



Relatório de Estágio de Mestrado em  
Engenharia Informática - Computação Móvel

***Implementação e Manutenção Corretiva e  
Evolutiva de um Sistema CORE***

**Luís Miguel Mendes Moreira**

Leiria, Março de 2015





Relatório de Estágio de Mestrado em  
Engenharia Informática - Computação Móvel

# ***Implementação e Manutenção Corretiva e Evolutiva de um Sistema CORE***

**Luís Miguel Mendes Moreira**

Relatório de Estágio de Mestrado realizada sob a orientação do Doutor Luís Marcelino,  
Professor Adjunto da Escola Superior de Tecnologia e Gestão do Instituto Politécnico  
de Leiria

Leiria, Março de 2015



## Agradecimentos

Gostaria de expressar o meu agradecimento a todas aquelas pessoas que sempre acreditaram e sempre me apoiaram ao longo desta viagem e que, mesmo quando as coisas não corriam bem, nunca abriram mão de mim.

Quero agradecer deixar um enorme e especial agradecimento aos meus pais e à minha família, porque sem eles não seria quem sou hoje nem teria chegado onde cheguei, nem nas alturas difíceis deixaram de acreditar e de me dar todo o apoio necessário.

A todos os docentes que me auxiliaram e apoiaram não só na elaboração desta tese, como também ao longo destes dois anos, a crescer não só tecnicamente como também pessoalmente, deixando um especial obrigado aos professores Luís Marcelino e Catarina Silva, que me orientaram durante a elaboração desta tese e ao professor Carlos Grilo, que me facultou esta oportunidade.

Aos amigos e colegas que me acompanharam ao longo destes últimos anos quer nas atividades curriculares quer nas atividades extra curriculares.

Por último mas não menos importante, quero também agradecer a todos os profissionais da *Deloitte* que me ajudaram a aprender e a evoluir, tanto profissionalmente como pessoalmente, ao longo do projeto e também dentro da empresa.

A todos o meu muito obrigado.

*Esta página foi intencionalmente deixada em branco.*

## Resumo

A constante evolução das tecnologias de informação faz com que estas sejam cada vez mais indispensáveis no nosso dia a dia. O aparecimento de novas tecnologias e o melhoramento das já existentes, fazem com que tudo o resto tenha também de evoluir ou ser melhorado. Para tal, as empresas não especializadas na área, recorrem a outras entidades para que estas satisfaçam as suas necessidades tecnológicas, de forma a que os produtos que fornecem possam dar seguimento às necessidades dos mercados. A estes serviços é dado o nome de consultoria. A consultoria é uma das áreas dominantes nos mais variados sectores e na informática não é exceção. Cada vez são mais as empresas que procuram este tipo de serviços de forma a satisfazerem as suas necessidades tecnológicas.

A consultoria tem seguido as tendências da tecnologia, crescendo e evoluindo de forma notável nos últimos anos, aproveitando ao máximo os novos recursos que lhe vão sendo disponibilizados e adequando-se também às novas realidades de que os clientes necessitam. A consultoria não deve ser vendida, deve ser comprada pelo cliente de acordo com as suas necessidades. Como tal, diversas instituições, nomeadamente bancárias, devido às novas exigências do negócio, veem a sua arquitetura ficar limitada e desatualizada, tendo então a necessidade de contratar os serviços de empresas externas, como é o caso da *Deloitte*, de forma a que estas lhes apresentem uma ou várias soluções que colmatem as limitações ao negócio impostas pela sua atual arquitetura.

**Palavras-chave:** Consultoria, Manutenção, Finanças, Banca.

## Abstract

The constant evolution of the information technology makes it increasingly indispensable to our lives. The emergence of new technologies and the improvement of the existing ones brings the need of everything else evolve or get improved.

This evolution requires that companies that aren't specialized in the IT area ask for help to another companies that can satisfy their technologic needs, in order to update their products and provide the best service to their clients. This services are called consultancy. The consultancy area is one of the dominant areas in the most serveral sectors, being more and more the enterprises that search for this services in order to get satisfied their technologic needs.

The consulting services has followed technology trends, growing and evolving remar-

kably in recent years, taking full advantage of the new features that become available and also adapting it to the new realities that customers need. The consultancy services should not be sold, they must be purchased by the customer according to their needs. As such, several institutions, including banking, due to new business requirements, see their architecture get limited and outdated, then having the need to hire the services of external companies, such as the *Deloitte*, in order that they provide them with one or more solutions that obviate the restrictions on business imposed by its current architecture.

**Keywords:** Consulting, Maintenance, Finances, Banking.

# Lista de Figuras

3.1	Processo de Criação de uma Apólice de Seguro . . . . .	14
3.2	O que é segurável? . . . . .	16
3.3	Cálculo Detalhado do Valor do Prémio . . . . .	18
3.4	Resolução de um Incidente . . . . .	19
4.1	Áreas de negócio da Unicre . . . . .	24
4.2	Arquitetura Weblogic . . . . .	32
4.3	Principais Noções do Modelo SCA . . . . .	33
4.4	Representação de um <i>Database Link</i> . . . . .	44
4.5	Criação de um Produto com Recurso a SCRUM . . . . .	48
4.6	AOSE do Projeto MMK . . . . .	52
4.7	Ciclo de vida de um documento . . . . .	54
4.8	Ciclo de geração de um documento . . . . .	56
4.9	Middleware agregado ao BI Publisher . . . . .	60
4.10	Estrutura da Mensagem de Correio Eletrónico . . . . .	64
4.11	Ciclo de um pedido de adesão DE . . . . .	66
4.12	Envio e disponibilização de um documento para impressão local . . . . .	67
4.13	Fluxo de Geração de um Documento . . . . .	72
4.14	Ciclo de um pedido de adesão DE . . . . .	75
4.15	Processo de Certificação de Faturas . . . . .	80
A.1	Arquitetura do Projeto MMK . . . . .	88
A.2	Exemplo de uma estrutura de testes no SOAPUI . . . . .	89
A.3	Adição de campo BCC na Expedição por Email . . . . .	89
A.4	Diferenciação de Documentos por Área de Negócio . . . . .	90

*Esta página foi intencionalmente deixada em branco.*

# Lista de Tabelas

3.1	Calendarização Academia AMS . . . . .	9
3.1	Calendarização Academia AMS (continuação) . . . . .	10
3.1	Calendarização Academia AMS (continuação) . . . . .	11
A.1	Detalhe dos Critérios de Expedição Interna . . . . .	90
A.1	Detalhe dos Critérios de Expedição Interna (continuação) . . . . .	91
A.2	Detalhe dos Critérios de Expedição Externa . . . . .	91
A.2	Detalhe dos Critérios de Expedição Externa (continuação) . . . . .	92

*Esta página foi intencionalmente deixada em branco.*

# Acrónimos

<b>Acrónimo</b>	<b>Significado</b>
API	<i>Application Programming Interface</i>
AOSE	Arquitetura Orientada a Serviços e Eventos
BCC	<i>Blind Carbon Copy</i>
BIP	<i>Business Intelligence Publisher</i>
BPEL	<i>Business Process Execution Language</i>
CRM	<i>Customer Relationship Manager</i>
DDL	<i>Data Definition Language</i>
DML	<i>Data Manipulation Language</i>
DE	Direção de Emissão
GT	Gestor de Tangíveis
GT_RTD	Gestor de Tangíveis - Repositório Temporário de Dados
GT_UPCP	Gestor de Tangíveis - Unidade de Processamento e Controlo de Produção
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
J2EE	<i>Java 2 Enterprise Edition</i>

<b>Acrônimo</b>	<b>Significado</b>
JAR	<i>Java ARchive</i>
JCB	<i>Japan Credit Bureau</i>
JNDI	<i>Java Naming and Directory Interface</i>
ODS	<i>Operational Data Storage</i>
MDS	<i>Metadata Services</i>
MMK	Mais Máquina Kernel
PA	Pedido de Adesão
PDF	<i>Portable Document Format</i>
PL/SQL	<i>Procedural Language Extension to Structured Query</i>
RTF	<i>Rich-Text Format</i>
RU	RedeUnicre
SAC_GIT	Suporte ApliCacional de Geração de Informação para Terceiros
SCA	<i>Service-Component Architecture</i>
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SPE	Suporte Perfeito e Evolutivo
STMP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
TIFF	<i>Tagged Image File Format</i>
XML	<i>eXtensible Markup Language</i>

<b>Acrónimo</b>	<b>Significado</b>
XSD	<i>XML Schema Document</i>
WSDL	<i>Web Service Definition Language</i>

*Esta página foi intencionalmente deixada em branco.*

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	2
1.2	Motivações e Objetivos . . . . .	3
1.3	Estrutura do Documento . . . . .	3
<b>2</b>	<b>Empresa</b>	<b>5</b>
2.1	Caracterização da Área de Negócio . . . . .	6
2.1.1	Application Management Services . . . . .	6
2.1.2	Consultoria . . . . .	8
<b>3</b>	<b>Formação</b>	<b>9</b>
3.1	Academia . . . . .	9
3.1.1	Oracle . . . . .	11
3.1.2	Microsoft . . . . .	12
3.1.3	The Insurance Application . . . . .	14
3.1.4	COBOL . . . . .	22
<b>4</b>	<b>Projeto Mais Máquina Kernel</b>	<b>24</b>
4.1	Responsabilidades e Tarefas . . . . .	25
4.1.1	Responsabilidades . . . . .	27
4.1.2	Tarefas . . . . .	28
4.2	Tecnologias . . . . .	29
4.2.1	Aplicações . . . . .	29
4.2.2	Oracle Weblogic . . . . .	30
4.2.3	Service-Oriented Architecture . . . . .	32
4.2.4	Business Intelligence Publisher . . . . .	40
4.2.5	Base de Dados Oracle . . . . .	42

4.3	Processo de Desenvolvimento . . . . .	45
4.3.1	Cadeias . . . . .	45
4.3.2	Repositório . . . . .	46
4.4	Comunicação e Metodologia . . . . .	47
4.4.1	Testes . . . . .	48
4.5	Arquitetura . . . . .	50
4.5.1	Arquitetura Orientada a Serviços e Eventos . . . . .	51
4.6	Gestor de Tangíveis . . . . .	52
4.6.1	Ciclo de Vida de um Documento . . . . .	53
4.6.2	BI Publisher no Gestor de Tangíveis . . . . .	58
4.7	Manutenção de Tangíveis . . . . .	61
4.8	Manutenção dos Processos de Expedição . . . . .	62
4.8.1	Expedição de Tangíveis . . . . .	62
4.8.2	Melhorias de Desempenho de Consultas e Processamento de Da- dos . . . . .	68
4.8.3	Melhoria do Processo de Geração de Documentos . . . . .	69
4.8.4	Manutenção de Serviços SOA . . . . .	71
4.9	Pedidos de Adesão . . . . .	73
4.9.1	Siebel . . . . .	74
4.10	Suporte . . . . .	76
4.11	Fusão entre Áreas de Negócio . . . . .	76
4.12	SAC_GIT . . . . .	77
4.13	Certificação de Faturação . . . . .	78
4.14	Gestão de Risco . . . . .	80
<b>5</b>	<b>Conclusão</b> . . . . .	<b>82</b>
<b>A</b>	<b>Anexos</b> . . . . .	<b>87</b>
A.1	Arquitetura do Projeto MMK . . . . .	88
A.2	SOAP UI . . . . .	89
A.3	Expedição por Email . . . . .	89
A.4	Expedição para Impressão Local . . . . .	90
A.5	Detalhe dos Critérios de Expedição Interna . . . . .	90
A.6	Detalhe dos Critérios de Expedição Externa . . . . .	91

*Esta página foi intencionalmente deixada em branco.*

# Capítulo 1

## Introdução

Vivemos num mundo cada vez mais evoluído e esta evolução reflete-se em quase tudo o que nos rodeia, nomeadamente nas tecnologias de informação. Esta evolução gera revoluções na maneira das pessoas pensarem, alterando mesmo alguns dos padrões a que estávamos habituados e que ainda há meia dúzia de anos pensávamos que seriam impensáveis tornarem-se no que são hoje. As alterações dos padrões e dos pensamentos fazem com que os clientes comecem a querer sempre mais de tudo, nomeadamente dos seus prestadores de serviços e as entidades bancárias não são uma exceção. Estas exigências levam a que estas entidades necessitem de contratar alguém que melhore ou renove por completo as suas infraestruturas de forma a continuarem a agradar aos seus clientes e também responderem às exigências dos mercados.

É inevitável reconhecer que onde quer que vamos, há sempre tecnologia por trás. Graças a isto, existem também diversas áreas em ascensão, nomeadamente a consultoria. Cada vez mais são as empresas que tiram partido desta maturidade e da grande capacidade de resposta das consultoras aos desafios que lhes são lançados.

Num projeto de consultoria surgem novos desafios a qualquer momento e nem em todos os desafios é fácil conjugar o conhecimento técnico com o conhecimento funcional, traduzir o negócio em *software* e é aqui que a consultoria é fulcral, é através dela que as coisas acontecem, é através dela que empresas de outras áreas que não TI conseguem acompanhar e satisfazer as suas necessidades tecnológicas e consequentemente as necessidades dos seus clientes.

Vivemos num mundo em que cada vez mais empresas e também mais pessoas recorrem aos seus serviços para atingir os seus objetivos e graças à força que a consultoria ganha e ao facto de se relacionar com tantas áreas, é de longe uma das melhores

opções para nos enriquecermos e melhorarmos enquanto profissionais.

## 1.1 Enquadramento

Este estágio foi realizado no âmbito do Mestrado em Engenharia Informática - Computação Móvel. Durante estes doze meses integrei os quadros de uma das *big four*<sup>1</sup>, a *Deloitte*, mais especificamente na área de *Application Management Services*. Esta área está essencialmente responsável por manter e/ou atualizar e evoluir sistemas já existentes, fazendo também análise e implementação de novas soluções.

Com início a 2 de Setembro de 2013 e término a 1 de Setembro de 2014, este estágio iniciou-se com uma pequena academia que permitiu testar, reconsolidar ou adquirir alguns conhecimentos que vim a precisar. A seguinte etapa foi a integração num projeto de *outsourcing* como analista e onde era expectável efetuar tarefas de consultoria, como implementação, análise e planeamento, e de manutenção corretiva e evolutiva, onde seriam corrigidos e melhorados componentes em produção.

A academia, baseada nos vastos anos de experiência que a empresa tem em lidar com recém-licenciados e ainda com as necessidades que os projetos requerem, fornece não só, uma revisão de alguns conteúdos que possam ter sido esquecidos ou menos bem consolidados durante o percurso académico como uma introdução a novas tecnologias. O projeto, por sua vez, consiste na renovação da arquitetura de uma das principais entidades bancárias a nível nacional e está dividido em duas grandes componentes: *acquiring* e *issuing*, distribuídas entre dois grandes projetos: o projeto de consultoria e o projeto de manutenção.

Uma vez que serei integrado numa área de integração, trabalhando essencialmente com tecnologias Oracle, serão apresentadas e explicadas as tecnologias utilizadas diretamente de forma a facilitar a suportar o restante documento. Será então descrita toda a experiência vivida neste projeto, começando pela fase de integração no mesmo, uma vez que era um projeto já em andamento, que na fase da minha integração já se encontrava em fase de implementação, até a uma fase pós produção, a fase de manutenção e evolução. Serão abordadas as mais diversas tarefas realizadas, alguns conceitos próprios do negócio e até mesmo as dificuldades encontradas, nunca deixando de passar pela sua colocação em produção e conseqüente manutenção.

---

<sup>1</sup> Nomenclatura utilizada para fazer referência às quatro maiores empresas na área de consultoria e auditoria [big4accountingfirms, 2014]

## **1.2 Motivações e Objetivos**

Este estágio teve como principal objetivo tornar-me um profissional capaz de compreender, implementar e gerir soluções em áreas financeiras, nomeadamente nas áreas da banca e dos seguros. Foi-me proposto que adquirisse um variado leque de competências, não só técnicas como também funcionais. As competências técnicas concentraram-se em tecnologias reconhecidas (as quais serão abordadas no capítulo 4) a nível mundial e com uma enorme presença no mercado, sendo expectável que não só adquirisse conhecimento ao ponto de analisar, desenhar e implementar soluções como também ser reconhecido como um especialista das mesmas. Numa vertente funcional, foi-me proposto que, gradualmente, ganhasse competências na área de gestão, nomeadamente nas componentes de gestão de projeto, como a organização de trabalho e isto de forma a atingir uma maturidade que me leve um dia mais tarde, a liderar e coordenar equipas de trabalho e também de gestão de infraestruturas, que acabam por ser um dos componentes indispensáveis em qualquer projeto de IT e sem fugir à exceção, neste mesmo projeto.

No projeto documentado, o cliente demonstrou a necessidade de reformar toda a sua arquitetura, já desatualizada, pouco eficiente e limitada, de forma a cumprir com as novas exigências dos tempos que correm.

Quando se efetua um trabalho de consultoria, o principal foco é a satisfação do cliente. Desta forma torna-se verdadeiramente desafiante resolver os desafios lançados pelos mesmos, existindo sempre a necessidade de ir não só ao encontro do que estes desejam como ainda superar as suas expectativas, fazendo assim com que estes ganhem confiança na empresa e que possam eventualmente procura-los no futuro para projetar novas soluções.

## **1.3 Estrutura do Documento**

Este documento está construído de forma a apresentar o trabalho desenvolvido, sendo introduzidos primeiros os componentes e as tecnologias associadas aos mesmos e só depois o trabalho realizado bem como os desafios encontrados, sendo algumas das vezes complementado por conhecimento relativo a este negócio em específico. A sua estrutura é composta por cinco capítulos.

No presente capítulo, capítulo um, é introduzido e enquadrado o estágio, bem como as motivações e objetivos do mesmo, servindo também para apresentar o documento.

No capítulo dois é dada a conhecer a **empresa** onde realizei o estágio, um pouco da sua história bem como do seu funcionamento e hierarquia montada dentro da mesma.

No capítulo três, de seu nome **formação**, é apresentada a calendarização das atividades desenvolvidas ao longo do estágio e também é feita uma exposição dos conhecimentos recebidos durante a academia para novos elementos.

No quarto capítulo, **Unicre**, é abordado o projeto realizado para o cliente, onde é apresentada numa primeira parte o projeto e o cliente onde fui inserido, as responsabilidades e as tarefas que me foram propostas. São também apresentadas as tecnologias utilizadas na implementação do projeto, desde a sua teoria a aspetos mais práticos relativos aos desenvolvimentos. Esta informação será seguida do trabalho desenvolvido que será sempre acompanhado do conhecimento adquirido relativamente à parte do negócio.

No último capítulo é feita a síntese de todo o trabalho desenvolvido ao longo destes meses. São também apresentadas algumas sugestões para trabalho futuro e por último é apresentada uma conclusão.

No fim do documento encontra-se ainda anexada informação que complementa o relatório que pela sua importância não poderiam ser omissos.

# Capítulo 2

## Empresa

A história da *Deloitte* remonta ao século XIX, e o seu nome internacional – *Deloitte Touche Tohmatsu* – deriva dos três percursores William Welch Deloitte, George Touche e Nobuzo Tohmatsu - líderes na área dos serviços profissionais.

*Deloitte* é a marca sob a qual dezenas de milhares de profissionais, trabalhando em firmas independentes espalhadas por todo o mundo, colaboram na prestação de serviços de auditoria, consultoria fiscal, consultoria e *corporate finance* a clientes nos mais diversos sectores de atividade. Estas entidades são firmas membro da *Deloitte Touche Tohmatsu Limited - DTTL*, uma sociedade privada de responsabilidade limitada do Reino Unido.

Cada firma membro presta serviços numa determinada área geográfica e está sujeita às leis e regulamentos profissionais do país ou dos países em que opera. A DTTL não presta diretamente serviços a clientes e suas as firmas membro são entidades separadas e legalmente distintas, não podendo uma obrigar as outras. A DTTL e cada firma membro é responsável pelos atos e omissões próprios não sendo responsável pelos dos outros. Cada firma membro opera sob os nomes *Deloitte*, *Deloitte & Touche*, *Deloitte Touche Tohmatsu*, ou outros nomes relacionados, sendo cada uma delas estruturada de modo diferenciado consoante as respetivas leis nacionais, regulamentos profissionais, costumes e outros fatores, podendo exercer a sua atividade nas respetivas áreas geográficas através de subsidiárias, participadas ou outras entidades.

Em Portugal e Angola, a *Deloitte* tem escritórios em Lisboa, Porto e Luanda e conta com cerca de 1.800 profissionais que assumem diariamente um compromisso com a excelência e presta serviços de auditoria, consultoria fiscal, consultoria e *corporate finance* através das seguintes entidades:

- *Deloitte & Associados*, SROC S.A. (auditoria, consultoria fiscal e gestão de risco);
- *Deloitte Consultores*, S.A. (consultoria e *corporate finance*);
- SGG – Serviços Gerais de Gestão, S.A. (*outsourcing* de serviços financeiros);
- *Deloitte & Touche – Auditores Limitada (Angola)* (auditoria, consultoria e consultoria fiscal).[Limited, 2014]

## 2.1 Caracterização da Área de Negócio

O *outsourcing* tornou-se uma ferramenta estratégica, uma vez que permite às empresas concentrarem-se no *core* do seu negócio enquanto uma entidade externa, especialista na área, se preocupa em tratar de outras tarefas que embora sejam essenciais ou negócio não são da sua competência.

Desta forma, a linha de serviços de *Business Process Solutions - BPS*, ajuda os seus clientes na melhoria de processos *non-core*, mas essenciais ao negócio e na gestão de aplicações, permitindo-lhes maximizar as suas vantagens competitivas, através de soluções globais de *outsourcing* da mais elevada qualidade.

Esta área de negócio faz parte da empresa SGG - Serviços Gerais de Gestão, que sendo uma filiada da marca *Deloitte*, presta serviços de *outsourcing* nas áreas de contabilidade administrativa e consultoria fiscal. A SGG é constituída por quatro divisões: *Tax Compliance*, *Human Resources*, *Finance & Accounting* e *Application Management Services*, estando o estágio inserida nesta última.

### 2.1.1 Application Management Services

A área de *Application Management Services - AMS*, apesar de ser uma das áreas mais recentes da *Deloitte* e tendo sido criada em Portugal, já possui um enorme prestígio no mundo da consultoria, chegando mesmo a prestar serviços e trabalhar sobre tecnologias que mais ninguém tem recursos para trabalhar.

Esta área, é definida por um conjunto de serviços que abrangem serviços de gestão, manutenção e evolução de *software*, estando dividida em duas grandes áreas FSI - *Financial Services Industry*, estando esta mais virada para área da Banca e Seguros e a área de *Manufacturing*, com clientes na área da Indústria, Banca e Retalho. Hierarquicamente, os integrantes da área podem ser classificados em cinco níveis principais:

- O *Partner* (Parceiro), que é o responsável pela área toda. É ele que gere e negocia projetos e contrata novos recursos. Noutras áreas da empresa, podem existir

vários *partners*, cada um deles responsáveis por uma subárea.

- Os *Managers* (Gestores) são responsáveis por fazer a gestão do projeto. São eles que negociam, dentro do projeto, diretamente com o cliente e que se encarregam de fazer a distribuição de tarefas e responsabilidades para as equipas.
- Os *Team Leaders* (Líderes de Equipa) são os responsáveis por gerir as equipas, ou seja, orientar, passar conhecimento acerca das tecnologias e do projeto de forma a que os analistas e os programadores possam executar as suas tarefas.
- Os *Programmers* (Programadores) que em termos de hierarquia se encontram logo abaixo dos líderes de equipa, são responsáveis por executar tarefas que exijam mais conhecimento, tanto técnico como de negócio. Estes começam também a acumular algumas tarefas de gestão pessoal de forma a que quando chegarem ao nível superior tenham a capacidade de gerir equipas de dimensão significativa.
- Os *Trainees* (Analistas), que se situam no fundo da pirâmide, são responsáveis pelas tarefas mais simples e com menor impacto no projeto. Geralmente são sempre acompanhados por alguém com experiência e conhecimento do negócio, como por exemplo, um programador.

Existem ainda outros níveis intermédios que dizem respeito aos *programmers*, que se encontra numa fase de transição da programação para a gestão, que é denominada de *experienced programmer*. No caso dos *managers* existe também uma distinção do mesmo género, havendo *pro-managers*, *managers* e *senior managers* conforme as responsabilidades e a experiência que cada um apresenta. Estas funções são atribuídas conforme o grau de experiência da pessoa não só dentro da empresa como também a nível tecnológico e a nível de gestão.

Em termos práticos, um projeto é fechado e negociado por um *partner*, podendo ter ou não interferência de um ou mais *managers* neste processo. Adjudicando-se o projeto, é o *manager* que fica responsável pela sua gestão, negociando e comunicando com os elementos das áreas funcionais, sendo ele que estará responsável por defender a restante equipa bem como perceber as necessidades do cliente. Logo abaixo dos *managers* encontram-se os *team leaders* que estão responsáveis por gerir as suas equipas, delegando e priorizando o trabalho conforme a sua importância, sendo eles que geralmente comunicam com as pessoas das áreas técnicas do cliente. Pontualmente e graças à sua experiência podem também desenvolver em conjunto com os programadores. Na base da hierarquia, encontram-se os programadores, cuja principal finalidade é implementar a solução pretendida pelo cliente.

