

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

CM+ Implementação de um Protótipo de CRM

Sara Inês Monteiro

Tese submetida para o Grau de Mestre
em Engenharia Informática

1 de Julho de 2014

CM+ Implementação de um Protótipo de CRM

Tese submetida para o Grau de Mestre
em Engenharia Informática

Autor:

Sara Inês Monteiro

Orientadores:

Prof^ª. Dr^ª. Maria José Marcelino

Eng. Pedro Douwens Prats

1 de Julho de 2014

Resumo

O CM+ é uma ferramenta que implementa processos de gestão de clientes, desenvolvida no seio da Novabase. Apesar de não ser um produto de CRM puro, tem revelado resultados promissores no que toca a funcionalidades típicas desse tipo de sistemas. A sua eficácia, o baixo custo de implementação, bem como o elevado nível de personalização têm sido factores cruciais para o sucesso do CM+ em projetos de CRM.

É no contexto do CM+ que surge o projeto descrito neste documento, realizado no âmbito do estágio do Mestrado em Engenharia Informática da FCTUC, no ano lectivo 2013/2014. Como objetivo central deste projeto tem-se o desenvolvimento de novas funcionalidades para o CM+ que contribuam de forma inovadora para a sua expansão e para a sua adaptação aos mais diversos sectores de negócio.

Mais concretamente, este trabalho debruça-se sobre a implementação de funcionalidades de gestão unificada de serviços e incidentes multicanal (*call center, loja, SMS, email* e redes sociais) com integração de redes sociais e no desenvolvimento de um módulo de recomendação de soluções de incidentes.

Palavras-Chave: CM+, CRM, Redes Sociais, Serviços, Incidentes, Interações, Recomendação, *Facebook*

Abstract

CM+ is a tool developed in Novabase that implements customer management processes. Despite not being a typical CRM system, it has been revealing promising results regarding the typical functionalities of this kind of systems. Its effectiveness, the low cost implementation as well as the high customization level have been crucial factors to CM+ success in CRM projects.

It is within the context of CM+ that the Master thesis presented in this document arises. The project's main goal is the implementation of new features that can contribute, in an innovative manner, to improve CM+, making it more suitable to multiple market sectors.

Specifically, this project focuses in the implementation of unified management features of multichannel incidents (call center, SMS, email, social networks), its integration with social networks and in the development of an incident solution recommendation system.

Keywords: CM+, CRM, Social Network, Services, Incidents, Interactions, Recommendation, *Facebook*

Agradecimentos

Ao Pedro Santos, pela prontidão com que aceitou a proposta de estágio e por todo o acolhimento em CRM e na Novabase.

Ao Prats, pelos ensinamentos, confiança, acompanhamento e toda a orientação concedida ao longo deste ano.

À professora Maria José Marcelino, pelas críticas, conselhos e ajuda.

À equipa do OM+ e ao João Cordeiro Santos. Pela disponibilidade constante, pela ajuda, conselhos e críticas. Acima de tudo, pela boa disposição, pelos pastéis de nata, chocolates e amabilidade com que me receberam e acolheram.

Ao Bruno, pelos intermináveis debates de ideias e dúvidas, pela ajuda, pela presença (distante) constante, pelas palavras, pela amizade.

Aos meus amigos, tão simplesmente pela amizade.

Aos meus pais e irmãos. Pelo suporte, amor, dedicação e educação.

À Ana João, ao Francisco e à Marta. Pelo apoio inocente e inconsciente disfarçado de risos e sorrisos.

A todos, um sincero e sentido “Obrigada”.

Índice

Glossário	XII
Acrónimos.....	XIII
1 Introdução.....	1
1.1 Entidade Acolhedora.....	1
1.2 Âmbito.....	1
1.3 Objetivos	2
1.4 Planeamento.....	2
1.4.1 Planeamento Total.....	2
1.4.2 Desvios ao Planeamento	3
1.5 Estrutura do Documento.....	4
2 Estado de Arte.....	5
2.1 Contexto	5
2.1.1 Mercado CRM.....	6
2.1.2 Tendências CRM.....	11
2.1.3 <i>Social</i> CRM.....	13
2.2 CM+	13
2.2.1 O que é?.....	13
2.2.2 Caso Prático.....	14
2.2.3 Entidades.....	16
2.2.4 Arquitetura	17
2.2.5 CM+ como CRM.....	20
3 Metodologia e Objetivos.....	23
3.1 Metodologia	23
3.2 Definição de Objetivos	24
4 Análise de Requisitos.....	27
4.1 Levantamento de Requisitos	27
4.1.1 Prototipagem de Cenários	28
4.2 Análise de Requisitos.....	34
4.2.1 Requisitos Funcionais	34
4.2.2 Requisitos Não Funcionais	38
5 Arquitetura e Especificação.....	41

5.1	Estrutura Geral do Sistema	41
5.1	Base e Modelo de Dados	42
5.2	Lógica.....	44
5.2.1	CM+.....	44
5.2.2	Módulo de Integração com o Facebook.....	45
5.2.3	Módulo de Recomendação de Soluções	46
5.3	Portal.....	48
5.4	Estrutura e Tecnologias	48
6	Implementação	51
6.1	Base e Modelo de Dados	51
6.2	CM+	53
6.2.1	Gestão de Incidentes.....	53
6.2.2	Integração com o Módulo Facebook	55
6.3	Módulo de Integração com o Facebook	57
6.4	Módulo de Recomendação de Incidentes	61
6.4.1	Pré-processamento	61
6.4.2	Motor de Pesquisa	62
6.5	Portal.....	63
7	Verificação e Validação	65
7.1	Testes Funcionais.....	65
7.1.1	Testes Unitários e de Integração	65
7.1.2	Testes de Sistema	66
7.2	Testes Não Funcionais.....	66
7.2.1	Testes de Qualidade (Sistema de Recomendação)	66
7.2.2	Testes de Performance.....	69
8	Conclusão	73
8.1	Trabalho Futuro	74
9	Referências	75
	Anexos.....	77
	Anexo A – Planeamento PERT.....	i
	Anexo B – Prototipagem Total do Sistema	iii
	Campanhas de Marketing.....	iii
	Detalhe de Campanha de Marketing.....	iii
	Dashboards de Marketing.....	iv

Detalhe de Oportnidades de Marketing	iv
Encomendas	v
Detalhe de Produto	v
Detalhe de Proposta.....	vi
Anexo C – Descrição Detalhada do Modelo de Dados do CM+	vii
Anexo D –Portal de Atendimento e OM+	xiii
Anexo E – Testes de Sistema.....	xv
Anexo F – Testes de <i>Performance</i> (Sistema de Recomendação)	xx
Anexo G – Testes de Qualidade (Sistema de Recomendação).....	xxii
Anexo H – Testes de <i>Performance</i> : Módulo de Integração com o <i>Facebook</i>	xxiii
Manutenção da Informação Atualizada.....	xxiii
<i>Performance</i> do Módulo.....	xxiii

Lista de Figuras

Figura 1.1 – Plano do Ano Letivo.....	3
Figura 2.1 – Áreas Estratégicas de CRM.....	5
Figura 2.2 - Mercado de CRM em 2012.....	6
Figura 2.3- Arquitetura do Sistema de Informação	15
Figura 2.4 - Arquitetura do CM+	17
Figura 2.5– Execução de uma Ordem.....	19
Figura 2.6 - Arquitetura do CM+ como CRM.....	20
Figura 2.7 – Arquitetura do CM+ como CRM	22
Figura 3.1– Fases do Processo de Implementação.....	23
Figura 4.1– Protótipo para a Página Inicial.....	28
Figura 4.2– Protótipo para a página de Incidentes	29
Figura 4.3– Protótipo para a página de Detalhes de Incidentes	30
Figura 4.4– Protótipo para a página de Resposta a Interações	31
Figura 4.5– Protótipo para a página de Criação de novas interações.....	32
Figura 4.6– Protótipo para a página de criação de novos incidentes	32
Figura 4.7– Protótipo para a página de detalhes de incidentes	33
Figura 4.8– Protótipo para a página de pesquisa de incidentes	33
Figura 4.9– Protótipo para a página de alertas	34
Figura 5.1– Arquitetura proposta para o sistema final.....	42
Figura 5.2– Representação do Modelo de Entidades	43
Figura 5.3– Arquitetura do Módulo de Integração com o Facebook	45
Figura 5.4– Arquitetura do Módulo Recomendação	47
Figura 6.1– Fases do Processo de Implementação.....	51
Figura 6.2– Screenshot da estrutura do modelo ADF	52
Figura 6.3– Screenshot do Detalhe da Entidade CmpIncident	52
Figura 6.4– Screenshot do Detalhe da Vista CmpIncident	53
Figura 6.5– Diagrama de Atividades para a “Criação de Novo Incidente Manual”	54
Figura 6.6– Exemplo de publicação no Facebook.....	57
Figura 6.7– Diagrama de Atividade para a “Criação de Interação Via Facebook”	58

Figura 6.8– Diagrama de Atividade para a “Resposta a Interação Via Facebook”	59
Figura 6.9– Diagrama de Atividades para a “Recomendação de Soluções”	61
Figura 6.10– Lógica do Sistema de Recomendação.....	61
Figura 6.11– Área de Gestão de Incidentes do Portal.....	63
Figura 6.12– Detalhe de Incidente	64
Figura 7.1 – Valor da métrica <i>Recall</i> para os casos de teste realizados	68
Figura 7.2– Valor da métrica Precisão para os casos de teste realizados.....	68
Figura 7.3– Variação do número de Transações por Segundo em função do número de pedidos e do número de clientes	70
Figura 7.4– Tempo de execução dos Pedidos ao módulo de integração com o Facebook (Em função do número de pedidos)	72

Lista de Tabelas

Tabela 2.1 - Características do CM+.....	22
Tabela 4.1 - Requisitos Funcionais.....	35
Tabela 4.2 - Requisitos Não Funcionais.....	38
Tabela 7.1– Tabela para o registo dos testes de sistema	66

Glossário

Activis – Software de gestão de processos de negócio distribuído pela *WeDo Technologies* que apresenta funcionalidades de administração de processos específicos da indústria das telecomunicações (tais como o controlo da portabilidade numérica, a gestão de compartilhamento de acessos locais, a gestão de processos de rede dentro do operador através de workflow, o provisionamento de números de um operador, a automação da integração de clientes parceiros. Ao mesmo tempo que otimiza o fluxo de negócio, o Activis apresenta a vantagem de ser fácil de integrar com outros sistemas.

ADF Oracle – Framework que possibilita um acesso unificado aos sistemas de backend através duma representação completa de um determinado modelo de dados (desde entidades às associações entre elas e à LOV). É de livre licenciamento e suportada pelo Oracle JDeveloper 11g.

EXTjs – Framework livre de licenciamento, disponibilizada sob a licença LGPL e compatível com os browsers mais utilizados no mercado (IE, *Firefox*, *Safari*, *Chrome*, etc). Esta framework de frontend fornece uma estrutura que facilita a integração dos componentes visuais com as regras de negócio. Os arquivos necessários à sua utilização podem ser incluídos na página HTML (compatível com HTML5) através da tag <SCRIPT> como qualquer *js*. A EXTjs apresenta uma arquitetura MVC, ou seja, contém uma componente *Model* que corresponde a uma coleção de campos e respectivos dados, uma componente *View* relativa a qualquer tipo de componente visual como grelhas, árvores, painéis, etc. e por fim a componente *Controller* onde se encontra o código responsável pelo funcionamento da aplicação, ou seja, onde está contida a lógica

Tibco – Software de EAI proprietário que permite fazer a integração entre sistemas e bases de dados distribuídos e incompatíveis, garantindo o mapeamento correto de mensagens e pedidos trocados entre os vários componentes. Este ano o TIBCO foi posicionado como líder no quadrante mágico da Gartner para plataformas de integração *On-Premise*.

Tribold EPM – Solução de software proprietária para EPM (Enterprise Product Management). Centraliza o seu funcionamento nos catálogos de produtos e serviços permitindo ao utilizador acompanhar e realizar os processos para a gestão do ciclo de vida do produto. A camada de integração do Tribold possibilita uma infraestrutura completa para a integração dos dados dos produtos de forma a estarem disponíveis para outros sistemas (como CRM, *Billing*, ERP) de forma eficiente e estruturada

WebLogic Server – Servidor *web* da Oracle que permite a centralização dos serviços da aplicação, como por exemplo as funcionalidades de servidor web, componentes de negócio e o acesso aos restantes serviços (CRM, ERP, etc.). Para além de promover um aumento da performance da aplicação, garante que o acesso aos serviços críticos da empresa é feito com um elevado nível de segurança.

O *WebLogic Server* é uma plataforma que consiste num *application server Java EE*, um portal empresarial, uma plataforma de EAI, um servidor de transações, uma plataforma de telecomunicações e um servidor *web http*. Neste sistema está implementada a versão 12c do *WebLogic Server* que é completamente compatível com HTML5.

Acrónimos

BSCS – *Business Support Control System*
CAGR – *Compound Annual Growth Rate*
CM+ - *Customer Management Plus*
CRM – *Customer Relationship Management*
DML – *Data Manipulation Language*
EPM – *Enterprise Product Management*
ERP – *Enterprise Resource Planning*
IPQ – Instituto Português da Qualidade
OM+ - *Order Management Plus*
PA – Portal de Atendimento
TCO – *Total Cost of Ownership*
UFE – *Unified Frontend*

Capítulo 1

Introdução

No âmbito da obtenção do grau de Mestre em Engenharia Informática pela Faculdade de Ciências e Tecnologias da Universidade de Coimbra, tem lugar um estágio na empresa Novabase, mais especificamente na competência de CRM com a duração de um ano lectivo.

O presente relatório visa a documentação do trabalho realizado, desde o momento de contextualização do tema até à fase de verificação e validação do sistema, passando por processos de análise de requisitos, definição de proposta, especificação da arquitetura e processos de implementação.

Neste primeiro capítulo pretende-se que o leitor conheça a entidade acolhedora e se familiarize com o âmbito do projeto e os objetivos gerais que o desencadearam.

1.1 Entidade Acolhedora

A Novabase é uma empresa de desenvolvimento de soluções de *software* à medida, fundada no ano de 1989. Foi a 1ª empresa portuguesa a obter certificação pelo IPQ, segundo a norma NP EN ISO 9001 e atualmente é líder em Tecnologias de Informação, estando cotada na *Euronext* de Lisboa desde o ano 2000.

Para além do desenvolvimento de *software*, a Novabase trabalha também nas áreas de integração de sistemas, TV Digital, bilhética para transportes e na oferta de infraestruturas avançadas.

No ano de 2012 o volume de negócios registado foi 212 milhões de euros, sendo que desses, 29% foram obtidos no exterior. Atualmente a empresa exerce funções em 40 países e conta com mais de 2000 colaboradores.

Com a visão de “Tornar a vida das pessoas e das empresas mais simples e mais feliz”, a Novabase tem investido na área de Investigação e Desenvolvimento para disponibilizar produtos e serviços ao nível de Telecomunicações, Serviços Financeiros, Governo, Serviços de Saúde, Energia, *Managed Services*, *Venture Capital* e outros (1).

Na área de *Business Solutions* a empresa apresenta várias competências, entre as quais CRM (*Customer Relationship Management*). Desta fazem parte mais de 100 colaboradores empenhados no desenvolvimento e implementação de soluções de negócio para a gestão da relação com o cliente ao nível empresarial (1). É nesta área que é acolhido este estágio da FCTUC, tendo ficado à responsabilidade do Engº Pedro Douwens Prats.

1.2 Âmbito

No âmbito do desenvolvimento de um Sistema de Informação para uma operadora de telecomunicações africana, uma equipa da Novabase desenvolveu um componente de *software*, denominado CM+ que desempenha funcionalidades de gestão de clientes.

Esse componente revelou-se verdadeiramente eficaz na execução dos processos à sua responsabilidade e, conseqüentemente, hoje em dia é um trunfo a utilizar noutros projetos. Como tal, a empresa tem todo o interesse em fazer crescer este sistema, dotando-o de novas funcionalidades coerentes com o que são as tendências de CRM e capazes de o tornar mais competitivo a nível de mercado.

Neste contexto, surgiu então a proposta de estudar o produto no seu estado atual e expandi-lo de modo a torná-lo mais completo e mais próximo daquilo que deve ser um sistema de CRM.

1.3 Objetivos

O CM+ foi desenvolvido para o sector específico das telecomunicações e responde às necessidades a ele inerentes. No entanto, a sua aplicabilidade em outros sectores, como saúde ou banca, é diminuta, sendo necessárias inúmeras alterações para que possa ser usado nesses sectores.

Por outro lado, apesar de desempenhar funcionalidades típicas de um sistema de CRM, quando comparado com produtos de CRM conceituados no mercado, é possível ver que o CM+ tem em falta muitas funcionalidades ao nível da gestão de *Marketing*, Vendas e Serviços.

Assim, uma especificação fundamental deste projeto é o desenvolvimento de novas funcionalidades no CM+ que o aproximem daquilo que deve ser um verdadeiro produto de CRM e que ao mesmo tempo contribuam para o tornar adaptável a vários sectores de mercado. Espera-se que essas novas funcionalidades reflitam as tendências que se verificam ao nível da área de CRM, conferindo ao CM+ características de inovação e competitividade.

De acordo com esses critérios, e considerando as tendências verificadas nas áreas de Serviços, Integração com Redes Sociais e Adaptação ao Contexto, define-se como objetivos centrais:

- Implementação de **mecanismos de gestão de serviços**: expandir o CM+ para possibilitar processos de gestão de incidentes multicanal de forma unificada (*call center*, loja, *email*, SMS, redes sociais);
- **Integração com redes sociais**: integrar o CM+ com redes sociais e, em específico, fazer a implementação ao nível da gestão de serviços, possibilitando a denúncia de dúvidas e problemas, seu processamento e resposta por esta via;
- Desenvolvimento de um **Sistema de Recomendação de Soluções** de Incidentes.

Ao longo dos capítulos 3 e 4 estes objetivos serão detalhados, subdivididos e convertidos em requisitos.

1.4 Planeamento

1.4.1 Planeamento Total

Na imagem seguinte encontra-se o Diagrama de Gantt que tem por objetivo expor o plano de trabalho geral estipulado para o desenvolvimento do projeto.

Podé subdividir-se a sua análise em 2 períodos principais, o 1º corresponde ao 1º semestre lectivo (Setembro – Janeiro), para o qual se previu o estudo do contexto do projeto, a ambientação aos conceitos *core* de CRM, a análise do sistema a trabalhar, o estudo do diferencial do CM+ atual para o que se pretende e, em função disso, a definição e análise de requisitos e desenho da arquitetura.

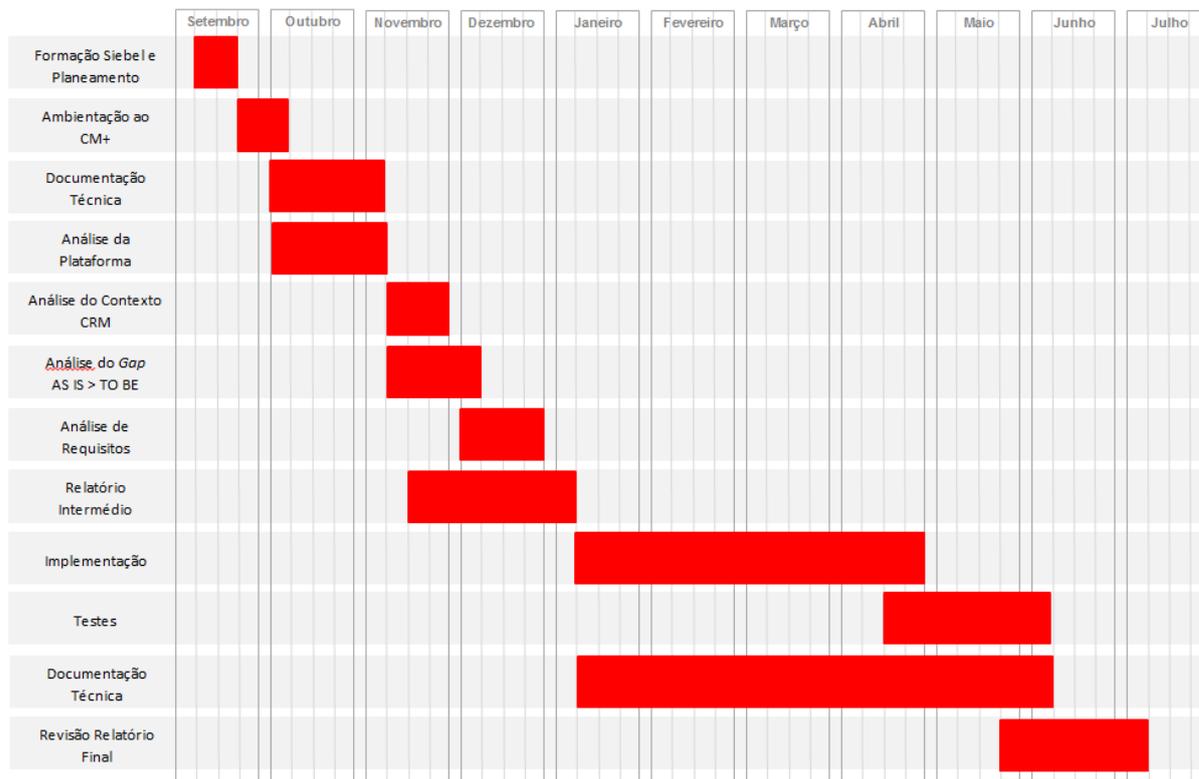


Figura 1.1 – Plano do Ano Letivo

Ao 2º semestre correspondem os processos de implementação das funcionalidades, a sua verificação e validação. Todo o trabalho desenvolvido deve ser acompanhado da documentação dos processos desenvolvidos para que no dia 30 de Junho de 2014 possa ser entregue o relatório de estágio final.

No anexo 1 podem encontrar-se um planeamento mais detalhado correspondente ao período de implementação do sistema.

1.4.2 Desvios ao Planeamento

O trabalho desenvolvido ao longo do 1º semestre sofreu algum adiantamento relativamente ao inicialmente planeado visto que a fase de Documentação Técnica foi ligeiramente menos demorada do que se esperava.

Como tal, a fase de implementação iniciou-se no mês de dezembro. Os processos implementados corresponderam fundamentalmente às alterações ao nível da base e modelo de dados do sistema como será analisado ao longo do documento.

Para complementar e detalhar o período de implementação pode observar-se, como já referido, no anexo 1 um plano mais detalhado, de acordo com a técnica de planeamento PERT, respeitante ao processo de implementação e às várias tarefas a ele inerentes. Através desse plano é possível verificar que, apesar de a previsão otimista estar de acordo com o especificado no diagrama de Gantt apresentado, o cálculo final da previsão aponta para um processo de implementação um pouco mais longo.

De acordo com o previsto pelo planeamento de PERT, o período de implementação prolongou-se mais que o desejado e, como tal, a fase de testes teve de ser encurtada. Como

o plano alertou à partida para esse fato, as tarefas de validação foram estruturadas para que o trabalho pudesse ser entregue atempadamente.

1.5 Estrutura do Documento

O presente relatório estrutura-se em 7 capítulos principais.

O primeiro corresponde ao estudo do **estado de arte** do contexto do projeto. Nesse capítulo é analisado não só o *software* alvo de estudo e alteração mas também o contexto mais global em que se integra, a área de CRM. Desta forma pretende-se que o leitor tenha um conhecimento mais abrangente daquilo que é CRM, das condições atuais desse mercado e de como o CM+ se integra nesse mesmo contexto. É também neste capítulo que se analisa o diferencial entre o CM+ e alguns dos produtos de CRM mais comuns.

No segundo capítulo descreve-se o processo de **análise de requisitos**, não só a fase de levantamento dos mesmos mas também o processo de seleção daqueles que devem ser implementados e respetiva descrição.

O terceiro capítulo apresenta a **metodologia** seguida para a implementação do trabalho desenvolvido.

No quarto capítulo descreve-se a proposta de projeto, apresentando a esquematização da **arquitetura** do sistema face aos requisitos definidos anteriormente. Esta fase é já mais inerente ao processo de implementação servindo-lhe quase como introdução.

Ao nível do quinto capítulo detalham-se os processos de **implementação** e desenvolvimento do projeto proposto. Com ele, espera-se que o leitor compreenda as fases envolvidas no trabalho, as decisões tomadas e o porquê.

O capítulo seguinte corresponde à descrição dos processos de **verificação e validação**. Neste são descritos os testes realizados que verificam se o sistema está ou não de acordo com os requisitos funcionais e não funcionais para ele definidos.

Por fim, na **conclusão** resume-se o trabalho desenvolvido, verificando se cumpre os objetivos propostos inicialmente e perspetivando o seu futuro e possíveis extensões.

Capítulo 2

Estado de Arte

Expandir o CM+ de forma a aproximá-lo daquilo que é um *software* de CRM puro envolve não só conhecer em profundidade o sistema em si, mas também conhecer e compreender o que é CRM e quais as características de um *software* deste tipo. Só desta forma se torna possível definir e implementar funcionalidades que se adequem aos objetivos estipulados.

Assim, este capítulo reflete o estudo realizado acerca do contexto do projeto. Inicialmente analisa a área e o mercado de CRM e descreve sucintamente alguns dos sistemas mais conceituados. Posteriormente dá a conhecer o CM+ e enquadra-o na área de CRM.

2.1 Contexto

CRM pode ser definido como tecnologia de negócio cuja aplicação conduz à otimização do lucro, das receitas e do grau de satisfação dos clientes. Pressupondo um ambiente centrado no cliente, um CRM procura a organização de todos os processos que envolvam clientes ao nível dos vários sectores empresariais (2).

De forma mais simples, podemos definir CRM como um sistema empresarial que possibilita a gestão do cliente e das relações com ele estabelecidas, ao longo do tempo. Permitindo o acompanhamento dos processos desde o pré-venda, **marketing**, passando pela **venda** até ao pós-venda, **serviços**, possibilita uma visão 360º daquilo que são o cliente e os processos em que está envolvido (figura 2.1).



Figura 2.1 – Áreas Estratégicas de CRM

Implementar um CRM possibilita não só uma janela de interação com o cliente de maior amplitude, como procura funcionar como um nível de integração entre o cliente e todos os canais empresariais. Isto conduz ao aumento da satisfação do cliente e

consequentemente potencia os lucros, tendo este factor vindo a contribuir fortemente para a generalização da sua utilização (2).

Mais que um sistema empresarial, um CRM deve ser encarado como uma estratégia definida e desenvolvida através de processos e tecnologias que maximizam o valor extraído do cliente ao longo do período de vida do seu relacionamento com uma dada empresa.

2.1.1 Mercado CRM

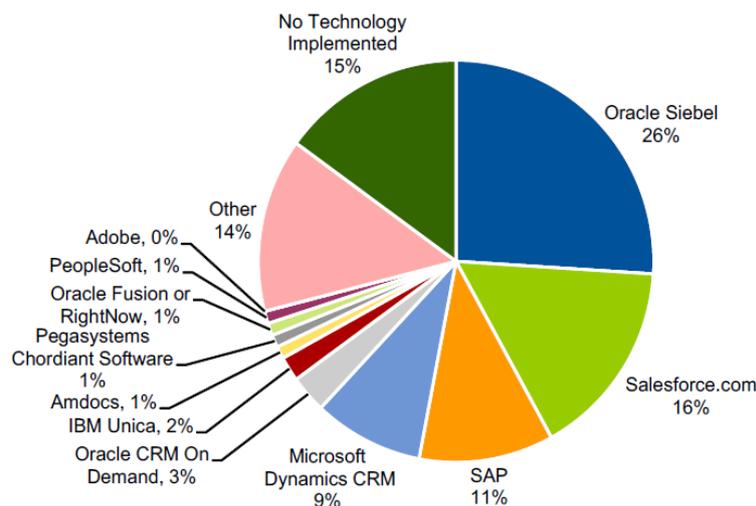
Ao nível do mercado de produtos de *software* empresariais, o sector específico de CRM tem revelado um crescimento acentuado e apresentado uma quota cada vez mais significativa. Só no ano de 2012 as receitas da venda desse sector cresceram cerca de 12.5%, tendo-se revelado um mercado de 20.6B de dólares (3).

O mercado de CRM foi também líder no que toca às projeções de crescimento apresentando um CAGR de 15,1% entre 2012 e 2017, prevendo-se que nesse ano supere o mercado de ERP, atualmente líder deste sector (3).

Patrick J. Sullivan e Ed Thompson justificam estes dados com uma maior consciencialização das empresas da importância de investir em tecnologias que aumentem o volume de novas vendas ao mesmo tempo que garantem a satisfação e manutenção dos clientes atuais (2).

Considerando um contexto económico cada vez mais exigente e um nível de concorrência elevado, tornam-se compreensíveis os dados apresentados. Atualmente, as empresas procuram estar onde o cliente está, compreender as suas necessidades e ir ao encontro delas, promovendo serviços exímios que maximizem a satisfação do cliente. Tudo isto pode ser conseguido através da implementação dos processos pressupostos num sistema de CRM.

Entre os mais de 1000 produtos de CRM, identificados pela *Gartner* no ano de 2012, encontram-se soluções proprietárias e *opensource*, líderes de mercado e *softwares* com presença residual. Na figura 2.2 pode observar-se a divisão de mercado de CRM pelos diferentes fornecedores de *software* (4).



Source: Gartner (October 2013)

Figura 2.2 - Mercado de CRM em 2012

Considera-se importante, neste momento, fazer uma visita geral por alguns dos produtos de CRM referenciados para que se compreenda melhor o que eles podem oferecer e também aquilo que os diferencia. O grande intuito desta análise é a aquisição de um conhecimento geral, mas sólido, relativamente ao conceito de *software* de CRM, de forma a criar um ambiente propício à compreensão do CM+ e de como ele se localiza face às soluções de CRM atualmente disponíveis.

Para esta análise optou-se por selecionar 3 dos produtos mais vendidos: *Siebel*, *Dynamics CRM* e *Salesforce*. De forma a alargar o espectro da análise e possibilitar uma visão mais abrangente, decidiu juntar-se à análise 2 produtos opensource, o *SugarCRM* e o *OpenCRX*.

Com estas escolhas pretende abranger-se uma parte representativa daquilo que são as soluções de CRM disponíveis, para que se conheçam quais as características gerais valorizadas e exigidas a um *software* deste tipo, bem como quais os factores que contribuem para o sucesso ou insucesso destas ferramentas.

Siebel

A primeira versão do *software Siebel* surgiu pelas mãos da empresa *Siebel Systems* fundada em 1993, apresentando nessa altura funcionalidades de automação de vendas. Mais tarde, no decorrer do ano de 2005, a *Oracle* comprou a empresa sendo que, atualmente, o *Siebel* é propriedade da *Oracle* e por ela distribuído (5).

Hoje em dia o *Siebel* é líder de mercado de CRM, apresentando funcionalidades transversais aos vários sectores económicos, desde saúde, banca, até ao sector público e disponibilizando processos de gestão de marketing, vendas e serviços.

A dimensão deste *software* possibilitou à *Oracle* dominar o mercado de CRM durante muito tempo. No entanto, nos últimos anos, tem registado perdas significativas que revelam as mudanças que se estão a desenrolar neste sector.

Se no ano de 2008 as previsões da *Gartner* apontavam para uma quota de mercado de 36%, esse valor foi diminuindo, alcançando o valor de 26% no ano de 2012 e uma previsão de 19% no presente ano. O *Siebel* apresenta então uma taxa de utilização cada vez menor, perdendo terreno para concorrentes como *Salesforce*, *SAP* e *Dynamics CRM*. Inclusivamente, a *Gartner* prevê que o *Salesforce* ganhe a liderança ainda este ano (4).

Os custos de implementação e manutenção elevados, a dificuldade de implementação e utilização bem como a desatualização do software são alguns dos factores que podem estar na origem desse decréscimo de utilização. Outro fato que pode também ter contribuído para esta situação é o aparecimento de inúmeros produtos de CRM e a melhoria de alguns já existentes, como o caso do *Dynamics CRM*.

Para este *software* destacam-se como pontos fortes o facto de ser o mais utilizado a nível mundial e por grandes empresas e o fato de ser o mais antigo o que contribuí para que seja uma ferramenta muito completa, muito testada e com um nível de suporte extremamente elevado.

A dificuldade e o custo de implementação são as maiores fraquezas do *Siebel* e que têm contribuído fortemente para o seu declínio.

Salesforce CRM

Foi no ano de 1999 que foi lançada a 1ª versão deste *software* de CRM, sendo que no ano de 2000 foi apresentado como a 1ª solução online CRM para automação de vendas. No mercado em estudo, este produto reveste-se de especial importância visto ser uma das primeiras plataformas *cloud* para CRM (6).

Com a evolução do sistema foram acrescentadas funcionalidades de gestão de marketing e serviços e a utilização deste produto foi-se generalizando.

Atualmente, o *Salesforce.com* conta com mais de 100.000 clientes entre os quais se encontram nomes sonantes como *Facebook*, *Bayer*, *Virgin*, *Philips* (7).

A *Gartner* calculou uma presença de utilização de 19% no ano de 2012. Face aos 26% do atual líder, é possível observar que este produto adquiriu uma posição muito relevante no mercado. Mais, o mesmo estudo indica que em 2013 esse número deve ascender aos 24% o que, paralelamente ao declínio do *Siebel*, pode levar este produto a alcançar a liderança do mercado de *software CRM* (4).

Microsoft Dynamics CRM

Em paralelo com a queda acentuada da *Oracle*, um dos produtos que tem feito o percurso inverso é o distribuído pela *Microsoft*, o *Dynamics CRM*. Em 2005 este apresentava uma taxa de participação em projetos CRM residual, em 2009 este valor era já de 7% e em 2012 tinha subido para 9%. Atualmente, estima-se uma utilização de 11% a nível do mercado mundial de CRM. No seu estudo “*CRM Applications Deployed by Consultancies in 2013 Show Which Skills Are Prevalent*” a *Gartner* prevê que este *software* continue num ritmo de crescimento acentuado (4).

A primeira versão deste produto foi lançada no ano de 2003, 10 anos depois do *Siebel*, e apresentava apenas funcionalidades mínimas de gestão de vendas e serviços.

Ao longo dos anos este produto foi sendo atualizado e melhorado, tornando-se mais completo e flexível versão após versão. A última versão foi lançada em 2013 e revela um produto altamente flexível e personalizável, transversal aos vários sectores de negócio, com um interface *user friendly* e completamente compatível com dispositivos móveis (8).

Apesar de ser mais recente que o *Siebel* e só após 2008 ter solidificado a sua presença no mercado, o *Dynamics CRM* é um produto de grande valor e potencial, tendo sido considerado como líder no *magic quadrant* de 2012 da *Gartner* para a automação de vendas, assim como o *Siebel* (9) (10).

