

REM

An Interface Between WebShops for eCommerce and ERP Systems at MM Software Consulting GmbH

INTERNSHIP REPORT

António Fiel Mendes Carvalho

MASTERS IN COMPUTER ENGINEERING



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

September | 2017

An Interface Between WebShops for eCommerce and ERP Systems at MM Software Consulting GmbH

INTERNSHIP REPORT

António Fiel Mendes Carvalho

MASTERS IN COMPUTER ENGINEERING

SUPERVISOR
Eduardo Leopoldo Fermé



UNIVERSIDADE da MADEIRA

*An interface between WebShops for eCommerce and ERP Systems
at MM Software Consulting GmbH*

António Fiel Mendes Carvalho

(Licenciado)

*Relatório de Estágio Submetido à Universidade da Madeira para a
Obtenção do Grau de Mestre em Engenharia Informática*

Bad Zwischenahn – Alemanha

(Setembro - 2017)

Orientador Interno:

Eduardo Leopoldo Fermé, PhD

Professor Associado da Faculdade de Ciências Exatas e da Engenharia da Universidade da Madeira

Orientador Externo:

Martin Melchert, Eng

CEO da empresa MM Software Consulting GmbH

ABSTRACT

This report intends to present the project developed and the experience lived during the accomplishment of the curricular internship for the conclusion of the Masters in Informatics Engineering of the University of Madeira.

An internship is an excellent opportunity to apply the skills acquired during the academic education in real practical cases. The main objectives for this internship were to create contacts and acquire more professional and personal experience in a new culture.

The internship was held at MMSC, a small company based in Germany. In this internship, a project was developed for the synchronization of data between Web stores for e-commerce and ERP systems.

In this report, the motivations for the realization of this project will be exposed, as well as the tools and technologies used to develop it. Subsequently, the planning and implementation of this project will be presented.

KEYWORDS

Software Engineering

ERP

E-Commerce

Web Shop

KobaltC

Interface

RESUMO

Este relatório pretende expor o projeto desenvolvido e a experiência vivenciada durante a realização do estágio curricular para a conclusão do Mestrado em Engenharia Informática da Universidade da Madeira.

Um estágio curricular é uma excelente oportunidade para aplicar as competências adquiridas durante a formação académica em casos práticos reais. Os principais objetivos para este estágio foram criar contactos e adquirir mais experiência profissional e pessoal numa nova cultura.

O estágio foi realizado na empresa MMSC, sediada na Alemanha. Neste estágio foi desenvolvido um projeto para a sincronização de dados entre lojas na Web para comércio eletrónico e sistemas ERP.

Neste relatório serão expostas as motivações para a realização deste projeto assim como as ferramentas e tecnologias utilizadas para o desenvolvimento do mesmo. Posteriormente, o planeamento e a implementação deste projeto serão apresentados.

PALAVRAS-CHAVE

Engenharia de Software

ERP

Comércio eletrônico

Loja na Web

KobaltC

Interface

AGRADECIMENTOS

O maior agradecimento dou à minha família, especialmente aos meus pais, por todo o apoio, força e carinho que demonstraram durante esta nova etapa para mim, apesar dos milhares de quilômetros que nos separavam. Em particular, gostaria de agradecer à minha irmã mais “nova” pelo seu incansável apoio e ânimo. Obrigado Cristina!

Um agradecimento especial para o meu orientador Professor Doutor Eduardo Fermé pelo acompanhamento e disponibilidade para este projeto de longa duração, assim como pelas suas orientações e sugestões preciosas para o desenvolvimento do projeto.

Quero também agradecer à MMSC e sua equipa por me terem recebido, particularmente ao Engenheiro Martin Melchert por ter partilhado comigo a sua longa experiência e pela sua confiança em mim ao oferecer-me um lugar na empresa. Gostaria também de dar um agradecimento especial ao Viktor Stoppel por todo o seu imprescindível apoio a nível profissional e a nível pessoal.

Para finalizar, quero agradecer a todos os amigos, aos que deixei na ilha e aos que fiz durante a minha estadia na Alemanha, pelas amizades e influências positivas para comigo. Tornaram tudo mais fácil.

Muito obrigado a todos!

ÍNDICE

ABSTRACT	III
KEYWORDS	IV
RESUMO	V
PALAVRAS-CHAVE.....	VI
AGRADECIMENTOS	VII
ÍNDICE.....	VIII
LISTA DE FIGURAS	X
LISTA DE TABELAS	XI
ACRÓNIMOS	XII
I. INTRODUÇÃO.....	1
I.1. APRESENTAÇÃO	2
I.2. CONTEXTO DO ESTÁGIO	3
I.3. CONTEXTO INSTITUCIONAL	4
I.3.1. Descrição da empresa.....	5
I.3.2. Localização da empresa.....	6
I.4. KOBALT•C ERP-DATA SERVER	7
I.5. OBJETIVOS	8
II. ESTADO DA ARTE	9
II.1. PLANEAMENTO DE RECURSOS EMPRESARIAIS.....	10
II.2. LOJA NA WEB PARA COMÉRCIO ELETRÓNICO.....	14
II.3. SISTEMAS SEMELHANTES NO MERCADO.....	18
II.3.1. cateno	18
II.3.2. sync4.....	19
II.3.3. ShopConnectFlex.....	19
II.3.4. Análise e considerações	20
II.4. FERRAMENTAS E TECNOLOGIAS UTILIZADAS.....	21
II.4.1. C++ e Code::Blocks	21
II.4.2. PostgreSQL e pgAdmin.....	22
II.4.3. HTML e JavaScript.....	23
II.4.4. JSON	23
II.4.5. TortoiseSVN e Mantis BugTracker	24
III. DESENVOLVIMENTO DO SISTEMA.....	25
III.1. PROJETO.....	26
III.2. METODOLOGIAS DE TRABALHO	28
III.2.1. As 5 dimensões de um projeto de software	28
III.3. REQUISITOS DO SISTEMA.....	30
III.3.1. Requisitos funcionais	30
III.3.2. Requisitos não funcionais	33
III.4. CASOS DE UTILIZAÇÃO	35
III.4.1. Atores.....	35
III.4.2. Diagrama de casos de utilização	35

III.4.3. Descrição dos casos de utilização	37
III.5. DIAGRAMAS DE ATIVIDADE	40
III.6. SOLUÇÃO TÉCNICA	43
III.7. ESTRUTURA DA BASE DE DADOS	45
III.7.1. Visão geral	45
III.7.2. Serviço	46
III.7.3. Lojas	47
III.7.4. Recursos	48
III.7.5. Elementos	49
III.7.6. Utilitários	50
III.8. INTERFACE DO UTILIZADOR	51
III.8.1. Estrutura da interface	51
III.8.2. Esboços da interface	52
III.9. IMPLEMENTAÇÃO DO SISTEMA	55
III.9.1. Protótipo	55
III.9.1.1. Estrutura da base de dados	55
III.9.1.2. Interface do utilizador	57
III.9.2. KbltShop	59
IV. PROJETOS DIVERSOS	67
IV.1. DESENVOLVIMENTO, MANUTENÇÃO E TESTE DE SOFTWARE	68
IV.2. SUPORTE AO CLIENTE	73
V. CONCLUSÃO	75
V.1. SOBRE O PROJETO DESENVOLVIDO	76
V.2. SOBRE O ESTÁGIO	77
V.3. PERSPETIVAS FUTURAS	78
VI. REFERÊNCIAS	79

LISTA DE FIGURAS

FIGURA I-1 - LOGOTIPO MM SOFTWARE CONSULTING GMBH.....	5
FIGURA I-2 - ENTRADA PARA OS ESCRITÓRIOS DO NORTE.....	6
FIGURA I-3 – LOGOTIPO KOBALT•C ERP-DATA SERVER.....	7
FIGURA II-1 - EXEMPLO DA ORGANIZAÇÃO DE UMA EMPRESA [1].....	10
FIGURA II-2 - ILUSTRAÇÃO DE UM PROCESSO SEM ERP [3].....	11
FIGURA II-3 - LOGOTIPO MICROTECH [6].....	13
FIGURA II-4 - TIPOS DE COMÉRCIO ELETRÔNICO [7].....	14
FIGURA II-5 - LOGOTIPO CATENO [11].....	18
FIGURA II-6 - LOGOTIPO SYNC4 [12].....	19
FIGURA II-7 - LOGOTIPO SHOPCONNECTFLEX [13].....	19
FIGURA II-8 - PROGRAMA OLÁ MUNDO EM C++.....	21
FIGURA II-9 - PGADMIN III.....	22
FIGURA II-10 - REPRESENTAÇÃO DO MVVM [20].....	23
FIGURA II-11 - ESTRUTURA DO FORMATO DE DADOS JSON [21].....	24
FIGURA II-12 - LOGOTIPO TORTOISESVN [22].....	24
FIGURA II-13 - LOGOTIPO MANTIS BUGTRACKER [23].....	24
FIGURA III-1 - AS 5 DIMENSÕES DE UM PROJETO DE SOFTWARE.....	28
FIGURA III-2 - DIAGRAMA DE FLEXIBILIDADE PARA ESTE PROJETO.....	29
FIGURA III-3 - DIAGRAMA DE CASOS DE UTILIZAÇÃO.....	36
FIGURA III-4 - DIAGRAMA DE ATIVIDADE PARA O CASO "LISTAR ELEMENTOS".....	40
FIGURA III-5 - DIAGRAMA DE ATIVIDADE PARA O CASO "ATUALIZAR ELEMENTOS".....	41
FIGURA III-6 - DIAGRAMA DE ATIVIDADE PARA O CASO "PROCESSAR ELEMENTO".....	42
FIGURA III-7 - DIAGRAMA DA GENERALIZAÇÃO DAS LOJAS.....	44
FIGURA III-8 - DIAGRAMA DA GENERALIZAÇÃO DOS RECURSOS.....	44
FIGURA III-9 - TABELAS PRINCIPAIS DA ESTRUTURA.....	45
FIGURA III-10 - TABELAS PARA GESTÃO DO SERVIÇO.....	46
FIGURA III-11 - TABELAS PARA CONFIGURAÇÃO DAS LOJAS.....	47
FIGURA III-12 - TABELAS PARA CONFIGURAÇÃO DA SINCRONIZAÇÃO DOS RECURSOS DAS LOJAS.....	48
FIGURA III-13 - TABELAS PARA O TRATAMENTO DOS ELEMENTOS.....	49
FIGURA III-14 - TABELAS PARA CONFIGURAÇÃO DO MAPEAMENTO DE VALORES E SEQUÊNCIAS.....	50
FIGURA III-15 - PÁGINA INICIAL DO KOBALTC WEBCLIENT.....	51
FIGURA III-16 – ESTRUTURA DA INTERFACE DO UTILIZADOR.....	52
FIGURA III-17 - ESBOÇO DO MENU "EINSTELLUNGEN".....	53
FIGURA III-18 - ESBOÇO DO MENU "SHOPS".....	53
FIGURA III-19 - DIAGRAMA ENTIDADE RELAÇÃO PARA O PROTÓTIPO.....	56
FIGURA III-20 - PÁGINA PARA A CONFIGURAÇÃO DOS PARÂMETROS DA API.....	57
FIGURA III-21 - PÁGINA PARA O MAPEAMENTO DE VALORES ENTRE ERP E SHOPWARE.....	58
FIGURA III-22 - PÁGINA PARA CONFIGURAR PARÂMETROS PARA O RECURSO ARTIGO.....	59
FIGURA III-23 - ESTRUTURA COMPLETA DA BASE DE DADOS.....	65
FIGURA IV-1 - KOBALTC DESKTOP HELPER.....	69
FIGURA IV-2 - INTERFACE PARA A CONFIGURAÇÃO DA RECOLHA DOS DADOS PARA UTILIZAR NO EMAIL.....	70
FIGURA IV-3 - CONFIGURAÇÃO DA COMPOSIÇÃO DO EMAIL.....	71
FIGURA IV-4- ILUSTRAÇÃO DO DESENVOLVIMENTO DE UM PROJETO DE SOFTWARE [26].....	72
FIGURA IV-5 - PROCESSO MDE ABGANG.....	72

LISTA DE TABELAS

TABELA III-1 - REQUISITOS FUNCIONAIS	33
TABELA III-2 - REQUISITOS NÃO FUNCIONAIS	34
TABELA III-3 - DESCRIÇÃO DO CASO DE UTILIZAÇÃO "LISTAR ELEMENTOS"	37
TABELA III-4 - DESCRIÇÃO DO CASO DE UTILIZAÇÃO "ATUALIZAR ELEMENTOS"	38
TABELA III-5 - DESCRIÇÃO DO CASO DE UTILIZAÇÃO "PROCESSAR ELEMENTOS"	38
TABELA III-6 - DESCRIÇÃO DO CASO DE UTILIZAÇÃO "CONFIGURAR SISTEMA"	39
TABELA III-7 - DESCRIÇÃO DO CASO DE UTILIZAÇÃO "EFETUAR MANUTENÇÃO"	39
TABELA III-8 – ESTADO DE IMPLEMENTAÇÃO DOS REQUISITOS PARA O SISTEMA.....	63

ACRÓNIMOS

API – Application Programming Interface
BDE – Betriebs Daten Erfassung
CEO – Chief Executive Officer
CSS – Cascading Style Sheet
eCommerce – Comércio eletrônico
EDI – Electronic Data Interchange
ERP – Enterprise Resource Planning
GmbH – Gesellschaft mit beschränkter Haftung
HTML – HyperText Markup Language
IDE – Integrated Development Environment
IMEX – Import Export
JSON – JavaScript Object Notation
KG – Kommanditgesellschaft
LMS – Lager Management System
MDE – Mobile Daten Erfassung
MMSC – Mayer Melchert Software Consulting
MS – Microsoft
REST – Representational state transfer
RPT – Reporting
SAP – System, Applications and Products
SQL – Structured Query Language
SOAP – Simple Object Access Protocol
UUID – Universally Unique Identifier

I. INTRODUÇÃO

I.1. APRESENTAÇÃO

Com o seguinte relatório pretendo descrever a experiência vivida ao longo de todo o estágio, que foi a minha opção para concluir o segundo ciclo de estudos na Universidade da Madeira, nomeadamente do Mestrado em Engenharia Informática. Um estágio, especialmente num país estrangeiro, traz imensas mais valias, quer a nível profissional como a nível pessoal, pois envolve a minha integração e envolvimento numa empresa, que é um ambiente bastante diferente do académico, assim como a minha adaptação a uma nova cultura e modo de viver.

O estágio foi realizado na íntegra na Alemanha, na empresa MM Software Consulting GmbH. Esta foi a minha primeira longa experiência fora do conforto de Portugal, mais especificamente, da Madeira, e não terminou com o estágio, já que me abriu uma nova janela de oportunidades com a possibilidade de ficar contratado nesta empresa.

Uma das mais-valias deste estágio foi a oportunidade de trabalhar com pessoas com muitos anos de experiência profissional e elevada competência na área e os contactos que obtive através dos mesmos.

I.2. CONTEXTO DO ESTÁGIO

O estágio curricular, no âmbito do Mestrado de Engenharia Informática da Universidade da Madeira, é uma das três hipóteses da unidade curricular Projeto/Estágio/Dissertação para a realização de um trabalho, com o objetivo de dar por concluído o segundo ciclo de estudos.

A opção pelo estágio foi a escolhida devido ao desejo de obter experiência e competências próprias que o contexto empresarial e de mundo real oferece, que difere bastante do académico e permite, ao mesmo tempo, aplicar o conhecimento adquirido ao longo do percurso académico. Para realizar o estágio, faltava apenas encontrar uma empresa com um projeto apetecível para todas as partes, sendo estas a empresa, a Universidade e eu próprio.

A oportunidade de realizar o estágio surgiu ao tentar diversos contatos que obtive durante os dois anos e meio em que estive vinculado ao meu primeiro emprego. Aproveito para acrescentar que tive conhecimento da vaga para o emprego através de um anúncio publicado por um dos docentes nos fóruns do Moodle da Universidade da Madeira.

O contato em que obtive sucesso foi com Viktor Stoppel, gestor de projetos e programador na empresa MM Software Consulting GmbH que está sediada na cidade de Bad Zwischenahn na Alemanha, a quem tive a oportunidade de conhecer a nível profissional e pessoal. Após diversas conversas que me deixaram a par da empresa e projetos correntes e futuros, sugeri então o estágio. Obtive uma resposta imediata e positiva com diversos projetos à escolha, as condições do estágio e perspectivas após conclusão deste mesmo. Como estas condições me agradaram imenso, parti então para a escolha de um dos projetos.

I.3. CONTEXTO INSTITUCIONAL

O estágio foi possível, como já foi referido, ao entrar em contato com o Viktor Stoppel, e posteriormente com o Engenheiro Martin Melchert, que lidera a empresa, este último que aceitou ser o meu orientador externo e ditou quais as condições do estágio. Estas condições alteraram-se com o decorrer do estágio das quais realço as mais importantes:

- A realizar na Alemanha;
- Possibilidade de futuramente obter de um contrato de forma a ficar efetivo e ter mais segurança a nível de saúde e financeiro;
- Disponibilidade para participar noutros projetos.

A proposta de estágio foi recebida de forma muito positiva pois a empresa, que é muito jovem, estava a precisar de mais colaboradores, sendo que um estagiário é uma solução financeira excelente. Outro fator para a aceitação do estágio, talvez o mais determinante, foi o fato de já nos termos cruzado profissionalmente no passado, tornando a minha integração mais fácil.

Considerando que uma das condições para a realização deste estágio seria realizá-lo na Alemanha, parti para Oldemburgo, cidade vizinha à sede da referida empresa.

I.3.1. Descrição da empresa

A MM Software Consulting GmbH, ou MMSC, é uma empresa pequena e jovem, fundada em dezembro de 2012, composta por cerca de 6 colaboradores e com planos para integrar mais elementos. A MMSC dedica-se a diversas atividades no sector das tecnologias de informação sempre relacionadas com sistemas ERP (sistemas de planeamento de recursos empresariais), área na qual a equipa possui um total superior a cinquenta anos de experiência. Destas atividades, destacam-se as seguintes:

- Desenvolvimento, manutenção, marketing e venda do próprio software Kobalt•C ERP-DataServer;
- Marketing e venda, em parceria com a Microtech, do sistema ERP denominado por ERP Complete;
- Revenda de dispositivos móveis para assistir à gestão de armazém;
- Consultoria em diversas áreas relacionadas com sistemas ERP, focando sobretudo, mas não exclusivamente o ERP Complete;
- Ensino e otimização de processos de gestão e de negócio relacionados com ERP, como gestão de riscos e de qualidade, a clientes ou parceiros;
- Suporte técnico e gestão dos servidores de clientes do Kobalt•C.



Figura I-1 - Logotipo MM Software Consulting GmbH

A empresa é também certificada no programa de *Berufsausbildung* (formação profissional), que lhe permite receber, sobretudo jovens, que tenham terminado os seus estudos e que queiram integrar numa formação profissional num contexto empresarial. No caso da MMSC, a formação é em *Fachinformatiker Anwendungsentwicklung* (especialista em tecnologias da informação e desenvolvimento de aplicações) e tem uma duração de três anos. No início do meu estágio, a empresa formava o Daniel

Gödecke, que se tornou efetivo após ter terminado a sua formação. Entretanto foram recebidas duas novas formandas, a Julia e a Rike.

I.3.2. Localização da empresa

A sede da empresa está localizada no norte da Alemanha, em Bad Zwischenahn, pertencente ao estado de Niedersachsen (Baixa Saxónia), com o seguinte endereço: Zum Pionierpark 6A, D-26160. A empresa possui também um escritório no sul da Alemanha, facilitando assim a logística para clientes e parceiros do sul da Alemanha, Suíça e Áustria.



Figura I-2 - Entrada para os escritórios do norte

Na Alemanha, o prefixo Bad, quando se referindo a uma cidade ou localidade, significa que se trata de um local cujas atividades estão centradas à volta do contato com a natureza, turismo de saúde e bem-estar. Este é o caso de Bad Zwischenahn, que também conta com uma população algo envelhecida, sendo um dos motivos pela qual decidi ficar na cidade vizinha de Oldemburgo durante o estágio.

I.4. KOBALT•C ERP-DATA SERVER

O produto principal da empresa é o Kobalt•C ERP-Data Server, um sistema modular composto por vários serviços especializados na recolha, transformação, armazenamento e envio de dados, dentro do contexto dos módulos com os quais os respetivos serviços interagem. A finalidade é dar suporte e estender as funcionalidades de sistemas de planeamentos de recursos empresariais (ERP).