Naturalmente que não é seguida uma estrutura rígida no que toca ao desempenho de

funções, podendo desta forma todas as pessoas dos variados cargos realizarem tarefas que vão além das suas competências, permitindo-lhes assim evoluírem e progredirem com maior facilidade na sua carreira.

### **2.1.2 Consultoria**

Uma vez que este conceito está sempre presente ao longo deste estágio, é importante contextualizar o que é a consultoria. A consultoria consiste no ato de fornecer conhecimentos especializados a terceiros, mais concretamente a entidades sem conhecimentos em determinada área. Um consultor pode tanto fornecer aconselhamento como apresentar um conjunto de soluções com vista a colmatar um determinado problema.

A consultoria poderá ser feita de duas formas:

- Internamente, quando um profissional é consultado por colaboradores ou departamentos internos da sua própria empresa;
- Externamente, quando um profissional presta serviços a clientes que não pertençam à sua empresa. Estes conhecimentos são fornecidos por um período de tempo a troco do pagamento de tarifas.

# Capítulo 3

## Formação

### 3.1 Academia

De forma a garantir uma boa preparação aos seus novos elementos, a *Deloitte* possibilita aos mesmos a integração numa das suas academias, neste caso, a academia AMS. Esta academia aborda algumas das tecnologias com maior importância nos projetos adjudicados bem como ainda aborda outras tecnologias que poderão ser úteis em potenciais clientes.

De forma a equilibrar e nos fornecerem um maior *know-how* possível, a academia foi dividida e calendarizada por nível de complexidade e importância, tal como é possível ver na tabela abaixo.

Tabela 3.1: Calendarização Academia AMS

Itens de Ação	Data Início	Data Fim
Academia Oracle		
PL/SQL	02/09/2013	04/09/2013
Oracle Forms	05/09/2013	09/09/2013
Oracle Reports	09/09/2013	11/09/2013
Exame Oracle	11/09/2013	11/09/2013

Tabela 3.1: Calendarização Academia AMS (continuação)

Itens de Ação	Data Início	Data Fim
Academia Microsoft		
SSIS - SQL Server Integration Services	12/09/2013	16/09/2013
SSRS - SQL Server Reporting Services	16/09/2013	17/09/2013
Exame Microsoft	17/09/2013	17/09/2013
Academia TIA		
Conceitos sobre Seguros	18/09/2013	18/09/2013
Conceitos TIA, Experiência de Utilizador e Modelo de Dados	18/09/2013	18/09/2013
Perfis de Utilizador e Autorizações	18/09/2013	18/09/2013
Editor de Produtos e Algoritmos de Tarifação	19/09/2013	19/09/2013
Conta e Comissões	20/09/2013	20/09/2013
Sinistros	20/09/2013	23/09/2013
Oracle Forms	23/09/2013	23/09/2013
Caso de Estudo	24/09/2013	01/10/2013
Academia Excel		
Conceitos Avançados sobre Excel	02/10/2013	02/10/2013
Casos práticos baseados em situações reais	02/10/2013	04/10/2013

Tabela 3.1: Calendarização Academia AMS (continuação)

Itens de Ação	Data Início	Data Fim
Academia Microsoft		
MVC 4.0 e Acesso a Dados	07/10/2013	07/10/2013
<i>Lightswitch</i> e <i>WebServices</i>	08/10/2013	08/10/2013
Caso Prático	09/10/2013	09/10/2013
<i>DotNetNuke</i>	10/10/2013	11/10/2013
Academia COBOL		
Estrutura e Sintaxe	14/10/2013	19/10/2013
Caso Prático	19/10/2013	19/10/2013

### 3.1.1 Oracle

A academia Oracle, que no início deveria ser apenas uma revisão de modelização de base de dados, SQL e PL/SQL [on the Net, 2014], rapidamente revelou que o que aprendi na escola é apenas uma gota no meio de um oceano. Conceitos como *packages* ou procedimentos que já haveriam sido esquecidos e que na escola pareciam um bicho-de-sete-cabeças, rapidamente se tornaram triviais quando comparados com o que ia aparecendo a cada dia da academia.

De forma a tornar a formação o mais proveitosa possível, os exercícios fornecidos durante a mesma, foram feitos sempre orientados à realidade, utilizando para tal base de dados de testes que provieram de clientes reais, para que fosse transmitida ao máximo uma ideia daquilo que estaria para vir.

Após ter passado por revisões relativas a funções proprietárias da Oracle, tipos ou mesmo otimizações de código, a primeira parte que foi realmente novidade, foram os Oracle *Forms*, que tal como o próprio nome indica, é uma tecnologia da Oracle que

permite desenhar e desenvolver aplicações empresariais de forma rápida e eficiente. Esta tecnologia permite criar, um pouco ao estilo do que fui habituado durante o meu percurso académico, formulários para trabalhar sobre *schemas* já existentes. Permite criar, por exemplo, as janelas para que seja possível adicionar, editar ou remover novos dados [Oracle, 2011].

Para complementar os *Forms*, foram ainda apresentados uns conceitos de relatórios (*Oracle Reports*), que à semelhança dos *Forms*, permite gerar de forma rápida e simples relatórios. Estes relatórios permitem recolher informação já existente nos *schemas* e apresenta-la de forma clara e estruturada para um determinado utilizador final [Oracle, 2014b].

### **3.1.2 Microsoft**

Tal como mencionado anteriormente, as tecnologias Microsoft tem bastante relevância na Deloitte, não fossem estas bastante utilizadas em diversos projetos da empresa. Esta academia abrangeu várias tecnologias da Microsoft nomeadamente os serviços de integração e relatórios.

#### **SQL Server: Serviços de Integração e Relatórios**

Esta formação permitiu adquirir conhecimentos base em ferramentas ETL (Extract, Transform and Load). O ETL é um processo essencial na gestão de base de dados, visto que permite manipular enormes volumes de dados, através de processamento paralelo, nomeadamente move-lo de uma origem para um destino num curto espaço de tempo [Janssen, 2014]. O primeiro contacto com as ferramentas ETL, foi feito através da plataforma SQL Server *Integration Services* da Microsoft, que permite resolver complexos problemas de negócio, relacionados com manipulação de dados [Microsoft, 2014b], tais como, a integração de dados, que pode ser um problema devido a heterogeneidade das fontes e formatos de dados. Foi ainda passado conhecimento dos SQL Server *Reporting Services* da Microsoft, que é uma ferramenta, em muito semelhante aos Relatórios da Oracle, cuja finalidade é fornecer ferramentas que permitem a criação, manutenção e entrega de relatórios [Microsoft, 2014c].

## **Microsoft Excel**

O Excel é uma ferramenta muito poderosa. No mundo empresarial, este assume ainda um poder e uma importância maior, sendo que algumas destas mesmas empresas apenas utilizam esta ferramenta para fazer a gestão dos seus recursos [Technology, 2014]. Esta formação serviu, não só, para aprender mais sobre a tecnologia mas também para aprender um pouco de negócio, visto que, à semelhança das formações anteriores, todos os exercícios criados foram baseados em dados reais provenientes de uma instituição bancária.

Os pontos chave desta formação foi aprender a manipular a informação, recorrendo para tal às funções nativas do excel e bem como o desenvolvimento de macros que permitem agilizar muitos outros processos.

## **Microsoft C# e Lightswitch**

O C# que é uma linguagem orientada a objetos desenvolvida pela Microsoft [Microsoft, 2014e] e que também já me acompanhou durante o meu percurso académico. Embora tenha servido apenas para introduzir a formação, o estudo da arquitetura Modelo-Vista-Controlador (MVC) [Microsoft, 2014a] foi o ponto de maior interesse desta formação, uma vez que anteriormente apenas me tinham sido transmitidas umas luzes do seu funcionamento ao invés de uma utilização num caso real.

A segunda parte da formação incidiu sobre uma ferramenta de desenvolvimento da Microsoft, o Lightswitch. Esta ferramenta tem como objetivo simplificar e encurtar o processo de desenvolvimento de aplicações de negócio. [Microsoft, 2014d]

## **DotNetNuke**

O DotNetNuke é uma plataforma para desenvolvimento web que foi construída sobre a plataforma ASP.NET, da Microsoft. Esta ferramenta está muito direcionada para o desenvolvimento de sites comerciais, intranets e extranets, mas pode também ser utilizado no desenvolvimento de muitas outras aplicações personalizadas [Rouse, 2013].

Esta formação foi inserida no contexto da formação da Microsoft, uma vez que é construída e utilizada num dos ambientes de desenvolvimento dos mesmos.

### 3.1.3 The Insurance Application

Muito mais que uma formação meramente técnica, esta formação permitiu também ter umas primeiras impressões de como funciona o mundo das seguradoras bem como alguns aspetos relativos aos seus modelos de negócio.

#### Seguros

A atividade de segurar tem como objetivo reduzir os riscos e as incertezas, de instituições ou indivíduos, através do estabelecimento de contratos de seguro. Por sua vez estes contratos são compostos três tipos de cláusulas: condições gerais que são imperativas e que geralmente são invariáveis e transversais a todos os contratos do mesmo tipo, condições particulares que são todas as condições que individualizam um contrato, tais como, o segurado, os beneficiários, os riscos cobertos, entre outros e por último, as condições especiais que são acordos contratuais que modificam a cobertura padrão.

O estabelecimento de uma apólice de seguro é bastante simples (ver figura 3.1), necessitando apenas especial atenção na avaliação dos riscos e conseqüente valor a ser pago pela apólice, uma vez que é este valor que permite às seguradoras continuarem com o seu negócio.

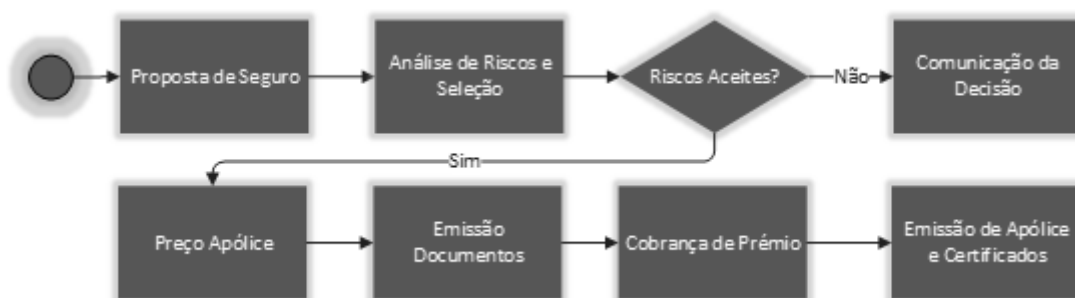


Figura 3.1: Processo de Criação de uma Apólice de Seguro

#### Conceitos

Nos tempos que decorrem, o conceito de seguro é algo que faz parte do nosso dia-a-dia. De forma a proteger-nos ou a proteger algo nosso, o seguro é um contrato no qual um segurador se compromete a indemnizar através de um prémio um segurado, no caso de ocorrer um sinistro.

Apesar de já existirem seguros bizarros nos dias de hoje, como por exemplo seguros contra *zombies*, nem tudo é segurável, uma vez que as seguradoras também têm de evitar reduzir o risco para si próprias, existindo mínimos que têm de ser cumpridos, como é o caso de o incidente ainda estar por acontecer (ver figura 3.2).

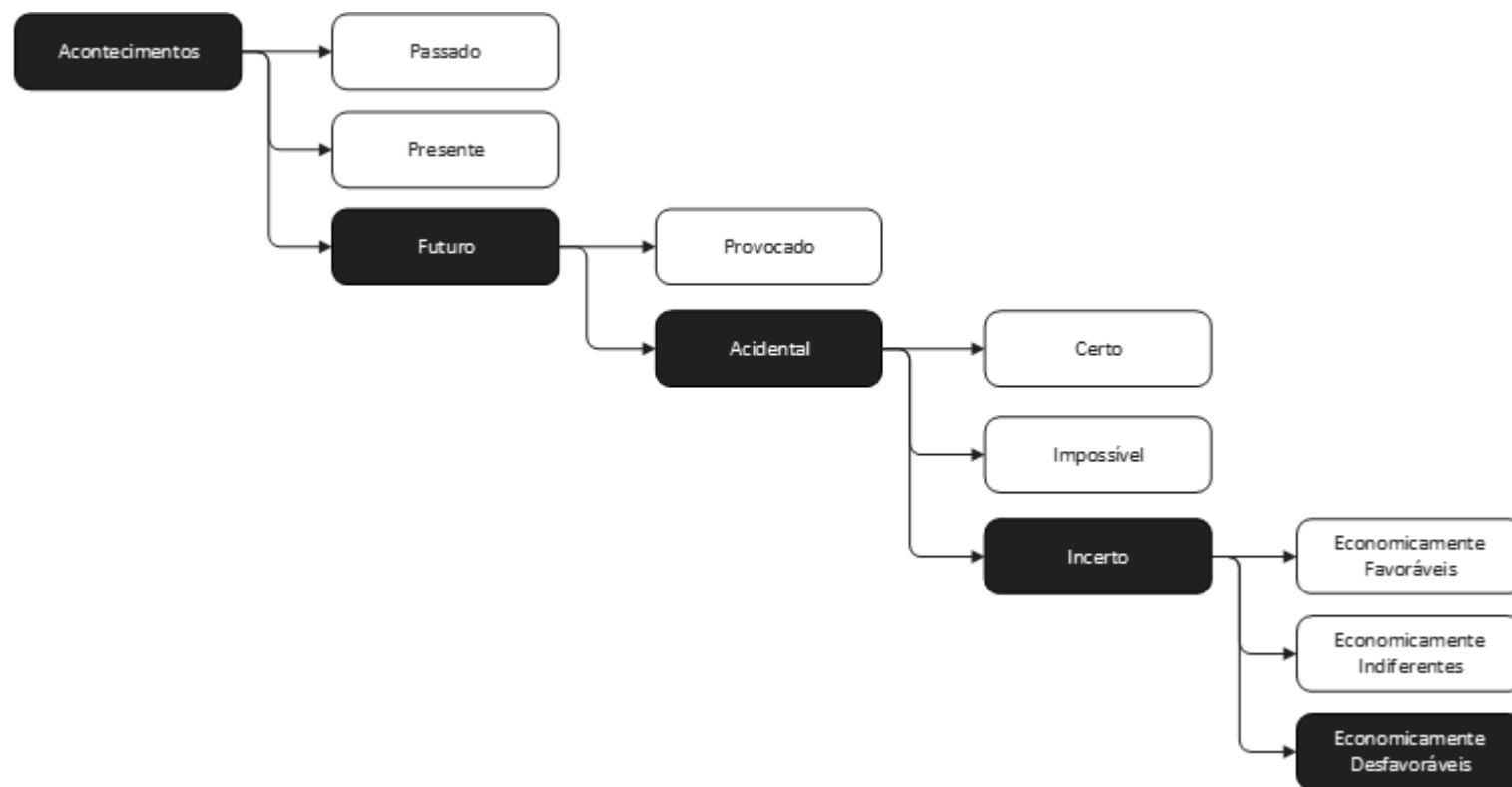


Figura 3.2: O que é segurável?

Começando pelo principal, o **segurador**, é a entidade legal que está autorizada a praticar a atividade seguradora, ou seja, é a entidade que subscreve um contrato com um tomador e fornece a proteção dos riscos. Por sua vez, o **tomador** é a entidade responsável pelo cumprimento do contrato contra o segurador, isto é, procura garantir que o segurador paga o prémio ao segurado em caso ocorrer um risco coberto pelo contrato. Do lado dos beneficiários existem outros tantos conceitos, sendo que no topo desta hierarquia se encontra o **segurado** que é a entidade, seja individual ou coletiva, cujos interesses irão estar na base da elaboração do contrato. Consequentemente, este contrato poderá ter vários **beneficiários**, não se limitando apenas ao segurado. Os beneficiários serão todas as entidades, individuais ou coletivas, que ficam também abrangidas pelo contrato de seguro. No caso de ser um seguro automóvel, existe ainda o **condutor habitual** que é a pessoa que geralmente conduz o carro segurado e que será considerada para a ativação do seguro.

Tal como noutras áreas financeiras, os seguros envolvem capitais, capitais estes que exigem garantias de ambas as partes envolvidas. Do lado do beneficiário, e de forma a proteger também a entidade seguradora, existe uma entidade, o **credor** que se compromete a prestar crédito ao segurado em caso de o mesmo não conseguir. Usualmente, o credor acaba por ser também um dos beneficiários do seguro.

Saindo das entidades e passando ao miolo do seguro, temos as **coberturas**, que tal como o próprio nome sugere, corresponde aos riscos individuais que são cobertos pelo seguro. Apesar dos chamados seguros contra todos os riscos, não existem realmente seguros que cubram todos os riscos, apenas existem seguros que cobrem maioritariamente os riscos de uma determinada área. Quantas mais forem as coberturas pretendidas na aquisição de um seguro, maior será o seu **prémio**. Além de ser calculado sobre as coberturas pretendidas pelo segurado, acrescentem-lhe ainda várias parcelas, um prémio base, um prémio comercial e sobre o prémio bruto ainda serão aplicadas as restantes cargas fiscais, tal como é possível ver na figura 3.3.

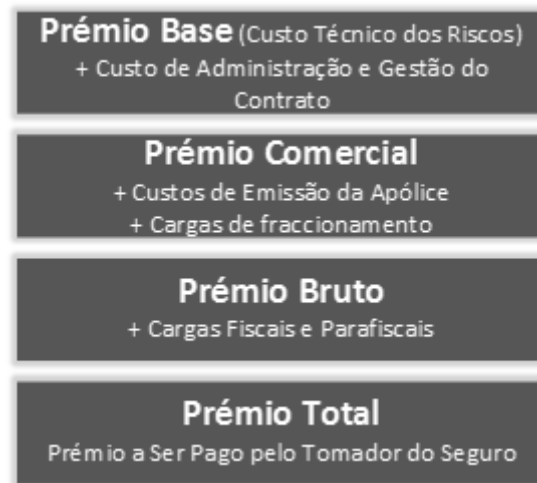


Figura 3.3: Cálculo Detalhado do Valor do Prémio

O prémio base é calculado, de uma forma muito simplista, pelo quociente entre incidências e o número de apólices. De forma a que seja possível à seguradora fazer um orçamento, é feita uma estimativa, designada por *preço a priori* onde o prémio pago pelo segurado irá, teoricamente, cobrir os encargos expectáveis. Como geralmente deste orçamento resulta uma margem larga, são feitos ajustes nos prémios, baseados nos incidentes que o segurado tenha, que é designado por *preço a posteriori*.

Como as seguradoras não têm capitais ilimitados, é necessário serem definidos limites e/ou proteções que protejam e garantam o futuro das mesmas. O excesso, é a parte do dano que o segurado terá de pagar no caso de ser responsável pelo acidente, esta medida tem como principal objetivo, partilhar o risco entre o segurado e a responsabilidade do mesmo. O excesso é inversamente proporcional ao prémio a ser pago, isto é, quanto maior for o excesso, menor será o prémio a ser pago.

Por outro lado, existem limites de garantias associadas aos contratos de seguros, sejam a nível financeiro ou mesmo a nível material.

A nível de contrato, existe uma duração associada ao mesmo, podendo esta ser anual, onde o contrato e o valor do prémio é automaticamente renovado e recalculado ao fim de um ano ou temporário, onde o contrato tem, usualmente uma curta duração e onde o prémio é calculado através da mesma. Se a incidência ocorrer a um nível considerável, são classificadas como catástrofes.

As **incidências**, são ocorrências de incidentes cobertas pelo contrato de seguro (apólice), que pode resultar na compensação do segurado por parte do segurador.

Durante a resolução de uma reclamação (figura de suporte 3.4), apuram-se as responsabilidades, identificam-se as vítimas e as reservas destinadas a este efeito são acionadas de forma a compensar os danos causados. A abertura do processo é o momento onde para a primeira avaliação e para o provisionamento dos custos de compensação e regularização do sinistro. Estas disposições irão ser reavaliadas em fases posteriores e a seguradora será a responsável por determinar de quem são as responsabilidades. Serão canceladas no momento de reivindicação.

O processo de **subscrição** é fundamental para empresa de seguros visto que este processo assegura a seleção e precificação de riscos adequada.



Figura 3.4: Resolução de um Incidente

### Re-seguro e Cosseguro

Como existem inúmeros clientes a estabelecerem contratos com as seguradoras, estas tornam-se incapazes de suportar todos os custos só por si próprias. Surge então o conceito de dispersão de risco, que pode ser feito através de um re-seguro ou de um co-seguro.

### Resseguro

O resseguro é um seguro sobre um seguro já existente distribuindo desta forma o risco verticalmente pelas diversas entidades envolvidas. A entidade seguradora efetua uma transferência de parte das responsabilidades assumidas para uma outra entidade seguradora, mediante o pagamento de um prémio. O segurado não precisará de ter conhecimento da ligação entre seguradoras.

### Co-seguro

É um tipo de seguro onde múltiplas seguradoras assumem o risco de forma equitativa.

Estes seguros são definidos numa única apólice, onde todas as seguradoras envolvidas indicam a proporção de risco coberto, sendo que geralmente existe um líder que emite esta apólice e as outras partes envolvidas apenas se limitam a assinar. Este líder ficará encarregue de gerir a apólice e tomar decisões, tanto em seu nome como em nome das outras seguradoras. As restantes partes envolvidas, apenas respondem perante as partes que tenham assumido responsabilidades.

### **Ramo Não-Vida**

Nos dias de hoje, é possível segurar quase tudo, desde seres humanos, animais até a bens materiais, como carros, casas, entre outros. Como é lógico, um seguro para um ser vivo terá de ser muito diferente de um seguro para algo que não possui vida, a estes seguros, dá-se o nome de Não-Vida.

Os seguros de veículos são, talvez, os mais conhecidos e mais utilizados no quotidiano. Tendo coberturas que podem ir desde as quebras de vidros até aos passageiros do automóvel, estes seguros têm como particularidade as suas tipologias: seguros para clientes particulares que possuam um ou mais carros, neste caso existem pacotes, que servem para que os mesmos possam segurar de uma só vez todos os seus veículos, seguros de frota, que servem para os clientes empresariais segurarem todos os veículos da sua empresa, seguros sobre a carta de condução, que facilita protege todos os clientes cuja profissão envolva lidar com vários veículos, como é o caso dos mecânicos ou mesmo dos taxistas e existem ainda os seguros de fronteira, cujo objetivo é segurar veículos fora de fronteiras sem haver a necessidade de criar um seguro no país em questão.

Os seguros de invalidez e de doença, embora sejam referentes a pessoas, são considerados seguros de não vida pois não se destinam diretamente ao segurador, mas sim a terceiros, neste caso a trabalhadores de uma determinada empresa. Estes seguros, tal como o próprio nome indica, cobrem os riscos que os trabalhadores correm, tanto dentro da empresa como fora dela. Estes seguros, além da componente obrigatória que garante que os trabalhadores são compensados em caso de acidente em trabalho, podem ainda ter cláusulas que se estendem às viagens que estes façam pelas empresas. Num âmbito mais geral, existem seguros contra todos os riscos que cobrem vários riscos tanto a nível pessoal como material. Outros exemplos são os seguros contra incêndios, que cobrem não só incêndios como catástrofes naturais, ou roubos, que protegem apenas os bens relativos à empresa e não os trabalhadores.

Ao género dos seguros de veículos, existem também os seguros marítimos e de trans-

portes, que visam cobrir riscos relativos ao transporte de bens, embarcações de recreio ou mesmo danos causados em navios.

Seguros de responsabilidade civil, que são utilizados por profissionais cujo trabalho envolva a afetação de outros indivíduos, como são os casos dos médicos, dos advogados ou mesmo de auditores. Em Portugal, por exemplo, existe ainda um seguro obrigatório por lei, que protege os trabalhadores de uma empresa no caso de sofrerem um acidente em trabalho.

De importante notar que estes são os tipos de seguros mais comuns, existindo ainda muitos outros tipos por este mundo fora.

### **Ramo Vida**

Estes seguros destinam-se unicamente a pessoas, pois apenas cobrem os riscos relativos às mesmas. Podem ser divididos em duas categorias: os seguros de vida e os seguros de morte. Sendo as coberturas semelhantes para ambos, a principal diferença está na altura em que estes são ativados, num seguro de morte apenas é pago o prémio, de uma vez ou de forma faseada, após o falecimento do segurado enquanto que os seguros de vida apenas são pagos apenas se o segurador se encontrar vivo à data de fim do contrato. Podem haver ainda seguros mistos, onde os prémios são pagos, em caso de morte, até um período contratado, ou, em caso de vida, no fim do período contratado.

### **A aplicação**

*The Insurance Application* é uma aplicação que acompanha todos os processos de seguros, por mais variados que sejam, disponibilizando para tal funcionalidades transversais bem como a possibilidade de gerir como se comportam todas as suas funcionalidades [Application, 2014].

Na aplicação existem alguns módulos, cada um deles destinado a lidar com cada componente referente ao mundo dos seguros. Esta aplicação tem como objetivo permitir aos seus utilizadores, controlarem todo o processo relativo à criação e gestão de apólices, incluindo a abertura e resolução de incidências.