Dado o seu potencial considera-se como um exemplo de boas práticas e cuja evolução deve ser acompanhada atentamente.

Como ponto forte deste produto salienta-se a sua usabilidade de forma transversal a vários sectores económicos. Para além disso salienta-se o facto de um projeto de CRM com este produto requerer ao nível de recursos humanos cerca de 4 consultores de negócio e 9 consultores técnicos face aos 6 consultores de negócio e 22 técnicos requeridos para a implementação do *Siebel*, por exemplo (4).

Como ponto fraco nota-se o fato de não estar preparado para projetos de dimensão tão grande como os de *Siebel*, ou seja, é aplicável para projeto se pequena/média dimensão.

SugarCRM

Os produtos analisados até ao momento têm um ponto em comum, todos eles são produtos proprietários. No entanto, no mercado de CRM, é possível identificar vários distribuidores de produtos *opensource*. É importante analisar também este tipo de produtos visto serem alvo de grande interesse por parte dos compradores pelo pressuposto de serem mais baratos e possibilitaram maior nível de personalização.

O *SugarCRM* é um desses produtos *opensource*, sendo um dos mais conceituados e vendidos. Este produto surgiu no decorrer do ano 2004 e está disponível quer na *cloud* quer *on-site* (11).

Hoje em dia este produto conta com um grande leque de clientes e constitui-se como uma opção ao nível dos produtos proprietários, adequando-se a empresas de qualquer

dimensão e sector. Por entre os clientes do *SugarCRM* destacam-se a *IBM*, *uShip*, e, um caso português, a *Estradas de Portugal* (12).

O *SugarCRM* está disponível em duas versões, uma livre de licenciamento com funcionalidades limitadas e outra com licenciamento por número de utilizadores e com a totalidade das funcionalidades disponível.

Visto ser possível aceder e alterar o código fonte do sistema, consegue obter-se um nível de personalização muito superior aos produtos já analisados. Esse fator, em conjunto com a ideia frequente de que os produtos *opensource* apresentam custos mais baixos, leva muitas empresas a questionarem a sua utilização.

No entanto, essas questões devem ser analisadas cuidadosamente. Aceder e alterar o código, por exemplo, é um processo complexo e que envolve muito conhecimento técnico. Por outro lado, em relação ao custo mais baixo, nem sempre isso se verifica visto que, num *software* empresarial, os custos não são limitados ao licenciamento. Assim, não é imediato dizer-se que esta opção será à partida mais efetiva que as anteriores, dependendo isso de caso para caso e das características particulares de cada negócio.

OpenCRX

O desenvolvimento do *OpenCRX* iniciou-se no decorrer do ano de 2003, sendo que o lançamento da primeira versão do produto ocorreu cerca de um ano depois (13).

O seu surgimento deveu-se ao facto dos fundadores terem detectado certas falhas no mercado de CRM, tais como:

- O gasto excessivo em soluções proprietárias que muitas vezes não se adequam inteiramente às regras de negócio específicas de cada empresa;
- A dependência de plataformas específicas, muitas vezes incompatíveis com outros sistemas da empresa;
- A dificuldade de personalização dos sistemas existentes.

Este produto tem vindo a ser aperfeiçoado a cada nova versão e no último lançamento, em Novembro de 2013, apresenta funcionalidades transversais aos 3 componentes fundamentais de CRM, de forma adaptável a vários sectores de negócio (13).

Comparativamente à solução anterior (*SugarCRM*), o *OpenCRX* falha essencialmente no que toca a questões de referências e suporte. No entanto, o facto de apresentar uma arquitetura técnica modular e independente de plataformas e não ter custos de licenciamento associados, são critérios de escolha muito relevantes e com um peso acentuado.

Análise Comparativa

Conhecidas, de forma geral, algumas soluções de CRM disponíveis no mercado, é importante compreender as características técnicas que lhe estão associadas e de que forma elas se diferenciam entre si.

Os produtos descritos apresentam um conjunto de funcionalidades comuns. Apesar de poder variar o nível de detalhe ou complexidade de sistema para sistema, o objetivo é comum. Assim, em termos de funcionalidades todos os sistemas apresentam:

- Gestão de Contactos;
- Gestão de Contas;

- Gestão de Tarefas;
- Gestão de Marketing (oportunidades e oportunidades potenciais);
- Gestão de Vendas;
- Processos de serviço ao cliente;
- Gestão de Incidentes;
- Integração com serviços de *email*, SMS e chamadas;
- Adaptação com dispositivos móveis;
- Integração com as redes sociais.

Pode dizer-se que os produtos em análise apresentam grandes semelhanças no que toca a funcionalidades. Independentemente destas poderem ser mais desenvolvidas nuns sistemas do que noutros, considera-se que, qualquer um deles, se compõe como uma solução de CRM completa que pode satisfazer as necessidades de uma empresa de qualquer sector.

Percebe-se então que as diferenças mais relevantes no que respeita aos vários produtos não residem unicamente nas funcionalidades técnicas mas em outros factores. Nesse conjunto encontramos factores como: o custo total de implementação e manutenção (TCO), o nível de suporte disponibilizado, a capacidade de adaptação e personalização do sistema, a credibilidade e estabilidade que lhe está associada.

Custo Total de Posse (TCO)

Num sistema empresarial, os custos não se devem apenas à aquisição do *hardware* e *software* a ele associados. De facto, muitas vezes, esse pode até ser um dos menores custos.

O custo total associado à aquisição e implementação de um sistema como ERP ou CRM, denominado de *Total Cost of Ownership*, envolve para além dos custos de aquisição, os gastos na gestão e suporte, na manutenção, na formação dos utilizadores e ainda nas despesas para o utilizador final (14).

Desta forma, se por um lado os sistemas *opensource* podem à partida ser mais atrativos pelo facto de disponibilizarem funcionalidades completas e capacidade de personalização com baixo custo (ou até mesmo nulo) de aquisição, por outro, há que ter em conta os outros gastos associados e fazer uma ponderação que permita perceber se a longo prazo os custos totais são realmente mais baixos.

Neste momento o objetivo não é o de apresentar custos detalhados de cada *software*. Mas, por forma a compreender melhor esta questão, pode apresentar-se o caso do *Dynamics CRM* e do *SugarCRM*.

Para um cenário com 500 utilizadores, se considerarmos as versões mais simples e baratas de cada um, verifica-se que o custo a 3 anos do *SugarCRM* é inferior a \$500.000 e o do *Dynamics CRM* pode ir até cerca de \$900.000 (15).

No caso apresentado, o custo do *SugarCRM* é, de facto, mais baixo. No entanto, se forem analisados os sistemas com a totalidade das funcionalidades disponíveis, a situação inverte-se, apresentando o *SugarCRM* um custo máximo de cerca de \$1.900.000 em contraste com os \$900.000 do *Dynamics CRM* (15).

Se num primeiro impacto o *SugarCRM* aparentava ter um custo muito menor, quando se opta pela solução total, o custo aumenta substancialmente, mostrando que a longo prazo o produto é significativamente mais caro que o *Dynamics CRM*.

Dado o investimento significativo que é para uma empresa apostar numa solução CRM, o TCO é então um factor crucial na escolha e que deve ser analisado cuidadosamente.

Nível de Personalização

Em função do sector e das regras de negócio próprias de cada empresa, os requisitos relativos ao sistema de CRM variam muito e como tal a capacidade de personalização de um sistema é um factor a ter em conta.

Neste ponto é fácil compreender que um *software opensource* se destaque. Ao disponibilizar o código fonte, o responsável pela implementação do sistema pode fazer alterações múltiplas de forma a adequar ao máximo os métodos e processos às características específicas do negócio.

No caso de sistemas proprietários, como o *Siebel* e o *Dynamics CRM*, a personalização do sistema é mais complexa e trabalhosa visto que não é possível aceder ao código fonte e, como tal, para criar funcionalidades à medida, é necessário adaptar as existentes de raiz em cada um dos sistemas.

Credibilidade e Estabilidade

Implementar um sistema de CRM é um processo com um elevado custo associado. Como tal, quando uma empresa decide fazê-lo tem de garantir que aposta em produtos credíveis e estáveis que lhe garantam o retorno do investimento.

Se produtos como o *Siebel*, *Dynamics CRM*, *Salesforce* e até mesmo o *SugarCRM* têm uma utilização generalizada, estão muito testados e como tal são muito confiáveis para o mercado, soluções mais recentes e menos generalizadas como é o caso do *OpenCRX* têm muitas vezes dificuldade em penetrar e expandir-se visto não terem esse renome.

Suporte

Durante a implementação de um sistema de CRM são frequentes as dúvidas e os problemas. O suporte tem um papel fundamental para as resolver e acima de tudo para o fazer em tempo útil sem prejudicar os tempos do projeto.

Dada a sua dimensão e robustez, um *software* proprietário, como os que aqui foram referidos, apresentam um suporte de elevado nível de qualidade, que garante não apenas resposta rápida mas também efetiva. Apesar do *OpenCRX* também disponibilizar suporte, não tem, como a *Oracle*, equipas especializadas e focadas nisso que inclusivamente em casos mais complicados se podem deslocar ao local para fornecer suporte.

No *OpenCRX*, por exemplo, o suporte é maioritariamente através do fórum o que faz com que o tempo de resposta possa ser demasiado e imprevisível o que pode ser prejudicial para o projeto.

2.1.2 Tendências CRM

Conhecidos os pressupostos gerais de CRM, é importante compreender quais as funcionalidades típicas desempenhadas por produtos de CRM, bem como quais as tendências mais promissoras na área.

Um *software* de CRM tradicional foca a sua atuação em 3 componentes principais: Marketing, Vendas e Serviços. Ao gerir estas 3 áreas capacita a empresa de um conhecimento transversal de cada cliente ao longo de todos os processos que este estabelece com a empresa.

No sector de **Marketing**, as funcionalidades mais comuns envolvem a gestão de oportunidades de vendas, de campanhas, a análise de quais as campanhas/produtos mais

adequados para cada cliente e ainda o estudo dos clientes potenciais. Neste nível, um sistema de CRM procura impulsionar as vendas e os lucros através da gestão de processos que permitem por um lado a manutenção de clientes, por outro, a aquisição de novos.

No componente de **Vendas** são disponibilizadas funcionalidades que possibilitam a gestão dos processos de venda, de produtos e suas especificações, propostas, encomendas e outros. Pretende-se que estas funcionalidades melhorem os processos associados às vendas, potenciando consequentemente os lucros gerados.

Com o pressuposto de que um cliente satisfeito se mantém fiel à empresa e atrai novos clientes, um sistema CRM deve atuar também ao nível dos **serviços** pré e pós venda fornecidos pelas empresas. Para tal, possibilita a criação e gestão de serviços prestados ao cliente tais como, incidentes, dúvidas, interações, etc. Estas interações com o cliente podem ocorrer via *email*, *call center*, fóruns ou até mesmo em loja ou edifício físico da empresa, dependendo do sector específico da mesma. Através de métodos de calendarização de serviços, relatórios e acompanhamento de cada processo, as especificações de serviços de um CRM procuram maximizar a qualidade dos serviços prestados aos clientes, aumentando a sua satisfação.

Esta visão do CRM atual é generalista, descrevendo as funcionalidades globais e *core* de um sistema deste tipo. Em função do sector em que ele é adoptado é necessário adequar os conceitos aos processos de negócio específicos. Por exemplo, se numa empresa de telecomunicações os conceitos de produtos, vendas, campanhas e serviços são bastante explícitos e facilmente individualizáveis, num serviço de Segurança Social esses conceitos não são tão imediatos e como tal, tem de haver uma adaptação do sistema a essa realidade.

Com o crescimento do CRM e com uma necessidade cada vez maior de compreender os clientes, novas tendências têm surgido. De acordo com um estudo desenvolvido por alguns elementos da equipa de CRM da Novabase no ano de 2013, as grandes tendências neste sector são:

- **Social CRM:** Baseia-se no recurso às redes sociais e à informação por elas disponibilizadas para capacitar as organizações de um maior nível de conhecimento daquilo que são os seus clientes e as suas necessidades (16);
- **Big Data:** Como meio para aumentar as capacidades de *Marketing* e Vendas, verifica-se que o desenvolvimento de ferramentas capazes de lidar com grande volume de dados se reveste de grande interesse (16);
- **Conhecimento do Contexto:** Refere-se à capacidade de um sistema se adaptar ao contexto, apresentando um interface adaptado, uma precisão superior no que respeita à aquisição de informação e outras características capazes de o tornar melhor para cada situação específica em que é utilizado (16);
- **Análise de Sentimentos:** Através do recurso a técnicas de processamento da linguagem natural, análise textual, entre outras, refere-se à capacidade de um sistema fazer uma análise de sentimentos e poder interpretar e analisar a subjetividade subjacente às informações disponibilizadas (16);
- **Appification:** Esta tendência refere-se ao desenvolvimento de vários módulos para a mesma aplicação de forma a que cada organização possa escolher aquele (s) que mais se adaptam às suas características e necessidades técnicas (16).

Das tendências referidas salientamos o *Social CRM* que, pela sua relevância no contexto atual, se analisa mais pormenorizadamente em seguida.

2.1.3 *Social CRM*

Hoje em dia as empresas sentem uma pressão cada vez maior para estar onde os seus clientes estão e o centro de atividade dos consumidores é cada vez mais o ambiente virtual, especificamente as redes sociais. Este sector reveste-se de um potencial imenso que não está ainda a ser explorado ao máximo (17).

Através das redes sociais as empresas têm ao seu dispor, de forma gratuita e genuína, informações relativas às preferências dos consumidores, o seu nível de satisfação face à marca, as suas necessidades, etc. Apresentam-se então como um veículo de informação que, quando utilizada de forma consistente e coerente, se pode revelar preciosa para fornecer produtos e serviços cada vez mais à medida daquilo que é o que o consumidor procura.

Uma integração com as redes sociais bem estruturada permite atuar não só ao nível do marketing, ao canalizar campanhas e propostas de forma mais eficiente, mas também ao nível das vendas e serviços, possibilitando um aumento significativo da qualidade dos serviços prestados (17).

O surgimento desta nova componente vem introduzir mudanças significativas no paradigma tradicional do CRM. As empresas perdem o controlo da relação com o cliente, tornando-se este último no condutor das interações. Esta mudança tem um impacto significativo na empresa e pode levar a uma melhoria das estratégias de marketing, vendas e serviços de uma forma sem precedentes.

O objetivo é atingir um sistema que, mais do que gerir os clientes, estabelece um compromisso com eles, tornando a empresa capaz de ir ao encontro das necessidades pessoais de cada um sem perder o foco nos objetivos próprios (18).

Muitas são as empresas que já adotaram SCRM e verifica-se que os resultados obtidos são bastante positivos, principalmente em áreas como alta tecnologia, bens de consumo e retalho, em que se exigem inúmeros processos de suporte ao cliente.

O SCRM é assim uma das vertentes do CRM a acompanhar de perto e na qual o investimento se pode revelar verdadeiramente interessante.

2.2 CM+

2.2.1 O que é?

O CM+ é um componente de *software*, desenvolvido por uma equipa da *Celfocus*, que implementa processos de gestão de clientes. Através do encapsulamento de lógica e regras de negócio, faz o mapeamento entre processos de *frontend* e as entidades de CRM de um dado sistema (19).

De forma simplificada pode dizer-se que o CM+ disponibiliza uma maneira prática e simples para a gestão da base de dados de clientes e serviços de uma dada empresa, bem como das entidades que lhes são relativas.

O seu desenvolvimento teve um alvo muito específico, a indústria das telecomunicações, cobrindo grande parte dos requisitos de CRM deste sector para armazenar e gerir a carteira de clientes da operadora. No entanto, em função da operadora e do projeto em específico, este componente pode ser alvo de customização ao nível *core* para que possa responder o melhor possível aos requisitos definidos (19).

O CM+ surgiu no âmbito do desenvolvimento de um sistema mais completo e abrangente que analisaremos em seguida. No entanto é possível individualizar o CM+ dos restantes componentes desse sistema e aplicá-lo em outros contextos. Em seguida analisa-se o contexto em que foi desenvolvido para que melhor se compreenda toda a envolvimento do CM+.

2.2.2 Caso Prático

Como dito anteriormente, o CM+ teve origem no âmbito do desenvolvimento de um projeto de maior dimensão. Esse projeto consistiu no desenvolvimento de um sistema de informação para uma operadora de telecomunicações africana.

A referida operadora lidera o mercado desde 2001 e atualmente, a tem mais de 8 milhões de clientes.

Antes de 2012, a empresa apresentava um Sistema de Informação complexo, heterogêneo e com grandes limitações de performance. À medida que se requeriam novos serviços, eram adicionados componentes sem ter em conta a arquitetura geral nem a integração com outros sistemas.

Isto gerou um sistema com inúmeros pontos de falha que comprometiam o bom funcionamento dos serviços de informação da empresa e, numa empresa com esta dimensão, exigia-se um sistema de informação de qualidade e *performance* asseguradas.

De forma a possibilitar um salto qualitativo, a *Celfocus* propôs desenvolver um sistema que possibilitasse a melhoria do serviço ao cliente mas também o aumento da eficiência processual e operacional e a capacidade para lançar serviços e produtos inovadores para o mercado.

Assim, a intervenção da equipa da *Celfocus* neste sistema teve em mente o cumprimento de dois objetivos fundamentais: reestruturar o sistema existente, potenciando a integração entre os componentes heterogêneos existentes e integrar novas funcionalidades quer de Gestão de Produtos e Serviços quer de CRM, sempre tendo em conta as melhores práticas para o domínio das telecomunicações.

Atualmente, o produto desenvolvido reflete um Sistema de Informação completo que agrega funcionalidades de Gestão de Clientes, Gestão de Produtos e Serviços, Gestão de Pedidos, de Gestão Empresarial, entre outros. Em termos funcionais o novo sistema permite a realização de procedimentos de **suporte ao cliente** (ao nível de loja e *call center*), de **vendas** e **gestão backoffice**. A figura 2.3 esquematiza a arquitetura do sistema implementado (20).

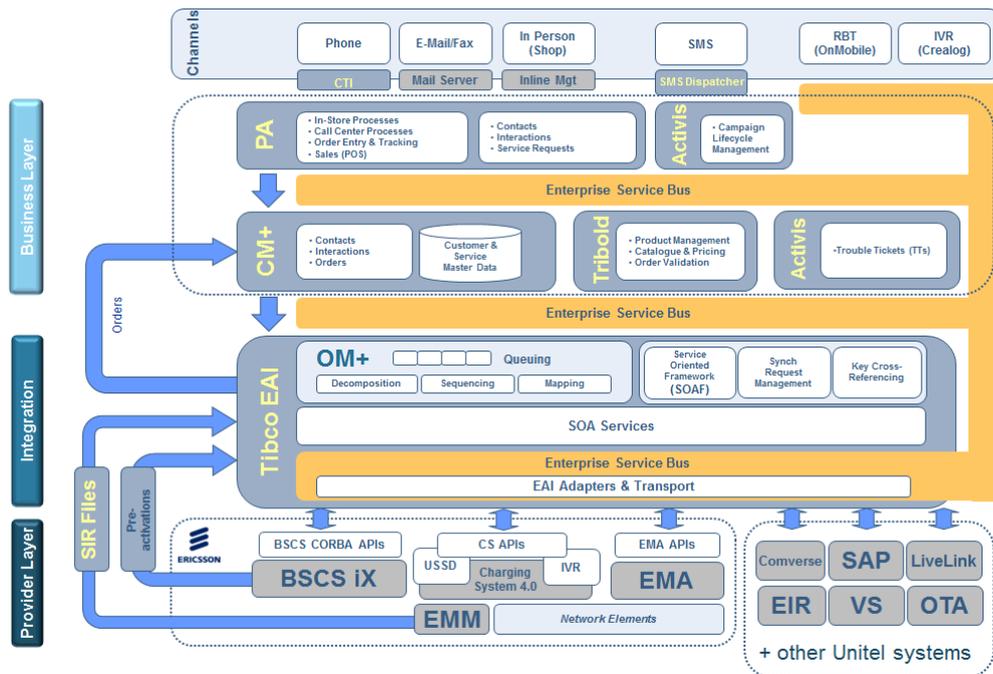


Figura 2.3- Arquitetura do Sistema de Informação

Nesta arquitetura podem distinguir-se 3 camadas:

Camada de Negócio: Aqui encontra-se o Portal de Atendimento (PA), onde é implementada a *interface* de comunicação com o utilizador, o servidor *web* e os canais de comunicação entre estes componentes e os restantes elementos da arquitetura. O PA é ainda responsável por algumas funcionalidades mais simples tais como a autenticação no sistema.

A este nível encontramos ainda o CM+, o *Tribold* e o *Activis*. O CM+ implementa a componente lógica do sistema para os processos de gestão de clientes. O *Tribold* é o responsável pela gestão de produtos, catálogos e serviços (EPM) da companhia. O *Activis* desempenha a este nível um papel importante quer para a gestão do ciclo de vida das campanhas quer para a gestão de *Trouble Tickets*.

Camada de Integração: Nesta camada encontra-se o sistema OM+ (*Order Management*), responsável pela gestão de pedidos, e ainda os componentes responsáveis pela Integração de Sistemas (*Enterprise Service Bus*). A Integração de Sistemas recorre ao uso do *software Tibco* e possibilita a conexão com sistemas de outras áreas tais como ERP, *Billing*, etc.

Apesar de não ser exclusiva desta camada, sendo transversal a todo o sistema, a integração encontra aqui uma dimensão mais considerável.

Camada de Fornecedores: A *Provider Layer* contém os restantes sistemas em integração com o principal, tais como SAP, *LiveLink*, *Converse*, etc.

Desta arquitetura há que destacar o CM+ e o *Tribold* visto serem a principal fonte de dados. O CM+ como fonte de dados de clientes, contas, serviços, tipificações, contactos, moradas, etc. e o *Tribold* como fonte de dados de produtos, serviços, taxas de ativação/reativação, taxas de renovação.

2.2.3 Entidades

Ainda antes de conhecer a arquitetura de acordo com a qual o CM+ está desenvolvido, é importante conhecer as entidades, mapeadas na base de dados que lhe está associada, e as relações entre elas, de forma a compreender melhor o funcionamento e estrutura deste sistema.

Em seguida apresenta-se uma listagem resumida das entidades consideradas mais relevantes. No anexo C pode encontrar-se uma descrição mais detalhada e aprofundada de todas as entidades.

Customer

Um *Customer* (cliente) refere-se a um comprador ou utilizador dos produtos da empresa ou, em alguns casos, pode referir-se a clientes de empresas semelhantes.

Account

Uma *Account* (conta) representa um conjunto de bens ou serviços passíveis de serem adquiridos ou usados por um cliente e pelos quais ele vai ser cobrado. Uma conta está associada a um e apenas um cliente, mas um cliente pode ter associadas várias contas.

Address

Um *Address* (endereço) pode representar ou um endereço físico do cliente ou o seu endereço electrónico. Esta entidade associa-se à entidade contacto, sendo que um contacto pode estar associado a zero ou mais endereços (19).

Contact

Um Contacto pode representar ou uma pessoa ou uma empresa. Um contacto está sempre associado a um cliente, sendo que um cliente apesar de poder ter vários contactos associados tem de ter sempre um selecionado como contacto principal (19).

Um contacto pode também estar associado a uma conta. Neste caso uma conta pode ter vários contactos associados mas, tal como na relação com o cliente, um desses contactos tem de ser selecionado como contacto principal de cobrança.

Interaction

Uma interação cria uma instância de comunicação entre a entidade Cliente (ou Conta) e a operadora, assim como instancia o seu objetivo e descrição (19).

Esta entidade pode ser utilizada para o registo de eventos tais como uma chamada para o *call center*, uma ida à loja, um *email*, um SMS, etc.

Product

Um produto representa, no contexto das telecomunicações, um bem ou serviço que é necessário entregar para que o cliente possa usar um certo serviço (19). A representação de produtos feita no CM+ é mínima, sendo que as únicas funcionalidades implementadas no CM+ relativamente a produtos é a sua criação nas tabelas *Prod* e *Prod_Assoc* e a sua associação aos *Assets* correspondentes. Todas as restantes funcionalidades inerentes à gestão de produtos são implementadas pelo *Tribold*.

Order

Uma ordem representa o pedido para criação ou modificação de um conjunto de *Assets* e atributos correspondentes (19). O pedido que origina a criação e configuração de uma ordem pode ser iniciado por um cliente (como por exemplo a suspensão ou alteração de um serviço) mas pode também ser um pedido gerado automaticamente em função do ciclo de vida de uma entidade.

2.2.4 Arquitetura

A figura 2.11 esquematiza a arquitetura do CM+. Como é observável, o CM+ comunica diretamente com a base de dados que armazena informação acerca de clientes e dados com estes relacionados, tais como contas, contactos, interações, etc. Em seguida descrevem-se as camadas segundo as quais se estrutura a arquitetura interna do CM+.

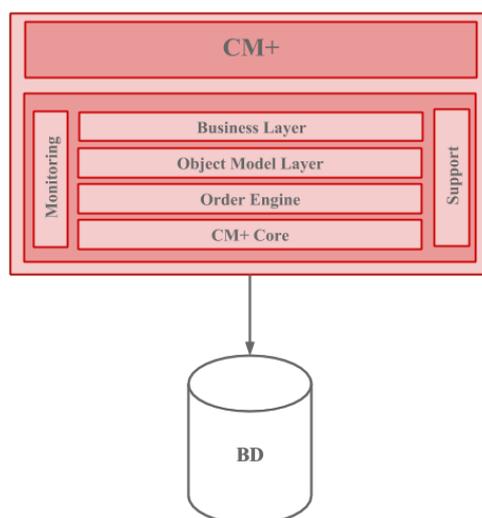


Figura 2.4 - Arquitetura do CM+

1) Camada de Negócio

Aqui encontram-se todos os serviços funcionais e a lógica de negócio que lhes corresponde. Esta camada está exposta aos outros sistemas via *Web Service* (JAX-WS) e mapeia os pedidos vindos de sistemas externos (como o PA) para o modelo de dados do CM+.

A este nível encontram-se também alguns métodos genéricos que ajudam no desenvolvimento de novos serviços. Assim, quando se pretende desenvolver novas funcionalidades ou personalizar as existentes, é nesta camada que se concentra a maior percentagem de trabalho requerido.

2) Modelo de Dados (ADF)

Esta camada representa o modelo de dados do CM+. Este foi desenvolvido com recurso à *framework* ADF *Business Components* da Oracle. O ADF é de livre licenciamento e mapeia as tabelas criadas ao nível da Base de Dados ao mesmo tempo que mantém os dados e a informação na aplicação antes de os submeter para a base de dados (21).

Os *Business Components* do ADF consistem em 3 elementos fundamentais:

- **Entity Object:** Representa uma linha numa tabela da base de dados e simplifica a modificação dos seus dados através da gestão “automática” da DML. Este componente encapsula lógica de negócio para cada linha da tabela, de forma a garantir que as regras de negócio estão a ser cumpridas consistentemente. Um *Entity Object* pode estar associado a outros *Entity Objects* para refletir as relações estabelecidas na base de dados.
- **View Object:** Representa uma *query* SQL. Estabelece a ligação entre vários *Entity Objects* de forma a criar hierarquias de qualquer complexidade. Quando o utilizador final modifica dados no PA os *View Objects* colaboram com os *Entity Objects* para que, de forma consistente, as modificações sejam validadas e gravadas.
- **Application module:** Componente de transição que os clientes usam para lidar com os dados de aplicação. Define um modelo de dados e procedimentos e funções de alto nível.

Criando vários *Entity Objects*, definindo as relações entre eles, através dos *View Objects*, e adicionando-os a *Application Modules* torna-se possível implementar regras de negócio, criando um modelo de dados coeso.

3) Motor de Regras

Responsável pela geração e submissão de ordens, compõe-se por várias etapas configuráveis em função dos diferentes cenários de aplicação. Implementa um mecanismo de *backup* utilizado para armazenar e recuperar a configuração de um certo momento (quando necessário). Sustenta pedidos futuros, encomendas (via OM+), pedidos massivos (via PA) e ainda implementa uma pool de execução de pedidos (22).

De notar que, apesar do CM+ possibilitar a representação de ordens, não suporta a sua gestão. No que respeita a ordens, o CM+ apenas as recebe, instancia-as, faz a sua validação através do *Rule Engine* (motor de validação de ordens) e depois reencaminha-as para um sistema de gestão de ordens, como o OM+. Esse sistema será depois o responsável pela decomposição, sequenciamento e mapeamento das ordens e, quando a ordem for completada, invocará o seu fecho ao CM+ (22).

Quando o CM+ recebe um pedido de um sistema de *frontend*, como o PA, para executar uma ordem, vai desencadear 2 processos principais: **asset to order** e **order to asset**. O primeiro consiste no mapeamento do pedido para o *asset* que lhe corresponde, criando uma ordem para essa entidade CRM.

Depois de criada a ordem, tem de ser validada por processos internos do CM+, implementados na *Rule Engine*, e posteriormente a ordem é concretizada e o *asset* modificado, ocorrendo assim o processo de **order to asset** (22).

Para clarificar este processo podemos tomar como exemplo um pedido de alteração de tarifário. Quando um operador executa esse pedido no PA ele é encaminhado para o CM+. Este por sua vez vai criar uma ordem, associada ao **Contrato de Serviço** da **Conta** respetiva. A este nível duas ações (no mínimo) têm de ocorrer:

- Desativação do tarifário atual;
- Ativação do novo tarifário.

Neste momento já ocorreu o processo *asset to order* e a ordem tem agora de ser validada de acordo com regras de negócio específicas. O **Rule Engine** faz a sua validação e caso esta seja bem-sucedida a ordem é encaminhada para o OM+ e a modificação é assumida para o *asset* correspondente. A figura 2.12 esquematiza este processo.

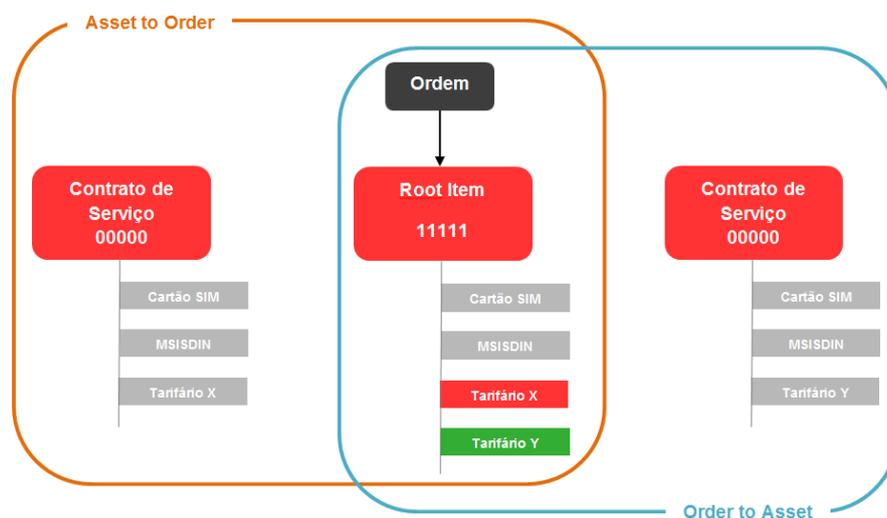


Figura 2.5– Execução de uma Ordem

4) Core

Fornece/disponibiliza suporte aos seguintes serviços: *Transaction Based Operations (commit/rollback)*, extração de dados, tratamento de exceções, gestão de prioridades, entre outros. Para além disso disponibiliza um motor de JMS (*java message service*) tolerante a falhas e permite *Pool Sessions* sincronizadas (o que melhora a performance do sistema e suporta elevada concorrência).

5) Monitoring/SLA

As funcionalidades implementadas nesta camada possibilitam a monitorização de eventos em tempo real. Estes eventos podem ser pré-configurados, através do portal de administração do CM+. Associados a cada evento podem estar intervalos de duração e ações por ele despoletadas (como por exemplo enviar um SMS, ou um *email*, ou cancelar um dado serviço).

Como exemplo pode ver-se o caso de um dado serviço que seja válido apenas por um período de 30 dias. Através do portal de Administração um evento é criado, e estabelecido um limite de 30 dias. Quando o sistema detecta a ocorrência desse evento, é despoletado o processo de cancelamento do serviço, de forma automática.

6) Suporte

É construída com base na *ADF Faces Framework* e possibilita visibilidade para diferentes componentes do CM+: configuração de metadados, configuração da *Order Engine*, etc.

2.2.5 CM+ como CRM

Descrição

O CM+ desempenha, como visto, funcionalidades de gestão de clientes de uma empresa. Mas, impõe-se a questão, “O CM+ é um sistema de CRM?” A resposta mais imediata neste momento poderia ser: “Não”, visto que o CM+ tem em falta funcionalidades de marketing, vendas e serviços que são o âmago do CRM.

No entanto, é importante uma resposta mais completa e correta que permita compreender qual a posição do CM+ no contexto de CRM e, acima de tudo, qual o seu potencial nessa área.

A resposta pode dividir-se em duas partes, uma mais conceptual e outra mais técnica, relacionada com as funcionalidades e metodologia de implementação.

Como visto anteriormente, o CM+ é um componente de *software* que engloba lógica e processos de negócio. Individualmente este componente pode ser transportado para outros contextos, no entanto, não é por si só um *software* completo. Isto é, o CM+ não tem implementação de serviços de *frontend* para interação com o utilizador, nem tem funcionalidades de monitorização de ordens, nem do catálogo de produtos.

Assim, em termos de conceito, vemos que para o CM+ ser considerado um *software* de CRM tem de, à partida, ter associado, no mínimo, um sistema de frontend (como o PA).

