

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

**ESTUDO DA INTERACÇÃO DO AIP
COM OUTRAS APLICAÇÕES
ORACLE RETAIL**

João Paulo Ribeiro Portásio

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Raul Moreira Vidal (Professor Associado)

Junho de 2010

Estudo da Interacção do AIP com Outras Aplicações Oracle Retail

João Paulo Ribeiro Portásio

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: João Carlos Pascoal de Faria (Professor Auxiliar)

Vogal Externo: Luis Manuel Borges Gouveia (Professor Associado)

Orientador: Raul Fernando de Almeida Moreira Vidal (Professor Associado)

28 de Junho de 2010

Resumo

Actualmente surgiu no mercado um novo sistema de informação dedicado ao mundo do retalho. Esse sistema, denominado AIP, tem como objectivo elevar a gestão da cadeia de abastecimento de um retalhista ao seu mais alto nível.

Como a aplicação ainda é bastante recente, as provas dadas por ela ainda são muito reduzidas, fazendo com que alguns retalhistas fiquem relutantes com a ideia de investir na mesma. Portanto é um passo fundamental fazer melhorias e estudos sobre este sistema de forma a torná-lo mais apetecível e credível no mercado do retalho.

Para tal com este projecto pretende-se realizar uma melhoria à interacção que o AIP tem com outras aplicações *Oracle Retail*. Este último, é o conjunto de aplicações *Oracle* dedicadas ao mundo do retalho. Melhorando a interacção do AIP fazemos com que o seu tempo de execução em determinadas tarefas diminua, permitindo assim dar mais garantias de fiabilidade da aplicação perante os retalhistas.

Então a abordagem ao projecto foi pensada da seguinte maneira: fazer uma análise às ferramentas que podem ser utilizadas para integrar aplicações *Oracle Retail*; realizar uma comparação entre as várias ferramentas seleccionadas, de forma a tirarmos conclusões do funcionamento das mesmas; fazer um estudo sobre como implementar alguma melhoria com os resultados obtidos na comparação anterior, isto se for possível melhorar a aplicação; implementar a melhoria indicando as mais valias que esta trouxe para a aplicação.

Para a comparação das ferramentas foram escolhidas o RIB, o RETL e o PL/SQL com SQL*Loader. Desta comparação concluiu-se que a última não trazia vantagens em relação às outras duas e que essas têm um funcionamento e uma aplicabilidade completamente diferentes no que toca à integração de aplicações *Oracle Retail*.

Com esta comparação conseguiu-se alcançar uma melhoria possível ao AIP. Essa melhoria passou por criar um novo conceito de interfaces de integração que consegue reunir as vantagens que tanto a ferramenta RIB como a RETL têm. A partir deste conceito implementou-se a interface tornando assim o AIP uma aplicação melhor e mais atraente para os seus potenciais clientes.

RESUMO

Abstract

Nowadays appears on market a new information system dedicated to the world of retail. This system, called AIP, aims to enhance the management of the supply chain of a retailer to its highest level.

Since the application is still fairly recent, it still has to prove its commercial value and some retailers remain reluctant about investing in it. Therefore it is a crucial step to improve this system and consequently increase its interest and credibility in the retail market.

So this project aims to achieve an improvement to the interaction of AIP with other *Oracle Retail* applications. The *Oracle Retail* is a suite of Oracle applications dedicated to retail market. Improving the AIP interaction makes its run-time decrease on certain tasks, thus giving more reliability to the application before the retailers.

So the approach to the project was designed as follows: analyze the tools that can be used to integrate *Oracle Retail* applications; make a comparison between the various tools selected, in order to draw conclusions from the tools operations; do a study on how to implement any improvements on AIP based in the results obtained in the previous comparison and if is possible to improve; implement the improvement and showing the gains that it brought to the application.

For comparison of the tools were chosen RIB, RETL and PL / SQL with SQL * Loader. From this comparison it was concluded that the last one brought no advantages over other two and the other two have a completely different operations, regarding to the integration of *Oracle Retail* applications.

With this comparison we were able to achieve a possible improvement to the AIP. This improvement went on to create a new concept of integration interfaces that can combine the advantages that RIB and RETL have. From this concept we implemented the new interface thus making the AIP an application better and more attractive to potential customers.

ABSTRACT

Agradecimentos

Ao meu orientador, Professor Raul Vidal, por ter acreditado que eu conseguiria realizar esta dissertação, mesmo depois de todos os problemas encontrados no início da mesma.

Ao meu orientador, Eng^o Rui Pinto, que dedicou-se a fazer com que o projecto proposto pela WIPRO se encaixasse o melhor possível dos requerimentos da dissertação da faculdade. Com este esforço conseguiu-se obter um projecto com bons resultados.

À minha namorada, que várias vezes durante este projecto teve de ouvir-me e que no final desses mesmos monólogos ainda conseguia dar-me grandes conselhos. Agradeço-lhe também toda a ajuda que me deu ao nível da escrita, tanto em Português como Inglês.

Aos meus pais, por me terem colocado numa posição sem pressão para que eu pudesse ter toda a liberdade para tomar qualquer decisão em relação à dissertação. Esta liberdade ajudou-me a continuar.

Ao Pedro Morgado, que sempre me ouviu e partilhou experiências de forma a que conseguíssemos ajudarmo-nos mutuamente.

Aos 4 amigos que trabalham comigo na Wipro, que me ajudaram a manter um sorriso durante grande parte dos dias deste projecto e que fizeram com que o dia-a-dia tivesse momentos fantásticos.

Aos meus amigos de casa, que fizeram com que os momentos extra projecto fossem extremamente agradáveis para que fosse mais fácil continuar o projecto no dia seguinte.

À minha tia, Maria José, por ter investido algum do seu tempo precioso a contribuir para que o resultado final deste projecto fosse melhor, pelo menos ao nível da qualidade de escrita.

AGRADECIMENTOS

Conteúdo

| | |
|---|-------------|
| Resumo | I |
| Abstract | III |
| Agradecimentos | V |
| Conteúdo | VII |
| Lista de Figuras | IX |
| Lista de Tabelas | XI |
| Abreviaturas e Símbolos | XIII |
| Capítulo 1 Introdução | 1 |
| 1.1. Motivação..... | 3 |
| 1.2. Objectivos | 3 |
| 1.3. Estrutura do Documento | 3 |
| Capítulo 2 Mundo Do Retalho | 5 |
| 2.1. Conceitos Básicos | 5 |
| 2.2. Áreas Críticas | 6 |
| 2.3. Retalho e Investigação | 8 |
| Capítulo 3 Gestão da Cadeia de Abastecimento no Mundo do Retalho | 9 |
| 3.1. Conceitos Básicos da Cadeia de Abastecimento..... | 9 |
| 3.2. Trabalhos da Cadeia de Abastecimento | 10 |
| 3.3. O Conceito de Gestão da Cadeia de Abastecimento | 12 |
| 3.4. Desafios da Gestão da Cadeia de Abastecimento | 13 |
| Capítulo 4 Oracle Retail Advanced Inventory Planning | 15 |
| 4.1. Oracle Retail | 15 |

