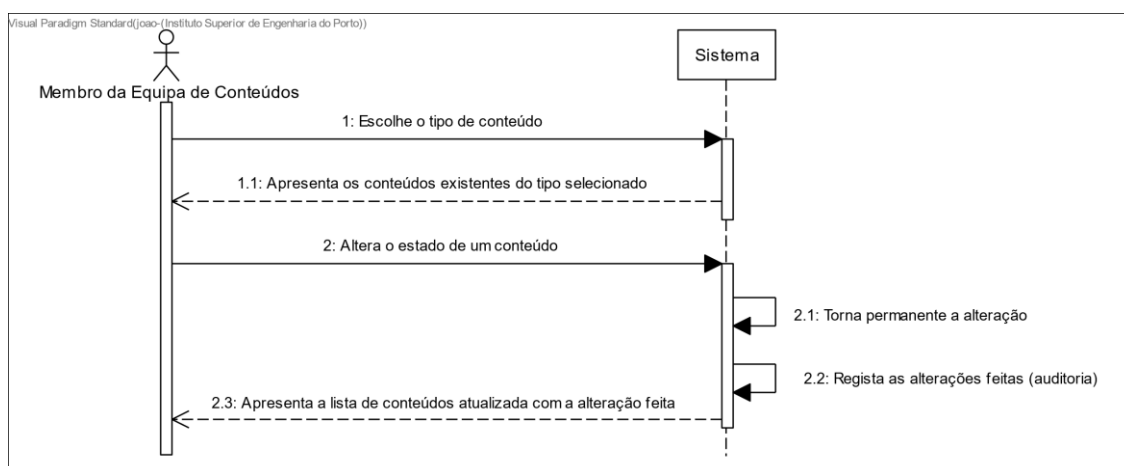


Figura 21 – Diagrama de sequência do sistema do UC25

6.4 Requisitos Não Funcionais

Os requisitos não funcionais correspondem às características de um sistema. No âmbito deste projeto, na presente secção, são apresentados os requisitos não funcionais, segundo o modelo FURPS+ (Lee, 2014).

Desta forma, no contexto deste projeto, na Tabela 28 são apresentados os requisitos não funcionais.

Tabela 28 – Requisitos não funcionais de acordo com modelo FURPS+

Classificação FURPS+	Requisito não funcional
Funcionalidade	Auditoria das alterações
Confiabilidade	Integridade dos dados
Desempenho	Pedidos de consulta de dados não podem ultrapassar os 500ms

Desempenho	Pedidos de alteração de dados não pode ultrapassar os 50ms
+ Desenho	Lógica de negócio tem de ser implementada na base de dados
+ Implementação	Tecnologias: PL/SQL e base de dados Oracle

No que diz respeito à funcionalidade, um dos objetivos identificados na secção 1.3, é manter registos automáticos de todas as alterações efetuadas nos conteúdos do ALERT®. Assim, é necessário garantir a auditoria de todas as alterações dos conteúdos. Para cada alteração realizada, tem de identificar-se:

- O utilizador responsável pela alteração;
- A data da alteração;
- O que foi alterado;
- A razão pela qual a alteração ocorre.

Relativamente à confiabilidade, é necessário garantir a integridade da lógica de negócios relativa aos diferentes conteúdos do ALERT®, de forma a que estes respeitem as regras de negócio definidas pela equipa de conteúdos.

No que concerne ao desempenho, tem de garantir-se que os pedidos de consulta de dados não podem ultrapassar os 500 milissegundos e os pedidos de alteração de dados (inserir e atualizar) não podem ultrapassar os 50 milissegundos.

No que diz respeito a restrições, foram identificadas restrições de desenho e de implementação. Relativamente à restrição de desenho, foi definido pela equipa de conteúdos que a lógica de negócio tem de ser implementada diretamente na base de dados. No âmbito da restrição de implementação, foi estabelecido pela equipa de conteúdos que os conteúdos do ALERT® são armazenados numa base de dados relacional Oracle e que o *back-end* é implementado com recurso à linguagem PL/SQL.

7 Design

No decorrer deste capítulo são abordados tópicos relacionados com o *design* da solução do projeto deste documento. Neste capítulo são apresentadas (1) alternativas de design, e identificada a alternativa que melhor se adequa ao projeto, apresentada a (2) vista de implantação do sistema, o (3) modelo relacional parcial da solução, e, por fim, é abordada a (4) estrutura da base de dados.

7.1 Alternativas de Design

Como abordado na secção 3.4, foram identificadas duas alternativas de design, no que diz respeito à arquitetura deste projeto: (1) implementar a lógica de negócio diretamente na base de dados e (2) implementar a lógica de negócio na camada de aplicação.

1. Uma das alternativas de design identificadas consiste em implementar a lógica de negócio diretamente na base de dados, ou seja, são desenvolvidas as funcionalidades que implementam a lógica de negócio diretamente na base de dados. Nesta alternativa, os dados relacionais relativos aos conteúdos do ALERT® encontram-se armazenados na base de dados e apenas podem ser geridos e acedidos a partir dessas funcionalidades, as quais se encontram implementadas também na base de dados.

Na Figura 22 é apresentado o diagrama de componentes referente a esta alternativa no qual se encontram representados os dois componentes principais – base de dados Oracle e *Content Server add-in*.

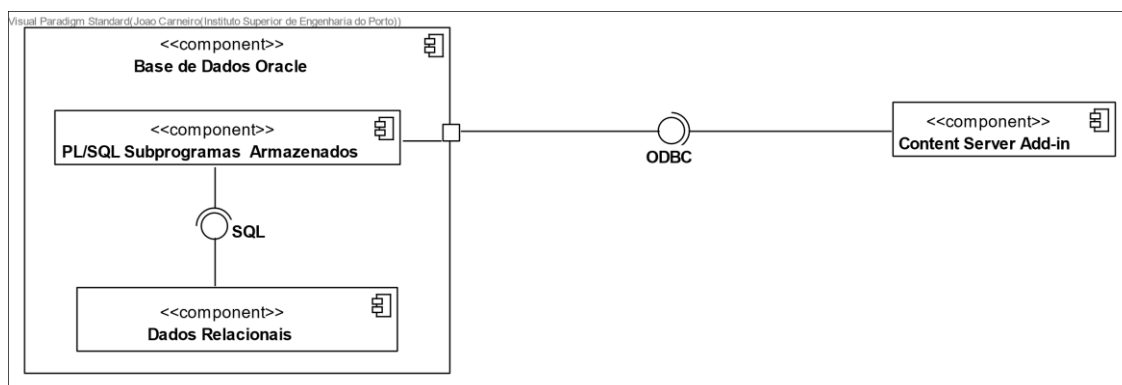


Figura 22 - Diagrama de componentes do sistema com a lógica de negócio implementada na base de dados

A base de dados Oracle, como referido anteriormente, armazena funcionalidades que refletem as regras de negócio definidas pela equipa de conteúdos. Estas funcionalidades consistem em subprogramas PL/SQL que são armazenados diretamente na base de dados e responsáveis por aceder e gerir os dados dos conteúdos do ALERT® a partir de instruções SQL. Assim, os dados relativos aos conteúdos apenas podem ser geridos a partir dos subprogramas PL/SQL armazenados na base de dados.

O *Content Server add-in*, como explicado na secção 2.4, fornece recursos adicionais à ferramenta Excel e permite aos membros da equipa de conteúdos fazer a gestão dos conteúdos recorrendo à ferramenta Excel. Nesta alternativa, o *Content Server add-in* invoca os subprogramas armazenados na base de dados Oracle a partir da interface ODBC⁵⁵.

2. A outra alternativa de design é implementar a lógica de negócio na camada de aplicação, tal como apresentada na Figura 23. Nesta alternativa a lógica de negócio é implementada numa aplicação, que é responsável pela implementação da lógica de negócio, e pode estruturar-se em diferentes camadas de computação lógica e física. A arquitetura de 3 camadas é uma arquitetura de desenvolvimento de software na qual são definidas três camadas distintas, nomeadamente:
 - Camada de apresentação;
 - Camada de aplicação;
 - Camada de dados.

Neste sentido, na Figura 23 é apresentado o diagrama de componentes do sistema, no qual estão representados três componentes – base de dados Oracle, *Content Server back-end API*⁵⁶ e *Content Server add-in*.

⁵⁵ ODBC “[...] fornece uma interface padrão que permite que uma aplicação possa aceder a diferentes fontes de dados.” (Oracle, s.d. f).

⁵⁶ Uma *Application Programming Interface (API)* “[...] é um conjunto de definições e protocolos para desenvolver e integrar software de uma aplicação.” [tradução livre do autor de (Red Hat, s.d.)].

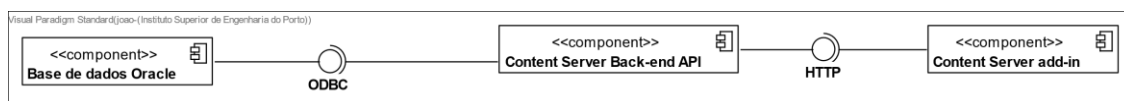


Figura 23 - Diagrama de componentes do sistema com a lógica de negócio implementada na camada de aplicação

A base de dados Oracle, que, segundo a arquitetura de três camadas se insere na camada de dados, é o componente cuja única responsabilidade é garantir a persistência dos dados relacionais dos conteúdos do ALERT®.

O componente *Content Server back-end API*, que, na perspetiva da arquitetura de três camadas se insere na camada de aplicação, é o componente responsável pelo processamento da informação de acordo com a lógica de negócio definida. De facto, é nesta camada que é implementada a lógica de negócio, sendo que esta comunica com a base de dados através da interface ODBC de forma a obter, inserir e alterar os dados relativos aos conteúdos do ALERT®.

Por fim, de acordo com a arquitetura de três camadas, o componente *Content Server add-in* insere-se na camada de apresentação. Este componente, como referido anteriormente, adiciona recursos à ferramenta Excel de forma a habilitar a gestão dos conteúdos do ALERT® aos membros da equipa de conteúdos. Nesta alternativa, este componente invoca as funcionalidades implementadas no componente *Content Server back-end API* através de pedidos *Hypertext Transfer Protocol* (HTTP)⁵⁷.

Na secção 3.4.3 é feita uma comparação detalhada entre as duas abordagens de design expondo os pontos fortes e fracos de cada uma das abordagens. No entanto, como referido na secção 3.4.3, a equipa apresenta como restrição de desenho a implementação da lógica de negócio diretamente na base de dados, alternativa representada na Figura 22. Esta restrição de desenho é um dos requisitos não funcionais descritos na secção 6.4. Assim, a alternativa de design escolhida para ser aplicada neste projeto, é a implementação da lógica de negócio diretamente na base de dados.

7.2 Visão de Implantação

Na Figura 24 é apresentado o diagrama de implantação do sistema do projeto do presente documento. Este sistema, apresenta um estilo de arquitetura cliente/servidor, em que se verifica que o sistema é segregado em duas aplicações distintas.

⁵⁷ HTTP “[...] é um protocolo de camada de aplicação [...] projetado para comunicação entre *web browsers* e servidores *web*, mas também pode ser usado para outros fins.” [tradução livre do autor de (Mozilla, 2020)].

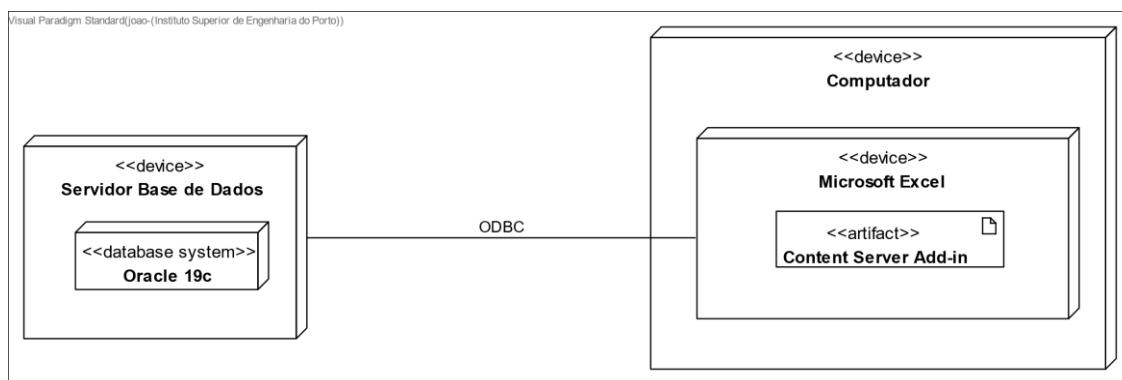


Figura 24 - Diagrama de implantação

Tal como se pode verificar na Figura 24, o cliente, que, neste caso, corresponde à aplicação Microsoft Excel, contém o *Content Server add-in* (abordado na secção 2.4). O *Content Server* adiciona um conjunto de funcionalidades que permite ao membro da equipa de conteúdos realizar a gestão dos conteúdos do ALERT® no Excel. Desta forma, o Excel faz pedidos ao servidor através da interface ODBC. O servidor, que, neste caso, corresponde à base de dados Oracle (versão 19c) armazena subprogramas PL/SQL que contêm toda a lógica de negócio responsável por receber os pedidos do cliente e retornar respostas aos pedidos.

7.3 Modelo Relacional

Como referido na secção 3.3.3, na qual se justifica a escolha, optou-se por recorrer a uma base de dados relacional, mais concretamente, uma base de dados Oracle na versão 19c. Apesar da eleição de uma base de dados relacional para armazenar os dados dos conteúdos, verifica-se que não existem muitos dados em comum entre as tabelas dos conteúdos. Assim, opta-se por expor um excerto do modelo relacional, representado na Figura 26, mais concretamente o modelo relacional do conteúdo exame.

O modelo relacional relativo à persistência dos dados dos exames do ALERT®, é constituído por 11 entidades, nomeadamente:

1. EXAM;
2. EXAM_GROUP_REL;
3. EXAM_GROUP;
4. EXAM_TYPE;
5. EXAM_BODY_STRUCTURE_REL;
6. EXAM_QUESTIONNAIRE;
7. BODY_STRUCTURE;
8. CORE_DATA.STATUS;
9. RESPONSE;
10. EXAM_CATEGORY;
11. QUESTIONNAIRE.

Tal como se pode verificar na Figura 26, em diferentes entidades, é possível identificar colunas que se encontram repetidas, entre as quais:

- ID_CONTENT – coluna que representa um identificador de um conteúdo cujo valor é gerado por uma ferramenta da autoria da ALERT e tem como finalidade identificar, inequivocamente, cada conteúdo;
- CODE_[NOME_ENTIDADE] – coluna que representa o código de tradução, em que “[NOME_ENTIDADE]” representa o nome de uma entidade, por exemplo, no caso da entidade exame o código de tradução seria CODE_EXAM. Este código permite relacionar um conteúdo com as diferentes traduções que este pode apresentar. Na Figura 25 é possível verificar que esta coluna é uma coluna virtual⁵⁸ e que o seu valor é determinado automaticamente segundo a seguinte expressão: [NOME_ENTIDADE].CODE_[NOME_ENTIDADE].ID_[NOME_ENTIDADE];
- INTERNAL_NAME - coluna que representa um identificador interno do conteúdo. Este identificador é um texto (*string*) único com um máximo de 30 caracteres que é utilizado pela equipa de conteúdos para conseguir identificar um conteúdo de uma forma mais simples e legível;
- ID_STATUS – coluna que representa um identificador do estado de um conteúdo, que pode, ao longo do tempo, apresentar diferentes estados;
- LAST_CHANGE – coluna que representa o *timestamp* da última alteração.

CODE_EXAM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	'EXAM.CODE_EXAM.' TO_CHAR('ID_EXAM')
-----------	-------------------------------------	--	--------------------------	---------------------------------------

Figura 25 – Coluna do código de tradução do conteúdo exame

⁵⁸ Uma coluna virtual no contexto de uma base de dados Oracle, consiste numa coluna cujo valor é determinado de forma automática, podendo usar valores de outras colunas da tabela.

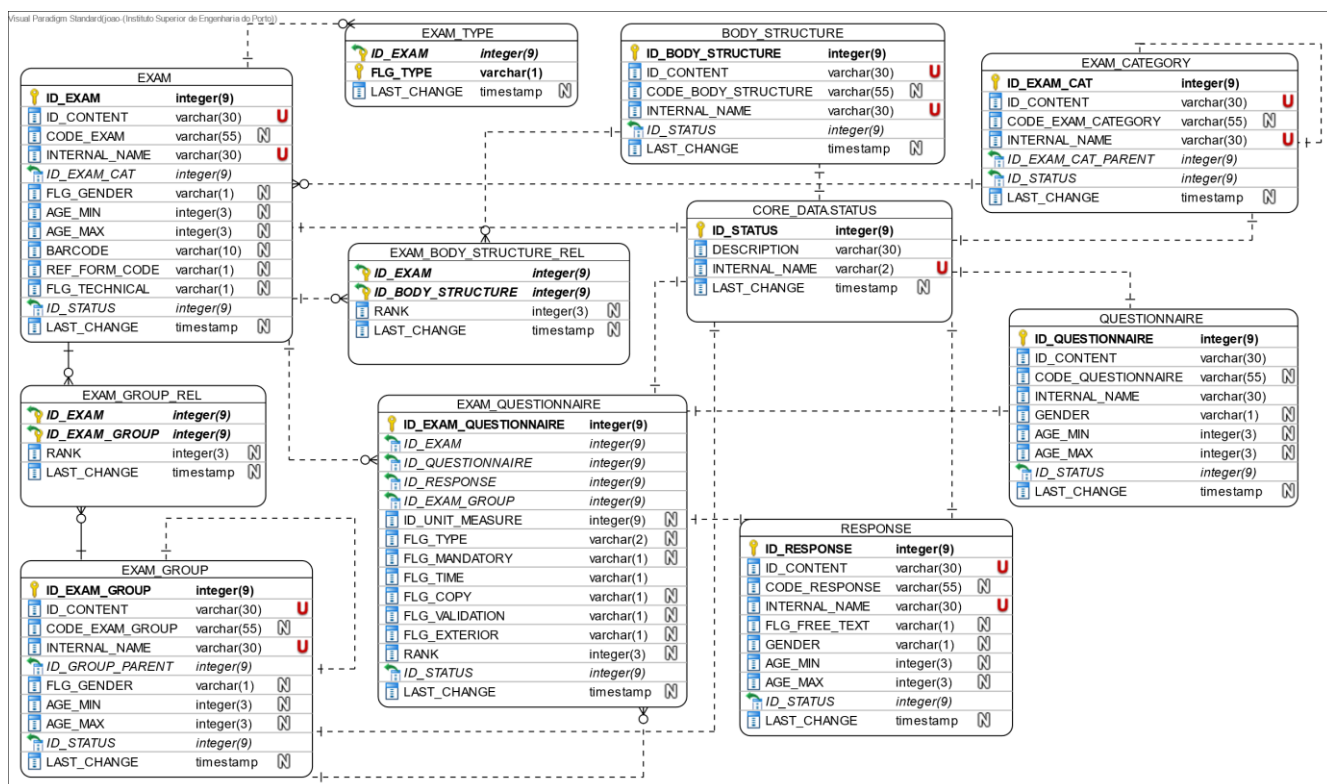


Figura 26 – Modelo relacional referente ao conteúdo exame

7.4 Base de Dados

Como referido na secção 3.3.3, a equipa de conteúdos estabeleceu como restrição o uso de base de dados Oracle para este projeto. Esta decisão prende-se com o facto da ALERT possuir licenças Oracle.

Assim, no contexto deste projeto, é criada uma base de dados Oracle para o armazenamento de subprogramas PL/SQL, os quais interagem com os dados relativos aos conteúdos do ALERT®, também armazenados na mesma base de dados Oracle. Neste âmbito, opta-se por estruturar esta base de dados recorrendo à criação de vários esquemas de base de dados (*database schemas*), por duas razões distintas – segurança e organização.

Relativamente à segurança, a utilização de esquemas de base de dados permite um controlo de acesso ao nível da base de dados, uma vez que, só podem ser acedidos os subprogramas que se espera que sejam acedidos. Desta forma, é garantida a segurança, na medida que existe um controlo de acesso, que resulta do facto de não ser possível inserir, editar ou aceder aos dados das tabelas diretamente.

Para além da segurança, a utilização de esquemas de base de dados permite uma separação lógica dos objetos da base de dados numa base de dados física, o que torna mais fácil a gestão da base de dados.

Na Figura 27 é apresentado um diagrama que representa a estrutura lógica da base de dados Oracle.

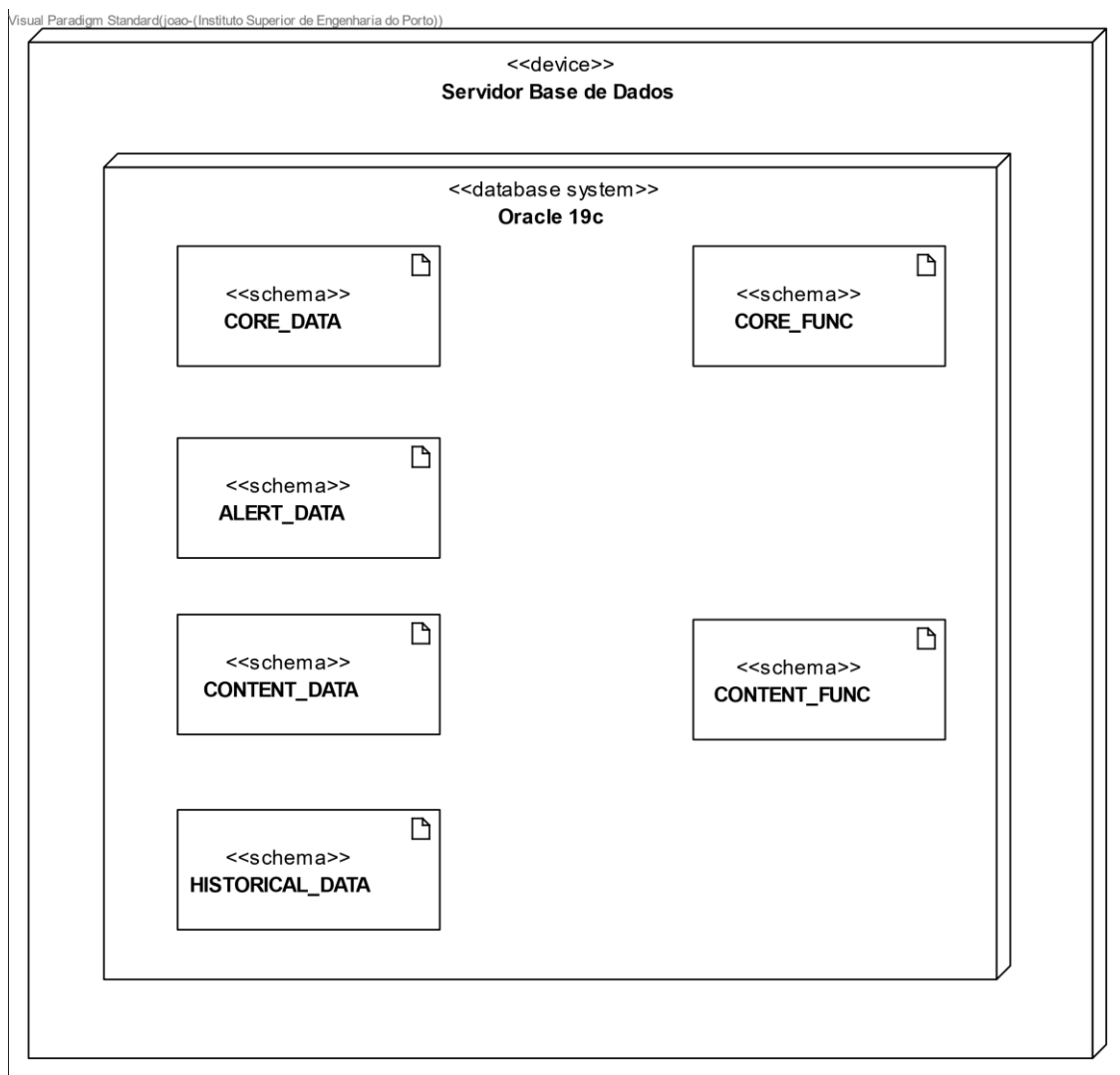


Figura 27 - Vista da estrutura lógica da base de dados

Como é possível verificar na Figura 27, a base de dados é estruturada logicamente através de vários esquemas de base de dados, nomeadamente:

- CORE_DATA – esquema responsável por armazenar objetos de base de dados relacionados com dados necessários a todos os esquemas;
- CORE_FUNC - esquema em que são desenvolvidos e armazenados subprogramas PL/SQL genéricos;
- ALERT_DATA – esquema responsável por armazenar objetos de base de dados relacionados com conteúdos específicos do ALERT®;
- CONTENT_DATA – esquema responsável por armazenar objetos de base de dados relacionados com os conteúdos clínicos e não clínicos do ALERT®;

- CONTENT_FUNC – esquema em que são desenvolvidos e armazenados subprogramas PL/SQL, responsáveis pela gestão dos conteúdos, onde é implementada a lógica de negócio;
- HISTORICAL_DATA – esquema responsável por armazenar objetos de base de dados relacionados com o histórico dos conteúdos.