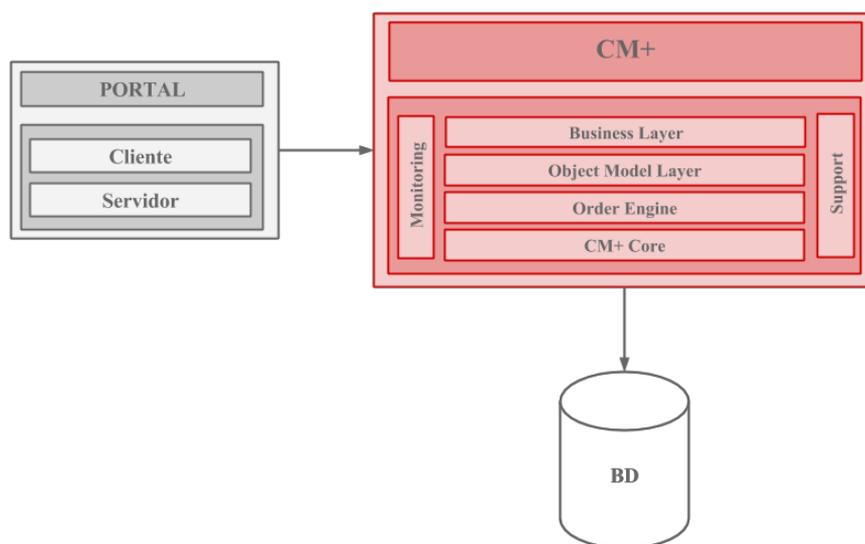


Figura 2.6 - Arquitetura do CM+ como CRM

A segunda parte da resposta tem um fundamento mais técnico. O CM+ mapeia entidades típicas de CRM como cliente, conta, contacto, entre outros. Através desse

mapeamento implementa algumas funcionalidades básicas de CRM como a gestão da carteira de clientes, a criação de incidentes/serviços, etc. No entanto, a noção de produto do CM+ é mínima e conceitos como venda, campanha ou oportunidade são inexistentes.

Dado isto, estamos perante um sistema sustentado por um modelo de entidades-relacionamento sólido e passível de extensão e já com um grande leque de funcionalidades implementadas. Apesar de não podermos dizer que o CM+ seja por si só um CRM, podemos dizer que se constitui como uma base sólida capaz de se tornar num sistema que abranja as funcionalidades consideradas nucleares de CRM.

Não sendo parte integrante do CM+, no sistema geral estudado, encontram dois elementos que desempenham funções que se consideram relevantes para este âmbito, o PA e o OM+. O PA visto ser o *frontend* que estabelece a ligação entre o *end-user* e o sistema, e o OM+ por se tratar do sistema gestor de ordens, um componente muito importante também num contexto de CRM. Como tal, considera-se importante analisá-los mais profundamente. No anexo D pode encontrar-se essa mesma análise.

Comparação com Outras Tecnologias CRM

Como visto, o CM+ não se constitui atualmente como um *software* de CRM ou, pelo menos, não como um software de CRM completo.

Tendo em mente o objetivo que guia o presente trabalho, é então importante compreender o que separa o CM+ de um *software* de CRM “puro”, como os apresentados anteriormente.

A tabela seguinte apresenta as funcionalidades anteriormente descritas como sendo transversais aos produtos analisados e esquematiza quais as que estão presentes ou não no CM+.

É saliente a ausência de funções fulcrais de CRM. No fundo, o CM+ funciona como uma janela para a base de dados de clientes e produtos da empresa, permitindo consultá-la e modificá-la, fazendo, no fundo, a sua gestão. Ficam de fora as dimensões de marketing e vendas e a área de serviços é assegurada de forma limitada.

O CM+ foi desenvolvido para um sector muito específico, o das telecomunicações e cumpre requisitos muito inerentes a esse mesmo sector, o que pode em parte justificar a ausência de muitas funcionalidades de CRM.

Um *software* como o *Dynamics CRM* ou o *OpenCRX* é concebido para se adaptar a qualquer sector e dimensão e isso faz com que abranja um conjunto de funcionalidades muito maior.

Quanto ao custo, visto ser um *software* desenvolvido à medida para cada negócio e não exigir licenciamento (à exceção da base de dados), o custo de implementação do CM+ acaba por ser menor do que aquele associado ao *Dynamics CRM*, o que no contexto atual representa uma enorme vantagem competitiva.

De acordo com as funcionalidades descritas para os produtos mais vendidos, apresentam-se na tabela 1 quais aquelas que estão presentes ou não no CM+.

Características	CM+
Gestão de Contas	●
Gestão de Contatos	●
Gestão de Produto	●
Gestão de Venda	●
Gestão de Tarefas	●
Gestão de Marketing	●
Serviço ao Cliente	●
Reporting	●
Gestão de Incidentes	●
Social CRM	●
Mobile CRM	●

Tabela 2.1 - Características do CM+

Neste ponto, considera-se interessante fazer um breve resumo de quais as vantagens e desvantagens do CM+ face à generalidade dos produtos CRM à luz de uma análise SWOT (figura 2.7).

Considera-se que em termos de produto a força deste sistema reside na sua simplicidade e flexibilidade. Em termos de competitividade externa relativamente a outros produtos, o CM+ tem como oportunidades um baixo custo e uma elevada capacidade de integração com outros sistemas devido à sua modularidade.

Por outro lado, a falta de competências de gestão de marketing e serviços revelam algumas fraquezas do CM+ enquanto produto de CRM e, o fato de não ser um sistema com uma grande carteira de clientes que possam confirmar a sua validade e credibilidade é também um ponto negativo a ter em conta.



Figura 2.7 – Arquitetura do CM+ como CRM

Capítulo 3

Metodologia e Objetivos

3.1 Metodologia

Face ao contexto de realização do presente projeto considera-se como modelo de trabalho uma metodologia tradicional com desenvolvimento iterativo. Este tipo de metodologia caracteriza-se por um conjunto de etapas bem definido em que, de cada qual, resulta um *output*, que pode ser um documento, um protótipo de *software* ou o produto final. Cada um desses resultados deve ser revisto e, consoante as falhas detectadas, dever ser sujeito a melhorias.

O esquema seguinte apresenta as etapas consideradas para o desenvolvimento deste projeto.

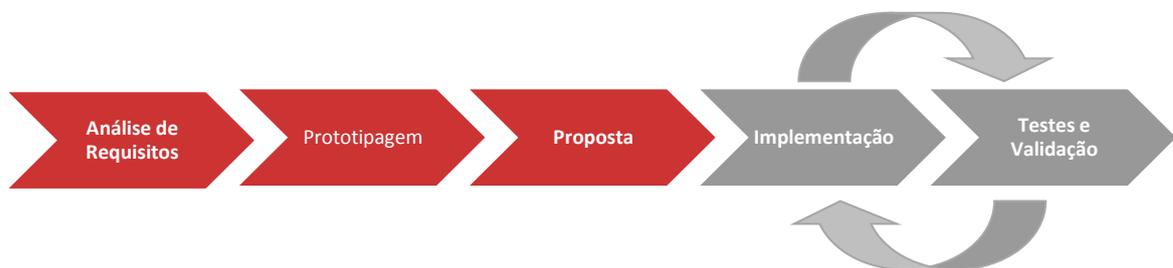


Figura 3.1– Fases do Processo de Implementação

No final das duas primeiras fases deve ter sido construído um documento que agregue a definição clara de quais os requisitos (funcionais e não funcionais) que devem ser implementados, bem como a prioridade de cada um deles. Para além disso deve ser apresentado um protótipo completo que simule o funcionamento do produto e que permita a compreensão clara daquilo que ele possibilita fazer.

O documento resultante deve gerar um debate de ideias entre aquilo que são as expectativas quanto ao sistema e aquilo que é concretizável e definir claramente o que será realizado. Consoante isso pode elaborar-se a proposta, definindo estratégias de implementação e estipulando quais os processos e em que componentes devem ser criados.

A partir do momento que sejam claros os requisitos e a forma de implementação pode prosseguir-se para a fase de implementação que decorrerá de forma iterativa em função dos requisitos definidos. Em função das decisões tomadas e dos possíveis problemas encontrados, pode haver reformulação dos requisitos.

Finalmente, decorre a fase de Validação, em que se detetarão possíveis erros e falhas do sistema. Em função deles, devem ocorrer processos de implementação para proceder à sua correção.

No final, como *output* de todo o processo deve resultar o produto prototipado, apto para utilização.

3.2 Definição de Objetivos

Como visto, o *Social CRM* é uma das grandes tendências do momento e praticamente todos os produtos analisados apresentavam componentes de integração com redes sociais. No entanto, verifica-se que essa integração se dá maioritariamente ao nível de funcionalidades de *marketing* e impulsão de vendas. No que respeita a serviços, o *Social CRM* não está ainda muito explorado e uma das maiores falhas verificadas é o facto de não ser possível a criação de serviços via redes sociais.

Paralelamente verifica-se que, ao nível de serviços, nos produtos analisados não há unificação da comunicação. Isto é, apesar de permitirem a criação de incidentes por vários canais (*call center*, loja, SMS), no sistema, eles são sempre criados como instâncias diferentes, mesmo quando correspondem a interações relativas ao mesmo caso.

Imagine-se o caso em que um cliente liga para um *call center* denunciando a falha de um serviço. No momento o operador sugere uma resolução e o cliente desliga e testa a solução. Caso não funcione e o cliente não possa ligar de volta mas possa publicar na página do *Facebook* o problema, a situação ideal era que o sistema à partida soubesse que aquele problema correspondia a um incidente já instanciado e fossem listadas todas as interações conjuntamente. Nos sistemas atuais isto não é ainda uma funcionalidade muito explorada.

Desta forma, considera-se que com o desenvolvimento de forma acoplada destas duas funcionalidades se consegue gerar um produto bastante interessante para o mercado atual. Por um lado tem-se a integração com redes sociais na área dos serviços, que é um tema emergente, por outro garante-se um sistema mais adaptado à realidade dos nossos dias, em que os canais de comunicação estão em constante oscilação.

Uma outra funcionalidade não muito comum nos sistemas de CRM analisados e que apresenta algum potencial prático é a implementação de um **sistema de recomendação de soluções de incidentes**. Mais uma vez, foco na área de serviços pela sua importância no que toca à satisfação dos clientes.

O objetivo desta funcionalidade é o de implementar um algoritmo que, perante os vários incidentes que vão sendo criados, apresente uma listagem das soluções mais adequadas para cada problema-tipo, baseando-se em casos anteriores. Esta funcionalidade contribuiria fortemente quer para a satisfação do cliente, quer para a redução do tempo de resolução de problemas, tornando o sistema mais eficiente.

Tendo tudo isto em conta e juntando o objetivo estipulado inicialmente, desenvolver funcionalidades que aproximem o CM+ de um sistema de CRM transversal a vários sectores, consideram-se os seguintes objetivos:

- **Desenvolvimento de Métodos de Gestão de Incidentes:** expansão das funcionalidades do CM+ de forma a possibilitar gestão de incidentes unificada, possibilitando múltiplas interações (de múltiplos canais) no âmbito de um mesmo incidente;
- **Integração com Rede Social para a gestão de Incidentes:** integrar o CM+ com rede social, permitindo a denúncia de problemas e dúvidas por esta via. A criação dos incidentes deve ser automática, o operador deve ser notificado e deve poder dar-lhe resposta pelo mesmo canal de comunicação.

- **Implementação de um Sistema de Recomendação de Soluções de Incidentes:** o sistema deve ser capaz de analisar os casos ocorridos anteriormente e sugerir as soluções que mais se adequam aos casos em aberto.

Em termos teóricos pretende-se que este projeto seja uma demonstração de como novos temas emergentes podem ser aplicados ao CM+ para que se torne mais inovador e esteja ao nível daquilo que são as exigências do mercado.

Capítulo 4

Análise de Requisitos

O processo de detalhe e análise de requisitos tem por objetivo descrever as funcionalidades e características do sistema a desenvolver face às necessidades dos clientes. Quanto mais completo e exímio for este processo maior a probabilidade de gerar um produto final mais próximo do desejado pelo cliente (23).

De acordo com *Bernd Bruegge* e *Allen H. Dutoit*, este processo deve dividir-se em duas partes principais, primeiro o levantamento de requisitos e posteriormente a sua análise.

No seu livro *Object-Oriented Software Engineering*, os autores defendem que um método eficaz, no que respeita à fase de levantamento de requisitos, é a apresentação de protótipos simples com algumas interações que permitam dar a entender ao utilizador como funcionará o sistema final. Estes protótipos esquematizam cenários de utilização e facilitam a comunicação entre quem desenvolve o sistema e o utilizador final. Acredita-se que desta forma se evitam erros que posteriormente causariam elevadas despesas para corrigir e ao mesmo tempo se garante que são cumpridas todas as necessidades do utilizador e não se desenvolvem funcionalidades obsoletas (23).

Para desenvolver a presente análise de requisitos seguiu-se esta abordagem. Como tal, inicialmente foram debatidas ideias sobre o que se pretendia e, de acordo com isso, elaborados alguns protótipos de cenários de utilização e com base neles foram decifradas as necessidades do sistema. Posteriormente passou-se à fase de análise de requisitos, com a divisão entre requisitos funcionais e não funcionais, como definido por Sommerville em *Software Engineering 9* (24).

4.1 Levantamento de Requisitos

Depois de debatidas algumas ideias com a equipa responsável pelo projeto foi possível delinear um esboço do que se espera do sistema final e, a partir desse esboço, desenhar alguns protótipos de utilização. Os protótipos implementados simulam o funcionamento final do sistema sem realmente realizar nenhuma ação em concreto e permitem uma melhor visualização do que será o produto final.

Os protótipos foram desenvolvidos tendo em conta o objetivo geral do projeto e a intenção de tornar o CM+ num sistema de CRM completo e transversal a diversos sectores de negócio. No entanto, esse plano é incompatível com a limitação temporal do presente projeto. Assim, apesar de ser apresentada a prototipagem total do sistema, serão selecionadas apenas algumas das funcionalidades para realizar neste âmbito.

Apresentar todos os protótipos desenvolvidos e descrever todas as funcionalidades encenadas resultaria num processo demasiado longo e possivelmente enfadonho para o leitor. Assim, opta-se por colocar em anexo a prototipagem do sistema total e, neste capítulo apenas se apresentam os motivos para a seleção de funcionalidades e os protótipos que lhes correspondem.

4.1.1 Prototipagem de Cenários

No capítulo anterior estreitou-se o caminho e foram definidos os objetivos concretos para o sistema a implementar. Como tal, neste momento, torna-se possível apresentar os protótipos relativos às ideias definidas como objetivos a implementar.

1) Página Principal

Este protótipo esquematiza a página inicial do sistema a desenvolver. Pode subdividir-se em 3 partes.

Do lado esquerdo encontram-se as áreas de funcionalidades disponíveis. Neste caso, de forma a aproximar-se ao máximo daquilo que é o conceito de CRM, divide as suas ações em 3 categorias: *Marketing*, *Vendas* e *Serviços*. A área de **Marketing** concentra as funcionalidades de gestão de campanhas e oportunidades e a elaboração de gráficos de *reporting*. Ao nível de **Vendas** apresentam-se funcionalidades de gestão de Propostas, Encomendas e Produtos/Serviços bem como serviços de *reporting*. Por fim, para os **Serviços** têm-se funções de gestão de Incidentes, Calendário de Serviços e, mais uma vez, *reporting*. Esta área lateral será transversal a todo o sistema de forma a que rapidamente o utilizador possa encontrar as funcionalidades que procura.

Na parte central são listados os avisos ou novidades relativos a acontecimentos importantes no sistema.

Do lado direito encontra-se a área “Cliente 360” onde se disponibilizam funcionalidades de gestão de clientes. Nesta área é possível pesquisar clientes por nome, número de telefone, etc., consultar detalhes a eles relativos e é ainda possível criar novos clientes.

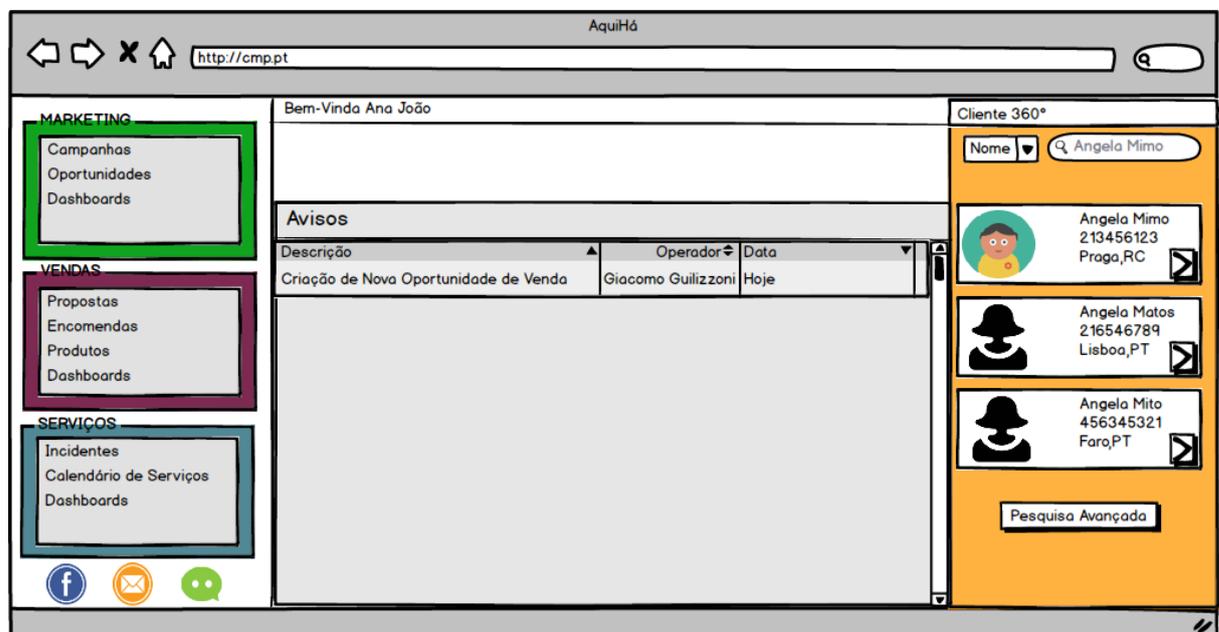


Figura 4.1– Protótipo para a Página Inicial

2) Incidentes

Escolhendo a opção “Incidentes” na área de Serviços, espera-se que seja apresentada uma janela semelhante à mostrada na figura 4.2.

Na parte central apresenta a listagem de todos os incidentes ocorridos recentemente. A esta listagem podem ser aplicados alguns filtros de forma a diminuir os resultados apresentados. Assim o utilizador pode optar por listar apenas os incidentes em aberto (ainda não resolvidos) ou listar apenas os fechados (aqueles que já foram resolvidos). Para cada incidente listado, o utilizador pode selecioná-lo e assim consultar todos os detalhes a ele referentes.

Na parte lateral direita o utilizador tem à disposição 3 tipos de funcionalidades. Através da funcionalidade de **pesquisa**, pode encontrar incidentes específicos baseados em informações que tenha à sua disposição. Por exemplo, caso pretenda pesquisar apenas incidentes do cliente X, ou pesquisar um incidente com um determinado *id*, pode fazê-lo por via dessa funcionalidade.

Ainda nessa área são apresentados **alertas** ao utilizador. Estes alertas correspondem a interações ou incidentes em aberto a cargo do utilizador corrente e aos quais ele deve dar resposta assim que possível. Ao aceder a essa funcionalidade, o utilizador poderá ver a listagem de interações a que deve responder e aceder ao processo de resposta.

Por fim, o utilizador pode **criar novos incidentes**. Esta opção é relevante para os casos em que as interações/incidentes não são criados automaticamente, como no caso do atendimento em loja ou via *call center*.

Relativamente aos incidentes é de notar que nos casos em que eles são reportados através de *email*, SMS ou por uma rede social, o incidente ou interação é criado automaticamente pelo sistema e apenas o utilizador responsável será notificado através de um alerta. Nos restantes casos, os incidentes são criados manualmente através da funcionalidade descrita anteriormente.

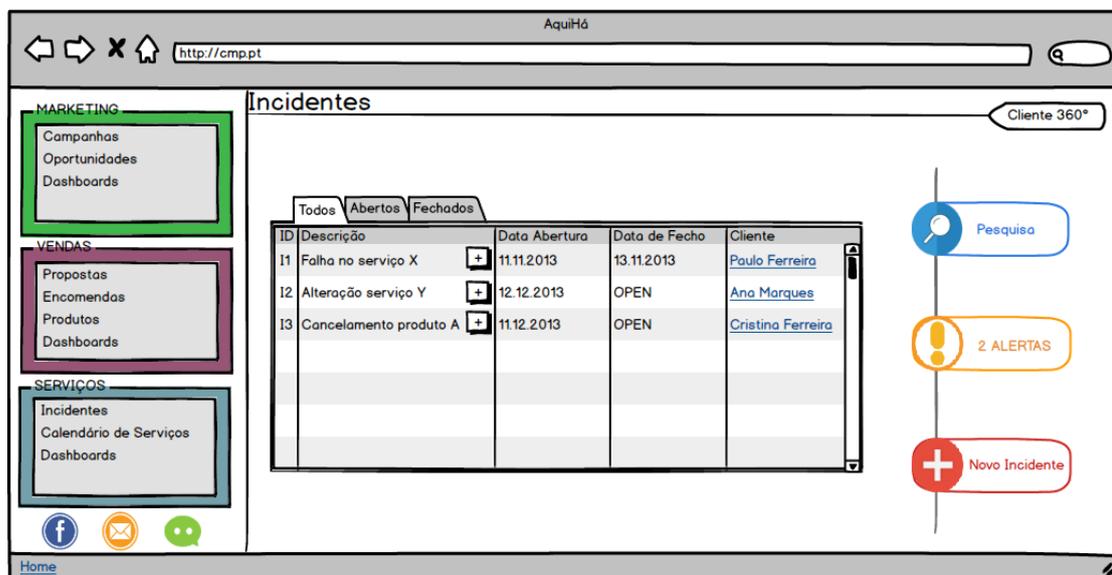


Figura 4.2– Protótipo para a página de Incidentes

3) Detalhe de Incidente

No protótipo anterior observou-se que o utilizador poderia seleccionar um incidente da lista de forma a visualizar os detalhes a ele referentes. Caso ele opte por isso, surge um ecrã semelhante ao apresentado na figura seguinte.

Neste ecrã mantém-se mais uma vez a área de funcionalidades na parte lateral esquerda e, na parte central, é possível encontrar a listagem de interações ocorridas no âmbito do incidente em causa. Nesta listagem encontra-se cada instância de comunicação, isto é, de cada vez que o utilizador comunica com a empresa no âmbito de um mesmo problema, qualquer que seja o canal utilizado, vai ser registada uma nova interação nesta lista.

Como exemplo pode observar-se o caso esquematizado. Um cliente faz uma chamada para o *call center* em que se mostra insatisfeito com o serviço Y. É criado um incidente (I2) e registada esta primeira interação que fica a cargo da utilizadora Marta Ribeiro. Esta, sugere a alteração para um de dois serviços, o A e o Z. e o utilizador fica de ponderar essa questão.

Essa interação fica resolvida mas o incidente continua em aberto. Mais tarde o utilizador comunica via Facebook que pretende aderir ao serviço Z. Na listagem de interações do incidente I2 é criada esta nova interação que fica em aberto até que a responsável confirme esta alteração.

Na parte lateral direita podem observar-se detalhes do incidente não referentes às interações. Nestes pode consultar-se o cliente a que se refere, o id de incidente, a data de início, etc.

Ainda nesta janela, na parte inferior pode visualizar-se uma área de sugestões. Pretende-se que esta apresente uma listagem de soluções possíveis para o incidente em causa. Estas sugestões baseiam-se na análise de semelhança de propriedades entre incidentes anteriores e o atual. Em função do nível de semelhança, o sistema sugere as soluções que levaram à resolução do problema nos casos mais semelhantes.

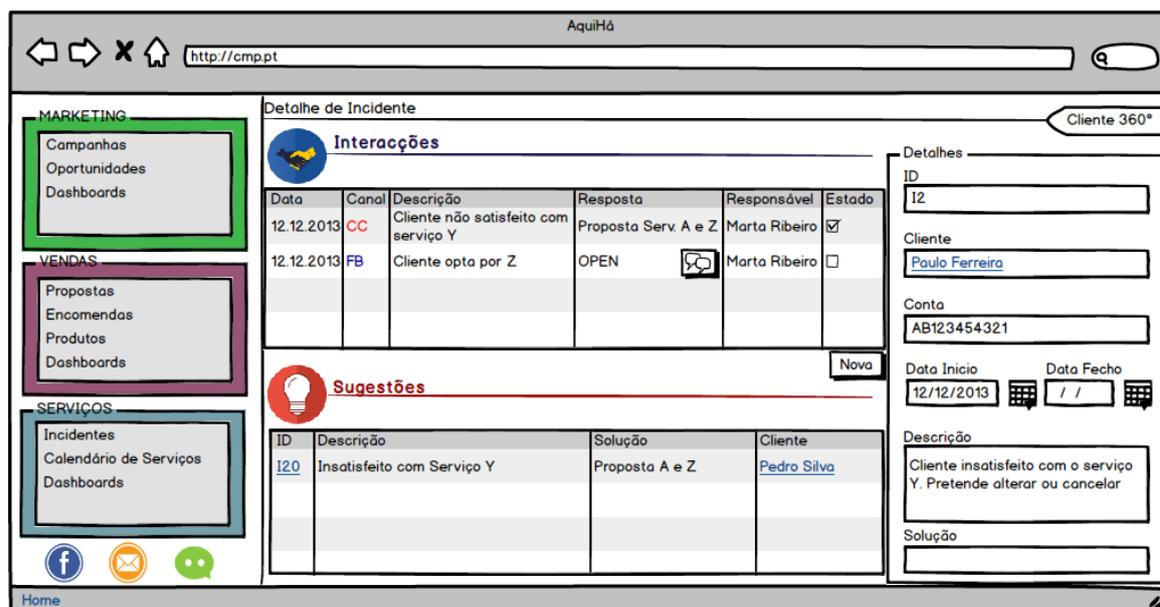


Figura 4.3– Protótipo para a página de Detalhes de Incidentes

4) Resposta a Interação em Aberto

No ecrã de detalhe de incidente, na listagem de interações o operador pode optar por responder a uma das interações em aberto (caso seja o responsável por ela). Nesse caso, surgirá um pop-up onde se determina o canal pelo qual se vai responder (por defeito é o mesmo da interação por parte do cliente) e qual a resposta.

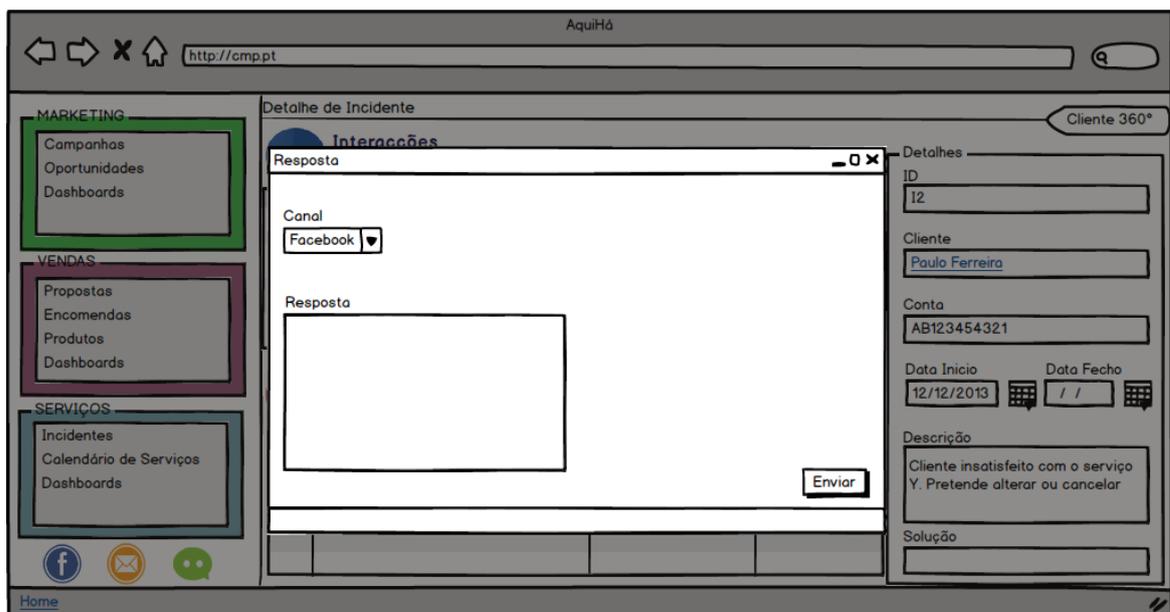


Figura 4.4– Protótipo para a página de Resposta a Interações

5) Nova Interação

No ecrã de detalhe de incidente pode também optar-se por criar uma nova interação, para casos em que isso não ocorre automaticamente como por exemplo, quando a interação é reportada via *call center* ou na loja.

Aqui o utilizador pode seleccionar o canal pelo qual recebeu a interação e fazer uma breve descrição relativa aos motivos e à descrição do cliente. Caso seja possível, pode logo dar uma resposta.

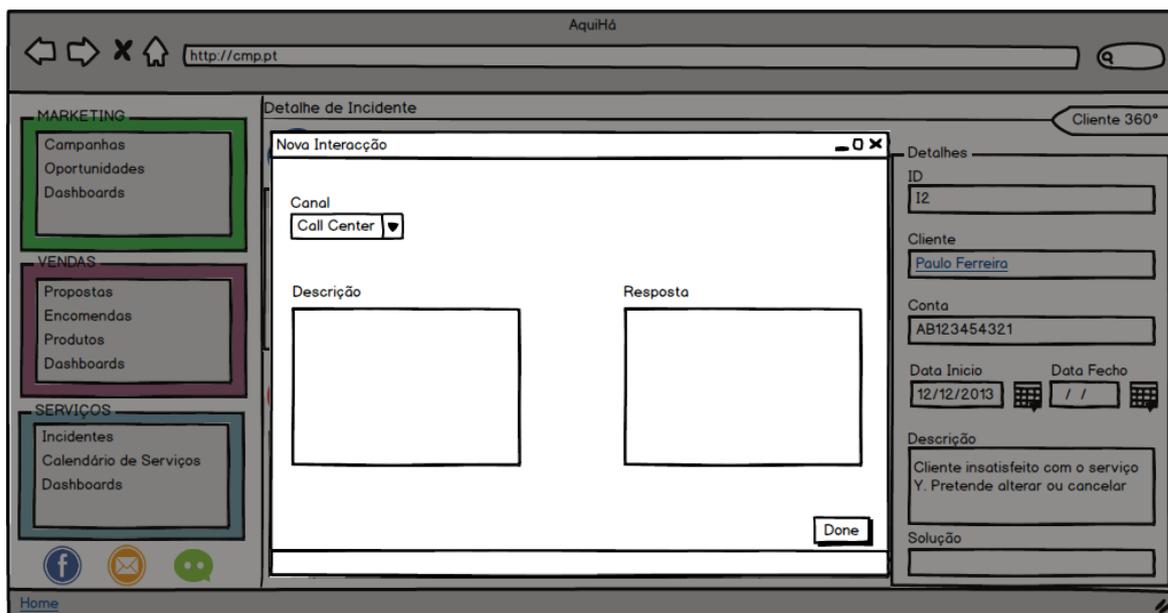


Figura 4.5– Protótipo para a página de Criação de novas interações

6) Novo Incidente

Caso o operador deseje criar um novo incidente (quando o canal não é automático, ou seja, via *call center* ou loja), aparecerá um ecrã semelhante ao esquematizado. Poderá pesquisar o cliente a que corresponde, ver quais os incidentes em aberto para ele e poderá discriminar a partida os pormenores da primeira interação.

Ao confirmar a criação do incidente, aparecerá algo semelhante ao esquematizado na figura 4.6. Ou seja, aparece o ecrã de detalhe de incidente para o novo incidente criado. Neste está já criada a 1ª interação.

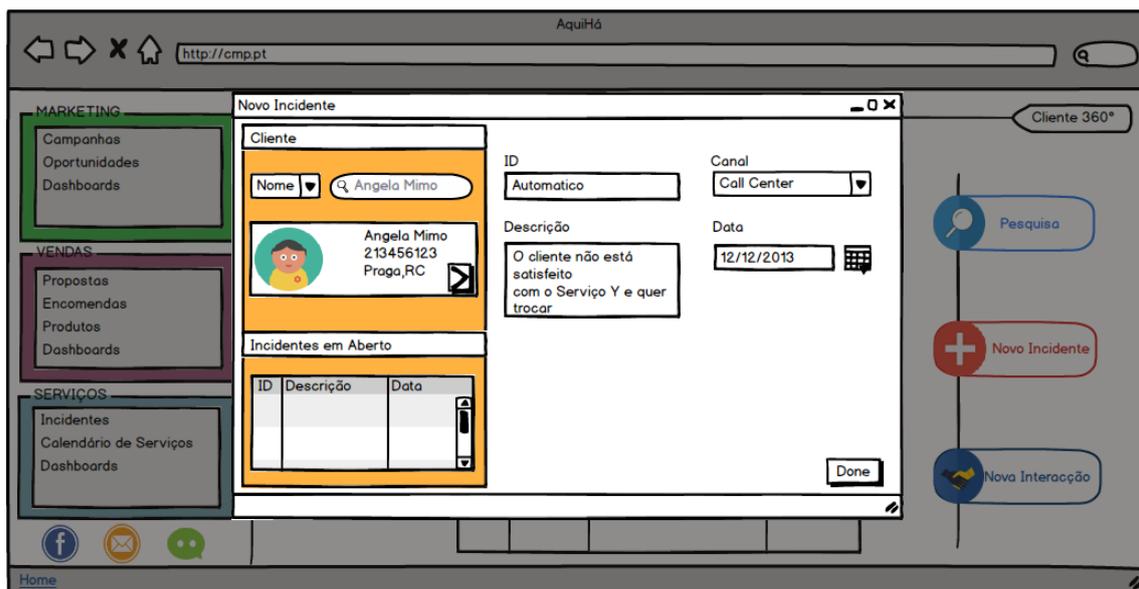


Figura 4.6– Protótipo para a página de criação de novos incidentes

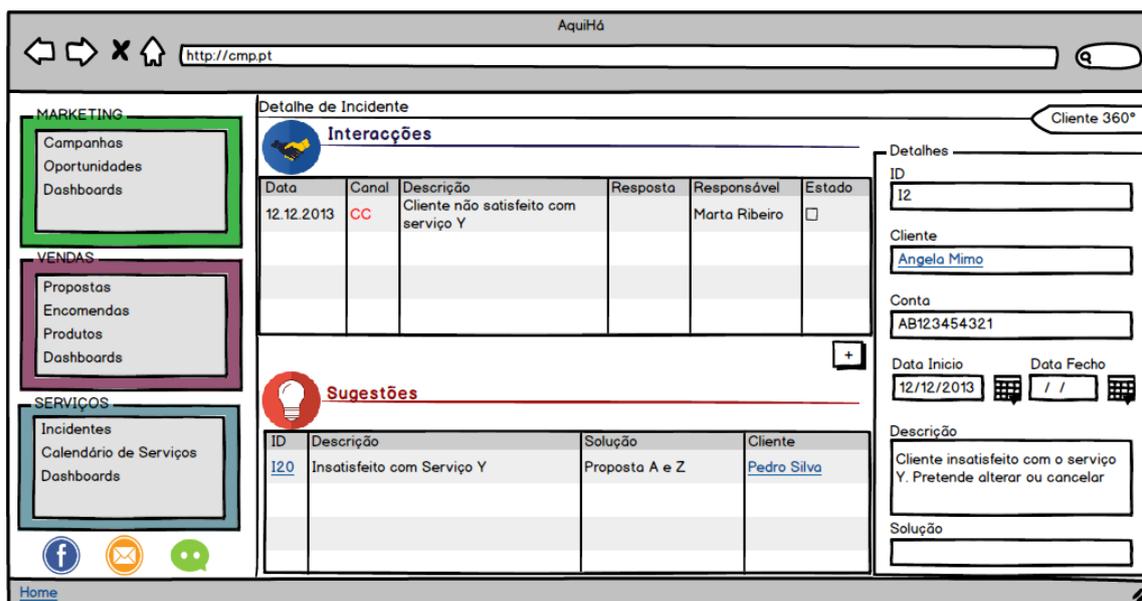


Figura 4.7– Protótipo para a página de detalhes de incidentes

7) Pesquisa

Esta funcionalidade possibilita que o operador pesquise incidentes, quer por ID, quer por nome ou número de cliente. Caso pesquise por id e esse corresponda a um incidente válido, o sistema devolve o incidente correspondente. No caso da pesquisa por cliente (nome ou número) o sistema vai devolver todos os incidentes relacionados com esse mesmo cliente.