CONTEÚDO

| | |
|--|-----------|
| 4.1.1. Conceito de Interface de uma Aplicação Oracle Retail..... | 16 |
| 4.1.2. Conceito de Ordens de Compra e Transferências no Mundo do Retalho | 17 |
| 4.2. Visão Geral do AIP..... | 17 |
| 4.3. Interação do AIP com Outros Módulos Oracle Retail | 20 |
| 4.3.1. Informação do Módulo RMS | 21 |
| 4.3.2. Informação do Módulo RO | 21 |
| 4.3.3. Informação do Módulo RDF | 21 |
| 4.4. Arquitectura do AIP..... | 22 |
| 4.5. Casos Reais de Implementação do AIP | 25 |
| 4.5.1. Retalhista Britânico | 25 |
| 4.5.2. Retalhista do Texas nos EUA..... | 26 |
| 4.6. Perspectiva de Melhoria à Interação do AIP..... | 26 |
| Capítulo 5 Comparação das Ferramentas de Integração de Aplicações Oracle Retail | 29 |
| 5.1. Oracle Retail Integration Bus | 29 |
| 5.1.1. Descrição do Funcionamento | 31 |
| 5.2. Oracle Retail Extract, Transform and Load..... | 32 |
| 5.2.1. Descrição do Funcionamento | 33 |
| 5.3. PL/SQL com Oracle SQL*Loader..... | 34 |
| 5.3.1. Descrição do Funcionamento | 34 |
| 5.4. Comparação das 3 Ferramentas | 35 |
| 5.4.1. Métodos Utilizados | 35 |
| 5.4.2. Resultado Obtidos | 36 |
| 5.4.3. Conclusões da Comparação das Ferramentas | 42 |
| Capítulo 6 Melhoria à Interação do AIP | 45 |
| Capítulo 7 Conclusões | 51 |
| 7.1. Dificuldades Encontradas | 52 |
| 7.2. Melhorias Possíveis no Trabalho Realizado | 53 |
| Referências | 55 |

Lista de Figuras

| | |
|---|----|
| Figura 3.1 – Exemplo de uma cadeia de abastecimento. | 10 |
| Figura 3.2 – Gráfico representativo dos trabalhos da cadeia de abastecimento. | 11 |
| Figura 4.1 - Processos de negócio do retalho abrangidos pelo Oracle Retail. | 16 |
| Figura 4.2 - Vantagens da sincronização do Fornecimento com a Procura de produtos proporcionado pelo AIP. | 18 |
| Figura 4.3 - Diferentes tipos de nós que possui uma cadeia de abastecimento de um retalhista. | 18 |
| Figura 4.4 - Explicação do porquê do AIP fazer um reaprovisionamento baseado no tempo. | 19 |
| Figura 4.5 - Arquitectura da interacção do AIP com módulos Oracle Retail. | 20 |
| Figura 4.6 - Arquitectura interna do AIP e sua interacção com os módulos Oracle Retail. | 23 |
| Figura 4.7 - Batch diário executado pelo AIP. | 24 |
| Figura 4.8 - Sugestão de melhoria à interacção do AIP. | 27 |
| Figura 5.1 - Diagrama do funcionamento do RIB. | 31 |
| Figura 5.2 - Diagrama de funcionamento do RETL. | 33 |
| Figura 5.3 - Diagrama do funcionamento do PL/SQL com SQL*Loader. | 34 |
| Figura 5.4 – Resultados obtidos na métrica do tempo que a interface, implementada em RIB, demora a passar grandes quantidades de informação. | 38 |
| Figura 5.5 - Resultados obtidos na métrica do tempo que a interface, implementada em RIB, demora a passar pequenas quantidades de informação. | 38 |

LISTA DE FIGURAS

- Figura 5.6 - Resultados obtidos na métrica do tempo que as interfaces, implementadas em RETL e PL/SQL com SQL*Loader, demoram a passar grandes quantidades de informação. 39
- Figura 5.7 - Resultados obtidos na métrica do tempo que as interfaces, implementadas em RETL e PL/SQL com SQL*Loader, demoram a passar pequenas quantidades de informação. 39
- Figura 6.1 – Arquitectura final da integração do AIP melhorada. 49

Lista de Tabelas

| | |
|---|----|
| Tabela 3.1 – Definições das actividades representadas na Figura 3.2..... | 11 |
| Tabela 3.2 – Definições dos membros da cadeia de abastecimento representados na Figura 3.2.. | 11 |
| Tabela 4.1 – Parâmetros para o cálculo do reaprovisionamento das lojas dos retalhistas..... | 22 |
| Tabela 5.1 – Definições dos elementos da Figura 5.1..... | 32 |
| Tabela 5.2 – Resultados obtidos na métrica de custo de desenvolvimento de uma interface de comunicação..... | 36 |
| Tabela 5.3 - Resultados obtidos na métrica de custo de alteração de uma interface de comunicação..... | 37 |
| Tabela 5.4 – Resultados obtidos na métrica do tempo que a interface demora a passar determinada quantidade de informação..... | 37 |
| Tabela 5.5 - Interpretação diferente dos resultados obtidos na Tabela 5.4 (unidades em registos/segundo)..... | 40 |
| Tabela 5.6 – Média dos resultados obtidos na Tabela 5.5..... | 40 |
| Tabela 6.1 – Volume de registos de ordens de compra e de transferência a serem transmitidos... | 46 |

LISTA DE TABELAS

Abreviaturas e Símbolos

| | |
|--------|--|
| AIP | <i>Oracle Retail Advanced Inventory Planning</i> |
| APC RO | <i>Analytic Parameter Calculation for Replenishment Optimization</i> |
| CA | <i>Cadeia de Abastecimento</i> |
| DM | <i>AIP Data Management</i> |
| EAI | <i>Enterprise Application Integration</i> |
| GCA | <i>Gestão da Cadeia de Abastecimento</i> |
| GMROI | <i>Gross Margin Return On Investment</i> |
| MDB | <i>Adaptador de Subscrição do RIB</i> |
| OM | <i>AIP Order Management</i> |
| PO | <i>Ordens de Compra (Purchase Orders)</i> |
| POS | <i>Pontos de Venda Electrónicos (Point Of Sale)</i> |
| RDF | <i>Oracle Retail Demand Forecasting</i> |
| RETL | <i>Oracle Retail Extract, Transform, and Load</i> |
| RIB | <i>Oracle Retail Integration Bus</i> |
| RMS | <i>Oracle Retail Merchandising System</i> |
| RO | <i>Oracle Retail Replenishment Optimization</i> |
| RPAS | <i>Retail Predictive Application Server</i> |
| SI | <i>Sistemas de Informação</i> |
| SRP | <i>AIP Store Replenishment Planning</i> |
| SSB | <i>Adaptador de Publicação do RIB</i> |
| TSF | <i>Transferências (Transfers)</i> |
| WRP | <i>AIP Warehouse Replenishment Planning</i> |
| XML | <i>Extensible Markup Language</i> |

ABREVIATURAS E SÍMBOLOS

Capítulo 1

Introdução

Actualmente o comércio do retalho está presente em todas as economias do mundo e este procura, cada vez mais, novas formas de otimizar os seus processos de negócio, de forma a melhorar o serviço prestado e os seus resultados operacionais.