Os esquemas de base de dados apresentados na Figura 27 estão estruturados de forma a que um esquema apenas possa aceder ao esquema do seu nível (na figura corresponde ao esquema que se encontra ao lado de um esquema) e aos esquemas que se encontram nos níveis acima de si mesmo. Assim, a título de exemplo, o esquema CONTENT_FUNC tem permissões para aceder ao esquema CONTENT_DATA e aos esquemas de nível superior – ALERT_DATA, CORE_DATA e CORE_FUNC. No entanto, o esquema CORE_FUNC apenas tem permissão para aceder ao esquema CORE_DATA.

8 Implementação

Neste capítulo são apresentados detalhes técnicos da implementação da solução deste projeto. Assim, inicialmente, é abordada a metodologia adotada no desenvolvimento da solução, de seguida é apresentada a estrutura do código e, por fim, apresentam-se detalhes relativos à implementação de determinados casos de uso do projeto.

8.1 Metodologia

Numa fase inicial (capítulo 6 – Engenharia de Requisitos), definiu-se, em conjunto com a responsável da equipa de conteúdos, os requisitos deste projeto, tanto a nível funcional, tendo sido estabelecidos os tipos de conteúdos do ALERT® a ter em consideração, como não funcional. Adicionalmente, a responsável pela equipa de conteúdos definiu, também, a prioridade dos requisitos.

Neste contexto, na implementação, adotou-se uma forma de trabalho sequencial, tendo em conta a priorização definida, na qual após o término de um requisito funcional, referente a um tipo de conteúdo, se avança para o seguinte requisito.

Para cada tipo de conteúdo, definiu-se, em conjunto com a responsável da equipa de conteúdos, um *workflow*, no que diz respeito à concretização do respetivo requisito. Neste sentido, primeiramente, realizou-se uma análise ao modelo de dados atual, referente ao tipo de conteúdo e ao respetivo ficheiro de configuração, com o objetivo de identificar aspetos que pudessem ser melhorados/reformulados e idealizou-se um novo modelo de dados. De seguida, em diálogo com a responsável pela equipa de conteúdos, validou-se este novo modelo de dados e, após esta validação, o primeiro passo consistiu na criação dos objetos de base de dados - tabelas e sequências – do tipo de conteúdo. Por fim, desenvolveram-se as respetivas funcionalidades que permitem a total gestão dos dados das tabelas do tipo de conteúdo, sem necessitar de um acesso direto à tabela.

8.2 Estrutura do Código

Como referido na secção 3.4.3, decidiu-se implementar as funcionalidades deste projeto diretamente na base de dados. Assim, as funcionalidades que representam as regras de negócio que permitem gerir os dados referentes aos conteúdos, foram implementadas com recurso à linguagem de programação PL/SQL no *schema* CONTENT_FUNC, tal como descrito na secção 7.4. Neste sentido, para cada conteúdo, criou-se um *package*, sendo que um *package*, do ponto de vista das bases de dados Oracle, agrupa logicamente variáveis e subprogramas (blocos de código escrito em PL/SQL) relacionados.

Relativamente à nomenclatura dos *packages*, esta definiu-se em conjunto com a equipa de conteúdos e consiste na seguinte expressão:

PK_[NOME_DA_TABELA]

, onde “[NOME_DA_TABELA]” corresponde ao nome da tabela de um conteúdo.

8.3 Caso de Uso

Nesta secção é apresentada informação relativa à implementação do caso de uso referente à gestão de conteúdos do tipo exame, previamente representado na Figura 15. Como referido na secção 6.3, os casos de uso de gestão de conteúdos foram simplificados, de forma a tornar mais simples a visualização dos requisitos funcionais. Assim, a gestão de conteúdos envolve três processos – criação, alteração e visualização de conteúdos.

Na Figura 28 é representado o modelo de domínio do conteúdo do tipo exame.

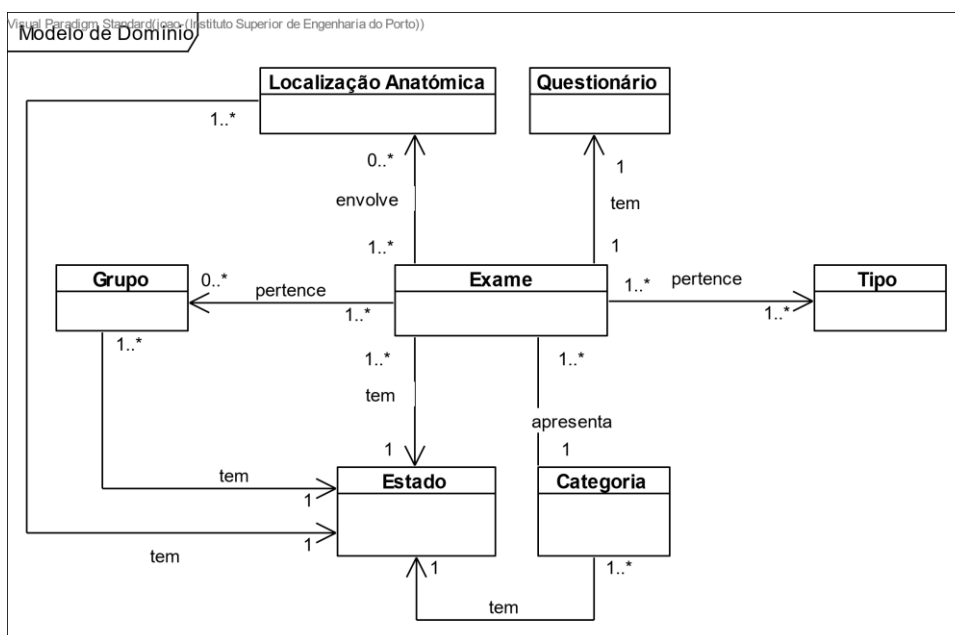


Figura 28 – Modelo de domínio do conteúdo do tipo exame

8.3.1 Criação de um conteúdo do tipo exame

No que diz respeito ao processo de criação de um conteúdo do tipo exame, este consiste na criação de um conteúdo em que o utilizador insere um conjunto de dados necessários para a criação de um exame.

Como referido na secção 2.4, o *Content Server*, cujo desenvolvimento é da responsabilidade de outro membro da ALERT, caracteriza-se como um *add-in*, implementado na ferramenta Excel, que consiste na componente gráfica deste projeto. Apesar do *Content Server* ser da autoria de outro elemento, considera-se fulcral a sua representação na Figura 29, para a total compreensão do projeto. Assim, na Figura 29, representam-se todas as interações entre um membro da equipa de conteúdos com o componente *Content Server*, que invoca as funcionalidades desenvolvidas no âmbito deste projeto, com o objetivo de criar um conteúdo do tipo exame.

Na Figura 29, verifica-se a participação de três intervenientes no diagrama de sequência, nomeadamente:

- CONTENTE_SERVER: componente da responsabilidade de outro membro da ALERT, que invoca funcionalidades que foram desenvolvidas no âmbito deste projeto. Trata-se da interface gráfica, através da qual um membro da equipa de conteúdos pode, de uma forma simplificada, gerir os conteúdos do ALERT®;
- CONTENT_FUNC.PK_EXAM: package armazenado diretamente na base de dados Oracle no *schema* CONTENT_FUNC, que agrupa logicamente variáveis e subprogramas relacionados (blocos de código escrito em PL/SQL), que permitem a gestão dos conteúdos do tipo exame;
- CONTENT_FUNC.PK_TRANSLATION: package armazenado diretamente na base de dados Oracle no *schema* CONTENT_FUNC, que agrupa logicamente variáveis e subprogramas relacionados, que permitem a gestão das traduções dos diferentes tipos de conteúdos.

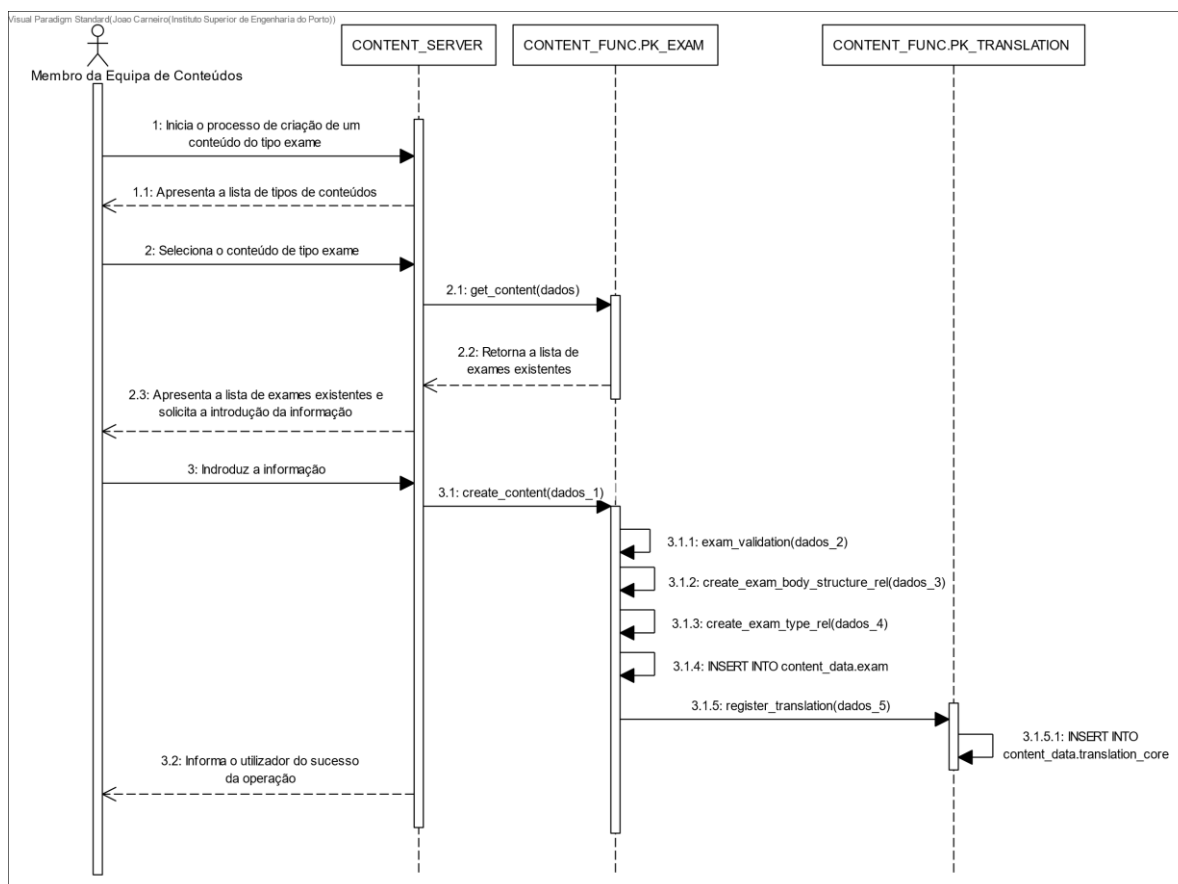


Figura 29 - Diagrama de sequência do caso de uso de criação de um conteúdo do tipo exame

No diagrama de sequência do caso de uso relativo à criação de um conteúdo do tipo exame, representado na Figura 29, verifica-se que o ator - membro da equipa de conteúdos - interage com o componente CONTENT_SERVER, o componente gráfico. Consta-se que, inicialmente, o ator interage com o componente CONTENT_SERVER, de forma a dar início ao processo de criação de conteúdo do tipo exame. Desta forma, o CONTENT_SERVER apresenta a lista de tipos de conteúdos, que podem ser geridos pelo ator. Neste caso, o ator opta pelo conteúdo do tipo exame, de seguida, o CONTENT_SERVER devolve a lista de conteúdos do tipo exame e o ator preenche os dados que permitem a criação de um conteúdo deste tipo. Na mensagem 3.1 do diagrama de sequência representado na Figura 29, verifica-se que o componente CONTENT_SERVER invoca um subprograma – *create_content* – que se encontra armazenado diretamente na base de dados, mais concretamente no *package* PK_EXAM no *schema* CONTENT_FUNC. Tal como referido na secção 7.4, é, neste *schema* – CONTENT_FUNC - que são desenvolvidos e armazenados subprogramas PL/SQL, nos quais são implementadas as regras de negócio que permitem a gestão dos conteúdos do ALERT®.

O subprograma invocado na mensagem 3.1 – *create_content* – é um subprograma do tipo *procedure* que recebe a informação introduzida pelo ator, de forma a criar conteúdos do tipo exame. Na mensagem 3.1 do diagrama de sequência representado na Figura 29 –

create_content(dados_1) -, optou-se por utilizar “dados_1”, que representa os parâmetros de entrada do *procedure*, de forma a simplificar o diagrama. No Extrato Código 1, encontra-se representada a especificação do *procedure create_content*.

```
PROCEDURE create_content(i_id_content      IN t_id_content,
                        i_internal_name    IN t_internal_name,
                        i_exam_category    IN t_id_content DEFAULT NULL,
                        i_body_structure    IN t_description DEFAULT NULL,
                        i_flg_img           IN t_flg DEFAULT NULL,
                        i_flg_pregnant      IN t_flg DEFAULT NULL,
                        i_flg_other         IN t_flg DEFAULT NULL,
                        i_flg_intervention  IN t_flg DEFAULT NULL,
                        i_flg_gender        IN t_gender DEFAULT NULL,
                        i_age_min           IN t_age DEFAULT NULL,
                        i_age_max           IN t_age DEFAULT NULL,
                        i_barcode           IN t_barcode DEFAULT NULL,
                        i_ref_form_code     IN t_ref_form_code DEFAULT NULL,
                        i_flg_technical     IN t_technical DEFAULT NULL,
                        i_id_status        IN t_id_status,
                        o_id_exam         OUT t_id_exam);
```

Extrato Código 1 – Especificação do *procedure create_content*

Tal como se pode verificar no Extrato Código 2, no *procedure create_content* é invocado outro *procedure* – *exam_validation* - que se encontra no mesmo *package* – PK_EXAM – no qual são feitas verificações a alguns dos dados introduzidos pelo ator. Adicionalmente, se nos dados de criação de um exame for definida uma categoria, procede-se à identificação do id (identificador) da mesma.

```

PROCEDURE create_content(i_id_content          IN t_id_content,
                        i_internal_name       IN t_internal_name,
                        i_exam_category      IN t_id_content DEFAULT NULL,
                        i_body_structure      IN t_description DEFAULT NULL,
                        i_flg_img            IN t_flg DEFAULT NULL,
                        i_flg_pregnant       IN t_flg DEFAULT NULL,
                        i_flg_other          IN t_flg DEFAULT NULL,
                        i_flg_intervention    IN t_flg DEFAULT NULL,
                        i_flg_gender         IN t_gender DEFAULT NULL,
                        i_age_min            IN t_age DEFAULT NULL,
                        i_age_max            IN t_age DEFAULT NULL,
                        i_barcode            IN t_barcode DEFAULT NULL,
                        i_ref_form_code      IN t_ref_form_code DEFAULT NULL,
                        i_flg_technical      IN t_technical DEFAULT NULL,
                        i_id_status          IN t_id_status,
                        o_id_exam           OUT t_id_exam) IS
    l_code_description t_code_exam; -- Store the translation code
    l_id_exam_category t_id_exam_cat; -- Id of exam category
BEGIN
    -- Exam validations
    exam_validation(i_id_content          => i_id_content,
                   i_internal_name       => i_internal_name,
                   i_age_min             => i_age_min,
                   i_age_max             => i_age_max,
                   i_exam_category       => i_exam_category,
                   i_barcode             => i_barcode,
                   o_id_exam_category    => l_id_exam_category);

```

Extrato Código 2 – Invocação do *procedure exam_validation*

No *procedure exam_validation* efetuam-se diversas verificações que permitem avaliar a validade dos dados introduzidos pelo ator. Caso os dados introduzidos não respeitem alguma das regras de validação definidas, a transação não ocorre e é retornada uma exceção a informar qual a regra que não foi cumprida. Assim, este *procedure* verifica as seguintes informações introduzidas:

- *i_id_content* – verifica se o valor não se encontra vazio e se não existem espaços no início e fim do valor introduzido;
- *i_internal_name* – verifica se o valor não se encontra vazio e se não existem espaços no início e fim do valor introduzido. Adicionalmente, através de uma expressão regular, verifica se o valor apenas contém letras (maiúsculas), números e o carácter ‘_’;
- *i_barcode* – verifica se o valor não se encontra vazio e se não existem espaços no início e fim do valor introduzido;
- *i_age_min* e *i_age_max* – verifica se o valor de “*i_age_min*” é de facto menor que o valor de “*i_age_max*”.

O parâmetro “*i_exam_category*” é opcional na criação do exame e representa o *internal_name* ou *id_content* da categoria de um exame. Assim, caso o valor de “*i_exam_category*” seja diferente de *NULL*, o *procedure exam_validation* identifica o id da respetiva categoria de exame introduzida pelo utilizador. Desta forma, implementou-se um *procedure* genérico no *schema* *CONTENT_FUNC*, mais concretamente no *package* *PK_UTILS*, que permite identificar o id de um

conteúdo através do seu *id_content* ou *internal_name*. No Extrato Código 3, encontra-se representado o excerto de código, no qual é feita a invocação do subprograma, de forma a determinar o id da categoria do exame. Caso não exista nenhuma categoria de exame com o *internal_name* ou *id_content* introduzido pelo ator, é retornada uma exceção a informar do sucedido e a transação não ocorre.

```
pk_utils.get_content_type_id_by_internal_name_or_content_id(i_content_type_name => 'exam_category',
                                                           i_content_type_table => 'content_data.exam_category',
                                                           i_id_content_type   => 'id_exam_cat',
                                                           i_input           => i_exam_category,
                                                           o_id             => o_id_exam_category);
```

Extrato Código 3 – Invocação do *procedure get_content_type_id_by_internal_name_or_content_id*

Após certificar que os dados introduzidos são válidos, invoca-se dois *procedures* - *create_exam_body_structure_rel* e *create_exam_type_rel* – cujo excerto de código se encontra representado no Extrato Código 4.

```
create_exam_body_structure_rel(i_body_structure => i_body_structure,
                              i_id_exam       => o_id_exam,
                              i_is_update     => false);

create_exam_type_rel(i_id_exam      => o_id_exam,
                    i_flg_img       => i_flg_img,
                    i_flg_other     => i_flg_other,
                    i_flg_pregnant  => i_flg_pregnant,
                    i_flg_intervention => i_flg_intervention,
                    i_is_update     => false);
```

Extrato Código 4 – Invocação dos *procedures create_exam_body_structure_rel* e *create_exam_type_rel*

Tal como se pode verificar no modelo de domínio relativo ao conteúdo do tipo exame, representado na Figura 28, constata-se que a criação de um exame pode, opcionalmente, envolver a definição de uma relação, ou mais relações, com diferentes localizações anatómicas (*body structure*). Assim, o *procedure create_exam_body_structure_rel*, apenas é invocado quando o parâmetro “*i_body_structure*” não é *NULL*, e permite o estabelecimento de relações entre um exame e diferentes localizações anatómicas. O *procedure create_exam_type_rel*, por sua vez, permite a criação de uma relação entre um exame e tipos de exame.

Tal como se pode verificar na Figura 28, um exame pode ter um, ou mais, tipos de exame. Neste sentido, atualmente, existem quatro tipos de exame, nomeadamente:

- Imagem – *i_flg_img*;
- Grávida – *i_flg_pregnant*;
- Procedimento – *i_flg_intervention*;
- Outro – *i_flg_other*.

Adicionalmente, é criado o exame, sendo os seus dados inseridos na respetiva tabela - *exam* - que se encontra armazenada no *schema* CONTENT_DATA, tal como é possível verificar no Extrato Código 5.

```
-- Insert into table
INSERT INTO content_data.exam
  (id_exam,
   id_content,
   internal_name,
   id_exam_cat,
   flg_gender,
   age_min,
   age_max,
   barcode,
   ref_form_code,
   flg_technical,
   id_status,
   last_change)
VALUES
  (o_id_exam,
   i_id_content,
   i_internal_name,
   l_id_exam_category,
   i_flg_gender,
   i_age_min,
   i_age_max,
   i_barcode,
   i_ref_form_code,
   i_flg_technical,
   i_id_status,
   current_timestamp)
RETURNING code_exam INTO l_code_description;
```

Extrato Código 5 – Inserção dos dados relativos a um exame na tabela *exam*

Por fim, regista-se a tradução através do *procedure* - *register_translation* - que se encontra no *package* PK_TRANSLATION. Para este registo é utilizado o valor da variável *l_code_description* que é obtido no momento da inserção dos dados do exame na respetiva tabela. Esta variável contém o valor da coluna *code_exam*, sendo esta, uma coluna virtual, ou seja, cujo valor é determinado automaticamente e permite relacionar um conteúdo com as diferentes traduções que este pode apresentar. A invocação deste *procedure* encontra-se representado no Extrato Código 6.

```
pk_translation.register_translation(i_code_translation => l_code_description,
                                   i_id_status         => i_id_status);
```

Extrato Código 6 – Invocação do *procedure* *register_translation*

8.3.2 Visualização dos conteúdos do tipo exame

O processo de visualização de conteúdos do tipo exame, representado na Figura 30, apresenta todas as interações entre um membro da equipa de conteúdos com o componente *Content Server*. Neste âmbito, o componente *Content Server*, invoca as funcionalidades desenvolvidas no âmbito deste projeto, com a finalidade de retornar todos os conteúdos do tipo exame.