A listagem de incidentes resultante da pesquisa permite que o utilizador selecione um deles e consulte ou altere os seus detalhes.

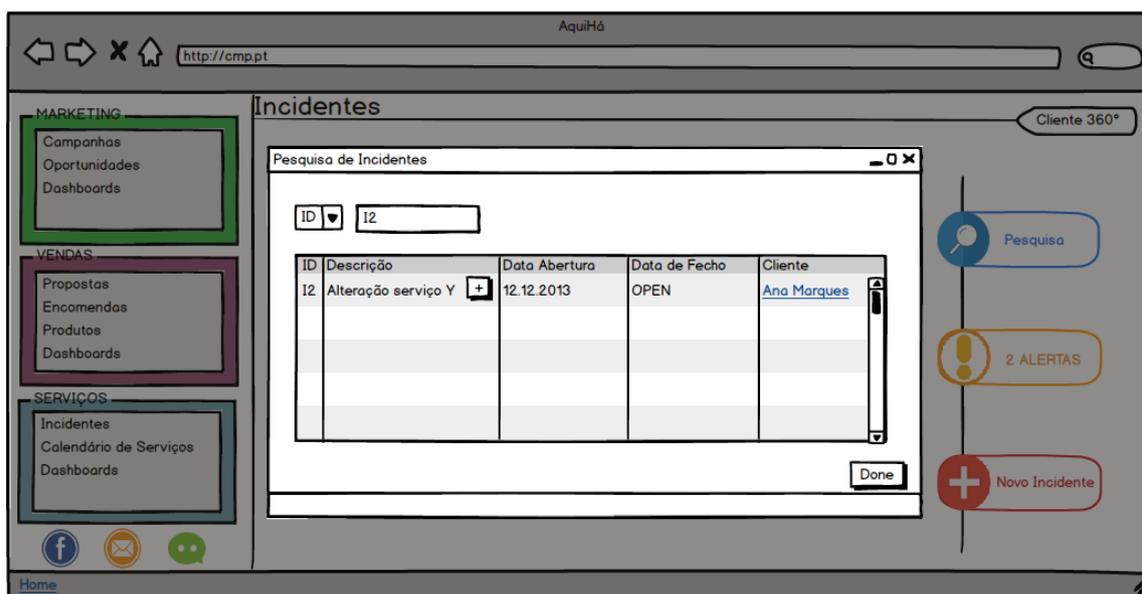


Figura 4.8– Protótipo para a página de pesquisa de incidentes

8) Alertas

Abrindo a opção de Alertas, o utilizador poderá visualizar a listagem de interações que tem a seu cargo por responder e pode optar por responder a alguma delas. Esta listagem corresponde a todas as interações que ele tenha por resolver, independentemente do canal.

Caso deseje consultar os detalhes referentes a cada uma delas, basta seleccioná-las e poderá, inclusivamente, dar-lhes resposta.

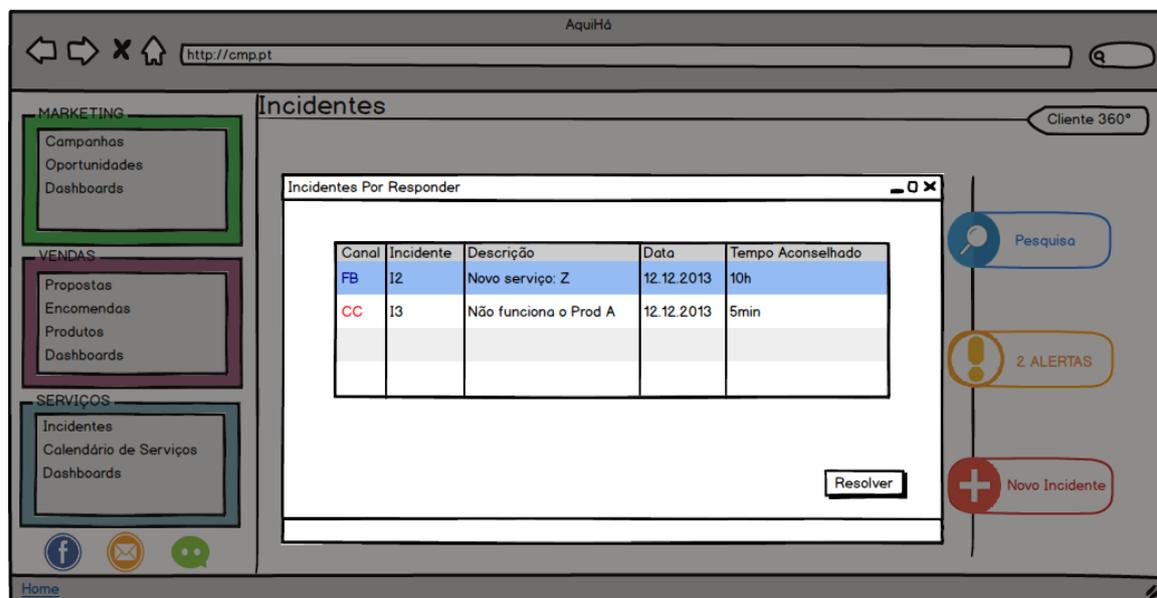


Figura 4.9– Protótipo para a página de alertas

4.2 Análise de Requisitos

Analisando, cuidadosamente, os protótipos desenvolvidos é agora possível detalhar quais as funcionalidades que se requerem ao sistema e analisá-las.

De acordo com *Sommerville* [18], divide-se esta análise em Requisitos Funcionais, correspondentes às funcionalidades que o sistema deve disponibilizar, bem como à forma que deve agir em certos contextos, e Requisitos Não Funcionais, correspondentes a características mais gerais do sistema, como restrições temporais.

4.2.1 Requisitos Funcionais

Na tabela seguinte apresentam-se os requisitos funcionais definidos e posteriormente é feita uma breve descrição de cada um deles.

ID	Requisitos Funcionais	Prioridade
Gestão de Incidentes		
1	Pesquisar Cliente e Visualizar Detalhes	
2	Criação Manual de Incidentes (Loja, <i>Call Center</i> , SMS)	
3	Criação Manual de Interações (Loja, <i>Call Center</i> , SMS)	
4	Pesquisar Incidente	
5	Consultar Listagem de Incidentes	
6	Consultar Detalhes de Incidente	Must Have
7	Consultar Detalhes de Interação	
8	Resposta a Interação Manual	
9	Associar Interação a Incidente existente	
10	Notificar Operador da Interação (caso seja automática)	
Integração com Redes Sociais		
13	Receber no CM+ Notificações do Facebook (Publicações, Comentários e Mensagens)	
14	Filtrar Incidentes de Outro Tipo de Publicações	
15	Criar Automaticamente Incidente e Interação	Must Have
16	Fazer Publicação no Facebook através do CM+ (Publicação, Comentário e Mensagem)	
17	Responder a Incidente via Facebook	
18	Criar Incidentes Via Twitter	
19	Responder a Incidentes via Twitter	Nice To Have
Sistema de Recomendação		
20	Apresentar Recomendações de Soluções para Incidente	
21	Consultar Detalhes de Incidente Recomendado	Must Have
22	Votar Recomendação	
23	Responder Automaticamente a Incidentes	Nice To Have

Tabela 4.1 - Requisitos Funcionais

Pesquisar Cliente e Visualizar Detalhes

O sistema original apresenta uma área designada de Cliente 360 onde é possível pesquisar e identificar clientes. No âmbito deste projeto deve ser implementada uma funcionalidade semelhante, reutilizando esse componente para que seja possível identificar os clientes, visualizar os seus detalhes e, quando necessário, criar novos incidentes relativos a esses mesmos clientes.

Criação Manual de Incidentes e Interações

O CM+, no estado de desenvolvimento inicial, possibilita a criação de incidentes através de Call Center, Loja ou SMS. No entanto, a metodologia atual de gestão de incidentes não ocorre de forma unificada e dado que esse tipo de tratamento envolve alteração ao nível de base de dados e de funcionalidades relativas aos serviços, essas funcionalidades terão de ser reformuladas para que sejam compatíveis com o novo sistema.

Assim, o sistema a desenvolver deve possibilitar ao operador criar quer novos incidentes, quer novas interações no contexto de um determinado incidente.

Pesquisar Incidente

O utilizador deve ter à disponibilidade uma forma de pesquisar incidentes. Isto é, dando o Id de cliente ou Id de incidente, o sistema deve devolver o (s) incidente (s) correspondente (s).

Este requisito é especialmente importante num contexto de *call center* ou loja em que o operador pode querer listar os incidentes de um determinado cliente ou criar uma interação no âmbito de um incidente específico.

Consultar Listagem de Incidentes, Interações e seus Detalhes

O sistema deve possibilitar ao operador listar os incidentes a seu cargo e caso pretenda, deve ser possível visualizar os detalhes de cada um deles. Nesses detalhes incluem-se dados como o id de incidentes, a data de início e fim (em caso disso) e também a listagem de interações ocorridas no seu âmbito.

Para essa listagem de interações deve ser também possível visualizar os detalhes de cada uma delas.

Resposta Manual a Interação

Para os casos de interações via manual (loja, SMS ou *call center*) o sistema deve possibilitar ao utilizador funcionalidades de resposta. Ou seja, dada uma certa interação, caso o operador pretenda dar-lhe resposta, deve poder aceder aos seus detalhes e editar e submeter a resposta, podendo definir se essa resposta encerra ou não o incidente em aberto.

Associar Interação a Incidente Existente

Como visto, um dos objetivos centrais do projeto é a unificação do processo de tratamento de incidentes. Como tal, devem ser desenvolvidas funcionalidades que permitam criar uma nova interação e associá-la a um incidente que esteja já registado e aberto no sistema.

Notificar Operador de Interação

Quando uma nova interação/incidente é criado através das redes sociais é necessário que haja um método que permita notificar o operador, para que ele lhe possa dar resposta. O operador deve ser alertado de qual a notificação, qual a origem e caso pretenda deve poder consultar os seus detalhes e dar-lhe resposta.

Receber no CM+ Notificações de Publicações do Facebook

Um dos objetivos, relativo à integração com o Facebook, é o de possibilitar a criação de incidentes no sistema a partir de declarações/denúncias de dúvidas ou problemas por parte dos clientes, na página de Facebook da empresa.

Para tal, para além de ser necessário criar um canal de comunicação com a rede social, é necessário que o sistema tenha a capacidade para ser notificado quando ocorrem novas publicações na página de Facebook da empresa para serem tratadas posteriormente.

Pretende-se que sejam recebidas notificações relativas a publicações, comentários e mensagens privadas.

Filtrar Incidentes de outro tipo de Publicações

Ao receber as notificações do *Facebook* é necessário que o sistema implemente um método de filtragem capaz de discernir:

- Se a notificação é proveniente de um cliente conhecido ou não;
- Se corresponde a um incidente (dúvida ou problema) ou se é simplesmente *spam* ou outro tipo de publicação;
- Se é referente a algum produto/Serviço em específico.

Criar Automaticamente Incidente e Interação

Quando o CM+ recebe a notificação proveniente do Facebook e caso confirme que se trata de um incidente e que é um cliente conhecido do sistema, deve ser criado e aberto um novo incidente no sistema, criada a primeira interação e feita a notificação ao operador anteriormente descrita.

Fazer Publicação no Facebook através do CM+

O canal de comunicação criado com o Facebook deve permitir que sejam feitas publicações e comentários na página da empresa e que se enviem também mensagens privadas. Estas funcionalidades são fundamentais para o requisito seguinte.

Responder a Incidente via Facebook

Quando um operador responde a uma dada interação no sistema, caso o canal seja o Facebook (caso se trate de uma resposta a uma dúvida/problema reportada na rede social), essa resposta deve ser submetida normalmente no sistema mas deve também ser enviada como comentário ou mensagem para o Facebook, em resposta à publicação do cliente.

Apresentar Recomendações de Soluções para Incidente

No contexto de um certo incidente o utilizador deve poder consultar as recomendações de soluções para esse mesmo incidente. O sistema deve devolver uma listagem de incidentes resolvidos que apresentem semelhança com o caso atual para que o operador veja qual a solução implementada nesses casos.

Consultar Detalhes de Incidente Recomendado

A listagem dos incidentes recomendados deve possibilitar que o utilizador selecione um deles e veja os seus detalhes.

Responder Automaticamente aos Incidentes

Esta funcionalidade aplica-se fundamentalmente aos canais: *Facebook*, *Email* e SMS. Seria interessante nestes casos, e baseando-se no algoritmo de recomendação de soluções anteriormente referido, o sistema gerar uma resposta automática (desde que houvesse uma solução com elevada probabilidade) de forma a reduzir o esforço humano e tornar este processo mais rápido.

4.2.2 Requisitos Não Funcionais

No que respeita ao comportamento não funcional do sistema consideram-se os requisitos estipulados na seguinte tabela.

Requisitos Não Funcionais	Prioridade
Informação Atualizada	
Escalabilidade	Must Have
Boa Performance de Integração	
Qualidade dos Resultados (Sistema de Recomendação)	
Usabilidade	Should Have

Tabela 4.2 - Requisitos Não Funcionais

Informação Atualizada

Visto que se trata de processos de comunicação entre a empresa e o cliente e que podem ser criadas várias instâncias de interação e por diferentes canais, é de extrema importância garantir que a informação seja atualizada quase em tempo real, de modo a que o operador não esteja a resolver problemas já superados ou quando já há novas informações.

Pretende-se que o sistema final mantenha um canal de comunicação coerente e credível e, como tal, atualizado e sincronizado com os vários canais possíveis.

Escalabilidade

O objetivo inicial estipulava que o sistema final deveria ter uma abrangência maior e uma capacidade de adaptação transversal. Estas capacidades referem-se não somente ao tipo de sector a que pertence a empresa mas também à sua dimensão. O produto desenvolvido deve apresentar performance semelhante seja uma empresa de venda a retalho com milhares de clientes, ou seja uma operadora de telecomunicações com milhões de clientes.

É importante ter este factor em conta durante o processo de desenvolvimento para que o alcance do objetivo central seja verdadeiramente cumprido.

Qualidade dos Resultados Obtidos

Um dos requisitos funcionais é a implementação de um sistema de recomendação de soluções de incidentes. Mais que gerar sugestões, é fundamental que essas sugestões tenham significado, sendo relevantes para o operador e contribuindo assim para a minimização do tempo de resolução de problemas.

Como tal, um requisito que deve ser alcançado é a obtenção de resultados de qualidade e com relevância na parte de recomendação de incidentes.

Usabilidade

Uma das vantagens apresentada como muito apelativa no que respeita a estes *softwares* é a sua usabilidade. Viu-se que produtos como o *Siebel* apresentavam *interfaces* complexos e pouco intuitivos para o utilizador. Contrariamente, outros, como o *Dynamics CRM*, apresentavam ambientes fáceis de usar, o que consequentemente os tornava mais apelativos.

Assim, um dos requisitos não funcionais a ter em conta é o desenvolvimento de um sistema fácil e intuitivo.

Capítulo 5

Arquitetura e Especificação

Depois de feita a análise de requisitos do sistema a desenvolver é agora imperatório que se defina qual a estrutura que vai estar na base desse sistema e como cada módulo da arquitetura vai agir e interagir de forma a implementar as funcionalidades requeridas.

Este capítulo apresenta a proposta de resolução para a implementação dos requisitos definidos. Para se atingirem os objetivos propostos é necessário fazer alterações à base de dados atual e a algumas funcionalidades lógicas já existentes. É também necessária a integração de novos módulos lógicos. Nesta análise vamos esquematizar a arquitetura final pensada para o sistema bem como perceber como se vai estruturar cada um dos módulos que será necessário alterar ou introduzir e como cada um deles será desenvolvido.

5.1 Estrutura Geral do Sistema

O sistema final estrutura-se em 3 camadas principais. A primeira camada, o **Portal**, corresponde ao sistema de *frontend* que estabelece a comunicação com o utilizador final e que encaminha os seus pedidos para a camada intermédia, a camada **lógica**, que compreende o **CM+** e os módulos de **integração com o Facebook** e de **recomendação**. Por fim, a última camada corresponde à **Base de Dados** com toda a informação requerida.

Ao nível do portal são desenvolvidas apenas as funcionalidades de *frontend* correspondentes aos requisitos estipulados.

Como referido, a camada lógica engloba 3 componentes: o **CM+**, o módulo de **integração com o Facebook** e o módulo de **recomendação de incidentes**. Ao nível do **CM+** são implementadas alterações na camada de negócio (camada lógica do **CM+**). Estas alterações correspondem à introdução de funcionalidades de gestão de incidentes.

Quanto à base de dados, será utilizada a que atualmente está associada ao **CM+**, sendo apenas ligeiramente modificada de forma a dar resposta às necessidades das novas funcionalidades.

A figura seguinte esquematiza a arquitetura que se pretende implementar.

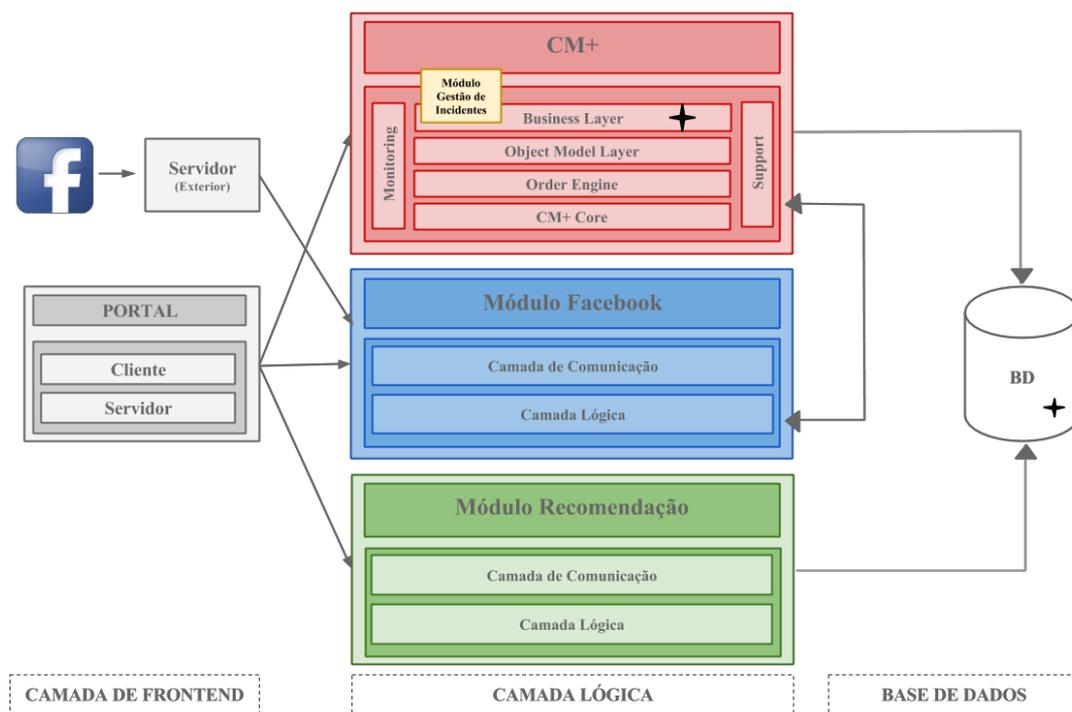


Figura 5.1– Arquitetura proposta para o sistema final

O **Portal** compõe-se pelas componentes Cliente e Servidor sendo que, na camada do servidor se podem encontrar as funcionalidades requeridas para a comunicação entre o sistema de *frontend* e os serviços de *backend*.

No que diz respeito à **Lógica**, individualizam-se 3 componentes. O primeiro, o CM+, constitui-se pelo sistema base existente, ao qual é acrescentado um módulo de gestão de incidentes na camada de negócio. Este engloba todos os processos de tratamento de incidentes, independentemente do canal de origem. Para além disso, ao nível do CM+ são ainda acrescentados métodos que permitem a comunicação com os restantes elementos da camada lógica e o tratamento dos dados que deles provêm.

O segundo componente, o módulo de integração com o Facebook implementa as funcionalidades necessárias para a comunicação com o Facebook, bem como os processos de filtragem das notificações recebidas, o seu processamento e os métodos que despoletam a criação de novos incidentes ou interações quando necessário.

O módulo de recomendação corresponde à implementação da lógica necessária quer para a construção da base de casos (incidentes), quer para o cálculo da semelhança entre casos e conseqüente *ranking* de recomendações.

Na Figura 5.1 pode observar-se um asterisco na base de dados associada ao CM+. Este simboliza a necessidade de efetuar algumas alterações à estrutura atual da base de dados. Essas alterações envolvem por um lado a criação de novos campos em certas entidades mas também a criação de novas entidades. As alterações e os novos elementos serão detalhados seguidamente.

5.1 Base e Modelo de Dados

Como referido anteriormente, um dos pontos fortes do CM+, que o valoriza enquanto produto de CRM, é a base e o modelo de dados que lhe estão associados. Estes apresentam

uma estrutura sólida e capaz de mapear de forma completa os conceitos e necessidades associadas a um sistema de CRM. Como tal, não se encontra necessidade de introduzir muitas alterações para a implementação dos requisitos definidos.

Associada ao CM+ encontra-se, atualmente, uma base de dados Oracle SQL com cerca de 50 tabelas, cada uma delas com um número de campos variável, sendo que a tabela com menor número de campos apresenta 7 e a com maior número, *CMP_ACCT*, apresenta 70 campos.

As tabelas constituintes da base de dados são mapeadas pelas entidades definidas ao nível do modelo de dados do sistema, implementado por via da *framework* Oracle ADF. É também ao nível dessa *framework* (camada de modelo de dados) que são definidas as relações existentes entre entidades.

Face aos requisitos estipulados, e tendo em conta a estrutura do sistema atual, verifica-se a necessidade da existência de duas novas entidades: *Incidente* e *Operador*. O *Incidente* pretende representar cada conjunto de interações ocorridas com um cliente, no âmbito de um mesmo problema. Tem sempre de corresponder a um determinado cliente e conta e tem associado um número variável de interações (que podem ter proveniência de canais distintos).

A entidade *Operador*, por sua vez, visa identificar cada utilizador do sistema. Para o contexto das funcionalidades a desenvolver no âmbito deste projeto, esta variável apresenta relevância no que toca a distribuir a responsabilidade pelas interações. Isto é, a cada nova interação criada deve ser associado um operador que terá a seu cargo a sua análise e respetiva resposta.

Relativamente às entidades atuais verifica-se que devem ser alteradas as entidades *Cliente* e *Interação*. A tabela de *Cliente* tem associada 38 campos entre os quais se encontram dados como nome, endereço e outros dados pessoais. A estes devem ser acrescentadas as referências para as contas pessoais do cliente no *Facebook* e/ou *Twitter*, de forma a tornar a comunicação com o cliente por estas vias possível. Estes dois novos campos são opcionais.

Ao nível da *Interação* é necessário que o campo “canal” possa assumir os valores: *Facebook* e *Twitter*, para além do SMS, *call center* e loja, que já assume. Para além disso, deve possibilitar-se dois tipos de descrição (atualmente existe apenas um campo descrição), descrição do cliente (correspondente ao problema exposto pelo cliente) e a descrição do operador (com a proposta de resolução ou esclarecimento).

O processo de criação das novas entidades e de implementação das referidas alterações envolve dois passos: primeiro a criação das tabelas e campos na base de dados e segundo, o seu mapeamento na camada de modelo de dados ADF. Na segunda fase inclui-se também a definição das relações entre as novas entidades criadas. Como resultado deste processo obtém-se um modelo de dados semelhante ao esquematizado na figura 5.2.

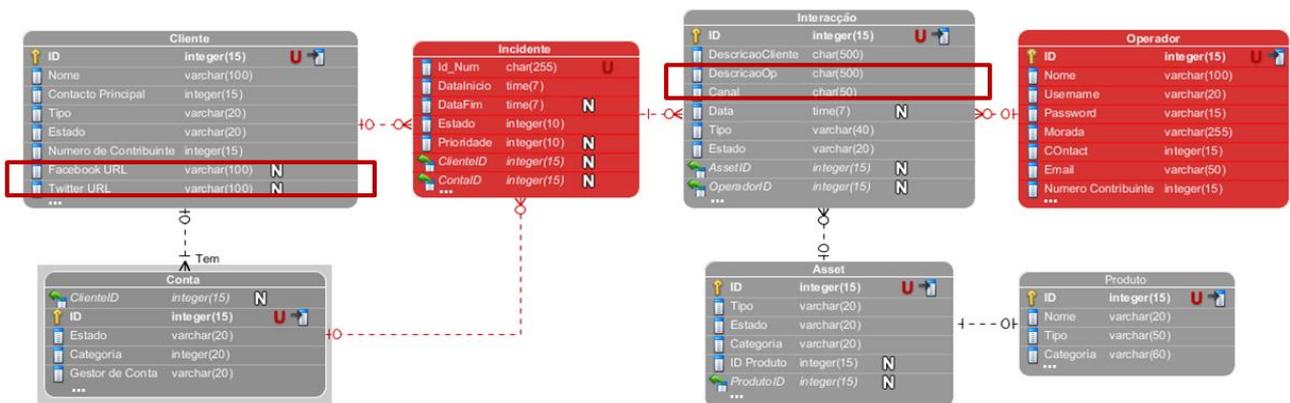


Figura 5.2– Representação do Modelo de Entidades

A vermelho assinalam-se os novos campos, novas entidades e novas relações criadas para o desenvolvimento das funcionalidades requeridas no âmbito deste projeto.

As entidades *Cliente*, *Conta*, *Produto*, *Asset* e *Interação* encontram-se associadas a outras entidades do sistema. No entanto, visto que neste contexto essas entidades são secundárias, optou-se por não as expor aqui, apresentando apenas aquelas que realmente são relevantes para o projeto.

Como é visível, não há necessidade de alteração estrutural muito significativa o que revela que o modelo inicial é de facto, bastante completo.

5.2 Lógica

É ao nível da camada lógica que é requerido o maior esforço para a implementação das funcionalidades estipuladas na análise de requisitos. O trabalho desenvolvido nesta camada subdivide-se em 3 grupos, correspondendo cada um aos módulos apresentados na figura 5.1, isto é, CM+, integração com o Facebook e módulo de recomendação.

No que respeita ao CM+ para além de algumas alterações aos processos de gestão de serviços pré-existentes, é necessário acrescentar novas funcionalidades que permitam quer a gestão de incidentes unificada, quer a integração com os dois novos módulos.

Os dois restantes elementos da camada lógica serão construídos de raiz. Em seguida, analisa-se em detalhe a estrutura de cada um destes 3 componentes lógicos.

5.2.1 CM+

A implementação da lógica de funcionalidades de gestão de incidentes que permitam um tratamento simples e unificado de serviços independentemente da sua origem vai ocorrer através de alterações e criações de métodos alojados na camada de negócio.

Consideram-se duas fases, a alteração/adaptação dos processos de gestão de serviços existentes e a criação de novos métodos que permitam a comunicação com o módulo de integração com o Facebook que, conseqüentemente, permitem a criação de incidentes e interações de forma automática

Relativamente aos processos já implementados serão alterados os métodos que atualmente correspondem à criação de interações via *call center* e loja. Estes terão de ser adaptados por um lado porque, como visto, o conceito da entidade interação foi modificado e, por outro, porque a sua metodologia de funcionamento tem de se adaptar ao novo contexto em que os incidentes são tratados de forma unificada.

O segundo módulo, correspondente à criação de uma nova metodologia de tratamento de incidentes envolve o desenvolvimento de funcionalidades que permitam a criação de interações e que, independentemente do canal de proveniência, as mapeie para os incidentes a que correspondem. Para além disso, aqui serão desenvolvidos métodos de consulta de detalhes, responsabilização de operadores, notificação de operadores das tarefas a seu cargo, resposta a interações multicanal, entre outras.

5.2.2 Módulo de Integração com o Facebook

Numa fase inicial pensou-se na integração com o Facebook como parte integrante da lógica do CM+. Dessa forma, os métodos necessários seriam implementados diretamente no sistema existente e o desenvolvimento seria feito à medida em função do perfil e das necessidades específicas do CM+.

No entanto, com uma análise mais detalhada foi-se percebendo que o desenvolvimento de um módulo independente do sistema, agregando a lógica de comunicação com o Facebook, poderia ser mais vantajoso.

Na perspectiva do CM+ enquanto produto, a modularidade permite que o cliente final escolha especificamente quais as funcionalidades que quer adquirir e torna esse processo mais simples para a equipa de desenvolvimento, que tem apenas de incluir ou não o módulo em função dos requisitos.

Por outro lado, e dado o contexto deste trabalho, é importante pensar como o resultado final pode ser utilizado e reutilizado, pensando numa perspectiva de otimização do trabalho desenvolvido. Ao se desenvolver um componente individualizável, passível de adaptação a qualquer sistema e apresentando funcionalidades mais globais, aumenta-se a probabilidade de reutilização do componente e simultaneamente a sua versatilidade.

Tendo em conta estas vantagens, a arquitetura para o módulo de integração com o Facebook foi desenhada e pode observar-se na figura seguinte. Também a forma como comunica com sistemas externos, neste caso, com o CM+ pode ser analisada na figura.

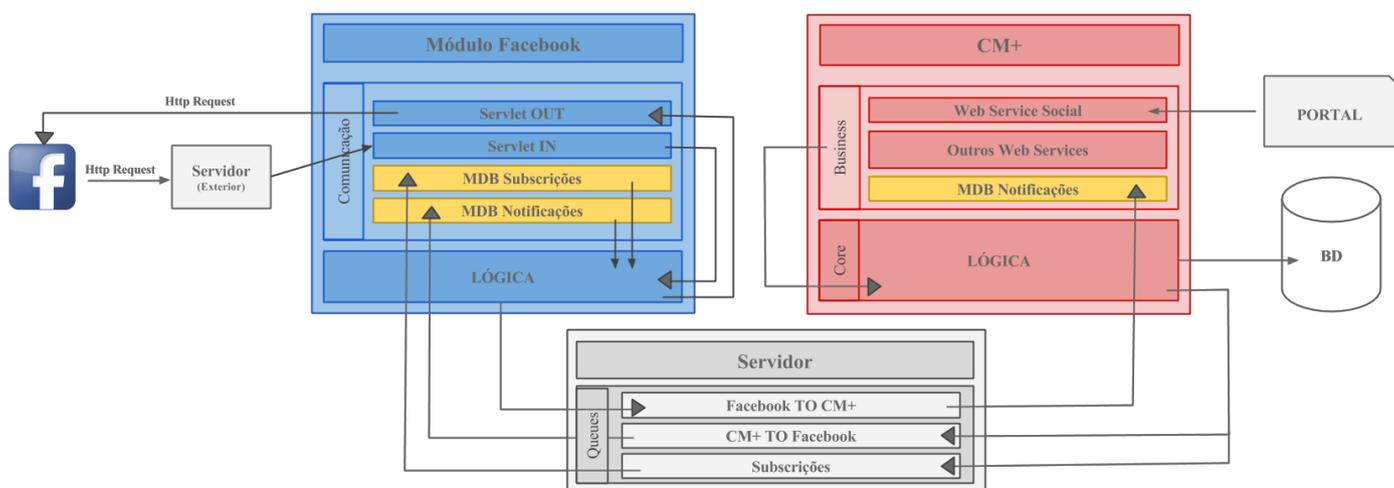


Figura 5.3– Arquitetura do Módulo de Integração com o Facebook

A comunicação entre o CM+ e o Facebook faz-se, como veremos detalhadamente no capítulo seguinte, através de pedidos Http. Para que seja possível receber notificações quando são feitas publicações, comentários ou enviadas mensagens, é necessário ter à disposição um endereço público para o qual o Facebook enviará também pedidos Http.

Recebidos no servidor, esses pedidos são reencaminhados para o módulo de integração, a azul na figura. Este subdivide-se em duas camadas. A primeira é a camada lógica e engloba os métodos responsáveis pelo processamento da informação que circula nos sentidos: Facebook – Sistema e vice-versa. A segunda camada corresponde aos

processos envolvidos na comunicação com sistemas externos, tais como *Facebook*, servidor e, neste caso, o sistema de CRM.

Para a comunicação com o CM+ foram analisadas duas possibilidades, a utilização de **Web Services** ou **MOM** (Message Oriented Middleware).

Arquitetura do Módulo

Ambas as possibilidades teriam aplicabilidade no sistema em desenvolvimento. Como tal, é necessário fazer uma pequena ponderação sobre as vantagens e desvantagens de cada uma delas.

Uma arquitetura com utilização de *Web Services* apresenta uma maior semelhança com o que se encontra implementado atualmente no CM+. Para além disso, expor este módulo através de *Web Services* contribuiria para uma comunicação mais rápida. No entanto, a interoperabilidade permitida por este tipo de arquiteturas é baixa e exige comunicação síncrona, desvantagens que têm de ser tidas em conta.

Por sua vez, sistemas baseados em MOM apresentam-se à partida como sistemas mais complexos e difíceis de implementar, mas com vantagens acrescidas no que toca a fiabilidade, escalabilidade e metodologia de comunicação.