Figura I-3 – Logotipo Kobalt•C ERP-Data Server

A MMSC pretende chegar a pequenas e médias empresas que procuram ou necessitam de soluções para agilizar e automatizar os seus processos de negócio através dos seguintes módulos do Kobalt•C:

- RPT – fornece ferramentas para configuração, geração e entrega automática de relatórios e alertas;
- IMEX – permite automatizar a exportação e importação de dados, em diversos formatos e meios de transporte;
- MDE – suporte para a recolha e tratamento de dados em tempo real de bens em movimento através de dispositivos móveis;
- LMS – sistema que efetua a gestão automática do armazém;
- BDE – sistema para a gestão dos colaboradores e das máquinas utilizadas ou serviços prestados por estes em função do tempo dos recursos empresariais;
- EDI – fornece as ferramentas para configurar a troca de dados entre duas empresas através dos formatos standardizados para EDI.

A acrescentar a estes acima, está também planeado o módulo SHOP, que servirá como interface entre lojas na web para eCommerce e sistemas ERP.

I.5. OBJETIVOS

Para o estágio curricular, a empresa propôs-me então a conceptualização e implementação do módulo SHOP com os seguintes objetivos:

- i. Desenvolvimento de um protótipo deste serviço que permita a sincronização de artigos e encomendas entre a loja na *Web Shopware* e o sistema ERP *ERP Complete* para um cliente específico.
- ii. Desenvolvimento de um serviço que permita troca e sincronização de dados entre diversas lojas e sistemas ERP, numa relação de n para n .
- iii. Desenvolvimento de uma interface gráfica que permita configurar o proposto no segundo objetivo.

A minha responsabilidade no primeiro ponto seria dar continuidade ao já existente esboço do protótipo a desenvolver, protótipo este que serve também, para me ambientar de uma forma prática com o ecossistema *KobaltC*, as suas ferramentas e de como interagir com as lojas na web.

A solução a desenvolver para o segundo e principal ponto seria inteiramente da minha responsabilidade, contando sempre com o apoio dos restantes colegas no decorrer do projeto quando necessário.

O objetivo proposto no terceiro ponto não é crítico, pelo que pode ser concluído após o término do estágio, seja por mim ou por um outro colaborador da empresa.

II. ESTADO DA ARTE

II.1. PLANEAMENTO DE RECURSOS EMPRESARIAIS

O desenvolvimento das tecnologias de informação e o mercado competitivo, leva as organizações empresariais a tirarem proveito dos sistemas de informação de forma a aumentarem a sua competitividade com uma melhor eficiência e eficácia na tomada de decisão. Estes sistemas suportam a empresa nas atividades das suas diferentes áreas de negócios, sejam estas internas tais como a produção, o processamento de encomendas e a gestão de recursos humanos, assim como, em interações externas com clientes, fornecedores e parceiros de negócio. [1]

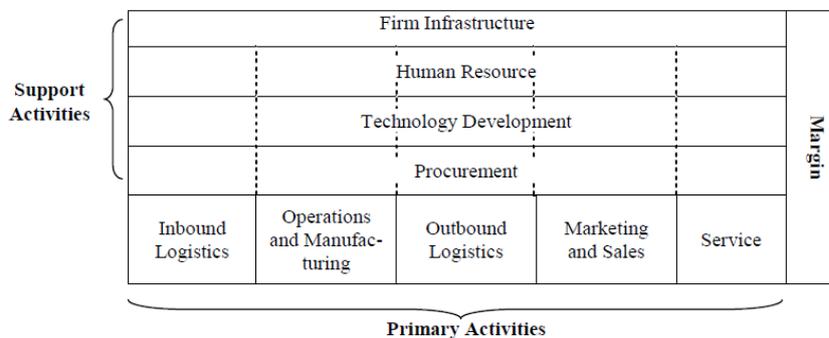


Figura II-1 - Exemplo da organização de uma empresa [1]

Os negócios empresariais têm funcionado como uma estrutura organizacional, na qual, as áreas de negócio funcionais estão separadas em diferentes departamentos [2], onde a gestão de cada departamento é realizada através de sistemas individuais [3]. Desta forma, cada departamento encontra-se isolado dos restantes e dos dados contidos nestes [2], o que pode levar ao risco de os dados da empresa estarem comprometidos devido a estarem guardados em diversos locais. Isto pode criar redundância desnecessária nos dados, pode também levar a que um departamento esteja a trabalhar e a tomar decisões com dados desatualizados, ou ainda, que os processos de negócio possam estar a levar mais tempo a serem concluídos devido à espera dos dados de outros departamentos necessários para conclusão do processo. [3] Os clientes podem também ser afetados como se pode constatar no cenário ilustrado na figura seguinte.



Figura II-2 - Ilustração de um processo sem ERP [3]

Devido à segregação dos dados e da informação que cada departamento tem ao seu dispor, o serviço acabou por não ser eficiente como o podia ser caso o atendimento tivesse ao seu dispor todas as informações relacionadas com o cliente, mesmo que estas sejam relativas a outros departamentos. Caso o atendimento tivesse ao seu dispor um sistema ERP e acesso à informação correta, o serviço prestado seria muito mais eficiente e a satisfação do cliente seria certamente maior. [3]

Um sistema ERP, ou sistema de planeamento de recursos empresariais, é uma aplicação de software modular complexa com a finalidade de integrar todos os processos e departamentos de negócio numa única plataforma. [1] Isto é conseguido com uma base de dados centralizada, à qual todas as áreas de negócio funcionais têm acesso, facilitando assim a automação de processos dentro da empresa e eliminando a necessidade de sincronização de dados entre os diferentes departamentos. [4] O sistema ERP permite os seguintes resultados:

- Fornecer um quadro de informação abrangente que integra funções, departamentos e níveis hierárquicos em uma cadeia composta de ação-resposta de eventos. Assim, a entrada de uma nova encomenda deduz automaticamente o material disponível no inventário, encomenda o material necessário aos fornecedores, atualiza a projeção de produção, revisa os horários de trabalho e prepara as novas projeções de mercado.
- Fornecer uma única e compreensiva base de dados, na qual todas as transações de negócio são inseridas, registadas, processadas, monitorizadas e relatadas. Isto reduz a entrada de dados e a monitorização,

facilita o uso do conhecimento tácito e permite coordenação de estratégias orientadas ao cliente (como personalização em massa).

- Aumenta a velocidade das transações de informações.
- Aumenta a conectividade estrutural das unidades e atividades. As consequências das transações tornam-se visíveis e os efeitos da escolha em uma unidade tornam-se aparentes por toda a empresa. [1]

Os sistemas ERP estão a se tornar mais necessários para qualquer empresa para aumentar a competitividade no mercado, e o sucesso da implementação destes sistemas levam a ganhos tangíveis e intangíveis, resultados estes, que levam a que haja um maior investimento de empresas em sistemas ERP. [1] Com uma implementação bem-sucedida de um sistema ERP são antecipados alguns benefícios, dentro dos quais: processos de negócio mais eficientes e com maior qualidade, reduzida complexidade e melhor harmonização da estrutura de sistemas de informação, melhor e mais rápido cumprimento dos requisitos legais, aumento da flexibilidade organizacional e redução de custos. [3]

A implementação de um sistema ERP numa empresa tem os seus pontos menos positivos, começando pelo seu custo. Este é um processo que poderá ser muito dispendioso em tempo e dinheiro, custo este que poderá aumentar consideravelmente consoante as necessidades de personalização e de integração de sistemas existentes, especialmente em organizações mais complexas e dispersas geograficamente. [1]

O grau de sucesso da implementação do sistema ERP na empresa pode variar consoante diversos fatores. Um destes fatores é a habilidade e experiência dos trabalhadores com o sistema. O fator, talvez o mais importante, é a capacidade de a empresa conseguir realizar as alterações organizacionais necessárias, e por vezes radicais, de forma a atingir o sucesso pretendido com o sistema ERP. [1]

Os sistemas ERP podem ser categorizados de diversas formas: pela sua dimensão, pelas tecnologias por detrás dos mesmos e pelas indústrias que o sistema pretende suportar. No mercado atual existe uma grande variedade e escolha de sistemas ERP, e curiosamente, as três maiores empresas de software mundiais têm uma forte presença nesta área, sendo estas a Microsoft, a Oracle e a SAP respetivamente.

[5] De referir ainda que a SAP se dedica a softwares de gestão de empresas e é a líder global de mercado em soluções nesta área.

A MMSC suporta neste momento o sistema ERP da empresa Microtech denominado como ERP Complete. Este sistema tem presença em mercados onde a língua principal é a alemã. Com este sistema a Microtech pretende chegar a pequenas e médias empresas nos países da Alemanha, Áustria e Suíça.



Figura II-3 - Logotipo Microtech [6]

II.2. LOJA NA WEB PARA COMÉRCIO ELETRÓNICO

O advento da internet nos anos 90 trouxe um novo paradigma para o mundo dos negócios. A internet, funcionando como uma rede global, associa à vantagem de ser universal, o facto de apresentar baixos custos de acesso e funcionamento, facilidade de uso, flexibilidade e interatividade. [7] A promessa das tecnologias de informação para permitir empresários e empresas vender e comprar os seus produtos através de redes digitais, tem sido uma das prioridades de desenvolvimento para a comunidade internacional desde o fim dos anos 90. Em 1999, a UNCTAD (United Nations Conference on Trade And Development), enfatizou que o comércio eletrónico tem o potencial para se um dos principais motores de comércio e desenvolvimento à escala global. [8]

O comércio eletrónico pode ser definido como compras e vendas efetuadas através de redes de computadores, utilizando múltiplos formatos e dispositivos, incluindo a internet e troca de dados eletrónicos, utilizando computadores pessoais, portáteis, tablets e telefones móveis de níveis de sofisticação variados. Este comércio envolve produtos físicos, digitais e serviços que podem ser entregues de uma forma digital. [8]

Considerando as diversas possibilidades para a classificação do comércio eletrónico, há uma preferência para se basear nos tipos de intervenientes envolvidos nas transações para tal. [7] Assim são reconhecidos quatro principais tipos de negócio: Business-to-Business (B2B), Business-to-Consumer (B2C), Business-to-Administration (B2A) e Consumer-to-Administration (C2A). [7]

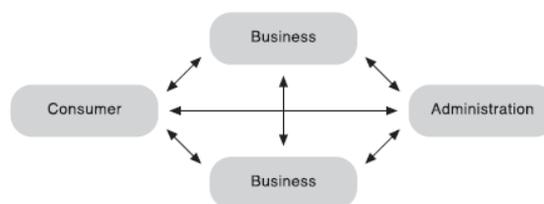


Figura II-4 - Tipos de comércio eletrónico [7]

- Business-to-Business (B2B): constitui a maior parte do negócio eletrónico, e envolve as transações efetuadas entre negócios ou empresas,

tais como entre fabricante e grossista, ou entre um grossista e um retalhista. [8]

- Business-to-Consumer (B2C): esta categoria corresponde à secção de retalho do comércio eletrónico, ou seja, negócio realizado entre empresas e consumidores. A venda direta a consumidores, através de tecnologias de informação podem ajudar micro e pequenos negócios a alcançarem novos mercados quer sejam domésticos ou internacionais. [8]
- Business-to-Administration (B2A): pertencem a esta categoria todas as transações eletrónicas entre empresas e administração pública, através de diversos serviços tais como áreas fiscal, segurança social, emprego, registos, etc... [7]
- Consumer-to-Administration (C2A): aqui estão abrangidas as transações eletrónicas entre indivíduos e a administração pública, tais como, marcação de consultas, realização de pagamentos, formação à distancia, entrega das declarações, entre outras. [7]

Em relação ao comércio tradicional, o comércio eletrónico tem associadas algumas vantagens e desvantagens. Entre as vantagens temos:

- O mercado abrangido é global, limitado sobretudo pela cobertura das redes informáticas;
- As informações recolhidas das interações eletrónicas com clientes, permitem, dar uma maior ênfase ao consumidor final através de produtos e serviços ajustados às preferências do cliente;
- Os serviços eletrónicos estão disponíveis 24 horas por dia.
- Os fornecedores podem colocar-se mais próximos dos clientes aumentando assim, a produtividade, competitividade e qualidade dos serviços e produtos por parte das empresas. Esta aproximação permite também aligeirar as cadeias de distribuição reduzindo, e por vezes eliminando, os intermediários entre fornecedores e clientes;

As empresas e os consumidores beneficiarão com as consequências de todos estes aspetos. As empresas serão beneficiadas com um aumento das oportunidades e

da sua competitividade, que por sua vez, beneficia os consumidores com a melhoria da qualidade e redução dos preços dos produtos e serviços disponibilizados. Não só vantagens, mas também desvantagens estão associadas ao comércio eletrônico, sendo algumas destas as seguintes:

- Existe uma maior dependência das empresas face às tecnologias da informação;
- As infraestruturas de comunicação que possam ser deficientes ou mal dimensionadas, nomeadamente ao nível da largura de banda e capacidade de processamento no acesso a determinados servidores. Algumas das consequências poderão ser uma acrescida dificuldade em aceder aos serviços pretendidos, assim como, ter uma experiência muito pouco fluída e frustrante ao utilizar os serviços;
- O elevado custo das telecomunicações e manutenção das mesmas, sobretudo, onde possa haver situações de monopólio dos operadores de telecomunicações;
- A legislação pode ser insuficiente para regular de uma forma adequada as novas atividades proporcionadas pelo comércio eletrônico;
- Uma cultura de mercado avessa ao comércio eletrônico, que poderá levar ao impedimento à aceitação do mesmo. Poderá ser, por exemplo, o caso de consumidores que prefiram tocar e/ou experimentar os produtos antes de os adquirir;
- Uma interação excessiva com os consumidores, já que, se esta for levada ao extremo, poderá a levar que as empresas não consigam dar resposta ao elevado número de pedidos que recebe. Um elevado número destes casos ou consumidores insatisfeitos afetará a imagem da empresa. [7]

A forma mais comum de comércio eletrônico para consumidores privados é através de lojas web ou virtuais. As lojas virtuais são páginas na internet, acedidas através de navegadores de internet ou aplicações móveis, com diversas características. Estas lojas possuem um inventário de produtos disponíveis para o cliente, através de imagens e informações diversas sobre cada produto, e normalmente é possível ao cliente realizar uma pesquisa para encontrar o produto pretendido mais facilmente.

Uma outra característica é a presença de um carrinho de compras, onde o utilizador pode facilmente adicionar os produtos que deseja. Através da plataforma de pagamento, o utilizador pode efetuar a transação e adquirir os produtos colocados no carrinho de compras. [9]

A disponibilidade de lojas Web ou aplicações para comércio eletrónico, centrado nos consumidores é enorme. Aqui as empresas têm a possibilidade de ter a sua própria loja, ou usar a loja de terceiros para fazer os seus produtos chegar aos consumidores. E é precisamente sob esta segunda opção que duas das maiores empresas globais se encontram. Estas são inclusivamente as líderes neste segmento, sendo estas respetivamente a Amazon.com e a eBay. [10] Ambas permitem que qualquer empresa e até mesmo qualquer consumidor, com relativa facilidade, possam mostrar os seus produtos e vendê-los.

II.3. SISTEMAS SEMELHANTES NO MERCADO

Antes de iniciar o desenvolvimento do sistema, foi feita uma pesquisa no mercado por sistemas semelhantes. O critério de escolha dos sistemas que mais pesou foi o sistema ser visto como um concorrente do sistema KobaltC com o módulo SHOP. Um concorrente foi então definido como um sistema que faça sincronização de dados entre o sistema ERP da Microtech e lojas na Web para comércio eletrónico. Através dos contactos e conhecimento do mercado obtidos dos colegas da MMSC, elaborei uma lista dos sistemas com relevância. A escolha recaiu sobre o sistema *cateno* da empresa *cateno GmbH & Co. KG*, sobre o sistema *sync4* da *Dupp GmbH*, e também, o sistema *ShopConnectFlex* da *Compusoft Hard- & Software GmbH*.

II.3.1. cateno

O sistema *cateno* foi lançado em 2002 para interligar os sistemas ERP da *Microtech* com lojas na web. Desde então é um sistema focado na vertente de comércio eletrónico, apesar de também incluir outros serviços como gestão de armazém e postos de venda (PoS).



Figura II-5 - Logotipo *cateno* [11]

Em termos de lojas na web, o suporte é mais diversificado, suportando as mais conhecidas lojas na web no território alemão como *Shopware*, *Magento* e *xt::Commerce*. Suporta também diversos marketplaces com *amazon.com*, *ebay* e *Rakuten* entre estes.

Neste sistema podemos destacar diversas funcionalidades. A primeira sendo a sincronização paralela dos dados do sistema ERP com uma ou mais lojas. Uma outra funcionalidade destacável é o recálculo automático dos preços dos produtos tendo em conta dados de outras lojas para oferecer o melhor negócio possível. Por fim, destaca-se também com a funcionalidade de obter automaticamente números de expedição de diversas empresas que realizam o transporte de mercadorias a nível internacional.

II.3.2. sync4

O sistema *sync4* é um sistema dedicado totalmente a suporte para lojas na web e conta com mais de 25 anos no mercado. Este tem suporte para por volta de dezanove sistemas ERP distintos, incluindo SAP e Microsoft Dynamics. Oferece também uma solução própria para gestão dos dados de artigos e encomendas caso não possua ou queira usar um sistema ERP para usar como base. Já em termos de lojas, suporta até catorze lojas diferentes incluindo *Shopware, xt:Commerce, amazon.com* e *ebay*.



Figura II-6 - Logotipo sync4 [12]

Para o funcionamento do sistema é requerido o sistema gestor de bases de dados *MS SQL Server* da *Microsoft*. Uma funcionalidade a destacar é poder criar novos atributos para artigos e gerar novas variantes para os artigos com relativa facilidade. Possibilita também, através da sua interface gráfica, especificar preços diferentes para certos artigos para determinados clientes. Um destaque final vai para as aplicações móveis disponíveis para as plataformas Android e iOS que permitem a monitorização e alguma capacidade de administração do sistema.

II.3.3. ShopConnectFlex

A *CompuSoft Hard- & Software GmbH* oferece produtos distintos para as diferentes lojas que suporta. Apenas o sistema ERP da *Microtech* é suportado e para cada loja suportada existe um produto ou software diferente. Até para a mesma loja, no caso da *Shopware*, existem mais que um produto para diferentes funcionalidades. As lojas suportadas são *Shopware, Presta, Oxid, Amazon* e *Ebay*.



Figura II-7 - Logotipo ShopConnectFlex [13]

Destaco nesta solução a capacidade de adicionar marcas de água nas imagens quando efetua o carregamento destas para a loja, como também, a capacidade que esta

tem de se conectar diretamente à base de dados da loja que permite ler os dados diretamente das tabelas.

II.3.4. Análise e considerações

A equipa da MMSC já estava a par de algumas das características destas aplicações, sobretudo da *cateno*. Esta é vista como a grande concorrente, pois, oferece alguns serviços semelhantes aos da MMSC para além do suporte para lojas na web, o que já o faz há mais de quinze anos. Os clientes têm uma preferência muito maior por um software que oferece tudo o que precisam, isto é, que os seus dados sejam tratados apenas por apenas uma empresa.

Foi observado que as lojas *Shopware*, *Amazon* e *ebay* são comuns no leque de oferta, assim como há interesse nas lojas *Oxid* e *xt:Commerce*. A loja *Magento*, apesar de ser referenciada apenas por um dos sistemas acima, é uma das lojas que a MMSC pretende suportar, visto que já recebeu diversos pedidos por esta.

No que toca à sincronização dos dados entre o sistema ERP e as lojas na web, as soluções apresentadas acima são todas muito semelhantes em relação aos dados que suportam em ambas as direções. Uma funcionalidade comum observada foi poder alterar, e até mesmo criar, diversos atributos dos artigos através do sistema, suportando assim campos que possam não existir ou ter um suporte adequado por uma das partes. Outra funcionalidade comum é a criação e envio automático de alertas quando detetada uma situação indesejada ou a simples criação de um relatório.

Uma funcionalidade muito interessante e potencialmente cativadora para clientes, é a de obtenção de números de expedição de encomendas de uma forma automática. Mesmo que o cliente não pretenda usar ou não tenha uma loja na web, isto pode ser um serviço que cativa um cliente a decidir por usar o KobaltC.

Na tabela seguinte faço uma análise comparativa destas soluções juntamente com a solução aqui desenvolvida para uma interface entre lojas e sistemas ERP. Não pretendo efetuar uma descrição exaustiva das características de cada uma das soluções, optando por identificar algumas das características globais de forma a facilitar a comparação entre as mesmas.

II.4. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Sendo este projeto a conceptualização e implementação de um serviço adicional ou extensão do software já existente, as ferramentas e linguagens a utilizar já estavam bem definidas à partida.

As ferramentas e tecnologias a utilizar para este projeto foram definidas pelo Engenheiro Melchert. Nos seus critérios pesaram bastante a sua longa experiência com algumas destas tecnologias e linguagens, as capacidades multiplataforma e de computação distribuída das mesmas e serem *open-source*¹.