Neste tipo de comércio, as empresas atingem grandes proporções ao nível da quantidade de lojas, produtos e colaboradores, entre outros. Portanto tornou-se imprescindível recorrer às tecnologias de informação para que tais dimensões sejam praticáveis. Hoje já poucas organizações do mundo do retalho conseguem sobreviver sem o recurso a estas tecnologias, pois elas são o segredo para a manutenção da organização da máquina tão complexa que é a gestão dos processos de negócio de um retalhista. Portanto os retalhistas investem muitos milhões de euros neste tipo de tecnologia.

Os sistemas de informação encontram-se já bastante desenvolvidos e evoluídos nesta área. Esse desenvolvimento foi-se realizando iterativamente, o que se reflecte nas arquitecturas dos mesmos. Normalmente estes encontram-se divididos por módulos, ou seja foram-se desenvolvendo aplicações consoante se descobriam novas maneiras de se melhorar os processos de negócio do retalho. Mas estas aplicações não podem trabalhar isoladamente. Toda a informação gerada no negócio tem de ser comum a todas elas e por isso é fundamental que se encontrem interligadas. Assim a interacção entre as mesmas é essencial e tem de ser bem pensada para a obtenção do máximo de eficiência possível.

Visto que o desenvolvimento deste tipo de aplicações é feito de forma iterativa, recentemente foi criada uma nova para o mercado do retalho. Esta nova aplicação é o AIP, que se insere numa suite de aplicações chamada *Oracle Retail*. Esta tem como objectivo principal melhorar a qualidade dos processos de reaprovisionamento de toda a cadeia de abastecimento, aumentando o número de funcionalidades que dispõe sobre a mesma. Sendo esta uma aplicação bastante recente, apresenta ainda alguns problemas e por isso ainda não é suficientemente

Introdução

apetecível a todos os seus clientes alvo. Só os retalhistas de grande dimensão financeira se arriscam numa aplicação tão recente. Para todos os outros, serão necessárias grandes melhorias, de forma a provar que a aplicação é 100% fiável e com resultados já demonstrados no mercado.

Assim sendo, actualmente estão em andamento muitos estudos sobre o AIP de forma a torná-lo um software mais rigoroso e eficaz. Portanto este é o grande objectivo do presente projecto. Mas dentro do AIP, focar-se-á apenas a sua interacção com outros módulos *Oracle Retail*, dada a brevidade do tempo disponibilizado para a realização do mesmo. A justificação para o foco do trabalho nesta área advém da experiência dos especialistas da área do AIP, que defendem que uma melhoria da interface de comunicação entre o AIP e uma das aplicações do *Oracle Retail* poderia ter um grande impacto ao nível da eficiência do funcionamento mesmo.

Por fim é importante reflectir acerca dos motivos pelos quais a eficiência desta aplicação se torna crucial para os seus clientes. Primeiramente, sendo o seu principal objectivo a gestão de todo o reaprovisionamento da cadeia de abastecimento de um retalhista, a gestão correcta dos tempos de execução dos diversos processos de negócio torna-se o factor mais crítico: todos os prazos têm de ser bem calculados para que tudo esteja “no sítio certo, à hora certa”. Em segundo lugar, está o facto de esta aplicação se basear em previsões, prevendo o quanto se venderá em determinado período para que se saiba quanto se terá de comprar de cada produto. Assim quanto mais eficiente for a aplicação mais garantias terá o retalhista de que tudo correrá dentro dos prazos estipulados, sobrando mais tempo de manobra para a resolução de algum evento inesperado. A maior eficiência do AIP terá também a vantagem de fornecer mais tempo para efectuar previsões, isto é quanto menos tempo o AIP demorar a ser executado mais tempo o retalhista terá para o cálculo de previsões. Ou seja imaginando que o resultado do AIP é necessário a determinada hora, quanto menor for o tempo de execução desta aplicação, mais tarde poderão ser feitas as previsões. E é sabido que, quanto mais em cima do acontecimento forem geradas as previsões, melhores resultados providenciarão. A este raciocínio se dá o nome de “sincronização da procura com o fornecimento de produtos”: aproxima-se o período em que se prevê vender os produtos com o período em que se faz as previsões. Quanto mais próximos estiverem estes dois períodos, melhores resultados operacionais terá o retalhista e consequentemente mais lucros.

Baseado nesse conhecimento, foi realizado um estudo debruçado nesta interacção e nas ferramentas que poderiam ser utilizadas para melhorar tal comunicação.

Os resultados alcançados poderão aumentar o interesse da aplicação perante os seus potenciais clientes.

1.1. Motivação

A motivação para este projecto foi o facto de se estar a trabalhar com um mundo aliciante em termos financeiros. Desenvolver aplicações para o mundo do retalho é sempre entusiasmante devido ao facto de estas serem sempre cobiçadas pelos retalhistas. O retalho é um mundo que movimenta grandes quantidades de dinheiro e como existe grande concorrência neste mundo, os retalhistas andam sempre à procura de investir em novas vantagens competitivas. Assim, os desenvolvimentos para esta área podem ser muito compensatórios.

Como motivação extra, também existiu o facto de dar a conhecer aos intervenientes do projecto um pouco mais sobre o mundo do retalho. É bom conhecermos um pouco do que está por trás das prateleiras, das quais tiramos os bens necessários e ficar a conhecer a máquina complexa que existe para que tudo esteja perfeito quando vamos comprar alguma coisa às lojas. Após este projecto uma ida às compras terá um sabor diferente.

1.2. Objectivos

O Oracle AIP é portanto um dos módulos do complexo sistema de informação que é o *Oracle Retail*. Sobre este pretende-se estudar se será possível melhorar uma interface de comunicação, que segundo especialistas na área parece susceptível de tal.

Assim sendo, como objectivos deste projecto temos:

- Realizar um estudo às ferramentas que podem implementar interfaces de comunicação entre aplicações *Oracle Retail*;
- Após os resultados do estudo anterior, tirar conclusões sobre a possibilidade ou a impossibilidade de serem realizadas melhorias;
- Se for passível de melhorias, o objectivo será implementá-las e explicar de que forma se procedeu para que se criassem mais vantagens competitivas no AIP.

1.3. Estrutura do Documento

Segundo uma visão geral do documento, este parte de um alto nível para o baixo nível em termos de conceitos e estado da arte, até chegar à implementação e resultados.

Portanto ao nível de conceitos e estado da arte, parte-se do Capítulo 2 e irá até ao Capítulo 4. No primeiro destes introduz-se o conceito mais geral deste projecto, o mundo do retalho. Nesta secção explica-se conceitos que para retalhistas são de consenso geral, mas que por parte de alguns leitores pode não ser.

Introdução

Posteriormente fala-se sobre uma das áreas do retalho que é a área onde o AIP se insere, gestão da cadeia de abastecimento. Aqui falam-se de conceitos importantes para que se consiga perceber o funcionamento desta aplicação.

No Capítulo 4 fala-se do AIP em si. Aqui explica-se o seu funcionamento, a sua arquitectura, a sua utilidade, entre outras coisas. É importante de referir que neste Capítulo se indica sobre qual interface se irá debruçar este projecto.