Na Figura 30, constata-se a participação de dois intervenientes no diagrama de sequência, nomeadamente:

- CONTENT_SERVER: componente da responsabilidade de outro membro da ALERT, que invoca funcionalidades que foram desenvolvidas no âmbito deste projeto. Trata-se da interface gráfica, através da qual um membro da equipa de conteúdos pode, de uma forma simplificada, gerir os conteúdos do ALERT®;
- CONTENT_FUNC.PK_EXAM: package armazenado diretamente na base de dados Oracle no *schema* CONTENT_FUNC, que agrupa logicamente variáveis e subprogramas relacionados (blocos de código escrito em PL/SQL), que permitem a gestão dos conteúdos do tipo exame.

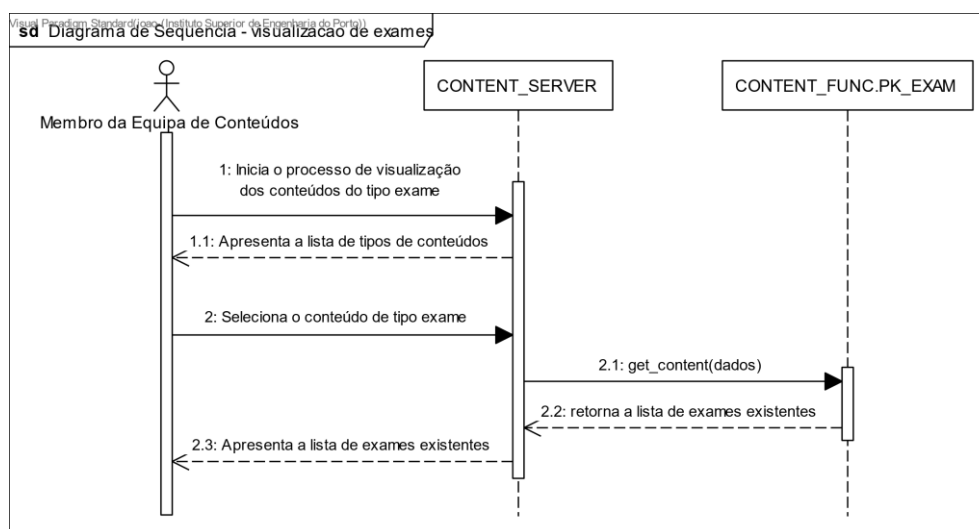


Figura 30 - Diagrama de sequência do caso de visualização dos conteúdos do tipo exame

No diagrama de sequência do caso de uso relativo à visualização dos conteúdos do tipo exame, representado na Figura 30, verifica-se que o ator - membro da equipa de conteúdos - interage com o componente CONTENT_SERVER. Constata-se que, inicialmente, o membro da equipa de conteúdos interage com o componente CONTENT_SERVER de forma a dar início ao processo de visualização dos conteúdos do tipo exame. Desta forma, o CONTENT_SERVER apresenta a lista de tipos de conteúdos que podem ser geridos pelo membro da equipa de conteúdos e o ator seleciona, neste caso, o tipo de conteúdo exame. Assim, o CONTENT_SERVER invoca a *function* *get_content*, cujo código da sua especificação se encontra representado no Extrato Código 7.

```

FUNCTION get_content(i_id_exam    IN t_id_exam DEFAULT NULL,
                   i_ids_exam    IN table_number DEFAULT table_number(NULL),
                   i_id_content  IN t_id_content DEFAULT NULL)
RETURN table_exam
PIPELINED;

```

Extrato Código 7 – Especificação da *function get_content*

A *function get_content* é um subprograma que admite a possibilidade de três situações de retorno de dados, nomeadamente:

- Todos os conteúdos do tipo exame – quando os três parâmetros – *i_id_exam*, *i_ids_exam* e *i_id_content* - da *function get_content* são *NULL*;
- Um conteúdo do tipo exame – o ator define qual o id do exame cujos dados pretende obter (através do parâmetro *i_id_exam*) ou, então, o ator define qual o identificador do conteúdo do exame através do parâmetro *i_id_content*;
- Uma lista de conteúdos do tipo exame – o ator define um conjunto de identificadores de exames cujos dados pretende visualizar através do parâmetro *i_ids_exam*.

No Extrato Código 8 é demonstrada a lógica implementada para a obtenção da lista de conteúdos do tipo exame a retornar e, neste sentido, optou-se pelo recurso à utilização de cursores.

```

IF i_ids_exam IS NULL THEN
OPEN l_cursor FOR
SELECT *
FROM content_data.exam e
WHERE coalesce(i_id_exam, e.id_exam) = e.id_exam
AND coalesce(i_id_content, e.id_content) = e.id_content;

ELSE

OPEN l_cursor FOR
WITH ids_exam AS
(SELECT
column_value
FROM TABLE(i_ids_exam) t)
SELECT *
FROM content_data.exam e
WHERE coalesce(i_id_content, e.id_content) = e.id_content
AND EXISTS
(SELECT 1
FROM ids_exam f1
WHERE coalesce(f1.column_value, e.id_exam) = e.id_exam);

END IF;

```

Extrato Código 8 – Implementação da lógica para obtenção da lista de exames a retornar

8.4 Configuração de Conteúdo

Como já referido anteriormente, na secção 2.1 cada conteúdo do ALERT pode ser configurado tendo em conta diferentes contextos. Assim, atualmente, existem seis contextos, nomeadamente:

- Idioma;
- Mercado;
- Software;
- Serviço clínico;
- Queixa;
- Versão (catálogo).

O contexto idioma está associado às diferentes traduções que um conteúdo pode apresentar. Verifica-se que existe a necessidade de criar conteúdos específicos para um determinado idioma, pelo que, através deste contexto, é possível configurar um determinado conteúdo para que este apresente tradução para um determinado idioma ou para um conjunto de idiomas.

Adicionalmente, a equipa constatou que existe necessidade de criar conteúdos específicos para determinados mercados, em que o ALERT® se encontra implementado. Desta forma, através deste contexto é possível que determinados conteúdos sejam configurados de forma que estes estejam disponíveis apenas para determinados mercados.

Como referido na secção 1.1.1, o ALERT® consiste num software que pode apresentar diferentes configurações no que diz respeito aos produtos (softwares) ALERT®. No site da ALERT (ALERT Life Sciences Computing, s.d. d) verifica-se que existe um conjunto de produtos para diferentes situações, nomeadamente:

- Hospitais - Figura 31;
- Cuidados primários - Figura 32;
- Clínicas privadas - Figura 33;
- Outras – Figura 34.

LISTA DE PRODUTOS ALERT® PARA HOSPITAIS

ALERT® PAPER FREE HOSPITAL composto por:

- ALERT® ADMISSION, DISCHARGE AND TRANSFER (ADT)
- ALERT® SCHEDULER
- ALERT® WAITING LINE
- ALERT® WAITING ROOM (WR)
- ALERT® ONLINE APPOINTMENT WITH VIDEOCONFERENCING
- ALERT® TRIAGE
- ALERT® EMERGENCY DEPARTMENT INFORMATION SYSTEM (EDIS)
- ALERT® OBSERVATION ROOM (OBS)
- ALERT® AMBULATORY (OUTPATIENT)
- ALERT® INPATIENT
- ALERT® OPERATING ROOM INFORMATION SYSTEM (ORIS)
- ALERT® PATIENT DATA MANAGEMENT SYSTEM (PDMS)
- ALERT® PRESCRIPTION
- ALERT® PHARMACY
- ALERT® MENTAL HEALTH
- ALERT® PSYCHOLOGY
- ALERT® REHABILITATION
- ALERT® HOME HEALTH CARE
- ALERT® HOME HEALTH CARE WITH VIDEOCONFERENCING
- ALERT® SOCIAL (SOCIAL WORKER)
- ALERT® REFERRAL
- ALERT® STANDARD CONTENT
- ALERT® DECISION TREES
- ALERT® CODING
- ALERT® BUSINESS INTELLIGENCE
- ALERT® PLANNING SYSTEM
- ALERT® eLIBRARY
- ALERT® eLEARNING
- INTER-ALERT® (INTERFACE PLATFORM)
- ALERT® CONFIGURATION TOOL
- ALERT® AUDIT TRAIL
- ALERT® CRISIS MACHINE
- ALERT® NETWORK OPERATIONS CENTER (NOC)

O HEALTH BOX® da ALERT, que inclui:

- ALERT® HEALTH INFORMATION EXCHANGE (HIE)
- ALERT® PERSONAL HEALTH RECORD (PHR)
- INTER-ALERT®

EM PARCERIA COM TERCEIROS:

- LABORATORY INFORMATION SYSTEM (LIS)
- RADIOLOGY INFORMATION SYSTEM (RIS)
- PACS

Confidencial © ALERT - Todos os direitos reservados.



Figura 31 – Lista de produtos ALERT® para hospitais. Retirado de (ALERT Life Sciences Computing, s.d. d).

LISTA DE PRODUTOS ALERT® PARA CUIDADOS PRIMÁRIOS

ALERT® PRIMARY CARE, que inclui:

- ALERT® ADMISSION, DISCHARGE AND TRANSFER (ADT)
- ALERT® SCHEDULER
- ALERT® ONLINE APPOINTMENT WITH VIDEOCONFERENCING
- ALERT® PRESCRIPTION
- ALERT® PHARMACY
- ALERT® SOCIAL (SOCIAL WORKER)
- ALERT® REFERRAL
- ALERT® STANDARD CONTENT
- ALERT® DECISION TREES
- ALERT® CODING
- ALERT® BUSINESS INTELLIGENCE
- ALERT® PLANNING SYSTEM
- ALERT® eLIBRARY
- ALERT® eLEARNING
- INTER-ALERT® (INTERFACE PLATFORM)
- ALERT® CONFIGURATION TOOL
- ALERT® AUDIT TRAIL
- ALERT® CRISIS MACHINE
- ALERT® NETWORK OPERATIONS CENTER (NOC)

O HEALTH BOX® da ALERT, que inclui:

- ALERT® HEALTH INFORMATION EXCHANGE (HIE)
- ALERT® PERSONAL HEALTH RECORD (PHR)
- INTER-ALERT®

EM PARCERIA COM TERCEIROS:

- LABORATORY INFORMATION SYSTEM (LIS)
- RADIOLOGY INFORMATION SYSTEM (RIS)
- PACS

Confidencial © ALERT - Todos os direitos reservados.



Figura 32 - Lista de produtos ALERT® para cuidados primários. Retirado de (ALERT Life Sciences Computing, s.d. d).

LISTA DE PRODUTOS ALERT® PARA CLÍNICAS PRIVADAS

ALERT® PRIVATE PRACTICE, que inclui:

- ALERT® ADMISSION, DISCHARGE AND TRANSFER (ADT)
- ALERT® SCHEDULER
- ALERT® ONLINE APPOINTMENT WITH VIDEOCONFERENCING
- ALERT® PRESCRIPTION
- ALERT® PHARMACY
- ALERT® SOCIAL (SOCIAL WORKER)
- ALERT® REFERRAL
- ALERT® STANDARD CONTENT
- ALERT® DECISION TREES
- ALERT® CODING
- ALERT® BUSINESS INTELLIGENCE
- ALERT® PLANNING SYSTEM
- ALERT® eLIBRARY
- ALERT® eLEARNING
- INTER-ALERT® (INTERFACE PLATFORM)
- ALERT® CONFIGURATION TOOL
- ALERT® AUDIT TRAIL
- ALERT® CRISIS MACHINE
- ALERT® NETWORK OPERATIONS CENTER (NOC)

O HEALTH BOX® da ALERT, que inclui:

- ALERT® HEALTH INFORMATION EXCHANGE (HIE)
- ALERT® PERSONAL HEALTH RECORD (PHR)
- INTER-ALERT®

EM PARCERIA COM TERCEIROS:

- LABORATORY INFORMATION SYSTEM (LIS)
- RADIOLOGY INFORMATION SYTEM (RIS)
- PACS

Confidencial © ALERT - Todos os direitos reservados.



Figura 33 – Lista de produtos ALERT® para clínicas privadas. Retirado de (ALERT Life Sciences Computing, s.d. d).

OUTROS PRODUTOS ALERT®

ALERT® ONLINE

- ALERT® ONLINE APPOINTMENT WITH VIDEOCONFERENCING
- HEALTH BOX® da ALERT
- ALERT® SELF TRIAGE
- ALERT® SALVAGE SITE (RECUPERAÇÃO DE INFORMAÇÃO CLÍNICA DIGITAL)
- ALERT® MIRROR SITE (ASSOCIADO AO Nº DO PROCESSO CLÍNICO)
- DASHBOARD COVID-19 , MEDIDAS DE CONTROLE POR PAÍS, ESTIMATIVAS DE RECURSOS CHAVE, DIÁRIO PANDEMIA, BREVES RELATÓRIOS POR PAÍS
- NOTÍCIAS DE SAÚDE

ALERT® STUDENT

Confidencial © ALERT - Todos os direitos reservados.



Figura 34 - Lista de outros produtos ALERT®. Retirado de (ALERT Life Sciences Computing, s.d. d).

Tendo em conta a vasta gama de produtos ALERT®, a equipa de conteúdos constatou a necessidade de configurar conteúdos específicos para produtos (softwares) ALERT®. Assim, através do contexto software, é possível configurar conteúdos específicos para determinados softwares.

Adicionalmente, a equipa de conteúdos aferiu a necessidade de configurar conteúdos para determinados serviços clínicos e diferentes queixas. No que diz respeito aos serviços clínicos, a necessidade de configuração dos conteúdos deve-se ao facto de as instituições de cuidados médicos apresentarem diferentes serviços clínicos. No que diz respeito às queixas, constatou-se a necessidade de configurar determinados conteúdos para que estes estejam disponíveis apenas para determinadas queixas dos pacientes.

Por fim, a equipa de conteúdos constatou a necessidade de configurar os conteúdos de forma que estes estejam disponíveis para determinadas versões do software ALERT®.

Assim, de forma a permitir a configuração dos conteúdos, foram criadas três tabelas auxiliares no *schema* CORE_DATA, nomeadamente:

- CONTEXT_TYPE – tabela onde são definidos os vários contextos;
- CONTENT_TYPE – tabela em que são registados os vários tipos de conteúdo;
- CONTENT_TYPE_CONTEXT – tabela onde são registadas as diversas configurações de um tipo de conteúdo, ou seja, as relações entre tipos de conteúdo e contextos.

Adicionalmente, no *schema* CONTENT_DATA, foram criadas as seguintes tabelas onde são estabelecidas as relações entre os diferentes conteúdos e os contextos:

- CONTEXT_CATALOGUE – tabela onde são registadas as associações entre o identificador de um tipo de conteúdo, o identificador do conteúdo e o identificador do catálogo;
- CONTEXT_CLINICAL_SERVICE – tabela onde são registadas as associações entre o identificador de um tipo de conteúdo, o identificador do conteúdo e o identificador do serviço clínico;
- CONTEXT_COMPLAINT – tabela onde são registadas as associações entre o identificador de um tipo de conteúdo, o identificador do conteúdo e o identificador da queixa;
- CONTEXT_MARKET – tabela onde são registadas as associações entre o identificador de um tipo de conteúdo, o identificador do conteúdo e o identificador do mercado;
- CONTEXT_SOFTWARE – tabela onde são registadas as associações entre o identificador de um tipo de conteúdo, o identificador do conteúdo e o identificador do software.

De forma a gerir as tabelas anteriores, foram implementadas funcionalidades no *schema* CONTENT_FUNC, sob a forma de subprogramas agrupados pelos seguintes *packages*:

- PK_CONTEXT_CATALOGUE – *package* que agrega um conjunto de subprogramas que permitem a gestão dos registos da tabela CONTEXT_CATALOGUE;
- PK_CONTEXT_CLINICAL_SERVICE – *package* que agrega um conjunto de subprogramas que permitem a gestão dos registos da tabela CONTEXT_CLINICAL_SERVICE;
- PK_CONTEXT_COMPLAINT – *package* que agrega um conjunto de subprogramas que permitem a gestão dos registos da tabela CONTEXT_COMPLAINT;
- PK_CONTEXT_MARKET – *package* que agrega um conjunto de subprogramas que permitem a gestão dos registos da tabela CONTEXT_MARKET;
- PK_CONTEXT_SOFTWARE – *package* que agrega um conjunto de subprogramas que permitem a gestão dos registos da tabela CONTEXT_SOFTWARE.

Nestes *packages* encontra-se a implementação de três subprogramas – *set_context*, *set_context_status* e, por fim, *get_context*. O *procedure set_context*, representado no Extrato Código 9, permite definir e atualizar a associação entre o identificador do tipo de conteúdo (*i_id_cnt_type*), o identificador do conteúdo (*i_id_cnt_item*) e o identificador do contexto (neste caso, mercado – *i_id_market*).

```
PROCEDURE set_context
(
    i_id_cnt_type IN t_id_cnt_type,
    i_id_cnt_item IN t_id_cnt_item,
    i_id_market   IN t_id_market,
    i_id_status  IN t_id_status
);
```

Extrato Código 9 – Especificação do *procedure set_context*

Por sua vez, o *procedure set_context_status*, representado no Extrato Código 10, habilita a possibilidade de atualizar o estado (*i_id_status*) da associação entre o identificador do tipo de um conteúdo (*i_id_cnt_type*), o identificador do conteúdo (*i_id_cnt_item*) e o identificador do contexto (neste caso, mercado – *i_id_market*).

```
PROCEDURE set_context_status
(
    i_id_cnt_type IN t_id_cnt_type,
    i_id_cnt_item IN t_id_cnt_item,
    i_id_market   IN t_id_market,
    i_id_status  IN t_id_status
);
```

Extrato Código 10 – Especificação do *procedure set_context_status*

Por fim, a *function* `get_context`, representada no Extrato Código 11, permite retornar a lista dos registos das associações entre um determinado tipo de conteúdo (*i_id_cnt_type*) e, neste caso, um determinado identificador do contexto mercado (*i_id_market*).

```

FUNCTION get_context
(
    i_id_cnt_type IN t_id_cnt_type,
    i_id_market   IN t_id_market,
    i_id_cnt_item IN t_id_cnt_item DEFAULT NULL,
    i_ids_cnt_item IN table_number DEFAULT NULL
) RETURN table_ctx_market
PIPELINED;

```

Extrato Código 11 – Especificação da *function* `get_context`

Assim, na Figura 35 é representado o diagrama de seqüência da configuração do contexto mercado num conteúdo do tipo exame.

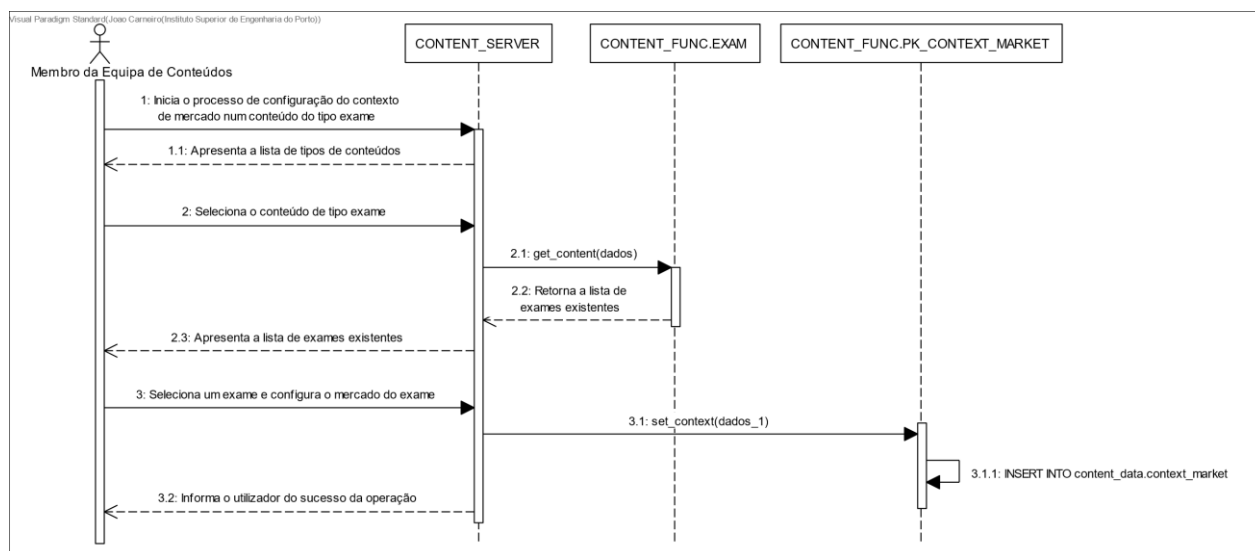


Figura 35 - Diagrama de seqüência referente à configuração do contexto mercado num conteúdo do tipo exame

Na Figura 35, constata-se a participação de três intervenientes no diagrama de seqüência, nomeadamente:

- **CONTENT_SERVER**: componente da responsabilidade de outro membro da ALERT, que invoca funcionalidades que foram desenvolvidas no âmbito deste projeto. Trata-se da interface gráfica, através da qual um membro da equipa de conteúdos pode, de uma forma simplificada, gerir os conteúdos do ALERT®;
- **CONTENT_FUNC.PK_EXAM**: package armazenado diretamente na base de dados Oracle no *schema* `CONTENT_FUNC`, que agrupa logicamente variáveis e subprogramas relacionados (blocos de código escrito em PL/SQL), que permitem a gestão dos conteúdos do tipo exame;

- CONTENT_FUNC.PK_CONTEXT_MARKET: package armazenado diretamente na base de dados Oracle no *schema* CONTENT_FUNC, que agrega logicamente um conjunto de variáveis e subprogramas relacionados (blocos de código escrito em PL/SQL), que permitem a gestão dos registos da tabela CONTEXT_MARKET.

No diagrama de sequência referente à configuração do contexto mercado num conteúdo do tipo exame, representado na Figura 35, constata-se que o ator – membro da equipa de conteúdos – interage com o componente CONTENT_SERVER. Verifica-se que, inicialmente, o ator interage com o componente CONTENT_SERVER, de forma a dar início ao processo de configuração do contexto de mercado num conteúdo do tipo exame. Neste sentido, o CONTENT_SERVER apresenta a lista de tipos de conteúdos e, neste caso, o ator opta pelo conteúdo do tipo exame. De seguida, o CONTENT_SERVER devolve a lista de conteúdos deste tipo, o ator seleciona um exame e insere os dados da configuração do mercado. Assim, o CONTENT_SERVER invoca um subprograma – *set_context(dados1)* –, cuja especificação está representada anteriormente, no Extrato Código 9, que se encontra armazenado diretamente na base de dados, mais precisamente, no *package* PK_CONTEXT_MARKET, no *schema* CONTENT_FUNC. Por fim, o *procedure set_context(dados1)* insere os dados na respetiva tabela – *context_market* – que se encontra armazenada no *schema* CONTENT_DATA, tal como se pode verificar no Extrato Código 12.

```

PROCEDURE set_context(i_id_cnt_type IN t_id_cnt_type,
                    i_id_cnt_item IN t_id_cnt_item,
                    i_id_market   IN t_id_market,
                    i_id_status IN t_id_status) IS
BEGIN
  BEGIN
    INSERT INTO content_data.context_market
      (id_cnt_type, id_cnt_item, id_market, id_status, last_change)
    VALUES
      (i_id_cnt_type,
       i_id_cnt_item,
       i_id_market,
       i_id_status,
       current_timestamp);
  EXCEPTION
    WHEN dup_val_on_index THEN
      UPDATE content_data.context_market m
        SET m.id_status = i_id_status, m.last_change = current_timestamp
        WHERE m.id_cnt_type = i_id_cnt_type
          AND m.id_cnt_item = i_id_cnt_item
          AND m.id_market = i_id_market;
  END;
END;

```

Extrato Código 12 – Implementação do *procedure set_context*

8.5 Histórico de Alterações

Tal como se pode verificar na secção 1.3 (Objetivos), um dos objetivos deste projeto consiste no desenvolvimento de funcionalidades que habilitem o registo automático de todas as alterações efetuadas ao nível dos conteúdos e respetivas configurações.