II.4.1. C++ e Code::Blocks

O C++ é uma linguagem de programação compilada que suporta diversos paradigmas, é multiplataforma, permite uma excelente utilização dos recursos do sistema e é considerada de nível intermédio, isto é, possui características de linguagens de alto e de baixo nível. A forma eficiente como utiliza os recursos tornou-a muito escolhida na implementação de serviços cujos recursos sejam limitados, e também em serviços de maior escala devido ao seu desempenho e flexibilidade. [14] Todos os serviços integrados no KobaltC estão implementados utilizando esta linguagem.

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Olá Mundo!" << endl;
    return 0;
}
```

Figura II-8 - Programa Olá Mundo em C++

O Code::Blocks é um IDE *open-source*, multiplataforma com uma licença de utilização gratuita para as linguagens C, C++ e Fortran. Esta ferramenta é um IDE completo com um grau de personalização muito alto e imensas opções para estender ainda mais as funcionalidades que oferece à partida. [15]

¹ *open-source* – código aberto, que é uma abordagem ao desenvolvimento de software onde o código fonte de um sistema é tornado público e os utilizadores são encorajados a participar no desenvolvimento do sistema [25]

II.4.2. PostgreSQL e pgAdmin

O PostgreSQL é um sistema de gestão de base de dados objeto-relacional, com mais de 15 anos de desenvolvimento em código aberto. Este é compatível com o conceito ACID (atomicidade, consistência, isolamento e durabilidade) e é compatível com qualquer dos maiores sistemas operacionais do mercado ganhando uma boa reputação em termos de fiabilidade, integridade dos dados e correção dos dados. É um sistema de nível empresarial que segue os padrões existentes em bases de dados relacionais, embora também seja completamente personalizável, sendo compatível com diversas outras linguagens para a extensão de funcionalidades, para além das funcionalidades sofisticadas que oferece por defeito como a replicação assíncrona de uma base de dados ou partes de uma por diversas máquinas ou nodos de uma rede, transações agrupadas e suporte para *NoSQL* através de JSON. [16]

A ferramenta de administração gráfica utilizada é pgAdmin III, suportando todas as funcionalidades da base de dados com uma interface fácil e intuitiva. [17]

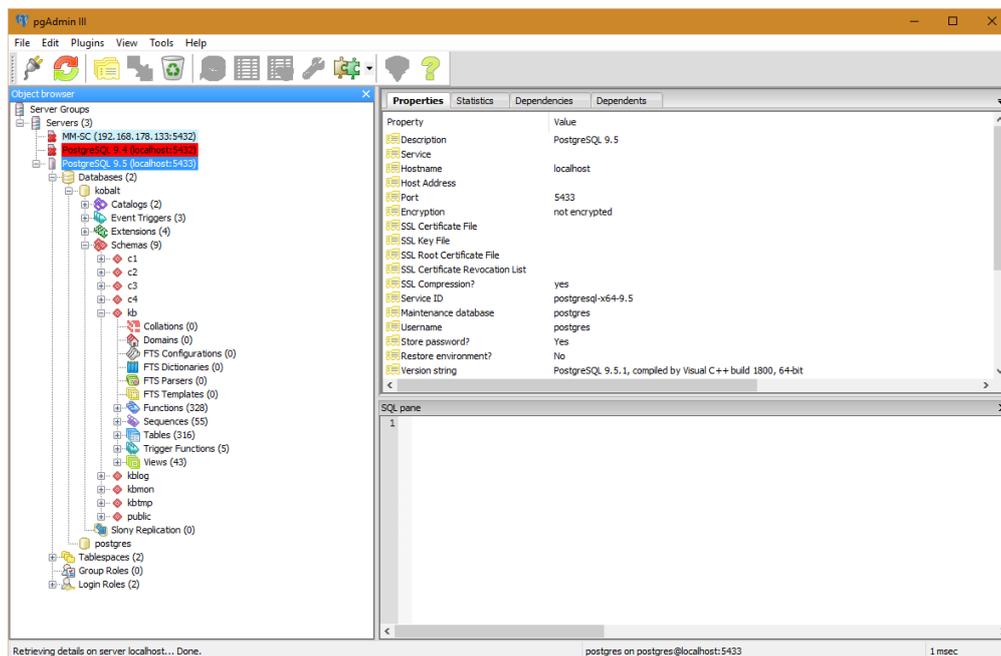


Figura II-9 - pgAdmin III

Esta ferramenta é também multiplataforma e *open-source*, incluída por omissão no pacote de instalação da base de dados do PostgreSQL. [17]

II.4.3. HTML e JavaScript

O HTML (HyperText Markup Language) é a linguagem de marcação estandardizada para a criação de páginas web. Juntamente com CSS e JavaScript é possível criar interfaces gráficas complexas que estão ao alcance de virtualmente qualquer navegador de internet e dispositivo, o que as torna uma escolha excelente para desenvolver as interfaces para a interação do utilizador com as aplicações e serviços em questão. [18]

As interfaces gráficas para o KobaltC são criadas sobre a biblioteca Javascript Knockout, que é *open-source* e livre para utilizar, totalmente desenvolvida em Javascript que a torna compatível com qualquer navegador e frameworks web e utiliza o padrão MVVM (model-view-view model). [19]

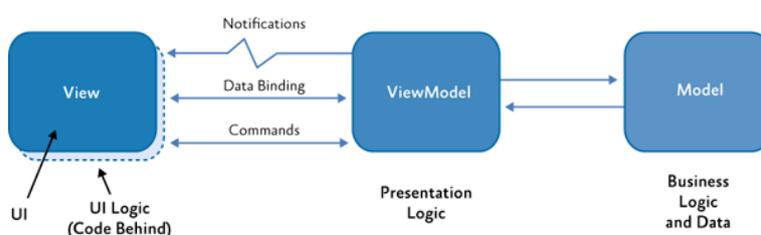


Figura II-10 - Representação do MVVM [20]

II.4.4. JSON

JSON (JavaScript Object Notation) é um formato de troca de dados fácil de ler e escrever para humanos, e também fácil para analisar e gerar para máquinas. É um formato de texto independente do idioma que utiliza convenções familiares aos programadores de inúmeras linguagens e é construído sobre estruturas de dados universais, sendo:

- Uma coleção de pares nome/valor;
- Uma lista ordenada de valores.

Estas estruturas fazem parte de quase todas as linguagens de programação modernas pelo que torna este formato ideal para troca de dados. [21]

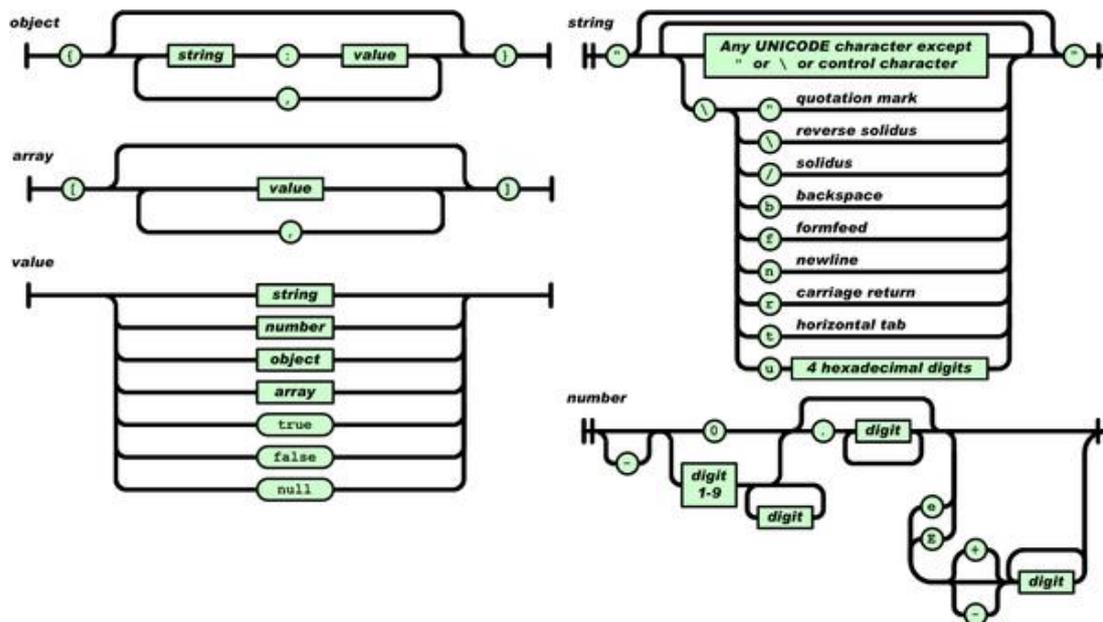


Figura II-11 - Estrutura do formato de dados JSON [21]

II.4.5. TortoiseSVN e Mantis BugTracker

O TortoiseSVN é um sistema de controlo de versão desenvolvido para Windows, com licença livre para uso e que providencia todas as funcionalidades esperadas e necessárias nestes tipos de sistemas, como um bom controlo do histórico do desenvolvimento, facilitar o trabalho em equipa, assim como obter versões anteriores por estarem marcadas como estáveis ou para depuração. [22]



Figura II-12 - Logotipo TortoiseSVN [22]

O Mantis BugTracker é uma ferramenta, baseada em web, criada com a finalidade de permitir colaboração entre as diversas entidades relacionadas, clientes e colaboradores da empresa, na deteção e gestão de falhas encontradas no projeto em questão. Embora a deteção e gestão de falhas seja a principal finalidade, esta ferramenta pode ser personalizada para que sirva como uma ferramenta de gestão de incidências e de projetos. [23]



Figura II-13 - Logotipo Mantis BugTracker [23]

III. DESENVOLVIMENTO DO SISTEMA

III.1. PROJETO

O projeto proposto é a concetualização e implementação de um sistema que funcione como uma interface de dados entre lojas na web para eCommerce e sistemas ERP. O principal propósito deste módulo será providenciar uma sincronização automática, controlada, eficiente e correta dos dados entre os diferentes sistemas.

O módulo SHOP está nos planos da empresa desde o início do planeamento e desenvolvimento do KobaltC. Devido à priorização dos projetos e recursos humanos limitados, este nunca havia sido implementado, embora fosse publicitado como um módulo disponível num futuro próximo. A priorização deste projeto elevou-se dado o interesse que os clientes, correntes e potenciais mostram por um sistema destes integrado na solução KobaltC.

Os clientes da MMSC são empresas cujos negócios se baseiam na venda de produtos e serviços, próprios ou de terceiros. Estas empresas utilizam um sistema ERP para gerir os seus processos de negócio, como a gestão do armazém, o tratamento de encomendas, entre outros. De forma a dar a mais visibilidade aos seus negócios, muitas destas empresas optam por também ter uma ou mais lojas virtuais a servir de montra para os seus produtos e serviços.

As lojas e o sistema ERP são geridos de uma forma independente entre si, o que dificulta a gestão dos dados destes sistemas. Um dos problemas mais recorrentes é o dos dados estarem desatualizados e, por vezes, incorretos, que podem trazer prejuízos enormes e até problemas com a lei. Um exemplo disto será quando o valor de um artigo mostrado ao cliente é muito inferior ao real, e por lei, o comerciante tem de aceitar a venda por esse valor, mesmo que este valor errado tivesse já sido retificado. Uma situação destas trará certamente prejuízos financeiros e de imagem para com os clientes e/ou parceiros. Esta será uma das diferentes situações mais graves que o módulo SHOP terá de prevenir.

Uma das maiores dificuldades de implementar um sistema assim encontra-se no facto de diferentes clientes/empresas utilizarem diferentes lojas e/ou sistemas ERP, cujas estruturas de dados e forma de interação com os mesmos diferem, inclusivamente na forma como estes consomem os mesmos dados. Uma possível solução para este

problema é a criação de interfaces especializadas para a troca de dados entre sistemas específicos, aliás, esta foi a solução adotada para o cumprimento do primeiro objetivo deste estágio. Mas esta solução só se torna viável se o cliente a financiar ou se a interface em questão for requisitada em número suficiente. Dito isto, o ideal é uma interface capaz de trocar dados com diversos sistemas diferentes sem a necessidade de ter de alterar a implementação ou, pelo menos, minimizar e facilitar estas alterações.

III.2. METODOLOGIAS DE TRABALHO

III.2.1. As 5 dimensões de um projeto de software

Na gestão de um projeto, há 5 dimensões a ter em conta: funcionalidades, qualidade, custo, tempo e equipa. Estas dimensões não são independentes entre si e na gestão de um projeto há que ter em conta que, por exemplo, se a equipa for aumentada poderá levar a uma redução do tempo, mas ao mesmo tempo a um custo mais elevado. [24]

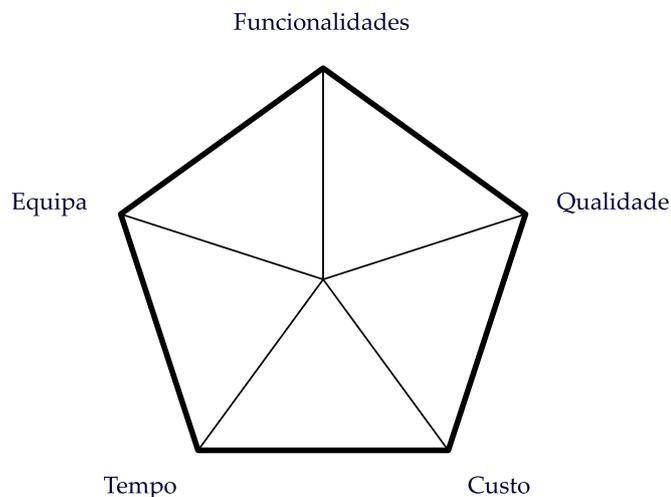


Figura III-1 - As 5 dimensões de um projeto de software

Cada uma destas dimensões pode assumir um de três papéis: controlador, restrição ou grau de liberdade. Um controlador é um objetivo principal do projeto, por exemplo, num projeto que deve ser concluído a tempo de aproveitar uma janela de oportunidade de negócio, o controlador será o tempo. Uma restrição é um fator limitador fora do controlo do líder de projeto, por exemplo, num projeto que a equipa para o mesmo esteja já definida, sem a possibilidade de adicionar ou remover elementos, a equipa será uma restrição. Por vezes um controlador do projeto pode ser visto como uma restrição caso este seja inegociável. As dimensões não sejam um controlador ou uma restrição tornam-se um grau de liberdade, isto é, o líder de projeto pode ajustar estas dimensões até um certo nível de forma a alcançar os objetivos globais do projeto. [24]

Um aspeto importante deste modelo não é quais das 5 dimensões são controladores ou restrições, mas sim a negociação das prioridades das dimensões do

projeto antecipadamente pela equipa do projeto. As 5 dimensões não podem ser todas controladores, assim como, também não podem ser todas restrições. Esta negociação ajuda a definir as regras e limitações do projeto. [24]

Uma forma de classificar as dimensões em uma das categorias é pensar na flexibilidade que o líder de projeto tem para cada dimensão. Uma restrição oferece ao líder de projeto pouca a nenhuma flexibilidade, um controlador oferece pouca flexibilidade e um grau de liberdade oferece a maior flexibilidade. Uma forma gráfica para apresentar a flexibilidade do projeto é através de um diagrama de Kiviat. Neste diagrama temos 5 eixos, um para cada dimensão, com uma escala arbitrária entre 0 e 10 (0 significa pouco flexível e 10 muito flexível). Em cada eixo é marcado um ponto no valor de flexibilidade determinado para a dimensão respetiva. [24]

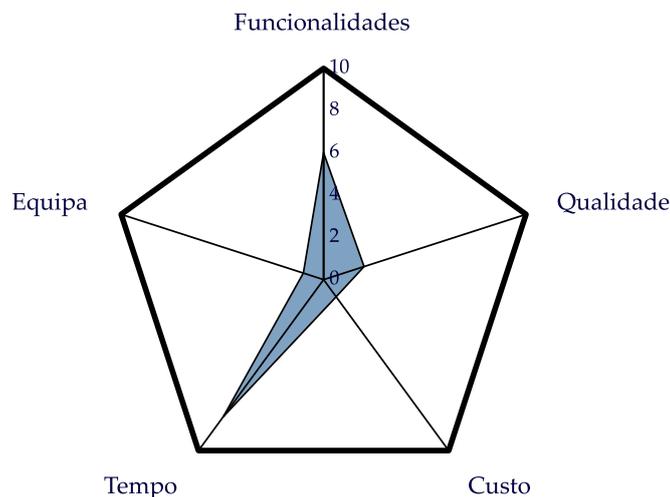


Figura III-2 - Diagrama de flexibilidade para este projeto

A figura anterior apresenta o diagrama de flexibilidade para o projeto do estágio. O sistema a desenvolver vai processar e atualizar dados nos sistemas dos clientes e tem de ser assegurado que estes estão corretos, pelo que o controlador deste projeto é a qualidade das funcionalidades implementadas. Neste projeto temos duas restrições: a equipa, que é constituída por mim, e o custo. Para o desenvolvimento deste projeto não há um prazo fixo e tendo controlo sobre as funcionalidades a implementar, as dimensões tempo e funcionalidades têm uma maior flexibilidade.

III.3. REQUISITOS DO SISTEMA

O processo de apurar, analisar, documentar e verificar as funcionalidades e restrições é denominado por engenharia de requisitos. Os requisitos de um sistema de software consistem de declarações do que é pretendido para o sistema em termos das funcionalidades do sistema, do seu comportamento, das suas propriedades e qualidades, assim como das suas restrições. [25]