Após esta introdução de conceitos, apresenta-se o estudo às ferramentas que podem implementar interfaces de comunicação entre aplicações *Oracle Retail*. Neste estudo apresentam-se os métodos utilizados e os resultados obtidos, tirando-se no final uma conclusão baseada nesse estudo.

No Capítulo 6, apresenta-se então a proposta de melhoria a realizar ao AIP. Que trabalho foi feito, como se conseguiu melhorar a interface de comunicação e que vantagens tem a nova implementação.

Por fim, apresentam-se as conclusões deste projecto. Ou seja, qual o saldo final do trabalho realizado.

Capítulo 2

Mundo Do Retalho

2.1. Conceitos Básicos

O retalho é o conceito de vender produtos que sejam para consumo final dos clientes. Isto é, os retalhistas são empresas que compram produtos em grandes quantidades para posteriormente venderem à unidade ou em quantidades mais pequenas [1].

Este tipo de negócio teve início há milhares de anos atrás com vendedores ambulantes a apregoarem as suas mercadorias pelos mercados fora. Actualmente este é bastante competitivo e as taxas de insucesso dos estabelecimentos de retalho é relativamente grande. O preço é a maior área de competição entre os retalhistas, mas existem outras características diferenciadoras como a localização das lojas, a selecção e exibição dos seus produtos, a atractividade das lojas e ainda a sua reputação. Os canais de distribuição do retalho atingem agora vários métodos como máquinas de venda automáticas, vendas porta a porta, vendas por telefone, marketing por endereço de e-mail, internet, lojas de desconto, lojas especializadas, supermercados e cooperativas.

As empresas de retalho podem variar bastante entre si relativamente à sua dimensão, por exemplo podem ir desde a maior do mundo, como a *Wal-Mart*, que vende quase todo o tipo de produtos até uma pequena farmácia gerida por uma família.

A frase que caracteriza o ideal do retalho é:

“Oferecer o **produto certo**, no **local certo**, no **momento certo** pelo **preço certo**.”

Marshall L. Fisher, Ananth Raman and Anna Sheen McClelland

Este ideal continua ainda bastante inatingível para grande parte dos retalhistas, apesar de existir já uma elite destes que se começa a aproximar bastante do mesmo [2].

O retalho é um negócio que move muito dinheiro. Por exemplo nos EUA durante o mês de Março de 2010, o negócio do retalho facturou cerca de 363 200 Milhões de Dólares, cerca de 286 700 Milhões de Euros. O que daria anualmente 3,5 Biliões de Euros [3]. Além de ser um negócio

que move Biliões, ainda assim é uma indústria muito competitiva tanto internamente (dentro do mesmo país) como globalmente. Portanto cada vez mais os retalhistas se expandem para o mundo de forma a aumentar o seu público-alvo.

Devido à dimensão deste negócio, muitas outras empresas o acham atractivo para que possam fazer dele a sua área de negócio. No contexto deste projecto, o negócio que utiliza o retalho como seu cliente é o negócio dos sistemas de informação (SI). Estes sistemas têm sido a ferramenta utilizada pelos retalhistas para que se consigam aproximar o mais possível do ideal do retalho.

Para que se possam diferenciar entre eles, os retalhistas tentam criar vantagens competitivas a partir dos SI e para tal investem muitos Milhões de Euros em empresas que os desenvolvam. Além disso, existem ainda empresas que integram nos seus próprios quadros de pessoal equipas especializadas para o desenvolvimento destes sistemas.

2.2. Áreas Críticas

No mundo do retalho existem quatro áreas críticas ao qual os retalhistas dão bastante atenção. Visto serem as áreas mais importantes deste mundo, os investigadores que desenvolvem soluções para o negócio do retalho têm dedicado mais tempo e dinheiro às mesmas. O que leva a que sejam as áreas que apresentam mais sistemas de informação capazes de as abranger. Estas áreas, segundo Marshall L. Fisher, Ananth Raman e Anna Sheen McClelland [2], são:

- Exactidão e disponibilidade de informação do negócio;
- Gestão do ciclo de vida da cadeia de abastecimento (CA);
- Previsão das vendas;
- Planeamento do inventário de toda a CA.

Na primeira área, os retalhistas utilizam normalmente pontos de venda electrónicos (POS) que registam as vendas que são efectuadas. Mas apesar de estes sistemas melhorarem bastante a exactidão e a disponibilidade da informação do negócio, existem mesmo assim problemas com a exactidão dessa informação. Normalmente a falta de precisão que mesmo assim existe provêm das devoluções, por exemplo, quando um cliente compra uma camisola de tamanho M e volta à loja para trocar a mesma pelo S. Como o empregado não quer incomodar o cliente, então não perde tempo a passar ambas as camisolas pelo POS, indicando ao sistema que voltou a entrar um M e a sair um S, logo perde-se a precisão do stock de cada um dos produtos. Um exemplo comum também é o facto de o retalhista comprar uma determinada quantidade de carne, que depois de partida é vendida ao cliente sem ossos e sem gorduras, logo a quantidade de carne que foi comprada é diferente da que foi vendida, tornando-se assim também num problema de exactidão dos dados. A aplicação Oracle que abrange esta área é o *Oracle Retail Merchandising System*

(RMS). No entanto é de salientar que esta aplicação já teve estudos e desenvolvimentos para que se conseguisse minimizar este problema de exactidão e para tal a mesma apresenta já algumas soluções bastante fiáveis.

A segunda área é importante porque os retalhistas pretendem sempre que o *lead-time*¹ da sua cadeia de abastecimento para cada produto seja o mais curto possível. Quanto mais curto for este tempo, mais tempo estes têm para prever bem as suas vendas, antes de entregarem ordens de compra aos seus fornecedores. Isto é quanto mais curto o *lead-time* mais tarde os retalhistas podem entregar as ordens de compra, de forma a terem o produto no dia desejado. Ou seja aproximam a necessidade do cliente à necessidade de envio das ordens de compra aos fornecedores, isto é melhoram a sua resposta perante o mercado. Portanto gerir bem o ciclo de vida da CA torna-se tão importante. Quanto menor for este ciclo de vida, mais flexibilidade e reactividade terá a CA. Ou seja responderá muito melhor a alterações não planeadas.

A terceira área é importante porque os retalhistas necessitam cada vez mais de melhores algoritmos e técnicas de prever as suas vendas. A quantidade de vendas que podem ser realizadas por um retalhista, num determinado dia, pode variar segundo bastantes factores (condições meteorológicas, estação do ano, festas ou eventos especiais, etc.). É portanto muito importante desenvolver técnicas que possam ser as mais exactas possíveis no cálculo destas previsões. Esta é uma área complicada para se encontrar soluções perfeitas. Actualmente a dimensão e as condições do mercado levam a que este seja bastante instável e de difícil previsão. O simples facto de existir muita variedade dos mesmos produtos leva a que ainda se torne mais difícil fazê-lo. Porque por exemplo seria mais fácil prever a quantidade de água que se venderia num determinado dia, do que a quantidade de cada uma das marcas de água. Para este caso, a aplicação que a Oracle possui é o *Oracle Retail Demand Forecast (RDF)*. Esta aplicação usufrui de bastantes algoritmos que são adequados a vários tipos de características de produtos.