O funcionamento da aplicação desta aplicação divide-se em diversos módulos, dos quais se destacam o módulo de gestão das **partes** constituintes de uma apólice, como o segurado, o segurador, entidades externas, entre outras, o módulo de **incidências** que permite gerir as reclamações efetuadas por parte de um segurado quando este ocorre num sinistro: gestão de correspondência, de pagamentos, efetuar um seguimento da

incidência, entre outros e um módulo de gestão das **apólices** onde é ainda gerada, distribuída e armazenada toda a documentação necessária. Relacionado com este módulo existe também um outro, o módulo de re-seguro, onde são registados os contratos e calculados os prémios relativos aos mesmos.

Existem também dois módulos financeiros **conta** e **comissão** que permitem ao utilizador gerir cobranças ou pagamentos relativos às entidades seguradas, sendo possível desta forma manter também um controlo dos fundos da seguradora. Numa perspetiva secundária, existem os módulos de gestão de disputas, que se integra com todos os outros módulos da aplicação e que permite controlar toda a documentação existente, um módulo de autorização onde são geridos os acessos da aplicação TIA.

### 3.1.4 COBOL

A *COmmon Business Oriented Language* é uma linguagem de programação - a segunda linguagem de alto nível mais antiga -, orientada para os negócios, cujo principal propósito é o processamento de informação em mainframes <sup>1</sup>. A COBOL é uma linguagem bastante prolixa e ao contrário do que o público em geral pensa, não está morta, muito antes pelo contrário. Muitos sistemas de banca ou seguros são implementados em COBOL devido à sua robustez e estabilidade. [Microfocus, 2003]

Apesar de ser uma linguagem bastante prolixa, esta tem bastantes aspetos que a tornam única, como é o caso da divisão em cinco colunas, da área de codificação, sendo que cada uma delas tem uma função diferente: número de sequência, que representa a linha de código, uma área de indicador, que apenas suporta três tipos de caracteres, dos quais e por exemplo, o asterisco serve para comentar, ou não, o linha de código e ainda possui duas outras colunas que são a área A e área B respetivamente. Cada uma destas áreas destina-se a diferentes níveis, embora muitos compiladores já não façam distinção entre estas duas áreas, a primeira só serve para registos do tipo 01, as restantes aparecem na área B. Estes registos são utilizados para especificar e distinguir os diferentes tipos de dados bem como para distinguir dados de outras operações, como por exemplo a cópia de valores entre dois tipos de dados.

Outro dos aspetos que torna a linguagem COBOL única, são as quatro divisões pelo qual o código é composto:

- Identificação, que contém a informação mais relevante relativa ao programador e

---

<sup>1</sup>Computador de alto desempenho utilizado para computação de larga escala, que necessita de uma melhor disponibilidade e uma melhor segurança que um simples computador de escritório

ao próprio programa;

- Ambiente, que permite configurar aquilo com que o programa irá interagir;
- Dados, onde é definida a alocação de memória a ser utilizada pelo programa;
- Procedimento, que é onde se coloca a lógica do programa, ou seja, o que é que este irá executar.

[zingCOBOL, 2014]

Mais uma vez e como nas outras formações, esta foi baseada em casos reais, nomeadamente baseado em casos utilizados em aplicações para bancos.

## Capítulo 4

### Projeto Mais Máquina Kernel

A 24 de Outubro de 2013, fui integrado num projeto de um dos atuais clientes da Deloitte, a Unicre. Este projeto, de nome Mais Máquina Kernel (MMK), consiste na renovação de toda a arquitetura da Unicre, sendo planeada e construída toda uma nova arquitetura em detrimento da anterior, que não cumpria com as atuais necessidades do cliente.

A Unicre foi criada em Abril de 1974 como uma *joint-venture*<sup>1</sup> de vários bancos portugueses com a finalidade de desenvolver o negócio de cartões de pagamento em Portugal, sendo que foi a única empresa autorizada a emitir cartões de crédito (negócio de *issuing*) até 1988, e a prestar serviços de *acquiring* para marcas internacionais até 1991.

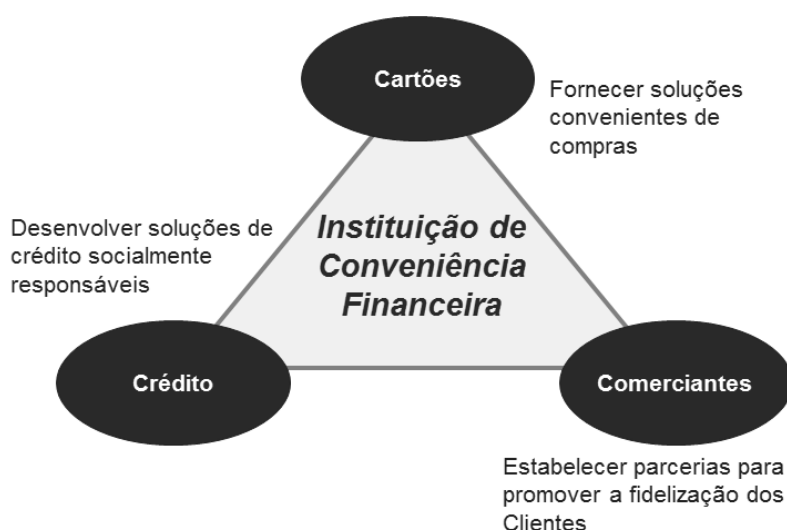


Figura 4.1: Áreas de negócio da Unicre

<sup>1</sup> Associação, definitiva ou não, de várias empresas de forma a explorar determinada área de negócio

Especializada na gestão e emissão de cartões de pagamento, a Unicre possui duas marcas de negócio e atua em três vertentes: cartões de crédito, soluções para aceitação de pagamentos e serviços para outras instituições.

Desta forma, a Unicre disponibiliza através destas marcas, serviços de *acquiring* que consistem na disponibilização de soluções para aceitação de pagamentos em estabelecimentos comerciais, presenciais e virtuais com cartões estrangeiros dos sistemas internacionais Visa, MasterCard, Visa Electron, Maestro, VPay, Diners e JCB. Por outro lado, os serviços de *issuing*, consistem na emissão e gestão de cartões de pagamento, com marca própria e em parceria, para clientes particulares e empresas, disponibilizando ainda, crédito pessoal. [Unicre, 2012]

## 4.1 Responsabilidades e Tarefas

Este projeto está em divido em duas áreas distintas, cada uma delas relativas a cada uma das áreas de negócio do cliente: uma componente relativa a clientes empresariais (componente de *acquiring*) e uma outra componente relativa aos clientes particulares (componente de *issuing*). Por sua vez, cada uma das áreas de negócio está sub-divida em componentes, existindo para tal diferentes equipas de forma a facilitar e agilizar a organização e o desenvolvimento destas mesmos componentes:

- **Blocos** - Que funciona como um intermediário de comunicação, é responsável por agendar reuniões com a direção da Unicre, encaminhar e registar os pedidos de passagens a produção e ainda são responsáveis por gerir e controlar o estado das ocorrências abertas pelo cliente, no fundo é como se fosse uma interface entre a Deloitte e o cliente.
- **Ágil** - Mesmo sendo o Mais Ágil um projeto distinto, esta equipa faz também parte do projeto MMK. Encarrega-se pelo desenvolvimento de uma aplicação implementada com recurso à tecnologia Softwell e à respetiva integração da mesma, via *webservice* com as restantes componentes do MMK;
- **Canais** - É a equipa que implementa e mantém todas as interfaces para o utilizador final, desde os portais de comerciantes até aos portais de *backoffice*. Esta equipa está também responsável por manter e gerir todos os *webservices* e *schemas* relacionados, não só, com os portais como também com a gestão de utilizadores.
- **Integração** - Sendo a maior equipa do projeto, esta está encarregue por toda a

componente de integração, ou seja, todo o motor do projeto, os diferentes meios de comunicação entre diferentes componentes. Esta equipa tem várias responsabilidades, estando mesmo dividida em componentes síncronos e componentes assíncronos. Estas responsabilidades passam pela manutenção de *schemas*, *webservices* tanto de comunicação interna como de intercomunicação de componentes, pela manutenção de documentação com informação proveniente das principais fontes de dados, nomeadamente do *core*, pela geração de relatórios operacionais, entre outras.

- **Gestão de Relação com o Cliente (Customer Relationship Management - CRM)** - Está responsável pela manutenção e gestão de uma das maiores ferramentas de interação com o cliente, o Siebel.
- **Business Intelligence** - A equipa de *Business Intelligence* está responsável pelo tratamento e transformação de dados provenientes do software de processamento de pagamentos (*core*) e ainda pelo repositório central de informação, onde esta e outra informação será armazenada. Toda a validação de informação a ser disponibilizada aos restantes componentes passa maioritariamente pelos componentes desta equipa.

Estas equipas são compostas por diversas secções de forma a garantir uma melhor organização e delegação de tarefas. A minha participação foi feita maioritariamente na equipa de integração, mais concretamente na equipa responsável por um dos componentes mais importantes do projeto, o Gestor de Tangíveis (ver secção 4.6). Como integrante desta equipa, que nos primeiros três meses foi composta por mim e por uma pessoa já com experiência na mesma ficando depois reduzida apenas a mim, fiquei responsável por me assegurar de finalizar os desenvolvimentos, manter e evoluir todas as peças deste mesmo componente: *webservices* SOA, ferramenta de *reporting*, desde a recolha de dados à geração dos documentos finais, pelo *middleware* Java responsável pelas diferentes expedições e operações avançadas sobre documentos, como justaposição e ainda pelas base de dados de suporte. Pontualmente, estive ainda integrado na equipa de *Business Intelligence*, a prestar suporte nas suas aplicações de processamento de informação desenvolvidas em Java.

Além do projeto MMK, viria ainda a existir um novo projeto, o Suporte Perfeito e Evolutivo (SPE). Este projeto consiste, tal como o próprio nome indica, no suporte e evolução dos componentes desenvolvidos pelo projeto MMK e que já se encontram em produção. O projeto SPE mantém-se em paralelo com o projeto MMK, uma vez que este passou apenas a concentra-se na área de *Issuing*.

### 4.1.1 Responsabilidades

As duas primeiras semanas e uma vez que entrei no projeto MMK, que já se encontrava em fase de implementação, foram passadas a estudar o modelo de negócio, de forma a ganhar algum conhecimento sobre termos e aspetos do negócio, como também os desenhos técnicos e funcionais das componentes em desenvolvimento, de forma a possibilitar um melhor entendimento e interação não só com as outras equipas mas como também com o próprio cliente. Uma vez que não tinha qualquer experiência com as tecnologias utilizadas, despendi algum deste tempo a familiarizar-me e explorar as mesmas, tentando sempre absorver o máximo de conhecimento das pessoas que já as trabalhavam.

Tal como em qualquer projeto, as responsabilidades foram crescendo com o passar do tempo. Comecei como *analista* na área de integração, que tendo alguém superior responsável por mim, foram-me sendo atribuídas tarefas de correção e evolução de funcionalidades já existentes acompanhadas de passagens de conhecimento que me permitiram, ao longo do tempo, ganhar conhecimento e tornar-me auto suficiente.

Neste aspeto fui um privilegiado, pois face à grande quantidade de trabalho a desenvolver, foram-me atribuídas tarefas e responsabilidades de níveis superiores. Ao fim de apenas três meses, graças à boa prestação e também ao facto de a pessoa responsável por mim ter sido movida para outro projeto, já estava responsável por um dos principais componentes do projeto, o Gestor de Tangíveis que será abordado e explicado no capítulo 4.6. Nesta altura estava responsável por finalizar desenvolvimentos ainda em aberto, tanto a nível da lógica e estrutura das base de dados de suporte ao componente, como da lógica presente nos fluxos dos *webservices* relacionados com o mesmo.

Numa ótica mais de manutenção e evolução, as minhas responsabilidades passaram também pela manutenção e evolução da lógica e estrutura das base de dados de suporte bem como da lógica nos fluxos dos *webservices*, inclusivamente melhorias no tratamento de erros e melhorias a nível de desempenho. A manutenção e evolução estendeu-se também aos documentos, tanto a nível de *templates* como, e mais uma vez, a nível de desempenho da geração dos mesmos e ainda na aplicação *middleware* responsável por capacitar a ferramenta de geração documental de funcionalidades que esta não suportava nativamente, onde foram feitos sobretudo trabalhos evolutivos, como o enriquecimento e aperfeiçoamento das mensagens de correio eletrónico expedidas pela mesma.

Após terminada a implementação e o período de garantia destes mesmos componentes de *acquiring*, transferi-me para o projeto SPE de forma a ganhar experiência na área da manutenção. Esta transferência acabou por me garantir novas responsabilidades, uma vez que neste novo âmbito fiquei responsável por outro componente em que já havia participado, o *schema* de base de dados para Suporte Aplicacional e Geração de Informação para Terceiros 4.12. Sobre este componente as minhas responsabilidades recaíram sobre a total gestão da base de dados, não me limitando apenas a melhorá-la tanto a nível de desempenho como dotando-a de novas funcionalidades, mas também a assumir novas responsabilidades de controlo de qualidade, onde tudo o que era desenvolvido, tudo o que era necessário alterar teria de ser aprovado por mim.

Graças ao facto do componente de Gestão de Tangíveis possuir um *middleware* desenvolvido à medida em JAVA, foi-me ainda atribuída a responsabilidade pelos módulos JAVA que transitaram do projeto MMK, relativos ao processamento de informações provenientes do sistema central.

#### 4.1.2 Tarefas

A minha passagem pelo MMK pode ser descrita em duas etapas: uma etapa de consultoria, onde ajudei a analisar e implementar algumas das soluções deste projeto e uma etapa de manutenção, na qual participei no suporte corretivo, perfectivo e evolutivo do MMK.

Aquando da minha entrada no projeto, as minhas tarefas recaíam sobre ganhar conhecimento e familiarizar-me com algumas tecnologias utilizadas no projeto, nomeadamente da Oracle: *Service-Oriented Architecture*, *Business Intelligence Publisher* e PL/SQL.

Passando realmente à interação com o projeto, as primeiras tarefas foram essencialmente a análise de desenhos técnico e funcional do componente. Estas tarefas foram complementadas por tarefas funcionais, tarefas estas que se relacionam mais com a lógica de negócio. Desta forma foi-me possível facilitar a comunicação e o entendimento das necessidades do cliente.

Ao longo do tempo e após ganhar alguma maturidade no projeto através da resolução de problemas detetados pelo cliente, comecei a realizar tarefas relativas à implementação e afinação do Gestor de Tangíveis (GT), sobretudo nos desenvolvimentos da vertente de *Acquiring*. Estas tarefas consistiram sobretudo em três categorias:

- Otimizações que passaram pela análise, identificação e correção de problemas de desempenho encontrados tanto nos desenvolvimentos de *webservices* SOA,

como também de base de dados (obtenção de informação, *stored procedures*) e ainda nos relatórios e modelos de dados utilizados pelo BI Publisher;

- Melhorias que consistiram em adicionar novas funcionalidades a desenvolvimentos já existentes, nas mesmas três vertentes que as otimizações;
- Implementações que resultaram de novas funcionalidades solicitadas pelo cliente e que antes não haviam sido identificadas, sendo que nesta se incluiu também a tecnologia Java.

Após a entrada do projeto de manutenção, juntaram-se algumas tarefas mais relacionadas com gestão, uma vez que fiquei encarregue de analisar e dar suporte a todas as ocorrências que fossem abertas pelo cliente ao mesmo. Estas ocorrências consistem em correções, relativas às alterações no negócio ou algum detalhe que não tenha cumprido com o desenho técnico, que o cliente solicita à Deloitte. As ocorrências podem ser também consideradas melhorias técnicas que não tenham sido identificadas em desenho técnico.

Nesta fase comecei a trabalhar diretamente e em primeira linha com a equipa de desenvolvimento do cliente, chegando mesmo a fazer análise de dados e dos sistemas responsáveis pela produção de informação real.

## **4.2 Tecnologias**

No projeto MMK existe um vasto leque de tecnologias que são utilizadas ao longo dos diversos componentes, chegando algumas a serem mesmo comuns a vários componentes. Numa ótica de integração, as tecnologias mais utilizadas são as da Microsoft, como os seus serviços de relatórios e integração, bem como os serviços da Oracle, sobre os quais os seguintes tópicos se vão focar, uma vez que são utilizadas pelos componentes da minha responsabilidade.

### **4.2.1 Aplicações**

Além das aplicações criadas e mantidas por entidades externas, tal como é o caso do Way4, existem aplicações criadas no âmbito do projeto MMK. Estas aplicações foram desenvolvidas nas mais variadas tecnologias, tendo sido implementadas à medida das novas necessidades do cliente, algumas destas necessidades por serem tão específicas tiveram de ser implementadas completamente de raiz e com o acompanhamento do cliente.

Graças à sua maturidade e ao seu custo, o JAVA acabou por ser adotado em diversos desenvolvimentos e mesmo que sejam relativos a componentes distintos, estes têm todos um objetivo em comum: o tratamento de informação, seja esta proveniente de componentes MMK ou a componentes externos.

Posteriormente estas aplicações serão integradas com o Oracle Weblogic, mais especificamente com o *Business Intelligence Publisher* e ainda com o Control-M, tal como será abordado em capítulos posteriores.

### **Responsabilidades e Tarefas**

As aplicações Java existentes no projeto, nomeadamente de processamento de dados e de tarefas afetas ao Gestor de Tangíveis, já se encontravam praticamente implementadas. Desta forma as minhas responsabilidades passaram por geri-las, mantê-las e evolui-las. Estas tarefas consistiram essencialmente em análise e implementação, investindo, numa primeira fase pré-produção, algum tempo a identificar lacunas, recolher informação da experiência dos utilizadores e validar que os fluxos, tais como duplicação de documentos, funcionavam corretamente. Graças a estas tarefas, foi ainda possível identificar pontos de melhoria, até porque com a realização de testes com os utilizadores da área de negócio, verificou-se que alguns pormenores funcionais, tais como os *templates* das mensagens de correio eletrónico não se encontrarem *on brand*<sup>2</sup> ou os pré-impresos utilizados nos documentos finais estarem desatualizados ou incorretamente posicionados.

Uma vez que ao longo de todo o projeto sempre surgiram questões relativas ao desempenho dos fluxos, realizei também tarefas relativas a melhoria de desempenho, tanto na própria lógica destas aplicações como nas invocações a aplicações ou bibliotecas externas. Estas tarefas foram sempre sustentadas por tarefas de análise, feitas através de ficheiros *log* de forma a perceber volumes de informação, tempos de execução e taxas de sucesso.

### **4.2.2 Oracle Weblogic**

O weblogic é um servidor de software que corre numa camada intermédia entre as base de dados e as aplicações. Este componente integra o *middleware* Oracle Fusion,

---

<sup>2</sup>Estar dentro dos padrões visuais da marca

que consiste em diversos produtos Oracle que abrangem as áreas de *business intelligence*, desenvolvimento colaborativo, gestão de conteúdo e serviços de integração [Alapati, 2011].

Tendo uma oferta constituída por três produtos distintos *Server Standart Edition*, *Server Enterprise Edition* e Suite todos baseados na plataforma Java 2 Platform, Enterprise Edition (J2EE), estes fornecem um conjunto de módulos e serviços que lidam com muita da programação e o comportamento da aplicação de forma automática. Neste âmbito, será apenas considerado a versão Weblogic Suite onde serão abordados também dois dos seus módulos: *Service-Oriented Architecture* e *Business Intelligence Publisher*.

Funcionalmente, o weblogic possui dois tipos de servidores: *admin* e *managed*. O **admin** está responsável por gerir e coordenar os servidores trabalhadores, fornecendo acesso a todas as operações de configuração e manutenção das aplicações executadas no weblogic. Existe sempre um servidor em cada *pool*<sup>3</sup> podendo este controlar diversos *clusters*<sup>4</sup> de servidores distintos (SOA, BIPublisher, OSB, etc). Por outro lado, os **managed** são os servidores trabalhadores onde são executadas as diversas aplicações. De forma a garantir redundância, balanceamento de carga e alta disponibilidade, é comum existir mais do que um servidor trabalhador. Nestes servidores é disponibilizado acesso, visualização e controlo centralizados a todas operações que sejam executadas no weblogic.

### **Responsabilidades e Tarefas**

O Weblogic é um servidor transversal a toda a arquitetura do projeto MMK, sendo assim utilizado por diversas equipas da área da integração. De forma a que exista uma boa coesão entre as mesmas, todas estas equipas tiveram de desenvolver tarefas extra, nomeadamente tarefas de documentação das suas configurações, como ligações à base de dados e tarefas de monitorização de sistemas. O componente de Gestão de Tangíveis não é uma exceção e tendo em conta as suas particularidades, como o servidor de geração documental e o *middleware* agregado à mesma, fui encarregue de executar tarefas de monitorização, de forma a despistar alguns erros relacionados com más configurações, aplicativos ou de infraestrutura, e ainda de forma a dar suporte às equipas da casa que estavam responsáveis pelas diversas infraestruturas.

---

<sup>3</sup>Uma *pool* representa um espaço em memória onde são criadas diversas *threads* de forma a executarem diversas tarefas.

<sup>4</sup>Um cluster consiste num grupo de servidores que trabalham como um só.

### 4.2.3 Service-Oriented Architecture

A arquitetura orientada a serviços (SOA) é a estrutura subjacente que suporta as comunicações entre serviços, por outras palavras, é o SOA que define como é que duas entidades interagem entre si. Estas comunicações podem envolver uma simples troca de dados ou a coordenação de execução de uma determinada atividade [Oracle, 2014c]. Uma vez que o termo *webservice* vai ser mencionado várias vezes ao longo deste documento, é importante compreender que estes consistem em encapsulamentos de lógica e processos que são expostos através de interfaces. Esta solução standard é utilizada na integração de sistemas e na comunicação de diferentes aplicações, mesmo que estas utilizem diferentes tecnologias. Esta standardização é garantida graças à utilização da linguagem XML, mesmo na definição das interfaces feita com recurso a contratos WSDL e também graças à comunicação com recurso aos protocolos SOAP<sup>5</sup> e HTTP.

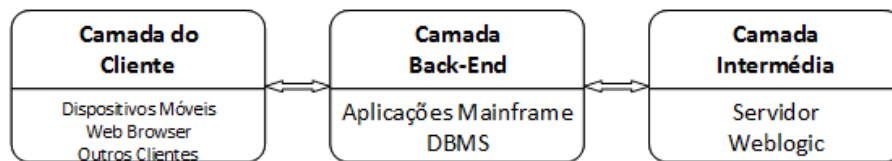


Figura 4.2: Arquitetura Weblogic

### Oracle SOA Suite

Para implementar esta arquitetura orientada a serviços, recorreu-se ao Oracle SOA Suite. Este *middleware* consiste num conjunto de produtos que permitem, de forma simples e rápida, desenhar, implementar, disponibilizar e gerir aplicações. Estas são construídas com recurso a um modelo baseado em componentes e serviços (SCA): cada aplicação - denominada por Composite - é composta por diversos subcomponentes cooperativos que implementam o processo de negócio pretendido.

<sup>5</sup>Protocolo para troca de informação estruturada em plataformas distribuídas.

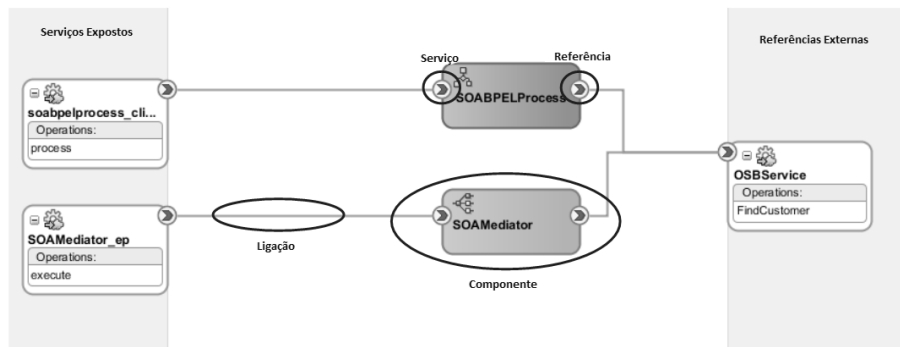


Figura 4.3: Principais Noções do Modelo SCA

Associado a este modelo além das noções de *composite*, de serviço e de componente, estão ainda associadas as noções de referência, que representam ligações a componentes externos, como base de dados, e de ligação que serve para estabelecer um fluxo entre dois componentes.

São quatro os tipos de componentes que implementam a lógica de negócio e a orquestração pretendida numa aplicação SOA:

1. Mediator funciona como um *router*, é o componente quem tem as responsabilidades de transformação e reencaminhamento simples de mensagens. Este reencaminhamento é feito com base no cabeçalho e no conteúdo das mensagens e ainda em regras definidas noutra componente.

O mediator permite também validação de dados com recurso a XSD e respetivo tratamento de erros. É igualmente possível validar os parâmetros de entrada do serviço com recurso a ficheiros *schematron*.

Através de uma única invocação é também possível definir diferentes sequências de acordo com a informação que é recebida, com a exceção de que os retornos dos serviços não podem ser utilizados para a invocação de um outro, neste caso existe a necessidade de recorrer a um BPEL.