Os requisitos de um sistema são geralmente classificados como requisitos funcionais ou requisitos não funcionais. Um requisito funcional é uma declaração de como os serviços providenciados pelo sistema devem funcionar, da forma que o sistema deve reagir a estímulos particulares e como se deve comportar em determinadas situações. Em alguns casos pode declarar o que o sistema não deve fazer. Um requisito não funcional é uma declaração de restrições ou funcionalidades providenciadas pelo sistema. Geralmente, estes requisitos aplicam-se ao sistema como um todo invés de a funcionalidades ou serviços específicos. [25]

O levantamento dos requisitos para este sistema foi efetuado com base nas funcionalidades requisitadas e definidas para o protótipo a desenvolver, no discutido em diversas reuniões com vários membros da empresa e com o decorrer do desenvolvimento do sistema.

III.3.1. Requisitos funcionais

Requisitos	Prioridade
RF1: O sistema deve suportar a troca de dados entre sistemas ERP e lojas na Web, numa relação de 1 para n .	Alta
RF2: O sistema deve suportar a troca de dados entre sistemas ERP distintos, numa relação de 1 para n .	Baixa
RF3: O sistema deve suportar a troca de dados entre lojas na Web distintas, numa relação de 1 para n .	Baixa
RF4: O sistema deve suportar a troca de dados de um recurso em ambas as direções quando disponível.	Alta

RF5: O sistema deve permitir ao administrador efetuar um <i>ping</i> à loja para determinar se está acessível.	Média
RF6: O sistema deve ser capaz de listar um ou mais elementos das lojas na Web e dos sistemas ERP.	Alta
RF7: O sistema deve ser capaz de criar novos elementos nas lojas na Web e nos sistemas ERP.	Alta
RF8: O sistema deve ser capaz de atualizar os dados dos elementos nas lojas na Web e nos sistemas ERP.	Alta
RF9: O sistema deve ser capaz de remover elementos das lojas Web. Esta funcionalidade deve ser somente permitida com a supervisão de um administrador.	Alta
RF10: O sistema não deve permitir remover elementos dos sistemas ERP.	Alta
RF11: O sistema deve iniciar a sincronização dos dados dos elementos quando alterações nos dados destes elementos é detetada.	Alta
RF12: O sistema deve iniciar sincronização de dados através de tarefas escalonadas.	Alta
RF13: O sistema deve manter um histórico dos dados dos elementos.	Média
RF14: O sistema deve manter um histórico das transferências de dados com as lojas na Web.	Alta
RF15: O sistema deve ser capaz de detetar novos elementos nas lojas na Web e nos sistemas ERP para sincronizar.	Alta
RF16: O sistema deve ser capaz de identificar quais os campos dos elementos que sofreram alterações.	Média
RF17: O sistema deve ser capaz de atualizar apenas os campos dos elementos que sofreram alterações.	Média
RF18: O sistema deve permitir ao administrador selecionar os campos dos elementos que devem ser sincronizados.	Alta

RF19: O sistema deve sincronizar apenas os elementos que cumpram os critérios pré-definidos. O administrador deve ser capaz de alterar os critérios para a sincronização.	Alta
RF20: O sistema deve alertar o administrador por email quando uma loja na Web ou sistema ERP estão incontactáveis após um determinado número de tentativas.	Média
RF21: O sistema deve detetar valores indesejados ou errados nos dados dos elementos.	Alta
RF22: O sistema deve atuar com ações e/ou alertas quando a situação do requisito anterior se verifica.	Alta
RF23: O sistema deve ter a capacidade de mapear valores entre os diferentes sistemas.	Alta
RF24: O administrador deve ser capaz de atuar manualmente no sistema.	Média
RF25: O sistema deve suportar importação e exportação em massa de dados.	Baixa
RF26: O sistema deve ser capaz de interagir com as lojas na web através do protocolo REST.	Alta
RF27: O sistema deve ser capaz de interagir com as lojas na web através do protocolo SOAP.	Média
RF28: O sistema deve suportar a troca de dados através do formato JSON.	Alta
RF29: O sistema deve suportar a troca de dados com os sistemas ERP, Büro+ e ERP Complete.	Alta
RF30: O sistema deve suportar a troca de dados com a loja Shopware.	Alta
RF31: O sistema deve suportar a troca de dados com a loja Magento.	Média
RF32: O sistema deve de poder comunicar com sistemas de transporte de encomendas e requerer números de expediente.	Média

RF33: O sistema deve permitir portar os dados contidos na base de dados entre diferentes instalações do KobaltC.	Média
RF34: O sistema deve ser capaz de utilizar dados portados de outra instalação do KobaltC.	Alta
RF35: O sistema deve permitir criar relatórios.	Média

Tabela III-1 - Requisitos funcionais

III.3.2. Requisitos não funcionais

Requisitos	Prioridade
RNF1: O sistema deve ser desenvolvido na linguagem C++.	Alta
RNF2: A base de dados a utilizar deverá ser o PostgreSQL.	Alta
RNF3: O sistema deve disponibilizar uma interface gráfica integrada no KobaltC WebClient para facilitar a administração do sistema.	Baixa
RNF4: O sistema deve poder sincronizar os dados com diferentes sistemas de simultaneamente.	Baixa
RNF5: O sistema deve facilitar a criação do suporte à sincronização de qualquer recurso disponível nas lojas na Web e sistemas ERP.	Alta
RNF6: O sistema deve facilitar a criação de suporte a novas lojas na Web e sistemas ERP.	Alta
RNF7: O sistema deve poder aplicar alterações aos parâmetros para a sincronização dos dados de uma forma hierárquica. Os parâmetros podem ser aplicados a elementos individuais, a todos os elementos de um recurso específico ou a todos os elementos de uma loja na Web ou sistema ERP.	Média
RNF8: O sistema deve permitir ao administrador alterar os parâmetros mais relevantes de funcionamento do sistema.	Média
RNF9: O sistema deve ser capaz de retomar as tarefas em progresso caso haja uma falha da máquina onde o sistema corre.	Alta

<p>RNF10: O sistema deve suportar uma execução distribuída, isto é, diferentes máquinas devem ser capazes de trabalhar nos mesmos dados.</p>	<p>Baixa</p>
<p>RNF11: O sistema deve disponibilizar a sincronização de artigos, de categorias de artigos, de imagens de artigos, de clientes e de encomendas por omissão para todas as lojas na Web e sistemas ERP suportados pelo sistema.</p>	<p>Baixa</p>

Tabela III-2 - Requisitos não funcionais

III.4. CASOS DE UTILIZAÇÃO

Um diagrama de casos de utilização é um modelo com o intuito de mostrar, de uma forma simples, uma visão geral das interações dos diferentes atores com o sistema. Cada caso de utilização representa uma tarefa que envolve algum tipo de interação externa com o sistema por parte de atores. Os atores representam papéis de utilizadores ou de sistemas externos que interagem com o sistema. [25]

III.4.1. Atores

Os atores identificados para o sistema foram:

Serviço – serviço responsável pela sincronização dos dados entre as diferentes lojas.

Administrador – responsável pela gestão do sistema que inclui configurar o sistema e agir sobre alertas e relatórios gerados pelo sistema.

Utilizador – utilizador que consulta relatórios e recebe alguns alertas gerados pelo sistema.

Loja – sistema externo com o qual efetuar sincronização de dados. Aqui estão englobadas lojas na Web e sistemas ERP.

Distribuidor – sistema de distribuição que providencia números de expediente para encomendas.

III.4.2. Diagrama de casos de utilização

De seguida temos o diagrama de casos de utilização, representando os casos que foram considerados como os principais para a compreensão do funcionamento do sistema.

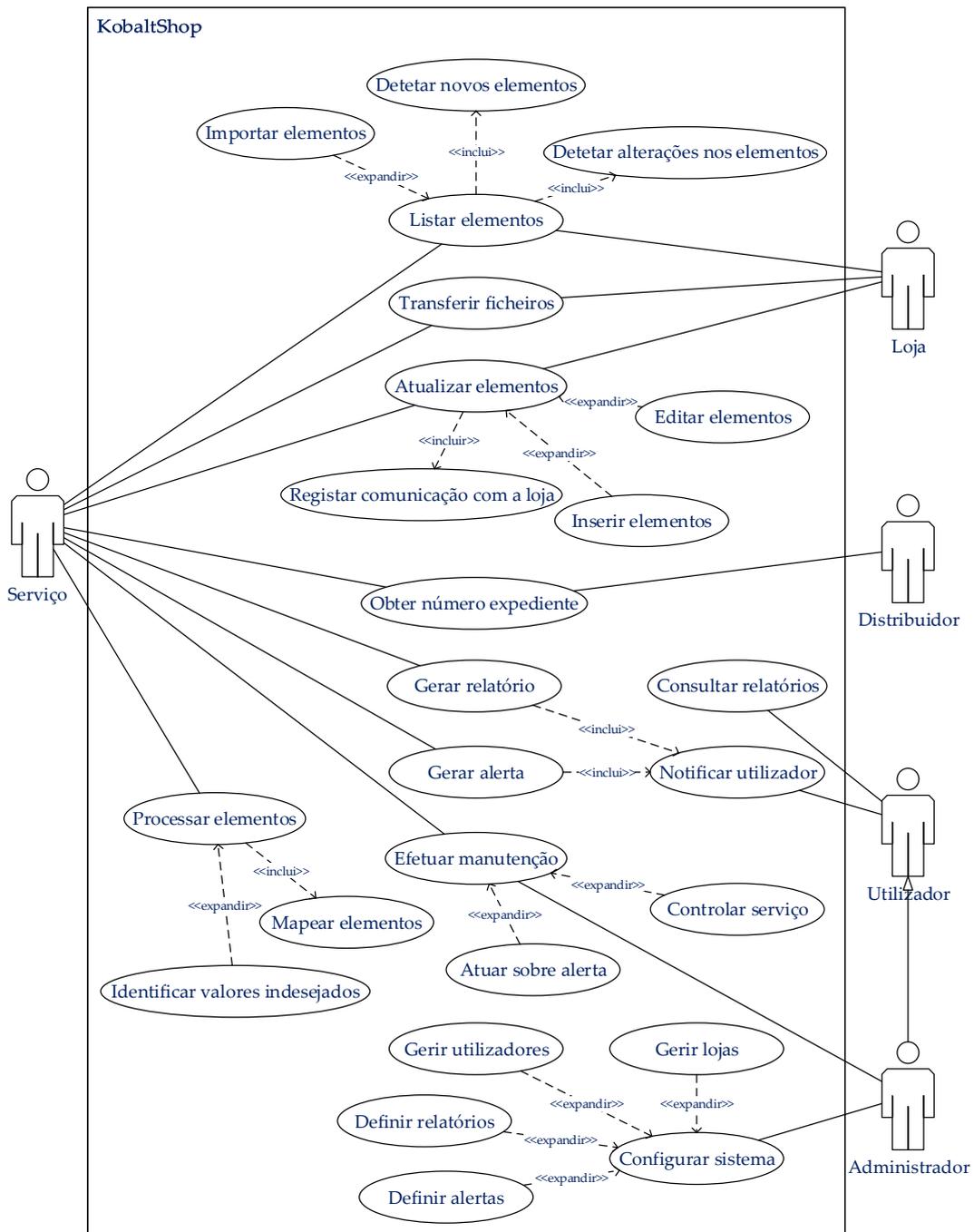


Figura III-3 - Diagrama de casos de utilização

De forma a auxiliar a compreensão do diagrama, foi feita uma descrição de alguns destes casos de utilização no ponto que se segue.

III.4.3. Descrição dos casos de utilização

Listar elementos

Atores	Serviço, loja
Descrição	<p>O serviço efetua um pedido à loja por uma lista de elementos que correspondam aos parâmetros para este pedido, como por exemplo, uma lista de todas as encomendas recebidas nos últimos 15 minutos.</p> <p>O serviço recebe a lista e determina para cada um dos elementos se estes os seus dados apresentam alterações ou os elementos são novos para o sistema. Se sim, os elementos são importados para o sistema e marcados para processamento posterior pelo serviço.</p>
Estímulos	Pedido efetuado pelo serviço despoletado por escalonamento, por um evento do sistema ou por uma instrução emitida pelo administrador.
Resposta	Uma lista com zero, um ou mais elementos.
Comentários	O caso de utilização “Processar elementos” realiza o processamento posterior acima referido.

Tabela III-3 - Descrição do caso de utilização "Listar elementos"

Atualizar elementos

Atores	Serviço, loja
Descrição	<p>O serviço efetua um pedido à loja para a atualização dos dados de um elemento na loja. Se o dito elemento não existir na loja, este será criado. Após a resposta indicando sucesso na operação por parte da loja, é criado um registo desta comunicação com a loja.</p>
Estímulos	Pedido efetuado pelo serviço despoletado por um evento do sistema ou por uma instrução emitida pelo administrador.
Resposta	Confirmação por parte da loja que o elemento foi atualizado e o <i>id</i> do elemento na loja.

Comentários	Os elementos aqui afetados foram previamente marcados, no caso de utilização “Processar elementos”, para serem atualizados na loja respectiva.
--------------------	--

Tabela III-4 - Descrição do caso de utilização "Atualizar elementos"

Processar elementos

Atores	Serviço
Descrição	<p>O serviço tem acesso a uma lista de elementos que foram marcados para processamento. Este processamento é para determinar se o elemento deve ser sincronizado com as lojas respectivas.</p> <p>Um por um, o serviço inicia o processamento destes elementos, que inclui determinar se estes elementos possuem valores indesejados.</p> <p>Quando for determinado que o elemento deve ser sincronizado, a estrutura dos dados do elemento será preparada e mapeada de acordo com a estrutura dos elementos das lojas alvo da sincronização. Os elementos são marcados para serem tratados pelo caso de utilização “Atualizar elementos”</p>
Estímulos	O serviço inicia esta tarefa através escalonamento ou através de um evento do sistema.
Resposta	Um código indicando o resultado do processamento que pode levar ao serviço a gerar um alerta
Comentários	No caso de o elemento possuir valores indesejados, o serviço irá corrigir os valores, ou irá marcar o elemento como inativo ou então gerar um alerta.

Tabela III-5 - Descrição do caso de utilização "Processar elementos"

Configurar sistema

Atores	Administrador
Descrição	<p>O administrador configura os parâmetros e regras para o funcionamento do sistema.</p> <p>O administrador pode adicionar e configurar as lojas entre as quais deve ser realizada a sincronização de dados, assim como definir as regras para a sincronização.</p> <p>O administrador pode também criar e gerir os utilizadores que terão acesso aos relatórios e alertas gerados pelo sistema. Estes relatórios e alertas podem também ser configurados pelo administrador.</p>
Estímulos	O administrador acede à área de configuração do sistema
Resposta	Confirmação que as alterações feitas à configuração do sistema foram aceites.

Tabela III-6 - Descrição do caso de utilização "Configurar sistema"

Efetuar manutenção

Atores	Administrador, serviço