Por fim e dentro do âmbito deste projecto, o planeamento de inventário é também fundamental para que os retalhistas consigam gerir todos os stocks ao longo da sua CA. A gestão dos stocks por toda a CA é fundamental para que nunca haja falta de produtos nos locais de venda ao cliente final. Nesta área joga-se com toda a informação gerada pelos sistemas anteriores para que o planeamento possa ser o mais preciso possível. Reunindo toda a informação anterior faz-se o planeamento das quantidades de stocks necessárias em cada ponto da CA. Ou seja segundo as previsões de vendas geradas, mais a informação dos stocks em cada ponto da CA e os seus *lead-times* consegue-se gerar o planeamento de inventário. Para esta área a Oracle lançou agora um novo produto que trará grandes melhorias em relação ao planeamento anteriormente realizado

¹ No mundo do retalho, o *lead-time* é o tempo desde que o retalhista entrega uma ordem de compra a um fornecedor até que recebe o produto no local pretendido.

pelo RMS. Esta aplicação tem o nome *Oracle Retail Advanced Inventory Planning* (AIP). Será sobre esta aplicação que será feito este projecto.

Em suma, todas estas áreas têm como principal intuito diminuir o máximo possível o excesso e/ou a falta de stock dos produtos de um retalhista. Ou seja tentar coincidir as necessidades dos clientes com a compra de produtos aos fornecedores. Assim boas soluções para as 3 primeiras áreas farão com que o planeamento de inventário possa ser melhor de forma a trazer mais vantagens competitivas aos retalhistas. É importante referir, que a última área tem sido onde tem havido mais investigação ultimamente. Segundo os investigadores é aqui que actualmente se pode criar mais vantagens competitivas aos clientes destes SI [4].

2.3. Retalho e Investigação

O mundo do retalho sendo uma indústria que move muito dinheiro todos os anos (valores referidos na secção 2.1), bastante competitiva e onde cada retalhista começa a estar presente globalmente emerge assim como uma escolha fascinante para os investigadores na área da gestão da cadeia de abastecimento (GCA). Ou seja a maioria dos estudos sobre CA são feitos sobre as CA do retalho, visto que bons resultados destes estudos podem levar a grandes lucros, por parte dos investigadores. Este tipo de indústria apresenta um vasto número de desafios estimulantes na área das operações e do planeamento de inventário. Portanto os avanços das capacidades computacionais e das tecnologias de informação, a hiper-competitividade nesta indústria, o aparecimento de múltiplas formas de retalho e canais de distribuição, uma tendência crescente para uma rede global de retalho e uma melhor compreensão da importância da colaboração de todos os elementos de uma CA alargada levaram a um aumento da investigação sobre tópicos de GCA do retalho [4].

Consequentemente toda a investigação permitiu que hoje os retalhistas tenham ao seu dispor uma variedade de produtos comerciais que efectuem previsão de vendas, optimização de preços, planeamento de inventário, entre outras e que são capazes de se integrar nos seus sistemas de informação.

Apesar de o retalho ser a indústria mais interessante a ser estudada pela comunidade científica, pelos motivos previamente referidos, é obviamente objectivo desta última, criar soluções genéricas que possam ser aplicadas a qualquer cadeia de abastecimento.

Em suma, toda esta investigação ajudou a que fosse possível criar a aplicação sobre o qual este projecto se vai centrar, o AIP.

Capítulo 3

Gestão da Cadeia de Abastecimento no Mundo do Retalho

A frase gestão da cadeia de abastecimento (GCA) carece de uma definição clara. Uma revisão da literatura revela que esta se tornou uma expressão utilizada em negócios para incluir tudo dentro do campo do marketing e da logística. Para dar um exemplo: Ross considerou que a GCA não é menos que um método, um conceito, uma filosofia, um sistema, um processo, uma estratégia e um estado de espírito [5]. Com esta representação abrangente é difícil entender o que é realmente a GCA, portanto durante este capítulo iremos ver vários elementos que constituem esta área.

A GCA é uma área importantíssima dentro do mundo do retalho. Será dentro desta área que o projecto se irá focar. Então para isso será introduzida informação útil para melhor compreensão da mesma. Também é importante referir que a GCA é a área sobre a qual o AIP se debruça ou seja, esta aplicação tem como objectivo levar a GCA ao mais alto nível.

3.1. Conceitos Básicos da Cadeia de Abastecimento

A cadeia de abastecimento (CA) é um sistema de organizações, pessoas, tecnologias, actividades e recursos dedicados a fazer mover produtos ou serviços desde fornecedores e

retalhistas até aos seus clientes. Normalmente uma CA é representada por um grafo dirigido², que possui vários tipos de nós com designações diferentes. No caso do retalho, é normal encontrarmos uma CA parecida com a que se pode ver na Figura 3.1.

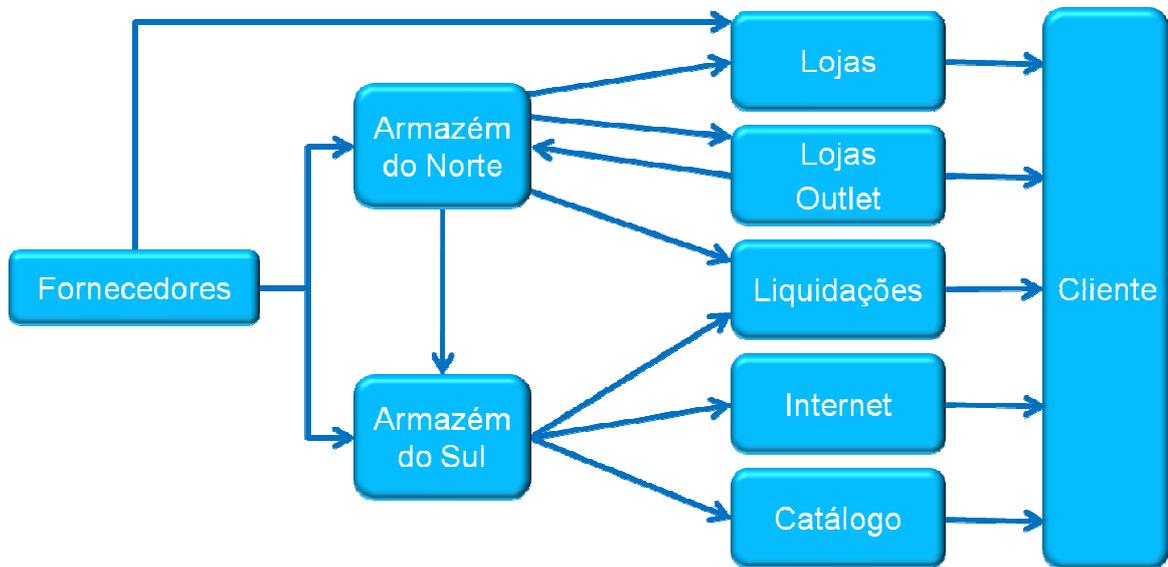


Figura 3.1 – Exemplo de uma cadeia de abastecimento.

Nesta Figura, existem 4 tipos de nós: os fornecedores, os armazéns, os pontos de venda e o cliente final. Depois existem as ligações dirigidas que indicam em que sentido os produtos podem viajar desde a sua origem até ao cliente. Uma cadeia de abastecimento tem de estar bem definida para que se possa fazer uma boa gestão da mesma.