Através de MOM é possível que um sistema envie uma mensagem para outro de forma assíncrona, ou seja, enviada a mensagem, o primeiro sistema pode prosseguir sem ter de esperar que o destinatário da mensagem a processe e responda. Este aspecto é especialmente vantajoso quando há grande volume de troca de mensagens, o seu processamento pode ser demorado e nos quais a resposta do outro sistema não é relevante para a continuidade do sistema.

O módulo de integração com o *Facebook* funciona como um receptor de notificações, seu processamento e conseqüente notificação do sistema externo (CM+). Para além do número de notificações ser variável e possivelmente elevado, a notificação do CM+ não envolve a necessidade de resposta (do CM+ ao módulo).

No sentido contrário de comunicação, o CM+ envia para o módulo pedidos para fazer publicações, comentários ou enviar mensagens, não exigindo esses processos resposta para que se possa prosseguir no fluxo do sistema.

Por último, o módulo em questão está vulnerável às condições da rede e dependente dos servidores externos do Facebook e da sua rapidez a processar pedidos. Este facto pode por vezes gerar períodos de latência ou falhas, que com uma arquitetura MOM não correspondem a perda dos pedidos nem à latência do sistema principal (CM+).

Assim, ponderando as características de cada uma, opta-se neste módulo pelo recurso a MOM, com a utilização de **queues FIFO** como plataforma de armazenamento das mensagens enviadas e com um modelo de envio *point-to-point*.

Ao nível da camada lógica são implementados os processos de processamento das notificações que circulam entre o sistema e o Facebook (em ambos os sentidos). Mais pormenores desta camada serão vistos no capítulo seguinte.

5.2.3 Módulo de Recomendação de Soluções

Um dos objetivos que se pretende que um CRM alcance é o de agilizar os processos de interação com o cliente, permitindo uma resposta mais rápida e eficaz às interações ocorridas. Através da implementação de um sistema de recomendação de soluções pretende-se construir o caminho para o alcance desse objetivo.

À semelhança do que acontece com o módulo anteriormente analisado, também este é realizado de forma independente do CM+. A arquitetura segundo a qual é implementado pode observar-se na figura seguinte.

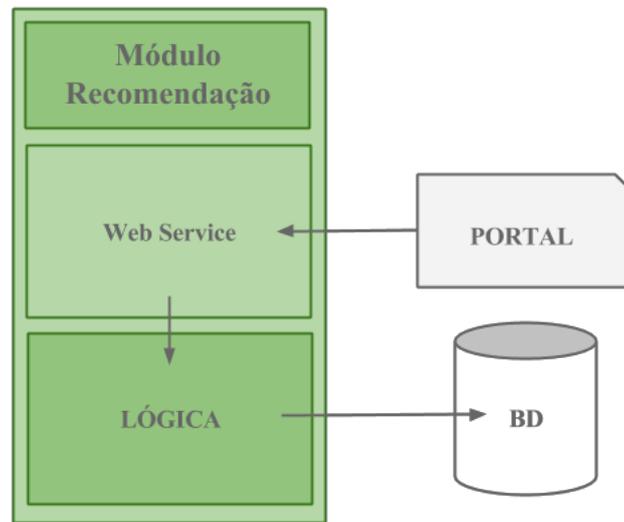


Figura 5.4– Arquitetura do Módulo Recomendação

A sua estrutura é significativamente mais simples do que a apresentada para o módulo de integração com o Facebook. Neste caso, o módulo comunica diretamente com a base de dados. Esta base de dados não tem de ser necessariamente a base de dados associada ao CM+, mas sim uma qualquer base de dados que contenha os casos que representam os incidentes.

O portal pode aceder diretamente às funcionalidades disponibilizadas por este módulo invocando métodos expostos via *Web Services*. Se no caso anterior se considerou que optar por MOM era a melhor solução, neste opta-se por uma arquitetura em que se expõem os serviços disponibilizados via *Web Services (JAX-WS)*. Foi tomada esta opção visto que as funcionalidades disponibilizadas exigem uma resposta, dependendo o fluxo do sistema dessa mesma resposta.

Na camada lógica do módulo pode encontrar-se a implementação do sistema de recomendação. No capítulo seguinte essa implementação será detalhada, mas de forma geral pode dizer que a camada lógica agrega dois grupos de processos: um responsável pelo **pré-processamento** dos casos e o outro que agrega funcionalidades de **motor de busca** (indexação de casos e cálculo de semelhança).

Metodologia

A metodologia a usar na implementação desse sistema será aprofundada no capítulo seguinte. No entanto, para que se compreenda em traços gerais o que será criado ao nível da camada lógica, explica-se brevemente em que consiste este sistema.

Introduzindo como *input* um conjunto de dados estruturados correspondentes a um produto ou serviço, espera-se que um sistema de recomendação gere recomendações com significado que possam interessar aos utilizadores finais. Extrapolando para o contexto deste projeto, o objetivo é que, partindo do conjunto de incidentes ocorridos anteriormente, e

comparando com um incidente em específico em aberto, se gerem recomendações de soluções possíveis (25).

A recomendação pode basear-se em atributos de perfil de itens (incidentes) e nesse caso é chamada de recomendação baseada no conteúdo ou pode prever resultados baseando-se em votações de utilizadores, caso em que se denomina de recomendação colaborativa. Existe ainda um outro modelo de recomendação, a recomendação híbrida, que combina as técnicas das duas anteriores (26).

Para o sistema em questão considera-se vantajosa a implementação do modelo híbrido. O objetivo é primeiramente desenvolver um método capaz de detetar semelhança entre perfis de incidentes (recomendação baseada no conteúdo) e numa segunda fase melhorar a capacidade preditiva desse sistema com o modelo colaborativo. Para a parte colaborativa coloca-se à disponibilidade do operador a possibilidade de votar na qualidade da solução recomendada.

O sistema híbrido descrito inclui-se numa estratégia em cascata. Neste tipo de estratégia são usados vários métodos de recomendação por uma certa ordem, sendo que cada método apenas atua sobre os resultados do método anterior. No caso concreto deste projeto a ideia é ter 2 métodos de recomendação. O primeiro baseia-se no conteúdo e gera um conjunto de x elementos. Posteriormente, esses elementos serão analisados de acordo com a sua votação (colaborativo) e serão selecionados $x-y$ elementos, apresentados por ordem decrescente de votos.

No fundo, através da adoção deste método pretende-se refinar os resultados obtidos, procurando que eles se aproximem daquilo que são as necessidades do utilizador final (27).

5.3 Portal

No exemplo prático analisado, no capítulo 2, foi visto que o sistema que fazia o *frontend* de interação com o utilizador final era o chamado “Portal de Atendimento”, implementado com recurso a um servidor *Weblogic* da *Oracle*, desenvolvido em *javascript*.

Para que seja visível o cumprimento dos objetivos definidos no âmbito deste projeto é necessário desenvolver um componente conceptualmente semelhante a esse portal de atendimento que permita servir de *interface* entre o utilizador e os serviços desenvolvidos.

Dois factores de realce num *software* de CRM são a sua usabilidade e o seu custo e ao nível do desenvolvimento do portal são é importante não deixar de os ter em conta ao longo do processo de desenvolvimento.

A prototipagem de cenários realizada permite ter uma breve noção do que se pretende conceptualmente ao nível das funcionalidades que devem ser disponibilizadas e modo de organização das mesmas.

De realçar que este portal funciona maioritariamente como protótipo para o mapeamento das funcionalidades lógicas implementadas, não mapeando todo o funcionamento do CM+.

5.4 Estrutura e Tecnologias

As ferramentas utilizadas na empresa para o desenvolvimento do CM+ são:

- Ambiente de Desenvolvimento Java (**Eclipse**, Netbeans, etc.);
- **Maven**, para a gestão das dependências de bibliotecas através de um ficheiro *pom*;
- **Oracle JDeveloper**, ferramenta para a gestão e alteração do modelo de dados ADF;
- **SOAP UI**, para a realização de testes de validação;
- **Oracle SQL Developer**, para gestão da Base de Dados;

O presente projeto parte do trabalho já realizado no âmbito do CM+ e intenta dar-lhe continuidade ao mesmo tempo que o mantém coerente com o já desenvolvido. Assim, opta-se por manter as tecnologias atualmente em utilização.

Como servidor *web* considera-se a utilização do *Weblogic Server da Oracle* e ao nível do portal recorre-se a HTML e *javascript*.

O CM+ encontra-se desenvolvido em *java* e, para implementação das funcionalidades atuais, implica a importação de algumas bibliotecas, tais como:

- **log4j** (para fazer log de dados, ajudando no *debug*);
- **quartz** (ferramenta que possibilita escalonamento de tarefas e como tal útil para a definição de *workflows* e para a gestão de sistemas);
- **hibernate** (para facilitar o mapeamento entre o modelo de dados definido na base de dados e o modelo de objetos);
- **Ehcache** (*cache java-based*);
- **Spring** (*framework* que funciona como *container* de inversão de controlo e injeção de dependência).

Em termos de estrutura *java*, Os recursos estão estruturados nos seguintes projetos Maven:

- **cmp_common**: Não possui código, simplesmente agrupa dependências Maven comuns;
- **cmp_core**: Agrupa todas as funcionalidades core da plataforma CM+, correspondendo à implementação da camada Core;
- **cmp_eai**: Contém as funcionalidades que são utilizadas para a interação com a camada de integração (EAI), inclui os métodos e API's necessários para enviar/receber mensagens http, JMS, etc.;
- **cmp_services**: Inclui os recursos que implementam lógica de negócio e serviços técnicos expostos por via JAX-WS, ou seja, em termos de arquitetura corresponde aos processos que ocorrem ao nível da camada de negócio;
- **cmp_parent**: Agrupa todos os projetos e permite levar a cabo operações Maven sobre todos eles;
- **cmp_weblogic**: ficheiros XML's e descriptors que serão usados pela weblogic;

Capítulo 6

Implementação

Ao longo deste capítulo pretende descrever-se todo o processo de implementação dos requisitos previamente estipulados.

O referido processo é desenvolvido de forma iterativa e iniciou-se pela implementação das alterações à base de dados existente. Na figura seguinte podem observar-se as várias fases do processo de desenvolvimento. No anexo 1 encontra-se um planeamento de PERT que apresenta os diferentes processos implementados em cada uma das etapas.

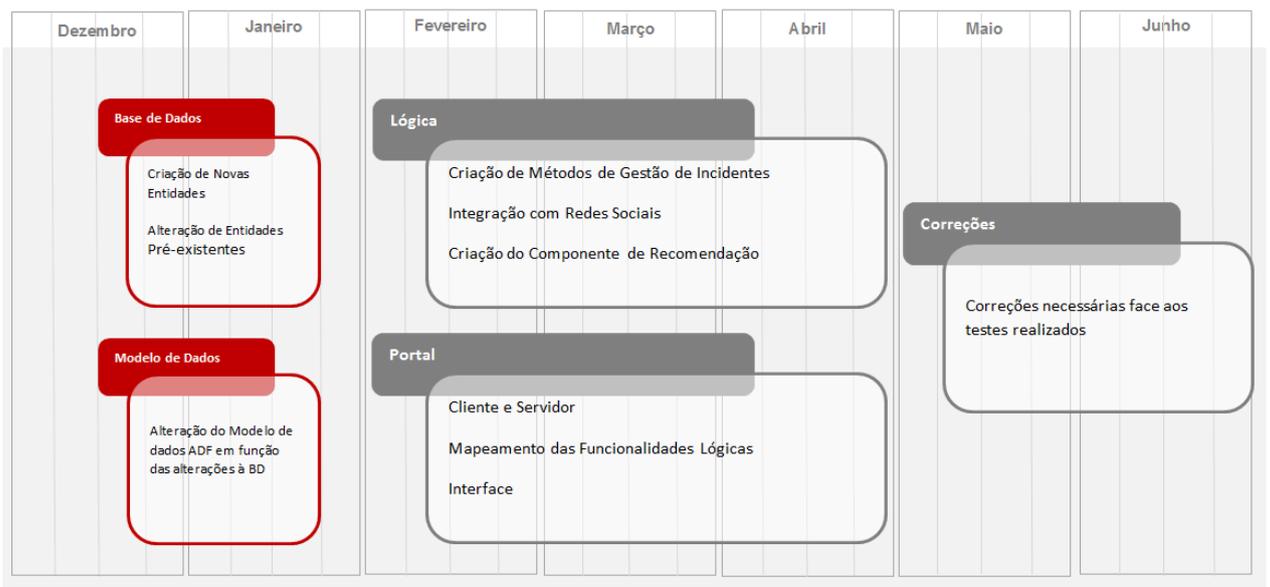


Figura 6.1– Fases do Processo de Implementação

6.1 Base e Modelo de Dados

Como analisado no capítulo anterior, o desenvolvimento das novas funcionalidades implica algumas alterações ao modelo de dados atual. Para tal é necessária a criação de novas tabelas na base de dados, a introdução de novos campos em algumas e, posteriormente, o mapeamento destas alterações ao nível do modelo de dados *ADF*.

Aceder e modificar o modelo de dados é um processo que pode ser realizado através do *software JDeveloper*. Na figura 6.2 encontra-se um *screenshot* com a estrutura típica de um modelo de dados desenvolvido através da *ADF*.

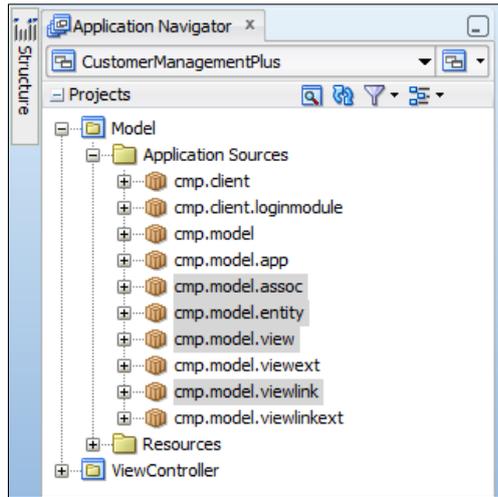


Figura 6.2– Screenshot da estrutura do modelo ADF

Sombreado a cinzento encontram-se os objetos que é necessário modificar para mapear as alterações feitas à base de dados e também para estabelecer as relações entre as entidades pretendidas.

No *package cmp.model.entity* são criados os *entity objects*, que vão mapear, neste caso, as tabelas da base de dados (noutros contextos poderiam mapear outros componentes da base de dados, ficheiros XML, etc.).

Na figura seguinte apresenta-se o mapeamento da tabela *CMP_INCIDENT*, criada para representar incidentes. Os campos da tabela são representados como atributos do objeto e é possível configurar os seus detalhes.

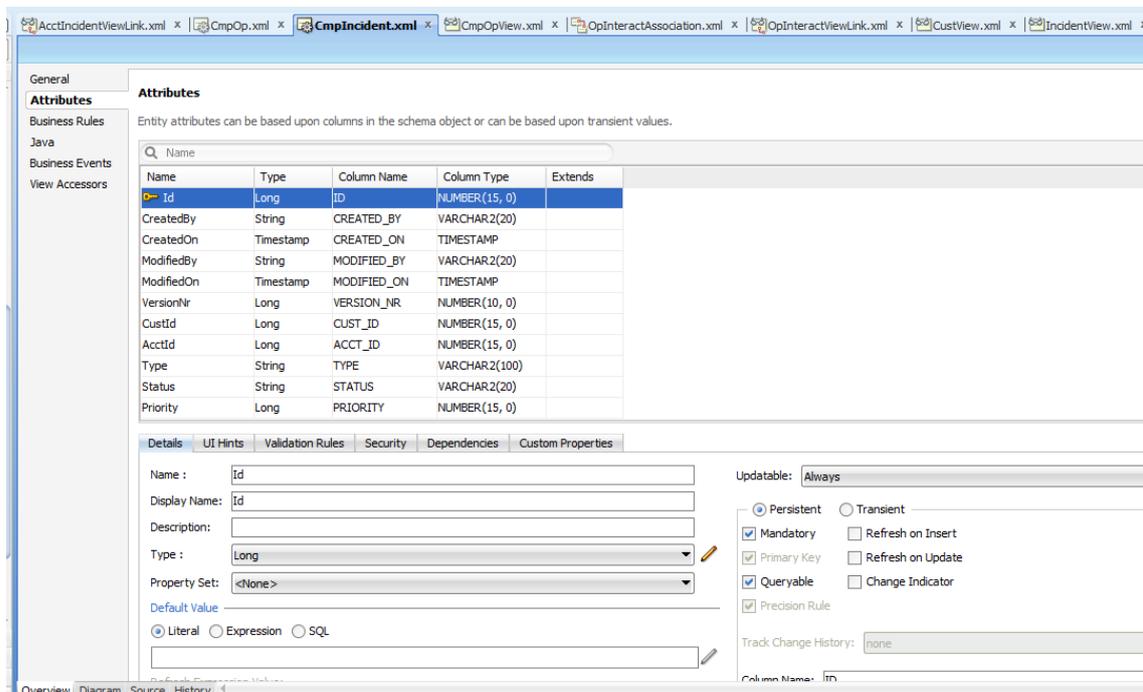


Figura 6.3– Screenshot do Detalhe da Entidade CmpIncident

A criação de *View Objects* é também um processo necessário ao nível do modelo de dados. Um objeto deste tipo permite fazer *queries* à base de dados, extraíndo dados de uma dada tabela. Assim, cada *view object* tem associado um conjunto de entidades à qual faz *queries* para recolher dados necessários.

Criadas as entidades e as *views* é necessário definir ligações entre elas. Para relacionar entidades criam-se *Associations* e, baseando-se nelas, é possível definir *View Links* que expressam a relação entre *views* associadas a *entities*.

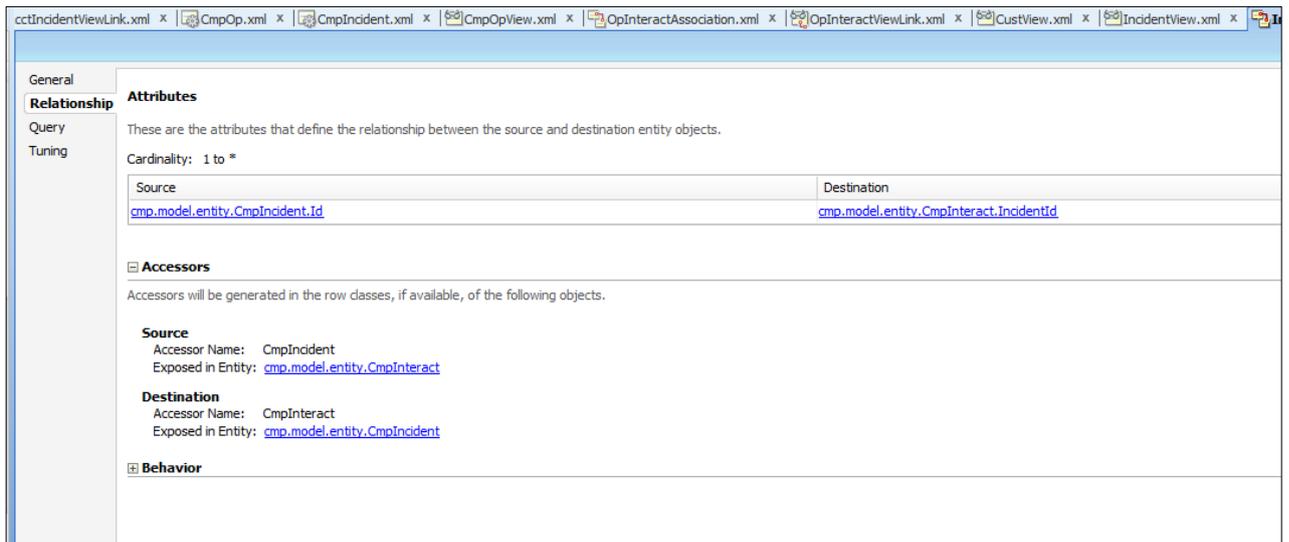


Figura 6.4– Screenshot do Detalhe da Vista CmpIncident

A figura 6.4 apresenta um exemplo de uma *Association* que relaciona a entidade *CmpIncident* com a *CmpInteract*. Como atributos de relação seleciona-se o Id de incidente, com uma cardinalidade de “1-*”, visto que um incidente pode ter várias interações associadas.

Com base nesta *Association* pode criar-se então o *View Link* para expressar a relação entre as entidades.

Para todas as entidades criadas e modificadas e respetivas relações é necessário executar estas alterações.

6.2 CM+

6.2.1 Gestão de Incidentes

Aplicadas as alterações necessárias ao nível da base e do modelo de dados, estão criadas as condições necessárias para tornar possível a implementação do elemento de gestão de incidentes no CM+.

De forma a clarificar o fluxo de funcionamento que se pretende alcançar com esta fase de implementação, apresenta-se o seguinte diagrama.

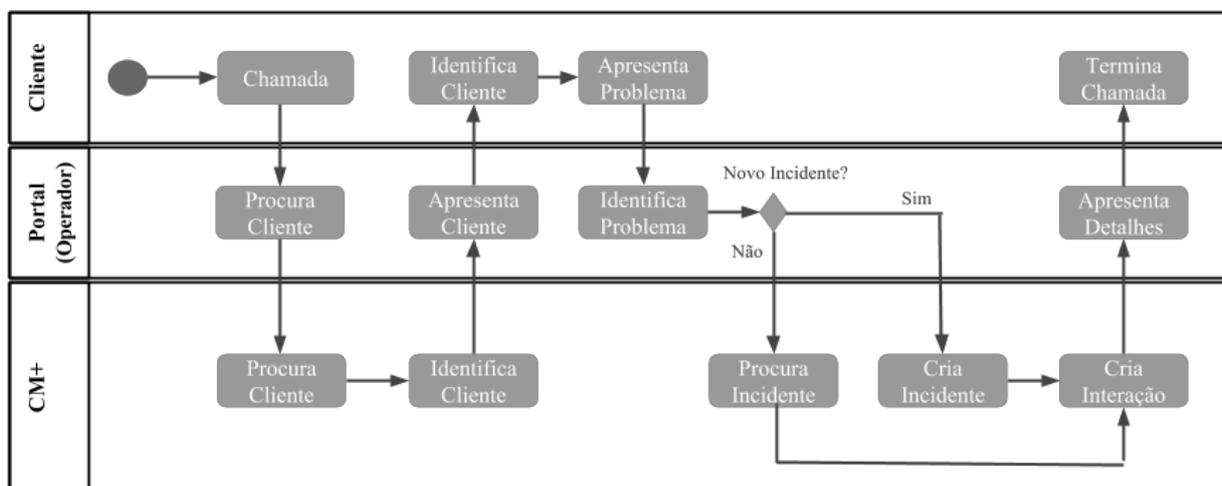


Figura 6.5– Diagrama de Atividades para a “Criação de Novo Incidente Manual”

Este diagrama representa a **criação de um novo incidente via Call Center**. A atividade é despoletada por uma chamada para os serviços da empresa que, ao nível do operador/portal, leva à identificação do cliente. Assim que o cliente apresenta a sua dúvida ou problema o operador pode decidir se se trata de um novo incidente ou se a interação corresponde a um incidente em aberto.

Tomada essa decisão pode ser submetida a criação do incidente ou interação, respetivamente, sendo que o sistema vai encaminhar essa informação para o CM+ que por sua vez a vai validar e submeter para a base de dados.

Criado o incidente e respetiva interação, são apresentados no portal os seus detalhes e o operador pode terminar a chamada.

Para tornar possível este fluxo divide-se a implementação em 3 fases:

1. **Comunicação com a Base de Dados:** Métodos necessários para fazer alterações à base de dados que permitam aceder, alterar e inserir dados nos novos campos e entidades criados.
2. **Métodos Lógicos:** Desenvolvimento de funcionalidades responsáveis pelo processamento de informação e validação de dados para posterior inserção na base de dados. (Nesta fase são também realizadas as alterações necessárias aos métodos pré-existentes do CM+ para ser possível a gestão de serviços tal como descrita na análise de requisitos.)
3. **Criação de Web Services:** Desenvolvimento dos *Web Services* através dos quais os novos serviços ficam expostos aos sistemas exteriores. É através destes componentes que se torna possível que sistemas externos, como o Portal, acessem às funcionalidades, invoquem serviços e que esses sejam validados e submetidos na base de dados.

6.2.2 Integração com o Módulo Facebook

Para além da implementação dos processos que permitem a gestão de incidentes, é necessário incluir no CM+ funcionalidades que permitam a comunicação e o enquadramento com a lógica implementada no módulo de integração com o Facebook.

A comunicação entre os dois elementos (CM+ e módulo) vai ser feita através do envio de mensagens no formato *xml* para *queues FIFO* no servidor *Weblogic*. Como tal, são incluídos no CM+ métodos que permitem quer a leitura quer a escrita para essas mesmas *queues*.

No servidor estão alojadas duas *queues* para as quais o CM+ pode enviar mensagens, uma para onde deve enviar os pedidos de subscrições (subscrições à integração com o Facebook e aos serviços específicos a que subscrive) e outra para onde devem ser enviados todos os pedidos relacionados com a gestão da página da empresa (fazer publicações e comentários, enviar mensagens, consultar publicações, *likes*, etc.).

As mensagens enviadas para estas *queues* devem estar no formato *xml* de acordo com um esquema previamente definido, colocando-as numa estrutura perceptível ao módulo do *Facebook*.

Existe ainda uma terceira *queue* no servidor da qual o CM+ recebe mensagens. É criado um *listener* que espera os pedidos dessa *queue* e que, quando os recebe, os trata e encaminha para as funcionalidades do CM+ que lhe correspondem.

No caso prático em que um cliente faz uma publicação na página da empresa e ela é identificada como incidente, ela é enviada para a *queue* do CM+. O *listener* à escuta dessa *queue* vai identificar a recepção de uma notificação e vai processá-la, convertendo a mensagem *xml* num objeto *java* familiar ao CM+. O objeto é analisado e como se trata de um incidente vai ser encaminhado para o componente de gestão de incidentes que o processará e o criará na base de dados.

Processamento das Notificações Provenientes do Facebook

Tratando-se de uma página de Facebook pública que permite publicações e comentários de todos os que lhe acedam é necessário implementar um filtro que permita ao CM+ distinguir entre o conteúdo que realmente lhe interessa. No âmbito deste projeto, apenas se pretende filtrar publicações ou mensagens que correspondam a incidentes (dúvidas ou problemas) de clientes atuais da empresa. No futuro estas funcionalidades podem ser estendidas de forma a mapear outras funcionalidades como por exemplo identificar situações de satisfação ou insatisfação dos clientes.

Para o âmbito deste trabalho, o que se pretende é que, ao receber uma notificação do *Facebook*, o sistema seja capaz de:

1. **Saber se o remetente é um cliente da empresa ou não**

Para tal, o componente do CM+ responsável pela comunicação com o Facebook implementa um método que, fazendo uma *query* à base de dados, verifica se o id do remetente da notificação corresponde a algum cliente registado da empresa. De reforçar que não basta que seja um cliente real da empresa, mas o seu *url/id* do perfil do Facebook tem de estar registado também.

Caso o cliente não esteja registado e se trate de um incidente, o sistema gera um comentário automático reportando o fato de o cliente não estar registado.

2. Identificar se a notificação corresponde a um incidente ou não (e descartar caso não seja)

De forma a perceber se a notificação se trata ou não de um incidente é necessário analisar o seu conteúdo. O ideal seria que o sistema o pudesse fazer sem que o cliente tivesse que fazer nada em particular para tal, sem ser enviar a descrição do problema. No entanto esse processamento envolveria um algoritmo de processamento de linguagem natural muito complexo que seria incompatível com o âmbito deste projeto.

Como tal, houve necessidade de criar uma alternativa a esse processo. Uma forma simples e eficaz encontrada foi a de colocar na publicação ou mensagem uma das *hashtags*: “#Incident”, “#Doubt” ou “#Problem”. Hoje em dia os utilizadores das redes sociais estão familiarizados com a utilização das *hashtags* e, como tal, considera-se ser uma boa alternativa. Sempre que este padrão é encontrado numa notificação é então considerado como sendo um incidente.

Apesar de não ser uma técnica 100% fidedigna no que respeita ao controlo de *spam*, considera-se que, aliada à identificação do remetente, permita uma boa filtragem das notificações, eliminando um número significativo de notificações não desejadas.

3. Identificar se é um incidente pré-existente ou não

Esta funcionalidade encontra-se inerente à anterior. Mais uma vez, seria ideal que o sistema tivesse capacidade de analisar o conteúdo e reconhecer se se trata de um incidente pré-existente ou não. Sendo uma implementação incompatível com o âmbito do projeto, de modo a perceber se é algo referente a um incidente existente ou proveniente de outro canal o cliente pode sinalizar isso através de uma *hashtag* como “#repost” ou “#open” ou ainda através da indicação do ID do incidente (aliando o id à *hashtag* que sinaliza o incidente).

4. Processar a notificação: identificar o produto ou serviço a que se refere, a descrição do problema e submeter o incidente no sistema

Caso a notificação seja sinalizada como sendo um incidente, o seu conteúdo vai ser processado. Em primeiro lugar vai ser analisado se o cliente aliou o problema a algum produto ou serviço. Ao referir um produto ou serviço com recurso a uma *hashtag* (“#Chávena” por exemplo) o sistema vai verificar se esse corresponde a um produto registado e existente na empresa e, em caso disso, vai associar esse mesmo produto ou serviço ao incidente.

A verificação da existência do produto na base de dados evita que sejam associados produtos inexistentes na empresa, e como tal que seja adicionada informação falsa na base de dados. Simultaneamente esta verificação faz com que todas as palavras associadas a *hashtags* que não correspondam a produtos sejam consideradas como parte da descrição.

É possível que ocorram frequentemente erros na escrita da denominação dos produtos e/ou serviços. De forma a minorar as situações de “descarte” de produtos incluídas nas descrições, quando é identificada uma *hashtag* que pode corresponder a um

produto e se pesquisa na base de dados se é ou não um produto, faz-se a pesquisa com recurso ao algoritmo *Jaro Winkler Distance*.

Este algoritmo permite calcular o nível de semelhança entre duas palavras (*strings*). Este método, uma variante do algoritmo *Jaro Distance*, devolve um valor entre 0 e 100 em que 100 corresponde a duas palavras exatamente iguais e 0 a duas palavras completamente diferentes. Este método parte do cálculo da distância de *Jaro*, dependente do número de caracteres iguais e do número de transposições e, posteriormente, de forma a otimizar esse cálculo, acrescenta um prefixo *p* que faz com que palavras com maior semelhança desde o início sejam melhor cotadas.

A opção por este método em detrimento de outros como a distância de *Levenshtein*, deveu-se ao fato de este ser mais apropriado para *strings* de pequena dimensão como o são os nomes de produtos/Serviços.

Exemplo de funcionamento

Para clarificar o funcionamento do processo de análise de conteúdo, tome-se como exemplo a publicação apresentada na imagem seguinte.

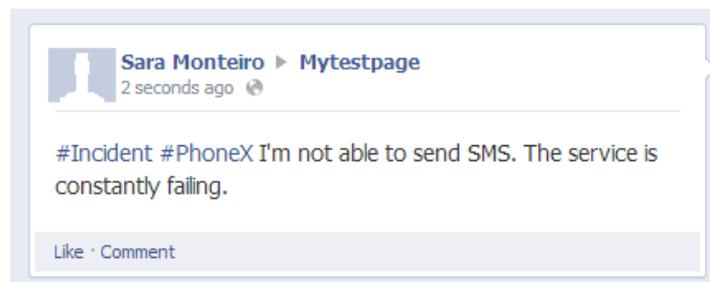


Figura 6.6– Exemplo de publicação no Facebook

Recebida a notificação da publicação e feita a verificação de que se trata de um cliente registado (ou não), o sistema vai analisar o conteúdo da publicação.

Ao encontrar a *tag* *#Incident* o CM+ assume tratar-se de um incidente e vai procurar referência a um produto. Encontra a *tag* *#PhoneX*, e procura na base de dados se existe algum produto denominado “PhoneX” (ou semelhante). Existindo, assume o produto do incidente como sendo esse e coloca o restante texto como descrição, procedendo à submissão na base de dados e notificação do utilizador. Caso houvesse mais *tags*, seriam consideradas como parte da descrição do problema.

6.3 Módulo de Integração com o Facebook

Este componente encontra-se dividido em duas camadas, a de comunicação, responsável pela recepção e envio de pedidos para sistemas externos e a lógica, onde se encontra o processamento desses pedidos.

Como visto, no sistema final, um cliente deve poder reportar dúvidas ou problemas através de vários canais. Nesses inclui-se o Facebook. De forma a compreender o fluxo de processos que ocorrem quando o cliente publica o problema no Facebook da empresa, apresenta-se em seguida o diagrama de atividades que modela esse processo.

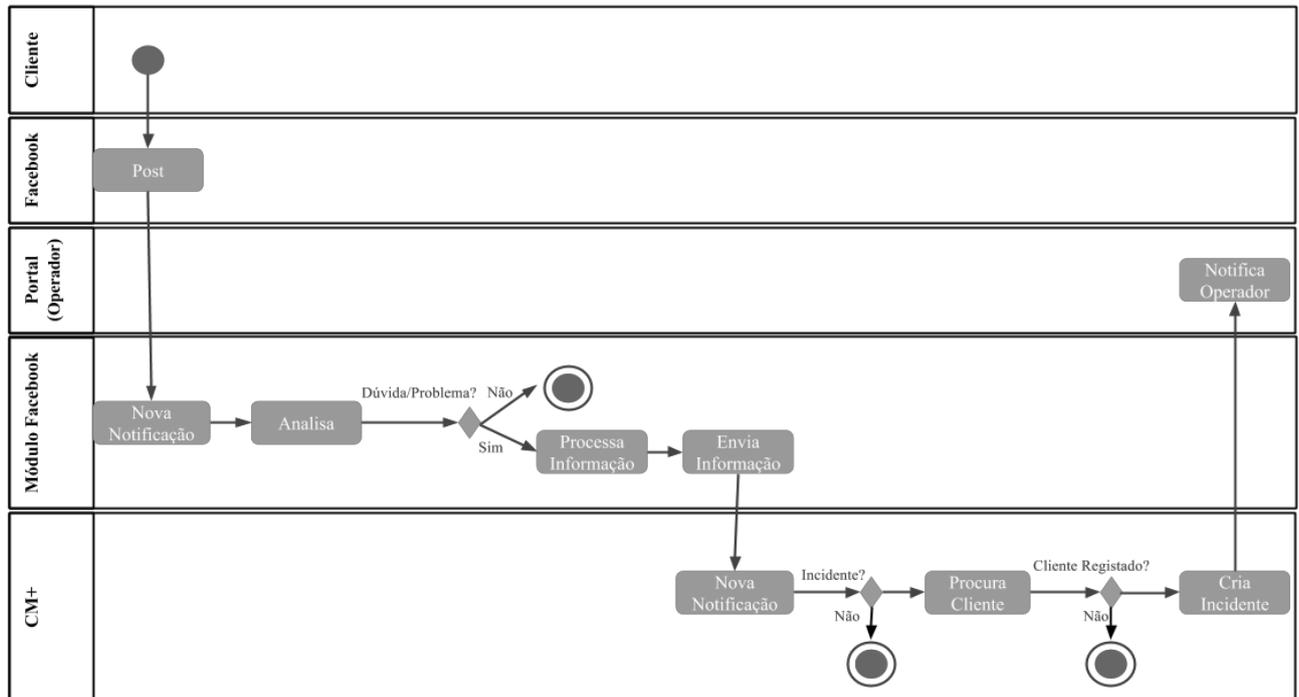


Figura 6.7– Diagrama de Atividade para a “Criação de Interação Via Facebook”

Salientam-se do diagrama os processos de **processamento de informação** e os **processos no CM+**. O primeiro corresponde os processos *core* que englobam a lógica do módulo. Nestes inclui-se o processamento para perceber que tipo de notificação se trata (*post*, comentário, etc.), a identificação do id da notificação, a recolha de informação necessária a ela relativa e conseqüente estruturação para envio para o CM+. Essa estruturação é feita em *xml* de acordo com um *xml schema* previamente estipulado.