Descrição	<p>O administrador pode ter a necessidade de efetuar manutenção no serviço, como o ter de atuar relativamente a um alerta gerado pelo serviço.</p> <p>O administrador pode ter também a necessidade de parar ou reiniciar o serviço.</p> <p>O serviço e o administrador podem iniciar tarefas de limpeza dos dados temporários e outras tarefas semelhantes com efeitos de melhorias de desempenho e salvaguardar recursos do sistema físico.</p>
Estímulos	Instruções enviadas pelo serviço ou pelo administrador
Resposta	Confirmação que as instruções foram aceites pelo sistema

Tabela III-7 - Descrição do caso de utilização "Efetuar manutenção"

III.5. DIAGRAMAS DE ATIVIDADE

Nesta secção temos alguns diagramas de atividade de alguns dos casos de utilização acima elaborados. Diagramas de atividade pretendem demonstrar as atividades que compõem um processo do sistema e o fluxo de controlo entre cada atividade, podendo também demonstrar como dados são processados no processo. [25]

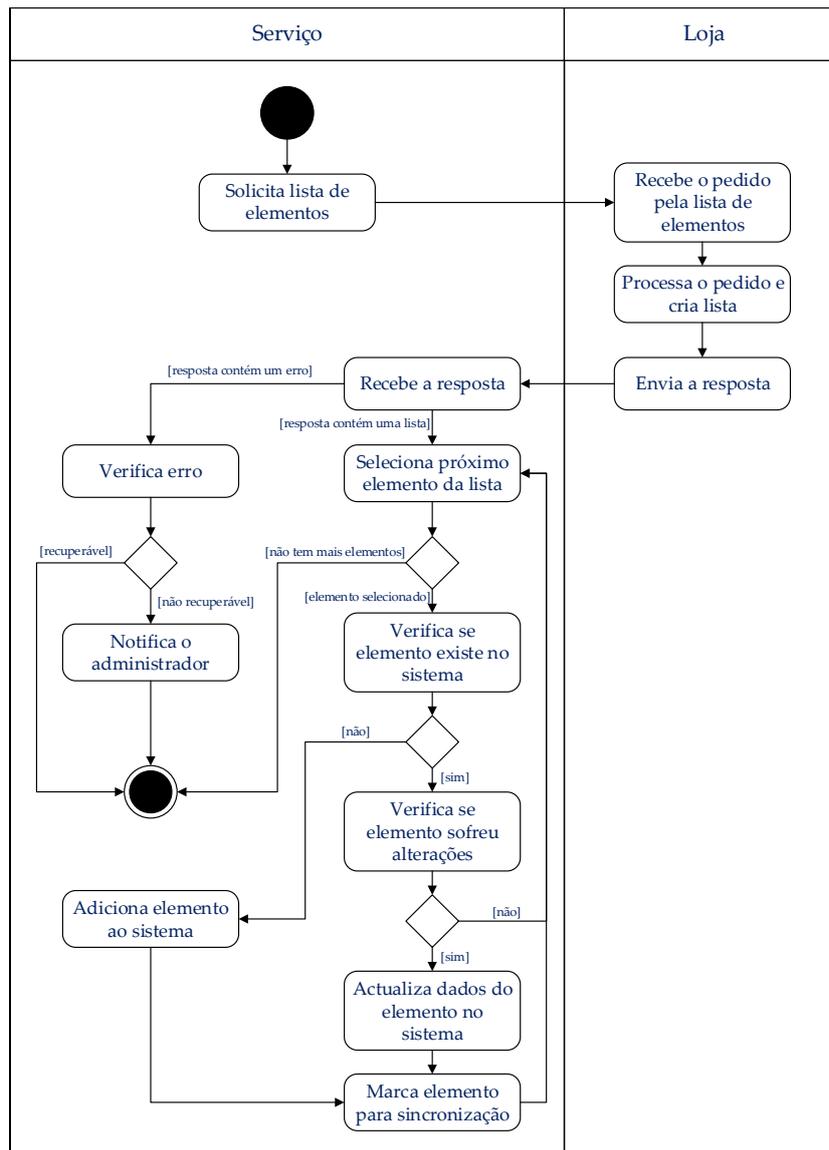


Figura III-4 - Diagrama de atividade para o caso "Listar elementos"

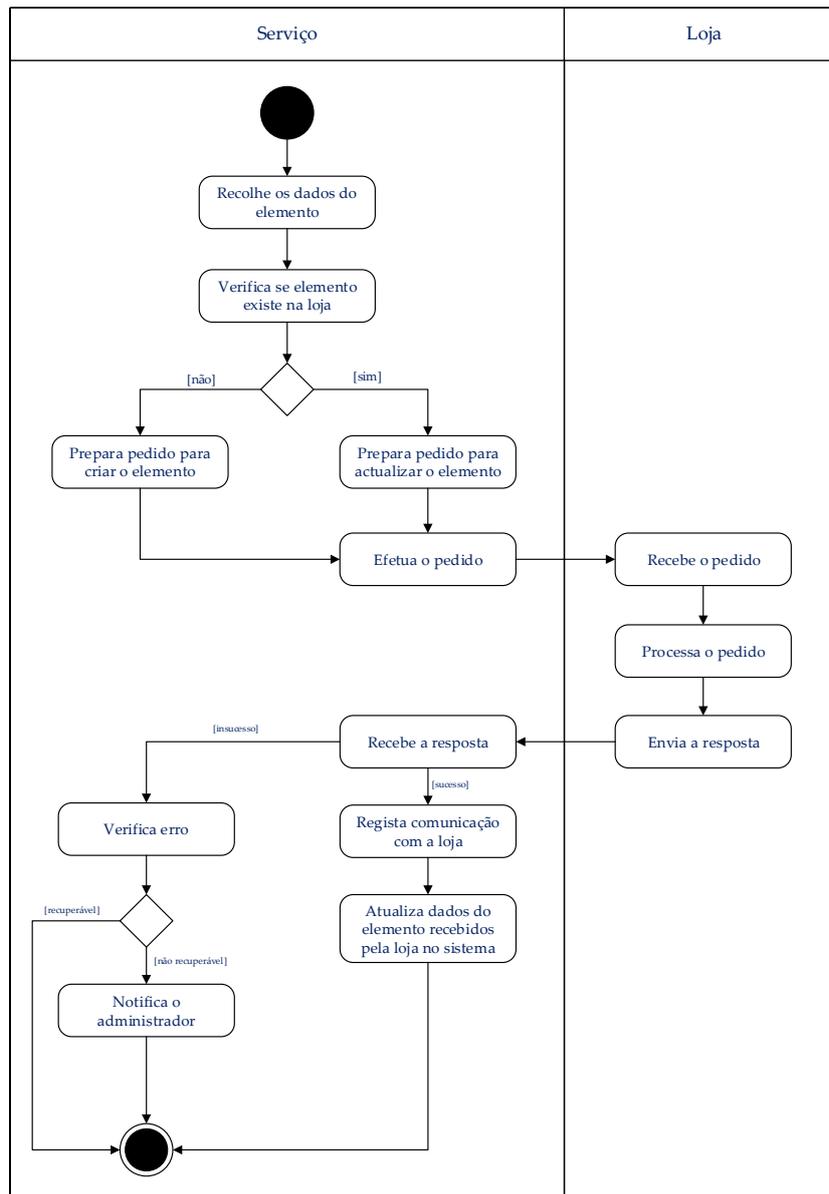


Figura III-5 - Diagrama de atividade para o caso "Atualizar elementos"

O registo da comunicação é um passo crítico do caso “Atualizar elementos”, sobretudo, para apurar responsabilidades de problemas com os dados de elementos na loja. Apesar de o sistema estar responsável pelos dados dos elementos, este está impotente a alterações nos dados dos dados diretamente na loja por parte de outros utilizadores.

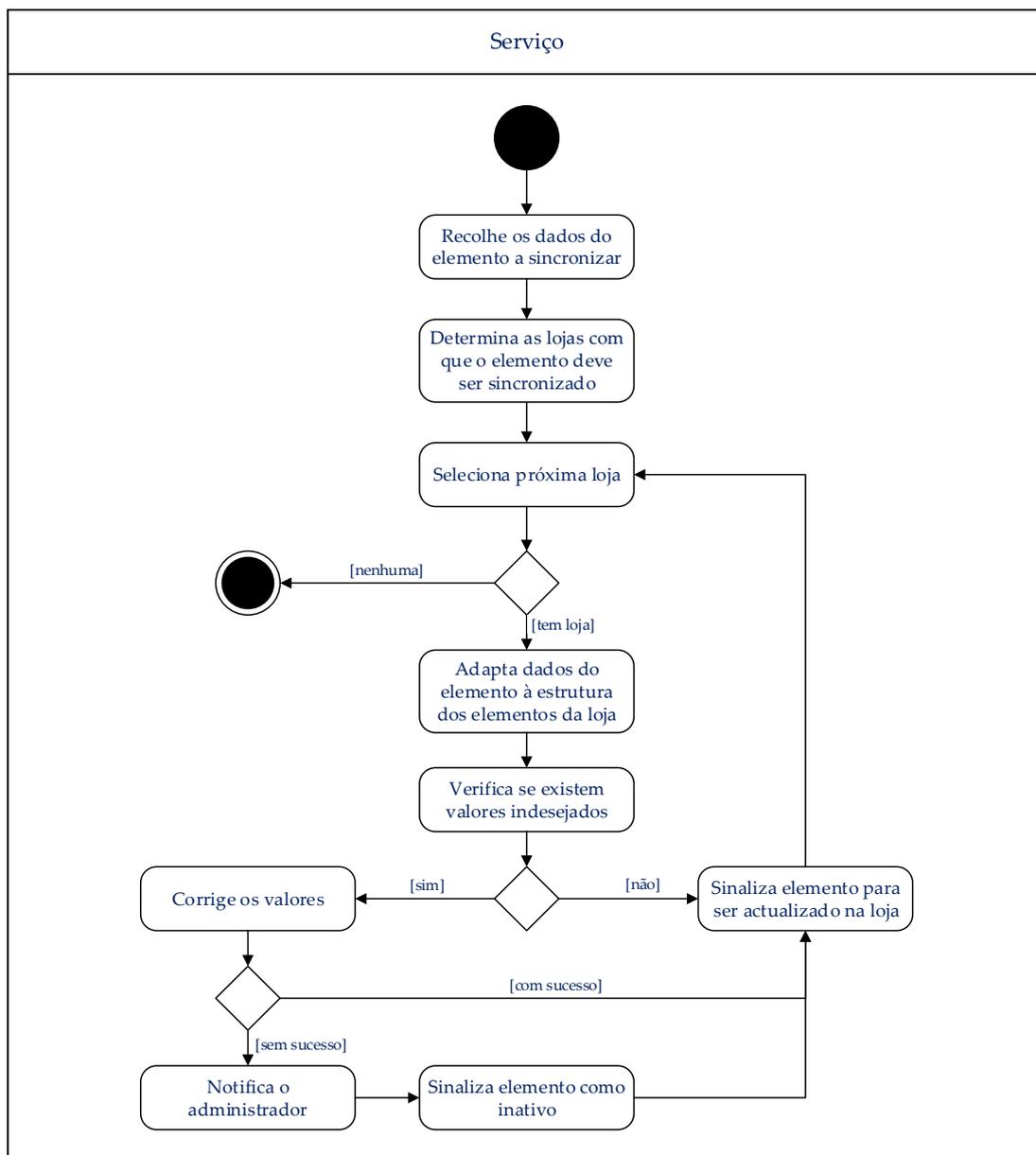


Figura III-6 - Diagrama de atividade para o caso "Processar elemento"

No caso de valores indesejados, como por exemplo, o preço de um produto ser zero, se o sistema não conseguir corrigir o valor, o administrador tem de ser notificado para este determinar o motivo deste valor.

O elemento ser sinalizado como inativo pode ser quer para sincronizações futuras até intervenção do administrador ou desativar ou esconder o elemento na loja para que nenhum utilizador da loja o consiga seleccionar com valores errados.

III.6. SOLUÇÃO TÉCNICA

De forma a satisfazer os requisitos para o sistema a níveis de robustez e expansibilidade cheguei à solução que vou apresentar de seguida.

De acordo com os requisitos **RF1**, **RF2**, **RF3**, **RNF5** e **RNF6** é pretendido um sistema que permita a troca de dados entre diversos sistemas externos, e que facilite adicionar o suporte à troca de dados para novos sistemas. A solução apresentada para satisfazer estes requisitos é utilizar generalização nos sistemas externos e nos recursos que estes fornecem.

A generalização é uma técnica onde se determina quais as características comuns entre as diferentes classes presentes no sistema de forma a criar uma estrutura hierárquica, de forma a manter as informações comuns nas classes mais acima na hierarquia e as especializações nas subclasses na hierarquia. [25]

Para o nosso sistema foram identificadas as características em comum entre os diferentes sistemas externos que são sistemas ERP e lojas na Web. Uma das características em comum é a disponibilidade de uma API² que permite ao nosso sistema a recolha e atualização dos dados do sistema externo em questão. Embora estas interfaces possam ser baseadas em tecnologias diferentes, permitem o mesmo, que é listar, inserir, editar e até mesmo remover dados ou elementos dos sistemas em questão.

Outra característica em comum encontrada é o facto destes sistemas terem os seus dados estruturados de formas semelhantes. Isto é, os dados relativos a artigos estão disponibilizados como o recurso artigo, sucedendo o mesmo para os dados relativos a encomendas, a clientes e por assim em diante. Nestes sistemas as estruturas dos dados são diferentes, embora se refiram a [14] recursos semelhantes nos seus sistemas respetivos.

² API - Uma interface de programação de aplicações, ou API, é um conjunto de definições de métodos, protocolos e ferramentas que permitem aceder às funcionalidades de um programa. Isto quer dizer que estas funcionalidades podem ser diretamente acedidas por outros programas. [25]

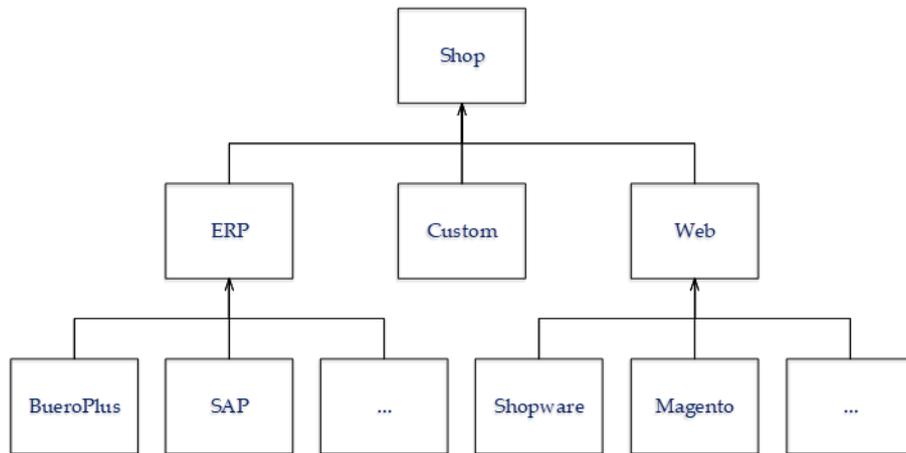


Figura III-7 - Diagrama da generalização das lojas

Assim sendo temos uma classe denominada por *Shop*, ou loja, que representa qualquer sistema externo que possui uma API e uma lista de recursos disponíveis. Os sistemas ERP estão representados na subclasse *ERP* e as lojas na Web na subclasse *Web*. Cada sistema será então uma subclasse destes anteriores que contém as suas especializações em relação às restantes classes.

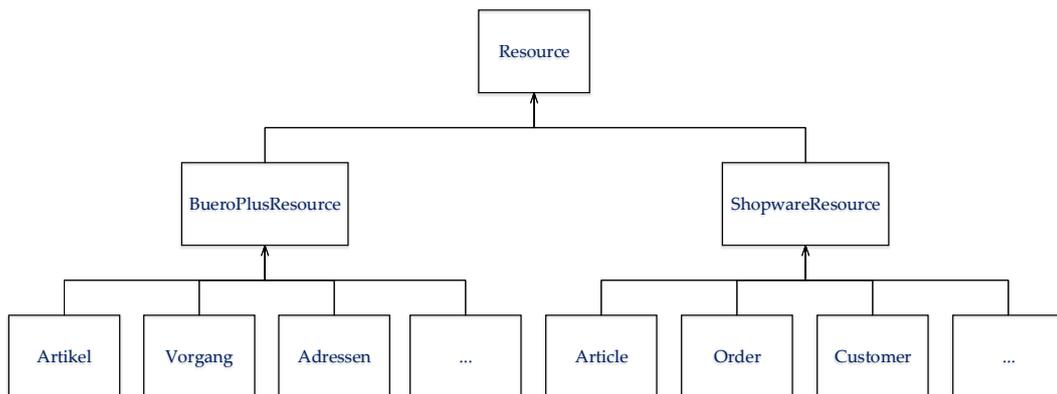


Figura III-8 - Diagrama da generalização dos recursos

Uma hierarquia semelhante existe para os recursos, representada na figura anterior. Nesta figura temos o exemplo dos recursos que pertencem às lojas *BueroPlus* e *Shopware* respetivamente. Os recursos das restantes lojas seguem a mesma estrutura.

Esta solução pretende facilitar a criação do suporte a novas lojas na medida que para cada nova loja e seus recursos é necessário apenas implementar as suas especializações, facilitando assim a manutenção do sistema.

III.7. ESTRUTURA DA BASE DE DADOS

De seguida é apresentada a solução para a estrutura da base de dados modelada para este projeto, consoante os requisitos levantados para o projeto. O módulo *Shop* pertence ao ecossistema *KobaltC* e alguns dos requisitos para o módulo estão já implementados noutros serviços. As tabelas que suportam estes serviços não serão apresentadas a pedido do engenheiro Melchert.

III.7.1. Visão geral

Nas tabelas do ecossistema *KobaltC* há uma regra de nomenclatura a seguir, em que todas as tabelas específicas para um módulo devem partilhar o mesmo prefixo. No caso do módulo *SHOP*, o prefixo é *shop*.

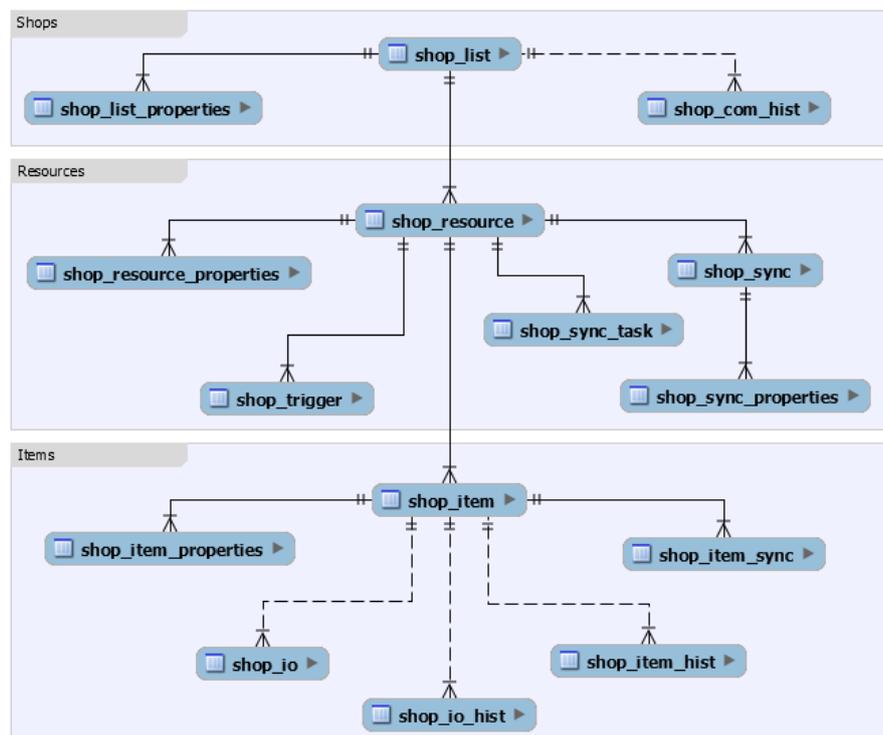


Figura III-9 - Tabelas principais da estrutura

Na figura anterior temos uma representação das entidades principais da estrutura da base de dados para o sistema. A entidade **shop_list** representa todas as n lojas configuradas no sistema, sendo estas lojas na Web e sistemas ERP. Para cada uma destas lojas temos n recursos (**shop_resource**) disponibilizados pela loja respetiva. Cada recurso pode ser sincronizado com outros n recursos, cujas regras de

sincronização entre os recursos são definidas em **shop_sync**. Por fim temos os elementos (**shop_item**), onde cada um destes n pertence a um único recurso.

De seguida temos uma descrição mais específica de todas as entidades constituintes da estrutura.

III.7.2. Serviço

O esquema que se segue, apresenta as tabelas necessárias para a configuração dos parâmetros do serviço e controlo do mesmo. A estrutura é idêntica à dos restantes serviços do ecossistema *KobaltC*, com uma tabela para a configuração dos parâmetros do serviço e três tabelas para gerir a execução e histórico das tarefas do serviço.



Figura III-10 - Tabelas para gestão do serviço

A tabela **shop_properties** armazena os parâmetros que controlam a execução do serviço. Os parâmetros aqui definidos são válidos em todo o serviço.

As tabelas **shop_joblist**, **shop_joblist_act** e **shop_joblist_hist** contêm, respetivamente, as tarefas que o serviço deve executar, as tarefas a serem executadas no momento e o histórico das tarefas já executadas. Esta estrutura assegura que as tarefas não são perdidas quando o serviço é encerrado (**shop_joblist**), permite retomar ou repetir as tarefas que não foram terminadas (**shop_joblist_act**) e mantém um histórico que pode ser usado para estatísticas e procura de falhas (**shop_joblist_hist**).

O serviço verifica periodicamente se existem tarefas da seguinte forma:

- verifica se existem tarefas na tabela **shop_joblist_act** e executa-as. Quando terminada esta será movida para a tabela **shop_joblist_hist**;
- de seguida, verifica na tabela **shop_joblist_hist** se existem tarefas a repetir. Se forem encontradas, estas tarefas são adicionadas à tabela **shop_joblist** e marcadas como repetidas;
- por fim, verifica na tabela **shop_joblist** se existem tarefas a executar, utilizando o princípio *FIFO*³ e ordenadas por prioridade. Estas tarefas são então movidas para a tabela **shop_joblist_act** e executadas. Quando terminadas, a tarefa é movida para **shop_joblist_hist**.

III.7.3. Lojas

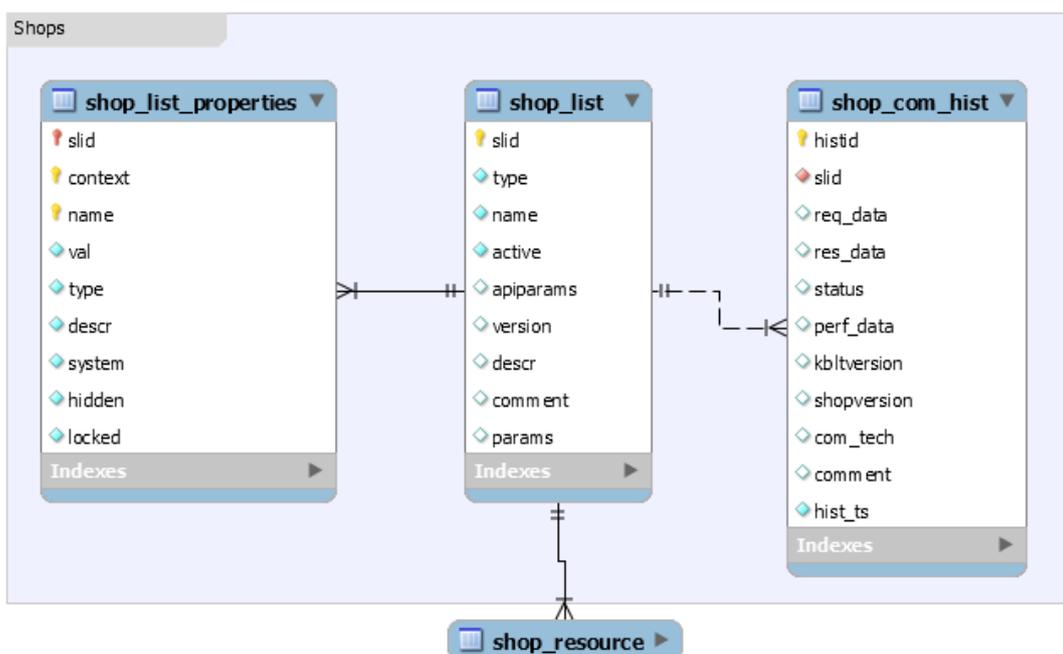


Figura III-11 - Tabelas para configuração das lojas

A tabela **shop_list** é a tabela principal, pois aqui estão armazenados os dados relativos a todas as lojas, independentemente do tipo de loja, com que o sistema deverá sincronizar. Esta tabela armazena dados como o tipo de loja, nome, versão da loja, descrição e outros. Os parâmetros para a comunicação do sistema com a loja são guardados no campo *apiparams*, que é texto numa estrutura JSON. Restantes

³ *FIFO* – first in first out, método de utilização dos dados de uma estrutura da forma análoga a uma fila, onde os dados são processados na ordem de chegada à estrutura

parâmetros únicos de cada loja serão guardados no campo *params*, também sob texto com uma estrutura JSON.

Todas as comunicações entre o sistema e as lojas que alterem dados nas lojas serão armazenadas na tabela **shop_com_hist**. Os dados incluem os parâmetros do pedido realizado à loja (*req_data*), a resposta da loja (*res_data*), versões do sistema e da loja aquando da comunicação e dados sobre o desempenho da comunicação.

Na tabela **shop_list_properties** serão guardados diversos parâmetros válidos no contexto desta loja. Os parâmetros aqui definidos sobrepõem os parâmetros definidos em **shop_properties** caso tenham o mesmo *context* e *name*.

III.7.4. Recursos

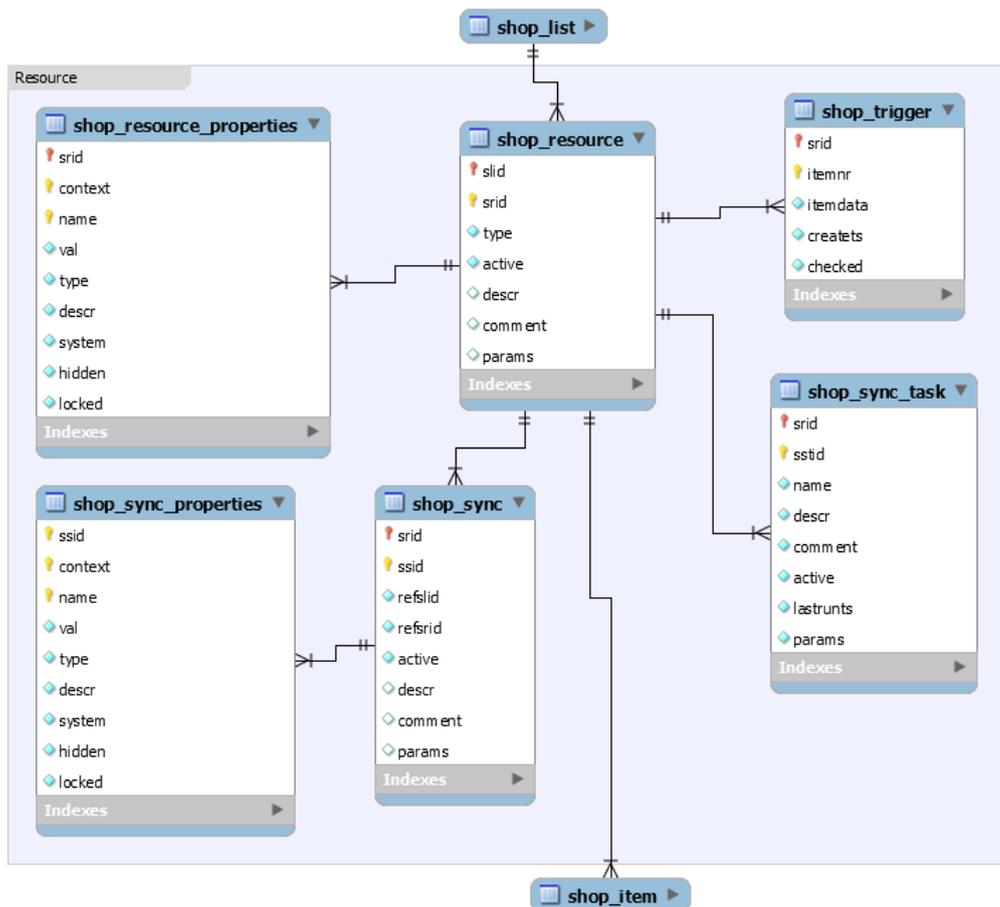


Figura III-12 - Tabelas para configuração da sincronização dos recursos das lojas

Na tabela **shop_resource** são armazenadas as configurações para cada recurso que cada loja oferece. Analogamente, os parâmetros específicos a cada tipo de recurso é armazenado no campo *params* numa estrutura JSON.

O serviço periodicamente verifica nas lojas se existem elementos a sincronizar, e estas tarefas estão definidas na tabela **shop_sync_task**. Cada recurso pode possuir mais que uma tarefa distinta para a recolha dos elementos da loja a sincronizar.

A tabela **shop_sync** armazena os dados da configuração que ditam como sincronizar os elementos do recurso em questão para o recurso alvo. Cada recurso pode ter mais que um recurso alvo, que implica uma configuração de sincronização. Os parâmetros e regras para a sincronização e mapeamento dos dados estão definidos na estrutura JSON que é guardada no campo *params*.

As tabelas **shop_resource_properties** e **shop_sync_properties** têm um propósito e funcionamento análogo à tabela **shop_list_properties**, no contexto dos recursos e das sincronizações dos recursos respetivamente.

Na tabela **shop_trigger** são inseridos os dados de elementos do recurso que o serviço deve averiguar se devem ou não ser processados, isto é, se reúnem as condições para serem sincronizados. Esta tabela serve como ponto de entrada para sincronizar todos os elementos.

III.7.5. Elementos

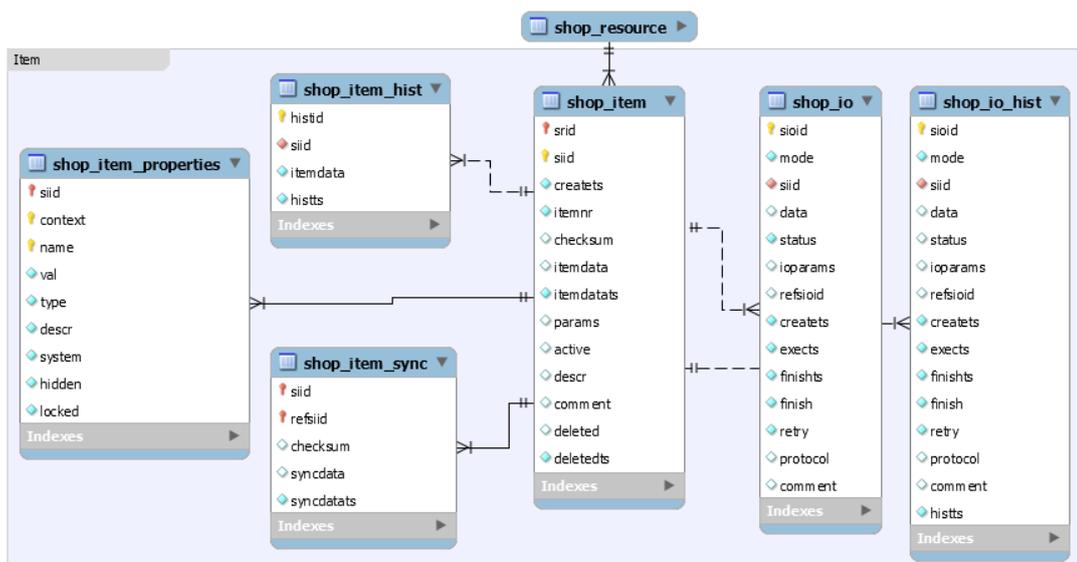


Figura III-13 - Tabelas para o tratamento dos elementos