3.2. Trabalhos da Cadeia de Abastecimento

Os trabalhos da cadeia de abastecimento podem ser representados por um gráfico desenvolvido por Stern, segundo fluxos de marketing. Nove fluxos entre os membros de um canal de distribuição estão identificados na Figura 3.2. Existem fluxos que se movem exclusivamente para a frente (physical, ownership e promotion), outros só se movem para trás (ordering e payment) enquanto os outros movem em ambos os sentidos (negotiation, financing, risking e information).

² Estruturas matemáticas utilizadas para modelar relações com uma certa direcção entre objectos. Um grafo é uma colecção de nós e ligações que ligam pares de nós. Num grafo dirigido essas ligações possuem um sentido no qual se indica a direcção do movimento, neste caso o movimento dos produtos.

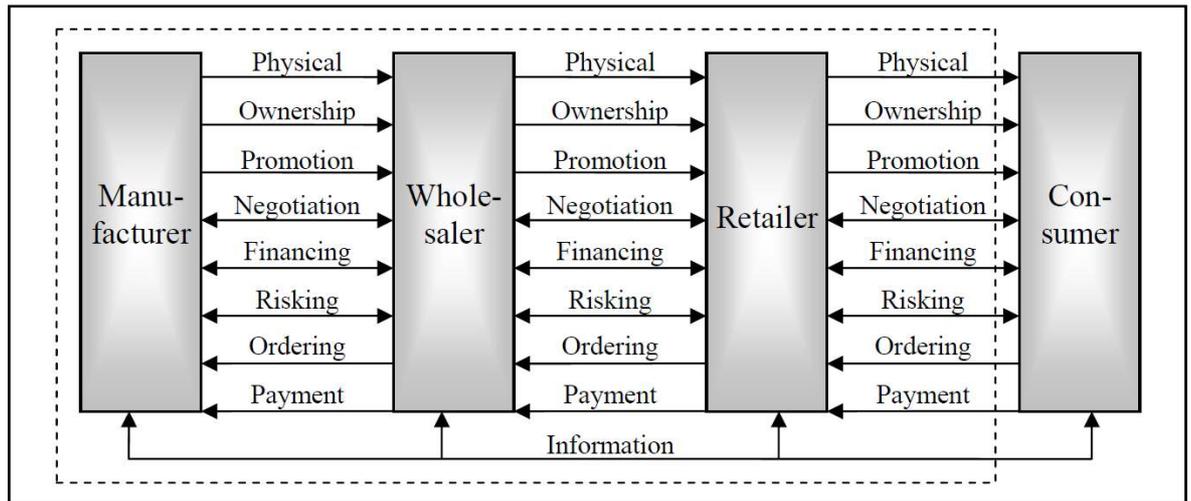


Figura 3.2 – Gráfico representativo dos trabalhos da cadeia de abastecimento.

Com o intuito de se tornar mais explícito o conteúdo da imagem anterior, apresentam-se definições dos conceitos da Figura 3.2 nas tabelas Tabela 3.1 e Tabela 3.2. A primeira tabela apresenta as definições das actividades e a segunda tabela apresenta as definições dos membros da cadeia de abastecimento.

| Actividade | Significado |
|-------------|--------------------------|
| Physical | Produtos manufacturados. |
| Ownership | Direitos de propriedade. |
| Promotion | Promoções. |
| Negotiation | Negociação. |
| Financing | Regras de financiamento. |
| Risking | Risco. |
| Ordering | Encomendas. |
| Payment | Pagamentos. |

Tabela 3.1 – Definições das actividades representadas na Figura 3.2.

| Membros da cadeia de abastecimento | Significado |
|------------------------------------|-------------|
| Manufacturer | Fornecedor |
| Wholesaler | Revendedor |
| Retailer | Retalhista |
| Consumer | Cliente |

Tabela 3.2 – Definições dos membros da cadeia de abastecimento representados na Figura 3.2.

Assim a passagem dos produtos manufacturados são sempre no sentido dos consumidores na CA, tanto como os direitos de propriedade dos mesmos e como as promoções. Um retalhista faz uma promoção usualmente se o fornecedor tiver feito também uma promoção a si. Negociação, financiamento e risco são partilhados e definidos entre os vários elementos da CA. Geração de encomendas e os pagamentos são realizados sempre no sentido dos fornecedores ou seja, o retalhista realiza estas actividades junto dos fornecedores. As actividades da Figura 3.2 têm de ser sincronizadas com a procura do segmento de mercado a que se destinam. Ou seja o mesmo objectivo de qualquer CA.

As preferências dos consumidores, segundo Stern baseiam-se nas quatro seguintes características:

- Oportunidade destes comprarem os seus produtos em pequenas quantidades, não os obrigando a adquirirem inventário desnecessário;
- As lojas dos retalhistas estarem perto dos mesmos, diminuindo assim os custos que estes têm a transportar os seus bens;
- Tempo que o retalhista demora a entregar os seus produtos (aplicado mais a retalhistas electrónicos);
- Por fim, a disponibilidade e variedade dos produtos.

Se todas estas características forem iguais entre 2 retalhistas diferentes, os clientes irão escolher o que apresenta melhor qualidade no serviço. Mas isto normalmente não acontece, porque quanto melhor é o serviço prestado pelo retalhista, maior terão de ser os preços dos seus produtos. Por isso, o preço torna-se também um factor de diferenciação dos retalhistas [6].

3.3. O Conceito de Gestão da Cadeia de Abastecimento

O termo gestão da cadeia de abastecimento (GCA) foi introduzido por consultores nos meados de 1980 e foi adoptado pela comunidade académica no final da década [7]. Após essa altura, a confusão entre logística e a GCA tem sido imensa. Alguns acreditam que são sinónimos, outros acreditam que existem realmente diferenças. Stern define logística como gestão do fluxo de materiais físicos [6], que posteriormente sofreu uma metamorfose para o termo GCA. Christopher tem um entendimento parecido sobre o assunto, acrescentando que a GCA é uma extensão à logística, ou seja, além da logística aborda também a lógica da logística [8].

O Council of Supply Chain Management Professionals (CSCMP), instituição que estuda a GCA, define a gestão da logística como a parte dos processos da CA que planeia, implementa e controla a eficiência e a eficácia na movimentação dos fluxos de bens armazenados, serviços e informação gerada desde o ponto de origem dos produtos até ao ponto de consumo. As actividades de gestão logística geralmente incluem a gestão da entrada e saída de transportes com

produtos, gestão da frota, armazenamento, manuseamento de materiais, atendimento de pedidos, design da rede de logística, gestão de stock, planeamento do abastecimento e gestão dos prestadores de serviços subcontratados. A diferentes níveis, a gestão logística inclui também fornecedores e contractos, planeamento de produção, empacotamento e montagem e atendimento ao cliente. Esta área está envolvida a todos os níveis do planeamento e execução estratégica, operacional e tática. A gestão logística é uma função integradora, que coordena e otimiza todas as actividades logísticas, bem como marketing, vendas, manufacturação, finanças e tecnologias de informação [9].