Assim, como referido na secção 7.4, é no *schema* HISTORICAL_DATA que se encontram armazenados os objetos de base de dados relacionados com o histórico das alterações dos conteúdos. Neste sentido, na Figura 36 é representado o modelo relacional referente ao histórico das alterações dos conteúdos, em que se destaca duas tabelas – STATUS_HISTORY e CONTENT_HISTORY.

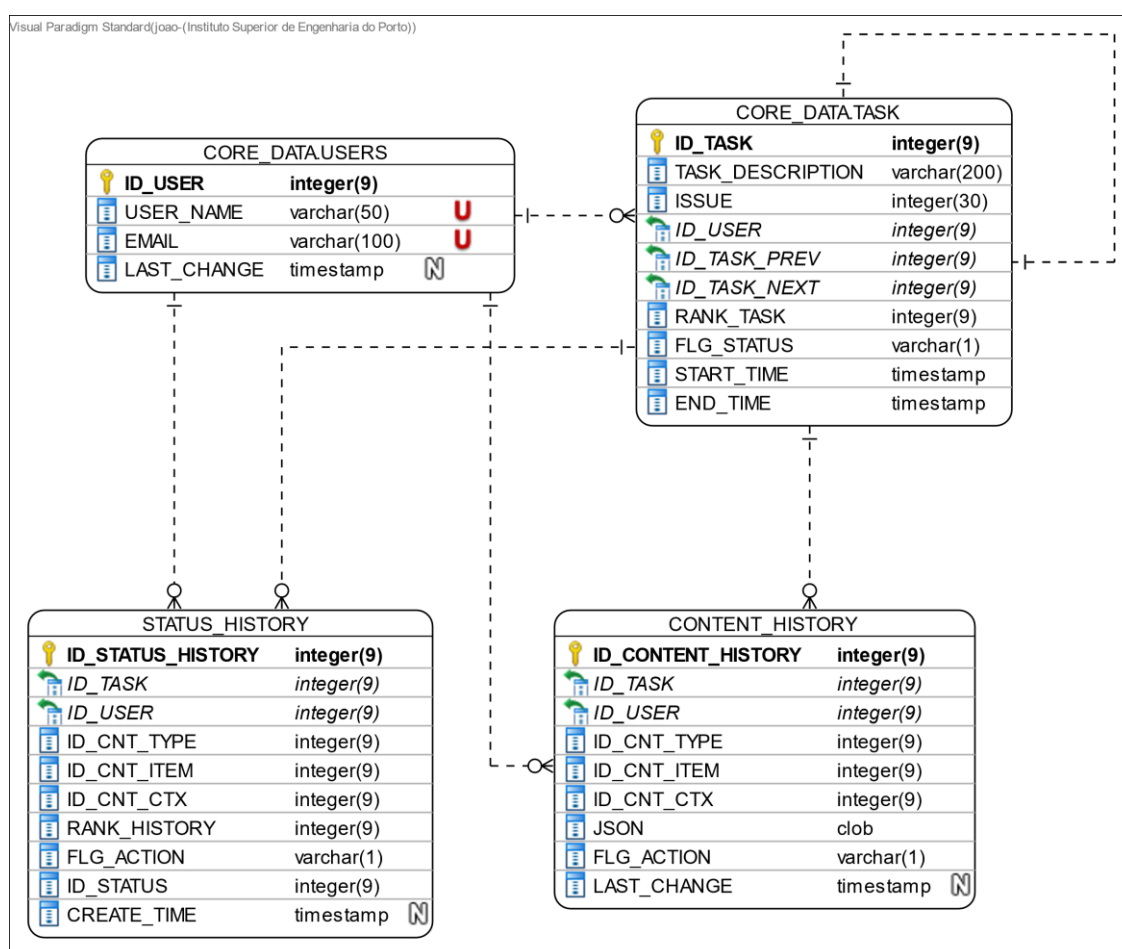


Figura 36 – Modelo relacional referente ao histórico das alterações dos conteúdos

A razão pela qual se optou pela criação de duas tabelas – STATUS_HISTORY e CONTENT_HISTORY - para armazenar o histórico das alterações dos conteúdos, deve-se ao facto de se ter constatado, juntamente com a equipa, que, muitas vezes, no processo de atualização de um conteúdo apenas se altera o estado do mesmo e que esta alteração de estado de um conteúdo é bastante frequente, devido ao ciclo de vida que um conteúdo sofre. Desta forma,

por um lado, constatou-se a necessidade de criar uma tabela – STATUS_HISTORY - dedicada ao armazenamento dos registos de alteração do estado (*ID_STATUS*) dos conteúdos e, por outro lado, a necessidade de criar uma tabela – CONTENT_HISTORY – onde são armazenados os registos de alterações das restantes colunas de um conteúdo. O facto de não se ter optado pela criação de uma só tabela para armazenar os registos das alterações dos conteúdos deve-se ao facto de que estes registos são armazenados sob a forma de um JSON, que representa o anterior valor das colunas antes da atualização. Neste contexto, tal como referido anteriormente, sendo que o mais comum é alteração do estado de um conteúdo, optou-se pela criação de duas tabelas distintas, com o intuito de poupar espaço na base de dados.

Neste âmbito, de forma a habilitar a gestão dos registos de histórico desenvolveu-se um package – PK_HISTORY – no qual se implementaram dois *procedures* - *set_content_history* e *set_status_history* – representados no Extrato Código 13.

```
PROCEDURE set_content_history(i_id_cnt_type      IN t_id_cnt_type,
                             i_id_cnt_item      IN t_id_cnt_item,
                             i_id_cnt_ctx      IN t_id_cnt_ctx,
                             i_id_task        IN t_id_task,
                             i_id_user        IN t_id_user,
                             i_json           IN t_json,
                             i_flg_action     IN t_flg_action,
                             o_id_content_history OUT t_id_content_history);

PROCEDURE set_status_history(i_id_cnt_type      IN t_id_cnt_type,
                            i_id_cnt_item      IN t_id_cnt_item,
                            i_id_cnt_ctx      IN t_id_cnt_ctx,
                            i_id_task        IN t_id_task,
                            i_id_user        IN t_id_user,
                            i_flg_action     IN t_flg_action,
                            i_id_status      IN t_id_status,
                            o_id_status_hist OUT t_id_status_hist);
```

Extrato Código 13 – Especificação dos *procedures set_content_history* e *set_status_history*

O primeiro *procedure* - *set_content_history* -, permite a inserção do histórico das alterações de um determinado conteúdo na tabela CONTENT_HISTORY e, o segundo – *set_status_history* – por sua vez, habilita a inserção do histórico das alterações do estado dos conteúdos na tabela STATUS_HISTORY.

9 Experimentação e Avaliação

Neste capítulo apresentam-se os indicadores adotados para a avaliação do projeto, de seguida elaboram-se as hipóteses, e, por fim, apresentam-se as metodologias de avaliação escolhidas para a avaliação do projeto, bem como os respetivos resultados.

9.1 Indicadores de Avaliação

De forma a avaliar o projeto referente a este documento, definiu-se um conjunto de indicadores de avaliação, nomeadamente:

- Cumprimento dos requisitos não funcionais de desempenho;
- Satisfação da equipa de conteúdos com a solução desenvolvida.

9.2 Hipóteses

Uma vez definidos os indicadores de avaliação na secção anterior, é necessário formular hipóteses para testar os indicadores. Neste âmbito, foram definidas as seguintes hipóteses:

- Cumprimento dos requisitos não funcionais de desempenho – verificar se os tempos de resposta das funcionalidades implementadas cumprem os tempos de resposta estipulados na secção 6.4;
- Satisfação dos membros da equipa de conteúdos superior a 4 (0-5) - hipótese que permite identificar o nível de satisfação da equipa de conteúdos.

9.3 Metodologia de Avaliação

Após definidos os indicadores e elaboradas as hipóteses de avaliação, é necessário escolher as metodologias de avaliação. Desta forma, na Tabela 29 é indicada a metodologia adotada para cada um dos indicadores definidos.

Tabela 29 – Metodologia adotadas para cada indicador

Indicador	Metodologia de Avaliação
Cumprimento dos requisitos não funcionais de desempenho	Testes de desempenho
Satisfação	Inquérito de satisfação e teste de hipóteses

Relativamente aos testes de desempenho, pretende-se que para cada tipo de operação de gestão de conteúdos – criação, alteração, atualização do estado e consulta - sejam realizados 85 testes para cada operação de criação, alteração e atualização do estado e 87 testes para a operação de consulta de conteúdos, sendo cada um dos testes executado 1000 vezes. Adicionalmente, para cada teste, é calculada a média dos tempos de resposta das respetivas 1000 execuções. O valor de referência a utilizar para cada teste é a média obtida.

De modo a aferir a satisfação da equipa de conteúdos relativamente à solução desenvolvida, optou-se por realizar um inquérito de satisfação. Com base nas respostas obtidas neste inquérito, elaborou-se um teste de hipóteses para avaliar o nível de satisfação da equipa. No teste de hipóteses, definiu-se o nível de significância como 0.05, valor que corresponde a um grau de confiança de 95%. Este inquérito, disponível no Anexo E, é constituído por oito questões diretas, de resposta fechada. Neste sentido, as respostas são dadas, tendo em consideração o modelo da escala de Likert, que admite cinco respostas possíveis, nomeadamente (Boone & Boone, 2012):

- Discordo totalmente;
- Discordo parcialmente;
- Indiferente;
- Concordo parcialmente;
- Concordo totalmente.

9.4 Avaliação de Resultados

Nesta secção é elaborada a avaliação dos resultados relativos às hipóteses anteriormente definidas (secção 9.2). Assim, são apresentados os resultados obtidos do uso de cada metodologia.

9.4.1 Testes de Desempenho

Na secção 6.4 definiram-se os requisitos não funcionais do presente projeto. Dos requisitos não funcionais definidos, dois são relacionados com o desempenho da solução, tendo-se estabelecido tempos de resposta para as diferentes funcionalidades, nomeadamente:

- Tempo de resposta não pode ultrapassar os 500 milissegundos para os pedidos de consulta de dados;
- Tempo de resposta não pode ultrapassar os 50 milissegundos para os pedidos de alteração de dados.

Neste âmbito, através da ferramenta *Load Tester* do IDE PL/SQL Developer desenvolveram-se testes, de forma a avaliar os tempos de resposta das funcionalidades desenvolvidas e armazenadas na base de dados. O servidor da base de dados Oracle, na qual se implementou as funcionalidades, encontra-se alojado numa máquina virtual com as seguintes especificações:

- 2 vCPUS: Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz;
- 4 GB RAM.

Desta forma, a Figura 37 representa alguns dos testes que foram desenvolvidos para avaliar o tempo de resposta das funcionalidades que permitem a criação de conteúdos.

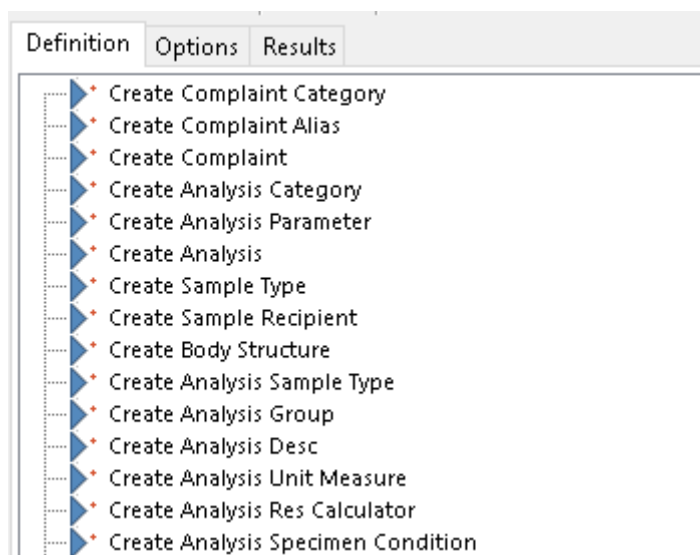


Figura 37 – Alguns dos testes para avaliar o tempo de resposta das funcionalidades de criação de conteúdos

O Extrato Código 16, presente no Anexo A, apresenta o exemplo do código de um teste do tempo de resposta da funcionalidade que permite a criação de conteúdos do tipo categoria de queixa, sendo que, em cada um dos testes, são criados 1000 registos do tipo de conteúdo do respetivo teste.

No Anexo A, a Figura 49, a Figura 50, a Figura 51, a Figura 52 e a Figura 53 apresentam os resultados dos testes desenvolvidos, isto é, a média do tempo de resposta de cada teste. Com base nestes resultados, confirma-se que a média do tempo de resposta de cada um dos testes é inferior ao tempo de resposta estabelecido para os pedidos de alteração de dados - 50 milissegundos.

Na Figura 38 encontram-se representados alguns dos testes desenvolvidos para testar o tempo de resposta das funcionalidades que permitem a atualização de conteúdos existentes.

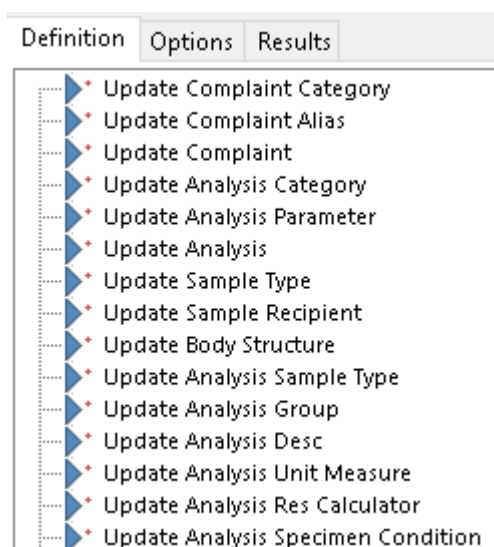


Figura 38 – Alguns dos testes para avaliar o tempo de resposta das funcionalidades de atualização de conteúdos

O Extrato Código 17, presente no Anexo B, consiste no código de um teste para avaliar o tempo de resposta da funcionalidade da atualização de conteúdos do tipo categoria de queixa, sendo que, em cada teste são atualizados valores dos 1000 tipos de conteúdos criados nos testes anteriores.

No Anexo B, a Figura 54, a Figura 55, a Figura 56, a Figura 57 e a Figura 58 apresentam os resultados dos testes desenvolvidos, por outras palavras, a média do tempo de resposta de cada teste. Tendo em conta estes resultados, verifica-se que a média do tempo de resposta de cada um dos testes é inferior ao tempo de resposta definido, no que diz respeito aos pedidos de alteração de dados - 50 milissegundos.

Adicionalmente, na Figura 39 representam-se alguns testes desenvolvidos, com o intuito de avaliar o tempo de resposta das funcionalidades, que permitem a mudança do estado de conteúdos existentes.

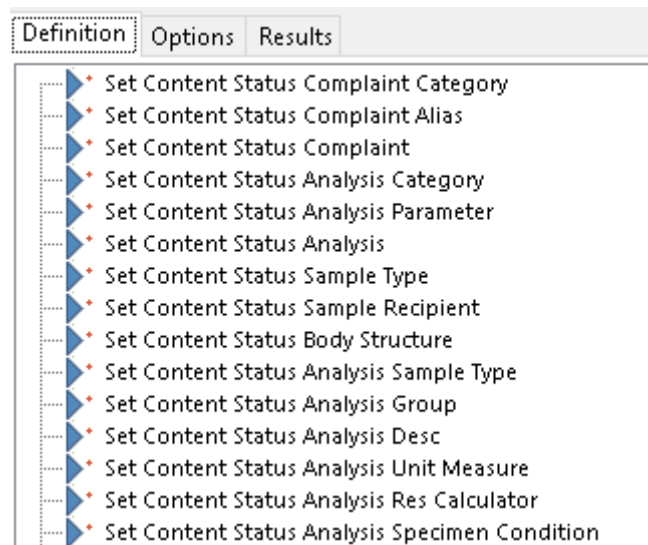


Figura 39 - Alguns dos testes para avaliar o tempo de resposta das funcionalidades de atualização do estado de conteúdos

O Extrato Código 18, presente no Anexo C, consiste no código de um teste para avaliar o tempo de resposta da funcionalidade que permite a atualização do estado de conteúdos do tipo categoria de queixa, sendo que, em cada teste são atualizados os estados de cada um dos 1000 conteúdos criados nos testes iniciais.

No Anexo C, a Figura 59, a Figura 60, a Figura 61, a Figura 62 e a Figura 63 apresentam os resultados dos testes desenvolvidos, isto é, a média do tempo de resposta de cada teste. Com base nestes resultados, constata-se que a média do tempo de resposta de cada teste é inferior ao tempo de resposta definido, no que diz respeito aos pedidos de alteração de dados - 50 milissegundos.

Por fim, a Figura 40 representa alguns dos testes desenvolvidos, com a finalidade de avaliar o tempo de respostas das funcionalidades, que habilitam a consulta de dados de cada tipo de conteúdos.

Definition	Options	Results
Get Content Complaint		
Get Content Complaint Alias		
Get Content Complaint Category		
Get Content Analysis Sample Type		
Get Content Analysis Sample Type Simple		
Get Content Analysis Sample Type Complex		
Get Content Analysis Group		
Get Content Analysis Category		
Get Content Analysis Parameter		
Get Content Analysis Desc		
Get Content Analysis Unit Measure		
Get Content Analysis		
Get Content Sample Type		
Get Content Sample Recipient		
Get Content Analysis Res Calculator		

Figura 40 - Alguns dos testes para avaliar o tempo de resposta das funcionalidades de obtenção dos dados de conteúdos

O Extrato Código 19, presente no Anexo D, consiste no código do teste para avaliar o tempo de resposta da funcionalidade que permite a obtenção de todos os conteúdos do tipo categoria de queixa, sendo que, em cada um dos testes são retornados os dados dos 1000 conteúdos criados nos testes iniciais.

No Anexo D, a Figura 64, a Figura 65, a Figura 66, a Figura 67 e a Figura 68 apresentam os resultados dos testes desenvolvidos, isto é, a média do tempo de resposta de cada teste. Assim, tendo em conta estes resultados, verifica-se que a média do tempo de resposta de cada teste é inferior ao tempo de resposta definido, relativamente aos pedidos de consulta de dados - 500 milissegundos.

9.4.2 Inquérito de Satisfação

O inquérito de satisfação, disponível no Anexo E, foi submetido a três dos quatro membros da equipa de conteúdos, uma vez que, à data da elaboração do documento, um dos membros se encontrava de baixa médica, ficando assim impossibilitado de responder ao inquérito. Como referido anteriormente, o inquérito é constituído por oito questões de resposta fechada, em que as respostas são baseadas no modelo da escala de Likert. A Tabela 30 demonstra a pontuação numérica a que correspondem cada uma das cinco respostas possíveis.

Tabela 30 – As respostas do inquérito de satisfação e a respetiva pontuação numérica

Respostas Possíveis	Pontuação Numérica
Discordo totalmente	1
Discordo parcialmente	2
Indiferente	3
Concordo parcialmente	4
Concordo totalmente	5

De seguida, são apresentados os resultados de cada uma das respostas do inquérito de satisfação submetido à equipa de conteúdos.

Relativamente à primeira questão – “O modelo de dados reformulado/atualizado apresenta as alterações identificadas.” – foram obtidos os resultados que se encontram representados na Figura 41.

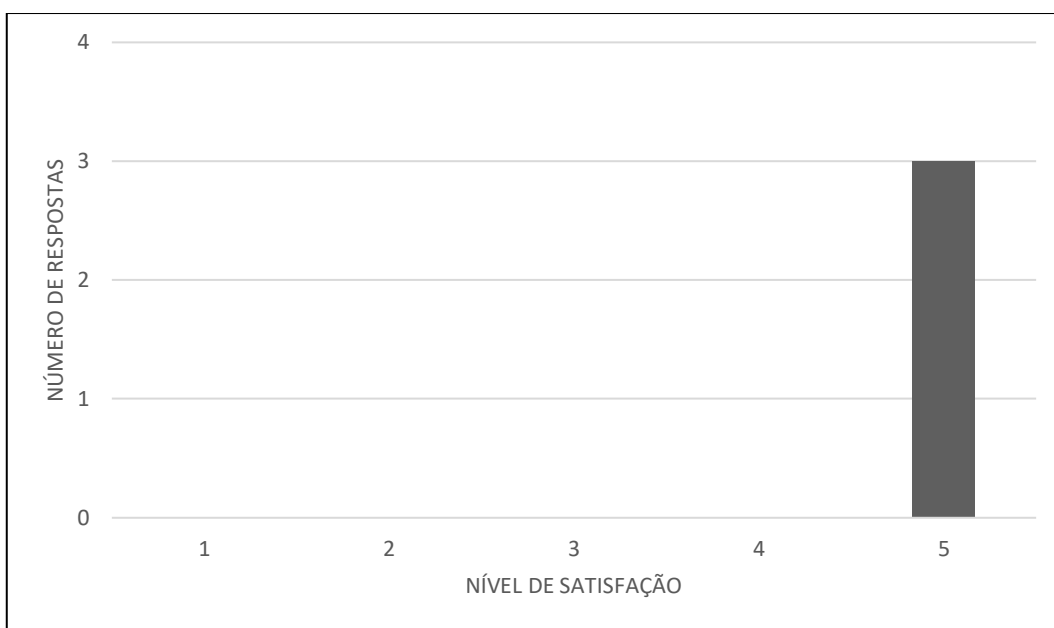


Figura 41 – Resultados da primeira questão do inquérito de satisfação

No que diz respeito à segunda questão – “As funcionalidades de criação de conteúdos foram implementadas corretamente.” – obtiveram-se os resultados representados na Figura 42.