A tabela **shop_item** armazena os dados e alguns parâmetros relativos aos elementos de cada recurso de cada loja. O campo *itemnr* guarda o número pelo qual o elemento é identificado na loja de origem.

O propósito e funcionamento da tabela **shop_item_properties** é análogo às restantes tabelas com nome terminado por *properties* mas no contexto do elemento.

A tabela **shop_item_hist** irá armazenar o histórico das alterações dos dados dos elementos, contendo os dados dos elementos na sua totalidade ou apenas os valores que se alteraram.

Na tabela **shop_item_sync** são armazenados os dados relativos à sincronização entre 2 elementos. Através desta tabela conseguimos descobrir com que elementos das lojas destino este elemento está a ser sincronizado, assim como os últimos dados sincronizados entre os 2 elementos.

As tabelas **shop_io** e **shop_io_hist** armazenam os dados relativos às tarefas de processamento dos elementos e histórico das mesmas.

III.7.6. Utilitários

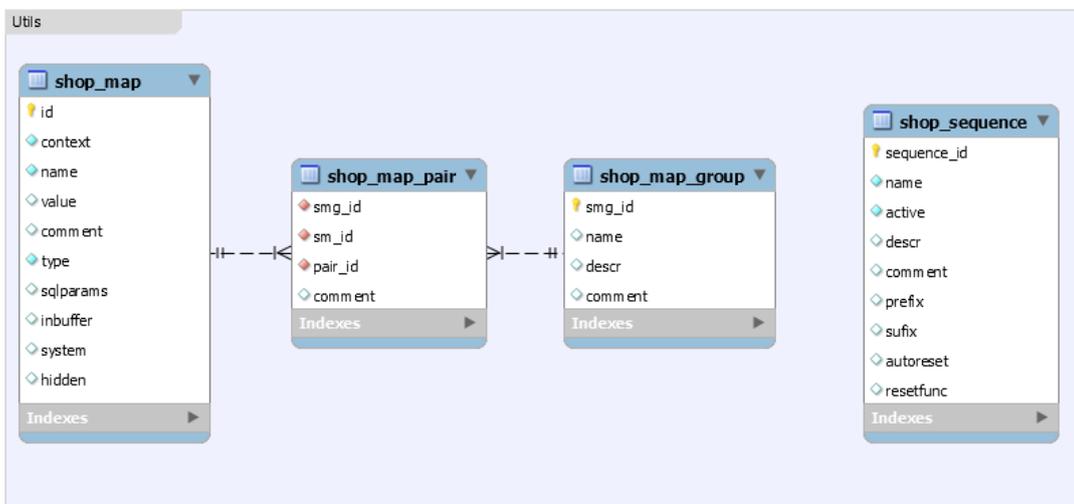


Figura III-14 - Tabelas para configuração do mapeamento de valores e sequências

Na tabela **shop_sequence** podemos definir sequências com prefixos e sufixos e parâmetros para a reposição das sequências.

As tabelas **shop_map**, **shop_map_pair** e **shop_map_group** permitem definir variáveis e configurar o mapeamento de valores entre as diferentes lojas.

III.8. INTERFACE DO UTILIZADOR

A interface do utilizador é importante para qualquer sistema, pois oferece aos utilizadores uma forma de interagirem com o sistema. Neste projeto, a interface oferece ao administrador uma interface mais apelativa para a configuração do sistema que a ferramenta de gestão da base de dados.



Figura III-15 - Página inicial do KobaltC WebClient

O ecossistema *KobaltC* possui uma aplicação web (*KobaltC WebClient*) que oferece aos administradores uma interface gráfica para a configuração do sistema, acessível a partir de qualquer dispositivo com um navegador de internet. A interface para o módulo *Shop* será também integrada nesta aplicação e que deverá seguir a mesma estrutura.

III.8.1. Estrutura da interface

A interface é composta por um cabeçalho, um corpo e um rodapé. No cabeçalho temos os dois menus principais da interface, que são dinâmicos e três botões estáticos. O corpo está dividido em três partes, um menu contextual à esquerda (minimizável), a área para a aplicação onde o utilizador pode interagir com os dados e uma área para ajuda à direita que se encontra escondida por omissão. O rodapé é uma barra de estado que apresenta mensagens do sistema, a aplicação corrente e a versão do sistema.

Os menus da aplicação são criados dinamicamente, consoante os serviços licenciados e ativos, as permissões do utilizador e os dados disponíveis para esse mesmo menu.

Na figura seguinte temos as partes mais importantes da estrutura da interface identificadas com as letras A, B, C e D e cores diferentes.

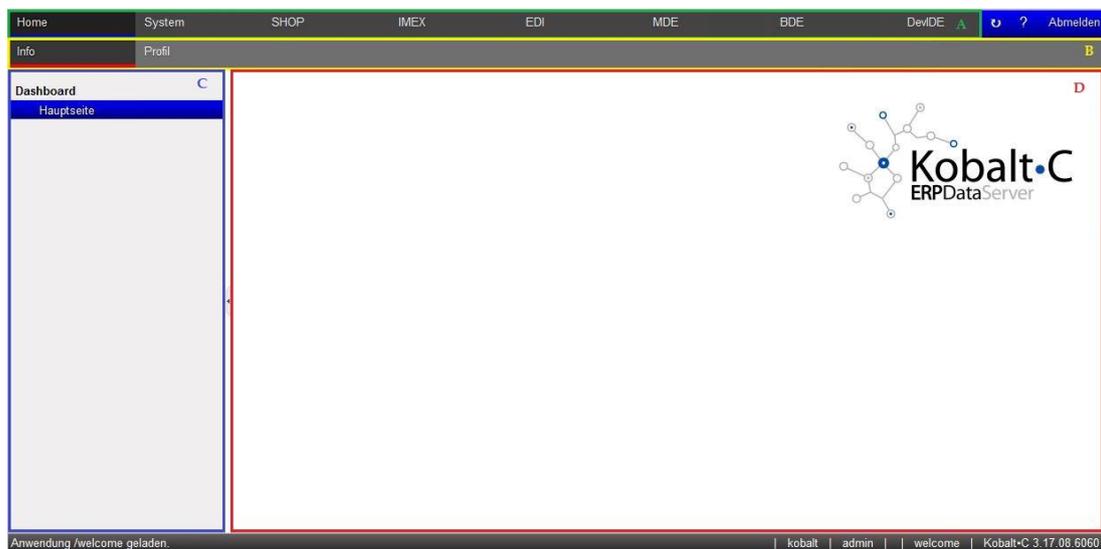


Figura III-16 – Estrutura da interface do utilizador

À direita do menu A temos três botões estáticos de sistema. As suas funcionalidades são, respetivamente, refrescar os menus da interface, ativar ou desativar o painel de ajuda e terminar a sessão do utilizador.

O menu A está reservado para os módulos e serviços instalados e ativos. O menu B é contextual em relação ao menu selecionado em A. A interface para o sistema aqui apresentado será acedida através do menu *SHOP*.

Em C temos um menu em estilo de árvore e que é contextual em relação à opção escolhida no menu B. As opções aqui disponíveis refletem normalmente os registos disponíveis na base de dados e, ao selecionar uma destas opções, é carregado na área D a aplicação e os dados respetivos à opção em C.

III.8.2. Esboços da interface

Os esboços que se seguem mostram uma solução para a estrutura dos menus e de como os dados podem ser disponibilizados na interface. Estes esboços focam na

apresentação dos dados mais críticos sendo os da configuração do serviço e das suas lojas e recursos, assim como de como apresentar os elementos das lojas.

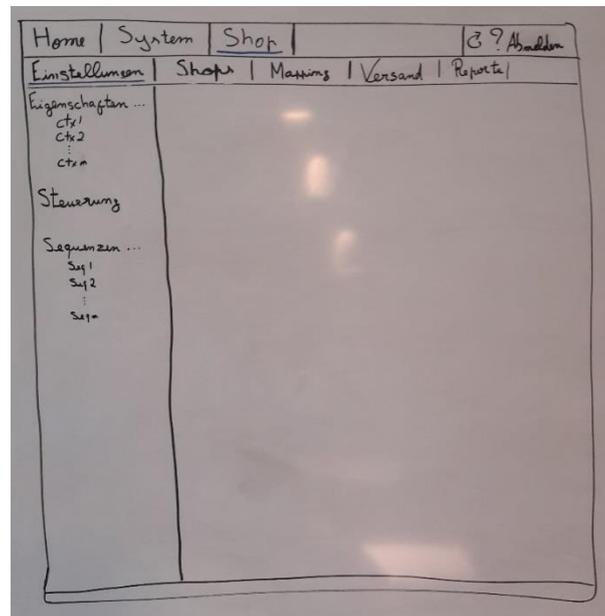


Figura III-17 - Esboço do menu "Einstellungen"

Esta figura refere-se à proposta para apresentar e alterar as configurações (*Einstellungen*) globais do serviço.

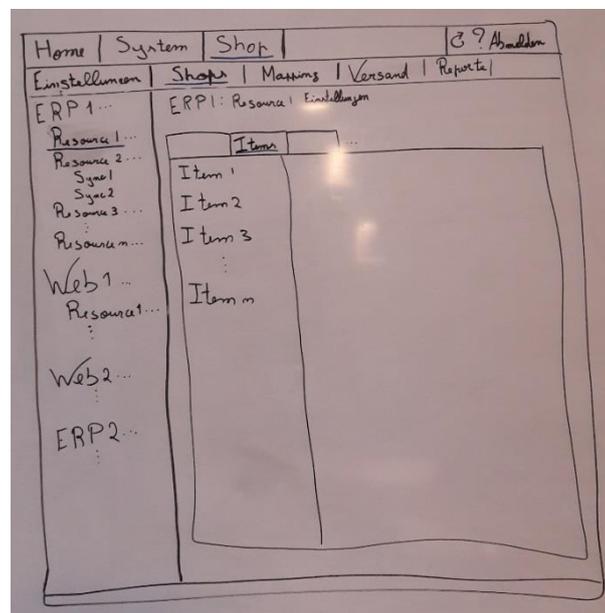


Figura III-18 - Esboço do menu "Shops"

Aqui nesta figura temos o proposto para apresentar ao administrador as configurações das lojas, recursos e elementos.

III.9. IMPLEMENTAÇÃO DO SISTEMA

Um dos objetivos deste estágio era o desenvolvimento e a implementação de um protótipo do sistema aqui apresentado. A implementação deste protótipo será apresentada no próximo ponto.

O protótipo tinha a finalidade de satisfazer as necessidades específicas de um cliente e também de servir como uma introdução ao ecossistema *KobaltC* e às lojas na Web.

Muitos aspetos deste projeto eram novos para mim, como a linguagem de programação *C++*, com a qual tive contacto apenas num projeto pequeno durante a formação académica, assim todo o ecossistema *KobaltC* e o tema lojas na Web, na perspetiva do programador. Houve um período de várias semanas de aprendizagem, em que a falta de documentação ou errada não facilitou, até dar início ao desenvolvimento do protótipo.

III.9.1. Protótipo

O protótipo foi desenvolvido e implementado utilizando como base as funcionalidades pretendidas pelo cliente.

O cliente utiliza o sistema ERP *BüroPlus* e quer sincronizar dados com duas lojas na Web do tipo *Shopware*. O cliente pretende que os dados relativos a artigos, incluindo categorias de artigos e as imagens dos mesmos, sejam sincronizados do sistema ERP para as lojas na Web. No sentido inverso, é pretendido que sejam sincronizados os dados relativos a encomendas e a clientes que efetuam encomendas.

III.9.1.1. Estrutura da base de dados

Na imagem seguinte temos a estrutura da base de dados que suporta o protótipo.

As tabelas cujo nome começa por **webjoblist**, controlam as tarefas que o serviço deve de executar. Estas tabelas fazem parte do esqueleto deste protótipo que foi preparado antes do meu envolvimento no projeto.

A estrutura da base de dados foi modelada consoante as necessidades específicas do cliente e a sincronização dos dados com a loja do tipo *Shopware*. As tabelas que no nome têm *shop_shopware* guardam algumas configurações para o tratamento dos dados e dos pedidos à loja. Nestas tabelas também são guardados os dados dos elementos de cada recurso, sendo os dados de cada recurso guardados nas suas tabelas respetivas.

III.9.1.2. Interface do utilizador

A interface do utilizador do protótipo permite ao administrador alterar os parâmetros para a sincronização. De seguida temos alguns exemplos da interface criada para este protótipo.

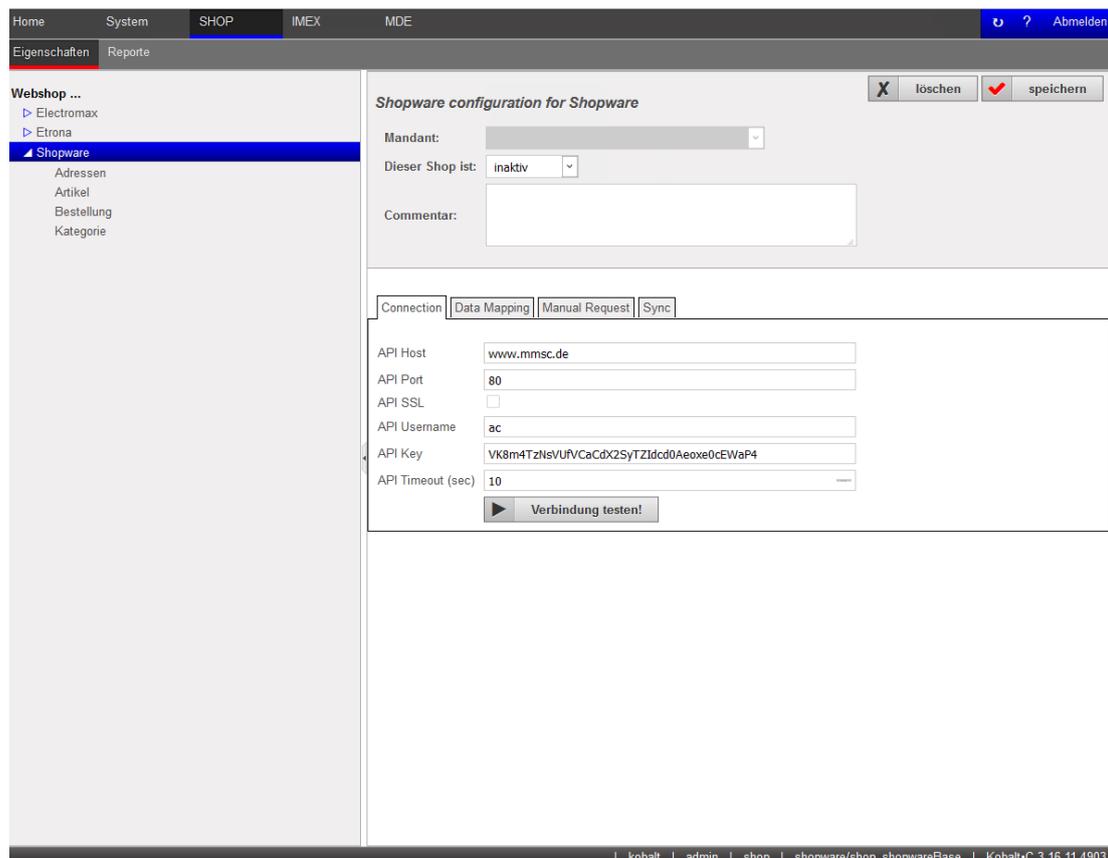


Figura III-20 - Página para a configuração dos parâmetros da API

Na página da figura anterior é possível alterar os parâmetros utilizados na API para comunicar com a loja, e também testá-los.

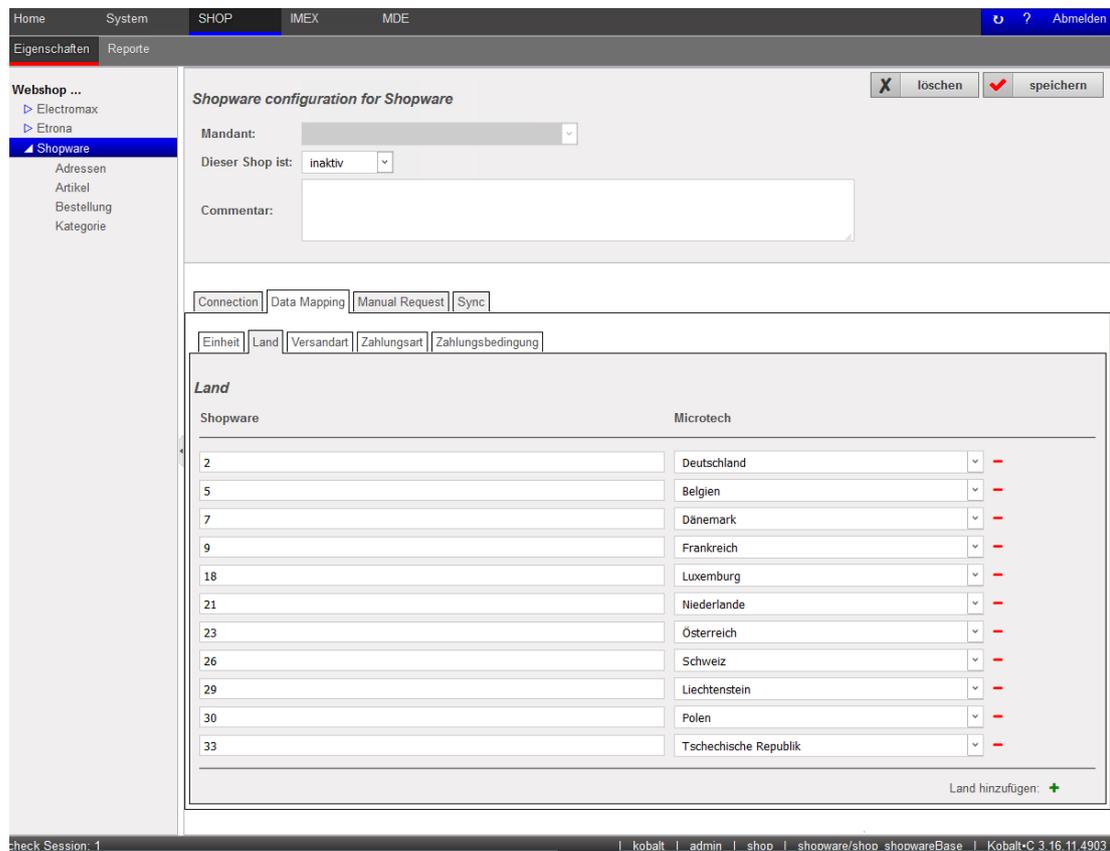


Figura III-21 - Página para o mapeamento de valores entre ERP e Shopware

Nesta área podemos configurar o mapeamento de diversos valores como as unidades, os países, os tipos de pagamento e os tipos de expedição.

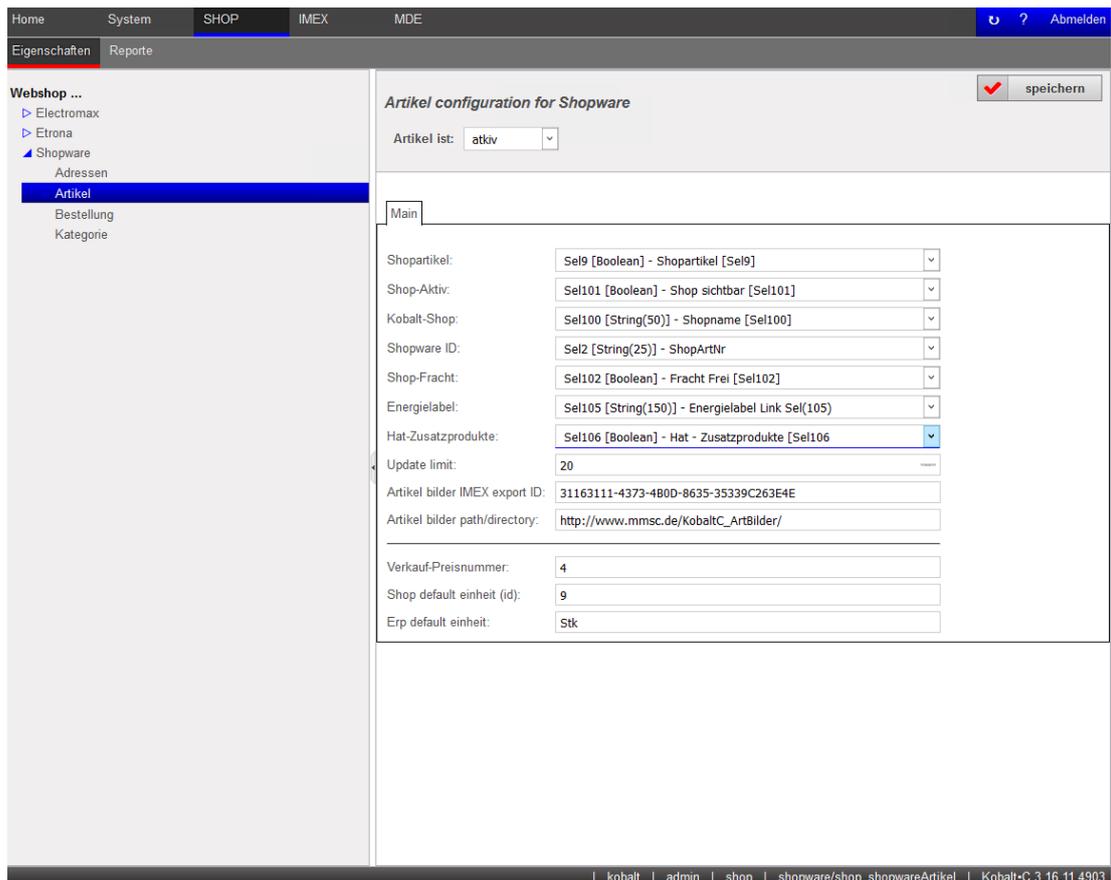


Figura III-22 - Página para configurar parâmetros para o recurso artigo

Nesta figura temos a área onde podemos configurar alguns parâmetros utilizados para a sincronização dos artigos da loja.

III.9.2. KbltShop

Na seguinte tabela temos os requisitos levantados para o sistema, a sua prioridade e o seu estado de implementação.

Requisito	Prio.	Implem.
RF1: O sistema deve suportar a troca de dados entre sistemas ERP e lojas na Web, numa relação de 1 para n .	Alta	Sim
RF2: O sistema deve suportar a troca de dados entre sistemas ERP distintos, numa relação de 1 para n .	Baixa	Sim
RF3: O sistema deve suportar a troca de dados entre lojas na Web distintas, numa relação de 1 para n .	Baixa	Sim

RF4: O sistema deve suportar a troca de dados de um recurso em ambas as direções quando disponível.	Alta	Sim
RF5: O sistema deve permitir ao administrador efetuar um <i>ping</i> à loja para determinar se está acessível.	Média	Sim
RF6: O sistema deve ser capaz de listar um ou mais elementos das lojas na Web e dos sistemas ERP.	Alta	Sim
RF7: O sistema deve ser capaz de criar novos elementos nas lojas na Web e nos sistemas ERP.	Alta	Sim
RF8: O sistema deve ser capaz de atualizar os dados dos elementos nas lojas na Web e nos sistemas ERP.	Alta	Sim
RF9: O sistema não deve ser capaz de remover elementos das lojas Web. Esta funcionalidade deve ser somente permitida com a supervisão de um administrador.	Alta	Sim
RF10: O sistema não deve permitir a remoção elementos dos sistemas ERP.	Alta	Sim
RF11: O sistema deve iniciar a sincronização dos dados dos elementos quando alterações nos dados destes elementos é detetada.	Alta	Sim
RF12: O sistema deve iniciar sincronização de dados através de tarefas escalonadas.	Alta	Sim
RF13: O sistema deve manter um histórico dos dados dos elementos.	Média	Sim
RF14: O sistema deve manter um histórico das transferências de dados com as lojas na Web.	Alta	Sim
RF15: O sistema deve ser capaz de detetar novos elementos nas lojas na Web e nos sistemas ERP para sincronizar.	Alta	Sim
RF16: O sistema deve ser capaz de identificar quais os campos dos elementos que sofreram alterações.	Média	Sim

RF17: O sistema deve ser capaz de atualizar apenas os campos dos elementos que sofreram alterações.	Média	Sim
RF18: O sistema deve permitir ao administrador selecionar os campos dos elementos que devem ser sincronizados.	Alta	Sim
RF19: O sistema deve sincronizar apenas os elementos que cumpram os critérios pré definidos. O administrador deve ser capaz de alterar os critérios para a sincronização.	Alta	Sim
RF20: O sistema deve alertar o administrador por email quando uma loja na Web ou sistema ERP estão incontactáveis após um determinado número de tentativas.	Média	Sim
RF21: O sistema deve detetar valores indesejados ou errados nos dados dos elementos.	Alta	Sim
RF22: O sistema deve atuar com ações e/ou alertas quando a situação do requisito anterior se verifica.	Alta	Sim
RF23: O sistema deve ter a capacidade de mapear valores entre os diferentes sistemas.	Alta	Sim
RF24: O administrador deve ser capaz de atuar manualmente no sistema.	Média	Sim
RF25: O sistema deve suportar importação e exportação em massa de dados.	Baixa	Não
RF26: O sistema deve ser capaz de interagir com as lojas na web através do protocolo REST.	Alta	Sim
RF27: O sistema deve ser capaz de interagir com as lojas na web através do protocolo SOAP.	Média	Não
RF28: O sistema deve suportar a troca de dados através do formato JSON.	Alta	Sim
RF29: O sistema deve suportar a troca de dados com os sistemas ERP, Büro+ e ERP Complete.	Alta	Sim

RF30: O sistema deve suportar a troca de dados com a loja Shopware.	Alta	Sim
RF31: O sistema deve suportar a troca de dados com a loja Magento.	Média	Não
RF32: O sistema deve de poder comunicar com sistemas de transporte de encomendas e requerer números de expediente.	Média	Não