Segundo a mesma instituição, por outro lado a GCA significa o planeamento e a gestão de todas as actividades envolvidas tanto na aquisição de produtos e adjudicação de contractos, como conversão e como todas as actividades de gestão logística. Importante também é o facto de incluir tanto a coordenação e colaboração de todos os parceiros por toda a CA, que podem ser fornecedores, intermediários, serviços subcontratados, revendedores e clientes. Na sua essência a GCA é considerada mais abrangente do que a gestão da logística porque integra não só materiais, valores e fluxos de informação, mas também todos os processos chave do negócio de todas as empresas da CA. A GCA integra a gestão da procura dos clientes com a gestão do fornecimento dos produtos e coordena processos e actividades através de funções como design do produto, manufacturação, marketing e vendas [9].

Por fim, os advogados consideram a GCA mais do que um novo nome para gestão da logística. A implementação da primeira implica identificar os membros importantes da cadeia ao qual é necessário se criar uma ligação, escolher que processos necessitam de ser ligados a esse membro e qual é o nível de integração necessário. A integração dos processos têm o objectivo de aumentar a eficiência e a eficácia em todos os membros da CA [10].

3.4. Desafios da Gestão da Cadeia de Abastecimento

A proximidade existente com o mercado dos consumidores dá aos retalhistas alguma informação rápida e precisa sobre questões como mudança nas preferências da moda e atractividade da oferta dos concorrentes.

Mesmo assim os retalhistas enfrentam no seu dia-a-dia imenso desafios. Desafios como diminuição do tempo de colocação dos produtos no mercado, visto que cada vez mais o tempo de vida destes é menor, aumento da variedade de produtos, que faz com que a previsão da procura dos produtos seja menos exacta e também diminuir cada vez mais os *lead-time* (ver explicação na nota de rodapé 1, na página 7) da cadeia de abastecimento [11].

Para tentar resolver tais desafios tem-se investido e estudado muito a área da GCA. No entanto não existe a melhor forma de gestão da cadeia de abastecimento. A maneira como os

retalhistas competem nos mercados de consumo deve definir a forma como estes se devem focar na GCA. Como nenhum retalhista pode ser tudo para todos os clientes então cria-se uma interdependência entre estes definindo assim o que determinada empresa é para o seu cliente.

Um pré-requisito para a sustentabilidade de um retalhista é haver uma relação muito bem definida entre o que é oferecido aos clientes e as actividades da CA. Não é suficiente ter muito conhecimento sobre as preferências dos competidores e dos clientes para se ter um bom desempenho. Disponibilizar os produtos de uma forma desorganizada e ineficiente irá dificultar a principal tarefa dos retalhistas, ter lucro. Isto é evidente com o desaparecimento de muitos retalhistas em que o seu canal de distribuição era exclusivamente a internet. Este tipo de retalhistas tinham pouca experiência na área logística, devido ao facto de não terem lojas físicas e por isso esta área falhava nas suas empresas, o que as levava à ruína. Os retalhistas electrónicos que ainda resistem, já existiam como retalhistas tradicionais e adicionaram só mais um canal de distribuição à sua empresa. Por isso, um conhecimento profundo no que toca ao tipo de produtos, padrões de procura, variedade de produtos, qualidade dos serviços e localização estão relacionados com o trabalho da CA e são cruciais a esta.

Capítulo 4

Oracle Retail Advanced Inventory Planning

A aplicação *Oracle Retail Advanced Inventory Planning* (AIP), como anteriormente referido, será o âmbito deste projecto. Então como esta aplicação é ainda bastante recente no mercado ainda é alvo de muitas tentativas de melhoria. Portanto, o objectivo genérico para o projecto será tentar melhorar esta aplicação para que esta possa ser mais aliciante para os seus clientes.

O *Oracle Retail AIP*, como o nome indica, é uma aplicação que se insere num contexto de aplicações que contemplam o *Oracle Retail*. Este último define-se por um conjunto de aplicações com o intuito de perseguir o objectivo de fazer com que os retalhistas atinjam a excelência dos seus métodos operacionais [12]. Deste modo, começa-se por explicar um pouco do mundo onde o AIP se insere. Posteriormente, explicar-se-á como esta aplicação se integra com os módulos do *Oracle Retail*, seguindo-se uma passagem pela sua arquitectura interna. No final, será explicitada a melhoria que se prevê ser possível fazer à interacção do *Oracle Retail AIP*.

4.1. Oracle Retail

O mundo *Oracle Retail* é um conjunto de aplicações desenvolvidas pela empresa *Oracle* com o objectivo de serem soluções para muitas áreas do retalho, principalmente as suas áreas críticas. Este mundo apresenta uma arquitectura que é facilmente interpretada se pensarmos nela como o trabalho de pequenas máquinas, cada uma com a sua tarefa, trabalhando todas para o

mesmo objectivo. Com isto quer-se dizer que este mundo está dividido essencialmente por módulos, cada um com a sua função específica dentro do mundo do retalho. Actualmente é composto por mais de 60 módulos onde nem todos são exclusivos do retalho mas que não deixam de ser essenciais a este [13]. Assim todos estes componentes do *Oracle Retail* têm como objectivo principal a colaboração de informação e processos entre os ramos de trabalho de um retalhista, representados na Figura 4.1.

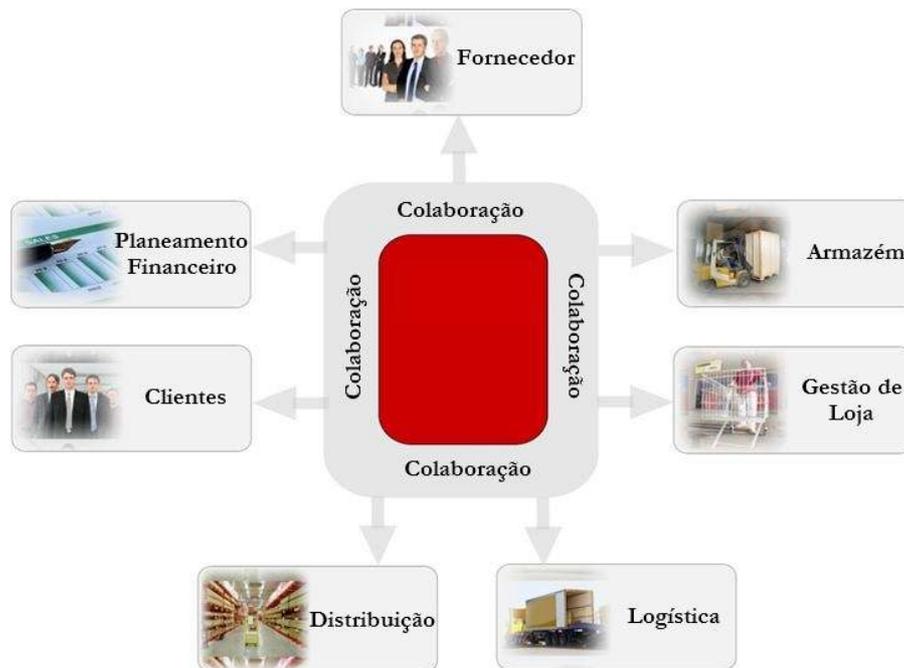


Figura 4.1 - Processos de negócio do retalho abrangidos pelo Oracle Retail.