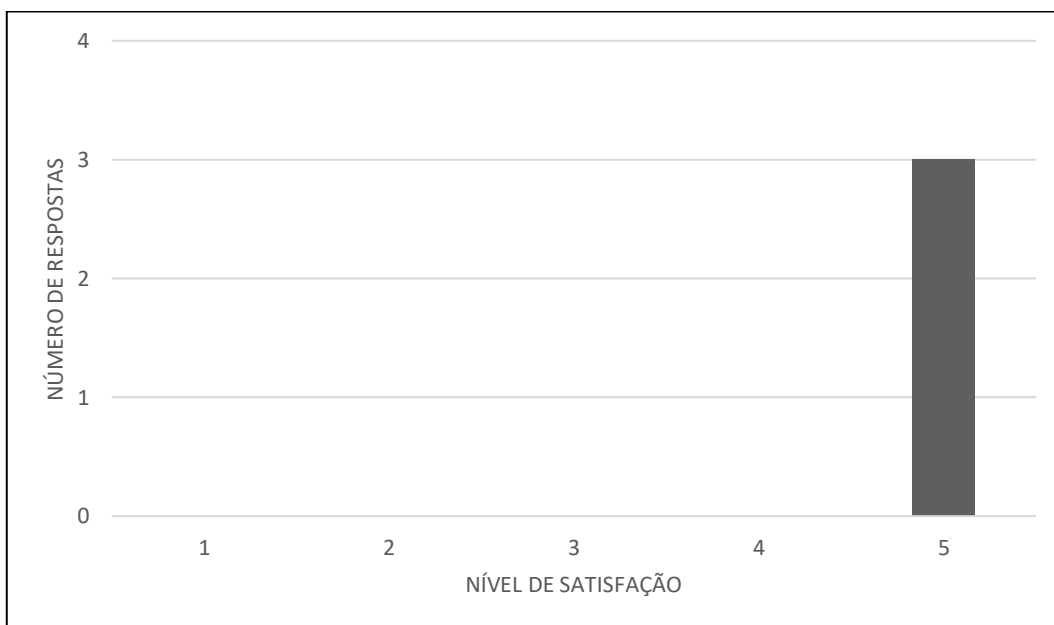


Figura 42 – Resultados da segunda questão do inquérito de satisfação

No que concerne a terceira questão – “As funcionalidades de atualização de conteúdos foram implementadas corretamente.” – foram obtidos os resultados representados na Figura 43.

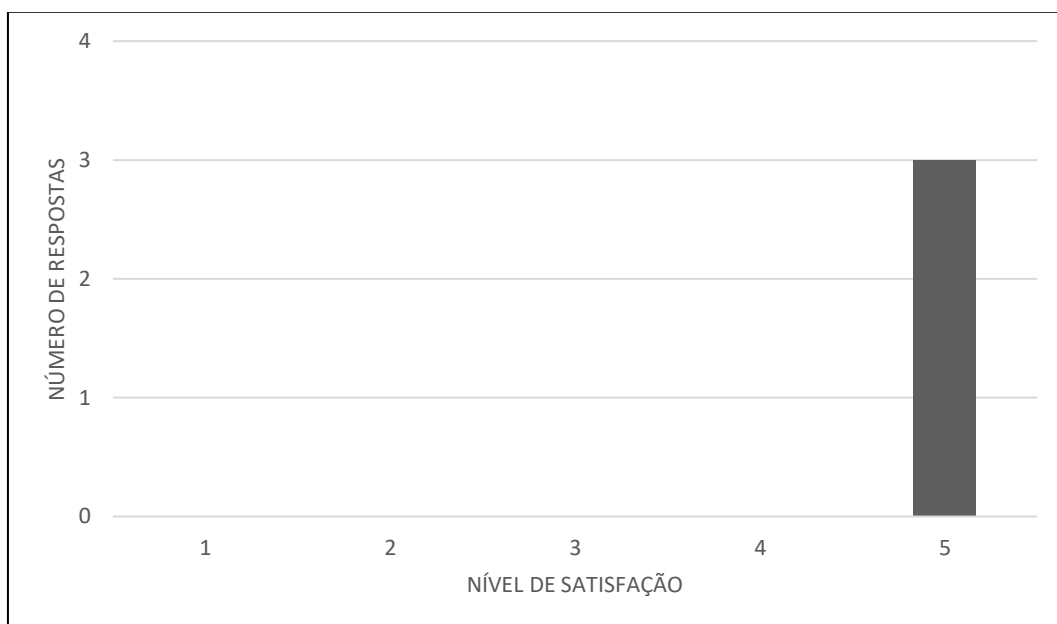


Figura 43 - Resultados da terceira questão do inquérito de satisfação

Quanto à quarta questão – “As funcionalidades de visualização de conteúdos foram implementadas corretamente.” - obtiveram-se os resultados representados na Figura 44.

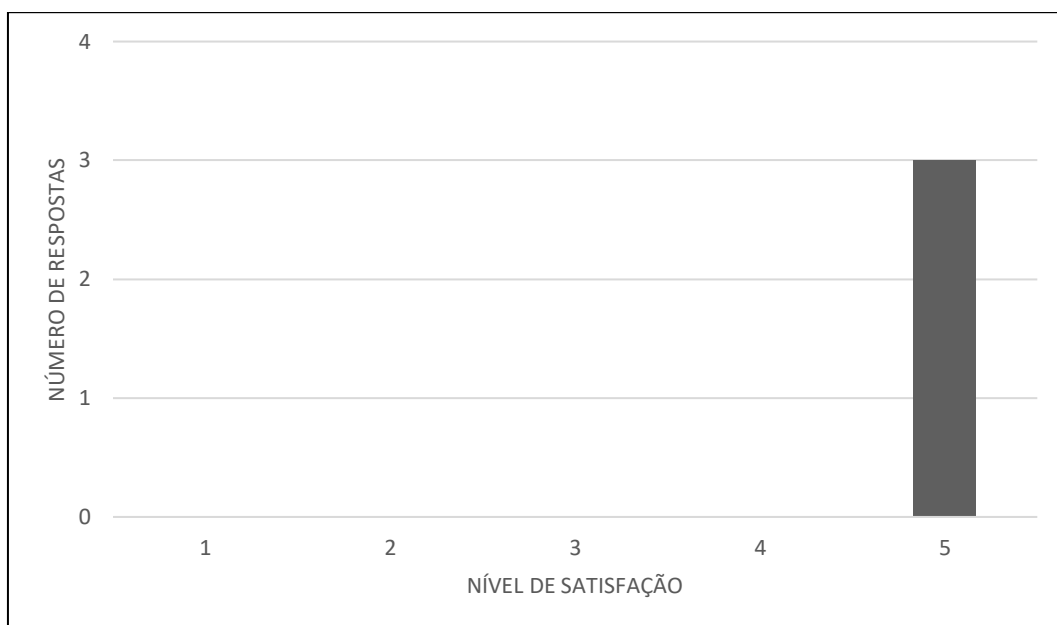


Figura 44 - Resultados da quarta questão do inquérito de satisfação

No que toca a quinta questão – “As funcionalidades desenvolvidas permitem a configuração dos conteúdos.” – foram obtidos os resultados representados na Figura 45.

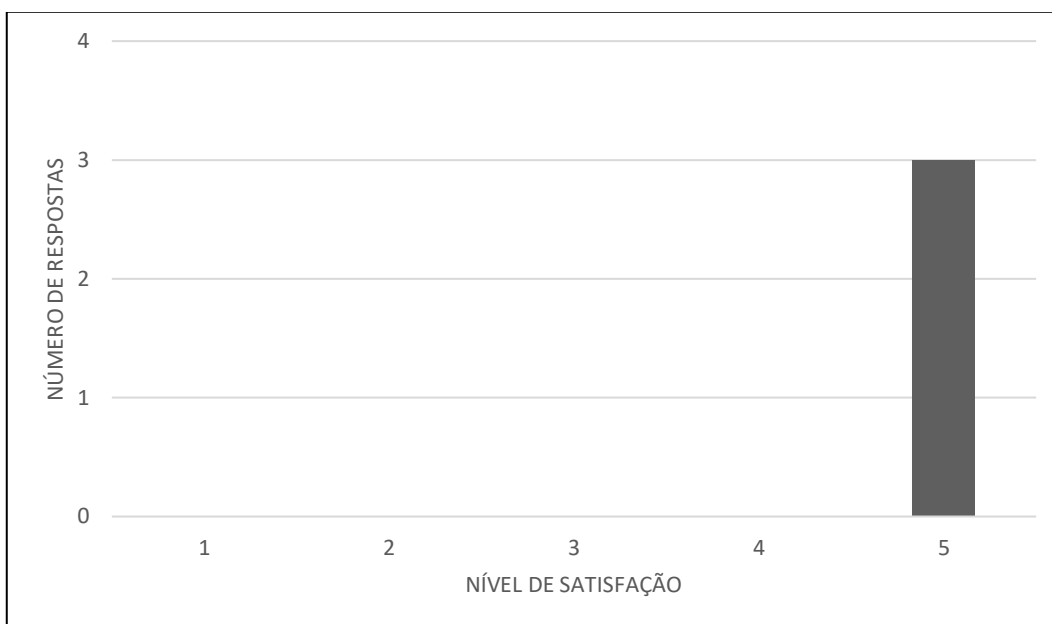


Figura 45 - Resultados da quinta questão do inquérito de satisfação

Em relação à sexta questão – “As funcionalidades desenvolvidas permitem o registo do histórico das alterações dos conteúdos.” - obtiveram-se os resultados representados na Figura 46.

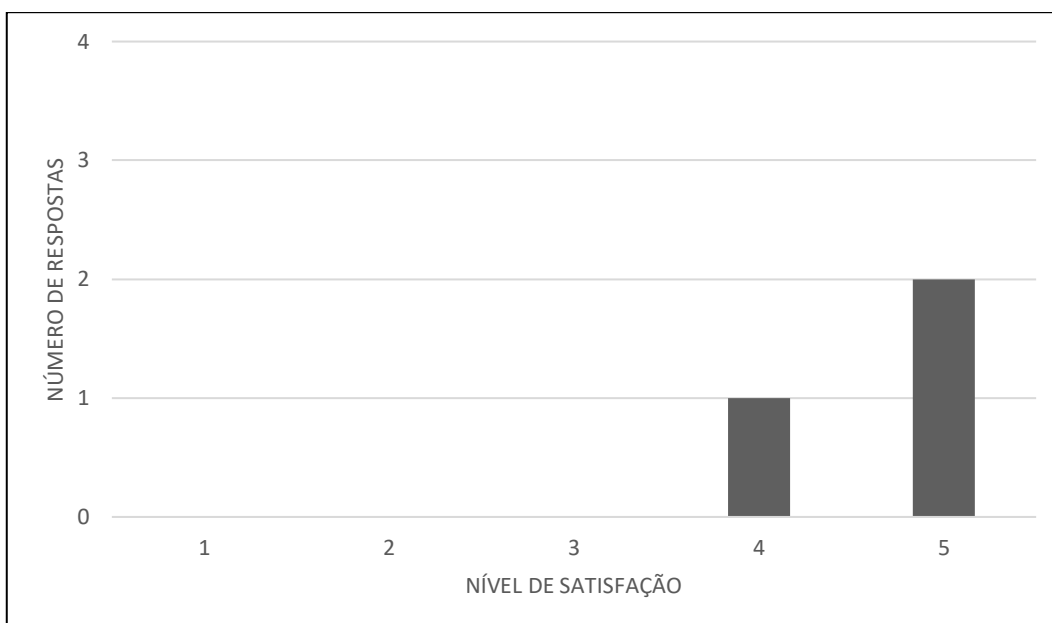


Figura 46 - Resultados da sexta questão do inquérito de satisfação

Relativamente à sétima questão – “As regras de negócio dos conteúdos foram implementadas corretamente.” – foram obtidos os resultados representados na Figura 47.

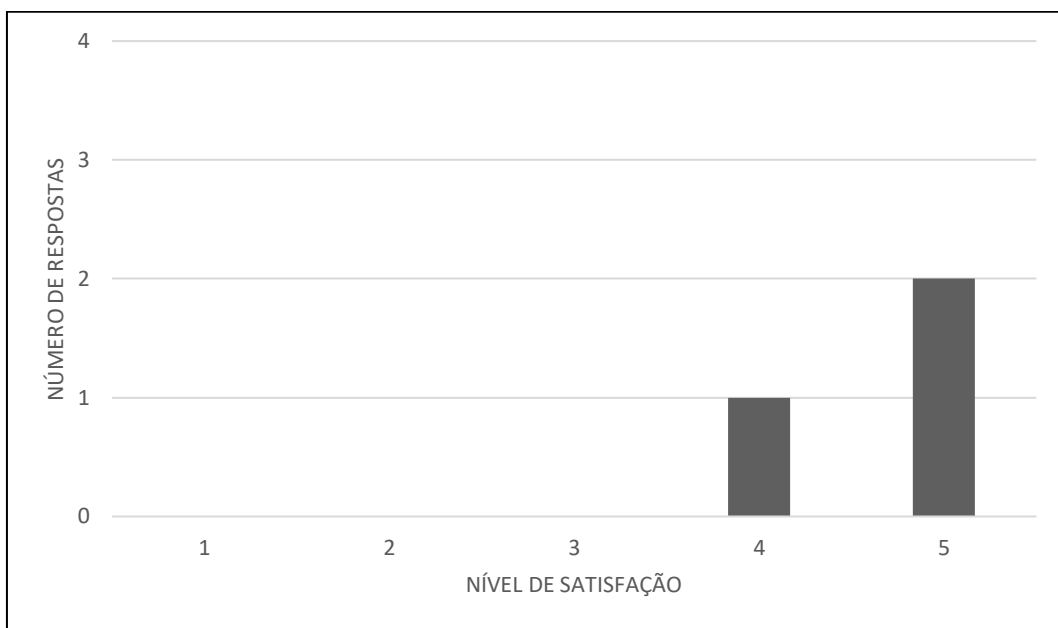


Figura 47 - Resultados da sétima questão do inquérito de satisfação

Por fim, no que diz respeito à oitava, e última, questão – “As funcionalidades desenvolvidas a implementar na plataforma *Content Server* facilitarão e ajudarão a tornar os processos de gestão de conteúdos mais rápido.” – foram obtidos os resultados representados na Figura 48.



Figura 48 - Resultados da oitava questão do inquérito de satisfação

Na Tabela 31 apresenta-se o número de respostas para cada tipo de resposta possível.

Tabela 31 – Número de respostas para cada tipo de resposta possível

Respostas Possíveis	Número de Resposta
1 – Discordo totalmente	0
2 – Discordo parcialmente	0
3 – Indiferente	0
4 – Concordo parcialmente	2
5 – Concordo totalmente	22

De forma a analisar os resultados do inquérito de satisfação, primeiramente verificou-se se os dados da Tabela 31 seguem, ou não, uma distribuição normal. Assim, recorreu-se ao teste de normalidade *Shapiro-Wilk* (Razali & Wah, 2011) e definiram-se as seguintes hipóteses:

- H_0 – Os valores da amostra seguem uma distribuição normal;
- H_1 – Os valores da amostra não seguem uma distribuição normal.

No Extrato Código 14 representa-se o teste de normalidade *Shapiro-Wilk* e o respetivo resultado.

```
> dados <- c(5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,4,4)
> shapiro.test(dados)

      shapiro-wilk normality test

data:  dados
W = 0.3155, p-value = 1.38e-09
```

Extrato Código 14 - Teste de normalidade *Shapiro-Wilk*

No Extrato Código 14, é aplicado o teste *Shapiro-Wilk* e, verifica-se que o valor de *p-value* é $1.38e-09$. Neste contexto, como o valor de *p-value* é menor do que o valor da significância definido (0.05), rejeita-se a hipótese H_0 , podendo-se afirmar que a amostra não segue uma distribuição normal.

Assim, é necessário aplicar um teste não paramétrico, em detrimento de um teste paramétrico, uma vez que a amostra não segue uma distribuição normal. Como referido na secção 9.2, definiu-se a seguinte hipótese – “Satisfação dos membros da equipa de conteúdos superior a 4 (0-5)”. Neste sentido, definiram-se as seguintes hipóteses para o teste não paramétrico de *Wilcoxon* (Investopedia, 2020):

- H_0 – A média das respostas obtidas é inferior ou igual a 4;
- H_1 – A média das respostas obtidas é superior a 4.

Neste âmbito, no Extrato Código 15 encontra-se o teste não paramétrico de *Wilcoxon* aplicado e o respetivo resultado.

```
> wilcox.test(dados, mu = 4, alternative = "greater" )  
      wilcoxon signed rank test with continuity correction  
data:  dados  
V = 253, p-value = 1.492e-06  
alternative hypothesis: true location is greater than 4
```

Extrato Código 15 – Teste não paramétrico de *Wilcoxon*

Com base no resultado representado no Extrato Código 15, verifica-se que o valor de *p-value* é de 1.492e-06, sendo este valor inferior ao nível de significância (0.05), rejeitando-se, desta forma, a hipótese H_0 . Assim, pode afirmar-se, com um grau de confiança de 95%, que o nível de satisfação da equipa de conteúdos é superior a 4.

10 Conclusões

Neste capítulo é apresentada uma breve visão geral do trabalho desenvolvido no âmbito do projeto de estágio na ALERT, são apresentados os objetivos alcançados, são descritas algumas linhas orientadoras para o trabalho a desenvolver no futuro, e, por fim, é elaborada uma apreciação final sobre o projeto.

10.1 Visão Geral

Este projeto tem como objetivo principal a simplificação da gestão de conteúdos, através do desenvolvimento de funcionalidades que auxiliem os processos de criação, alteração e visualização de conteúdos do ALERT®. Desta forma, pretende-se que estas funcionalidades sejam invocadas numa plataforma intuitiva e segura, desenvolvida por outro elemento da ALERT.

Após a identificação dos problemas, seguiu-se a fase de análise das diferentes soluções que permitem alcançar o objetivo definido e, assim, colmatar os problemas identificados. Neste sentido, foram identificadas soluções para duas componentes considerados nucleares deste projeto – o tipo de base de dados e em que camada implementar a lógica de negócio. Relativamente ao tipo de base de dados, identificaram-se duas soluções – base de dados relacional e base de dados não relacional – e, em seguida, efetuou-se um estudo comparativo entre ambas, de forma a determinar qual seria a mais indicada para este tipo de projeto. Neste âmbito, optou-se por uma base de dados do tipo relacional, uma vez que, a equipa de conteúdos definiu como requisito não funcional de implementação a utilização de base de dados Oracle. No que diz respeito ao estudo referente à decisão acerca de onde implementar a lógica de negócio, identificaram-se duas alternativas – na camada de dados (diretamente na base de dados) ou na camada de aplicação – e procedeu-se a uma análise comparativa entre ambas, com o objetivo de determinar qual a mais indicada para este tipo de projeto. Tendo em conta esta análise, e o facto de a equipa de conteúdos ter definido como requisito não funcional de desenho a implementação da lógica de negócio na camada de dados (diretamente na base de dados), determinou-se que esta seria a alternativa a adotar.

Em suma, optou-se pela adoção da base de dados do tipo relacional Oracle para o armazenamento dos dados dos conteúdos do ALERT® e para armazenar as funcionalidades implementadas através da linguagem de programação PL/SQL, que permitem a gestão dos conteúdos.

10.2 Objetivos Alcançados

Nesta secção estão presentes duas tabelas – Tabela 32 e Tabela 33 – que representam, respetivamente, os objetivos estabelecidos e grau de implementação, inicialmente, na secção 1.3 e, ainda, os casos de uso definidos na secção 6.3 e o respetivo grau de implementação.

Tabela 32 – Objetivos do projeto e respetivo grau de implementação

Objetivo	Grau de Implementação
Reformular/Atualizar o modelo de dados existente	Completo
Desenvolver funcionalidades que permitam: <ul style="list-style-type: none"> • Listar todos os conteúdos existentes para cada tipo de conteúdo e respetivas configurações; • Efetuar alterações a cada um dos conteúdos e configurações, salvaguardando a coerência dos dados; • Manter um registo automático de todas as alterações efetuadas. 	Completo
Documentar todo o desenvolvimento efetuado, bem como as regras implementadas	Completo
Testar as funcionalidades desenvolvidas	Completo

Tabela 33 – Casos de uso e respetivo grau de implementação

Caso de Uso	Grau de Implementação
UC01 – Gestão de conteúdos do tipo queixa	Completo
UC02 – Gestão de conteúdos do tipo exame	Completo
UC03 - Gestão de conteúdos do tipo categoria de exame	Completo
UC04 - Gestão de conteúdos do tipo localização anatómica	Completo
UC05 - Gestão de conteúdos do tipo análise de laboratório	Completo
UC06 - Gestão de conteúdos do tipo serviço clínico	Completo
UC07 - Gestão de conteúdos do tipo sangue/hemoderivados	Completo
UC08 - Gestão de conteúdos do tipo dieta	Completo
UC09 - Gestão de conteúdos do tipo alta	Completo
UC10 - Gestão de conteúdos do tipo instrução de alta	Completo
UC11 - Gestão de conteúdos do tipo hábito	Completo
UC12 - Gestão de conteúdos do tipo plano de saúde	Completo
UC13 - Gestão de conteúdos do tipo intervenção	Completo
UC14 - Gestão de conteúdos do tipo necessidade	Completo
UC15 - Gestão de conteúdos do tipo ocupação	Completo
UC16 - Gestão de conteúdos do tipo origem	Completo

UC17 - Gestão de conteúdos do tipo posicionamento	Completo
UC18 - Gestão de conteúdos do tipo questão clínica	Completo
UC19 - Gestão de conteúdos do tipo nota de resultado	Completo
UC20 - Gestão de conteúdos do tipo plano de intervenção social	Completo
UC21 - Gestão de conteúdos do tipo especialidade	Completo
UC22 - Gestão de conteúdos do tipo material	Completo
UC23 - Gestão de conteúdos do tipo transporte	Completo
UC24 - Gestão de conteúdos do tipo folha de resumo	Completo
UC25 - Gestão de conteúdos do tipo registo hídrico	Completo
UC26 - Gestão de conteúdos do tipo <i>checklist</i>	Completo
UC27 - Gestão dos contextos dos tipos de conteúdos	Completo
UC28 – Alterar o estado de conteúdos	Completo

10.3 Trabalho Futuro

No presente projeto foram desenvolvidas, com sucesso, as funcionalidades de gestão dos 25 tipos de conteúdo propostos pela equipa de conteúdos. Não obstante, atualmente existem mais de 100 tipos de conteúdos no ALERT®, pelo que, no futuro, vislumbra-se uma forte possibilidade de desenvolvimento das funcionalidades de gestão dos restantes tipos de conteúdos. Com o desenvolvimento destas funcionalidades, pretende-se que a plataforma que invoca estas funcionalidade e se encontra a ser desenvolvida por outro elemento da ALERT, seja capaz de suportar todos os conteúdos do ALERT®.

10.4 Apreciação Final

Durante o desenvolvimento deste projeto, foram adquiridos vastos conhecimentos, sobretudo, no que diz respeito ao funcionamento de um software clínico. Adicionalmente, este projeto permitiu o contacto com uma nova abordagem, relativamente à implementação da lógica de negócio na camada de dados, bem como, à aprendizagem de uma nova linguagem de programação – PL/SQL.

A solução desenvolvida no âmbito deste projeto reflete o cumprimento de todos os objetivos definidos inicialmente, assim como os requisitos funcionais e não funcionais. Neste âmbito, considera-se que este projeto contribui para a eficiência do trabalho dos membros da equipa de conteúdos, na medida em que permite a simplificação da gestão dos conteúdos. Adicionalmente, o projeto permite a disponibilização mais rápida de novos conteúdos e atualização de conteúdos existentes, através das funcionalidades desenvolvidas, que são invocadas por uma plataforma implementada por outro membro da ALERT®.