RF33: O sistema deve permitir portar os dados contidos na base de dados entre diferentes instalações do KobaltC.	Média	Sim
RF34: O sistema deve ser capaz de utilizar dados portados de outra instalação do KobaltC.	Alta	Sim
RF35: O sistema deve permitir criar relatórios.	Média	Sim
RNF1: O sistema deve ser desenvolvido na linguagem C++.	Alta	Sim
RNF2: A base de dados a utilizar deverá ser o PostgreSQL.	Alta	Sim
RNF3: O sistema deve disponibilizar uma interface gráfica integrada no KobaltC WebClient para facilitar a administração do sistema.	Baixa	Não
RNF4: O sistema deve poder sincronizar os dados com diferentes sistemas simultaneamente.	Baixa	Não
RNF5: O sistema deve facilitar a criação do suporte à sincronização de qualquer recurso disponível nas lojas na Web e sistemas ERP.	Alta	Sim
RNF6: O sistema deve facilitar a criação de suporte a novas lojas na Web e sistemas ERP.	Alta	Sim
RNF7: O sistema deve poder aplicar alterações aos parâmetros para a sincronização dos dados de uma forma hierárquica. Os parâmetros podem ser aplicados a elementos individuais, a todos os elementos de um recurso específico ou a todos os elementos de uma loja na Web ou sistema ERP.	Média	Sim

RNF8: O sistema deve permitir ao administrador alterar os parâmetros mais relevantes de funcionamento do sistema.	Média	Sim
RNF9: O sistema deve ser capaz de retomar as tarefas em progresso caso haja uma falha da máquina onde o sistema corre.	Alta	Sim
RNF10: O sistema deve suportar uma execução distribuída, isto é, diferentes máquinas devem ser capazes de trabalhar nos mesmos dados.	Baixa	Parcial
RNF11: O sistema deve disponibilizar a sincronização de artigos, de categorias de artigos, de imagens de artigos, de clientes e de encomendas por omissão para todas as lojas na Web e sistemas ERP suportados pelo sistema.	Baixa	Sim

Tabela III-8 – Estado de implementação dos requisitos para o sistema

Anteriormente foi mencionado que alguns dos requisitos estão implementados noutros módulos e serviços do ecossistema *KobaltC*, e estes são:

- **RF12**, implementado pelo serviço *KbltSched*, o escalonador de tarefas do *KobaltC*. Para o nosso serviço, é necessário configurar tarefas de escalonamento que, através de funções SQL, iniciam uma tarefa no serviço SHOP.
- **RF20**, **RF22**, implementado no serviço *KbltPostMaster*, aqui podem ser configurados modelos de emails a enviar, que podem ser utilizados por outros serviços, inclusivamente o SHOP.
- **RF35**, implementado no módulo *RPT*, onde se podem criar e configurar relatórios. O módulo SHOP tem apenas de fornecer os dados.

Aproveito para deixar uns comentários sobre alguns dos requisitos. Os requisitos **RF9** e **RF10** são para proteger a empresa de situações desagradáveis com clientes. Em momento algum devem ser removidos elementos, quer de lojas Web ou quer de sistemas ERP. Esta responsabilidade recai sobre o cliente.

Os requisitos **RF21** e **RF22** são para proteger o cliente de dados corromptos ou errados, sejam causados por erro humano ou do sistema. Por exemplo, não é desejado que artigos tenham preços iguais a 0. Situações destas devem ser prevenidas.

Para o requisito **RF24**, o administrador pode atuar no sistema alterando dados diretamente na base de dados ou através de funções SQL para o efeito.

As lojas que se pretende suportar inicialmente, utilizam REST, por isso o requisito **RF27** será deixado para o futuro caso uma futura loja o necessite.

Os requisitos **RF33** e **RF34** referem-se à situação onde configuramos um sistema localmente para testes e depois estas configurações são importadas na máquina do cliente. Isto foi conseguido ao utilizar UUID⁴ nas chaves primárias das tabelas de configuração do sistema.

A interface gráfica para configuração e administração do sistema (**RNF3**), não foi implementado, pois a prioridade é baixa, já que o mesmo é possível através da interface gráfica da base de dados. Será implementado num futuro próximo, assim como o requisito **RF32**.

O requisito **RNF10** está parcialmente implementado, no sentido que o serviço está preparado para suportar distribuição de processamento. Na tabela **shop_joblist** temos o campo *knuuid* que identifica a máquina física que a tarefa em questão deve ser executada. O sistema não suporta esta funcionalidade neste momento, mas a estrutura está preparada.

Na imagem seguinte está representada a estrutura completa da base de dados para este sistema.

⁴ UUID – universal unique identifier (identificador único universal), é um valor padronizado constituído por 32 caracteres alfanuméricos e quatro hífen (8-4-4-4-12), cuja a probabilidade de colisão de entre dois valores dentro do mesmo contexto é quase ínfima que o torna ideal para ser usado como uma chave

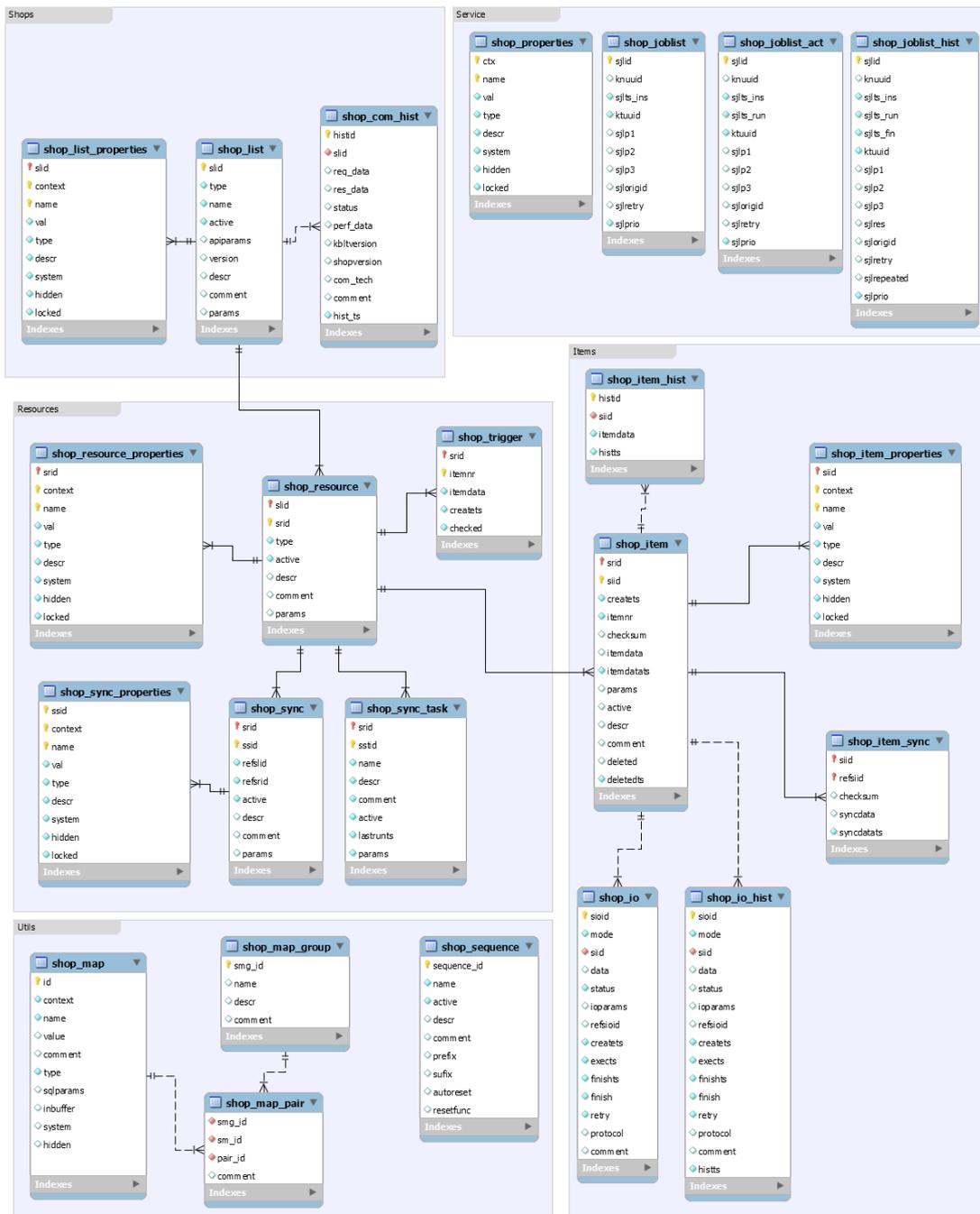


Figura III-23 - Estrutura completa da base de dados

IV. PROJETOS DIVERSOS

IV.1. DESENVOLVIMENTO, MANUTENÇÃO E TESTE DE SOFTWARE

Uma das condições para a realização deste estágio é que houvesse disponibilidade minha para participar noutros projetos. Inicialmente, a minha colaboração nesses projetos foi requisitada para assegurar a conclusão de diversos projetos nos prazos delineados para os mesmos e também para substituir colegas ausentes. O maior motivo foram mudanças na priorização dos projetos que retirou por diversas vezes o foco do projeto de estágio. Isto levou-me a uma rápida integração nos processos da empresa e a um conhecimento mais profundo e detalhado do software, pelo que mais e maiores responsabilidades me foram atribuídas na manutenção e adição de novas funcionalidades. Tive sobretudo responsabilidades de desenvolvimento, manutenção e realização de testes no global do *KobaltC*. Desde acrescentar novas funcionalidades, ou melhorar funcionalidades já existentes, identificar e eliminar falhas, e testar as novas versões de lançamento. Apresento de seguida alguns dos projetos mais relevantes em que participei.

IV.1.1. KobaltC Desktop Helper

O *KobaltC Desktop Helper*, como foi batizado oficialmente, é uma ferramenta criada com o intuito de facilitar a manutenção e o suporte do *KobaltC*. As ferramentas para tal já existiam, mas requerem conhecimento do utilizador sobre a linha de comandos do Windows, também como, requerem a repetição das mesmas linhas de comandos sempre que queiramos executar a tarefa em questão. Um bom exemplo para uma destas situações é aceder aos registos dos diversos serviços do ecossistema em tempo real, em que é necessário iniciar uma aplicação, através da linha de comandos do Windows, como por exemplo: `tail -f "C:\MMSC\KobaltC\log\KbltShop_2017-10-15.log"`. Isto é algo que se torna desnecessariamente repetitivo e passível a erros tipográficos, sobretudo em fases de depuramento, em que é necessário repetir os mesmos comandos frequentemente. Propus então desenvolver uma interface gráfica para facilitar estas tarefas e, para tal, optei por *C#* utilizando a *framework .NET* devido à experiência e preferência pessoal.

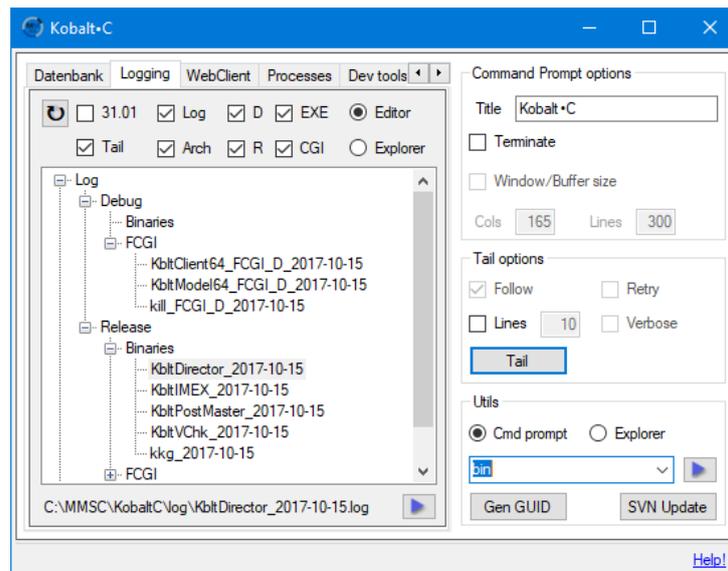


Figura IV-1 - KobaltC Desktop Helper

A tarefa de aceder aos registos dos serviços do *KobaltC* fica agora à distância de dois cliques no rato. Para além de facilitar o acesso aos registos, através desta aplicação é também possível efetuar alguma manutenção da base de dados, iniciar as aplicações do *KobaltC*, gerar as licenças para a utilização dos diversos módulos do *KobaltC*, entre diversas outras tarefas. Todas estas tarefas eram realizadas sempre através da linha de comandos do Windows.

Esta ferramenta passou a ser incluída no pacote de instalação do *KobaltC ERP Data Server* para facilitar o suporte realizado no sistema do cliente por parte da MMSC. Algumas das tarefas, como gerar licenças, estão indisponíveis nos sistemas dos clientes. Internamente, esta ferramenta ficou conhecida como: *The Lazy Helper*.

IV.1.2. LogX

No âmbito da monitorização do *KobaltC*, foi-me proposto a criação e manutenção de uma ferramenta para recolher e processar dados sobre determinados eventos e enviar por email para análise. Esta ferramenta foi batizada como *LogX* e foi rapidamente estendida para enviar avisos sobre diversas situações indesejadas ou que precisem de um olhar humano para determinar a ação a ter sobre a situação, assim como o envio de relatórios dinâmicos em texto simples ou em formato HTML com ou sem anexos.

Uma mensagem ou email *LogX* é composto por duas partes, nomeadamente, a recolha dos dados para o email e a composição e envio do mesmo. Os dados para o

email são obtidos da base de dados, através de uma declaração SQL completamente personalizável e à escolha do utilizador. Isto pode ser configurado na interface gráfica mostrada na imagem abaixo, juntamente com diversas outras opções para a recolha dos dados, tais como se o email deve ser enviado mesmo se nenhum dado foi retornado pela base de dados.

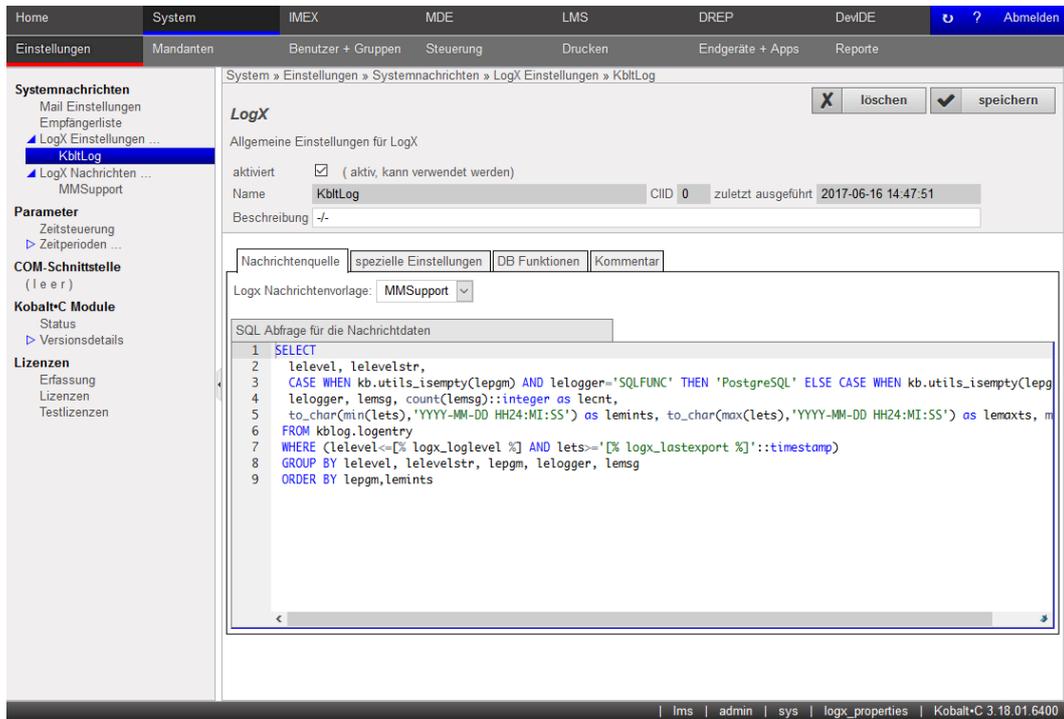


Figura IV-2 - Interface para a configuração da recolha dos dados para utilizar no email

Após a recolha dos dados, é iniciada a composição do email, cuja interface para configuração deste pode ser visualizada na próxima imagem. Aqui podemos personalizar o email com opções como: os destinatários, o assunto e corpo do email, este dividido em cabeçalho, corpo e rodapé, e ainda a opção de este último ser marcado como conteúdo HTML ou texto simples. Os dados anteriormente recolhidos podem ser utilizados em qualquer campo do email através da seguinte máscara [% {CAMPO} %], onde {CAMPO} é o nome atribuído à coluna na recolha dos dados e a máscara será substituída, de uma forma literal, pelo valor associado a esta coluna. Isto permite a total personalização do email.

Diversas outras opções estão também disponíveis, tais como a adição de anexos, limitação do tamanho do email, dividindo-o em diversos mais pequenos e de qual conta de email deve ser o email enviado.

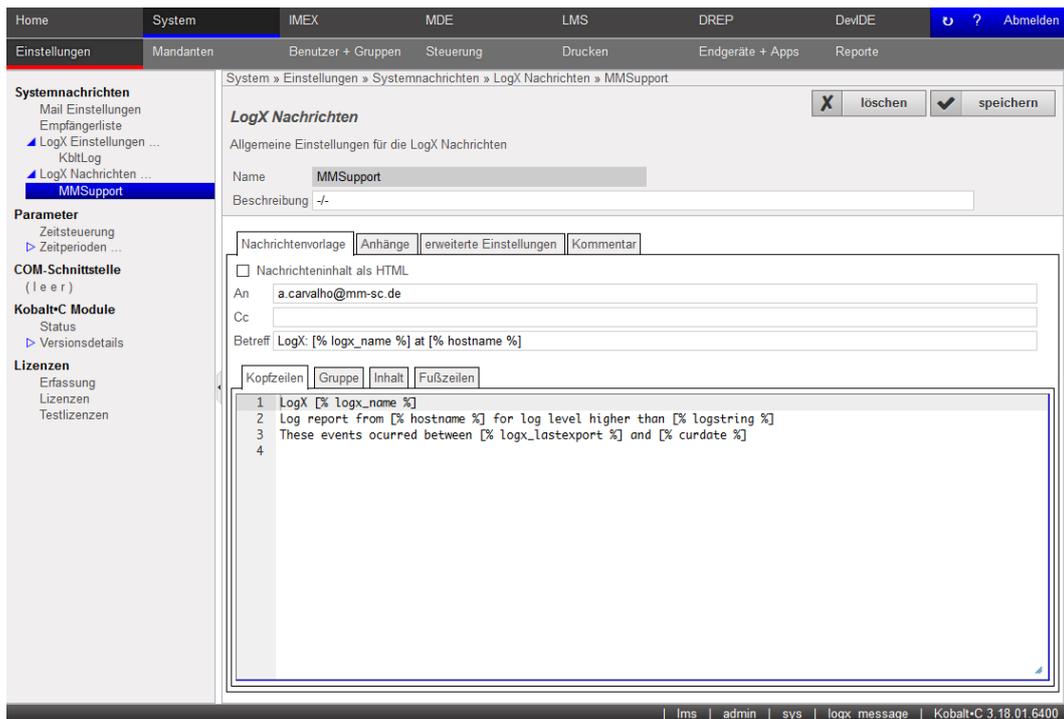


Figura IV-3 - Configuração da composição do email

IV.1.3. Módulo MDE

O módulo MDE foi sem dúvida o projeto em que estive mais envolvido fora o deste relatório. Neste projeto estive responsável pela concetualização, implementação, manutenção e suporte de diversos processos MDE, muitos destes sendo específicos para determinados clientes. Ainda relacionado com este módulo, fui também responsável pela criação de um serviço, semelhante a outros já existentes, específicos para lidar com documentos de entrada ou saída de bens, que permite, em conjunto com os processos MDE desenhados para tal, a criação e manipulação de documentos e outros objetos de negócios dentro do ERP através de dispositivos móveis. Devido a muitos dos processos serem desenhados em volta das necessidades ou dos desejos de clientes específicos, tive a oportunidade de interagir de uma forma indireta com diversos clientes e, de facto, pude observar que situações como a seguinte imagem descreve (não literalmente), são por vezes realidade. Isto acontece principalmente pelo facto de o cliente não saber exatamente o que necessita ou o que se adequa melhor à sua situação, e também devido a um não planeamento ou falta de documentação para a implementação do processo.

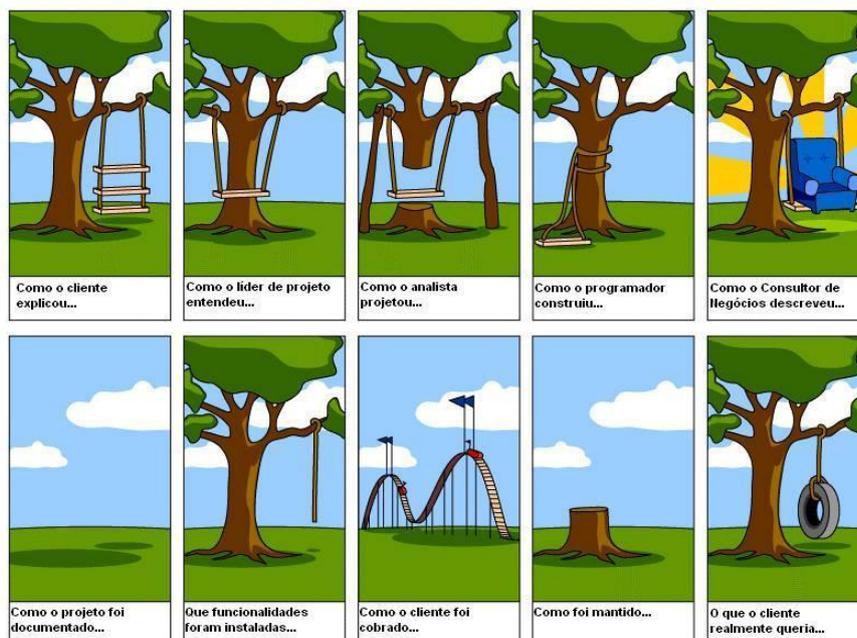


Figura IV-4- Ilustração do desenvolvimento de um projeto de software [26]