Do lado do CM+, ao receber e ler a mensagem caso se trate de uma publicação de um incidente vai proceder aos processos descritos na secção anterior para a criação do incidente na base de dados.

Para que se compreenda também a comunicação no sentido oposto (CM+ - Facebook) apresenta-se o seguinte diagrama de atividade.

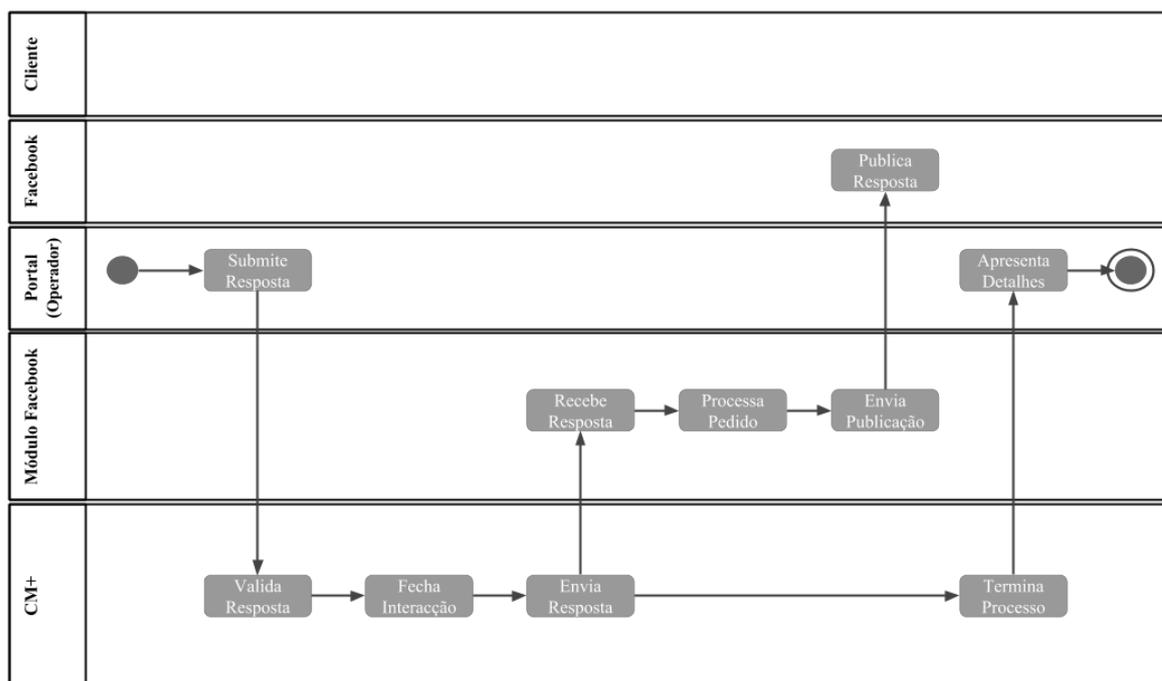


Figura 6.8– Diagrama de Atividade para a “Resposta a Interação Via Facebook”

A partir do momento em que o utilizador é notificado que tem uma interação por responder pode consultar os seus detalhes e dar-lhe resposta. Quando submete a resposta, o pedido é encaminhado para o CM+ que vai validar a resposta, associando-a à interação e determinando o canal pelo qual se deve dar resposta e posteriormente dá como fechada a interação.

O CM+ envia a informação para o módulo do Facebook que vai processar o pedido e posteriormente enviar o comentário (ou publicação ou mensagem) para o Facebook. Enviada a mensagem, o CM+ dá o processo como terminado e envia os detalhes para o portal.

Neste módulo são ainda implementados métodos que permitem: fazer e obter publicações e comentários, obter dados estatísticos da página e ainda responder a mensagens. Algumas destas funcionalidades não são necessárias para o cumprimento dos requisitos definidos (como fazer e obter publicações e estatísticas da página). No entanto, enriquecem este módulo conferindo-lhe maior capacidade para desenvolver outras funcionalidades ou ser utilizado noutros contextos.

Comunicação com o Facebook

A comunicação com o *Facebook* é feita através de pedidos http (POST e GET). Quando são feitas alterações na página (quando um cliente faz uma publicação, por exemplo) os servidores do *Facebook* geram um pedido http POST para o servidor do sistema. Esse pedido é reencaminhado para o módulo que depois faz o processamento da notificação.

No sentido contrário, sempre que o módulo recebe pedidos para fazer comentários, ou consultar publicações, etc., o módulo gera um pedido http (GET ou POST) para o *Facebook* que executa o pedido.

Processamento das Notificações e Pedidos

Este componente funciona como um intermediário entre o *Facebook* e o CM+ (ou outro sistema), transformando as mensagens e pedidos recebidos (quer de um lado quer de outro) em objetos estruturados e compreensíveis e, encaminhando-os para os locais certos.

Desta forma, quando recebe uma notificação do *Facebook*, por exemplo, vai em primeiro lugar perceber qual o campo alterado e fazer um pedido para obter todas as informações dessa alteração. Posteriormente agrega essas informações e coloca-as num objeto *java* que é enviado para a *queue* no servidor.

O CM+ tem o *schema xml* que permite interpretar o objeto e como tal vai depois trata-lo como descrito na secção anterior.

Por outro lado, quando o CM+ envia pedidos para o módulo, este vai interpretá-los e utilizar a informação necessária para fazer o pedido pretendido ao *Facebook*.

Mecanismo de Throttling

A documentação da API do *Facebook* revela que existe um limite para o número de pedidos. No entanto, como esse limite depende do esforço que cada um exige a nível de CPU, etc. não é estipulado um número fixo.

Algumas pesquisas no fórum de dúvidas da API apontam para um limite próximo de 600 pedidos em 600 segundos. Atingido esse limite, os pedidos da hora seguinte são bloqueados. Nestes pedidos não se incluem os “*Real Time Updates*”, ou seja as notificações provenientes do *Facebook* não contribuem para essa contabilidade.

Para evitar que se gere esse bloqueio há a necessidade de criar um mecanismo que evite que sejam gerados mais de X pedidos em Y segundos. Esse mecanismo é criado ao nível deste módulo que, antes de enviar os pedidos confirma o número de pedidos enviados nos últimos Y segundos e, caso seja superior a um limite X, guarda em fila de espera.

6.4 Módulo de Recomendação de Incidentes

Em termos de resultado final, este componente do sistema deve possibilitar um fluxo como o exemplificado no diagrama da figura 6.6.

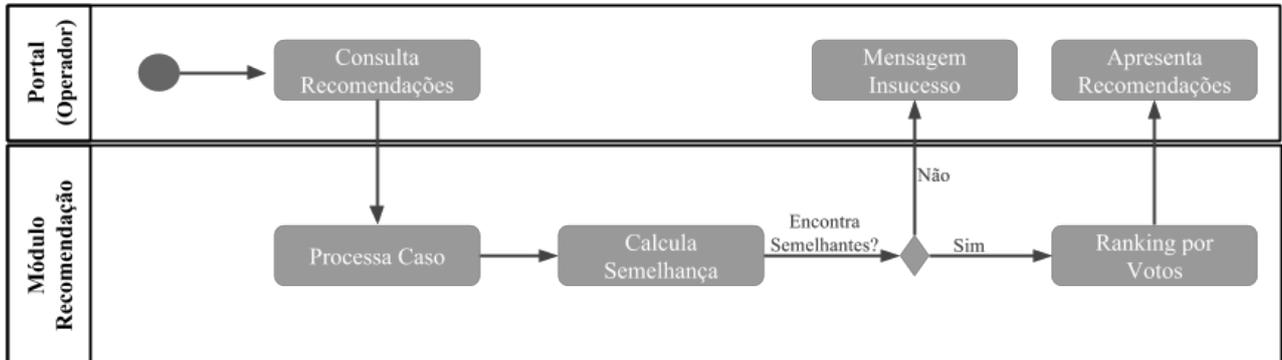


Figura 6.9– Diagrama de Atividades para a “Recomendação de Soluções”

Podem salientar-se duas ações fundamentais: o **Cálculo de Incidentes Semelhantes** e a **Procura das Melhores Soluções**. Estas, em conjunto, representam o funcionamento central do sistema de recomendação em desenvolvimento.

Tal como definido no capítulo da arquitetura e especificação do sistema, o componente de recomendação constitui-se por uma camada lógica em comunicação com a base de dados e está exposto aos serviços externos via *Web Services*. Para que a parte lógica implemente as funcionalidades requeridas é implementada de acordo com a figura seguinte.

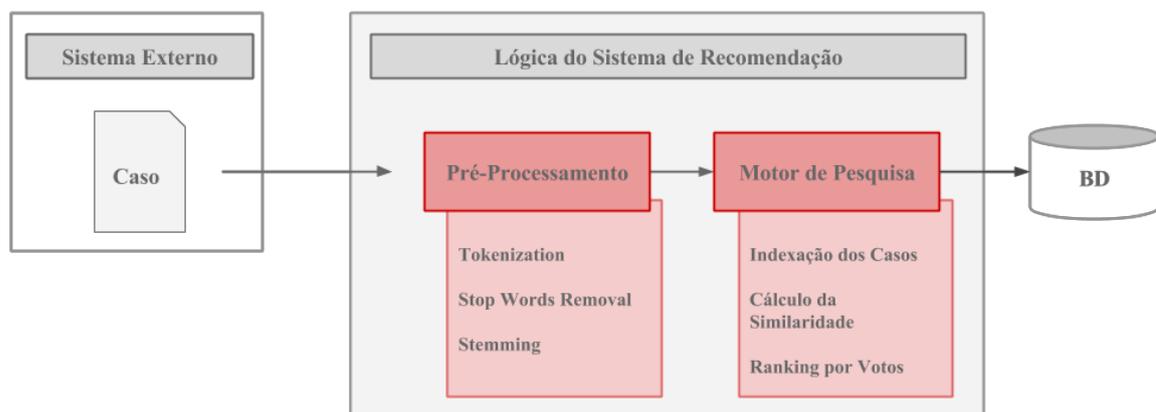


Figura 6.10– Lógica do Sistema de Recomendação

6.4.1 Pré-processamento

A fase de pré-processamento representa os métodos necessários para extrair da descrição do caso os termos que verdadeiramente têm significado e que, como tal, devem ser indexados. Para atingir esse objetivo são implementados 3 processos diferentes:

1. **Tokenization:** Este processo permite separar as palavras que constituem a descrição do problema e simultaneamente elimina sinais de pontuação, hífen e números.
2. **Stop Words Removal:** A remoção de *stop words* faz com que sejam eliminadas palavras como determinantes, pronomes ou outros que se usam com muita frequência mas que não atribuem significado à descrição.
3. **Stemming:** Este último processo representa a capacidade de reduzir as palavras à sua raiz. Este passo é de grande relevância por exemplo para os verbos. Muitas vezes usamos um verbo em diferentes tempos verbais e, ao reduzir esses tempos à raiz da palavra, aumenta a probabilidade de encontrar correspondências. Por exemplo, um operador pode usar a descrição “*Product X not working*” e outro pode usar “*Product X doesn’t work*”. Reduzindo o “*working*” à sua raiz (*work*) vai haver um grau de correspondência maior com o 2º caso.

Esta fase de pré-processamento ocorre em dois momentos do fluxo do sistema, sempre que um caso é inserido na base de casos e quando chega um pedido de recomendação. No segundo caso, o pedido de recomendação inclui um objeto “incidente” com a descrição que é processada para que fique reduzida ao conteúdo significativo e posteriormente seja comparada com as descrições da base de casos.

6.4.2 Motor de Pesquisa

A lógica do sistema de recomendação engloba outro componente, o qual é referido na imagem como motor de pesquisa. Neste, são implementados fundamentalmente dois processos: a indexação dos casos e o cálculo da semelhança entre casos.

O número de incidentes mapeados por este sistema pode variar muito e, em função de cada sector em específico, pode ser muito significativo. Este fato cria a necessidade de indexar os casos adicionados à base de casos para que a pesquisa seja mais rápida e, conseqüentemente, mais eficiente.

Para a indexação dos casos opta-se por recorrer à biblioteca *Lucene*. Esta é inteiramente desenvolvida em *Java* e para além de disponibilizar funcionalidades completas para a implementação de motores de pesquisa, apresenta também elevadas capacidades no que respeita à criação de índices. Através do recurso a esta biblioteca é criado um índice invertido. Contrariamente ao que acontece nos índices normais, nestes, a cada termo é associado uma listagem dos casos que o contém.

Até ao momento foram descritos os processos associados à criação da base de casos e à sua indexação. Estes correspondem à criação do ambiente necessário para que seja possível a pesquisa e aquisição de casos semelhantes a um caso em específico. Segue-se agora a análise do processo que ocorre quando é requerida recomendação para um caso específico.

Quando um sistema externo invoca este módulo através do *web service* disponibilizado para obter recomendações relativas a um caso deve enviar informação relativa à descrição de ao produto desse mesmo caso.

Ao receber a informação, este componente vai em primeiro lugar processar o caso, extraíndo apenas a parte relevante e posteriormente vai procurar os casos mais relevantes e apresentar um ranking desses mesmos casos. Como visto anteriormente pretende-se que a recomendação ocorra em duas fases, primeiro sejam encontrados os casos mais relevantes em função do seu conteúdo e, seguidamente, sejam selecionados os mais votados e apresentado o seu ranking em função dos votos de cada um.

Para o cálculo baseado no conteúdo procura-se que sejam encontrados os casos cujo produto/serviço seja o mesmo que o caso-chave e que apresentem a maior semelhança possível ao nível da descrição. Para obter a semelhança entre descrições de dois casos faz-se

uso do algoritmo *Nearest Neighbor*. Considerando estes dois factores o algoritmo devolve uma lista ordenada dos casos mais relevantes (mais semelhantes).

Desta lista são escolhidos os primeiros 20 casos (ou seja, os 20 casos mais relevantes) e esses são sujeitos à avaliação colaborativa. Destes 20 casos serão então escolhidos os 10 casos mais votados, ou seja, aqueles que, por votação dos utilizadores, são considerados mais relevantes.

6.5 Portal

De forma a mapear as funcionalidades implementadas quer no CM+ quer nos dois novos módulos desenvolvidos, foi implementado em HTML5 um portal de acesso básico. O objetivo deste portal é servir como protótipo ao expor as funcionalidades desenvolvidas, mostrando de forma prática como se poderiam encaixar num sistema de CRM de uma empresa.

A título de exemplo apresenta-se (na figura 6.11) a área de Gestão de Incidentes. Nesta área são listados os incidentes em aberto (atribuídos ao utilizador da sessão). O utilizador pode seleccionar um dos incidentes para ver os seus detalhes (3), pode fazer pesquisa de incidentes e pode ainda criar um novo incidente (Canais manuais)(2).

A partir do momento em que faz *login* terá sempre na barra superior (1) a área de alertas onde pode consultar notificações e onde aparecem os alertas de novas notificações. Para criar o mecanismo de notificações em tempo real foi utilizado a metodologia de *SSE* (*Server Sent Events*). Assim, sempre que o módulo de integração com o *Facebook* notifica o CM+ de um novo incidente e este o cria na base de dados, gera também uma mensagem para o portal que faz surgir no utilizador a notificação de um novo alerta. Caso o utilizador esteja *offline* a notificação fica na *queue* e é entregue quando ele ficar *online*.

ID	Description	Customer Id	Status	Date
1353	[* Test 6*]	9999999	Created	22/6/2014
1352	[* Test 5*]	9999999	Created	22/6/2014
1351	[* Test 4*]	9999999	Created	22/6/2014
1350	[* Test 3*]	9999999	Created	22/6/2014
1349	[* Test 2*]	9999999	Created	22/6/2014
1348	[* Test 1*]	9999999	Created	22/6/2014

Figura 6.11– Área de Gestão de Incidentes do Portal

Caso opte por consultar os detalhes de um incidente surge um ecrã semelhante ao anterior onde são listadas as interações ocorridas nesse âmbito (figura 6.12). As interações não respondidas são listadas a vermelho e o utilizador pode seleccionar cada uma delas para ver os seus detalhes e dar-lhe resposta.

Na área de detalhes de incidente o utilizador pode ainda seleccionar a opção “*Similar Cases*”. Esta opção desencadeia um pedido ao sistema de recomendação para que seja gerada uma lista de incidentes semelhantes e respetivas soluções. Perante essa lista o utilizador pode consultar os detalhes de cada um dos elementos e eventualmente votar em algumas soluções.

Ainda nesta área o operador pode criar novas interações (5). Escolhendo o canal, descrição e/ou solução pode criar a interação e esta ficará associada ao incidente em contexto.

CMP Admin Incident Details

5 [+New](#) [Similar Cases](#) 4

ID	Data	Channel	Description	Answer	Responsable
50001545	25/6/2014	CC	Test 1 Interaction	null	4
50001470	22/6/2014	FB	[" Test 6"]	Unanswered	4

Figura 6.12– Detalhe de Incidente

O portal desenvolvido com recurso a HTML5 e javascript permite assim aceder às funcionalidades de gestão de incidentes com integração com os módulos do *Facebook* e de recomendação.

Capítulo 7

Verificação e Validação

De forma a validar e avaliar o sistema desenvolvido é necessário desenvolver um plano de testes capaz de detetar falhas na implementação das funcionalidades, perceber se o sistema se comporta de acordo com os requisitos e objetivos definidos e avaliar a qualidade e *performance* do produto desenvolvido.

Para a fase de testes deste projeto, opta-se por uma abordagem de duas fases. Na primeira fase considera-se a realização de **testes funcionais**, em que se examinam os resultados obtidos pelo sistema e se confrontam com aquilo que era o esperado. Na segunda fase realizam-se **testes não funcionais**. Estes pretendem validar o comportamento do sistema no que diz respeito aos requisitos não funcionais. A este nível analisam-se factores como a usabilidade, *performance* e qualidade de resultados do sistema implementado.

A escolha desta abordagem em duas fases relaciona-se com o fato de este capítulo pretender analisar e verificar se os requisitos estipulados inicialmente foram ou não atingidos. Visto que esses requisitos se dividiam num componente funcional e num não funcional, considera-se que também a fase de validação se deve subdividir nestas duas etapas.

7.1 Testes Funcionais

7.1.1 Testes Unitários e de Integração

Ao longo da fase de desenvolvimento, as diversas unidades criadas foram testadas, tendo sido corrigidas falhas detetadas em algumas funcionalidades, métodos e subrotinas. A realização deste conjunto de testes evitou que se gerassem falhas maiores e com impacto significativo no produto final.

Para a realização deste tipo de testes estipulou-se para cada unidade desenvolvida um conjunto de dados de entrada e os dados de saída esperados. Nos casos em que se verificou discrepância entre os dados obtidos e os esperados, foram efetuadas as alterações e correções necessárias.

Numa fase mais avançada do desenvolvimento, em que várias unidades estavam já implementadas e testadas foi necessária a realização de testes de integração de forma a compreender como as unidades interagem entre si e compreender se essa interação ocorre sem falhas e de acordo com o espectável.

Para esta fase foi adotada uma abordagem *Bottom-up* em que se foram agrupando as unidades resultantes da fase anterior e se foi testando a integração entre elas. Mais uma vez foram definidos dados de entrada e comportamento esperado de cada unidade e correções foram implementadas ao registar comportamentos não esperados.

7.1.2 Testes de Sistema

Desenvolvidas e testadas as várias unidades do sistema e a integração entre elas pode passar-se à fase seguinte do processo de validação, os testes de sistema. Neste tipo de testes são confrontados os requisitos funcionais definidos inicialmente com as funcionalidades do sistema produzido.

Para fazer esta validação foram estudadas quais as funcionalidades requeridas para cada requisito e qual o resultado que se espera obter por parte do sistema. O registo de cada teste de sistema foi feito de acordo com o especificado na imagem seguinte.

<i>ID – Teste</i>	<i>X</i>
Prioridade	Elevada ou Média ou Baixa
Requisito	Breve descrição do requisito funcional a que corresponde
Condições	Indica as condições/hipóteses necessárias para a realização do teste
Dados de Entrada	Define quais os dados e operações a realizar para efetuar o teste
Resultado Esperado	Apresenta qual o comportamento e resultados que se espera que o sistema tenha
Estado	✔ - Passou ou ✘ - Não Passou
Detalhes	Espaço para observações relevantes relativas à execução do teste

Tabela 7.1– Tabela para o registo dos testes de sistema

Este registo baseia-se num dos formatos utilizados pela equipa de testes da Novabase para a realização de testes de sistemas.

No anexo 5 podem encontrar-se todos os testes de sistema realizados registados de acordo com esta estrutura.

A realização deste processo de validação permitiu concluir que as funcionalidades implementadas respondem aos requisitos estipulados, verificando-se apenas pequenas falhas em algumas funcionalidades. Essas falhas foram assinaladas e estão em processo de correção.

7.2 Testes Não Funcionais

7.2.1 Testes de Qualidade (Sistema de Recomendação)

Um dos requisitos não funcionais do sistema é a qualidade de resposta do Sistema de Recomendação. De fato, deste sistema espera-se não apenas que gere recomendações, mas que gere recomendações com significado e utilidade para o utilizador. Para tal foi desenvolvido um plano de testes de qualidade deste módulo.

Base de Casos

Dada a natureza do ambiente em que foi desenvolvido o CM+ não foi possível obter uma base de incidentes reais e como tal houve necessidade de procurar uma alternativa.

Estudadas algumas possibilidades optou-se pela utilização de casos descritos num conhecido *site* de perguntas e respostas, o *StackOverflow* (www.stackoverflow.com). Neste *site* são colocadas dúvidas/problemas pelos utilizadores e outros apresentam possíveis soluções e

sugestões. Estas soluções podem ser votadas de acordo com a sua utilidade. Para além disso, a descrição do problema é acompanhada de *tags* que têm o intuito de identificar o tema da questão.

É possível neste site encontrar uma estrutura semelhante à que se procura:

- Descrição do problema, correspondente à descrição do incidente;
- Tags representativas do tema, que pode ser entendida como a definição de um produto ou serviço a que se associa o incidente;
- Solução do problema, que pode corresponder à solução do incidente;
- Ranking, equivalente ao número de votos da recomendação.

Esta semelhança entre a estrutura dos casos deste projeto em concreto e os definidos pelo site esteve na origem da decisão de se optar por usar estes casos para fazer a validação do módulo.

Foram extraídos cerca de 1000 casos do site e armazenados na base de casos de acordo com a estrutura definida. Estes casos foram extraídos por tema (produto), sendo extraídos cerca de 30 temas diferentes.

Métricas de Validação

O objetivo deste processo de avaliação é analisar a eficiência do sistema, neste caso alcançada através da geração de resultados relevantes para o problema apresentado.

Consideram-se duas métricas para a avaliação dos resultados: *Recall* e *Precisão* (28).

Recall: Reflete a capacidade do sistema mostrar todos os resultados úteis.

$$Recall = \frac{R}{R + NR}$$

R – N° de Casos Relevantes Extraídos

NR – N° de casos Relevantes Não-Extraídos

Precisão: Reflete a capacidade do sistema de extrair apenas resultados úteis.

$$Precisão = \frac{R}{R + NRE}$$

R – N° de Casos Relevantes Extraídos

NRE – N° de casos Não - Relevantes Extraídos

O objetivo é maximizar as duas métricas. Mas é necessário ter em atenção que, a otimização de uma pode corresponder a uma diminuição da outra. Por exemplo, ao aumentar o número de casos extraídos do sistema, aumenta a probabilidade de apresentar mais casos relevantes, aumentando o valor da *recall*. No entanto, aumenta também a quantidade de casos irrelevantes e consequentemente obtém-se uma precisão mais baixa.

Por outro lado, invertendo o cenário também se verificaria que o valor da *recall* seria mais baixo.

Metodologia de Validação

Foram realizados 25 casos de teste. Para cada um foi feita uma pesquisa no *StackOverflow* sobre um tema existente na BD e foram extraídos os 10 casos considerados mais relevantes pelo *site*. Esses 10 casos foram adicionados à BD, acrescentando aos mais de 800 casos lá existentes (referentes aos mesmo temas e outros).

Posteriormente, foram gerados pedidos *SOAP* ao sistema de recomendação com a mesma *query* de pesquisa efetuada no *site*. Os resultados foram anotados e comparados com os obtidos no *site*. Para cada teste foram calculados os valores de *Recall* e Precisão. Em seguida analisam-se esses resultados.

Análise de Resultados

Os valores obtidos para a métrica *recall* são bastante satisfatórios. Como é observável no gráfico seguinte, 19 dos 25 casos apresentam todos os casos relevantes. No total dos casos de testes, o valor de *recall* é de 96% (a tabela detalhada com os resultados encontra-se em anexo).

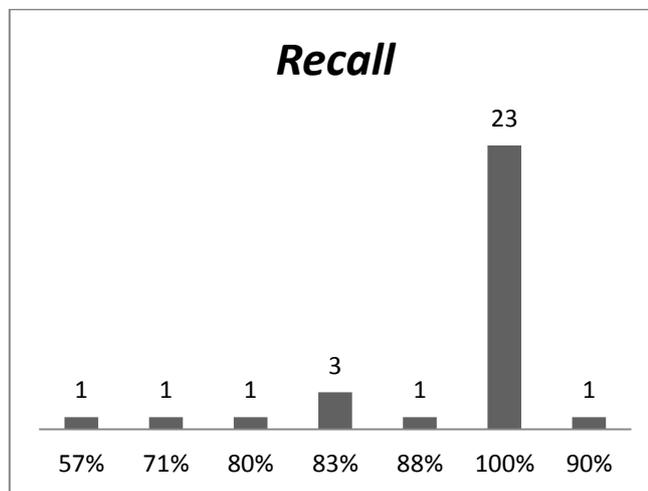


Figura 7.1 – Valor da métrica *Recall* para os casos de teste realizados

Quanto aos valores de Precisão, os resultados não são tão satisfatórios. O valor médio de precisão é de 60%, tendo o máximo sido atingido por dois casos (80%) e o mínimo também por dois casos (40%).

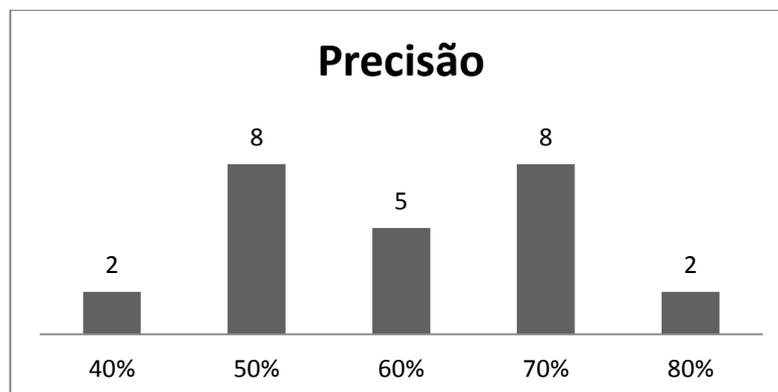


Figura 7.2– Valor da métrica Precisão para os casos de teste realizados

Este valor de precisão mais baixo do que o pretendido é facilmente justificável. Ao nível da implementação optou-se pela extração de 10 casos em todas as recomendações, independentemente do nível de semelhança. Esta decisão relacionou-se com o fato de ser implementado um sistema de recomendação híbrido e afeta a precisão nos casos em que o número de casos relevantes é inferior a 10.

Os casos selecionados para teste apresentavam todos um número de casos relevantes entre 6 e 8, o que valida este valor de precisão.

Por outro lado, caso o número de incidentes relevantes seja superior a 10, esta metodologia vai beneficiar a precisão mas afetar o valor da *recall*.

Assim, considera-se que deve ser futuramente analisada a possibilidade de definir um limiar de semelhança. Ou seja, utilizar o valor de semelhança obtido pelo algoritmo e apenas selecionar os casos com uma semelhança superior a um limiar x . Aplicando posteriormente a recomendação colaborativa. Dessa forma espera-se otimizar as duas métricas analisadas.

7.2.2 Testes de Performance

Para validar alguns dos requisitos não funcionais definidos, tais como a escalabilidade e a garantia da atualidade da informação realizaram-se alguns testes de *performance*.

Módulo de Recomendação

Para testar a escalabilidade do sistema de recomendação e perceber como este se comporta quando sujeito a um maior volume de pedidos foram realizados testes de carga. Para a sua realização recorreu-se à ferramenta de testes de carga do *software SOAP UI*.

Foram testados cenários de 50, 200, 400, 600,800, 1000 e 5000 pedidos. Para cada número de pedidos foi variado o número de utilizadores virtuais (UV's). O número mínimo de UV's usado foi de 20 e o número máximo 180.

O intervalo entre a execução de cada pedido ao sistema foi estipulado no valor de 3000ms. No anexo F encontra-se a tabela com os resultados obtidos para cada um dos testes realizados.

A análise da tabela permite verificar que o sistema apresenta um comportamento estável. A partir dos 200 pedidos, variar o número de utilizadores virtuais não leva a uma inconsistência do sistema. O gráfico seguinte apresenta o valor das transações por segundo para alguns dos testes realizados. De forma a simplificar a análise do gráfico representam-se apenas os testes realizados para 30, 120 e 180 UV's.

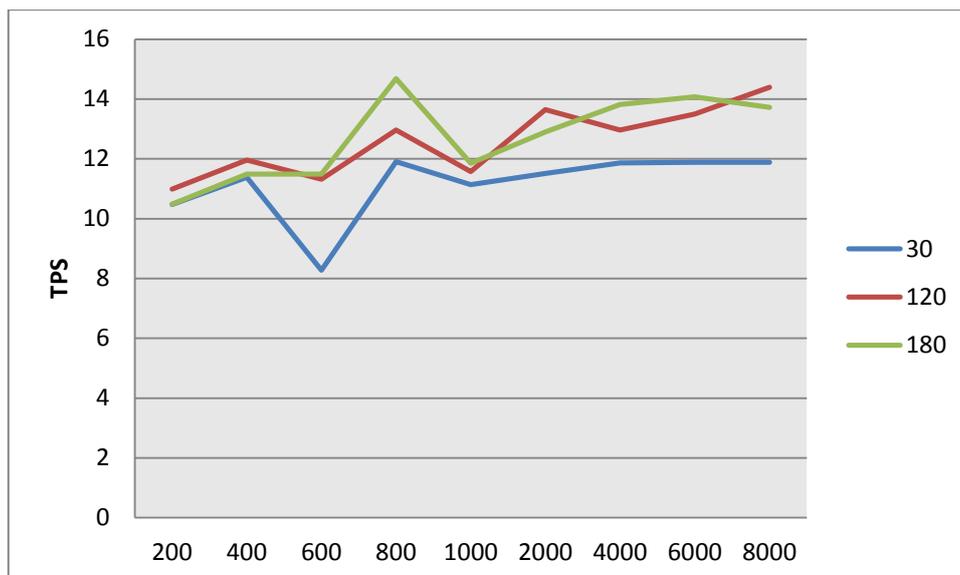


Figura 7.3– Variação do número de Transações por Segundo em função do número de pedidos e do número de clientes

Como se verifica, o sistema comporta-se de forma semelhante e estável quando sujeito a variações de número UV's. Apesar de no gráfico apenas se representarem 3 dos 9 casos testados, por questões de perceptibilidade, a realização dos testes comprovou que o número de transações por segundo que o sistema é capaz de realizar não varia significativamente ao variar quer o número de utilizadores, quer o número de pedidos.

O gráfico pretende mostrar exatamente esse fato. Não se verifica uma variância significativa no comportamento do sistema pelo fato de se alterar o número de pedidos ou utilizadores. Analisando todos os dados obtidos (através da tabela em anexo) verifica-se que o mínimo número de transações por segundo realizadas foi de 7,78 (para 20 UV's e 200 pedidos) e o máximo foi de 16,04 (para 140 UV's e 4000 pedidos) o que representa uma diferença de 8,26 transações por segundo, um número baixo que reflete o comportamento estável do módulo de recomendação quando escalado.

Na tabela em anexo verifica-se que 180 é o número a partir do qual se começam a verificar falhas nos pedidos. Estas falhas correspondem a problemas de conectividade aos serviços, ou seja denotam a incapacidade do sistema para receber mais ligações.

Módulo de Integração com o Facebook

Ao nível deste módulo pretende-se verificar e validadas duas capacidades:

- Manter a informação entre o Facebook e o sistema atualizada;
- Escalabilidade.

Informação Atualizada

Para validar que a comunicação entre o sistema e o Facebook ocorre praticamente em tempo real foram realizados testes para medir o intervalo de tempo desde que há publicação até que a outra parte é notificada.

No sentido do CM+ para o *Facebook* é difícil fazer essa validação de forma minuciosa. Apesar de se ver a publicação a ocorrer, o tempo apenas apresenta precisão de minutos o que nos impede de precisar o intervalo de tempo entre a submissão no CM+ e a publicação

no *Facebook*. No entanto, nos testes realizados o intervalo de tempo foi sempre inferior a 1 minuto.

No sentido contrário, a notificação enviada pelo *Facebook* apresenta um parâmetro com o *timestamp* relativo à hora de publicação com precisão de milissegundos. Anotando esse valor e os valores relativos à chegada da notificação ao módulo e ao CM+ é possível ter uma noção da rapidez com que a informação é atualizada.

Visto que as máquinas do *Facebook* e a VM onde corre o sistema podem não estar sincronizadas estes valores podem apresentar algum erro associado.