Referências

ALERT Life Sciences Computing, s.d. a. *Reconhecimento* | *ALERT-ONLINE.COM - Português*. [Online]

Available at: <https://www.alert-online.com/pt/results>

[Acedido em 3 Fevereiro 2021].

ALERT Life Sciences Computing, s.d. b. *Quem somos* | *ALERT-ONLINE.COM - Português*. [Online]

Available at: <https://www.alert-online.com/pt/mission-values>

[Acedido em 3 Fevereiro 2021].

ALERT Life Sciences Computing, s.d. c. *Introdução à ALERT*. [Online]

Available at: <https://www.alert-online.com/pt/introduction>

[Acedido em 3 Fevereiro 2021].

ALERT Life Sciences Computing, s.d. d. *Software clínico* | *ALERT-ONLINE.COM - Português*. [Online]

Available at: <https://www.alert-online.com/pt/presentation>

[Acedido em 30 Abril 2021].

Alexander, M. & Kusleika, D., 2018. *Access 2019 Bible*. s.l.:s.n.

Al-Harbi, K. M. A.-S., 2001. Application of the AHP in project management. *International Journal of Project Management*, 1 1.

Anon., s.d. *About ICNP* | *ICN - International Council of Nurses*. [Online]

Available at: <https://www.icn.ch/what-we-do/projects/ehealth-icnptm/about-icnp>

[Acedido em 11 22 2020].

ANSI, 2018. *The SQL Standard - ISO/IEC 9075*. [Online]

Available at: <https://blog.ansi.org/2018/10/sql-standard-iso-iec-9075-2016-ansi-x3-135/>

[Acedido em 4 Janeiro 2021].

Atlassian, s.d a. *What is a Kanban Board?*. [Online]
Available at: <https://www.atlassian.com/agile/kanban/boards>
[Acedido em 15 Abril 2021].

Atlassian, s.d. b. *Kanban - A brief introduction*. [Online]
Available at: <https://www.atlassian.com/agile/kanban>
[Acedido em 15 Abril 2021].

Automations, A., s.d. *Allround Automations*. [Online]
Available at: <https://www.allroundautomations.com/products/pl-sql-developer/>
[Acedido em 10 12 2020].

Bagchi, S. & Tulsikie, B., 2000. *e-business Models: Integrating Learning from Strategy Development Experiences and Empirical Research*. s.l., s.n.

Belliveau, P., Griffin, A. & Somermeyer, S., 204. *The PDMA ToolBook 1 for New Product Development*. s.l.:John Wiley & Sons.

Booch, G., Rumbaugh, J. & Jacobson, I., 1999. Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series). *J. Database Manag.*, 1 Janeiro.

Boone, H. N. & Boone, D. A., 2012. Analyzing likert data. *Journal of extension*.

Borges, K. N. R., 2014. LibreOffice Para Leigos. *Facilitando a vida no escritório*.

Codd, E. F., 1983. A relational model of data for large shared data banks. *Communications of the ACM*, 1 Janeiro.

Cudney, E. & Elrod, C., 2011. *Quality Function Deployment in Continuous Improvement*. s.l.:s.n.

DB-Engines, 2021. *DB-Engines Ranking*. [Online]
Available at: <https://db-engines.com/en/ranking/relational+dbms>
[Acedido em 4 Janeiro 2021].

F3M, s.d a. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/empresa/apresentacao>
[Acedido em 18 junho 2020].

F3M, s.d b. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/gestao-de-unidades-de-saude>
[Acedido em 18 junho 2021].

F3M, s.d. i. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/processo-clinico-e-social>
[Acedido em 18 junho 2021].

- F3M, s.d. c. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/clinica>
[Acedido em 18 junho 2021].
- F3M, s.d. d. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/internamento>
[Acedido em 18 junho 2021].
- F3M, s.d. e. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/internamento-lar>
[Acedido em 18 junho 2021].
- F3M, s.d. f. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/unidades-de-cuidados-continuados>
[Acedido em 18 junho 2021].
- F3M, s.d. g. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/prescricao-eletronica>
[Acedido em 18 junho 2021].
- F3M, s.d. h. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/gestao-hospitalar>
[Acedido em 18 junho 2021].
- F3M, s.d. j. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/bloco-operatorio>
[Acedido em 18 junho 2021].
- F3M, s.d. k. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/tratamento-simplificado-de-toda-a-movimentacao-de-stock>
[Acedido em 18 junho 2021].
- F3M, s.d. l. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/processo-integrado-do-utente>
[Acedido em 18 junho 2021].
- F3M, s.d. m. *F3M*. [Online]
Available at: <https://www.f3m.pt/pt/software/saude/mpds>
[Acedido em 18 junho 2021].
- Faculdade de Ciências da Universidade de Lisboa, s.d.. *Histologia*. [Online]
Available at: <https://ciencias.ulisboa.pt/pt/taxonomy/term/1580>
[Acedido em 7 Abril 2021].
- Feuerstein, S. & Pribyl, B., 2005. *Oracle PL/SQL Programming*. s.l.:O'Reilly Media, Inc..

Feuerstein, S. & Pribyl, B., 2014. *Oracle PL/SQL programming*. 6º ed. Beijing ; Cambridge ; Farnham ; Köln ; Sebastopol ; Tokyo: O'Reilly.

Ganesh Chandra, D., 2015. BASE analysis of NoSQL database. *Future Generation Computer Systems*, 1 Novembro.

Glantt, s.d. a. *Sobre a Glantt*. [Online]

Available at: <https://www.glantt.com/pt/o-que-somos/sobreaglantt/Paginas/default.aspx>

[Acedido em 6 Abril 2021].

Glantt, s.d. b. *Globalcare*. [Online]

Available at: <https://www.glantt.com/pt/o-que-fazemos/ofertas/SoftwareSolutions/Paginas/Globalcare.aspx>

[Acedido em 6 Abril 2021].

Glantt, s.d. c. *Globalcare - Glantt*. [Online]

[Acedido em 6 Abril 2021].

Glantt, s.d. d. *Hospital Management System*. [Online]

Available at: <https://globalcare.glantt.com/hospital-management-system/>

[Acedido em 7 Abril 2021].

Glantt, s.d. e. *Clinical*. [Online]

Available at: <https://globalcare.glantt.com/clinical/>

[Acedido em 7 Abril 2021].

Glantt, s.d. f. *MCDT*. [Online]

Available at: <https://globalcare.glantt.com/mcdt/>

[Acedido em 7 Abril 2021].

Glantt, s.d. g. *Pharma & Logistics*. [Online]

Available at: <https://globalcare.glantt.com/pharma-logistics/>

[Acedido em 7 Abril 2021].

Glantt, s.d. h. *Interfaces*. [Online]

Available at: <https://globalcare.glantt.com/interoperability/>

[Acedido em 7 Abril 2021].

Glantt, s.d. i. *Patient Engagement*. [Online]

Available at: <https://globalcare.glantt.com/patient-engagement/>

[Acedido em 7 Abril 2021].

Gray, J. & Reuter, A., 1992. *Transaction Processing: Concepts and Techniques*. s.l.:Elsevier.

Hospital da Luz, s.d.. *Anatomia patológica: o que é? | Hospital da Luz*. [Online]

Available at: <https://www.hospitaldaluz.pt/pt/guia-de-saude/dicionario-de->

saude/A/201/anatomia-patologica-o-que-e

[Acedido em 7 Abril 2021].

Hospital de Cascais, s.d. . *Triagem de Manchester*. [Online]

Available at: <https://www.hospitaldecascais.pt/pt/servicos/Paginas/triagemmanchester.aspx>

[Acedido em 7 Abril 2021].

IBM, 2014b. *What is a database management system?*. [Online]

Available at:

www.ibm.com/support/knowledgecenter/en/zosbasics/com.ibm.zos.zmidbmg/zmiddle_46.htm

[Acedido em 3 Janeiro 2021].

IBM, 2014. *Data definition language*. [Online]

Available at: www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/sqlp/rbafysqltech.htm

[Acedido em 6 Janeiro 2021].

IBM, 2020a. *Relational Databases*. [Online]

Available at: <https://www.ibm.com/cloud/learn/relational-databases>

[Acedido em 4 Janeiro 2021].

IBM, 2020b. *Relational database*. [Online]

Available at: <https://www.ibm.com/analytics/relational-database>

[Acedido em 4 Janeiro 2021].

IBM, 2020c. *SQL vs. NoSQL Databases: What's the Difference?*. [Online]

Available at: <https://www.ibm.com/cloud/blog/sql-vs-nosql>

[Acedido em 6 Janeiro 2021].

IBM, 2020d. *NoSQL Databases*. [Online]

Available at: <https://www.ibm.com/cloud/learn/nosql-databases>

[Acedido em 6 Janeiro 2021].

IBM, 2020e. *What is Three-Tier Architecture*. [Online]

Available at: <https://www.ibm.com/cloud/learn/three-tier-architecture>

[Acedido em 11 Janeiro 2021].

International, B2B, s.d. *What is the Value Proposition Canvas?*. [Online]

Available at: <https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/>

[Acedido em 20 12 2020].

Investopedia, 2020. *How the Wilcoxon Test Is Used*. [Online]

Available at: <https://www.investopedia.com/terms/w/wilcoxon-test.asp>

[Acedido em 24 junho 2021].

Kahraman, C., Ertay, T. & Büyüközkan, G., 2006. A fuzzy optimization model for QFD planning process using analytic network approach. *European Journal of Operational Research*, 1 06.

Kambalyal, C., 2010. 3-Tier Architecture. *Retrieved On*.

Kyte, T. et al., s.d.. *Data Concurrency and Consistency*. [Online]
Available at: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/data-concurrency-and-consistency.html#GUID-E8CBA9C5-58E3-460F-A82A-850E0152E95C>
[Acedido em 11 Janeiro 2021].

Kyte, T. et al., s.d.. *Introduction to Oracle Database*. [Online]
Available at: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/introduction-to-oracle-database.html#GUID-43F9DD5C-8D8C-4E61-A2B4-5C05907D3CEC>
[Acedido em 11 Janeiro 2021].

Kyte, T. et al., s.d.. *Oracle Relational Data Structures*. [Online]
Available at: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/tables-and-table-clusters.html#GUID-72E247B5-F39A-47F1-9445-72D9221F57E3>
[Acedido em 11 Janeiro 2021].

Kyte, T. et al., s.d.. *Tables and Table Clusters*. [Online]
Available at: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/tables-and-table-clusters.html#GUID-F845B1A7-71E3-4312-B66D-BC16C198ECE5>
[Acedido em 11 Janeiro 2021].

Leavitt, N., 2010. Will NoSQL Databases Live Up to Their Promise?. *Computer*, Fevereiro.

Lee, M.-C., 2014. Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. *Current Journal of Applied Science and Technology*, 2 Junho.

LibreOffice, s.d.. *Base | LibreOffice - Conjunto de programas de escritório*. [Online]
Available at: <https://pt.libreoffice.org/descobrir/base/>
[Acedido em 18 Abril 2021].

Lindič, J. & Marques da Silva, C., 2011. Value proposition as a catalyst for a customer focused innovation. *Management Decision*, 1 1.

Liu, B., Chen, H. & Junmei, H., 2020. *Design and Implementation of University Continuing Education Informatization Platform Based on SaaS Model*. s.l., s.n.

Lotfy, A. E., Saleh, A. I., El-Ghareeb, H. A. & Ali, H. A., 2016. A middle layer solution to support ACID properties for NoSQL databases. *Journal of King Saud University - Computer and Information Sciences*, 1 Janeiro.

MedicineOne, s.d. a. *Empresa - MedicineOne*. [Online]

Available at: <https://www.medicineone.net/empresa>

[Acedido em 9 Abril 2021].

MedicineOne, s.d. b. *MedicineOne*. [Online]

Available at: <https://www.medicineone.net/medicineone2>

[Acedido em 9 Abril 2021].

Melton, J., Bernus, P., Mertins, K. & Schmidt, G., 1998. Database Language SQL. Em: *Handbook on Architectures of Information Systems*. Berlin, Heidelberg: Springer.

Melton, J. & Simon, A. R., 1993. *Understanding the New SQL*. s.l.:Morgan Kaufmann.

Microsoft Azure, 2018. *Non-relational data and NoSQL - Azure Architecture Center*. [Online]

Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>

[Acedido em 6 Janeiro 2021].

Microsoft, s.d. *Add or remove add-ins in Excel*. [Online]

Available at: <https://support.microsoft.com/en-us/office/add-or-remove-add-ins-in-excel-0af570c4-5cf3-4fa9-9b88-403625a0b460>

[Acedido em 29 12 2020].

Miles, L. D., 2015. *Techniques of Value Analysis and Engineering*. s.l.:Miles Value Foundation.

Mohamed, M., Altrafi, O. & Ismail, M., 2014. Relational Vs. NoSQL databases: A survey.

International Journal of Computer and Information Technology (IJCIT), 1 Maio.

MongoDB, s.d. a. *What is NoSQL? NoSQL Databases Explained*. [Online]

Available at: <https://www.mongodb.com/nosql-explained>

[Acedido em 6 Janeiro 2021].

MongoDB, s.d. b. *What is a Key-Value Database?*. [Online]

Available at: <https://www.mongodb.com/key-value-database>

[Acedido em 6 Janeiro 2021].

MongoDB, s.d. c. *What Is A Non-Relational Database?*. [Online]

Available at: <https://www.mongodb.com/non-relational-database>

[Acedido em 6 Janeiro 2021].

MongoDB, s.d. d. *NoSQL vs SQL Databases*. [Online]

Available at: <https://www.mongodb.com/nosql-explained/nosql-vs-sql>

[Acedido em 7 Janeiro 2021].

Mozilla, 2020. *HTTP | MDN*. [Online]

Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP>

[Acedido em 21 Janeiro 2021].

Nicola, S., 2020. *Análise de Valor*. Porto: s.n.

Olewnik, A. & Lewis, K., 2008. Limitations of the House of Quality to provide quantitative design information. *International Journal of Quality & Reliability Management*, 1 1.

Oracle, s.d. a. *What is a relational database?*. [Online]

Available at: <https://www.oracle.com/database/what-is-a-relational-database/>
[Acedido em 3 Janeiro 2021].

Oracle, s.d. b. *What is a database?*. [Online]

Available at: <https://www.oracle.com/database/what-is-database/>
[Acedido em 1 Janeiro 2021].

Oracle, s.d. c. *Server-Side Programming: PL/SQL and Java*. [Online]

Available at:
https://docs.oracle.com/cd/E11882_01/server.112/e40540/srvrside.htm#CNCPT1758
[Acedido em 7 Janeiro 2021].

Oracle, s.d. e. *Database PL/SQL Language Reference*. [Online]

Available at:
https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/overview.htm#LNPLS143
[Acedido em 8 Janeiro 2021].

Oracle, s.d. f. *Database Development Guide*. [Online]

Available at: https://docs.oracle.com/database/121/ADFNS/adfns_odbc.htm#ADFNS1113
[Acedido em 20 Janeiro 2021].

Osterwalder, A., Pigneur, Y., Bernarda, G. & Smith, A., 2014. Value Proposition Design: How to Create Products and Services Customers Want. Em: *Value Proposition Design: How to Create Products and Services Customers Want*. s.l.:John Wiley & Sons.

Razali, N. M. & Wah, Y. B., 2011. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*.

Red Hat, s.d. . *What is an API?*. [Online]

Available at: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
[Acedido em 21 Janeiro 2021].

Russo, R. d. F. S. M. & Camanho, R., 2015. Criteria in AHP: A Systematic Review of Literature. *Procedia Computer Science*, 1 1.

S.A, P. I., 2021. *Dicionário Priberam*. [Online]

Available at: <https://dicionario.priberam.org/ambulatório>
[Acedido em 7 Abril 2021].

Saaty, T. L., 1988. *What is the Analytic Hierarchy Process?*. Berlin, Heidelberg, Springer.

Saaty, T. L., 2002. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*.

Schussel, G., 2001. *Client/Server: Past, Present and Future*. s.l., s.n.

Unilabs, 2020. *Citologia: o que é, que tipos existem e para que servem?*. [Online]
Available at: <https://blog.unilabs.pt/exames-medicos/citologia/>
[Acedido em 7 Abril 2021].

Visual Paradigm, s.d.. *Requirement Analysis Techniques*. [Online]
Available at: <https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/>
[Acedido em 13 Janeiro 2021].

Wang, M. & Wang, H., 2006. From process logic to business logic—A cognitive approach to business process management. *Information & Management*, 1 Março.

World Health Organization: WHO, 2020. *Coronavirus disease (COVID-19)*. [Online]
Available at: <https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19>
[Acedido em 21 12 2020].

Anexo A. Testes de tempo de resposta – criação de conteúdos

Neste anexo apresenta-se um excerto de código que representa a criação do teste de tempo de resposta da criação de conteúdos do tipo categoria de queixa (Extrato Código 16) e, por fim, figuras que apresentam os resultados dos 85 testes de tempo de resposta relativos à criação de conteúdos, nomeadamente, Figura 49, Figura 50, Figura 51, Figura 52 e Figura 53.

```

[Step0]
Name=Create Complaint Category
Enabled=True
UID=2
RepeatCount=1000
RepeatForEachParentRecord=False
RepeatTime=0
DelayBefore=0
DelayBeforeRandom=False
DelayAfter=0
DelayAfterRandom=False
RecordsToFetch=0
FetchAllRecords=True
RecordsPerFetch=25
FetchLOBs=True
SQL>-- Create Content
=DECLARE
=OUT_ID_COMPLAINT_CATEGORY content_func.pk_complaint_category.t_id_complaint_category;
=BEGIN
= content_func.pk_complaint_category.create_content(i_internal_name =>
'COMPLAINT_CATEGORY_TEST'||(:l_id+:l_base),
=             i_id_content => 'CNT.'||(:l_id+:l_base),
=             i_id_status => 1,
=             o_id_complaint_category => OUT_ID_COMPLAINT_CATEGORY);
=END;
Parent=-1
VariableCount=2

[Step0-Variable0]
Enabled=True
Name=l_id
DataType=Sequence within run
Parameters=

[Step0-Variable1]
Enabled=True
Name=l_base
DataType=Integer
Parameters=1000

```

Extrato Código 16 – Teste de tempo de resposta da funcionalidade de criação de conteúdos do tipo categoria de queixa

Steps

Create Complaint Category : Count = 1000, Avg = 1,75 ms, Min = 0,76 ms, Max = 0,22 sec

Create Complaint Alias : Count = 1000, Avg = 1,75 ms, Min = 0,95 ms, Max = 0,22 sec

Create Complaint : Count = 1000, Avg = 2,10 ms, Min = 1,17 ms, Max = 0,22 sec

Create Analysis Category : Count = 1000, Avg = 1,25 ms, Min = 0,74 ms, Max = 0,21 sec

Create Analysis Parameter : Count = 1000, Avg = 1,54 ms, Min = 0,81 ms, Max = 0,22 sec

Create Analysis : Count = 1000, Avg = 2,00 ms, Min = 1,24 ms, Max = 0,22 sec

Create Sample Type : Count = 1000, Avg = 1,76 ms, Min = 0,91 ms, Max = 0,22 sec

Create Sample Recipient : Count = 1000, Avg = 1,39 ms, Min = 0,93 ms, Max = 0,21 sec

Create Body Structure : Count = 1000, Avg = 1,13 ms, Min = 0,72 ms, Max = 0,20 sec

Create Analysis Sample Type : Count = 1000, Avg = 11,02 ms, Min = 9,51 ms, Max = 0,34 sec

Create Analysis Group : Count = 1000, Avg = 3,89 ms, Min = 2,11 ms, Max = 0,22 sec

Create Analysis Desc : Count = 1000, Avg = 2,63 ms, Min = 2,05 ms, Max = 0,21 sec

Create Analysis Unit Measure : Count = 1000, Avg = 4,43 ms, Min = 2,80 ms, Max = 0,33 sec

Create Analysis Res Calculator : Count = 1000, Avg = 6,46 ms, Min = 5,07 ms, Max = 0,23 sec

Create Analysis Specimen Condition : Count = 1000, Avg = 1,49 ms, Min = 0,75 ms, Max = 0,21 sec

Create Exam Category : Count = 1000, Avg = 1,41 ms, Min = 0,77 ms, Max = 0,21 sec

Create Exam : Count = 1000, Avg = 3,53 ms, Min = 1,95 ms, Max = 0,22 sec

Create Exam Group : Count = 1000, Avg = 6,15 ms, Min = 4,83 ms, Max = 0,33 sec

Create Clinical Service : Count = 1000, Avg = 6,27 ms, Min = 4,71 ms, Max = 0,34 sec

Figura 49 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos

Steps

Create Hemo Type : Count = 1000, Avg = 1,75 ms, Min = 0,73 ms, Max = 0,21 sec

Create Hemo Type Analysis : Count = 1000, Avg = 8,96 ms, Min = 8,02 ms, Max = 0,23 sec

Create Discharge Dest : Count = 1000, Avg = 1,63 ms, Min = 0,74 ms, Max = 0,21 sec

Create Transport Entity : Count = 1000, Avg = 1,00 ms, Min = 0,76 ms, Max = 45,24 ms

Create Discharge Reason : Count = 1000, Avg = 6,34 ms, Min = 5,25 ms, Max = 0,23 sec

Create Discharge Reason Dest : Count = 1000, Avg = 9,57 ms, Min = 8,24 ms, Max = 0,23 sec

Create Diet : Count = 1000, Avg = 7,25 ms, Min = 5,37 ms, Max = 0,33 sec

Create Questionnaire : Count = 1000, Avg = 2,43 ms, Min = 0,91 ms, Max = 0,22 sec

Create Response : Count = 1000, Avg = 1,54 ms, Min = 0,94 ms, Max = 0,22 sec

Create Questionnaire Response : Count = 1000, Avg = 9,40 ms, Min = 8,08 ms, Max = 0,24 sec

Create Exam Questionnaire : Count = 1000, Avg = 14,67 ms, Min = 12,18 ms, Max = 0,35 sec

Create Analysis Questionnaire : Count = 1000, Avg = 20,24 ms, Min = 15,91 ms, Max = 0,36 sec

Create Blood Products Questionnaire : Count = 1000, Avg = 14,48 ms, Min = 12,38 ms, Max = 0,24 sec

Create Po Param : Count = 1000, Avg = 2,44 ms, Min = 1,74 ms, Max = 0,21 sec

Create Po Param Sets : Count = 1000, Avg = 5,38 ms, Min = 4,70 ms, Max = 82,11 ms

Create Po Param Um : Count = 1000, Avg = 2,17 ms, Min = 1,87 ms, Max = 50,61 ms

Create Po Param Mc : Count = 1000, Avg = 2,27 ms, Min = 1,20 ms, Max = 0,22 sec

Create Discharge Instructions Group : Count = 1000, Avg = 1,90 ms, Min = 0,73 ms, Max = 0,22 sec

Create Discharge Instructions : Count = 1000, Avg = 5,44 ms, Min = 4,54 ms, Max = 0,22 sec

Figura 50 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 1)

Steps

Create Health Plan Entity : Count = 1000, Avg = 0,99 ms, Min = 0,74 ms, Max = 55,89 ms