2. O BPEL é um componente que permite a construção da orquestração de um serviço. Além de fornecer atividades básicas e estruturadas de programação, como são o caso das cláusulas IF e do tratamento de falhas e erros, é através deste componente que são feitos os acessos externos ao serviço, com recurso a *partner links*. Os *partner links* têm associados um tipo que caracteriza a conversação entre dois serviços, definindo as regras (*roles*) desempenhadas por cada um destes e que especifica o *port type* utilizado por cada serviço para receber mensagens, uma vez que um serviço pode expor várias operações e cada uma delas conter

uma estrutura própria [Oracle, 2013]. A **idempotent** é uma propriedade definida nos *partner links* em que uma atividade não é repetida. Necessário para evitar que o servidor reprocessasse mensagens após shutdown.

3. A tarefa humana (*Human Task*) estende um processo automatizado para integração com a intervenção humana. Estas tarefas fornece uma *worklists* que é uma interface disponibilizada para que cada utilizador possa gerir as suas tarefas.
4. As regras de negócio servem para incluir a lógica de negócio nos fluxo de um determinado serviço. Estas regras estão separadas do código para permitir a sua modificação por parte dos utilizadores não dotados tecnicamente [Oracle, 2009].

### Implementação de um serviço SOA

O ambiente de desenvolvimento (IDE) utilizado para a implementação de serviços SOA é o JDeveloper, uma ferramenta proprietária da Oracle. Esta disponibiliza a *framework* de componentes SCA e diversos utilitários de apoio ao desenvolvimento.

Tal como acontece noutras linguagens de programação que foram abordadas a nível académico, este IDE fornece duas interfaces de visualização: um modo de desenho e um modo de código. Através deste IDE é ainda possível, de forma simples, a integração com servidores *weblogic*, base de dados e com serviços de metadados.

Outra semelhança com as linguagens de programação são as variáveis. As variáveis no SOA Suite são definidas através de elementos ou mensagens definidas num XSD. Estas variáveis são utilizadas para passar valores de entrada e receber valores de saída tanto nos BPEL, nos *composites* como nos adaptadores, bem como podem ser utilizados como variáveis auxiliares, como por exemplo, definir um contador.

A atribuição de valores para estas variáveis é feita através *assign*, que servem para fazer atribuições e mapeamentos simples e através de transformações, que servem para efetuar atribuições e mapeamentos mais complexos. Independentemente da atividade utilizada, a variável de destino será sempre uma enquanto que, por outro lado, a informação que lhe vai ser atribuída provém de N variáveis.

Existe também um conceito de **preferências** que consistem em configurações a usar em tempo de execução. Estas podem ser alteradas pelo *configuration plan*, tal como será descrito mais à frente, e permitem definir valores de variáveis a usar em processos BPEL. Estas variáveis são normalmente usadas para definir *timeouts* consoante o ambiente, definir o número de tentativas e os nomes de ficheiros.

Um serviço SOA é composto por diversos objetos, que interligados entre si permitem a execução de um fluxo que quando invocado desencadeia uma ou mais ações.

## Adaptadores

Os adaptadores servem para interligar os *composites* com as restantes aplicações através da criação automática de interfaces de acesso via serviço.

Existem diversos tipos de adaptadores, permitindo cada um deles acesso a um objeto externo. De todos os tipos que existem, apenas quatro são utilizados no projeto MMK:

- Adaptadores de base de dados que são, de longe, os mais utilizados. Estes adaptadores permitem invocar *stored procedures* e códigos SQL, como *inserts*, *updates*, *deletes*, entre outros, e aquando da sua seleção, é criada uma interface de interação com a base de dados. As transações efetuadas à base de dados através do adaptador, serão geridas pelo mesmo, criando uma transação quando é despoletado e terminando-a quando a sua execução chega ao fim.

Quando são executados em *design time* usam as ligações definidas no JDeveloper enquanto que em *runtime* utilizam JNDI <sup>6</sup> para aceder às *connection pools* definidas no servidor. No servidor será criado uma *jdbc datasource*<sup>7</sup> para cada *schema*.

- Adaptadores spring permitem interagir com classes Java e são utilizados para realizar processamentos complexos. No momento da sua criação, é automaticamente adicionada uma interface via *webservice* que com as operações equivalentes aos métodos da classe Java, sendo que o objeto de retorno é igualmente convertido numa mensagem de saída.
- Adaptadores de ficheiros são utilizados para interagir com ficheiros, tanto para escrita como para leitura. Os ficheiros podem ser consumidos através de uma estrutura bem definida, sem processamento de conteúdo (*trigger files*) ou ainda podem ser consumidas de forma binária.

É possível definir propriedades dos adaptadores na lógica da aplicação e também durante a sua execução. Para atribuir um valor a uma propriedade em tempo de execução basta fazer um ou vários *assigns* para as variáveis definidas, podendo assim o adaptador utilizar estas propriedades. Por outro lado, as atribuições a nível da aplicação podem ser definidas com valores estáticos. Uma das propriedades que são definidas ao nível dos dados da mensagem, as **correlações**, permitem interligar pedidos como respostas e vice-versa. Estas garantem o encaminhamento correto de mensagens assíncronas em fluxos com N invocações a um mesmo *Web Service*, estabelecendo para

---

<sup>6</sup>Java Naming and Directory Interface é uma API que permite aceder a diferentes serviços de diretorias ou de gestão de nomes.

<sup>7</sup>Java DataBase Connectivity fornece acesso e configuração de acesso a uma base de dados.

tal conversações através de ID's.

## Atividades

As atividades consistem em objetos SCA utilizados no desenvolvimento de aplicações SOA. Estes objetos podem ser agrupados em duas categorias:

- Atividades básicas, que incluem as invocações de interfaces, atualização de variáveis e tratamentos de erros e falhas:

- Numa perspetiva de manipulação de dados, a atividade de **assign** permite a atribuição de valores a variáveis recorrendo a valores fixos ou mapeamentos simples de outras variáveis. Estas atividades são utilizadas tipicamente para preencher variáveis de entrada e/ou saída de uma determinada interface, podendo ainda ser utilizadas para atribuir valores a variáveis de controlo.

A utilização destas atividades são recomendadas para atribuições simples e com pouca lógica associada, conseqüentemente, estas atividades fornecem um maior desempenho pelo que deverá ser dada preferência à sua utilização sempre que possível;

O **transform** serve para modificar informação com recurso a transformações XSL (XSLT). São executadas num processador específico permitindo a criação de uma lógica mais complexa, como por exemplo o mapeamento entre diferentes tipos de dados. Ao contrário da *assign*, esta atividade constrói todos os elementos e nós do XML, deixando sem valor todos aqueles nós que não possuam um mapeamento.

- As invocações a outras interfaces (**invoke**), dividem-se em síncronas e assíncronas, sendo o seu tipo escolhido em função do propósito do serviço implementado. Os pedidos que recorrem a sincronismo, são feitos nos dois sentidos, ou seja, existe um pedido bloqueante que aguardará sempre por uma resposta. Por outro lado, os pedidos assíncronos podem ser feitos num único sentido através de um pedido executado e esquecido (*fire-and-forget request*) ou ainda através de um pedido não bloqueante que poderá ou não obter uma resposta (*request-callback*).

- As atividades de interação com *partner links* também pode ser divididas em síncronas, onde as interações são feitas com recurso a *invokes*, e em assíncronas, podendo ser feitas num sentido ou em ambos, sendo que quando feita em ambos recorre-se ainda a uma atividade *receive*. Tipicamente um serviço síncrono implementado com recurso a um BPEL é iniciado por um

*receive* e terminado com um *reply*.

- Os erros e falhas podem ser gerados explicitamente pela atividade *throw* ou pela ocorrência de uma falha de sistema não contemplada. Estes erros e estas falhas são capturadas com recurso às atividades de *catch*, que apenas capturam um tipo particular de falha, e *catch all* para apanhar qualquer tipo.
  - Existem ainda outras atividades básicas que são frequentemente utilizadas na construção dos fluxos, como são os casos das atividades *wait*, que permitem parar/esperar X segundos durante uma execução de um serviço, das atividades *terminate* que são utilizadas para forçar explicitamente o término do serviço e das atividades *do nothing* que consistem em objetos vazios que podem ser utilizadas como uma simbolização de um desenvolvimento ainda por fazer ou como o tratamento de uma falha que deverá ser identificada mas não deverá ser tratada.
- Atividades estruturadas que permitem controlar os fluxos e respetivos comportamentos:
    - Os **Scopes** definem os segmentos de execução com controlos de erros, definição de variáveis e propriedades;
    - As **sequências** servem unicamente para agregar atividade e permitir uma simplificação da complexidade;
    - Condições **IF-THEN** implementadas com recurso a *switch*. O *switch* funciona como um desmultiplicador, onde de acordo com uma ou mais condições, o fluxo assume um caminho diferente;
    - Repetições de atividades, que funcionam do mesmo modo que um ciclo *while* numa outra linguagem de programação;
    - A execução paralela é garantida através das atividades *flow* e *flowN*, sendo que a primeira executa um determinado número de ramos definidos em *design time* enquanto que a outra executa um mesmo ramo N vezes, onde o N é definido em *runtime*.Juntamente com a execução paralela são utilizadas atividades como o *onMessage* e o *onAlarm* que recebem eventos específicos ou uma interrupção passados N segundos, respetivamente, executam um ou mais comportamentos. Existe ainda uma atividade *pick*, que é utilizado no interior de um *onMessage* e que permite seleccionar um tipo de mensagem específico e desencadear a atividade apropriada.

Em determinadas casos podem existir dependências de informação proveniente de outros processos. Para garantir a sincronização entre processos BPEL distintos, são utilizados **sinais** que recorrem a duas atividades BPEL *signal* e *receiveSignal* para estabelecer a troca de informação. Neste comportamento um dos processos invoca o segundo e assume um papel de mestre, isto é, assume o comando da comunicação. Para tal é necessário ativar uma propriedade na invocação ao processo, ***invoke as detail***, para que os dois processos partilhem o mesmo contexto.

### **Deploy de um serviço SOA**

Após o desenvolvimento local de um serviço, existe a necessidade de torna-lo acessível. Antes da colocação nos servidores dos diferentes ambientes, tem de ser efetuada a configuração dos *endpoints* em função do ambiente destino e ainda substituir as configurações de *design time* pelas de *run time*, sendo que estas configurações são colocadas em ficheiros próprios, os *configurations plans*. Os *config plans* consistem em ficheiros XML com instruções de substituição de valores respetivos a variáveis de ambiente. Estas instruções podem ser aplicadas a valores definidos dentro da especificação de um *composite* e ainda a valores de localização de ficheiros XSD incluídos ou importados em ficheiros WSDL.

Com todas as condições reunidas, o *deploy* pode ser feita diretamente do IDE de desenvolvimento, o jDeveloper ou ainda com recurso à ferramenta ANT. O método com recurso ao ANT consiste na execução de um *script* JAVA, que permite automatizar a compilação de aplicações, muito ao género dos *makefiles* em C.

Para as diferentes tecnologias existem diferentes ambientes, tal como será explicado num capítulo mais à frente, como tal é necessário a existência de vários ficheiros de configurações (*config plans*), cada um deles adaptado e configurado para o respetivo ambiente.

Uma vez que todos os ambientes têm localizações distintas, existem também ficheiros de propriedades que armazenam toda a informação necessária, tanto para apontar o *deploy* para a máquina bem como as credenciais para efetuar a autenticação na mesma.

### **Metadata Service**

O serviço de metadados (MDS) consiste numa tecnologia de repositório orientado a objetos, que fornece aos seus utilizadores uma gestão simplificada de todos os artefactos utilizados pelos componentes *middleware*.

Com o MDS é possível partilhar e reutilizar os metadados entre as aplicações, podendo

ao mesmo tempo ser gerido o ciclo de vida de um artefacto bem como as suas configurações específicas para um determinado ambiente.

Uma vez que o projeto MMK é constituído por diversos componentes distintos, existe a necessidade destes saberem como comunicar entre si. Desta forma, todos os artefactos, mais concretamente os WSDL's e XSD's que definem as interfaces dos serviços, que permitem consumir estes mesmos serviços, encontram-se guardados neste mesmo repositório MDS. O MDS contém ainda configurações, separadas por ambiente, relativas aos endereços de todos os serviços e servidores envolvidos, como são os casos dos serviços de gestão documental ou mesmo dos servidores SOA.

### **Responsabilidades e Tarefas**

Como mencionado anteriormente, o SOA é uma das aplicações integrantes do weblogic e tal como este, existem diversas equipas a fazer uso do mesmo. Assim sendo, cada equipa deverá estar responsável por manter e evoluir os serviços associados ao seu componente.

Dado a sua importância e dimensão no projeto, todas as equipas tiveram de assumir responsabilidades na comunicação de alterações aos seus serviços bem como, análise de forma a identificar e mitigar os impactos que as mesmas possam ter nas componentes circundantes. É ainda crucial que as equipas efetuem constantes tarefas de monitorização de serviços para que estes não falhem ou que quando falhem, o seu não funcionamento não afete nenhum outro componente.

É também da responsabilidade de todas as equipas que possuam componentes que façam uso do SOA, manterem bem documentadas e especificadas as interfaces dos seus serviços bem como as localizações dos mesmos, de forma a que quando surgir a necessidade de integração com outros componentes, esta seja simples e limpa. Aqui surge uma outra responsabilidade realizada com o MDS, é essencial que todas as equipas realizem tarefas de atualização de ficheiros WSDL e XSD no mesmo, uma vez que é por aqui que todos os componentes vão importar as definições de *webservices* externos.

No meu caso em específico, tive a responsabilidade de garantir a correta documentação, especificação e exposição de todos os *webservices* relacionados com o Gestor de Tangíveis, não só no que diz respeito ao componente de SOA mas como também ao *middleware* agregado ao BI Publisher.

## 4.2.4 Business Intelligence Publisher

O Business Intelligence Publisher [Oracle, 2014a] é uma solução que oferece a mais eficiente solução de geração e disponibilização de documentos para ambientes complexos e distribuídos. Fornece uma arquitetura central de geração e fornecimento, de informação de uma forma segura, e no formato correto.

O BIP trata-se de uma aplicação J2EE Standard e é composta pelos seguintes módulos aplicativos:

- Motor de extração de dados (*Data Extraction Engine*) que possibilita a ligação, extração e o processamento de dados provenientes das diferentes fontes de dados suportadas;
- Motor de geração de documentos (*Layout Rendering Engine*) no qual é efetuado o mapeamento entre os diversos modelos de documento e os seus respectivos XML de dados, resultando no documento final;
- Motor de distribuição de documentos (*Document Delivery Engine*) que permite a entrega dos documentos finais através de diversos canais de expedição (correio eletrônico, impressão, fax, WebDAV <sup>8</sup>, FTP, SFTP, AS2, HTTP ou outros canais personalizados). Neste módulo é também feito o acompanhamento de cada entrega, existindo também a possibilidade de reprocessar uma entrega mal sucedida.

Destes módulos destacam-se quatro processos específicos que estão diretamente ligados com a fase de composição do documento e para os quais existem blocos respetivos na arquitetura do BIP, responsáveis por lidar com as fontes de informação, modelos e documentos:

1. Processador de Documentos (*Document Processor*) permite ao utilizador criar documentos a partir de múltiplas fontes de informação (dados), combinações de modelos de layout ou ainda através da criação de documentos individuais para cada combinação.  
Fornecendo ao BIP o XML de dados que contém múltiplas fontes de dados e os respetivos modelos de layout, irá resultar na geração de múltiplos documentos finais;
2. Controlador de Dados (*Data Handler*) que é composto por um ou mais motores

---

<sup>8</sup> *Web Distributed Authoring and Versioning* ou WebDAV é uma extensão do protocolo HTTP que permite publicar, bloquear e gerir recursos na Web.

de geração de XML. Pode ser utilizado qualquer motor de geração de XML, tais como: Oracle Reports, Service Beans, entre outros;

Gestor de *Templates* (*Template Manager*) consiste num interface *user-friendly* construído com recurso a uma coleção de API's que permitem ao utilizador registar e manter definições de dados (modelo de dados) e de modelo de layout.

Os modelos de layout usados para produzir documentos finais, são armazenados e geridos no *Template Manager*. Estes são criados com recurso a ferramentas de trabalho com as quais os utilizadores estão familiarizados, como por exemplo o MS Word, MS Excel ou Adobe Acrobat.

3. Gestor de Traduções (*Translation Handler*) oferece ao utilizador a capacidade de registar e manter definições de dados e de modelos de layout associados aos relatórios, num interface *user-friendly*. Aumenta a facilidade e flexibilidade de manter relatórios diários e documentos de negócio, suportando igualmente a funcionalidade de tradução em tempo de execução quando necessária.
4. Gestor de Disponibilização/Entrega de Documentos (*Delivery Manager*) que está responsável por tratar o documento final gerado e disponibiliza-lo ao destino final, através do envio para destinos como: impressora, correio eletrónico, fax, localização *File System*, localização FTP, entre outros.

Focando as funcionalidades do BIP apenas no âmbito deste projeto, este é utilizado como ferramenta de geração documental, uma vez que é aqui que são gerados todos os documentos de negócio, como será abordado no capítulo 4.6.

O processo de geração de documentos é feita com recurso a:

- **Fontes de dados** podendo estas serem dos mais variados tipos, desde ficheiros de texto a base de dados;
- **Modelos de dados** que consistem em objetos que variam consoante as fontes de dados utilizadas, sendo que estes contêm um instruções que permitem ao BIP devolver dados estruturados para determinado relatório. No âmbito deste projeto, esta estrutura é definida pelo utilizador com recurso à linguagem SQL. Estes objetos por sua vez irão dar origem a ficheiros XML que serão, posteriormente aplicados com recurso a XPath, sobre os modelos dos documentos definidos.
- **Templates** que são os modelos dos documentos que irão ser gerados. Mais uma vez e à semelhança das fontes de dados, o BIP suporta uma variada tipologia de documentos que podem ser gerados. No MMK e devido à sua finalidade (gerar cartas para utilizadores finais) são utilizados modelos em *Rich Text Format* (RTF).

Desta forma facilita-se a sua gestão, manutenção e implementação uma vez que esta tipologia permite aproveitar uma outra ferramenta já utilizada: o Microsoft Word.

### **Responsabilidades e Tarefas**

Sendo o BI Publisher uma solução exclusiva do componente de tangíveis, a sua gestão e manutenção era toda da minha responsabilidade, mesmo que feita noutros âmbitos que não a gestão de tangíveis, estas tinham de ser aprovadas por mim. Esta manutenção passava por tarefas diversas desde a atualização ou alteração dos modelos de dados de suporte à geração do documento até à atualização ou alteração dos *templates* e respetivas parametrizações de geração.

Uma vez que a aplicação *middleware* da GT está agregada ao BI Publisher, a sua gestão e manutenção ficou também ao meu encargo, realizando tarefas de análise, manutenção e desenvolvimento de forma a ir de encontro às necessidades do cliente, quer funcionais, quer em termos de desempenho.

### **4.2.5 Base de Dados Oracle**

A base de dados, composta por vários *schemas*, é um dos pontos centrais de todo o projeto, visto que é nela que é armazenada grande parte da informação essencial ao funcionamento da Unicre.

De forma a manter as alterações feitas aos diversos *schemas* e também para facilitar a uma utilização transversal às equipas, foi criado um repositório SVN com todos os *scripts* de base de dados necessários à criação das mesmas.

Estes *scripts* foram organizados em vários *scripts PL*

*SQL*:

- *Scripts* que definem os cabeçalhos e os corpos dos diferentes *packages*;
- DDL (*Data Definition Language*) que definem a estrutura da base de dados ou do *schema*;
- DML (*Data Manipulation Language*) que contêm instruções de gestão de dados dentro dos *schemas*;
- *Scripts* de parametrização inicial que são utilizados para introduzir configurações essenciais à execução dos processos;
- *Scripts SQLLoader* que permitem a importação de informação, via ficheiro, para

a base de dados. Estes *scripts* encontram-se ainda divididos em controlo e em lógica, tal como será explicado já de seguida;

- *Scripts* de gestão, tais como, desligar todas as sessões ativas no momento do *deploy* ou mesmo a atribuição de permissões de leitura de tabelas por parte de outros *schemas*.

Outra necessidade que surge com a existência de diversos componentes, onde alguns têm modos de operação completamente distintos, é a importação de ficheiros como folhas de cálculo excel ou ficheiros de texto exportados de outros componentes, diretamente para as base de dados. Surgiu então a necessidade de criar *scripts* SQLLoader que lidassem com este mesmo tratamento de dados. Estes *scripts* são compostos por um ficheiro de controlo, um ficheiro de execução e ainda um ficheiro *load\_file*. Os de execução apenas servem para receber os parâmetros dos diversos ambientes bem como executar os ficheiros de controlo que contêm as cláusulas de inserção a serem utilizados para os dados obtidos pelo *load\_file*.

Posteriormente, todos estes dados serão manipulados, transformados e guardados recorrendo aos *packages* PL/SQL destinados a cada processamento para que mais tarde possam ser reutilizados. À semelhança do que acontece nos serviços SOA, o *deploy* dos desenvolvimentos relativos às base de dados são feitos com recurso à ferramenta ANT. Para tal, foram criados *scripts* que efetuam os passos necessários à colocação dos novos desenvolvimentos nos respetivos *schemas*: execução de código DDL e respetivo DML, seguido dos procedimentos PL/SQL e por último a respetiva compilação e recompilação, só para validar que não se verifica um erro por má compilação. Para estes dois últimos e à semelhança do que ocorre com o encerramento das ligações já existentes aos *schemas* antes do processo de *deployment*, foram criados *scripts* genéricos de forma a que pudessem ser transversais a todas as passagens relativas a base de dados Oracle.

Tal como mencionado anteriormente, existem vários componentes e diversas fontes de informação distribuídos por diversas máquinas. Uma vez que existe a necessidade de alguns destes *schemas* partilharem informação entre si, é necessário recorrer a *database links* para estabelecer estas comunicações. Os *DBLinks* consistem em apontadores que definem ligações unidirecionais entre dois servidores de base de dados.

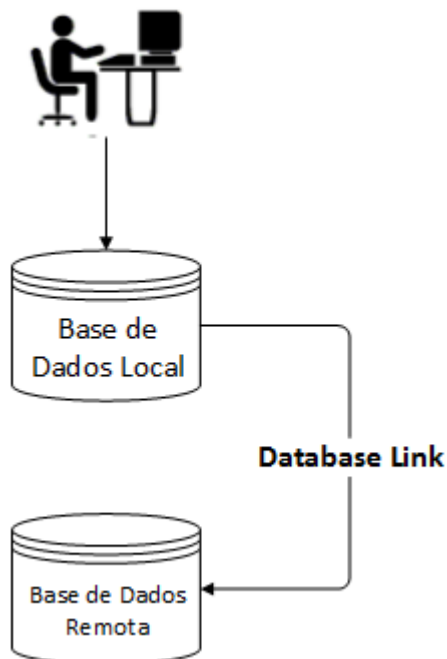


Figura 4.4: Representação de um Database Link

### Responsabilidades e Tarefas

As bases de dados são, destacadamente, o ponto mais central do projeto, todos os componentes do mesmo fazem utilização destas. Como tal, todas as equipas do projeto, há exceção de blocos, têm a responsabilidade de manter as mesmas em perfeito estado de funcionamento, não só de compilação como de desempenho, respeitar as normas de standardização de nomenclaturas e estruturação de código, uma vez que a qualquer momento, qualquer outra equipa pode necessitar de informação proveniente das mesmas.

Tal como nos serviços SOA, é essencial que todas as equipas assumam responsabilidades de comunicação de alterações, uma vez que quando ocorrem estas alterações existe a necessidade de identificar e mitigar riscos que possam vir a aparecer. É ainda importante que as equipas assumam tarefas de monitorização que previnam falhas ou que em caso das mesmas ocorrerem, sejam facilmente recuperadas.

No componente que me era afeto, o Gestor de Tangíveis, tive a responsabilidade de gerir três *schemas* da base de dados de integração. A gestão deste componente obrigava-me a uma constante comunicação com as outras equipas do projeto, uma vez que toda a informação necessária para gerar tangíveis provém de todos os outros componentes.

Devido ao crescimento exponencial destas tabelas, desempenhei ainda diversas tarefas de melhorias de desempenho, tais como a criação de índices, e tarefas de expurgo, de forma a eliminar dados já irrelevantes.

## 4.3 Processo de Desenvolvimento

Ter diversos ambientes distintos é fulcral na execução de um projeto, pois só desta forma é possível criar novos desenvolvimentos sem impactar o que já se encontra em funcionamento real. No projeto MMK existem cinco tipos de ambientes completamente distintos e que serão apresentados por ordem de relevância:

Um ambiente local, que existe um para cada elemento de equipa e que só este tem acesso ao mesmo, serve para se poder fazer mudanças a nível da lógica sem afetar o funcionamento e o trabalho das outras partes envolvidas. Desta forma, caso aconteça alguma desgraça, os danos colaterais são mínimos e facilmente recuperáveis.

Um ambiente de desenvolvimento, que contém todos os desenvolvimentos que já se encontrem estáveis e minimamente testados. Este ambiente serve também de backup, visto que nele estão todos os desenvolvimentos em pleno funcionamento.

Um ambiente de testes integrados, que tal como no ambiente de desenvolvimento, todas as implementações no mesmo existentes, se encontram em pleno funcionamento e duplamente testados, visto que já passaram por mais uma etapa, o ambiente de desenvolvimento.