No anexo G encontra-se a tabela com os valores obtidos. Esses valores mostram que o intervalo de tempo médio entre a publicação no *Facebook* e a notificação no CM+ é de 46 segundos. Desse tempo, 91.30% corresponde ao tempo entre a publicação e a recepção da notificação no módulo. Estes valores revelam que o sistema tem a capacidade de manter a informação atualizada de forma bastante rápida (inferior a 1 minuto).

Para finalizar esta parte é importante referir que estes valores não dependem exclusivamente do sistema implementado, estando dependentes do Facebook, dos seus servidores e capacidade. O fato de depender de um sistema externo pode levar a que os intervalos de tempo sejam superiores mesmo que o sistema implementado esteja em pleno funcionamento.

Escalabilidade

Para analisar a escalabilidade do sistema pensou-se primeiramente em seguir a metodologia seguida para o módulo de recomendação. No entanto, neste caso, essa tarefa revelou-se verdadeiramente mais complexa.

Apesar de ao nível do CM+ os serviços de interação como *Facebook* estarem expostos através de *web services*, a comunicação entre o CM+ e o módulo faz-se por via de *queues*, como visto no capítulo de implementação. Aplicando o método anterior seria apenas possível testar a capacidade de suportar carga ao nível do *web service*, não sendo perceptível se essa carga seria comportável *queues*.

Da mesma forma, aplicando carga às *queues*, seria apenas observável a capacidade de cada uma delas e não a capacidade do sistema como um todo.

Assim, a estratégia pensada envolve a monitorização do comportamento do sistema em dois momentos: quando o pedido é recebido no *web service* e antes de ser enviado para o *Facebook*. Em cada um destes momentos são registados numa tabela o id do pedido e o *timestamp* em que ocorre. Posteriormente, estes dados são cruzados de forma a garantir que todos os pedidos submetidos são enviados para o *Facebook* e de forma a calcular o tempo de resposta do sistema.

Visto que o objetivo é testar o sistema e a sua capacidade, os pedidos não são na realidade enviados para o *Facebook* de forma a evitar gerar *spam* desnecessário. Os pedidos gerados são apenas reencaminhados para um servidor de teste. De notar que devido ao limite de número de pedidos diário à API e consequente mecanismo de *throttling* ao nível do momento de envio, a chegada dos pedidos ao *Facebook* pode sofrer algum *delay*.

No anexo G apresenta-se uma tabela com as médias dos tempos obtidos para cada caso de teste. O gráfico seguinte sintetiza os resultados obtidos.

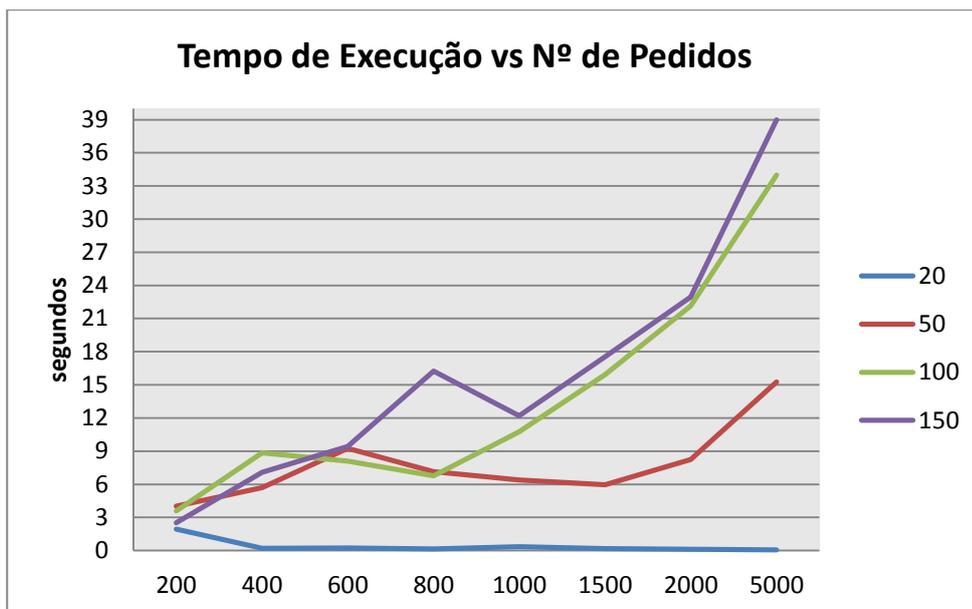


Figura 7.4– Tempo de execução dos Pedidos ao módulo de integração com o Facebook (Em função do número de pedidos)

Verifica-se que um aumento do número de pedidos tem um impacto significativo no tempo de execução do sistema, sendo que, quando acompanhado de um aumento de clientes, esse impacto é ainda mais significativo.

Este aumento do tempo de resposta do sistema está diretamente relacionado com a capacidade das *queues* mas também com a capacidade de processamento do próprio sistema. A partir das 150 UV's começaram a verificar-se falhas de conexão motivadas pela incapacidade do *web service* de suportar mais ligações.

Capítulo 8

Conclusão

O presente trabalho desenvolveu-se com foco num objetivo fundamental, expandir o CM+, dotando-o de novas funcionalidades transversais aos vários sectores de mercado e de acordo com as tendências emergentes da área de CRM.

Este objetivo foi, numa fase inicial, subdividido de forma a tornar mais perceptível qual o âmbito específico do trabalho e de que forma deveria ser estruturado. Dessa divisão resultaram três objetivos fundamentais:

- Implementação de funcionalidades de gestão de serviços multicanal;
- Integração com o *Facebook* (para a gestão de incidentes);
- Desenvolvimento de um sistema de recomendação de soluções de incidentes.

A etapa seguinte envolveu um processo de análise de requisitos e também um processo de planeamento, quer da arquitetura de desenvolvimento e implementação do sistema, quer planeamento temporal de execução de tarefas.

Estudadas as competências necessárias para o trabalho e criado o ambiente de desenvolvimento foi possível dar início ao processo de implementação das funcionalidades estipuladas. Esta fase agrupou-se em 3 etapas, correspondendo cada uma a implementações de diferentes módulos: CM+, Módulo de Integração com o Facebook e Módulo de Recomendação. Paralelamente a estes 3 módulos foi ainda implementado um portal em HTML5 cujo objetivo é, fundamentalmente, mapear as funcionalidades do sistema implementado de forma a tornar possível a visualização do trabalho desenvolvido e, funcionando como prova de conceito, mostrar como poderia ser aplicado num contexto real.

A última fase do projeto correspondeu à verificação e validação do sistema. Parte da validação ocorreu ao longo da implementação através da realização de testes unitários e de integração a cada uma das unidades desenvolvidas e à integração entre elas, respetivamente. De forma a assegurar que o sistema cumpre os requisitos estipulados, foram realizados testes de sistema e, por fim, alguns testes de *performance* e de qualidade de resultados com o intuito de aferir se o sistema estava de acordo com os requisitos não funcionais definidos.

Uma análise ao trabalho desenvolvido permite concluir que os objetivos estipulados no início deste estágio foram plenamente alcançados. A sua implementação comprovou não só que é possível que o CM+ apresente uma gestão unificada de incidentes multicanal, mas também que é possível complementá-la com a integração com redes sociais e que se pode tirar partido desta para estar mais perto do cliente e das suas necessidades atuais, caminhando lado a lado com as tendências do mercado.

Quanto ao sistema de recomendação, verificou-se que é capaz de gerar sugestões válidas e contextualizadas que podem contribuir para otimizar o processo de resolução de incidentes.

O processo de validação demonstrou que, apesar do sistema responder positivamente aos testes elaborados, podem ser feitas melhorias. Por exemplo, a precisão do sistema de recomendação pode ser melhorada ao ser complementada com outras métricas.

Para além de cumprir os objetivos funcionais definidos, considera-se que este trabalho veio contribuir para uma nova visão daquilo que é o CM+ e abrir horizontes para a sua expansão na vertente da gestão de serviços e não só.

8.1 Trabalho Futuro

A área de CRM e em particular a tendência do *Social CRM* são muito vastas e com um potencial de aplicação enorme não só no que respeita à área de serviços, abordada neste projeto, mas também na área de vendas e *marketing*. Também a área de conhecimento de contexto é um tema cada vez mais emergente nesta área e com inúmeras aplicabilidades.

O trabalho desenvolvido foca-se numa pequena parte de tudo o que pode ser explorado na área e, como tal, pode ser encarado como um ponto de partida para um sistema maior, que extenda as suas funcionalidades para as restantes áreas de CRM e até para outras áreas.

No que respeitas às funcionalidades desenvolvidas em específico para este projeto considera-se que se podem implementar melhorias quer ao nível da integração com o *Facebook*, quer ao nível do sistema de recomendação.

O processamento das notificações recebidas poderia ser sujeito a um processamento de linguagem natural capaz de identificar se é uma dúvida ou não, sem ser necessário que o utilizador o marque como tal (com recurso à *hashtag*). Uma outra melhoria interessante seria analisar os comentários feitos por outros clientes à publicação e tê-los em conta no processo de resolução do incidente.

Ao nível do sistema de Recomendação, o cálculo do limite de similariedade apropriado (validado por testes) contribuiria para o aumento da precisão dos resultados. Também a este nível se poderiam aplicar mais técnicas de processamento de linguagem natural, tais como *Topic Modeling*, que contribuiriam para o aumento da qualidade e relevância de resultados obtidos.

Referências

1. *Novabase*. [Online] [Citação: 31 de 10 de 2013.] <http://www.novabase.pt/pt/Novabase/Pages/Historia.aspx>.
2. **Patrick J. Sullivan, Ed Thompson**. *Magic Quadrant for CRM Service Providers, Worldwide*. s.l. : Gartner, Setembro 2013.
3. **Joanne M. Correia, Yanna Dharmasthira, Chris Pang**. *Market Snapshot: Customer Relationship Management, Worldwide, 2012*. s.l. : Gartner, Maio 2013.
4. **Ed Thompson, Patrick J. Sullivan**. *CRM Applications Deployed by Consultancies in 2013 Show Which Skills Are Prevalent*. s.l. : Gartner, Outubro 2013.
5. Founding Universe. [Online] [Citação: 21 de 11 de 2013.] <http://www.fundinguniverse.com/company-histories/siebel-systems-inc-history/>.
6. Salesforce. [Online] [Citação: 21 de 11 de 2013.] <http://www.salesforce.com/eu/company/foundation/>.
7. Salesforce. [Online] [Citação: 21 de 11 de 2013.] <http://www.salesforce.com/eu/customers/>.
8. Community Dynamics. [Online] [Citação: 21 de 11 de 2013.] [.http://community.dynamics.com/crm/b/survivingcrm/archive/2013/09/25/history-of-microsoft-s-crm-software.aspx](http://community.dynamics.com/crm/b/survivingcrm/archive/2013/09/25/history-of-microsoft-s-crm-software.aspx).
9. NxtBook (CRM Magazine). [Online] [Citação: 21 de 11 de 2013.] <http://www.nxtbook.com/nxtbooks/crmmedia/crm0708/#/30>.
10. Gartner. [Online] [Citação: 21 de 11 de 2013.] <http://www.gartner.com/technology/reprints.do?id=1-1HA4SNM&ct=130717&st=sb>.
11. SugarCRM. [Online] [Citação: 21 de 11 de 2013.] <http://www.sugarcrm.com/newspress/sugarcrm-2004-review>.
12. SugarCRM. [Online] [Citação: 21 de 11 de 2013.] <http://www.sugarcrm.com/case-studies>.
13. OpenCRX. [Online] [Citação: 25 de 11 de 2013.] <http://www.opencrx.org/project.htm#roadmap>.
14. Gartner Glossary. [Online] [Citação: 12 de 12 de 2013.] <http://www.gartner.com/it-glossary/total-cost-of-ownership-tco>.
15. **Bucholtz, Chris**. *CRM Total Cost of Ownership: Fees, Subscriptions and Hidden Costs*. s.l. : SugarCRM Whitepapers.
16. **Prats, Pedro Dowens**. Trend Watch Siebel CRM. *CRM Newsletter*. 2013.
17. **Parasnis, Carolyn Heller Baird and Gautam**. *From social media to Social CRM, What customers want*. s.l. : IBM Institute For Business Value.
18. —. *From social media to Social CRM, Reinventing the Customer Relationship*. s.l. : IBM Institute For Business Value.

19. *CMP Overview*. **Sousa, André**. Lisboa : s.n., 2013.
20. *Projeto CRM: Arquitetura da Solução (Formação Técnica)*. **Pereira, Miguel**. Luanda : s.n., 2012.
21. Oracle. [Online] [Citação: 8 de 10 de 2013.] <http://www.oracle.com/technetwork/testcontent/bslayer-095695.html>.
22. **Oliveira, André Guilherme Domingos de**. *Sistema para Validação da Correção e Completude de Encomendas*. 2012.
23. **Bruegge Bernd, Dutoit Allen H**. *Object-Oriented Software Engineering*.
24. **Sommerville**. *Software Engineering 9*.
25. **Sindhwani, Prem Melville and Vikas**. [RECOMMENDER SYSTEMS]. *IBM T.J. Watson Research Center*.
26. **Tuzhilin, Gediminas Adomavicius and Alexander**. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.
27. **Burke, Robin**. *Hybrid Web Recommender Systems*.
28. **Olmo, Félix Hernández del and Gaudioso, Elena**. *Evaluation of recommender systems: A new approach*. s.l. : Unidversidad Nacional de Educación a Distancia, Departamento de Inteligencia Artificial, Madrid, Spain.
29. Oracle. [Online] [Citação: 10 de 10 de 2013.] http://docs.oracle.com/cd/E13222_01/wls/docs61/intro/chap1.html.
30. *Projeto EAI/OM OM+ (Order Management)*. **Pereira, Miguel**. Luanda : s.n., 2012.

Anexos

Anexo A – Planeamento PERT

A tabela seguinte expõe detalhadamente os vários processos envolvidos na implementação do sistema descrito, refletindo o plano desenvolvido para esta fase.

O planeamento foi realizado de acordo com a técnica de PERT, apresentando uma previsão para a duração de cada tarefa (em dias). O cálculo da duração esperada faz-se de acordo com:

$$\text{Tempo Esperado} = \frac{\text{Otimista} + (4 \times \text{Provável}) + \text{Pessimista}}{6}$$

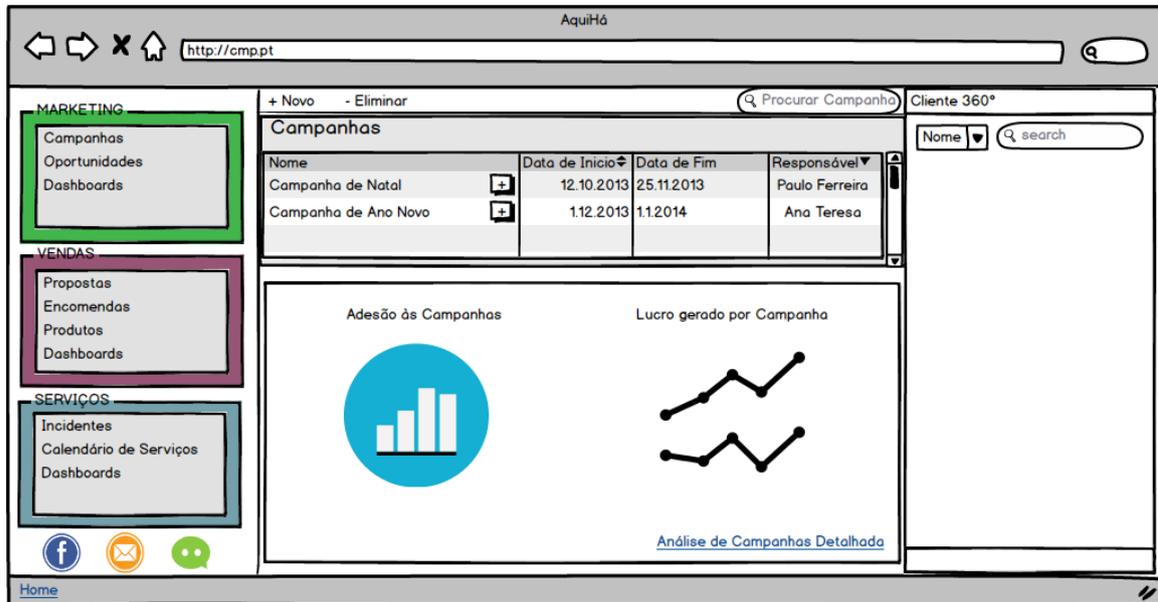
Desenvolvimento	Otimista	Provável	Pessimista	Esperado
Base e Modelo de Dados				
Alterações à Base de Dados	3	5	8	5,17
Alterações ao Modelo de Dados	7	8	10	8,17
Lógica e Portal				
Criação Ambiente Cliente-Servidor	4	7	11	7,17
Login no Sistema	2	3	5	3,17
Área Cliente 360	3	3	6	3,5
Gestão de Incidentes (CM+)				
Criação Manual de Incidentes	2	3	5	3,17
Criação de Interações Manual	2	3	5	3,17
Resposta a Interações Manual	2	3	5	3,17
Resposta a Interações Automática	3	4	6	4,17
Listar Incidentes	1	1	3	1,33
Pesquisar Incidente	1	1	2	1,17
Consultar Detalhes de Incidentes	1	1	3	1,33
Consultar Detalhes de Interações	1	1	3	1,33
Criação de Alertas de Incidentes/Interações	3	4	5	4,00
Notificação de Alertas de Incidentes/Interações	4	5	6	5,00
Consulta de Alertas de Incidentes/Interações	2	2	3	2,17
Mapeamento das Funcionalidades no Portal	4	6	9	6,17
Módulo de Integração com o Facebook				
Criação do Ambiente de Comunicação com o Facebook	4	5	8	5,33
Subscrição de Serviços	1	2	3	2,00
Subscrição de Notificações	2	4	5	3,83
Recepção de Notificações	3	4	5	4,00
Tratamento das Notificações Recebidas	2	4	6	4,00
Encaminhamento das Notificações para sistemas externos (CM+)	2	3	4	3,00
Recepção de Pedidos de Sistemas Externos (CM+)	1	1	2	1,17
Tratamento de Pedidos de Sistemas Externos (CM+)	2	3	4	3,00
Encaminhamentos de Pedidos Externos para o Facebook	3	4	5	4,00
Mapeamento das Funcionalidades no Portal	4	4	5	4,17
Sistema de Recomendação				

Criação da Base de Casos	1	2	4	2,17
Pré-Processamento de Casos	3	4	6	4,17
Indexação de Casos	1	1	2	1,17
Cálculo da Semelhança entre Casos	2	3	4	3,00
Votação de Casos	2	3	5	3,17
Lista de Recomendações para um Incidente	1	1	2	1,17
Mapeamento das Funcionalidades no Portal	3	4	6	4,17
TOTAL	82	112	171	116,83

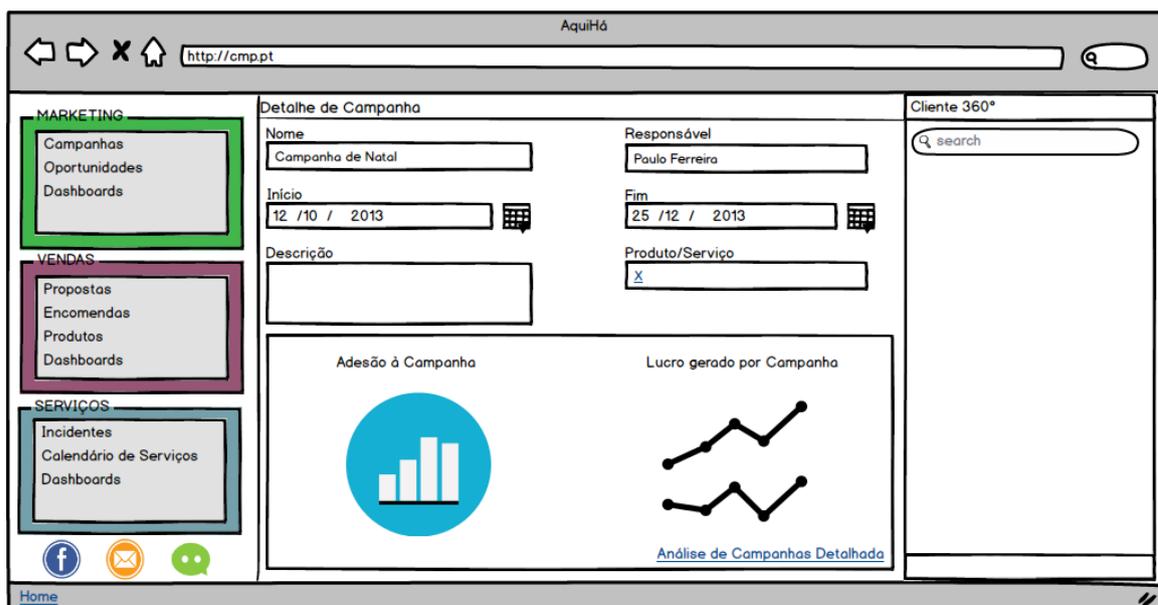
Anexo B – Prototipagem Total do Sistema

Campanhas de Marketing

Apresenta as Campanhas a decorrer ou já decorridas na empresa.

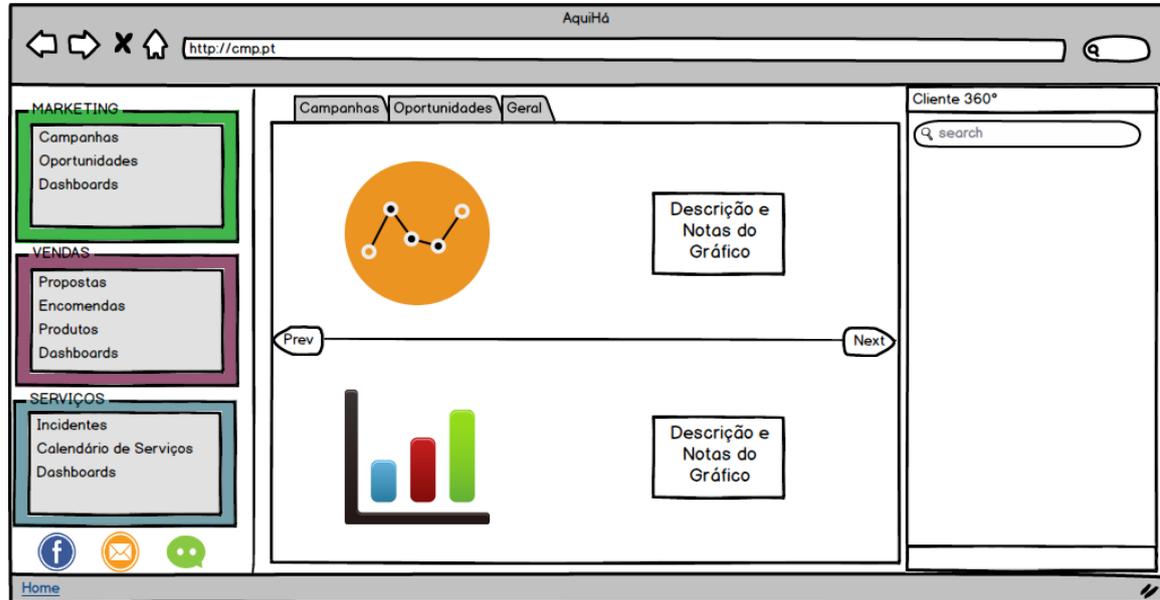


Detalhe de Campanha de Marketing



Dashboards de Marketing

Apresenta dados e resultados relativos às Campanhas de *Marketing*.



Detalhe de Oportnidades de Marketing

The screenshot shows the same web browser window with the URL `http://cmp.pt`. The page title is "AquiHá". The interface is divided into several sections:

- MARKETING**: Contains links for "Campanhas", "Oportunidades", and "Dashboards".
- VENDAS**: Contains links for "Propostas", "Encomendas", "Produtos", and "Dashboards".
- SERVIÇOS**: Contains links for "Incidentes", "Calendário de Serviços", and "Dashboards".
- Detalhe de Oportunidade**: A main content area with the following fields:
 - Tópico**: Stock Produto X
 - Data de Fecho**: 12 /12 / 2013
 - Cliente**: Paulo Ferreira
 - Produto/Serviço**: X
 - Probabilidade**: 80%
 - Concretizada
 - Responsável**: Ana Teresa
 - Descrição**: (Empty text area)
 - Interesse**: Muito Interessante
 - Lucro Potencial**: 11.000
- Cliente 360***: A sidebar on the right with a search bar labeled "search".
- Done**: A button at the bottom right.
- Home**: A button at the bottom left.

Encomendas

Lista as Encomendas realizadas e alguns dos seus detalhes.

The screenshot shows a web browser window with the URL `http://cmp.pt`. The page title is "AquiHá". The main content area displays a table of orders under the heading "Encomendas". The table has columns for "Produto", "Quantidade", "Data E.", "Receita", "Cliente", "Classificação", and "Fechada". There are two rows of data:

Produto	Quantidade	Data E.	Receita	Cliente	Classificação	Fechada
M	18	12.8.2011	11.000	Paulo Ferreira	MI	<input checked="" type="checkbox"/>
Y	36	1.12.2013	5.000	Ana Teresa	I	<input type="checkbox"/>

On the left side, there is a navigation menu with three sections: "MARKETING" (Campanhas, Oportunidades, Dashboards), "VENDAS" (Propostas, Encomendas, Produtos, Dashboards), and "SERVIÇOS" (Incidentes, Calendário de Serviços, Dashboards). At the top right, there is a search bar labeled "Procurar Campanha" and a client identifier "Cliente 360°".

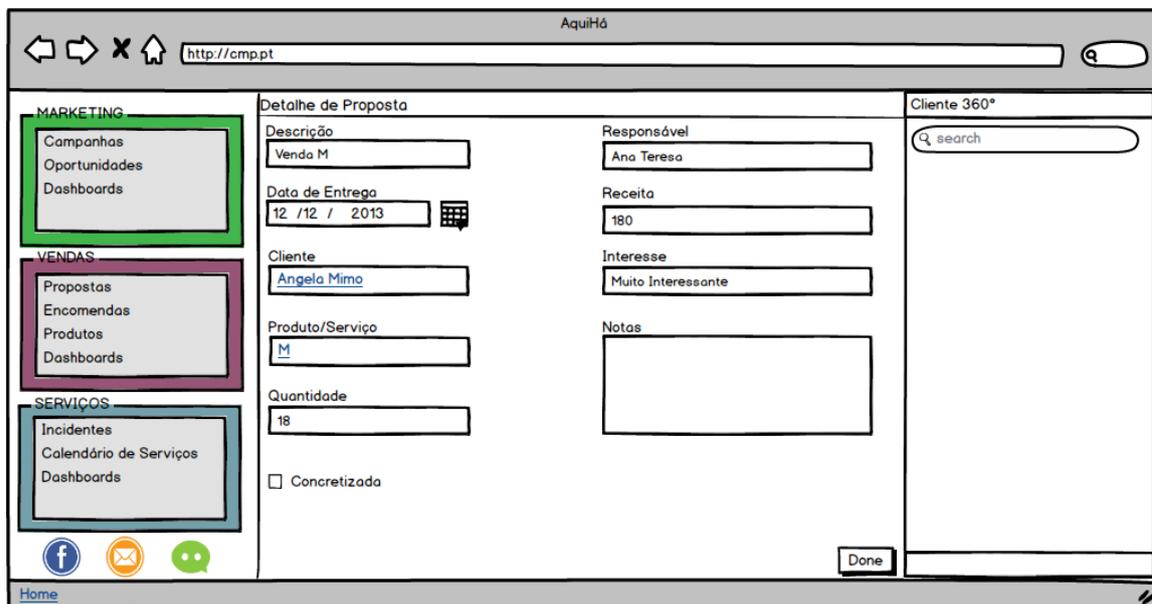
Detalhe de Produto

The screenshot shows the same web browser window, but now displaying the "Detalhe de Produto" page. The left navigation menu is identical to the previous screenshot. The main content area is divided into several sections:

- Nome:** M
- Responsável:** Ana Teresa
- Data Inicio Comercialização:** 12 / 12 / 2013
- Preço de Venda:** 10
- Stock:** 100
- Aparência:** An image of a blue armchair.
- Características:** A list of radio buttons with labels: "option 1 (selected)", "option 2", "option 3 (indeterminate)", "option 4 (disabled)", and "option 5 (disabled and selected)". An "Add" button is at the bottom right.
- Componentes:** A list of radio buttons with labels: "option 1 (selected)", "option 2", "option 3 (indeterminate)", "option 4 (disabled)", and "option 5 (disabled and selected)". An "Add" button is at the bottom right.
- Campanhas:** A list of links: "Campanha de Natal", "Campanha Namorados", and "Campanha RUX".

At the top right, there is a search bar and a client identifier "Cliente 360°".

Detalhe de Proposta



Anexo C – Descrição Detalhada do Modelo de Dados do CM+

Customer

Um *Customer* (cliente) refere-se a um comprador ou utilizador dos produtos da empresa. Em certos casos esta entidade pode também referir-se a clientes de empresas semelhantes (19).

Um cliente pode ser univocamente identificado através de um ou mais documentos de identificação.

De uma forma genérica um cliente pode ser um de dois tipos: Particular ou Empresarial. Em termos de ciclo de vida, a entidade cliente passa pelos estágios representados na figura 2.4.

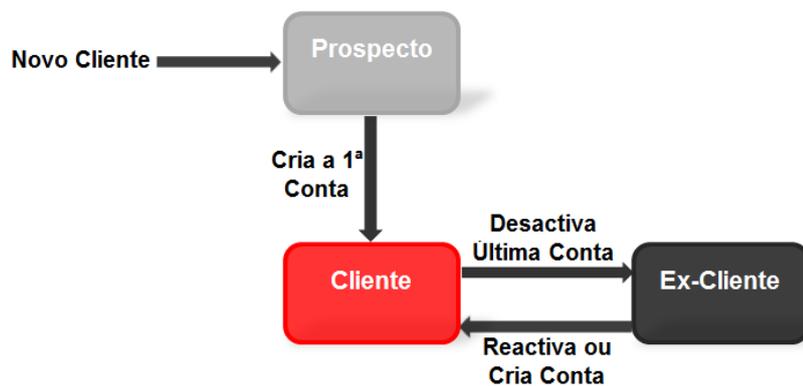


Figura 9.1 – Ciclo de vida da Entidade Cliente

Account

Uma *Account* (Conta) representa um conjunto de bens ou serviços passíveis de serem adquiridos ou usados por um cliente e pelos quais ele vai ser cobrado (19).

Em termos relacionais, uma conta pertence sempre a um e um só cliente. Já um cliente pode ter várias contas associadas.

Uma conta é identificada univocamente através de um número de identificação gerado automaticamente por um sistema de facturação (no caso analisado, o BSCS).

Geralmente existe nos sistemas de facturação uma entidade correspondente à conta, à qual ela está associada por via da definição de uma chave de integração. A figura 2.5 representa esta relação entre *Customer* e *Account*.

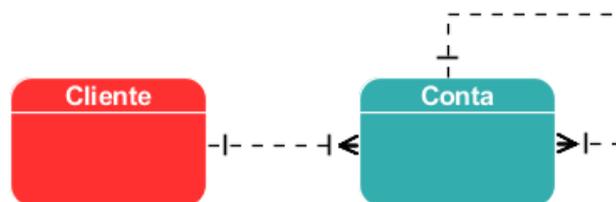


Figura 1.2 – Relação entre Cliente e Conta definida pelas regras de negócio

Em alguns sistemas os conceitos de *Billing* e de Conta de Serviço são representados como entidades *stand-alone*. No entanto, no caso do CM+, a entidade *Account* funde esses dois conceitos gerando uma relação como a esquematizada na figura 2.6. Assim, o ciclo de

vida da entidade *Account* e o seu estado ao longo do tempo vai depender dos estados de cobrança ou serviços que lhe estão associados.

Esta entidade mapeia ainda o conceito de *Dealer Account*. Uma *Dealer Account* representa o perfil de um *Dealer*, uma pessoa ou empresa que vende produtos e/ou serviços através de operadora de telecomunicações.

Os *Dealers* estão tipicamente associados à companhia através de um determinado contrato e em algumas circunstâncias têm limites geográficos de actuação.

No cenário particular em que um *Dealer* é também um cliente dos serviços da operadora, as contas de facturação e de *Dealer* podem ser associadas ao mesmo cliente.

É importante referir que as *Dealer Accounts* não são geridas pelo CM+. Elas são criadas e actualizadas por um sistema de ERP (como o SAP) e depois, através de um mecanismo de replicação implementado pelo sistema de integração (TIBCO), são visíveis do lado do CM+.

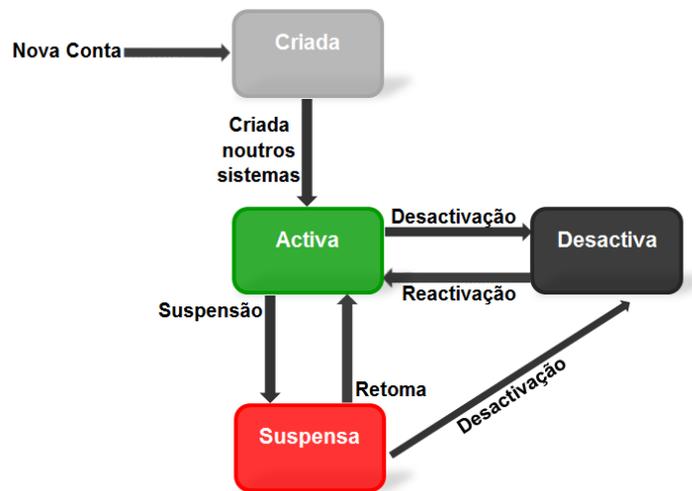


Figura 1.3 – Ciclo de Vida da Entidade Conta influenciado pelo sistema de *Billing* e pelos Serviços associados

Address

Um *Address* (endereço) pode representar ou um endereço físico do cliente ou o seu endereço electrónico (19).

Um contacto pode estar associado a zero ou mais endereços, sendo que em certos casos podem ser definidas regras que forçam que haja contactos que tenham de ter obrigatoriamente um endereço associado (em casos de faturação, por exemplo).

Um endereço só pode ser criado quando associado a um determinado contacto.

Contact

Um Contacto pode representar ou uma pessoa ou uma empresa. Um contacto está sempre associado a um cliente, sendo que um cliente apesar de poder ter vários contactos associados tem de ter sempre um seleccionado como contacto principal (19).

Um contacto pode também estar associado a uma conta. Neste caso uma conta pode ter vários contactos associados mas, tal como na relação com o cliente, um desses contactos tem de ser seleccionado como contacto principal de cobrança.