Create Health Plan : Count = 1000, Avg = 5,00 ms, Min = 4,46 ms, Max = 95,62 ms

Create Task Goal : Count = 1000, Avg = 1,06 ms, Min = 0,73 ms, Max = 58,08 ms

Create Intervention Plan : Count = 1000, Avg = 2,80 ms, Min = 2,18 ms, Max = 0,22 sec

Create Hidrics Type : Count = 1000, Avg = 1,64 ms, Min = 0,75 ms, Max = 0,21 sec

Create Hidrics Occurs Type : Count = 1000, Avg = 1,19 ms, Min = 0,74 ms, Max = 0,21 sec

Create Hidrics : Count = 1000, Avg = 3,64 ms, Min = 2,66 ms, Max = 0,22 sec

Create Way : Count = 1000, Avg = 1,21 ms, Min = 0,73 ms, Max = 0,21 sec

Create Hidrics Charact : Count = 1000, Avg = 1,46 ms, Min = 0,75 ms, Max = 0,20 sec

Create Hidrics Device : Count = 1000, Avg = 1,26 ms, Min = 0,79 ms, Max = 0,21 sec

Create Hidrics Configurations : Count = 1000, Avg = 1,07 ms, Min = 0,63 ms, Max = 0,20 sec

Create Hidrics Location : Count = 1000, Avg = 1,96 ms, Min = 1,22 ms, Max = 0,21 sec

Create Hidrics Way Rel : Count = 1000, Avg = 13,41 ms, Min = 10,88 ms, Max = 45,16 ms

Create Hidrics Device Rel : Count = 1000, Avg = 13,67 ms, Min = 12,05 ms, Max = 0,25 sec

Create Hidrics Charact Rel : Count = 1000, Avg = 13,46 ms, Min = 11,79 ms, Max = 0,25 sec

Create Hidrics Location Rel : Count = 1000, Avg = 9,28 ms, Min = 7,99 ms, Max = 0,23 sec

Create Supply Type : Count = 1000, Avg = 1,39 ms, Min = 0,75 ms, Max = 0,22 sec

Create Supply : Count = 1000, Avg = 5,77 ms, Min = 5,05 ms, Max = 0,22 sec

Create Supply Reason : Count = 1000, Avg = 0,96 ms, Min = 0,74 ms, Max = 26,48 ms

Figura 51 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 2)

Steps

Create Supply Relation : Count = 1000, Avg = 5,73 ms, Min = 5,14 ms, Max = 0,23 sec

Create Supply Context : Count = 1000, Avg = 3,00 ms, Min = 1,54 ms, Max = 0,22 sec

Create Positioning : Count = 1000, Avg = 1,55 ms, Min = 0,74 ms, Max = 0,23 sec

Create Checklist : Count = 1000, Avg = 1,04 ms, Min = 0,65 ms, Max = 0,21 sec

Create Checklist Version : Count = 1000, Avg = 4,65 ms, Min = 4,06 ms, Max = 0,22 sec

Create Checklist Item : Count = 1000, Avg = 1,97 ms, Min = 1,24 ms, Max = 0,21 sec

Create Checklist Item Dep : Count = 1000, Avg = 1,45 ms, Min = 1,00 ms, Max = 0,22 sec

Create Checklist Profile Template : Count = 1000, Avg = 1,38 ms, Min = 0,95 ms, Max = 0,21 sec

Create Habit Characterization : Count = 1000, Avg = 1,64 ms, Min = 0,74 ms, Max = 0,22 sec

Create Habit : Count = 1000, Avg = 2,47 ms, Min = 1,74 ms, Max = 0,21 sec

Create Analysis Collection : Count = 1000, Avg = 6,56 ms, Min = 4,89 ms, Max = 0,34 sec

Create Analysis Collection Int : Count = 1000, Avg = 6,38 ms, Min = 4,59 ms, Max = 0,23 sec

Create Supply Sup Area : Count = 1000, Avg = 1,68 ms, Min = 1,25 ms, Max = 0,20 sec

Create Supply Loc Default : Count = 1000, Avg = 1,48 ms, Min = 1,27 ms, Max = 37,08 ms

Create Hidrics Occurs Type Rel : Count = 1000, Avg = 9,19 ms, Min = 8,00 ms, Max = 0,24 sec

Create Speciality : Count = 1000, Avg = 1,40 ms, Min = 0,73 ms, Max = 0,21 sec

Create Necessity : Count = 1000, Avg = 1,56 ms, Min = 0,80 ms, Max = 0,22 sec

Create Occupation : Count = 1000, Avg = 1,56 ms, Min = 0,72 ms, Max = 0,21 sec

Create Result Notes : Count = 1000, Avg = 1,03 ms, Min = 0,73 ms, Max = 42,38 ms

Figura 52 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 3)

Create Origin : Count = 1000, Avg = 1,47 ms, Min = 0,79 ms, Max = 0,23 sec
Create Rehab Area : Count = 1000, Avg = 1,16 ms, Min = 0,72 ms, Max = 0,21 sec
Create Rehab Session Type : Count = 1000, Avg = 1,19 ms, Min = 0,74 ms, Max = 0,21 sec
Create Codification : Count = 1000, Avg = 1,40 ms, Min = 0,87 ms, Max = 0,21 sec
Create Intervention Category : Count = 1000, Avg = 0,92 ms, Min = 0,73 ms, Max = 31,81 ms
Create Intervention : Count = 1000, Avg = 4,95 ms, Min = 2,86 ms, Max = 0,33 sec
Create Rehab Area Interv : Count = 1000, Avg = 1,12 ms, Min = 0,93 ms, Max = 24,12 ms
Create Rehab Area Interv Session Type : Count = 1000, Avg = 1,82 ms, Min = 0,90 ms, Max = 0,22 sec
Create Intervention Codification : Count = 1000, Avg = 9,36 ms, Min = 7,94 ms, Max = 0,24 sec
Create Intervention Questionnaire : Count = 1000, Avg = 13,94 ms, Min = 12,25 ms, Max = 0,35 sec

Figura 53 - Resultados dos testes de tempo de resposta das funcionalidades de criação de conteúdos (continuação - 4)

Anexo B. Testes de tempo de resposta – atualização de conteúdos

Neste anexo apresenta-se um excerto de código que representa o teste de tempo de resposta da alteração de conteúdos do tipo categoria de queixa (Extrato Código 17) e, por fim, figuras que apresentam os resultados dos 85 testes de tempo de resposta relativos à atualização de conteúdos, nomeadamente, Figura 54, Figura 55, Figura 56, Figura 57 e Figura 58.

```

[Step0]
Name=Update Complaint Category
Enabled=True
UID=2
RepeatCount=1000
RepeatForEachParentRecord=False
RepeatTime=0
DelayBefore=0
DelayBeforeRandom=False
DelayAfter=0
DelayAfterRandom=False
RecordsToFetch=0
FetchAllRecords=True
RecordsPerFetch=25
FetchLOBs=True
SQL=DECLARE
=i_id_complaint_category number;
=i_id_content varchar2(30);
=BEGIN
= i_id_content := 'CNT.'||(:i_id+:i_base);
=
= SELECT id_complaint_category
= INTO i_id_complaint_category
= FROM content_data.complaint_category cc
= where cc.id_content = i_id_content;
=
= content_func.pk_complaint_category.update_content(i_id_complaint_category =>
i_id_complaint_category,
=
=           i_id_content      => i_id_content,
=           i_internal_name   => '_COMPLAINT_CATEGORY_TEST_'||(:i_id+:i_base)
=           /*i_id_status     => 2*/);
=END;
Parent=-1
VariableCount=2

[Step0-Variable0]
Enabled=True
Name=i_id
DataType=Sequence within run
Parameters=

[Step0-Variable1]
Enabled=True
Name=i_base
DataType=Integer
Parameters=1000

```

Extrato Código 17 - Teste de tempo de resposta da funcionalidade de atualização de conteúdos do tipo categoria de queixa

Steps

Update Complaint Category : Count = 1000, Avg = 1,49 ms, Min = 0,69 ms, Max = 0,30 sec

Update Complaint Alias : Count = 1000, Avg = 0,77 ms, Min = 0,61 ms, Max = 43,00 ms

Update Complaint : Count = 1000, Avg = 2,18 ms, Min = 1,58 ms, Max = 0,34 sec

Update Analysis Category : Count = 1000, Avg = 2,00 ms, Min = 1,01 ms, Max = 0,35 sec

Update Analysis Parameter : Count = 1000, Avg = 0,94 ms, Min = 0,64 ms, Max = 88,41 ms

Update Analysis : Count = 1000, Avg = 1,69 ms, Min = 0,75 ms, Max = 0,31 sec

Update Sample Type : Count = 1000, Avg = 1,73 ms, Min = 0,91 ms, Max = 0,31 sec

Update Sample Recipient : Count = 1000, Avg = 1,66 ms, Min = 0,81 ms, Max = 0,31 sec

Update Body Structure : Count = 1000, Avg = 0,87 ms, Min = 0,65 ms, Max = 44,83 ms

Update Analysis Sample Type : Count = 1000, Avg = 4,80 ms, Min = 3,04 ms, Max = 0,35 sec

Update Analysis Group : Count = 1000, Avg = 1,87 ms, Min = 0,94 ms, Max = 0,31 sec

Update Analysis Desc : Count = 1000, Avg = 2,20 ms, Min = 0,68 ms, Max = 0,31 sec

Update Analysis Unit Measure : Count = 1000, Avg = 9,15 ms, Min = 7,06 ms, Max = 0,36 sec

Update Analysis Res Calculator : Count = 1000, Avg = 1,98 ms, Min = 1,04 ms, Max = 0,32 sec

Update Analysis Specimen Condition : Count = 1000, Avg = 1,52 ms, Min = 0,72 ms, Max = 0,31 sec

Update Exam Category : Count = 1000, Avg = 1,68 ms, Min = 0,64 ms, Max = 0,31 sec

Update Exam : Count = 1000, Avg = 3,77 ms, Min = 2,15 ms, Max = 0,35 sec

Update Exam Group : Count = 1000, Avg = 1,35 ms, Min = 0,95 ms, Max = 56,95 ms

Update Clinical Service : Count = 1000, Avg = 1,54 ms, Min = 0,67 ms, Max = 0,31 sec

Figura 54 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos

Steps

Update Hemo Type : Count = 1000, Avg = 1,00 ms, Min = 0,64 ms, Max = 85,78 ms

Update Hemo Type Analysis : Count = 1000, Avg = 9,99 ms, Min = 8,41 ms, Max = 0,32 sec

Update Discharge Dest : Count = 1000, Avg = 0,80 ms, Min = 0,65 ms, Max = 27,96 ms

Update Transport Entity : Count = 1000, Avg = 0,81 ms, Min = 0,63 ms, Max = 47,72 ms

Update Discharge Reason : Count = 1000, Avg = 3,70 ms, Min = 2,10 ms, Max = 0,31 sec

Update Discharge Reason Dest : Count = 1000, Avg = 9,92 ms, Min = 8,16 ms, Max = 0,33 sec

Update Diet : Count = 1000, Avg = 2,56 ms, Min = 1,34 ms, Max = 0,31 sec

Update Questionnaire : Count = 1000, Avg = 1,57 ms, Min = 0,97 ms, Max = 0,31 sec

Update Response : Count = 1000, Avg = 1,34 ms, Min = 0,92 ms, Max = 67,32 ms

Update Questionnaire Response : Count = 1000, Avg = 9,56 ms, Min = 8,09 ms, Max = 0,32 sec

Update Exam Questionnaire : Count = 1000, Avg = 7,27 ms, Min = 5,73 ms, Max = 0,32 sec

Update Analysis Questionnaire : Count = 1000, Avg = 2,35 ms, Min = 1,01 ms, Max = 0,39 sec

Update Blood Products Questionnaire : Count = 1000, Avg = 8,57 ms, Min = 5,81 ms, Max = 0,32 sec

Update Intervention Questionnaire : Count = 1000, Avg = 3,08 ms, Min = 0,96 ms, Max = 0,36 sec

Update Po Param : Count = 1000, Avg = 3,16 ms, Min = 1,62 ms, Max = 0,44 sec

Update Po Param Sets : Count = 1000, Avg = 5,93 ms, Min = 4,71 ms, Max = 0,17 sec

Update Po Param Um : Count = 1000, Avg = 3,59 ms, Min = 2,14 ms, Max = 0,52 sec

Update Po Param Mc : Count = 1000, Avg = 1,69 ms, Min = 1,06 ms, Max = 0,31 sec

Update Discharge Instructions Group : Count = 1000, Avg = 2,71 ms, Min = 0,67 ms, Max = 0,31 sec

Figura 55 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 1)

Steps

Update Discharge Instructions : Count = 1000, Avg = 1,41 ms, Min = 0,60 ms, Max = 0,32 sec

Update Health Plan Entity : Count = 1000, Avg = 1,26 ms, Min = 0,65 ms, Max = 0,33 sec

Update Health Plan : Count = 1000, Avg = 1,58 ms, Min = 0,67 ms, Max = 0,31 sec

Update Task Goal : Count = 1000, Avg = 1,20 ms, Min = 0,68 ms, Max = 0,31 sec

Update Intervention Plan : Count = 1000, Avg = 2,75 ms, Min = 1,50 ms, Max = 0,31 sec

Update Hidrics Type : Count = 1000, Avg = 1,87 ms, Min = 0,68 ms, Max = 0,31 sec

Update Hidrics Occurs Type : Count = 1000, Avg = 0,85 ms, Min = 0,66 ms, Max = 50,39 ms

Update Hidrics : Count = 1000, Avg = 3,03 ms, Min = 2,29 ms, Max = 0,31 sec

Update Way : Count = 1000, Avg = 1,46 ms, Min = 0,64 ms, Max = 0,31 sec

Update Hidrics Charact : Count = 1000, Avg = 2,04 ms, Min = 0,65 ms, Max = 0,91 sec

Update Hidrics Device : Count = 1000, Avg = 1,13 ms, Min = 0,63 ms, Max = 0,30 sec

Update Hidrics Configurations : Count = 1000, Avg = 0,74 ms, Min = 0,61 ms, Max = 24,90 ms

Update Hidrics Location : Count = 1000, Avg = 0,70 ms, Min = 0,50 ms, Max = 24,90 ms

Update Hidrics Way Rel : Count = 1000, Avg = 13,40 ms, Min = 10,11 ms, Max = 0,32 sec

Update Hidrics Device Rel : Count = 1000, Avg = 13,41 ms, Min = 11,70 ms, Max = 0,32 sec

Update Hidrics Charact Rel : Count = 1000, Avg = 12,68 ms, Min = 11,52 ms, Max = 0,16 sec

Update Hidrics Location Rel : Count = 1000, Avg = 1,77 ms, Min = 0,59 ms, Max = 0,34 sec

Update Supply Type : Count = 1000, Avg = 0,79 ms, Min = 0,64 ms, Max = 26,84 ms

Update Supply : Count = 1000, Avg = 6,54 ms, Min = 4,81 ms, Max = 0,32 sec

Figura 56 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 2)

Steps

Update Supply Reason : Count = 1000, Avg = 1,19 ms, Min = 0,70 ms, Max = 0,31 sec

Update Supply Relation : Count = 1000, Avg = 5,80 ms, Min = 4,91 ms, Max = 87,07 ms

Update Supply Context : Count = 1000, Avg = 3,91 ms, Min = 1,62 ms, Max = 0,31 sec

Update Positioning : Count = 1000, Avg = 1,13 ms, Min = 0,64 ms, Max = 0,30 sec

Update Checklist : Count = 1000, Avg = 1,16 ms, Min = 0,65 ms, Max = 0,30 sec

Update Checklist Version : Count = 1000, Avg = 5,15 ms, Min = 0,00 ms, Max = 0,31 sec

Update Checklist Item : Count = 1000, Avg = 1,66 ms, Min = 0,00 ms, Max = 0,31 sec

Update Checklist Item Dep : Count = 1000, Avg = 0,81 ms, Min = 0,00 ms, Max = 45,83 ms

Update Checklist Profile Template : Count = 1000, Avg = 1,70 ms, Min = 0,00 ms, Max = 0,32 sec

Update Habit Characterization : Count = 1000, Avg = 0,94 ms, Min = 0,65 ms, Max = 74,95 ms

Update Habit : Count = 1000, Avg = 2,75 ms, Min = 1,40 ms, Max = 0,31 sec

Update Analysis Collection : Count = 1000, Avg = 1,70 ms, Min = 1,09 ms, Max = 0,30 sec

Update Analysis Collection Int : Count = 1000, Avg = 1,55 ms, Min = 0,93 ms, Max = 0,31 sec

Update Supply Sup Area : Count = 1000, Avg = 1,55 ms, Min = 0,89 ms, Max = 0,31 sec

Update Supply Loc Default : Count = 1000, Avg = 1,84 ms, Min = 0,94 ms, Max = 0,31 sec

Update Hidrics Occurs Type Rel : Count = 1000, Avg = 0,79 ms, Min = 0,58 ms, Max = 32,16 ms

Update Speciality : Count = 1000, Avg = 0,90 ms, Min = 0,65 ms, Max = 44,01 ms

Update Necessity : Count = 1000, Avg = 1,04 ms, Min = 0,70 ms, Max = 0,12 sec

Update Occupation : Count = 1000, Avg = 1,54 ms, Min = 0,63 ms, Max = 0,33 sec

Figura 57 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 3)

Update Result Notes : Count = 1000, Avg = 1,09 ms, Min = 0,63 ms, Max = 0,31 sec
Update Origin : Count = 1000, Avg = 1,72 ms, Min = 0,65 ms, Max = 0,31 sec
Update Rehab Area : Count = 1000, Avg = 2,22 ms, Min = 0,56 ms, Max = 0,31 sec
Update Rehab Session Type : Count = 1000, Avg = 1,31 ms, Min = 0,57 ms, Max = 0,32 sec
Update Codification : Count = 1000, Avg = 1,46 ms, Min = 0,68 ms, Max = 0,31 sec
Update Intervention Category : Count = 1000, Avg = 1,35 ms, Min = 0,56 ms, Max = 0,31 sec
Update Intervention : Count = 1000, Avg = 2,14 ms, Min = 1,57 ms, Max = 0,32 sec
Update Rehab Area Interv : Count = 1000, Avg = 0,69 ms, Min = 0,60 ms, Max = 11,32 ms
Update Rehab Area Interv Session Type : Count = 1000, Avg = 0,76 ms, Min = 0,63 ms, Max = 11,92 ms
Update Intervention Codification : Count = 1000, Avg = 2,01 ms, Min = 0,61 ms, Max = 0,32 sec

Figura 58 - Resultados dos testes de tempo de resposta das funcionalidades de atualização de conteúdos (continuação – 4)

Anexo C. Testes de tempo de resposta – atualização do estado de conteúdos

Neste anexo apresenta-se um excerto de código que representa o teste de tempo de resposta da alteração do estado de conteúdos do tipo categoria de queixa (Extrato Código 18) e, por fim, figuras que apresentam os resultados dos 85 testes de tempo de resposta relativos à atualização do estado de conteúdos, nomeadamente, Figura 59, Figura 60, Figura 61, Figura 62 e Figura 63.

```

[Step0]
Name=Set Content Status Complaint Category
Enabled=True
UID=2
RepeatCount=1000
RepeatForEachParentRecord=False
RepeatTime=0
DelayBefore=0
DelayBeforeRandom=False
DelayAfter=0
DelayAfterRandom=False
RecordsToFetch=0
FetchAllRecords=True
RecordsPerFetch=25
FetchLOBs=True
SQL=DECLARE
=i_id_complaint_category number;
=i_id_content varchar2(30);
=BEGIN
= i_id_content := 'CNT.' || (:i_id+:i_base);
=
= SELECT id_complaint_category
= INTO i_id_complaint_category
= FROM content_data.complaint_category cc
= where cc.id_content = i_id_content;
=
= content_func.pk_complaint_category.set_content_status(i_id_complaint_category =>
i_id_complaint_category, i_id_status => 4);
=END;
Parent=-1
VariableCount=2

[Step0-Variable0]
Enabled=True
Name=i_id
DataType=Sequence within run
Parameters=

[Step0-Variable1]
Enabled=True
Name=i_base
DataType=Integer
Parameters=1000

```

Extrato Código 18 - Teste de tempo de resposta da funcionalidade de atualização do estado de conteúdos do tipo categoria de queixa

Steps

Set Content Status Complaint Category : Count = 1000, Avg = 0,96 ms, Min = 0,45 ms, Max = 0,31 sec

Set Content Status Complaint Alias : Count = 1000, Avg = 0,60 ms, Min = 0,45 ms, Max = 12,16 ms

Set Content Status Complaint : Count = 1000, Avg = 0,90 ms, Min = 0,45 ms, Max = 0,31 sec

Set Content Status Analysis Category : Count = 1000, Avg = 0,66 ms, Min = 0,50 ms, Max = 17,24 ms

Set Content Status Analysis Parameter : Count = 1000, Avg = 0,61 ms, Min = 0,45 ms, Max = 12,66 ms

Set Content Status Analysis : Count = 1000, Avg = 0,61 ms, Min = 0,44 ms, Max = 9,42 ms

Set Content Status Sample Type : Count = 1000, Avg = 1,26 ms, Min = 0,44 ms, Max = 0,32 sec

Set Content Status Sample Recipient : Count = 1000, Avg = 0,74 ms, Min = 0,51 ms, Max = 91,29 ms

Set Content Status Body Structure : Count = 1000, Avg = 0,60 ms, Min = 0,44 ms, Max = 10,35 ms

Set Content Status Analysis Sample Type : Count = 1000, Avg = 0,90 ms, Min = 0,44 ms, Max = 0,31 sec

Set Content Status Analysis Group : Count = 1000, Avg = 0,91 ms, Min = 0,44 ms, Max = 0,32 sec

Set Content Status Analysis Desc : Count = 1000, Avg = 0,89 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Analysis Unit Measure : Count = 1000, Avg = 1,07 ms, Min = 0,56 ms, Max = 0,32 sec

Set Content Status Analysis Res Calculator : Count = 1000, Avg = 0,93 ms, Min = 0,46 ms, Max = 0,30 sec

Set Content Status Analysis Specimen Condition : Count = 1000, Avg = 0,63 ms, Min = 0,47 ms, Max = 12,37 ms

Set Content Status Exam Category : Count = 1000, Avg = 0,98 ms, Min = 0,56 ms, Max = 0,31 sec

Set Content Status Exam : Count = 1000, Avg = 0,85 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Exam Group : Count = 1000, Avg = 0,51 ms, Min = 0,42 ms, Max = 11,46 ms

Set Content Status Clinical Service : Count = 1000, Avg = 1,14 ms, Min = 0,42 ms, Max = 0,31 sec

Figura 59 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos

Steps

Set Content Status Hemo Type : Count = 1000, Avg = 0,52 ms, Min = 0,43 ms, Max = 9,54 ms

Set Content Status Hemo Type Analysis : Count = 1000, Avg = 0,72 ms, Min = 0,57 ms, Max = 13,33 ms

Set Content Status Discharge Dest : Count = 1000, Avg = 0,88 ms, Min = 0,44 ms, Max = 0,32 sec

Set Content Status Transport Entity : Count = 1000, Avg = 0,57 ms, Min = 0,44 ms, Max = 12,74 ms