Por último e antes do ambiente de produção, existe um ambiente denominado de qualidade, onde são efetuados testes de aceitação e funcionais por parte do pessoal da Unicre, tanto do pessoal técnico como do pessoal que tem o conhecimento da lógica de negócio.

Só assim será possível garantir a correta evolução de um pacote até chegar ao seu último fase, a entrada em produção. Estes pacotes - muitos deles *deploy* com recurso à ferramenta ANT - variam o seu conteúdo conforme os componentes a que estão associados, podendo ir desde blocos de código SQL até a modelos de layout RTF.

### 4.3.1 Cadeias

De forma a automatizar e agilizar diversas das tarefas que são executadas diariamente, sejam estas de processamento, de expurgo ou mesmo de geração de documentos,

surgiu a necessidade de recorrer a uma aplicação de automatização de processos, o Control-m [BMC, 2014]. Esta aplicação foi desenvolvida de forma a disponibilizar funcionalidades como o agendamento diário, semanal ou mensal de processos tais como *file watchers* que observam uma determinada pasta e que despoletam uma ou mais ações sempre que lá é colocado um ficheiro.

À sequência de processos executados por esta ferramenta, dá-se o nome de cadeia. Existem diversas cadeias, adaptadas a cada componente, sendo que algumas delas têm dependências com outras, uma vez que há processos que não fazem sentido serem iniciados sem que outros estejam concluídos, como por exemplo, não faz sentido iniciar a geração de tangíveis se não tiver sido terminado o carregamento de dados necessários para os mesmos.

**Responsabilidades e Tarefas** Uma percentagem significativa dos componentes do projeto MMK não necessitam de interação humana para serem realizadas. Desta forma, as equipas afetas a cada um destes componentes ficou responsável por criar e gerir as cadeias que fazem com que o esforço humano na realização destas tarefas, seja praticamente nulo. Esta criação e gestão de cadeias significou uma definição das etapas de cada uma delas e ainda mecanismos intermédios que permitem a interação da aplicação Control-M com certos tipos de desenvolvimentos.

Uma vez que o GT é um dos componentes que quase não necessita de interação humana para que funcione, fiquei encarregue pelas tarefas de acompanhamento de execução de cadeias em ambientes de testes integrados e de produção. Sempre que os desenvolvimentos o obrigavam, efetuei ainda tarefas de automatização ou redefinição de cadeias.

### 4.3.2 Repositório

Tal como em qualquer outro projeto, seja este de qualquer uma das engenharias existentes no mundo, existe a necessidade de todos os seus artefactos de planeamento, gestão ou implementação fiquem salvaguardados de alguma forma. Apesar da sua baixa complexidade, este componente é de uma elevada importância, uma vez que pode ajudar na recuperação de uma falha catastrófica que envolva perda de informação até mesmo a manter um controlo das alterações que foram feitas, o que poderá ser muito útil quando determinado *bug* surge de uma versão estável para uma nova atualização.

Desta forma e de maneira a que todas as alterações fiquem registadas, todos os de-

desenvolvimentos feitos no âmbito do projeto MMK estão salvaguardados num repositório de dados, neste caso, num repositório *subversion* (SVN). Estes desenvolvimentos estão separados por componente, de forma a facilitar a sua utilização e a sua gestão, e também por área de negócio, de forma a facilitar as passagens e os desenvolvimentos, uma vez que as passagens das mesmas foram feitas em alturas diferentes.

**Responsabilidades e Tarefas** O repositório funciona como um armazém de informação, não só a nível de desenvolvimento como a nível de documentação. É da responsabilidade geral que sempre que existam alterações estas fiquem registadas neste mesmo armazém, não só para garantir que em caso de catástrofe existe a possibilidade de recuperar desenvolvimentos, como também para ajudar a perceber como é que determinada alteração afetou determinado processo, sendo isto feito com recurso ao histórico de versões.

## 4.4 Comunicação e Metodologia

Nos desenvolvimentos pré-produção e tal como já foi mencionado no capítulo 4, antes de entrar em produção a maior parte da comunicação entre o cliente e a Deloitte era feita através de uma equipa específica para lidar com este tipo de assuntos.

Por sua vez, os problemas encontrados pelo cliente, são identificados e colocado num portal destinado a este mesmo fim, a abertura e controlo de ocorrências.

No projeto MMK é utilizada a metodologia de desenvolvimento SCRUM. Esta é uma metodologia ágil que pode ser aplicada em diversos tipos de projetos, mas que na maioria dos seus casos, é aplicada em projetos de desenvolvimento de software. Este processo é adequado a projetos que sofram mudanças rápidas ou que tenham requisitos emergentes, pois facilita a organização entre as diversas equipas.

Os desenvolvimentos na SCRUM são feitos através de iterações às quais se dá o nome de *sprints*, sendo que cada um destes é iniciado com uma breve reunião de planeamento e concluído com uma breve reunião de revisão. [SCRUM.org, 2010]

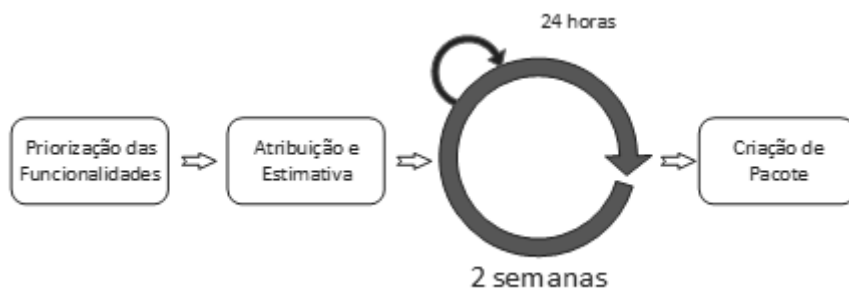


Figura 4.5: Criação de um Produto com Recurso a SCRUM

Uma vez que o projeto MMK tem macro-componentes que por sua vez se subdividem em componentes mais específicos, definiu-se que seriam feitas reuniões diárias entre os elementos das equipas responsáveis pelos macro-componentes e contra o que é mais comum na aplicação desta metodologia, a realização de reuniões semanais com os responsáveis por cada uma das macro-componentes.

Desta forma é possível garantir que todos os elementos estão cientes do que se está a desenvolver e garante-se que todas as equipas de todos os componentes estão alinhadas e a trabalhar para um mesmo fim. Para além de uma melhor organização, o SCRUM facilita ainda a comunicação bem como o planeamento, uma vez que é possível, com base do que é discutido nas reuniões, alocar equipas ou elementos a determinado desenvolvimento que interfira com diversos componentes.

#### 4.4.1 Testes

Os testes são sempre um dos procedimentos fulcrais no sucesso de um projeto. Na Unicre existem diversas equipas, sendo algumas delas apenas funcionais e outras mais dotadas tecnicamente.

Como é lógico, o nosso maior envolvimento e colaboração recai sobre a equipa da Direção de Sistemas de Informação (DSI), que mesmo sendo a equipa mais técnica do cliente, são quem melhor consegue traduzir os aspetos funcionais em aspetos técnicos e vice versa. Esta equipa está responsável por funcionar como primeira linha das restantes equipas do cliente, sendo que quando existe um problema relativo aos desenvolvimentos é a estes que eles devem recorrer. Graças aos componentes nos quais fui integrado, estive também em contacto com as equipas de Gestão de Transações Reclamadas (GTR), cujo nome é bastante intuitivo, uma vez que é a equipa responsável por gerir as reclamações relativas às transações que são feitas pelos comerciantes, com a

equipa da RedeUnicre, cujo propósito é gerir a área de faturação relativa ao negócio de *Acquiring* e por último a Direção Financeira e de Meios (DFM), que esteve envolvida na validação dos desenvolvimentos da certificação de faturação.

As sessões de testes servem, não só, para o cliente ter um primeiro contacto com os novos desenvolvimentos mas também para validar e reparar a lógica de negócio associada aos mesmos. Por norma, estas sessões são feitas diretamente com as pessoas de negócio, sendo apenas necessária a presença de elementos do projeto MMK.

## SOAP UI

De forma a testar os desenvolvimentos relacionados com serviços, mais especificamente com os serviços SOA, deu-se a necessidade de arranjar uma ferramenta que facilitasse e que de certa forma, automatizasse estes testes. A opção foi o SOAP UI que é uma ferramenta multi-plataforma e é utilizada para criar e executar de forma rápida e simples conjuntos de testes a *webservices*. A estrutura (anexo A.2) destes testes pode ser dividida em diversos níveis:

- Os passos (*TestSteps*) são o esqueleto dos testes, uma que são eles que definem o que irá ser testado. Existem diversos testes que podem ser efetuados mas os mais utilizados neste âmbito são o **envio de um pedido SOAP**, que pode ser enviado em HTTP, HTTPS ou JMS e a obtenção de uma resposta, no caso de haver uma, seja o serviço síncrono ou assíncrono e as propriedades, que podem ser utilizadas para receber ou transferir valores entre as várias invocações;
- Um caso de teste (*TestCase*) é uma coleção que contém vários passos que são configurados de forma a validarem uma determinada funcionalidade de um determinado serviço. A ordem de visualização destes passos deve ser sempre tida em conta uma vez que é ela que define a sequência por que estes serão executados;
- No nível superior, encontram-se a coleção de testes (*TestSuites*), cuja sua função é agrupar os testes funcionais em unidades lógicas.

Com esta ferramenta é fácil criar e executar testes automatizados, facilitando assim o teste exaustivo dos serviços desenvolvidos, uma vez que ao mesmo tempo temos a possibilidade de efetuar testes de carga como poupamos tempo por não termos de estar sempre a criar as invocações manualmente.

## 4.5 Arquitetura

A Unicre definiu uma Arquitetura de Referência para os seus Sistemas de Informação, com o objetivo de uniformizar e padronizar conceitos, de disciplinar o processo de desenvolvimento e de estabelecer mecanismos de controlo de qualidade e de reduções de esforços de desenvolvimento. Com base no levantamento de todas as aplicações Core e Satélites e de acordo com os princípios de desenho propostos, foram definidas diversas camadas de integração cuja infraestrutura se pretende autónoma e independente do Sistema Core. Esta arquitetura (ver anexo A.1) está estruturada em:

- *Front-Ends* onde estão todos os mecanismos que garantem a integração e contextualização dos diferentes *front-ends*, bem como a agilização de processos com recurso a mecanismos de automatização;
- Camada de canais onde se encontram as funcionalidades relativas à gestão de comunicações entre processadores e o novo sistema *core*. Esta camada inclui os serviços e os ecrãs de suporte aos diferentes *websites* pretendidos pelo cliente;
- Sistema de Suporte são todas as aplicações que dão suporte ao projeto, seja para garantir a persistência da informação, como são os casos dos repositórios de *subversion*, seja para suportar outras aplicações já anteriores ao projeto MMK, como é o caso da gestão documental;
- Sistemas Operacionais é a camada onde estão contidos os sistemas *core* do cliente, tanto o sistema pré-MMK como o novo sistema, que embora não faça parte do projeto MMK, foi implementado em paralelo com o mesmo;
- Sistemas de Apoio à Decisão
- Entidades externas onde são criadas as interfaces para comunicação com entidades externas ao projeto MMK;
- Segurança lógica na qual encontramos toda a lógica relativa a autenticação, controlo de acessos e perfis de utilizadores.
- Camada de Integração encontram-se os componentes que efetuam o maior processamento e tratamento de informação, obtendo-a, transformando-a e disponibilizando-a de acordo com as necessidades dos diferentes componentes. De acordo com as especificidades do negócio, os componentes podem ser síncronos ou assíncronos.

## 4.5.1 Arquitetura Orientada a Serviços e Eventos

Uma arquitetura orientada a serviços é um conjunto de princípios e metodologias para desenhar e desenvolver serviços de *software* que possam ser distribuídos e geridos através da rede da organização. A reutilização destes componentes é relativamente simples uma vez que podem ser complementemente independentes uns dos outros, podem ser modificados sem afetar qualquer outro dos componentes circundantes. Graças a esta independência e modularidade, estes componentes de *software* podem ser utilizados ou reaproveitados para desenvolver uma imensa variedade de novas aplicações, podendo mesmo ser disponibilizados a parceiros e fornecedores que pretendam integrar as suas aplicações no negócio.

A **Arquitetura Orientada a Serviços e Eventos (AOSE)** assenta nos princípios e metodologias SOA, sendo construída com base numa coleção de serviços e eventos que operam entre si através de interfaces bem definidas. Esta arquitetura caracteriza-se por capacidades como a utilização de serviços que podem ser descobertos tanto em tempo de desenho ou em execução por outros serviços, não só pelo seu identificador unívoco como também pela identidade da interface que o expõe.

Um serviço é um grupo de componentes de *software* que suporta um determinado processo de negócio, na concretização de um objetivo em particular, permitindo a comunicação entre estes componentes e os utilizadores finais.

Na implementação da arquitetura do projeto MMK, os serviços são implementados através de três formas distintas:

1. *Request-Reply* via *Web Services* utilizando o protocolo *Simple Object Access Protocol* (SOAP) e *Web Services Description Language* (WSDL);
2. Implementação de soluções de integração orientadas a mensagens/eventos (*Message Oriented Middleware*);
3. Sistemas de publicação/subscrição como é o caso do *Java Messaging Service* (JMS).

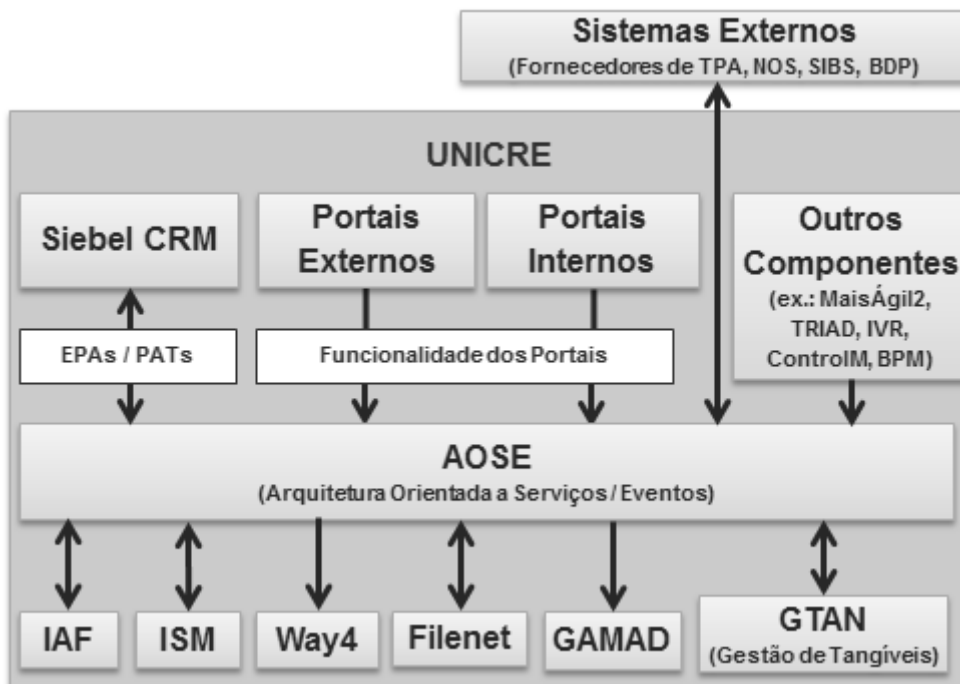


Figura 4.6: AOSE do Projeto MMK

Esta camada é um *middleware* assente no produto Oracle SOA Suite 11g e constitui uma componente crítica na arquitetura de referência. A AOSE possibilita a integração dos vários sistemas através da disponibilização das diversas funcionalidades existentes em módulos interoperáveis, reutilizáveis e partilhados transversalmente.

## 4.6 Gestor de Tangíveis

O Gestor de Tangíveis (GT) é um componente que, tal como o próprio nome sugere, está encarregue de gerar todo o tipo de documentos que sejam necessários para o negócio. Além da geração, este componente tem ainda a capacidade de tratar de diversos tipos de expedição dos mesmos, tal como irá ser abordado mais à frente.

Fazendo uso de diversas tecnologias, o GT irá receber eventos provenientes de qualquer outro componente presente na arquitetura MMK para gerar e/ou expedir documentos para um ou mais destinos. As componentes da arquitetura concetual, que fazem parte deste componente, são:

- Um controlador de eventos da AOSE que terá como funções disponibilizar uma estrutura para configurar os parâmetros pré-definidos para os eventos da aplicação

de Gestão de Tangíveis, receber os pedidos da camada de integração, aceder a serviços de extração de dados e ainda, reencaminhar o pedido e o resultado destas mesmas extrações para a aplicação Oracle BI Publisher;

- O ODS, no qual será obtida a informação massiva relativa à geração de documentos;
- Aplicação de Suporte e Integrador Assíncrono de Ficheiros que são responsáveis por gerar os eventos de produção de tangíveis.

Antes de se abordarem os tangíveis propriamente ditos, é importante reter três conceitos essenciais: modelo, documento final e expedição.

O modelo é um documento, no formato *Rich Text-Format* (RTF), composto por vários elementos fixos e/ou variáveis, que irá ser a base de todas as cartas. No modelo serão definidas as regras de carregamento e seleção da informação.

Desta forma garante-se que não existe uma relação de um-para-um entre os modelos e os documentos gerados, havendo desta forma um maior dinamismo (podendo haver um único modelo para diversas cartas), uma maior qualidade e ainda uma maior simplicidade no processo de geração de cartas.

O documento final é o resultado da junção da informação existente nas fontes de dados com os modelos das cartas. Esta junção será conseguida com recurso a uma ferramenta Oracle, o Business Intelligence Publisher.

Por último, a expedição, que contém é um conjunto de processos que garantem que estes mesmos documentos cheguem aos seus destinos.

#### **4.6.1 Ciclo de Vida de um Documento**

O documento tem associado a si um ciclo de vida, sendo este iniciado com o despoletamento de um evento por parte de um sistema origem até à sua expurgar (quando aplicável).

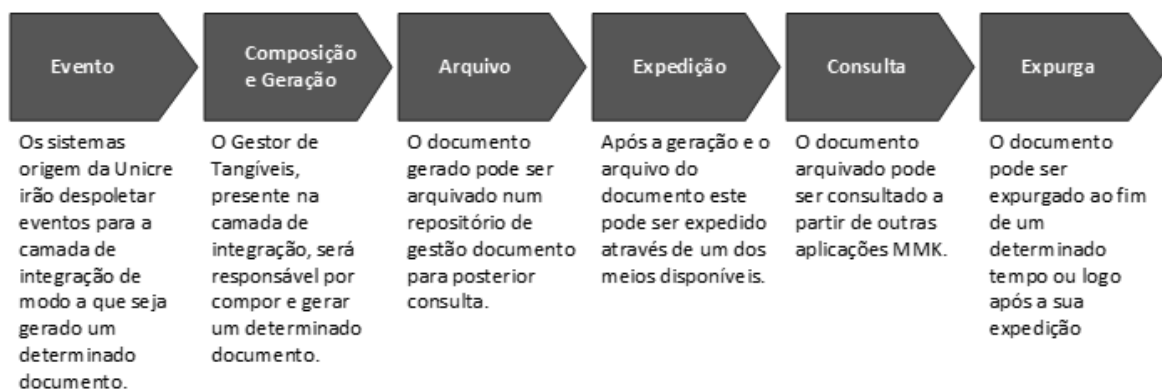


Figura 4.7: Ciclo de vida de um documento

1. Os **eventos** podem ser provenientes de vários componentes integrantes do projeto. Por sua vez, estes serão atendidos pela AOSE que por consequentemente transmitirá, ao componente de gestão de tangíveis, a informação necessária para gerar o documento. Pode ainda dar-se o caso onde o evento é também complementado com o envio de um documento que irá ser utilizado para a geração de outros documentos.

Os dados que alimentam estes modelos RTF, poderão ser obtidos via XML, que são construídos com informações provenientes de esquemas da base de dados, sejam eles centrais, tal como é o caso do ODS, ou esquemas de apoio à geração dos documentos por parte da GT.

2. A **composição** está encarregue de ler, através dos modelos de dados associados ao BI Publisher, e mapear estes mesmos dados para os respetivos campos, enquanto que a **geração** está encarregue de gerar o PDF final bem como aplicar-lhe os pré-impessos (se aplicável). Na eventualidade de ocorrerem erros derivados à má formatação destes dados, a GT irá alertar a origem desta mesma situação. Por sua vez, se o problema não for derivado de terceiros, o componente é suficientemente inteligente e automatizado para desencadear um processo que resolva ou informe acerca do mesmo.
3. Após a geração no Gestor de Tangíveis, o próximo passo é o **arquivo**. O ficheiro PDF gerado é arquivado na ferramenta de gestão documental (FileNet), caso as suas parametrizações assim o indiquem.

Depois do Gestor de Tangíveis ter enviado o documento no seu estado final (ficheiro \*.pdf) para o FileNet, devendo este notificar o GT com o retorno do resul-

tado do arquivo do documento, o GUID <sup>9</sup> do documento, bem como a respetiva pasta de processo.

Este retorno devolvido ao GT deve ser registado internamente para posterior comunicação aos sistemas origem que pretendam receber esta notificação.

4. A expedição dos documentos representa todos os canais para ou por onde os documentos poderão ser enviados. Este assunto será abordado com maior detalhe na secção 4.6.1.
5. A **consulta** permite ao utilizador aceder e visualizar os documentos gerados e arquivados através de três aplicações distintas: Siebel, sites e da ferramenta de gestão documental, Filenet.
6. A *expurga* de documentos consiste na eliminação de documentos que já não sejam mais necessários ao Gestor de Tangíveis. Existem dois processos distintos para proceder a esta expurga:
  - Expurga de documentos logo após a sua geração e respetivo envio para expedição. Neste processo são incluídos os documentos não arquivados no repositório de gestão documental. Por exemplo, as cartas enviadas aos utilizadores com as passwords de acesso aos canais de *Homebanking*;
  - Expurga de documentos arquivados no repositório de gestão documental. Este processo é feito com recurso aos mecanismos de expurga da própria Unicre.

Em termos de carregamento de dados para geração do documento, existem três elementos que são cruciais:

- A base de dados, que contém as tabelas com os dados de suporte à geração de cada tipo de tangível. Estas tabelas podem ser carregadas de três formas distintas: via SQLLoader, onde são processados ficheiros exportados do sistema central (Way4), via *webservice*, onde o componente que invoca o gestor de tangíveis transmite também a informação necessária à geração dos mesmos ou ainda via fluxos automatizados pelo control-m;
- O modelo de dados, que irá carregar a informação das tabelas existentes na base de dados para um documento XML, com um formato variável consoante o tipo de tangível em questão;
- O modelo associado ao tangível, no qual serão colocados os dados obtidos e transformados pelo modelo de dados.

---

<sup>9</sup>GUID - Identificador Único Global é um identificador que será único em qualquer que seja o contextos

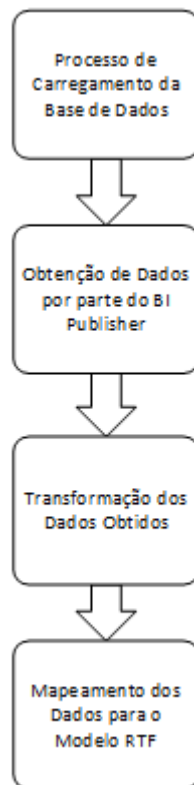


Figura 4.8: Ciclo de geração de um documento

Durante o processamento e carregamento dos dados para as tabelas de suporte à geração de tangíveis, também designadas por Repositório Temporário de Dados (RTD), a informação vai sendo igualmente distribuída pelas tabelas da Unidade de Processamento e Controlo de Produção (UPCP), onde fica armazenada a informação essencial à expedição dos tangíveis. Esta informação percorre um ciclo, que poderá ser variável de acordo com os meios de expedição especificados. Este ciclo começa com a tabela *documentos\_dados* onde são armazenados alguns dados dos clientes, das suas contas, cartões, tal como será explicado posteriormente na secção 4.6.1. Complementarmente, é adicionada informação na tabela *documentos\_finais*, que tal como o próprio nome indica, tem informação básica mas essencial acerca dos tangíveis gerados, como a sua localização e a sua tipologia. Estas duas tabelas fornecem toda a informação necessária para alimentar os diversos processos de expedição, onde cada expedição de um único tangível é designada de comunicação.

## Critérios de Expedição

Várias comunicações fazem parte de uma tarefa que por sua vez fazem parte de um trabalho. Um trabalho representa um processamento de uma determinada tipologia de tangível, podendo resultar na geração de um ou mais documentos, enquanto que por sua vez, a tarefa representa todos os passos necessários à geração de um documento:

1. A primeira etapa é o carregamento da informação, que com recurso aos procedimentos criados no *schema* RTD, tratam e colocam a informação necessária à composição dos documentos nas respetivas tabelas;
2. A tarefa de composição consiste na criação de um pedido de geração de documentos ao BI Publisher. Tal como mencionado na secção 4.6.1, recorrendo a três elementos, o BIP constrói um XML e aplica-o sobre um modelo RTF, criando desta forma um ou mais tangíveis;
3. A justaposição consiste na aplicação de um pré-impresso, que varia consoante a área de negócio e o departamento responsável, sobre o documento gerado pela tarefa de composição;
4. A tarefa de duplicação consiste na atualização do campos *Original* para *Duplicado* ou *Cópia do Documento Original*, dependendo do tangível;
5. As tarefas de expedição são todos os canais por onde os quais os tangíveis podem ser enviados para o seu destino.