O meu maior envolvimento neste módulo levou a aumentar consideravelmente os meus conhecimentos em base de dados e SQL, sobretudo a nível de desempenho de consultas à base de dados. Os processos MDE são desenhados com uma interface muito simples, em HTML, pois os dispositivos alvo para estes processos têm uma grande ênfase na robustez de construção e duração de bateria para serem utilizados no armazém com o mínimo de interrupções possíveis. Estes processos são controlados sobretudo através de SQL e funções SQL permitindo assim adaptar e construir os processos às necessidades específicas de cada cliente.

Na próxima imagem, exemplifico um dos processos MDE que desenvolvi, o processo *Abgang*, cujo intuito é a retirada de mercadoria do armazém. É um processo relativamente simples, onde é lido o código de barras que identifica o artigo o qual se pretende retirar uma quantidade x do armazém em questão.

Abgang	Präsentation	ac
LagNr		
1		
ArtNr		
9904		
Bremsen Charge und MHD		
Menge	<input type="text" value="10"/>	Einh
		Stk
zurück		
ok		

Figura IV-5 - Processo MDE Abgang

IV.2. SUPORTE AO CLIENTE

A maioria dos projetos da empresa requer adaptações às necessidades dos clientes pelo que torna cada projeto diferente um do outro, apesar de partilharem a mesma base. Ao longo do estágio, com a minha envolvência nestes projetos, as responsabilidades de suporte ao respetivos clientes e projetos ficaram da minha responsabilidade, embora não interaja diretamente com os clientes, devido a ainda não dominar o idioma alemão. Tive também a oportunidade de realizar visitas a clientes, para discutir projetos correntes e futuros.

Um destes projetos foi a implementação de uma interface para comunicar com um sistema de armazém automático, o *KASTOlvr*, através da troca de ficheiros com os dados no formato *Cobol/PIC*.

O suporte aos clientes inclui também a manutenção do *KobaltC* e, em certos casos, da máquina onde este está instalado, fazer as configurações necessárias para que os serviços estejam de acordo com as necessidades do cliente, assim como resolver ou esclarecer alguma dúvida ou situação indesejada na execução do *KobaltC*.

A empresa deu-me também a oportunidade de frequentar um curso de alemão para que consiga interagir mais e com melhor qualidade com os clientes e com os restantes colegas da empresa no idioma alemão.

V. CONCLUSÃO

V.1. SOBRE O PROJETO DESENVOLVIDO

O primeiro objetivo do estágio era o desenvolvimento de um protótipo do módulo SHOP. Este era uma solução rápida para cumprir um contrato com um cliente, e, citando o engenheiro Melchert, serviu para testar a ideia do sistema mais completo a desenvolver. A pesquisa e ilações tiradas durante o desenvolvimento deste serviu como base para o desenvolvimento do módulo SHOP, pois antes eu não tinha experiência nesta área.

O protótipo foi dado como concluído e pronto a ser utilizado pelo cliente satisfazendo todas as suas necessidades. Por este motivo considero que o primeiro objetivo do estágio foi alcançado com sucesso.

A solução desenvolvida para cumprimento do segundo objetivo do estágio é uma solução robusta e escalável. Permite adicionar facilmente suporte a novas lojas na Web e sistemas ERP. Visto que este serviço processa dados sensíveis ao negócio dos clientes foi dada a maior importância em manter estes dados corretos em todas as fases da sincronização.

A médio e longo prazo o sistema pode vir a consumir imensos recursos, nomeadamente, espaço no disco devido aos requisitos de manter os históricos de comunicações e alterações dos dados. Esta é uma situação desejada porque salvaguarda tanto o cliente como a empresa caso haja algum problema que precise de ser apurado.

Creio que o segundo objetivo foi também cumprido visto que a estrutura da base de dados e o serviço implementado cumprem com os requisitos mais importantes para este sistema.

O terceiro objetivo do estágio não foi alcançado devido a outras prioridades e à importância do mesmo para a empresa. A interface gráfica, que neste caso seria utilizada para a configuração inicial do serviço, ou seja, utilizada apenas após a instalação do módulo, foi considerada como boa de se ter, mas não crítica pelo engenheiro Melchert. Esta interface será implementada gradualmente consoante a disponibilidade dos recursos humanos.

V.2. SOBRE O ESTÁGIO

A realização deste estágio foi uma experiência muito enriquecedora que me permitiu crescer bastante quer a nível profissional como a nível pessoal, ainda mais por ter sido realizado na Alemanha. Esta experiência marcou-me ao ponto de decidir permanecer na Alemanha após terminado o estágio.

A nível profissional fui capaz de adquirir novos conhecimentos e competências. Isto deveu-se sobretudo à colaboração com profissionais competentes nas suas áreas respetivas que me guiaram e partilharam comigo o seu conhecimento e experiência. Tenho de realçar a experiência obtida através das interações, de uma forma direta ou indireta, com clientes da empresa, algo que no percurso académico só é adquirível com a realização de um estágio.

Ao longo de todo o estágio, fui aprendendo o idioma alemão, que juntamente com os contatos criados, me serão muito úteis para um futuro profissional na Alemanha.

Em termos pessoais tive a oportunidade de conhecer e interagir com pessoas de diferentes culturas e credos, assim como de vivenciar uma nova cultura que me permitiu crescer bastante a um nível pessoal.

Pelo crescimento pessoal e profissional conseguido através desta experiência, aconselho vivamente a realização de um estágio, especialmente numa cultura diferente.

V.3. PERSPETIVAS FUTURAS

Em relação a este projeto, as funcionalidades a implementar no futuro próximo devem ser:

- obter números de expedição – a prioridade desta funcionalidade elevou-se já que pode ser muito apelativa a cliente novos e correntes.
- a interface de utilizador – apesar de ser possível utilizar a interface gráfica da base de dados e a linha de comandos para configurar e controlar o sistema, esta é uma forma que não apela a possíveis clientes. Uma interface gráfica tradicional ajuda a vender este módulo a clientes, pois, estes vêm algo com que se podem familiarizar e utilizar.
- adicionar suporte a mais lojas.

O sistema desenvolvido comunica com diversos sistemas externos. Estes sistemas estão sujeitos a sofrerem alterações e a serem atualizados, o que requer manutenção contínua para assegurar que o *KobaltShop* continue a suportar os sistemas afetados. Esta situação ocorreu uma vez durante o desenvolvimento do projeto, com a loja *Shopware*, onde após uma atualização, o nosso sistema era incapaz de comunicar com a loja devido a uma alteração na estrutura dos dados para a comunicações por parte da loja.

Como já fora mencionado acima, durante o decorrer do estágio, foi-me oferecido um lugar efetivo na empresa. A direção da empresa também me comunicou que gostaria de me atribuir uma maior responsabilidade no planeamento e tomada de decisão em todos os projetos de software. Aceitei este novo papel porque me sinto confortável na empresa, tenho colegas capazes com os quais sei que posso contar e sinto confiança que o consigo desempenhar. Dito isto, o meu futuro próximo, a nível profissional, será continuar nesta empresa.

VI. REFERÊNCIAS

- [1] A. Tambovcevs e T. Tambovceva, "ERP system implementation: benefits and economic effectiveness," em *Proceedings of the 2013 International Conference on Systems, Control, Signal Processing and Informatics*, Rodes, 2013.
- [2] S. Sadrzadehrafiei, A. G. Chofreh, N. K. Hosseini e R. Sulaiman, "The Benefits of Enterprise Resource Planning (ERP) System Implementation in Dry Food Packaging Industry," em *The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)*, Malásia, 2013.
- [3] M. Dunaway e S. Bristow, *Importance and Impact of ERP Systems on Industry and Organization*, Arkansas, 2015.
- [4] E. Fosser, O. H. Leister, C. E. Moe e M. Newman, *ERP Systems and Competitive Advantage: Some Initial Results*, Manchester.
- [5] Forbes, "The World's Biggest Public Companies, Software/Programming," [Online]. Available: <https://www.forbes.com/global2000/list/#industry:Software%20%26%20Programming>. [Acedido em Junho 2017].
- [6] "MM Software Consulting GmbH," [Online]. Available: <http://mm-software-consulting.de/warenwirtschaft/microtech>. [Acedido em Junho 2017].
- [7] ANACOM, O Comércio electrónico em Portugal - O quadro legal e o negócio, ICP - Autoridade Nacional de Comunicação, 2004.
- [8] UNCTAD, "Information Economy Report: 2015: Unlocking The Potential Of E-Commerce For Developing Countries," United Nations Publications, Geneva, 2015.
- [9] N. Ratnakar, "Virtual Online Store". Estados Unidos da América Patente US20080195507 A1, 1 Janeiro 2007.
- [10] Forbes, "The World's Biggest Public Companies, Internet & Catalog Retail," [Online]. Available: <https://www.forbes.com/global2000/list/#industry:Internet%20%26%20Catalog%20Retail>. [Acedido em Junho 2017].
- [11] "Warenwirtschaft und E-Commerce - Profilösungen nach Maß | cateno.de," cateno GmbH, [Online]. Available: <https://www.catenode/>. [Acedido em Juli 2017].
- [12] "sync4 Schnittstelle zwischen Warenwirtschaft und Onlineshop," Dupp GmbH, [Online]. Available: <http://www.sync4.de/>. [Acedido em 07 2017].
- [13] "Compusoft Hard- & Software GmbH," Compusoft Hard- & Software GmbH, [Online]. Available: <https://www.compusoft-fn.de/index.php>. [Acedido em 07 2017].
- [14] "A Brief Description - A C++ Information," [Online]. Available: <http://www.cplusplus.com/info/description/>. [Acedido em Junho 2017].

- [15] "Code::Blocks," [Online]. Available: <http://www.codeblocks.org/home>. [Acedido em Julho 2017].
- [16] "PostgreSQL: About," [Online]. Available: <https://www.postgresql.org/about/>. [Acedido em Junho 2017].
- [17] "pgAdmin: PostgreSQL administration and management tools," [Online]. Available: <https://www.pgadmin.org/index.php>. [Acedido em Junho 2017].
- [18] "W3C HTML," [Online]. Available: <https://www.w3.org/html/>. [Acedido em Junho 2017].
- [19] "Knockout: Introduction," [Online]. Available: <http://knockoutjs.com/documentation/introduction.html>. [Acedido em Junho 2017].
- [20] "Implementing the MVVM Pattern Using the Prism Library 5.0 for WPF," [Online]. Available: [https://msdn.microsoft.com/en-us/library/gg405484\(v=pandp.40\).aspx](https://msdn.microsoft.com/en-us/library/gg405484(v=pandp.40).aspx). [Acedido em Julho 2017].
- [21] "JSON," [Online]. Available: <http://www.json.org/>. [Acedido em Junho 2017].
- [22] "About TortoiseSVN," [Online]. Available: <https://tortoisesvn.net/about.html>. [Acedido em Junho 2017].
- [23] "Mantis Bug Tracker," [Online]. Available: <https://www.mantisbt.org/index.php>. [Acedido em Junho 2017].
- [24] K. E. Wiegers, *Standing on Principle*, 1997.
- [25] I. Sommerville, *Software engineering*, Addison-Wesley Longman, Amsterdam, 2011.
- [26] "Projecto de Software," [Online]. Available: <http://tecnociencia.etikweb.com/Article-48-Projecto-de-Software.html>. [Acedido em Julho 2016].
- [27] M. R. S. S. Buschmann, *Patternorientierte Softwarearchitektur*, Addison-Wesley, 1998.
- [28] Bloch, Joshua - Principal Software Engineer, Google, "How to Design a Good API and Why it Matters".