Set Content Status Discharge Reason : Count = 1000, Avg = 0,53 ms, Min = 0,43 ms, Max = 10,32 ms

Set Content Status Discharge Reason Dest : Count = 1000, Avg = 0,66 ms, Min = 0,54 ms, Max = 12,93 ms

Set Content Status Diet : Count = 1000, Avg = 0,54 ms, Min = 0,42 ms, Max = 12,98 ms

Set Content Status Questionnaire : Count = 1000, Avg = 1,15 ms, Min = 0,43 ms, Max = 0,31 sec

Set Content Status Response : Count = 1000, Avg = 0,52 ms, Min = 0,43 ms, Max = 9,20 ms

Set Content Status Questionnaire Response : Count = 1000, Avg = 1,14 ms, Min = 0,43 ms, Max = 0,31 sec

Set Content Status Exam Questionnaire : Count = 1000, Avg = 0,66 ms, Min = 0,52 ms, Max = 13,51 ms

Set Content Status Analysis Questionnaire : Count = 1000, Avg = 0,90 ms, Min = 0,47 ms, Max = 0,32 sec

Set Content Status Blood Products Questionnaire : Count = 1000, Avg = 1,24 ms, Min = 0,54 ms, Max = 0,30 sec

Set Content Status Intervention Questionnaire : Count = 1000, Avg = 1,88 ms, Min = 0,52 ms, Max = 0,32 sec

Set Content Status Po Param : Count = 1000, Avg = 0,55 ms, Min = 0,43 ms, Max = 12,49 ms

Set Content Status Po Param Sets : Count = 1000, Avg = 0,54 ms, Min = 0,44 ms, Max = 12,11 ms

Set Content Status Po Param Um : Count = 1000, Avg = 1,24 ms, Min = 0,52 ms, Max = 0,31 sec

Set Content Status Po Param Mc : Count = 1000, Avg = 0,52 ms, Min = 0,42 ms, Max = 11,69 ms

Set Content Status Discharge Instructions Group : Count = 1000, Avg = 0,85 ms, Min = 0,42 ms, Max = 0,31 sec

Figura 60 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 1)

Steps

Set Content Status Discharge Instructions : Count = 1000, Avg = 0,54 ms, Min = 0,42 ms, Max = 15,97 ms

Set Content Status Health Plan Entity : Count = 1000, Avg = 0,55 ms, Min = 0,43 ms, Max = 18,19 ms

Set Content Status Health Plan : Count = 1000, Avg = 0,85 ms, Min = 0,43 ms, Max = 0,30 sec

Set Content Status Task Goal : Count = 1000, Avg = 0,81 ms, Min = 0,42 ms, Max = 0,30 sec

Set Content Status Intervention Plan : Count = 1000, Avg = 0,82 ms, Min = 0,43 ms, Max = 0,30 sec

Set Content Status Hidrics Type : Count = 1000, Avg = 0,82 ms, Min = 0,42 ms, Max = 0,30 sec

Set Content Status Hidrics Occurs Type : Count = 1000, Avg = 0,83 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Hidrics : Count = 1000, Avg = 1,45 ms, Min = 0,42 ms, Max = 0,32 sec

Set Content Status Way : Count = 1000, Avg = 0,50 ms, Min = 0,42 ms, Max = 9,82 ms

Set Content Status Hidrics Charact : Count = 1000, Avg = 0,81 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Hidrics Device : Count = 1000, Avg = 0,52 ms, Min = 0,43 ms, Max = 9,24 ms

Set Content Status Hidrics Configurations : Count = 1000, Avg = 0,81 ms, Min = 0,41 ms, Max = 0,30 sec

Set Content Status Hidrics Location : Count = 1000, Avg = 0,82 ms, Min = 0,40 ms, Max = 0,31 sec

Set Content Status Hidrics Way Rel : Count = 1000, Avg = 0,55 ms, Min = 0,45 ms, Max = 13,43 ms

Set Content Status Hidrics Device Rel : Count = 1000, Avg = 0,63 ms, Min = 0,51 ms, Max = 12,31 ms

Set Content Status Hidrics Charact Rel : Count = 1000, Avg = 0,88 ms, Min = 0,45 ms, Max = 0,31 sec

Set Content Status Hidrics Location Rel : Count = 1000, Avg = 0,52 ms, Min = 0,44 ms, Max = 12,54 ms

Set Content Status Supply Type : Count = 1000, Avg = 0,51 ms, Min = 0,42 ms, Max = 9,29 ms

Set Content Status Supply : Count = 1000, Avg = 0,51 ms, Min = 0,42 ms, Max = 9,44 ms

Figura 61 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 2)

Steps

Set Content Status Supply Reason : Count = 1000, Avg = 0,52 ms, Min = 0,42 ms, Max = 11,00 ms

Set Content Status Supply Relation : Count = 1000, Avg = 0,84 ms, Min = 0,45 ms, Max = 0,30 sec

Set Content Status Supply Context : Count = 1000, Avg = 0,65 ms, Min = 0,51 ms, Max = 13,01 ms

Set Content Status Positioning : Count = 1000, Avg = 0,88 ms, Min = 0,44 ms, Max = 0,30 sec

Set Content Status Checklist : Count = 1000, Avg = 0,86 ms, Min = 0,43 ms, Max = 0,31 sec

Set Content Status Checklist Version : Count = 1000, Avg = 0,62 ms, Min = 0,00 ms, Max = 5,44 ms

Set Content Status Checklist Item : Count = 1000, Avg = 1,16 ms, Min = 0,00 ms, Max = 0,31 sec

Set Content Status Checklist Item Dep : Count = 1000, Avg = 0,54 ms, Min = 0,00 ms, Max = 2,62 ms

Set Content Status Checklist Profile Template : Count = 1000, Avg = 1,48 ms, Min = 0,00 ms, Max = 0,32 sec

Set Content Status Habit Characterization : Count = 1000, Avg = 0,53 ms, Min = 0,43 ms, Max = 16,44 ms

Set Content Status Habit : Count = 1000, Avg = 0,52 ms, Min = 0,42 ms, Max = 6,98 ms

Set Content Status Analysis Collection : Count = 1000, Avg = 1,35 ms, Min = 0,53 ms, Max = 0,32 sec

Set Content Status Analysis Collection Int : Count = 1000, Avg = 1,52 ms, Min = 0,78 ms, Max = 0,30 sec

Set Content Status Supply Sup Area : Count = 1000, Avg = 0,60 ms, Min = 0,48 ms, Max = 6,46 ms

Set Content Status Supply Loc Default : Count = 1000, Avg = 0,95 ms, Min = 0,53 ms, Max = 0,31 sec

Set Content Status Hidrics Occurs Type Rel : Count = 1000, Avg = 0,88 ms, Min = 0,45 ms, Max = 0,31 sec

Set Content Status Speciality : Count = 1000, Avg = 0,84 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Necessity : Count = 1000, Avg = 1,14 ms, Min = 0,42 ms, Max = 0,31 sec

Set Content Status Occupation : Count = 1000, Avg = 1,14 ms, Min = 0,43 ms, Max = 0,30 sec

Figura 62 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 3)

Set Content Status Result Notes : Count = 1000, Avg = 0,83 ms, Min = 0,43 ms, Max = 0,31 sec

Set Content Status Origin : Count = 1000, Avg = 0,83 ms, Min = 0,42 ms, Max = 0,30 sec

Set Content Status Rehab Area : Count = 1000, Avg = 0,82 ms, Min = 0,42 ms, Max = 0,30 sec

Set Content Status Rehab Session Type : Count = 1000, Avg = 0,51 ms, Min = 0,42 ms, Max = 6,11 ms

Set Content Status Codification : Count = 1000, Avg = 0,50 ms, Min = 0,42 ms, Max = 5,94 ms

Set Content Status Intervention Category : Count = 1000, Avg = 0,83 ms, Min = 0,43 ms, Max = 0,31 sec

Set Content Status Intervention : Count = 1000, Avg = 0,51 ms, Min = 0,43 ms, Max = 7,56 ms

Set Content Status Rehab Area Interv : Count = 1000, Avg = 0,90 ms, Min = 0,50 ms, Max = 0,31 sec

Set Content Status Rehab Area Interv Session Type : Count = 1000, Avg = 1,26 ms, Min = 0,54 ms, Max = 0,31 sec

Set Content Status Intervention Codification : Count = 1000, Avg = 1,50 ms, Min = 0,50 ms, Max = 0,31 sec

Figura 63 - Resultados dos testes de tempo de resposta das funcionalidades de atualização do estado de conteúdos (continuação – 4)

Anexo D. Testes de tempo de resposta – obtenção dos dados de conteúdo

Neste anexo apresenta-se um excerto de código que representa o teste de tempo de resposta da obtenção de conteúdos do tipo categoria de queixa (Extrato Código 19) e, por fim, figuras que apresentam os resultados dos 87 testes de tempo de resposta relativos à obtenção dos dados de conteúdos, nomeadamente, Figura 64, Figura 65, Figura 66, Figura 67 e Figura 68.

```
[Step2]
Name=Get Content Complaint Category
Enabled=True
UID=3
RepeatCount=100
RepeatForEachParentRecord=False
RepeatTime=0
DelayBefore=0
DelayBeforeRandom=False
DelayAfter=0
DelayAfterRandom=False
RecordsToFetch=0
FetchAllRecords=True
RecordsPerFetch=25
FetchLOBs=True
SQL=select (1) from content_func.PK_COMPLAINT_CATEGORY.get_content(i_id_complaint_category =>
10, i_ids_complaint_category => null);
Parent=-1
VariableCount=0
```

Extrato Código 19 - Teste de tempo de resposta da funcionalidade de obtenção de dados de conteúdos do tipo categoria de queixa

Steps

Get Content Complaint : Count = 100, Avg = 0,14 sec, Min = 0,14 sec, Max = 0,37 sec

Get Content Complaint Alias : Count = 100, Avg = 0,12 sec, Min = 0,11 sec, Max = 0,42 sec

Get Content Complaint Category : Count = 100, Avg = 1,07 ms, Min = 0,77 ms, Max = 11,90 ms

Get Content Analysis Sample Type : Count = 100, Avg = 0,22 sec, Min = 0,20 sec, Max = 0,40 sec

Get Content Analysis Sample Type Simple : Count = 100, Avg = 0,29 sec, Min = 0,27 sec, Max = 0,58 sec

Get Content Analysis Sample Type Complex : Count = 100, Avg = 3,93 ms, Min = 3,23 ms, Max = 49,36 ms

Get Content Analysis Group : Count = 100, Avg = 0,16 sec, Min = 0,15 sec, Max = 0,47 sec

Get Content Analysis Category : Count = 100, Avg = 0,24 sec, Min = 0,23 sec, Max = 0,36 sec

Get Content Analysis Parameter : Count = 100, Avg = 30,61 ms, Min = 28,31 ms, Max = 53,40 ms

Get Content Analysis Desc : Count = 100, Avg = 48,51 ms, Min = 45,14 ms, Max = 77,12 ms

Get Content Analysis Unit Measure : Count = 100, Avg = 70,75 ms, Min = 61,49 ms, Max = 0,38 sec

Get Content Analysis : Count = 100, Avg = 37,65 ms, Min = 31,61 ms, Max = 0,34 sec

Get Content Sample Type : Count = 100, Avg = 34,69 ms, Min = 32,16 ms, Max = 65,55 ms

Get Content Sample Recipient : Count = 100, Avg = 93,83 ms, Min = 87,34 ms, Max = 0,12 sec

Get Content Analysis Res Calculator : Count = 100, Avg = 0,25 sec, Min = 0,23 sec, Max = 0,55 sec

Get Content Analysis Specimen Condition : Count = 100, Avg = 34,97 ms, Min = 27,07 ms, Max = 0,33 sec

Get Content Exam : Count = 100, Avg = 0,84 ms, Min = 0,54 ms, Max = 17,97 ms

Get Content Exam Group : Count = 100, Avg = 0,21 sec, Min = 0,19 sec, Max = 0,51 sec

Get Content Exam Category : Count = 100, Avg = 0,24 sec, Min = 0,23 sec, Max = 0,53 sec

Figura 64 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos

Steps

Get Content Body Structure : Count = 100, Avg = 30,94 ms, Min = 25,30 ms, Max = 0,33 sec

Get Content Clinical Service : Count = 100, Avg = 0,15 sec, Min = 0,14 sec, Max = 0,44 sec

Get Content Hemo Type : Count = 100, Avg = 29,04 ms, Min = 26,76 ms, Max = 55,66 ms

Get Content Hemo Type Analysis : Count = 100, Avg = 0,14 sec, Min = 0,14 sec, Max = 0,20 sec

Get Content Discharge Dest : Count = 100, Avg = 30,61 ms, Min = 25,18 ms, Max = 0,34 sec

Get Content Discharge Reason : Count = 100, Avg = 0,29 sec, Min = 0,27 sec, Max = 0,59 sec

Get Content Transport Entity : Count = 100, Avg = 39,52 ms, Min = 31,77 ms, Max = 0,34 sec

Get Content Discharge Reason Dest : Count = 100, Avg = 48,79 ms, Min = 42,42 ms, Max = 83,12 ms

Get Content Diet : Count = 100, Avg = 49,77 ms, Min = 40,90 ms, Max = 0,36 sec

Get Content Questionnaire : Count = 100, Avg = 41,41 ms, Min = 32,96 ms, Max = 0,35 sec

Get Content Response : Count = 100, Avg = 40,07 ms, Min = 33,40 ms, Max = 67,67 ms

Get Content Questionnaire Response : Count = 100, Avg = 0,32 sec, Min = 0,31 sec, Max = 0,62 sec

Get Content Exam Questionnaire : Count = 100, Avg = 62,67 ms, Min = 58,29 ms, Max = 84,60 ms

Get Content Analysis Questionnaire : Count = 100, Avg = 1,39 ms, Min = 0,87 ms, Max = 31,75 ms

Get Content Blood Products Questionnaire : Count = 100, Avg = 59,18 ms, Min = 50,12 ms, Max = 0,36 sec

Get Content Po Param : Count = 100, Avg = 0,17 sec, Min = 0,16 sec, Max = 0,48 sec

Get Content Po Param Sets : Count = 100, Avg = 0,14 sec, Min = 0,13 sec, Max = 0,45 sec

Get Content Po Param Um : Count = 100, Avg = 41,83 ms, Min = 33,79 ms, Max = 0,10 sec

Get Content Po Param Mc : Count = 100, Avg = 0,15 sec, Min = 0,14 sec, Max = 0,44 sec

Figura 65 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 1)

Steps

Get Content Discharge Instructions : Count = 100, Avg = 37,06 ms, Min = 32,37 ms, Max = 56,54 ms

Get Content Discharge Instructions Group : Count = 100, Avg = 29,99 ms, Min = 25,12 ms, Max = 42,74 ms

Get Content Health Plan : Count = 100, Avg = 0,16 sec, Min = 0,15 sec, Max = 0,46 sec

Get Content Health Plan Entity : Count = 100, Avg = 32,37 ms, Min = 24,39 ms, Max = 0,34 sec

Get Content Intervention Plan : Count = 100, Avg = 54,74 ms, Min = 45,47 ms, Max = 0,35 sec

Get Content Task Goal : Count = 100, Avg = 29,87 ms, Min = 24,37 ms, Max = 41,38 ms

Get Content Hidrics Type : Count = 100, Avg = 48,30 ms, Min = 35,76 ms, Max = 0,35 sec

Get Content Hidrics : Count = 100, Avg = 89,97 ms, Min = 76,91 ms, Max = 0,39 sec

Get Content Way : Count = 100, Avg = 34,97 ms, Min = 28,49 ms, Max = 52,45 ms

Get Content Hidrics Charact : Count = 100, Avg = 36,23 ms, Min = 24,43 ms, Max = 0,33 sec

Get Content Hidrics Occurs Type : Count = 100, Avg = 30,46 ms, Min = 24,27 ms, Max = 43,15 ms

Get Content Hidrics Device : Count = 100, Avg = 45,21 ms, Min = 26,26 ms, Max = 0,35 sec

Get Content Hidrics Configurations : Count = 100, Avg = 23,66 ms, Min = 20,47 ms, Max = 34,97 ms

Get Content Hidrics Location : Count = 100, Avg = 2,76 ms, Min = 1,91 ms, Max = 67,46 ms

Get Content Hidrics Way Rel : Count = 100, Avg = 35,72 ms, Min = 30,49 ms, Max = 59,29 ms

Get Content Hidrics Device Rel : Count = 100, Avg = 35,07 ms, Min = 30,02 ms, Max = 51,04 ms

Get Content Hidrics Charact Rel : Count = 100, Avg = 45,33 ms, Min = 29,86 ms, Max = 0,35 sec

Get Content Hidrics Location Rel : Count = 100, Avg = 0,31 sec, Min = 0,30 sec, Max = 0,39 sec

Get Content Supply Type : Count = 100, Avg = 35,06 ms, Min = 27,92 ms, Max = 50,61 ms

Figura 66 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 2)

Steps

Get Content Supply : Count = 100, Avg = 97,32 ms, Min = 86,99 ms, Max = 0,18 sec

Get Content Supply Reason : Count = 100, Avg = 31,41 ms, Min = 26,20 ms, Max = 44,98 ms

Get Content Supply Relation : Count = 100, Avg = 37,96 ms, Min = 28,26 ms, Max = 0,35 sec

Get Content Supply Context : Count = 100, Avg = 0,32 sec, Min = 0,31 sec, Max = 0,40 sec

Get Content Positioning : Count = 100, Avg = 38,61 ms, Min = 26,45 ms, Max = 0,34 sec

Get Content Checklist : Count = 100, Avg = 28,48 ms, Min = 21,15 ms, Max = 0,33 sec

Get Content Checklist Version : Count = 100, Avg = 0,14 sec, Min = 0,13 sec, Max = 0,17 sec

Get Content Checklist Item : Count = 100, Avg = 60,21 ms, Min = 52,02 ms, Max = 91,49 ms

Get Content Checklist Item Dep : Count = 100, Avg = 0,23 sec, Min = 0,22 sec, Max = 0,52 sec

Get Content Checklist Profile Template : Count = 100, Avg = 0,13 sec, Min = 0,13 sec, Max = 0,19 sec

Get Content Habit : Count = 100, Avg = 46,58 ms, Min = 37,25 ms, Max = 0,35 sec

Get Content Habit Characterization : Count = 100, Avg = 34,81 ms, Min = 25,69 ms, Max = 0,33 sec

Get Content Analysis Collection : Count = 100, Avg = 0,23 sec, Min = 0,22 sec, Max = 0,29 sec

Get Content Analysis Collection Int : Count = 100, Avg = 0,14 sec, Min = 0,13 sec, Max = 0,43 sec

Get Content Supply Sup Area : Count = 100, Avg = 0,23 sec, Min = 0,22 sec, Max = 0,25 sec

Get Content Supply Loc Default : Count = 100, Avg = 0,23 sec, Min = 0,22 sec, Max = 0,26 sec

Get Content Hidrics Occurs Type Rel : Count = 100, Avg = 0,32 sec, Min = 0,30 sec, Max = 0,62 sec

Get Content Speciality : Count = 100, Avg = 33,38 ms, Min = 23,96 ms, Max = 0,34 sec

Get Content Necessity : Count = 100, Avg = 47,19 ms, Min = 40,56 ms, Max = 70,23 ms

Figura 67 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 3)

Get Content Occupation : Count = 100, Avg = 32,11 ms, Min = 25,62 ms, Max = 44,67 ms
Get Content Result Notes : Count = 100, Avg = 37,33 ms, Min = 28,82 ms, Max = 0,34 sec
Get Content Origin : Count = 100, Avg = 35,89 ms, Min = 31,17 ms, Max = 52,95 ms
Get Content Rehab Area : Count = 100, Avg = 32,46 ms, Min = 23,80 ms, Max = 0,33 sec
Get Content Rehab Session Type : Count = 100, Avg = 32,13 ms, Min = 23,94 ms, Max = 0,33 sec
Get Content Codification : Count = 100, Avg = 30,77 ms, Min = 26,47 ms, Max = 45,24 ms
Get Content Rehab Area Interv Session Type : Count = 100, Avg = 0,13 sec, Min = 0,12 sec, Max = 0,46 sec
Get Content Rehab Area Interv : Count = 100, Avg = 0,32 sec, Min = 0,30 sec, Max = 0,62 sec
Get Content Intervention : Count = 100, Avg = 0,11 sec, Min = 89,65 ms, Max = 0,40 sec
Get Content Intervention Category : Count = 100, Avg = 30,86 ms, Min = 26,33 ms, Max = 41,71 ms
Get Content Intervention Codification : Count = 100, Avg = 0,32 sec, Min = 0,31 sec, Max = 0,63 sec
Get Content Intervention Questionnaire : Count = 100, Avg = 63,30 ms, Min = 50,05 ms, Max = 0,36 sec

Figura 68 - Resultados dos testes de tempo de resposta das funcionalidades de obtenção dos dados dos conteúdos (continuação – 4)

Anexo E. Inquérito de Satisfação

Neste anexo a Figura 69, a Figura 70 e a Figura 71 representam o inquérito de satisfação submetido à equipa de conteúdos da ALERT, cujo objetivo é aferir o nível de satisfação da equipa, relativamente à solução desenvolvida no âmbito deste projeto. Para a elaboração do questionário, recorreu-se à ferramenta Google Forms, na qual foram desenvolvidas oito questões diretas, de resposta fechada, tendo em consideração a escala de Likert, que admite cinco respostas possíveis, nomeadamente:

- Discordo totalmente;
- Discordo parcialmente;
- Indiferente;
- Concordo parcialmente;
- Concordo totalmente.

Questionário de Satisfação

Este questionário surge no âmbito do projeto de estágio realizado na empresa ALERT Life Sciences Computing (ALERT), no âmbito da unidade curricular do segundo ano, Tese/Dissertação/Estágio (TMDEI) do Mestrado em Engenharia Informática, na área de especialização de Engenharia de Software do Instituto Superior de Engenharia do Porto (ISEP).

O trabalho desenvolvido tem como foco a gestão dos conteúdos utilizados no ALERT® e procura atingir uma solução que permita facilitar a gestão dos mesmos, por parte da equipa de conteúdos, e, também, reduzir o tempo necessário para a sua disponibilização para os clientes.

Neste sentido, o presente questionário destina-se aos membros da equipa de conteúdos e pretende aferir o seu nível de satisfação relativamente à solução do projeto proposto.

O modelo de dados reformulado/atualizado apresenta as alterações identificadas. *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

As funcionalidades de criação de conteúdos foram implementadas corretamente. *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Figura 69 - Inquérito de satisfação

As funcionalidades de atualização de conteúdos foram implementadas corretamente. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

As funcionalidades de visualização de conteúdos foram implementadas corretamente. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

As funcionalidades desenvolvidas permitem a configuração dos conteúdos. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

Figura 70 - Inquérito de satisfação (continuação - 1)

⋮

As funcionalidade desenvolvidas permitem o registo do histórico das alterações dos conteúdos. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

As regras de negócio dos conteúdos foram implementadas corretamente. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

As funcionalidades desenvolvidas a implementar na plataforma Content Server facilitarão e ajudarão a tornar os processos de gestão de conteúdos mais rápido. *

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente

Figura 71 - Inquérito de satisfação (continuação - 2)