Com apoio do *middleware* JAVA acoplado ao BIP, o GT tem a capacidade de expedir tangíveis para seis destinos diferentes:

- A expedição para **arquivo**, que é totalmente independente dos outros cinco canais, consiste no envio de documentos para uma aplicação externa, o Filenet. O Filenet é uma plataforma de gestão documental [IBM, 2014], cujo principal fim no projeto MMK é o armazenamento e posterior disponibilização de documentos. Para a sua posterior obtenção, é necessário que no processo de expedição seja construído um ficheiro de metadados com informação referente ao documento em questão;
- O GT tem também a capacidade de disponibilizar tangíveis para **entidades** externas, como entidades bancárias, através da expedição para outras entidades que utiliza o protocolo FTP para expedir estes os documentos para outras entidades;
- A **impressão local** consiste no envio de documentos para sharepoint. Estes documentos poderão, posteriormente, ser impressos ou consultados nos componentes de canais;
- Na expedição por **email**, os tangíveis são enviados em anexo a uma mensagem

de correio eletrónico. Todos os componentes da mensagem, tais como o assunto, o remetente e até mesmo o seu corpo, são parametrizáveis para cada tipologia de tangível;

- A expedição por **email certificado** segue a mesma lógica da expedição por email mas com uma pequena nuance: toda a mensagem é assinada antes de ser enviada.

Estes critérios de expedição respeitam regras de prioridade e de exclusividade, tal como é apresentado nos anexos A.5 e A.6.

Todos estes meios de expedição têm ainda duas maneiras de serem processados: o modo online, que está presente em todos os processamentos que este componente pode efetuar, consiste na execução das operações logo quando a geração do documento e o modo massivo, por sua vez, só é invocado pelo control-M e corre periodicamente, podendo este período ser diário, semanal, quinzenal ou mesmo mensal.

Alguns tangíveis, devido ao seu elevado volume, são classificadas como documentos massivos. Uma vez que a informação utilizada por estes tipos de documentos é de uma dimensão muito superior quando comparado com outros, estes necessitam que seja efetuado um pré-processamento de dados antes de proceder ao normal carregamento e conseqüente geração.

### **Parametrização de Documentos**

Devido ao vasto universo de documentos a ser gerado pelo Gestor de Tangíveis, surgiu a necessidade de parametrizar estes mesmos modelos, de forma a adequá-los às necessidades do cliente sem ser necessário proceder a desenvolvimentos adicionais.

Estes parâmetros passam pelos mais variadas configurações, sejam estas relativos ao documento final, tal como o código do envelope a ser utilizado ou a assinatura do responsável a incluir no documento, ou relativos aos parâmetros de expedição, que indicam para onde e como vai ser enviado o documento.

## **4.6.2 BI Publisher no Gestor de Tangíveis**

Tal como explicado anteriormente, O BIP que permite a geração de documentos. O processo de geração de um documento é, em termos de visibilidade para o utilizador (não final) bastante simples, visto que apenas necessita de invocar um serviço SOA que tratará de obter a informação necessária para que o BIP consiga ir buscar a informa-

ção às base de dados de suporte à geração de tangíveis, transforma-la e envia-la para o orquestrador. Aqui é onde realmente a geração acontece, utilizando a informação recebida, o BIP consegue consultar, com recurso aos modelos de dados, os dados necessários à criação dos documentos finais. Por sua vez, estes dados que serão obtidos e transformados em XML, serão diretamente mapeados para os modelos RTF, dando origem aos PDF's finais que serão devidamente expedidos posteriormente. De notar que ao invés de a invocação ser invocada manualmente pelo utilizador da DSI, esta pode ser despoletada por outros componentes ou mesmo pela aplicação de automatização de processos, o Control-M.

Não sendo uma ferramenta pura de *reporting*, o BI Publisher tem algumas limitações. Como tal, foi necessário desenvolver um conjunto de serviços à medida e que respondesse às necessidades do projeto MMK.

## **Middleware GT**

Este *middleware* consiste numa aplicação desenvolvida em JAVA e que se encontra acoplada ao BI Publisher e a sua principal função é colmatar as falhas existentes neste mesmo, ou seja, funcionalidades que não são suportadas nativamente ou flexíveis ao ponto desejado.

O *middleware* pode ser separado em duas componentes, que são interligados por filas de espera tal como será descrito de seguida, e que são:

A componente de lógica, que está encarregue de disponibilizar as operações que o módulo disponibiliza. É aqui que são criadas as mensagens com a informação necessária à sua execução, tais como, a sub-operação a ser executada, uma vez que uma operação pode conter várias destas (exemplo: uma conversão pode ser feita para diferentes formatos). Posteriormente, esta componente irá disponibilizar estas mensagens nas respetivas filas de espera.

A componente de processamento está encarregue de desencadear as ações em si. Esta componente encontra-se sempre à escuta nas diferentes filas de espera, de forma a que quando é colocado lá um novo pedido, este seja instantaneamente processado. As principais operações disponibilizadas por este componente são: o envio, conversão e justaposição de documentos.

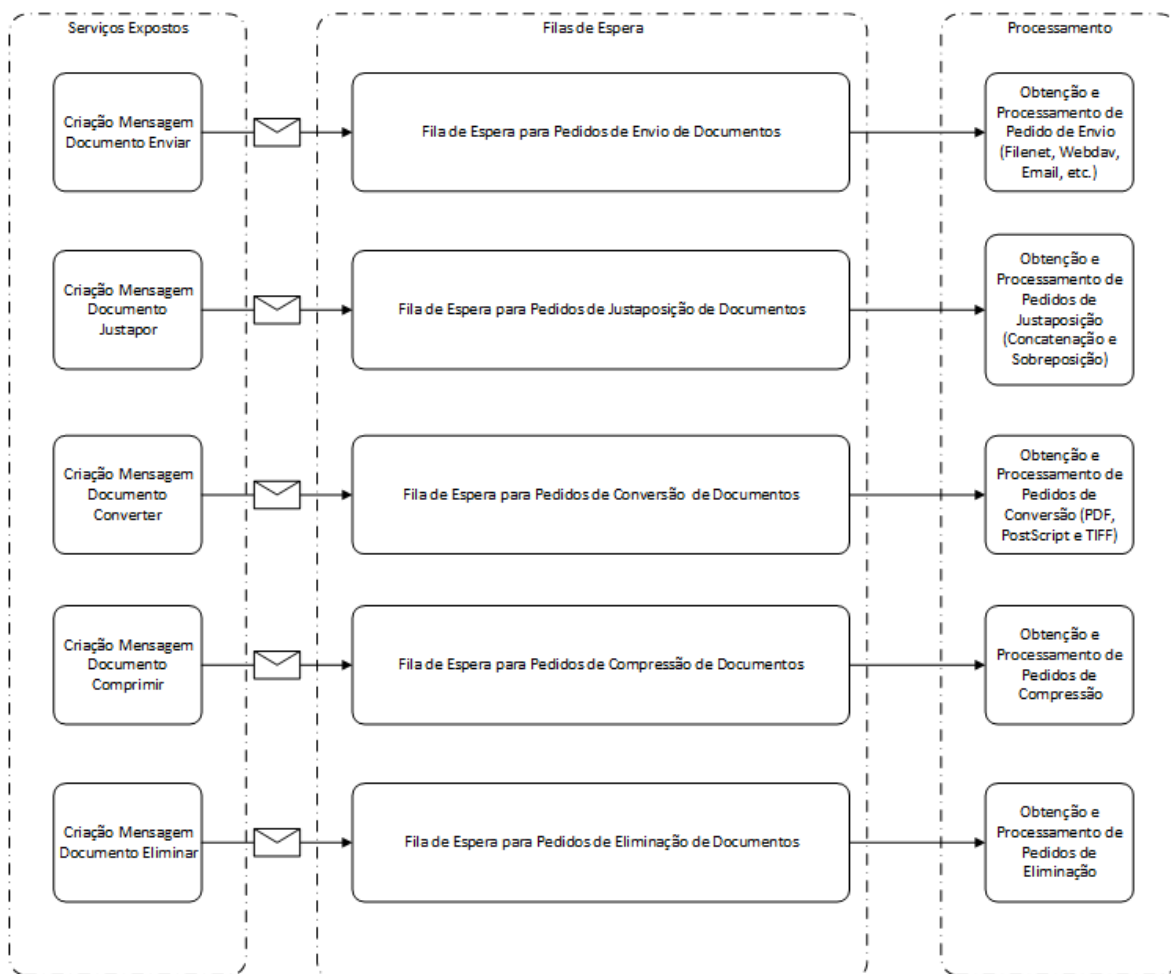


Figura 4.9: Middleware agregado ao BI Publisher

As principais funcionalidades que só são conseguidas graças a este módulo são:

- Justaposição de documentos, onde são aplicados os pré-impresos da Unicre sobre os documentos gerados pelo Gestor de Tangíveis;
- Duplicação de documentos, onde além do documento original ser duplicado, ainda é substituída a frase estática relativa ao documento original por uma frase estática relativa ao duplicado;
- Conversão de documentos provenientes de componentes externos, tanto de documentos em formato word como documentos digitalizados em formato TIFF ou PNG, para PDF;
- Expurga: a expurga consiste na eliminação de documentos que já se tenham tornado obsoletos ou que não sejam mais necessários;
- Envio de Emails que consiste na construção da mensagem de correio eletrónico

com base nos dados recebidos e envio de documentos sem certificação, via correio eletrónico.

- Envio de Emails Certificados que à semelhança do envio de emails, constrói e envia mensagens de correio eletrónico, mas com anexos certificados com base nas chaves privadas da própria Unicre;
- Arquivo que tal como o próprio nome indica, consiste em arquivar os documentos em repositórios específicos para este mesmo fim;
- Impressão consiste na expedição das cartas para filas de impressão. Este modo é o único que não tem um modo online ou massivo mas, que por sua vez, tem um modo local e um modo entidade. O modo local, tal como o próprio nome indica, consiste na impressão local, ou seja, dentro da própria Unicre, de documentos gerados. O modo entidade, por sua vez, consiste no envio de grandes quantidades de cartas, para que uma entidade externa efetue a sua impressão e respetivo despacho.

## 4.7 Manutenção de Tangíveis

Os tangíveis são uma das peças vitais deste componente, de outra forma nem faria sentido existir este um componente de gestão de tangíveis.

Sendo os tangíveis um ponto de comunicação com o utilizador final, estes estão em constante mudança/melhoria, de forma a refletir as alterações à lógica de negócio ou até mesmo à legislação portuguesa. Estas constantes alterações, nomeadamente na empresa que fornece os serviços de impressão e de *finishing* à Unicre, resultou na necessidade de serem adequadas as formatações dos campos de endereçamento de forma a que se ajustassem à janela dos envelopes. A reestruturação, não só das formatações mas também das margens dos documentos fez com que o pré-impresso não escondesse os campos de informação relativa ao código do modelo. Esta situação fez com que partisse à descoberta do que está por detrás do processo de duplicação de documentos, que tal como mencionado anteriormente é feita no *middleware* agregado ao BI Publisher. O meu primeiro pensamento foi que o problema estivesse na classe de justaposição do *middleware* JAVA, que esta tivesse o posicionamento do texto mal definido, uma vez que havia sido mudado de sitio, o que resultava numa má justaposição. Após análise deste processo, concluí que afinal não era o *middleware* que colocava o

texto diretamente no tangível, mas sim que existiam documentos estáticos que eram sobrepostos de acordo com a tipologia passada. Esta alteração passou então por refletir as novas formatações dos campos de endereçamento nestes mesmos documentos estáticos.

Outras alterações envolveram melhoria de desempenho ou inclusão de campos condicionais, tal como será descrito ao longo deste mesmo documento.

## **4.8 Manutenção dos Processos de Expedição**

Os processos de expedição são um dos pontos fulcrais do projeto MMK, uma vez que é através deles que se entregam muitos dos dados ao utilizador final, seja através de documento ou através de uma mensagem de correio eletrónico. Com a evolução e a inovação dos sistemas, não só da Unicre mas como também das entidades externas, foram solicitadas algumas alterações ou melhorias aos processos de expedição já existentes.

### **4.8.1 Expedição de Tangíveis**

Tal como mencionado num capítulo anterior, existem diversas cadeias no projeto MMK que visam automatizar e facilitar processos, nomeadamente processos relacionados com a geração e expedição de tangíveis. É claro que toda esta automatização só é possível quando os processos atingem um determinado grau de maturidade, o que representa um problema visto que as cadeias, mais especificamente as cadeias de expedição, são genéricas e não específicas para cada tipologia ou mesmo para cada documento. Identificada a limitação destes processos de expedição, concluiu-se que seria necessária uma alteração transversal, uma alteração que refletisse o que já existia no processo de carregamento de dados, a especificação do identificador para o qual se pretende executar o expedição.

Os serviços SOA são expostos através de uma interface, sendo que esta interface é definida com recurso a um ficheiro XSD. No âmbito do GT, além dos ficheiros que definem a interface, existe ainda um ficheiro transversal que possui campos comuns entre diversas interfaces, como por exemplo os campos de expedição. De forma a que fosse possível selecionar apenas um determinado identificador de documentos para ser expedido, foi necessário atualizar este XSD transversal adicionando um novo elemento, o IDTangível.

Após a alteração da interface do serviço, houve a necessidade de mapear o novo campo e refleti-lo nos procedimentos de carregamento de dados. Estes mapeamentos são garantidos com recurso aos *assigns* e também às transformações, que após terem sido feitas as alterações nos procedimentos da base de dados, apenas tiveram de ser atualizados para refletir o novo parâmetro de entrada. Esta operação foi repetida para cada um dos seis métodos de expedição existentes.

### **Expedição por Email**

Nos dias de hoje são cada vez menos as pessoas que ainda pedem/pretendem receber o seu extrato em papel, muito graças à evolução e simplificação da utilização e acessibilidade às novas tecnologias.

A Unicre possui um vasto lote de clientes, clientes estes que já aderiram maioritariamente aos extratos eletrónicos. Após a entrada em produção, constatou-se que alguns dos clientes não estavam a receber corretamente as mensagens de correio eletrónico, pois recebiam a mensagem em si mas sem o devido anexo.

Após uma análise exaustiva, concluiu-se que este cenário se verificava apenas com clientes que possuíam contas de correio eletrónico em certos servidores de email. Uma vez que esta situação representava uma alteração na forma como estas mensagens são construídas, desencadeou-se uma revolução em todo o processo de construção das mesmas.

### **Construção de uma Mensagem de Correio Eletrónico**

Finda a primeira execução da geração de tangíveis em ambiente de produção, foram recebidas queixas de diversos clientes que haviam recebido o email sem o respetivo anexo.

As mensagens de correio eletrónico são constituídas por duas grandes secções: um cabeçalho, onde vai definida toda a informação relativa à mensagem e um corpo, que contém o conteúdo propriamente dito.

Uma vez que nem todos os servidores de correio eletrónico estavam a lidar bem com a estrutura da mensagem usada na primeira abordagem, mais concretamente com o tipo *Multi-Purpose Internet Mail Extensions*<sup>10</sup> definido: *Multipart/Mixed* onde os conteúdos são apresentados pela ordem definida na criação da mensagem, sendo as partes

---

<sup>10</sup>O MIME é uma extensão ao protocolo de email SMTP que permite trocar diferentes tipos de dados na Internet

completamente independentes umas das outras. Nesta abordagem todos os conteúdos da mensagem, como os anexos e o conteúdo do email, estavam definidos em conjunto dentro da mesma mensagem e conseqüentemente do mesmo tipo de dados.

Foi então necessário proceder à reestruturação de toda a mensagem, começando pela nova definição do seu conteúdo, desta vez como *Multipart/Alternative* e onde foi dividido o corpo da mensagem em duas partes distintas: conteúdo e anexos. Nesta nova definição, o conteúdo é enviado de forma organizada, tão organizada que o cliente é que decide o que prefere apresentar.

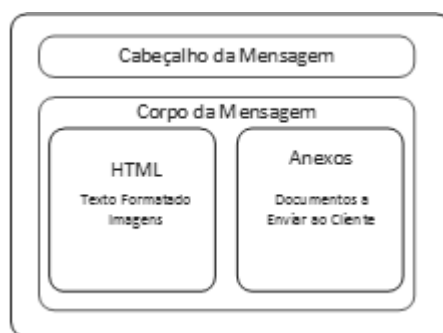


Figura 4.10: Estrutura da Mensagem de Correio Eletrônico

Para tirar partido de todo o potencial das tecnologias relativas às mensagens de correio eletrônico, resolveu-se também atualizar o processo de forma a suportar formatação, ou seja, o conteúdo das mensagens passou a ser código HTML em vez de texto simples. Com isto foi não só possível responder às exigências feitas pela equipa de Marketing relativamente ao *brand* da Unicre como também garantir que as mensagens são apresentadas de forma igual para todos os clientes. Com a introdução das mensagens no formato HTML, foi também possível formatar as mesmas na própria parametrização do documento existente na base de dados.

Fugindo um pouco da estrutura da mensagem e à semelhança do que já existe para outros meios de expedição, resolveu-se concatenar os documentos relativos a um pedido e enviar num único anexo, uma vez que não faz sentido enviar todas as partes constituintes do mesmo de forma individual.

De forma a monitorizar ainda o correto e completo envio das mensagens de correio eletrônico, acordou-se a implementação do envio de emails com BCC para uma conta de email associada à Unicre, parametrizada por parte dos elementos da DSI.

Esta funcionalidade envolveu alterações ao longo de toda a lógica associada ao envio de mensagens, ou seja, desde o carregamento de dados pelos procedimentos existentes

na base de dados até ao processamento de dados efetuados pelo SOA. Começando pela origem dos dados, foi necessário não só adicionar uma nova coluna nas parametrizações do documento, uma vez que cada tipo de tangível fica associado a uma determinado área de negócio e cada área de negócio tem o seu email de reclamações. Esta informação teria de ser obrigatoriamente replicada por todo o processo, adicionando novas colunas e refletindo as mesmas nos tipos da base de dados, até chegar às tabelas de comunicações, que tal como mencionado anteriormente, são estas que são utilizadas pelo *middleware* responsável pela expedição. Como é lógico, quem serve de ponte entre os dois é o SOA, havendo necessidade de proceder também a alterações no mesmo: foi necessário atualizar os adaptadores de forma a que estes atualizassem os XSD *schemas* que estão associados à invocação dos procedimentos PL/SQL. Desta forma foi então possível guardar e encaminhar a nova informação através do fluxo até à chamada do *middleware* para a sua posterior expedição (Anexo: A.3).

Desta forma apenas ficava em falta a adequação do *middleware*. Como este componente possui diversas interfaces, não só para a expedição mas como para outras operações que não o BI Publisher não suporte, foi necessário atualizar as interfaces relativas às expedições por email e por email certificado. Uma vez que estes serviços são implementados com recurso à linguagem JAVA ao invés de serem implementados no SOA, não foi atualizado o XSD *schema* mas sim as classes e respetivos objetos que definem estas mesmas interfaces, adicionando para tal uma nova propriedade e os seus respetivos *gets* e *sets*. Esta alteração teve de ser também refletida também na estrutura da mensagem que iria ser construída e colocada na fila de espera bem como no módulo JMSBean que esta responsável por observar a fila de espera e processar os pedidos, neste caso específico, ler a informação contida nesta mensagem, construir o email, no caso do envio certificado encripta também a mensagem, e por último a expedição da mesma.

### **Emails Dinâmicos**

Devido à natureza variável dos emails, houve a necessidade de criar consultas dinâmicas que permitissem adequar os conteúdos dos mesmos por forma a serem o mais dinâmicos possível. Estas consultas permitem ao utilizador, parametrizar quais campos que terão informação dinâmica nas mensagens de correio eletrónico. Na altura do envio das mensagens, todos os campos definidos na mensagem, que são identificados por parênteses retos e cujo nome do campo é igual à coluna onde será consultado, serão substituídos pelos respetivos valores.

## Expedição para Impressão Local

A impressão local é um dos seis métodos de expedição que a Gestão de Tangíveis disponibiliza, que tal como mencionado anteriormente, consiste no envio de documentos para listas sharepoint, para que mais tarde possam ser consultados e impressos, pelos utilizadores, através de portais.

Após ter ficado definido em desenho funcional que os documentos iriam ser disponibilizados de três formas: online, massivo agregado e massivo individual e assim ter sido implementado, surgiu a necessidade de agregar também os documentos enviados através do modo online. Inicialmente esta parecia uma alteração bastante simples, uma vez que seria apenas necessário invocar o serviço de concatenação à lógica já existente em SOA. Porém, isto não se verificou uma vez que ao efetuar a concatenação, o pré-impresso aplicado nos documentos gerados, estava a ser removido pela aplicação externa que está responsável pela concatenação. Este revelou-se um problema algo engenhoso, uma vez que a perda de pré-impresso podia ser derivada de diversos aspetos, desde o documento em si até à própria aplicação responsável pela concatenação.

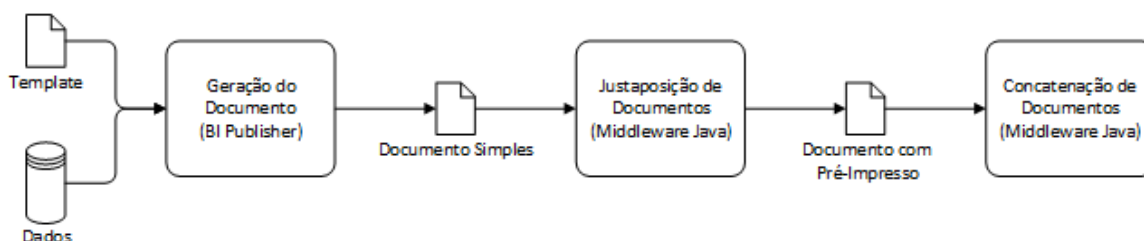


Figura 4.11: Ciclo de um pedido de adesão DE

Verificou-se que o problema era no processo de justaposição do PDF gerado pelo BI Publisher com o documento estático que inclui o pré-impresso, a biblioteca Java aplicava uma nova "camada" no PDF gerado que era ignorada pela aplicação externa durante o processo de concatenação. Após a deteção do problema, a resolução foi bastante simples, uma vez que bastou migrar a justaposição da biblioteca Java para a aplicação externa responsável pela concatenação, garantindo-se desta forma que todos os processos eram compatíveis entre si.

## Grupos para Impressão Local

Após os documentos serem colocados nas listas sharepoint, irão ser consultados atra-

vés de portais disponibilizados aos utilizadores. Visto que estas listas são genéricas, ou seja, não existe uma lista para cada utilizador, existe a necessidade de associar determinado documento a determinado utilizador. Para tal, é necessário que o processo de expedição para impressão local que envie, para além do documento a ser consultado, informação relativa ao mesmo e ao utilizador ao qual este se destina. Esta informação é enviada em formato de metadados, ou seja, é enviado um documento XML construído em tempo de execução pelo fluxo SOA, que contém o nome do documento e para quem se destina, bem como uma descrição para ser apresentada no portal.

Tal como descrito, esta solução disponibiliza um documento para um determinado utilizador. Esta funcionalidade torna-se inviável quando o utilizador a quem se destina o documento não pode estar presente por qualquer razão. Desta forma foi necessário criar uma solução que contornasse esta falha: alterar o destinatário do documento de um utilizador para um grupo de utilizadores.

Esta alteração solicitou basta sincronismo com a equipa de canais, visto que esta é a responsável pela tecnologia sharepoint e respetivos desenvolvimentos, de forma a que ambas as alterações fossem corretamente integradas e evoluídas ao mesmo tempo, para não empacotar os diferentes sistemas. Na matéria que foi necessária a minha intervenção, neste caso na Gestão de Tangíveis, apenas foi necessário parametrizar os documentos já existentes de acordo com os grupos acordados com a equipa de canais.

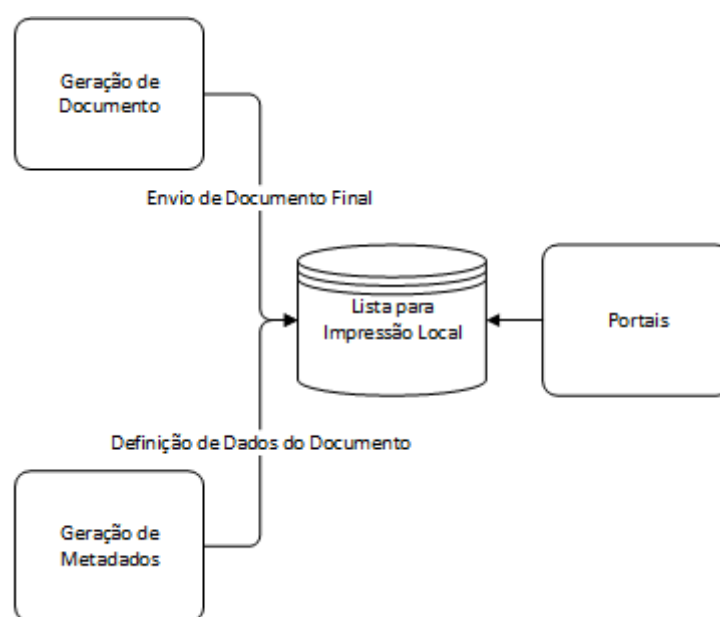


Figura 4.12: Envio e disponibilização de um documento para impressão local

## **Diferenciação de Documentos**

Como já falado anteriormente, a Unicre trabalha sobre duas áreas de negócio. Cada uma destas áreas tem particularidades a nível do negócio o que resulta em tangíveis com regras próprias e distintas entre as duas áreas. Consequentemente e fruto desta diferença entre áreas, os portais utilizados para consultar estes documentos também são distintos. Inicialmente e como o negócio de *Acquiring* foi o primeiro a entrar em produção, não foi tida em conta esta distinção entre áreas, resultando numa clara lacuna no que toca a disponibilização de documentos para impressão local.