Um contacto pode ou não estar associado a um ou mais endereços. Apesar de não ser à partida uma relação obrigatória, pode haver casos em que as regras de negócio exijam essa relação.

Customer Document

Um Documento de Cliente representa o conjunto de informação que pode identificar univocamente esse mesmo cliente, sendo que não podem existir dois clientes com o mesmo documento. Um cliente tem de ter obrigatoriamente, no mínimo, um Documento de Cliente (19).

Esta entidade existe unicamente nos contextos de cliente e contacto.

Interaction

Uma interação cria uma instância de comunicação entre a entidade Cliente (ou Conta) e a operadora, assim como instancia o seu objetivo e descrição (19).

Esta entidade pode ser utilizada para o registo de eventos tais como uma chamada para o *call center*, uma ida à loja, um *email*, um SMS, etc.

É possível a criação de sub-interacções para ligar ações executadas no âmbito de uma interação.

Dependendo do cenário, uma interação pode ser criada automaticamente, semi-automaticamente ou manualmente.

Esta entidade não tem ciclo de vida, uma vez que é criada não pode ser alterada ou removida.

Uma interação está sempre associada a um cliente, a uma conta ou a um contacto e pode ter associada uma ou mais interações agregadas.

Note

A Nota é usada para adicionar informações que se considerem relevantes para descrever uma entidade ou suas características (19). Pode ser pública ou privada.

Task

Uma tarefa corresponde a uma actividade que uma pessoa deve realizar, pode também representar um *email* que exige resposta manual (19).

Attachment

Um anexo pode ser utilizado para guardar uma referência para um ficheiro externo. Podem ser armazenados no CM+ como dados binários ou pode ser criado um *link* para um sistema de gestão de documentos externo (19).

Queue

Representa um conjunto de trabalhos pendentes que devem ser realizados por uma dada pessoa. Quando uma tarefa é criada é associada a uma *queue* pelo CM+ (19).

Agent

Consiste na pessoa ou equipa que faz a gestão e monitorização da fila. A relação entre todos estes conceitos pode ser esquematizada como mostra a figura 2.7 (19).

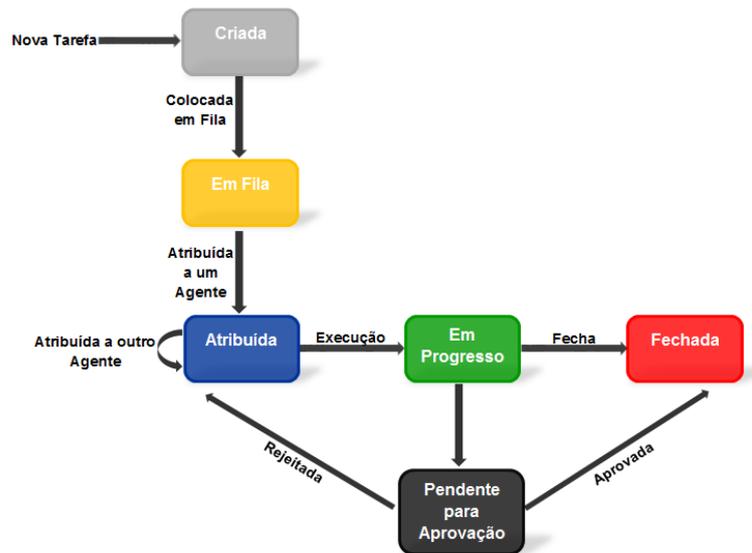


Figura 1.4 – Progresso da entidade *Task* ao longo do tempo

Product

Um produto representa, no contexto das telecomunicações, um bem ou serviço que é necessário entregar para que o cliente possa usar um certo serviço (19). A representação de produtos feita no CM+ é mínima, sendo que as únicas funcionalidades implementadas no CM+ relativamente a produtos é a sua criação nas tabelas *Prod* e *Prod_Assoc* e a sua associação aos *Assets* correspondentes. Todas as restantes funcionalidades inerentes à gestão de produtos são implementadas pelo *Tribold*.

Asset

Um ativo representa a instanciação de um produto, sendo que um produto pode ter vários *Assets* associados (19). Por exemplo, considerando numa operadora um produto combinado de ‘net+voz+tv’, *assets* possíveis seriam o *router*, o telefone, a box, etc. Os *assets* definidos podem ser: *Service Contract*, *Service Asset* ou *Account Asset*.

Um *Service Contract* representa um conjunto de bens e serviços para serem entregues e usados juntos, tal como um serviço de voz móvel. Este *asset* tem um ciclo de vida gerido de acordo com regras de negócio e, em alguns casos, pode ser conduzido pelo ciclo de vida da Conta a que está associado.

Um *Service Asset* representa um serviço que é necessário entregar ou usar no âmbito de um determinado *Service Contract*. Pode corresponder a um elemento físico (como um cartão SIM) ou um recurso (como o MSISDN), um tarifário ou até mesmo uma campanha ou desconto. O seu ciclo de vida é construído em função do ciclo de vida do *Service Contract* a que está associado. No entanto pode ser ativado ou desativado independentemente dele.

Por fim, um *Account Asset* corresponde a um serviço que é necessário entregar ou cobrar ao longo de todos os *Service Contracts* geridos por uma determinada conta. O seu ciclo de vida é conduzido pelo ciclo de vida da conta mas, tal como o anterior, pode ser ativado ou desativado de forma independente.

Attribute

Corresponde a uma característica específica de um ativo de serviço ou de conta. Por exemplo, caso se pretenda anotar a cor do objeto, o MSISDN, o número de série, etc., é

através desta entidade (19). O CM+ pode associar os atributos aos *assets* mas é o Tribold que tem a função de definir quais atributos são necessários para cada produto do catálogo.

Para melhor compreensão, a figura 2.8 esquematiza a relação entre estas entidades.

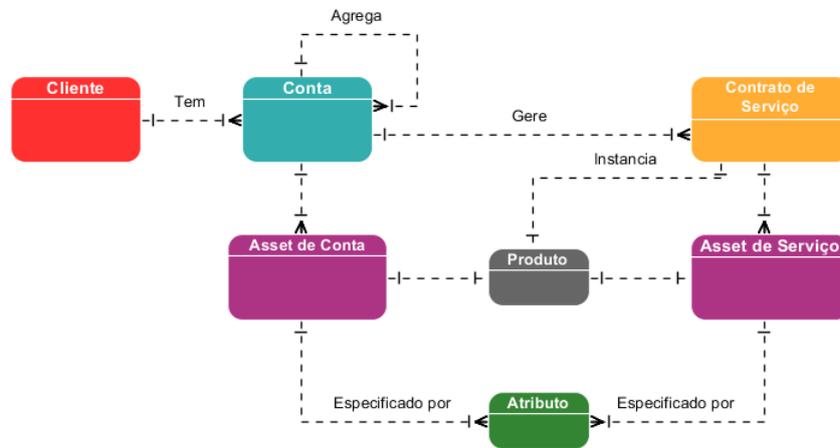


Figura 1.5 – Relação entre Clientes, Contas, Produtos e Activos

Order

Uma ordem representa o pedido para criação ou modificação de um conjunto de *Assets* e atributos correspondentes (19). O pedido que origina a criação e configuração de uma ordem pode ser iniciado por um cliente (como por exemplo a suspensão ou alteração de um serviço) mas pode também ser um pedido gerado automaticamente em função do ciclo de vida de uma entidade.

Uma ordem compõe-se por um conjunto de Itens que indicam o tipo de ação que deve ocorrer. Esses itens podem ser criados a partir de um conjunto pré-existente de *assets* (à exceção de casos em que se pretende ativar serviços). É também possível incluir numa ordem itens que não são localizados como *assets*.

Quando o CM+ recebe uma ordem proveniente do PA, tem de executar um determinado conjunto de validações de forma a garantir que a ordem não põe em causa a integridade do sistema. Se esse processo decorrer com sucesso, o CM+ encaminha a ordem para um sistema de gestão de ordens, como o OM+, que se responsabilizará pela decomposição, sequenciamento e execução da ordem.

Através de uma ordem é possível consultar as acções realizadas no âmbito do pedido para criar ou modificar entidades. Uma ordem consiste em um ou mais *Root Items*.

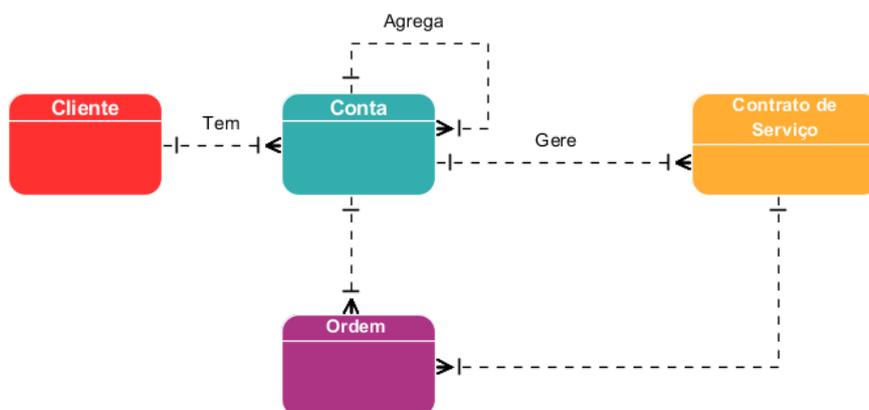


Figura 1.6 – Relação entre entidades

Root Item

Um *Root Item* baseia-se, geralmente, num *Service Contract* existente (quando o objetivo é modificar a configuração dele) ou nas configurações do produto instanciado por um *Service Contract* (quando se pretende criar ou ativar um novo *Service Contract*) (19).

Child Item

Um *Root Item* é composto por um conjunto de *Child Items*. Estes baseiam-se em *Service Assets* (para representar modificações) ou na configuração de qualquer produto instanciado pelo *Root Item* (quando o objetivo é ativar um novo *Service Asset* num dado *Service Contract*) (19).

No caso das modificações, o *Service Asset*, que está a ser modificado, é associado a um *Child Item* para que, quando a ordem for concluída pelo OM+, o CM+ possa fazer a atualização das instâncias do *Service Asset*, efetivando a sua nova configuração.

Item Attribute

Um *Item Attribute* representa uma característica específica de um *Root Item* ou *Child Item* (19). O número de atributos que podem ser configurados é definido pelo produto associado a esse item.

Esta entidade é usada para configurar a cor de um acessório, o número de série de um cartão SIM, entre outras características.

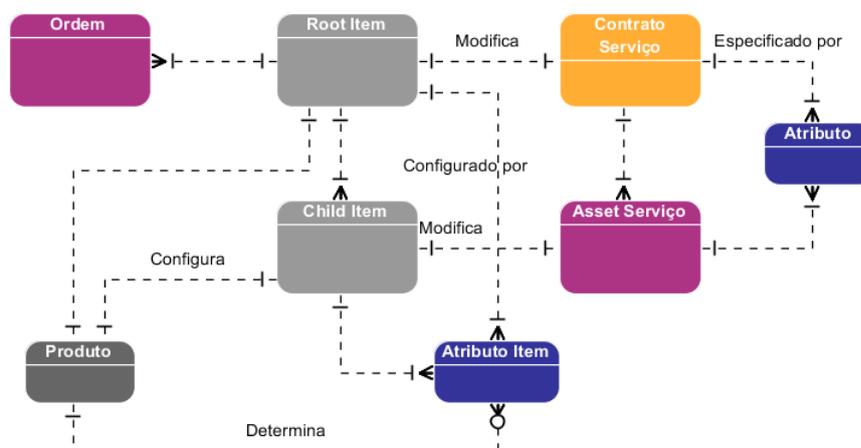


Figura 1.7 – Relação entre entidades relacionadas com a entidade Ordem

Anexo D –Portal de Atendimento e OM+

O presente anexo visa o detalhe e descrição dos sistemas Portal de Atendimento e O+;+, presentes no sistema de informação analisado.

Portal de Atendimento (PA)

Arquitetura e Desenvolvimento

O Portal de Atendimento (PA), muitas vezes referido como UFE (*Unified Front-End*) funciona como camada intermédia entre o utilizador final e os sistemas da empresa, tais como CRM, ERP, etc. O seu desenvolvimento baseia-se no recurso ao *Weblogic Server* da Oracle, “uma plataforma para desenvolvimento e entrega de aplicações empresariais multicamada” (29).

De acordo com os princípios definidos pelo *Weblogic Server*, a arquitetura da aplicação deve estar dividida em 3 componentes principais:

- **Cliente:** contém os programas executados pelo utilizador (tais como o browser);
- **Intermédia:** contém o servidor *Weblogic*;
- **Backend:** relativa aos sistemas da empresa, tais como bases de dados, CRM, ERP, etc.

Assim, e de acordo com esses princípios, a arquitetura do PA pode representar-se como mostra a figura 2.14.

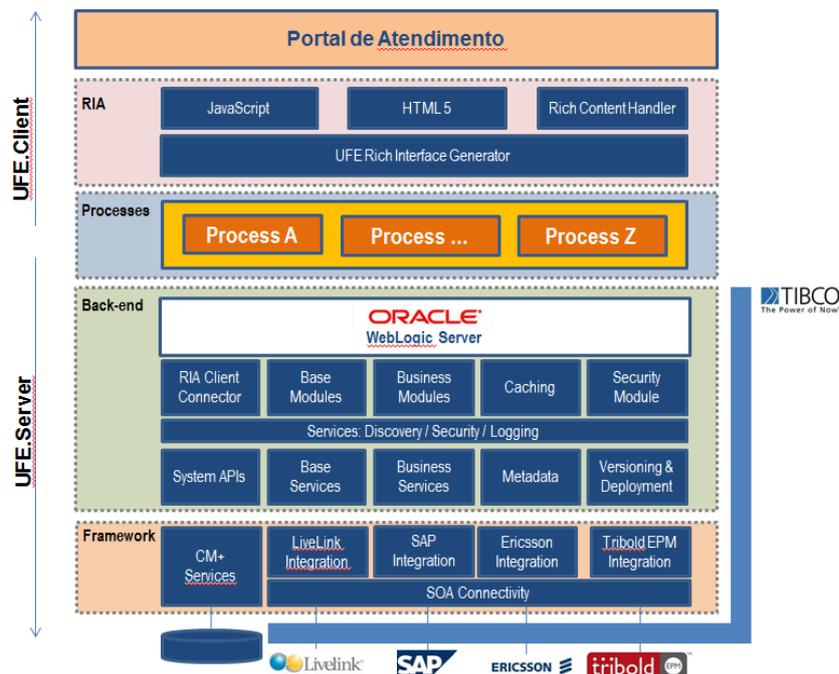


Figura 1.8 – Arquitetura do Portal de Atendimento desenvolvido no caso em estudo

Nos níveis superiores encontra-se a camada de **cliente**. Os programas aqui executados foram implementados com recurso à *framework* **EXTjs**, biblioteca *Javascript* que disponibiliza ferramentas para criar interfaces para páginas e sistemas *web-based* garantindo performance e possibilitando elevado nível de customização.

Através de *standard interfaces* é estabelecida a comunicação com o *Weblogic Server* que se encontra na camada **intermédia** (Backend). A este nível faz-se a mediação da comunicação,

pedidos e respostas entre o cliente final e os sistemas da empresa. Tal como qualquer outro servidor *web*, o *browser* pode executar pedidos ao *Weblogic* usando o protocolo http. As páginas *web* são produzidas através de servlets e JSP's.

Por fim, a última camada, **Framework**, corresponde aos restantes sistemas da empresa. Desde CRM, SAP, LiveLink, até o Tribold, etc. Estes sistemas são responsáveis pelo acesso às bases de dados e pela atribuição de lógica à aplicação. Este último nível corresponde às componentes críticas e cruciais do negócio que como tal devem estar protegidas e devem ter acesso controlado e limitado. O servidor *Weblogic* possibilita esse acesso seguro e controlado fundamental para este tipo de situações.

O PA encontra-se posicionado na camada de Negócio, comunicando com dois sistemas de *backend*: O CM+ e o EAI. A comunicação com o CM+ corresponde à implementação das funcionalidades de gestão de clientes. Os restantes pedidos, como *Billing*, Rede, ERP, gestão de produtos, etc. são encaminhados para o EAI que posteriormente se responsabilizará pelo seu mapeamento para os sistemas correspondentes.

OM+

O OM+ é um componente independente, responsável pela cadeia de Gestão de Ordens. A este nível executam-se funcionalidades como a recepção de pedidos, o seu sequenciamento, mapeamento, gestão de erros entre outros (30).

O OM+ recebe ordens pré-validadas, usando regras de negócio, por um dado sistema, como o CM+. Depois prossegue para a sua decomposição, mapeamento, sequenciamento e gere a concretização de todos os pedidos técnicos em cada um dos sistemas de backend intervenientes na ordem.

Para além disso, o OM+ faz a gestão de erros e, sempre que ocorre um erro na execução de uma dada ordem, propaga as consequências para os sistemas devidos.

O OM+ possibilita o controlo e acompanhamento dos serviços de ordens bem como coloca à disposição um interface para controlo de operações.

Este sistema recebe e processa dois tipos de ordens: "Asset-based" e "Not-asset-based". O primeiro tipo é geralmente proveniente do CM+ e possibilita a gestão e configuração de contas/serviços, criar ordens para novos produtos/serviços baseando-se em *assets* pré-existentes, visibilidade para ciclo de vida de assets durante interações com clientes, entre outros. As ordens "Not-asset-based" são todas aquelas que não envolvem a gestão de entidades do CM+ (30).

Independentemente do tipo de ordem, todas elas são de natureza assíncrona e o OM+ vai usar os mesmos ciclos de vida e *workflow* para as tratar.

Anexo E – Testes de Sistema

Para a validação das funcionalidades do sistema desenvolvido, que permitem a confirmação que este corresponde aos requisitos funcionais estipulados, foram realizados testes de sistema que se apresentam em seguida.

ID – Teste	1
Prioridade	Elevada
Requisito	Criar Incidente Manualmente
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes
Dados de Entrada	Identificação do cliente que reportou o incidente Canal de Comunicação Descrição do Problema e/ou Produto
Resultado Esperado	O incidente deve ser criado na BD Deve ser criada a primeira interação do incidente O utilizador deve ser encaminhado para a página de detalhes do incidente
Estado	✓
Detalhes	-

ID – Teste	2
Prioridade	Elevada
Requisito	Criar Interação Manualmente
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes Acesso a Detalhes de um Incidente
Dados de Entrada	Canal de Comunicação Descrição do Problema Descrição da Resposta (Opcional)
Resultado Esperado	Deve ser criada a interação na BD A interação deve ser associada ao incidente correspondente O utilizador deve permanecer na página de detalhes de incidente, podendo visualizar a nova interação
Estado	✓
Detalhes	-

ID – Teste	3
Prioridade	Baixa
Requisito	Pesquisar Incidente
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes Acesso à Área de Pesquisa
Dados de Entrada	ID do Cliente (Pesquisado na respetiva área) ou ID do Incidente
Resultado Esperado	O sistema deve encaminhar o utilizador para a página do incidente pesquisado ou Listar os incidentes que correspondem à pesquisa
Estado	✓
Detalhes	O sistema apenas responde como esperado caso o resultado seja apenas um incidente

ID – Teste	4
Prioridade	Elevada

Requisito	Listar Incidentes
Condições	<i>Login</i> no sistema efetuado com sucesso
Dados de Entrada	Acesso à Área de Gestão de Incidentes
Resultado Esperado	Deve ser apresentada a lista de incidentes a cargo do operador
Estado	✓
Detalhes	-

ID – Teste	5
Prioridade	Elevada
Requisito	Consultar Detalhes de Incidente
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes
Dados de Entrada	Seleção do Incidente a Consultar
Resultado Esperado	O utilizador deve ser encaminhado para a página de detalhes de incidente, vendo uma listagem das interações ocorridas
Estado	✓
Detalhes	-

ID – Teste	6
Prioridade	Média
Requisito	Consultar Detalhes de Interação
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes Selecionar o incidente pretendido
Dados de Entrada	Selecionar interação pretendida
Resultado Esperado	O utilizador deve visualizar os detalhes da interação pretendida
Estado	✓
Detalhes	-

ID – Teste	7
Prioridade	Elevada
Requisito	Resposta Manual a Interação
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes Selecionar o incidente pretendido
Dados de Entrada	Selecionar interação pretendida Introduzir resposta Selecionar canal de resposta
Resultado Esperado	O sistema deve validar submeter a resposta à interação na BD e caso assinalado, deve dar como encerrado o incidente O utilizador deve ser encaminhado para a página de detalhes de incidente (Caso o canal seja o Facebook, ver teste a Resposta Manual a Interação)
Estado	✓
Detalhes	-

ID – Teste	8
Prioridade	Elevada
Requisito	Associar Interação a Incidente existente
Condições	<i>Login</i> no sistema efetuado com sucesso Acesso à Área de Gestão de Incidentes Selecionar o incidente pretendido

Dados de Entrada	Criar nova interação Colocar descrição e canal
Resultado Esperado	O sistema deve criar uma nova interação e associá-la ao incidente em contexto
Estado	✓
Detalhes	-

ID – Teste	9
Prioridade	Média
Requisito	Receber Notificações no Portal
Condições	<i>Login</i> no sistema efetuado com sucesso
Dados de Entrada	
Resultado Esperado	Recepção de uma notificação provenientes do CM+
Estado	✓
Detalhes	-

ID – Teste	10
Prioridade	Elevada
Requisito	Receber no CM+ Notificações do Facebook
Condições	Servidor do CM+ ligado
Dados de Entrada	Fazer publicação, comentário ou enviar mensagem na/para a página de Facebook pública
Resultado Esperado	O CM+ deve receber notificação, processar e, em caso disso, criar novo incidente/interação Caso seja um incidente deve ser gerada a notificação para o portal
Estado	✓
Detalhes	-

ID – Teste	11
Prioridade	Elevada
Requisito	Filtrar incidentes de outro tipo de publicações
Condições	Servidor do CM+ ligado
Dados de Entrada	Fazer publicação, comentário ou enviar mensagem na/para a página de Facebook pública #Cenário 1: Fazer publicação de incidente, recorrendo ao uso da <i>hashtag</i> “#Incident” (ou “#Problem” ou “#Doubt” ou palavras similares a estas em termos de escrita, como “#Doubt”) #Cenário 2: Fazer publicação com <i>hashtags</i> erradas ou sem <i>hashtags</i>
Resultado Esperado	#Cenário 1: O sistema deve identificar que se trata de um incidente e como tal deve criá-lo na BD com os dados corretos #Cenário 2: O sistema deve identificar que não se trata de um incidente e como tal deve descartar e não criar nada na BD nem gerar notificações.
Estado	✓
Detalhes	-

ID – Teste	12
Prioridade	Média
Requisito	Identificar Produto em Publicação
Condições	Servidor do CM+ ligado

Dados de Entrada	<p>Fazer publicação, comentário ou enviar mensagem na/para a página de Facebook pública</p> <p>#Cenário 1: Fazer publicação de incidente, recorrendo ao uso da <i>hashtag</i> “#Incident” (ou “#Problem” ou “#Doubt” ou palavras similares a estas em termos de escrita, como “#Doubt”), colocando <i>hashtag</i> com produto existente na BD, por exemplo “PhoneX” (ou palavras semelhantes)</p> <p>#Cenário 2: Fazer publicação com <i>hashtags</i> de incidente mas sem referência a produto</p>
Resultado Esperado	<p>#Cenário 1: O sistema deve identificar que se trata de um produto conhecido e como tal deve registá-lo no incidente como sendo esse mesmo produto</p> <p>#Cenário 2: O sistema não deve identificar produto e na BD o incidente não deve ter nenhum produto associado.</p>
Estado	✓
Detalhes	-

ID – Teste	13
Prioridade	Elevada
Requisito	Fazer, através do CM+, publicação, comentar ou mensagem no Facebook
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Área de resposta a interação
Dados de Entrada	Texto de resposta
Resultado Esperado	Deve ser feita automaticamente uma publicação no Facebook em resposta à interação do cliente
Estado	✓
Detalhes	-

ID – Teste	14
Prioridade	Elevada
Requisito	Apresentar Recomendações para Incidente
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Detalhe de Incidente
Dados de Entrada	-
Resultado Esperado	Deve ser apresentada uma listagem de incidentes semelhantes e respetivas soluções
Estado	✓
Detalhes	-

ID – Teste	15
Prioridade	Média
Requisito	Consultar Detalhes de Incidente Recomendado
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Detalhe de Incidente Área de Recomendações de Incidente
Dados de Entrada	Seleccionar o Incidente
Resultado Esperado	Devem ser apresentados os detalhes do incidente seleccionado

Estado	✓
Detalhes	-

<i>ID – Teste</i>	<i>16</i>
Prioridade	Média
Requisito	Votar de Incidente Recomendado
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Detalhe de Incidente Área de Recomendações de Incidente
Dados de Entrada	Votar Incidente
Resultado Esperado	O número de votos do incidente na base de casos deve ser atualizado
Estado	✓
Detalhes	-

<i>ID – Teste</i>	<i>17</i>
Prioridade	Baixa
Requisito	Pesquisar Cliente (Área 360)
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Área Cliente 360
Dados de Entrada	Inserir um dos dados do cliente
Resultado Esperado	Deve ser apresentada a listagem de clientes correspondente aos critérios estipulados
Estado	✓
Detalhes	-

<i>ID – Teste</i>	<i>18</i>
Prioridade	Baixa
Requisito	Consultar Detalhes de Cliente
Condições	Servidor do CM+ ligado <i>Login</i> no sistema efetuado com sucesso Área Cliente 360 Pesquisar Cliente
Dados de Entrada	Selecionar cliente pretendido
Resultado Esperado	Devem ser apresentados os detalhes do cliente selecionado
Estado	✓
Detalhes	-

Anexo F – Testes de *Performance* (Sistema de Recomendação)

A tabela seguinte apresenta os resultados dos testes de carga realizados ao Sistema de Recomendação implementados.

Como visto ao longo do documento este sistema está exposto aos exterior através de *Web services*. Como tal, para simular carga no módulo optou-se por recorrer ao *software* SOAP UI. Este disponibiliza mecanismos de testes de carga a *web services*, monitorizando os pedidos feitos, o tempo consumido por cada pedido, o tempo máximo e mínimo ao longo da execução dos pedidos, a média e ainda o número de transações por segundo (TPS).

Foram considerados 9 cenários diferentes em função do número de Utilizadores Virtuais (UV's). Para cada um desses cenários foram executados 7 testes variando o número de pedidos entre 50 e 5000.

Para a execução de todos os testes foi definido um *delay* entre pedidos de 3000ms.

Os valores obtidos encontram-se sistematizados na tabela seguinte. Verificou-se que o número médio de transações por segundo (TPS) é de 17.1 e que na execução de todos os testes ocorreram apenas 616 falhas face aos 67900 testes realizados.

As falhas ocorridas foram todas falhas de conectividade ao serviço e 80.84% delas ocorreram para um número de utilizadores igual a 180.

UV's	Nº de Pedidos	TPS	Falhas	Min. (ms)	Máx. (ms)	Média (ms)
20	50	6.52	0	68	420	145.48
	200	7.91	0	66	922	155.42
	400	7.96	0	61	967	119.19
	600	8.19	0	55	939	123.28
	800	8.43	0	47	452	93.09
	1000	8.11	0	59	1240	155.71
	5000	8.33	0	53	5228	113.87
30	50	8.38	0	70	1101	314.22
	200	11.01	0	66	810	203.75
	400	11.37	0	57	1795	241.46
	600	12.11	0	50	1168	135.76
	800	12.04	0	56	1413	181.58
	1000	11.76	0	50	2278	250.75
	5000	11.74	0	48	3150	278.01
40	50	9.63	0	75	2269	830.16
	200	12.56	0	67	1064	538.37
	400	14.54	0	45	2952	259.64
	600	13.68	0	60	3854	492
	800	13.79	0	61	3915	475.3
	1000	13.26	0	61	6449	509.05
	5000	14.82	0	48	6420	273.77
50	50	9.77	0	78	2565	1116.46
	200	14.93	0	58	3782	662.11
	400	13.32	0	61	7707	1114.55

	600	13.92	0	56	7198	1053.32
	800	14.22	0	50	8981	1061.36
	1000	13.08	0	71	7066	1333.71
	5000	12.80	0	48	8897	1035.96
100						
	100	11.81	0	92	4085	2066.55
	200	12.47	0	71	12289	3781.02
	400	14.19	0	89	12239	3542.89
	600	15.59	0	57	7962	3569.34
	800	13.66	0	91	15053	4477.89
	1000	14.03	0	66	25901	4309.91
	5000	14.74	0	40	21226	3836.58
120						
	120	10.56	0	110	7019	3656.2
	200	11.85	0	100	13211	4913.91
	400	12.79	0	120	11987	5936.82
	600	12.64	0	77	17624	5982.12
	800	13.1	0	72	1602	5980.02
	1000	13.18	0	66	16083	5227.79
	5000	14.89	1	50	30840	4258.76
140						
	140	11.06	0	147	9877	5652.42
	200	12.08	0	143	14618	7521.11
	400	12.67	0	150	27626	8152.1
	600	16.47	0	72	13831	520028
	800	12.85	0	82	26884	7557.56
	1000	14.67	0	53	20960	5784.93
	5000	14.44	27	57	21721	6378.14
160						
	160	11.68	0	105	11428	7934
	200	12.59	0	180	13929	7255.75
	400	13.08	0	105	17617	7316.81
	600	13.68	0	72	27100	7327.89
	800	13.16	0	75	22163	7417.13
	1000	16.01	0	62	20821	6380.23
	5000	16.32	90	16	39021	6533.45
180						
	180	12.21	15	196	12545	77717.03
	200	14.21	68	185	11718	5213.66
	400	14.89	54	171	13449	7131.06
	600	14.60	56	107	23265	7663.79
	800	16.48	43	113	38280	7567.70
	1000	17.1	262	164	50136	7266.59
TOTAL	67900	17.1	616	16	50136	77717.03

Anexo G – Testes de Qualidade (Sistema de Recomendação)

Para validar e avaliar a qualidade dos resultados gerados pelo sistema de recomendação, bem como para garantir que gera resultados relevantes para o contexto, realizaram-se testes de qualidade ao sistema de recomendação.

Esses testes, descritos no capítulo de validação, consistem em pedidos de recomendação ao *web service* do módulo e na comparação dos resultados obtidos com os obtidos para a mesma pesquisa no *site StackOverflow*.

Para cada um dos casos testados são calculadas a *Recall* e a *Precisão*. A tabela seguintes agrega os resultados obtidos para os 31 casos de teste.

ID	R	NR	NRE	Recall	Precisão
1	7	1	3	88%	70%
2	7	0	3	100%	70%
3	7	0	3	100%	70%
4	7	0	3	100%	70%
5	4	3	6	57%	40%
6	7	0	3	100%	70%
7	8	0	2	100%	80%
8	7	0	3	100%	70%
9	5	1	5	83%	50%
10	7	0	3	100%	70%
11	7	0	3	100%	70%
12	5	0	5	100%	50%
13	5	0	5	100%	50%
14	6	0	4	100%	60%
15	4	1	6	80%	40%
16	6	0	4	100%	60%
17	6	0	4	100%	60%
18	8	0	2	100%	80%
19	5	1	5	83%	50%
20	5	0	5	100%	50%
21	5	2	5	71%	50%
22	6	0	4	100%	60%
23	6	0	4	100%	60%
24	5	0	5	100%	50%
25	5	0	5	100%	50%
26	6	0	4	100%	60%
27	10	0	0	100%	100%
28	9	1	1	90%	90%
29	6	0	4	100%	60%
30	5	1	5	83%	50%
31	6	0	4	100%	60%
TOTAL	192	11	118	95%	62%

Anexo H – Testes de *Performance* : Módulo de Integração com o *Facebook*

Manutenção da Informação Atualizada

Como descrito no capítulo de validação do sistema foram realizados testes no sentido de garantir que a informação é actualizada em todos os componentes necessários. Para além de garantir essa actualização, este conjunto de testes teve também por objetivo perceber qual o *delay* (típico) entre o momento de publicação na rede social e a notificação do CM+.

As medições foram efetuadas em três pontos: o tempo de publicação (presente na mensagem *json* enviada para o módulo), o tempo de receção no módulo de integração e o tempo de receção no CM+.

Verifica-se que o tempo médio desde que é feita uma publicação até que o sistema (CM+) é notificado é de 46 seg.. A maior parte do tempo corresponde ao intervalo entre a publicação na rede social e a chegada da notificação ao módulo.

Na medida em que este módulo está sujeito a fatores externos não controláveis, estes valores podem sofrer alterações independentes das funcionalidades implementadas.

Publicação	Módulo	CM+	FB -> Módulo	Módulo -> CM+	Total
15:25:21	15:25:55	15:26:01	00:00:34	00:00:06	00:00:40
15:35:07	15:36:16	15:36:23	00:01:09	00:00:07	00:01:16
15:48:40	15:49:15	15:49:19	00:00:35	00:00:04	00:00:39
15:52:18	15:52:55	15:52:59	00:00:37	00:00:04	00:00:41
15:56:25	15:57:01	15:57:07	00:00:36	00:00:06	00:00:42
16:04:23	16:05:01	16:05:04	00:00:38	00:00:03	00:00:41
16:19:00	16:19:46	16:19:48	00:00:46	00:00:02	00:00:48
16:25:55	16:26:02	16:26:05	00:00:07	00:00:03	00:00:10
16:21:58	16:23:15	16:23:19	00:01:17	00:00:04	00:01:21
16:24:54	16:25:33	16:25:37	00:00:39	00:00:04	00:00:43
TOTAL			00:00:42	00:00:04	00:00:46

Performance do Módulo

Para validar a escalabilidade deste módulo foram implementados testes de carga. A metodologia seguida para a sua realização é descrita no capítulo 7.

Foram considerados 4 cenários diferentes: 20, 50, 100 e 150 utilizadores virtuais e para cada um deles foi variado o número de pedidos gerados (entre 200 e 5000 pedidos). Para cada um dos pedidos foi calculado o tempo percorrido desde o momento que o pedido chega ao *web service* até que é enviado para o *Facebook*. Com esses registos foram calculados os tempos médios (em segundos) para cada um dos cenários e os resultados obtidos encontram-se na tabela seguinte.

UV's	200	400	600	800	1000	1500	2000	5000
20	1,943	0,214	0,239	0,158	0,341	0,177	0,127	0,062
50	4,016	5,701	9,255	7,139	6,38	5,957	8,241	15,262
100	3,596	8,834	8,107	6,777	10,755	15,92	22,148	33,974
150	2,52	7,092	9,411	16,237	12,208	17,55	22,956	38,99