Mais uma vez, este desenvolvimento foi feito em conjunto com a equipa de GT Issuing e as equipas de Canais. Planeou-se não só a informação que seria necessária enviar, para que posteriormente seja possível consultar o documento, bem como os grupos de impressão local a utilizar.

Iniciou-se então a alteração do fluxo no serviço SOA, que considerasse a área de negócio definida no documento. De acordo com este parâmetro, será consultada a parametrização do respetivo portal de destino, mais especificamente, da respetiva lista onde será armazenado o documento. Estas e outras parametrizações são armazenadas no *schema* UPCP numa tabela específica, a tabela de configurações gerais, que varia os valores das suas parametrizações consoante de ambiente para ambiente. Após acordadas e adicionadas as parametrizações do lado do gestor de tangíveis, apenas foi necessário adicionar o condicional que validasse a área de negócio e que de acordo com a mesma, obtivesse a configuração do portal e respetiva fila de impressão, tal como é possível verificar no esquema do anexo A.4.

## **4.8.2 Melhorias de Desempenho de Consultas e Processamento de Dados**

A Gestão de Tangíveis é um componente que lida com grandes quantidades de dados e dos mais variados tipos, muito graças ao facto de grande parte dos componentes do projeto interagirem com ela, maioritariamente, com o fim de serem gerados documentos. O formato dos dados varia conforme as componentes de onde provém, podendo variar desde um formato *comma separated values*, onde será lido e transformado com recurso a *scripts* SQLLoader, até um formato XML contido num envelope SOA, onde será processado e transformado através de um ou mais fluxos SOA.

Em termos de processamentos, existem dois tangíveis realmente morosos: os extratos

da direção de emissão, devido ao grande número de clientes *Issuing* e as cartas de informação financeira, que apesar de não serem geradas em tão grande número e serem geradas apenas uma vez por mês, exigem diversos cálculos aritméticos. Desta forma, estes dois processamentos são efetuados durante a noite de forma a não causarem entropias no normal funcionamento do Gestor de Tangíveis.

Algum tempo após a entrada em produção e resultando do aumento de informação armazenada nas base de dados, este processo tornou-se ainda mais moroso, de tal maneira que os processamentos além de se estenderem pela noite fora e por vezes até meio da tarde, começaram a causar *timeouts* na base de dados e problemas de bloqueio com outras gerações que vão sendo solicitadas ao longo dia.

Tal como mencionado anteriormente, o processamento destas cartas envolve uma grande número de operações aritméticas, neste caso concreto um número elevado de somas, o que resulta num elevado número de acessos às base de dados. Para minimizar o impacto causado por estas operações, converteu-se a obtenção e cálculo dos totais por cada iteração, onde cada iteração representa um comerciante, num cálculo massivo dos totais com recurso a tabelas temporárias, reduzindo desta forma o número de acessos às base de dados e a informação gerada, uma vez que estas tabelas eliminam a informação após ser terminada a transação.

Outra das melhorias feitas foi a exclusão de dados desnecessários bem como a agregação dos diferentes cursores num único, reduzindo também desta forma o acesso às diferentes tabelas e conseqüentemente o tempo de processamento.

A verdadeira dificuldade desta tarefa foi a complexidade da lógica do negócio, uma vez que das mesmas tabelas do ODS provêm os dados relativos tanto às marcas representadas (VISA, Mastercard, etc) ou não pela Unicre bem como os produtos financeiros (VISA Mastercard, VISA Crédito, etc) relativos aos diversos comerciantes e respetivos estabelecimentos.

### **4.8.3 Melhoria do Processo de Geração de Documentos**

Um dos primeiros problemas que foi identificado nas primeiras execuções, estava relacionado com *timeouts* relativos ao SOA. E porquê? O SOA tem um tempo limite de espera de vinte minutos quando faz pedidos assíncronos, neste caso a procedimentos PL/SQL.

## Base de Dados

Como estes procedimentos estavam a fazer operações morosas, foi necessário analisar e proceder a melhorias que permitissem ter melhor desempenho. O problema resolveu-se juntando todos os cursores existentes num só, eliminando assim X iterações para cada N cursores. Em conjunto com esta alteração fizeram-se também junções (*inner join*) entre tabelas de forma a eliminar registos incompletos, entenda-se por registos incompletos todos aqueles registos que através da utilização de *left joins* não contêm toda a informação necessária ao processamento, o que resultou na conseqüente redução de informação carregada em memória. Outra tarefa realizada afim de melhorar todo este processo, foi a remoção de funções proprietárias da oracle e que envolvem elevado e em certos casos, um desnecessário processamento, como é o caso das funções LPAD - função que acrescenta, antes de uma *string*, um determinado conjunto de caracteres até que a string atinja o tamanho pretendido - e TRUNC que no caso do GT, transforma uma data para um formato definido.

Resolvido os problemas de desempenho existentes a nível da base de dados, seria expectável que finalmente se conseguissem gerar cartas individuais e cartas massivas. Esta situação não se verificou, o BI Publisher estava com dificuldades em concretizar a geração da grande quantidade de cartas que determinados processamentos exigiam.

## BI Publisher

Numa fase posterior às correções efetuadas a nível de *scripts* e anteriores à afinação do BI Publisher, observou-se que cada carta estava a demorar um tempo anormal na sua geração. Este problema deve-se à forma como são lidos os dados dos ficheiros XML que permitem popular os modelos das cartas.

Procedeu-se então à análise e à correção do XPath utilizado para estas mesmas operações. Estas correções passaram por fazer uma análise completa à estrutura do XML e redefinir os seus XPaths, abandonando as procuras por varrimento completo ao documento e adotando o caminho específico dos elementos através do seu XPath completo. De forma a evitar pesquisas desnecessárias à árvore do XML, definiram-se ainda os quantificadores dos nós, evitando assim a repetição de procuras por determinados elementos que se repitam várias vezes.

O BI Publisher cria vários lotes para processamento de documentos, como tal precisa de bastante memória física para poder armazenar todo o conteúdo que irá ser colocado nos ficheiros, mas infelizmente, depois de processar a mesma, o *garbage collector* não funcionava corretamente, o que causava, ao fim de um período de tempo, um *memory*

*overflow*. Foi necessário configurar, testar, reconfigurar e retestar o BI Publisher até encontrar um conjunto de configurações que fosse capaz de cumprir estas tarefas, com grande volume de dados, do princípio ao fim. Por fim e uma vez que o *garbage collector* continuava com alguns problemas, a solução foi aumentar o tamanho dos lotes, ou seja, processar mais documentos de cada vez, uma vez que dentro do mesmo lote, o *garbage collector* até conseguia atuar de forma eficaz.

#### **4.8.4 Manutenção de Serviços SOA**

E como a gestão de tangíveis não consiste só em base de dados, surgiu também a necessidade de efetuar algumas intervenções a nível do SOA, nomeadamente a nível de mapeamentos, isto é, encaminhar a informação através dos diferentes fluxos de um *web service*, bem como, alterar transformações sobre informação proveniente de outras partes. O primeiro contacto sério com os serviços foi feito através da funcionalidade de arquivar tangíveis. Além do arquivo poder ser feito de modo online ou de modo massivo, como já foi mencionado anteriormente [], esta funcionalidade tem ainda dois fluxos internos distintos: um fluxo para as cartas, que geralmente representam quantidades reduzidas, e um outro para todos os extratos e quando falamos em extratos falamos sempre em quantidades consideravelmente grande. Após geração das cartas de informação financeira, deparamos-nos com um erro, que embora não fosse impeditivo, representava que algo no fluxo estava a falhar, a atualização do estado do arquivo. Depois da análise dos XML's utilizados para a troca de mensagens, concluiu-se que era um problema de mapeamentos, a informação relativa ao identificador do ficheiro estava em falta. A resolução deste problema até se revelou bastante simples, pois após ser identificada a etapa onde estava a ser efetuada esta mesma transformação foi uma questão de mapear corretamente a informação e proceder ao *re-deploy* do serviço.

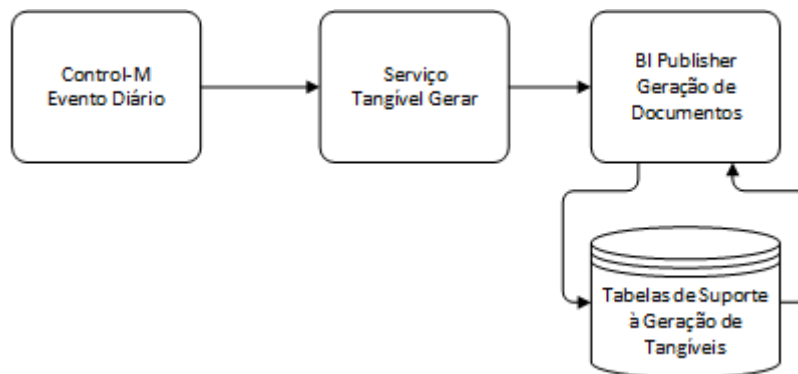


Figura 4.13: Fluxo de Geração de um Documento

### Cartões Matriz

Tal como em tantas outras instituições bancárias, a Unicre possui também um sistema de *homebanking* que permite aos seus clientes acederem e gerirem as suas contas bem como efetuar pagamentos, transferências bancárias, entre muitas outras operações. Para tal e à semelhança do que também existe noutras instituições, é necessário um cartão matriz para validar a identidade do titular da conta de forma a evitar situações fraudulentas. Estes cartões consistem em combinações únicas de números, identificados por linhas e colunas, que ficam associados a uma única conta. Após uma sessão de testes com os utilizadores funcionais, chegou-se à conclusão não apresentavam a estrutura desejada, nem que possuíam linhas e colunas suficientes para proporcionar a segurança desejável.

Surgiu então a necessidade de atualizar o componente de GT, mais concretamente a interface de comunicação com os portais, de forma a ter capacidade para suportar um novo formato de cartão enviado, um formato que fosse dinâmico para precaver novas alterações aos cartões. Assim sendo, qualquer que fosse o novo formato a adotar, o GT teria a capacidade de armazenar, tratar e entregar à entidade externa de impressão, Contiforme, a informação proveniente do Backoffice DE. Esta campo passou a ser composto por tuplos, cada um com uma chave e um valor, onde a chave representa a posição da matriz onde será colocado o dígito existente no campo valor.

Como já foi mencionado anteriormente, quando existe a necessidade de incluir novos campos para serem utilizados por um serviço, existe sempre a necessidade de adequar a sua interface, desta forma e à semelhança do que já havia sido feito anteriormente, foi necessário alterar o XML *schema* que define a interface do serviço de geração de tangíveis relacionados com os pedidos provenientes dos componentes da equipa de canais.

O grande desafio destas alterações, foi a necessidade de adequar os tangíveis de forma a que suportassem qualquer tipo de cartão matriz, pois não fazia sentido ter dinamismo a nível de lógica e depois ficar-se limitado pelos tangíveis. Uma vez que esta informação é armazenada numa única string, onde os pares chave-valor são separados por um hífen e os tuplos estão separados por um ponto e vírgula, foi necessário recorrer às funções avançadas do BI Publisher. Para os modelos RTF, o BI Publisher tem um conjunto de funções SQL e XSL que permitem, entre outras coisas, procurar caracteres dentro das strings (INSTR) e cortar as mesmas (SUBSTR). Com recurso à função XSL LENGHT, é possível determinar a dimensão do cartão matriz e desta forma, com auxílio de variáveis, fazer o substring e construir a matriz.

## 4.9 Pedidos de Adesão

Um pedido de adesão consiste no processo de adesão de um cliente, particular ou empresarial a um serviço prestado pela Unicre, quer seja de *Acquiring* quer seja de *Issuing*.

Como tal, existem uma série de conceitos próprios desta área de negócio, conceitos estes que tiveram de ser explorados, solicitados e compreendidos de forma a dar um melhor suporte e compreender melhor os desenvolvimentos pretendidos.

Um pedido de adesão é composto por diversas entidades que se relacionam entre si. Se dispusermos estas entidades hierarquicamente, encontramos no topo as empresas. As empresas que podem ser clientes particulares ou empresas. Cada pedido de adesão é feito exclusivamente para uma única empresa, mas estas por sua vez, podem aderir a vários produtos ou serviços fornecidos pela Unicre. Estas têm associadas às mesmas, uma ou mais lojas, que representam os estabelecimentos físicos ou mesmo virtuais, como são o caso das lojas de *e-commerce*. Por sua vez, as lojas têm um ou mais terminais de pagamento automático (TPA). Os terminais dividem-se em dois modos de funcionamento distintos: uma modalidade *online*, onde o terminal só aceita pagamentos feitos com chip e comunica uma transação no preciso momento em que esta é feita e uma modalidade *offline*, onde o terminal aceita pagamentos *contactless* e guarda todos os pagamentos para posteriormente os submeter, por exemplo ao fim do dia. Os terminais têm associados produtos financeiros que representam os produtos disponibilizados pelas diferentes empresas de serviços financeiros, como são os casos da VISA e da Mastercard. De acordo com o tipo de terminal, chip ou *contactless*, irão

ser consideradas ou não tarifas *online* ou *online e offline*. Estas tarifas representam os valores que serão cobrados ao cliente sobre cada transação. Estas podem refletir as tarifas da entidade responsável SIBS ou podem ser definidas na altura do pedido de adesão.

#### **4.9.1 Siebel**

O componente de CRM é um dos vários que interage diretamente com a gestão de tangíveis. Este componente está encarregue, entre outras responsabilidades, de gerir e gerar os pedidos de adesão dos clientes a novos produtos ou serviços da RedeUnicre. De entre as várias funcionalidades que este componente fornece, a criação de pedidos de adesão são a que mais importa para a gestão de tangíveis, pois será através deles que será feita a interação com este componente.

Os pedidos de adesão (PA) podem ser divididos em dois tipos: na componente de *Issuing*, temos os pedidos de adesão de produtos da direção de emissão (DE) enquanto que na componente de *Acquiring*, existem os pedidos de adesão de condições particulares. A geração de tangíveis referentes a pedidos de adesão pode ser feita de duas maneiras distintas:

1. Os documentos mais simples e que requerem poucos mapeamentos e pouca informação, podem ser gerados pelo próprio Siebel e enviados para o GT apenas para conversão para PDF e posterior expedição;
2. Os documentos que exijam maior lógica e processamento são gerados do lado do Gestor de Tangíveis. A informação que é utilizada para estes documentos pode provir tanto do próprio Siebel, passando toda a informação necessária à geração do documento através da invocação do serviço, ou através do ODS, sendo que neste caso as invocações feitas pelo Siebel apenas têm o identificador do pedido de adesão.

#### **Geração de Pedidos de Adesão**

Em parceria com a equipa de CRM, desenvolveu-se um processo que dá suporte e automatiza a aceitação, ou não, de pedidos de adesão DE efetuados fora dos escritórios da Unicre. De forma a registar estes pedidos, existe a necessidade de preencher um documento igual ao que é utilizado pelo processo eletrónico, no qual e com recurso mais uma vez ao gestor de tangíveis, é preenchido automaticamente com a informação

introduzida no portal RUOnline.

Visto que estes pedidos têm de ser digitalizados e posteriormente introduzidos no Filenet em formato de imagem (TIFF), para que possam ser aplicados filtros que melhoram a qualidade da imagem, e uma vez que o gestor de tangíveis não tem a capacidade de trabalhar com este tipo de documento, surgiu a necessidade de implementar uma solução que colmatasse esta lacuna, ou seja, que convertesse o pedido de adesão digitalizado em formato TIFF para um formato suportado pela GT que neste caso será um PDF.

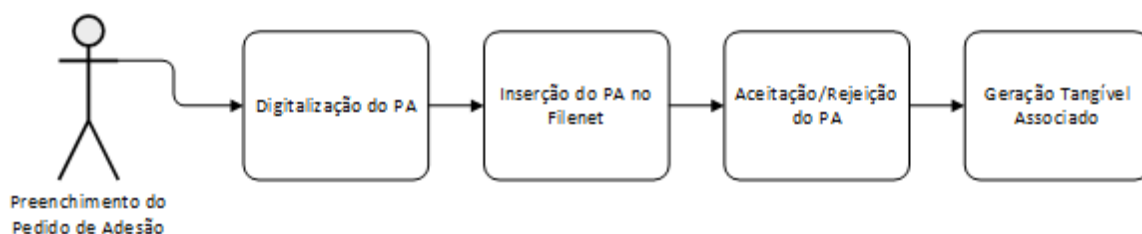
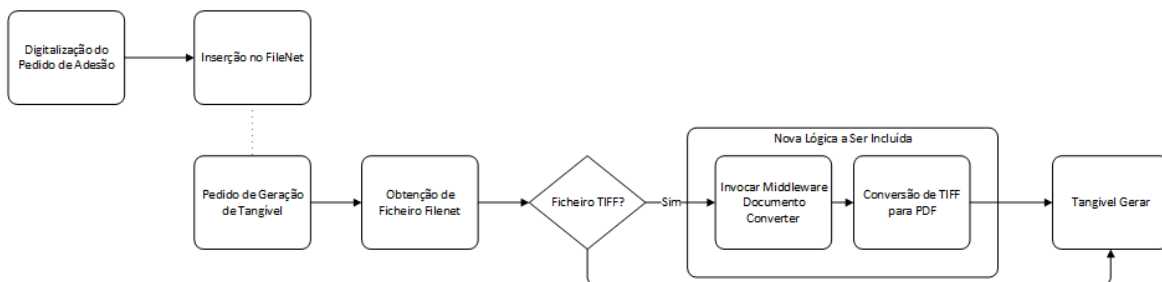


Figura 4.14: Ciclo de um pedido de adesão DE

Depois analisar o que já se encontrava implementado e perceber onde iriam incidir as novas alterações, concluiu-se que o mais viável seria adicionar uma nova funcionalidade ao já existente *middleware* que contorna as limitações da ferramenta BI Publisher.

Esta nova funcionalidade passa por adicionar um novo método no serviço documento converter, onde será recebido um pedido com o identificador da digitalização armazenada no *Filenet*. Posteriormente irá ser invocada uma aplicação externa que efetua a conversão da mesma para o formato PDF.

Uma vez que todo este processo é controlado pelos serviços SOA, foi também necessário efetuar alterações ao fluxo relativo ao controlo da composição. Estas alterações passaram, essencialmente, por verificar o tipo de ficheiro obtido do *Filenet*, através do tipo do conteúdo da mensagem, uma vez que é devolvido o conteúdo do ficheiro e não o ficheiro em si, e invocar, ou não, a conversão de TIFF para PDF.



## 4.10 Suporte

Após realizadas sessões de testes com os responsáveis do negócio, surgiu a necessidade de alterar a lógica de expedição associada aos pedidos de adesão: as cartas, que haviam sido pensadas apenas para serem expedidas para a impressão local, passaram a ser expedidas por email. Para tal, foi necessário efetuar alterações a nível da lógica associada à base de dados, adequação dos procedimentos de carregamento para as tabelas de *staging*<sup>11</sup> e ainda dos procedimentos de carregamento para as tabelas de suporte à geração de documentos. A nível de SOA, foram remapeados os campos, alteradas as transformações e recarregados os esquemas XSD de forma a incluírem as alterações feitas na base de dados.

Consequentemente, estas alterações levantaram novas questões associadas à expedição destes mesmos tangíveis, foi necessário considerar a agregação de documentos quando estes fossem expedidos por correio eletrónico, para que o cliente não receba uma mensagem com vários anexos, ou ainda para impressão local, visto que os formulários dos pedidos de adesão podem ser impressos em branco para ser feita a angariação de clientes na rua. Para tal, foi necessário adequar os serviços SOA associados a estes dois métodos de expedição, de forma a considerarem no seu processamento, a agregação de documentos, que já se encontrava implementada no módulo JAVA agregado ao BI Publisher, uma vez que a expedição para arquivo também segue esta mesma lógica.

## 4.11 Fusão entre Áreas de Negócio

As áreas de negócio sobre a qual a Unicre trabalha, *Acquiring* e *Issuing*, apesar de

<sup>11</sup>tabelas intermédias que servem para guardar temporariamente informação

terem bastantes aspetos em comum têm também algumas particularidades que as diferenciam. Estas particularidades também se refletem na lógica da Gestão de Tangíveis, nomeadamente nos serviços de expedição e no carregamento de informação para a geração dos documentos.

Esta diferenciação entre áreas de negócio não foi refletida desde o início da implementação do projeto MMK, uma vez que a sua disponibilização foi feita, tal como mencionado anteriormente, em diferentes alturas. Esta diferença de *deadlines*<sup>12</sup>, resultou numa posterior adequação dos serviços e processos já existentes. Visto que havia a necessidade contínua de manter e dar suporte ao ambiente de produção, estas alterações tiveram que ser mantidas num *branch*<sup>13</sup> diferente dos desenvolvimentos de *Acquiring*, para que não houvesse risco de ser feita uma passagem que estragasse o ambiente de produção. De forma a não inutilizar o ambiente de produção, uma vez que tempo é dinheiro e o negócio do dinheiro não para, houve a necessidade de efetuar uma fusão entre desenvolvimentos. Esta fusão consistiu em utilizar uma cópia do ambiente de produção e aplicar as alterações necessárias para o funcionamento do componente de *Issuing*, de forma a criar um pacote que para além das alterações ainda aplicasse correções de forma a que a aplicação do pacote fosse efetuada com o máximo sucesso.

## 4.12 SAC\_GIT

O SAC\_GIT também conhecido por Suporte ApliCacional para Geração de Informação para Terceiros, é um *schema* que contém dados que dá suporte aos componentes que geram informação para entidades externas, como é o caso do Banco de Portugal. O SAC\_GIT acaba por funcionar apenas como um sistema de suporte ao Integrador Assíncrono de Ficheiros, cuja principal funcionalidade é processar ficheiros, e ao Gestor de Tangíveis, cujos dados poderão ser provenientes do Way4 e do sistema central ODS.

**Melhorias de Desempenho no Carregamento de Dados da SAC\_GIT** Uma vez que existem processamentos de grandes quantidades de informação, revelou-se que seria necessário alterar alguns dos processos já existentes, de forma a não só não consumirem tanto espaço mas como também a serem mais eficientes, uma vez que existem diversas dependências com outros processos.

---

<sup>12</sup>Prazo de entrega de um desenvolvimento

<sup>13</sup>Ramificação paralela dos desenvolvimentos

Estas melhorias necessitaram de uma análise profunda aos diversos carregamentos, de forma a perceber que informação é que é realmente utilizada. Desta forma conseguiu-se uma simplificação e união das consultas, o que acaba por reduzir o número das iterações e dos acessos desnecessários às tabelas do *schema*. Outro ponto de melhoria identificado foi a utilização excessiva e desnecessária de funções Oracle com um grande *overhead*, tais como o *trunc* ou mesmo o *lpad*, sendo que estes foram eliminados e as datas utilizadas nos *truncs* foram todas uniformizadas.

## 4.13 Certificação de Faturação

Com as constantes alterações que são feitas à nossa constituição, surgiu a necessidade de implementar um módulo que fosse capaz de certificar as faturas emitidas pelo sistema central Way4. Portanto e devido aos requisitos de certificação de faturas impostos pelas portarias número 22-A/2012 [Tributaria, 2012], que regulamenta a utilização obrigatória de programas informáticos de faturação certificados e a emissão de documentos por equipamentos ou programas não certificados e número 160/2013 [Tributaria, 2013] que consiste em duas alterações, uma à portaria número 321-A/2007 que define o ficheiro modelo de auditoria tributária e uma outra à portaria 363/2010 que regulamenta a certificação prévia dos programas informáticos de faturação do Código do Imposto sobre o Rendimento das Pessoas Coletivas, surgiu a necessidade de implementar um novo módulo que fosse capaz de cumprir com estas novas obrigações, visto que o atual sistema de processamento de pagamentos não tem a capacidade de efetuar este tipo de certificação.

Estas alterações abrangem vários componentes do projeto MMK, sendo um deles o Gestor de Tangíveis e visto que já seria envolvido neste componente devido às alterações que os tangíveis necessitam, foi-me ainda delegado o desenvolvimento de um módulo de controlo responsável pelo processamento do *hash*, visto que, tal como mencionado anteriormente, o GT faz utilização de um módulo JAVA para efetuar processamento à medida.

O gerador de *hash* foi desenvolvido com o intuito de conseguir processar o maior número de faturas no menor intervalo de tempo possível. Para tal, utiliza um número parametrizável de *threads*, que processam a informação obtida da respetiva tabela existente no ODS. Este processamento consiste em obter a informação da fatura anterior à fatura a ser processada (se aplicável) e cifra-la. Para efetuar esta cifra, o gerador utilizar uma

chave privada associada à entidade, neste caso à Unicre.

A utilização de chaves privadas do tipo PEM revelou-se uma verdadeira dor de cabeça, visto que o JAVA não suporta de forma nativa este mesmo tipo. A solução para este problema passou por utilizar uma biblioteca externa, que dá suporte a este tipo de chaves por parte do JAVA. Com este processo, consegue-se garantir que as faturas não são adulteradas, visto que diferentes dados geram diferentes cifras. Findo este processo, o módulo irá atualizar o ODS com a informação gerada.

O resultado final foi um executável JAR que recebe um ficheiro com informação relativa a todas as faturas a processar e que, após efetuar o *hash* e o assinar com a chave privada em questão, atualiza os dados no ODS com a respetiva assinatura gerada.

### **Integração com o Gestor de Tangíveis**

Como mencionado na secção 4.13, a informação relativa à certificação de faturação foi incluída no GT, mais concretamente nos documentos gerados pelo mesmo.

Estes desenvolvimentos dividiram-se em duas etapas distintas: pré e pós certificação, onde a mudança de uma etapa para a outra foi ditada pela aprovação deste mesmo processo pela autoridade tributária.

Uma vez que surgiu estes desenvolvimentos não se fazem de um dia para o outro e ainda para mais quando necessitam da intervenção de uma entidade externa, houve a necessidade de proceder a alterações que demonstrassem que estavam a ser feitos os desenvolvimentos de forma a considerar esta exigência. Esses desenvolvimentos consistiram na alteração dos modelos associados aos tangíveis a serem considerados para o processo de certificação, neste caso as faturas da área de negócio de *Acquiring*. Por sua vez, os desenvolvimentos relativos à certificação em si, exigiram algo mais, exigiram alterações relativas ao negócio, nomeadamente a consideração e inclusão de nova informação para dar suporte à geração de documentos. Uma vez que a legislação portuguesa obriga a que as faturas originais sejam impressas uma única vez, sempre que for solicitada novamente a mesma fatura deverá ser impresso o duplicado ao invés da original, este processo serviu ainda para identificar se uma fatura foi emitida para efeitos de formação e/ou de testes, utilizando para tal um campo condicional que irá ativar um texto estático de acordo com o ambiente onde o mesmo for gerado. Foi ainda necessário armazenar na base de dados informação relativa ao código de entidade certificadora atribuído à Unicre bem como quatro caracteres do *hash* gerado, que validam a autenticidade do documento e que irão aparecer no rodapé do mesmo.

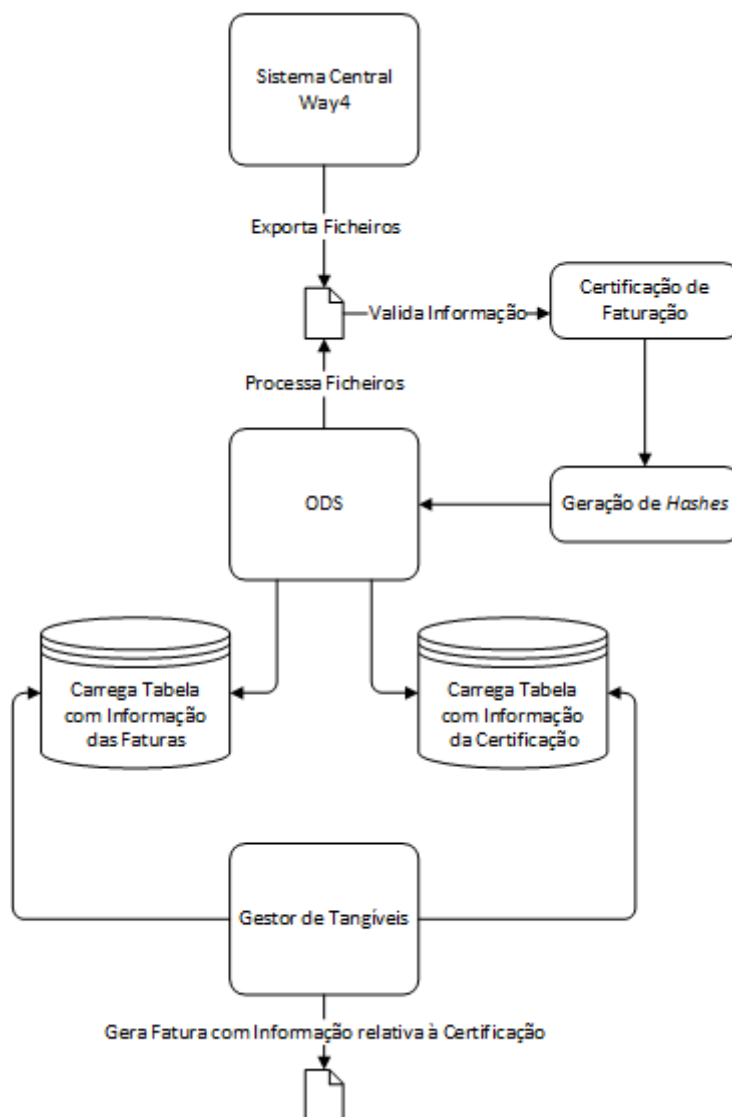


Figura 4.15: Processo de Certificação de Faturas

## 4.14 Gestão de Risco

Como mencionado num capítulo anterior, a Unicre já tem uma vasta experiência nas áreas de **acquiring** e de **issuing** resultante de vários anos de negócio, tantos anos ao ponto da informática ainda não ter a dimensão que tem hoje. Como tal, muitos dos dados relativos a clientes existentes no sistema anterior são escassos em informação, nomeadamente ao nível de números de contribuinte, visto que no antigamente não havia a obrigatoriedade de fornecer o mesmo aquando da adesão a um produto bancário.

De forma a centralizar os dados da RedeUnicre num único sistema, houve a necessidade de implementar mecanismos que permitissem tratar e uniformizar estes mesmos dados. Outro tema relativo à base de dados de risco, foi a melhoria de desempenho na pesquisa, via portal de *backoffice*, de entidades. Como estas pesquisas envolvem um elevado grau de possibilidades e de comparações (através da cláusula *LIKE*) que possibilitam a procura de uma entidade através de informação parcial, como por exemplo, uma parte de uma morada. Visto que os tempos verificados não eram aceitáveis, foi necessário afinar os acessos à base de dados, isto é, criar índices e reduzir a informação comparada e processada nestes mesmos acessos. Finda esta análise, procedeu-se à criação de índices nas colunas sobre as quais são efetuadas as consultas. Alterou-se também o código SQL de forma a só efetuar a comparação parcial de campos que assim o justifiquem e se estes forem diferentes de vazio (NULL).

# Capítulo 5

## Conclusão

Durante este estágio e muito devido à exigência de um projeto de consultoria, ganhei experiência e maturidade em vertentes funcionais, técnicas e de negócio. Esta experiência fez de mim um melhor e mais completo profissional, conseguindo não só ser um simples consultor com conhecimentos técnicos avançados mas também um consultor dotado em diferentes áreas financeiras.

Graças há constante necessidade de criar soluções num curto espaço de tempo, e esta necessidade obriga-nos a ter e a procurar, não só conhecimento das potencialidades e limitações das tecnologias bem como conhecimento funcional relativo à lógica de negócio montada, revelei uma necessidade de adquirir conhecimento, que fora de um ambiente profissional, ainda não se havia revelado. Esta procura de conhecimento fez com que desenvolvesse a minha autonomia e as minhas capacidades de comunicação, dado que muitas das vezes esta tinha de ser feita com pessoas do cliente. Esta comunicação dotou-me também com algo muito importante que é a capacidade de comunicar com pessoas sem conhecimentos técnicos, uma vez que são estas que têm o conhecimento funcional que necessitamos para que as nossas soluções façam sentido mas que em contrapartida não têm qualquer conhecimento tecnológico. Toda esta interação com o cliente providenciou-me uma destreza e maturidade não só a compreender o que um utilizador não técnico pretende.

Adicionalmente e devido às responsabilidades e tarefas definidas pela hierarquia da *Deloitte*, houve também uma notável evolução a nível de organização e planeamento, havendo constantemente a necessidade de planear e estimar o meu trabalho. Este planeamento forçou-me sempre a manter tudo bem organizado, nomeadamente na produção de evidências e preparação de sessões de esclarecimento com pessoas apenas

funcionais, o que obrigou sempre a um trabalho extra para simplificar e clarificar as questões técnicas que para nós se revelam triviais.

Além de todo o enriquecimento pessoal e ao facto de ter saído um pouco do âmbito da computação móvel, este estágio trouxe-me ainda valências a nível tecnológico, uma vez que tive a oportunidade de aprender e especializar-me em tecnologias com que nunca antes me tinha cruzado, nomeadamente as tecnologias SOA e PL-SQL da Oracle, cuja procura e reconhecimento no mercado continua em grande. O facto de ter assumido a responsabilidade exclusiva de um dos componentes críticos do projeto, tornou-me alguém muito mais pro-ativo e muito mais autodidata, tendo em conta que para conseguir compreender e ganhar algum conhecimento sobre o componente em si e das suas tecnologias, tive de lutar, explorar e aprender a extrair e aproveitar o conhecimento das pessoas com mais experiência, onde mais uma vez a comunicação, embora que desta vez com os colegas de empresa, tenha sido crucial para o meu sucesso.

Toda esta experiência, desde o primeiro dia na academia até aos dias de hoje no projeto, providenciaram-me conhecimentos em duas outras áreas que embora não tivesse qualquer tipo de conhecimento, se relacionam de forma intensa com a informática: a área financeira e a área dos seguros, embora que nesta última o contacto tenha sido breve. Estes conhecimentos e este contacto ensinou-me, além de diversos conceitos e funcionamento de instituições financeiras, a ver mais além, a compreender como as coisas funcionam realmente, a compreender o porquê do código que estou a desenvolver. Este estágio fez de mim uma pessoa muito mais completa, aprendi a ter um pensamento *one step ahead* que sem uma experiência profissional deste nível não teria sido possível, aprendi a ver bem para lá da informática.

# Bibliografia

[Alapati, 2011] Alapati, S. (2011). *Oracle WebLogic Server 11g Administration Handbook*. Oracle Press.

[Application, 2014] Application, T. I. (2014). Tia.  
<http://www.is4insurance.com/TIA-678.html>.

[big4accountingfirms, 2014] big4accountingfirms (2014). Who are the big 4 accounting firms? <http://www.big4accountingfirms.org/>.

[BMC, 2014] BMC (2014). Control-m workload automation.  
<http://www.bmc.com/it-solutions/control-m.html>.

[IBM, 2014] IBM (2014). Filenet - an ibm ecm product family.  
<http://www-01.ibm.com/software/ecm/filenet/>.

[Janssen, 2014] Janssen, C. (2014). Extract transform load (etl).  
<http://www.techopedia.com/definition/24170/extract-transform-load-etl>.

[Limited, 2014] Limited, D. T. T. (2014). About deloitte portugal.  
<http://www2.deloitte.com/pt/pt/legal/about-deloitte-portugal.html>.

[Microfocus, 2003] Microfocus (2003). *Introduction to the COBOL Language*.  
<https://supportline.microfocus.com/Documentation/books/nx40/lrintr.htm>.

[Microsoft, 2014a] Microsoft (2014a). Model-view-controller.  
<http://msdn.microsoft.com/en-us/library/ff649643.aspx>.

[Microsoft, 2014b] Microsoft (2014b). Sql server integration services ssis.  
<http://technet.microsoft.com/en-us/library/ms141026.aspx>.

[Microsoft, 2014c] Microsoft (2014c). Sql server reporting services ssrs.  
<http://msdn.microsoft.com/en-us/library/ms159106.aspx>.

[Microsoft, 2014d] Microsoft (2014d). Understanding lightswitch.  
<http://msdn.microsoft.com/en-us/vstudio/htmlclient.aspx>.

[Microsoft, 2014e] Microsoft (2014e). Visual c#. <http://msdn.microsoft.com/en-US/library/kx37x362.aspx>

[on the Net, 2014] on the Net, T. (2014). Oracle/plsql.  
<http://www.techonthenet.com/oracle/>.

[Oracle, 2009] Oracle (2009). *Getting Started with Oracle Business Rules*. Oracle.

[Oracle, 2011] Oracle (2011). *Oracle Reports Services 11g*.

[Oracle, 2013] Oracle (2013). *Fusion Middleware Developer's Guide for Oracle SOA Suite*. Oracle.

[Oracle, 2014a] Oracle (2014a). Introducing oracle business intelligence enterprise edition.

[Oracle, 2014b] Oracle (2014b). Oracle forms services 11g.  
<http://www.oracle.com/technetwork/developer-tools/forms/overview/index-098877.html>.

[Oracle, 2014c] Oracle (2014c). Service-oriented architecture (soa) definition.  
[http://www.service-architecture.com/articles/web-services/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html).

[Rouse, 2013] Rouse, M. (2013). Dnn (dotnetnuke).  
<http://whatis.techtarget.com/definition/DNN-DotNetNuke>.

[SCRUM.org, 2010] SCRUM.org (2010). What is scrum?  
<https://www.scrum.org/resources/what-is-scrum/>.

[Technology, 2014] Technology, A. (2014). What is microsoft excel and when would i use it? [http://spreadsheets.about.com/od/tipsandfaqs/f/excel\\_use.htm](http://spreadsheets.about.com/od/tipsandfaqs/f/excel_use.htm).

[Tributaria, 2012] Tributaria, A. (2012). Portaria no. 22-a/2012.  
<http://info.portaldasfinancas.gov.pt/NR/rdonlyres/8C23078D-3B1F-45EE-9D02-C01C2DCA6949/0/portaria22A2012.pdf>.

[Tributaria, 2013] Tributaria, A. (2013). Portaria no. 160/2013.  
<http://www.ascmi.com.pt/portaria160-2013.pdf>.

[Unicre, 2012] Unicre (2012). Actividade. <http://www.unicre.pt/site/>.

[zingCOBOL, 2014] zingCOBOL (2014). A beginner's guide to cobol programming.  
<http://cobol.404i.com/>.

# **Apêndice A**

## **Anexos**

# 8 A.1 Arquitetura do Projeto MMK

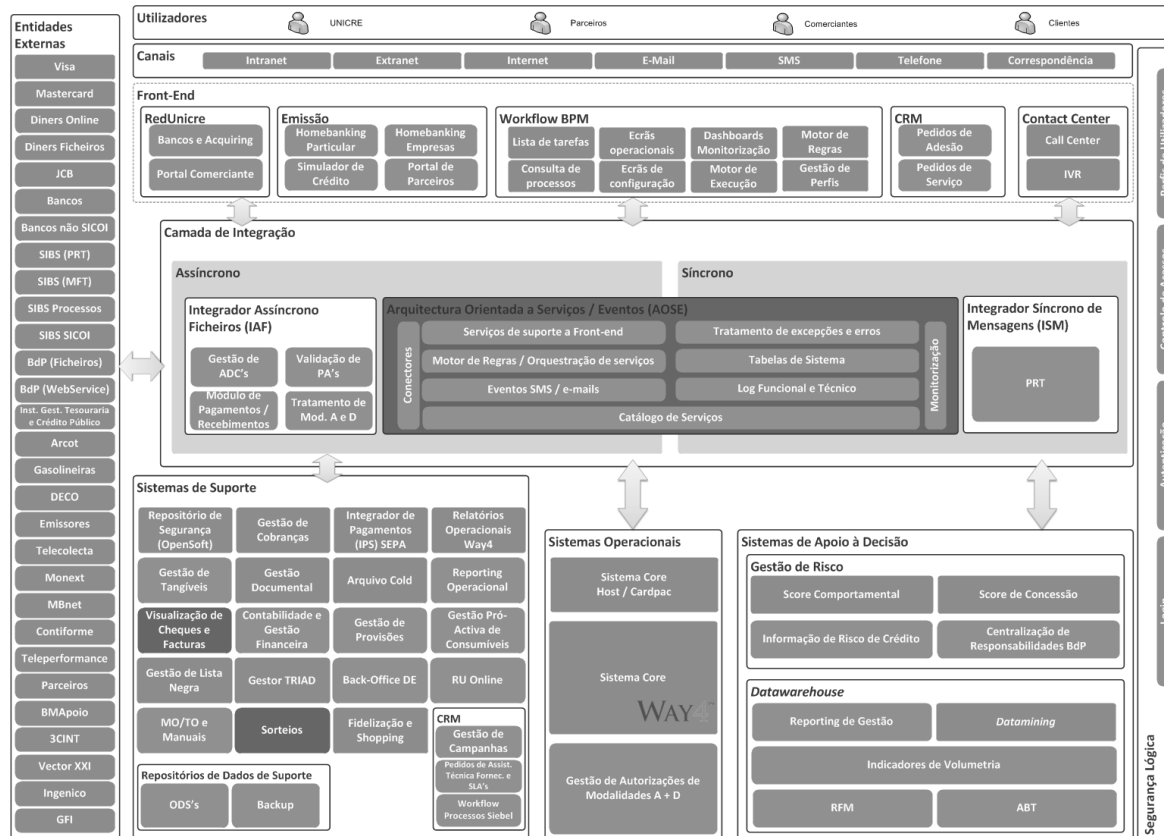


Figura A.1: Arquitetura do Projeto MMK

## A.2 SOAP UI

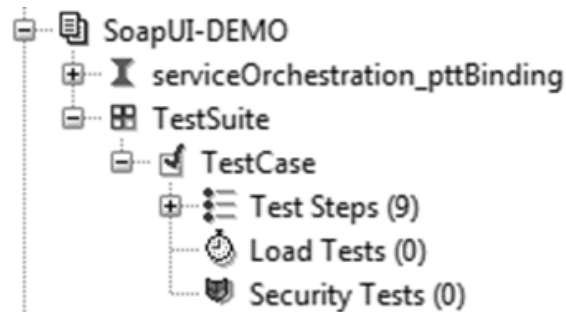


Figura A.2: Exemplo de uma estrutura de testes no SOAPUI

## A.3 Expedição por Email

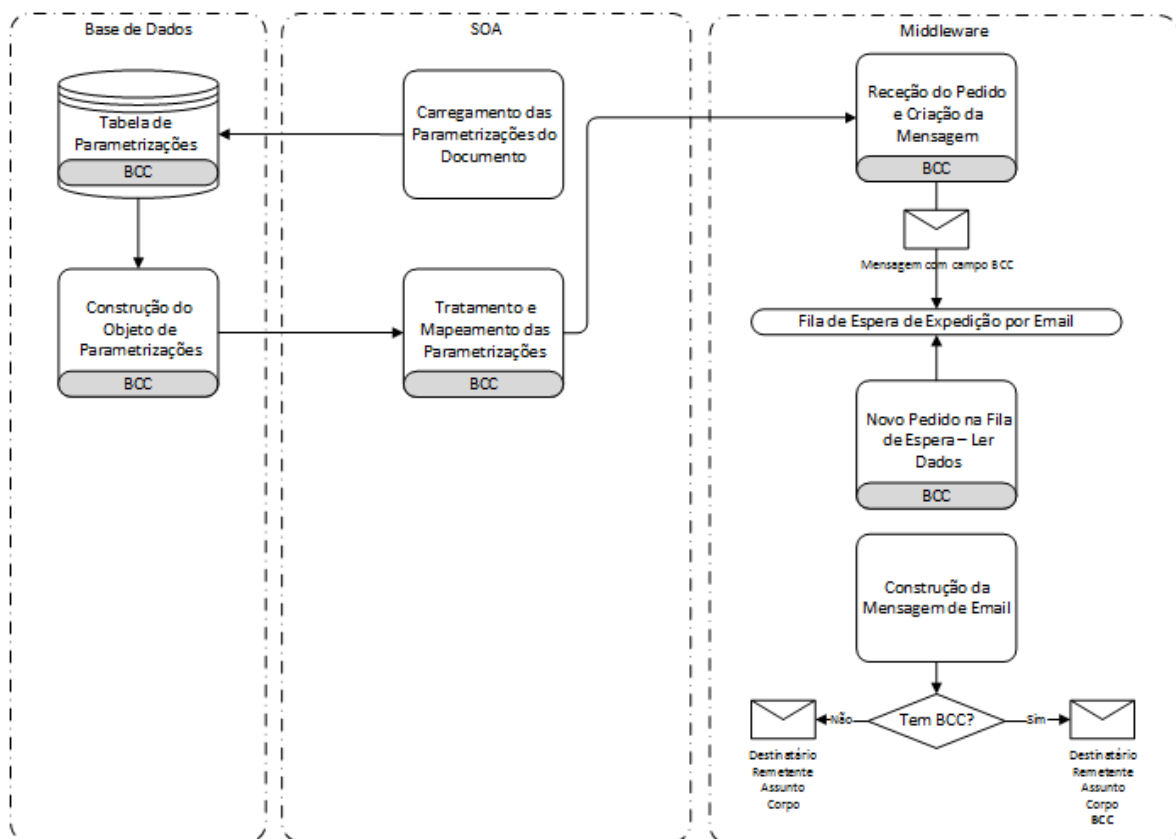


Figura A.3: Adição de campo BCC na Expedição por Email

## A.4 Expedição para Impressão Local

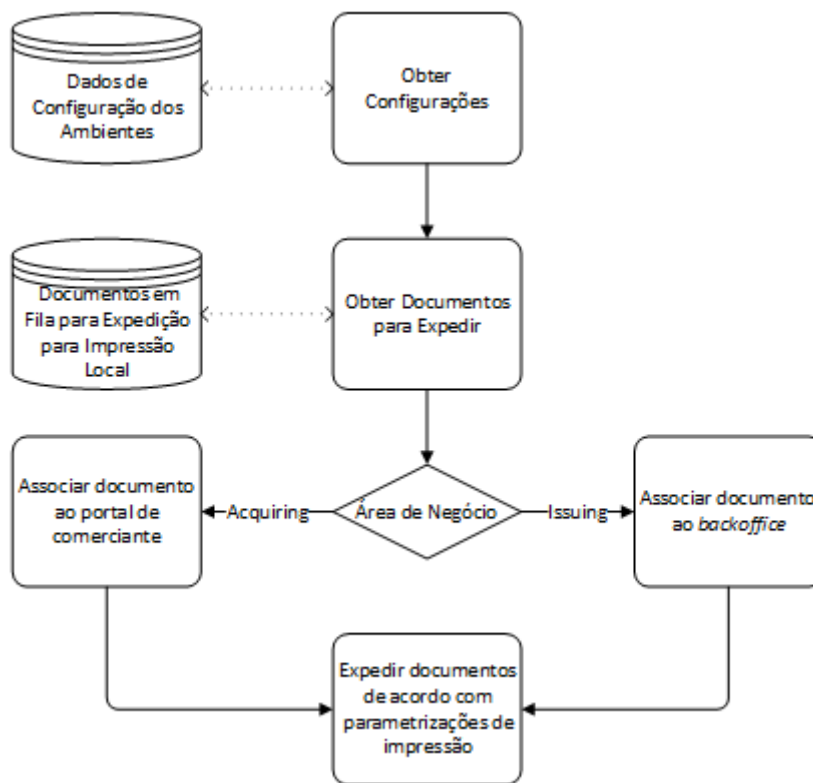


Figura A.4: Diferenciação de Documentos por Área de Negócio

## A.5 Detalhe dos Critérios de Expedição Interna

Tabela A.1: Detalhe dos Critérios de Expedição Interna

Impressão Local	Outras Entidades	Descrição
SIM	NÃO	Assume expedição via Impressão Local, impedindo o envio para expedição externa.
NÃO	SIM	Assume expedição via Outras Entidades, impedindo o envio para expedição externa.

Tabela A.1: Detalhe dos Critérios de Expedição Interna (continuação)

Impressão Local	Outras Entidades	Descrição
SIM	SIM	Assume expedição via Impressão Local e Outras Entidades, impedindo o envio para expedição externa.

## A.6 Detalhe dos Critérios de Expedição Externa

Tabela A.2: Detalhe dos Critérios de Expedição Externa

Email	Email Certificado	Finishing	
SIM	NÃO	NÃO	Assume unicamente a expedição via Email, falhando caso o cliente não tenha email definido ou o contacto preferencial definido como correio.
NÃO	SIM	NÃO	Assume unicamente expedição via Email Certificado, falhando caso o cliente não tenha email definido ou o contacto preferencial definido como correio.
NÃO	NÃO	SIM	Assume unicamente expedição via <i>Finishing</i> .

Tabela A.2: Detalhe dos Critérios de Expedição Externa (continuação)

Email	Email Certificado	Finishing	
SIM	NÃO	SIM	Caso o contacto preferencial seja distinto de correio (entenda-se não definido, ou restantes possibilidades) e o cliente tenha email definido, assume expedição via email. Restantes cenários realiza a expedição via <i>Finishing</i> .
NÃO	SIM	SIM	Caso o contacto preferencial seja distinto de correio (entenda-se não definido, ou restantes possibilidades) e o cliente tenha email definido, assume expedição via email Certificado. Restantes cenários realiza a expedição via <i>Finishing</i> .
SIM	SIM	NÃO	Assume unicamente expedição via Email Certificado (falhando caso o cliente não tenha email parametrizado).
SIM	SIM	SIM	Caso o contacto preferencial seja distinto de correio (entenda-se não definido, ou restantes possibilidades) e o cliente tenha email definido, assume expedição via Email Certificado. Restantes cenários realiza a expedição via <i>Finishing</i> .