Portanto, a *Oracle* sempre teve atenção a todos estes factores no negócio do retalho tendo assim aplicações para gerir cada uma das áreas.

Mas para que se consiga entender melhor o AIP é muito importante introduzir alguns conceitos adquiridos no mundo do retalho. Assim apresenta-se seguidamente o conceito de interface, de ordens de compra e de transferências.

4.1.1. Conceito de Interface de uma Aplicação Oracle Retail

No mundo do retalho a comunicação entre as diferentes aplicações das diferentes áreas é muito importante para que tudo funcione. Logo, a informação gerada em cada sistema tem de ser partilhada com os demais e para tal utilizam-se interfaces.

Uma interface é um mecanismo que existe nas demais aplicações do *Oracle Retail* e que tem como objectivo recolher informação do ponto de origem, tratá-la devidamente, colocando-a na aplicação de destino.

Basicamente o conceito aproxima-se bastante a conceitos de outras áreas, tal como um meio capaz de promover a comunicação ou interação entre dois ou mais sistemas [14].

É importante dizer-se que no mundo *Oracle Retail* existem duas ferramentas dedicadas à implementação destas interfaces, que são o RETL e o RIB.

4.1.2. Conceito de Ordens de Compra e Transferências no Mundo do Retalho

Ordens de compra e transferências são conceitos adquiridos no mundo do retalho. E portanto para qualquer retalhista a sua definição é óbvia. Neste mundo, designam-se ordens de compra por PO e transferências por TSF.

Assim uma ordem de compra (PO) é um contracto escrito de venda, entre comprador e vendedor, detalhando exactamente os produtos ou serviços a serem prestados por um único vendedor. Este contracto também especifica os termos de pagamento, as datas de entrega, a identificação dos itens, quantidades dos mesmos, termos de envio e outras condições/obrigações.

Uma ordem de compra é normalmente um documento numerado gerado pelo sistema de gestão financeira de um retalhista que regista os detalhes de uma compra e que regista que o pagamento será feito [15].

Transferências (TSF) assemelham-se às ordens de compra em tudo excepto no facto de serem contractos em que ambas as partes são elementos da mesma empresa. Ou seja no caso das PO é um contracto entre retalhistas e os seus fornecedores, enquanto uma TSF pode ser um contracto entre um armazém e uma loja, do mesmo retalhista. Basicamente é um documento que regista a passagem dos produtos entre os determinados pontos da cadeia de abastecimento.

4.2. Visão Geral do AIP

O *Oracle AIP* é uma solução empresarial líder de mercado desenhada para retalhistas que têm de gerir constantemente o reaprovisionamento do seu inventário. Esta solução intuitiva e colaborativa assegura a sincronização das encomendas realizadas aos fornecedores com a procura dos produtos por parte dos clientes.

Uma frase de um dos *Senior Specialist* da *Wipro Technologies*, que caracteriza muito bem o AIP é:

“Fazer **Mais** com **Menos**, ou seja tornar o inventário mais produtivo.”

Kannan Ravi

Um bom exemplo das vantagens que surgem da sincronização entre a procura e o fornecimento dos produtos de um retalhista pode ser visto na Figura 4.2.

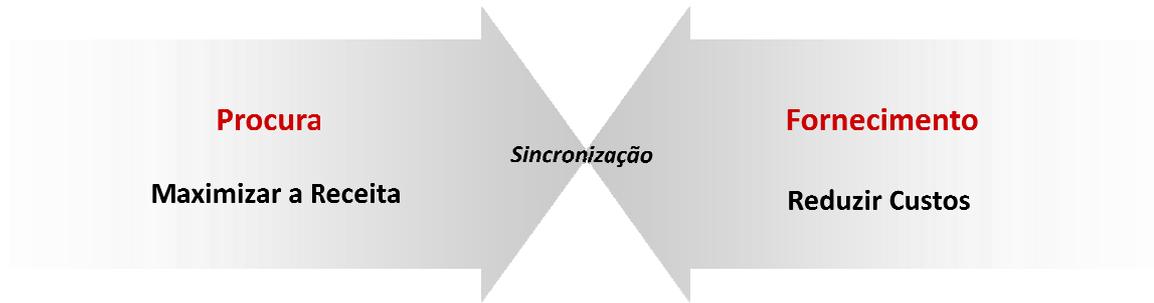


Figura 4.2 - Vantagens da sincronização do Fornecimento com a Procura de produtos proporcionado pelo AIP.

É então objectivo do AIP sincronizar a procura com o fornecimento para que os retalhistas possam maximizar a receita reduzindo custos.

A receita pode ser maximizada nos seguintes pontos:

- Melhorar as percentagens de entrada de stock;
- Optimizar as posições do inventário;
- Melhorar o nível dos serviços prestados;
- Mais promoções eficazes;
- Redução da perda de vendas.

Os custos podem ser reduzidos em:

- Inventário;
- Transportes;
- Armazenamento;
- Recursos Humanos;
- Manuseamento.

Mais detalhadamente, descreve-se as vantagens que esta aplicação proporciona em cada um dos diferentes pontos da cadeia de abastecimento (CA). Estes pontos da CA podem ser vistos na Figura 4.3.

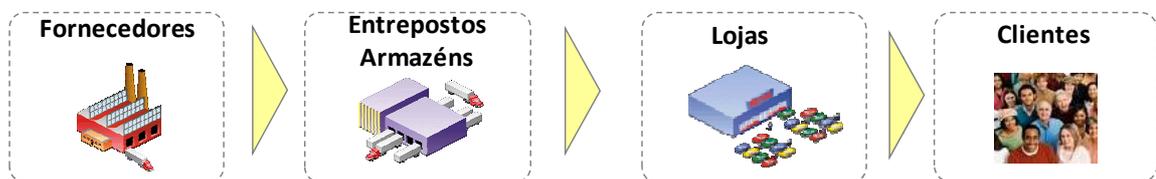


Figura 4.3 - Diferentes tipos de nós que possui uma cadeia de abastecimento de um retalhista.

Fornecedores:

- Melhor colaboração entre o retalhista e os fornecedores;
- Obtenção de informação mais eficiente e precisa sobre os produtos;
- Melhor gestão de promoções.

Entrepostos/Armazéns:

- Reaprovisionamento automático e otimizado;
- Otimização da cadeia de abastecimento dentro das restrições;
- Melhor percepção do inventário e do plano futuro de ordens de compra.

Lojas:

- Melhor precisão do inventário;
- Reaprovisionamento automático e otimizado;
- Reduzir perdas de produtos e produtos sem stock;
- Otimização dos processos dentro da loja.

Cliente:

- Atrair e manter clientes leais;
- Melhor disponibilidade de produtos;
- Otimização de preços segundo a procura dos produtos.

Uma das características mais importante de AIP é fazer o reaprovisionamento baseando-se nos *lead-times* (ver explicação na nota de rodapé 1, na página 7) da cadeia de abastecimento. Isto é facilmente percebido analisando a Figura 4.4.



Figura 4.4 - Explicação do porquê do AIP fazer um reaprovisionamento baseado no tempo.

Ou seja esta aplicação tem a capacidade de fazer os cálculos do reaprovisionamento, tendo atenção que uma ordem de compra tem de chegar ao fornecedor uma quantidade de dias antes ao dia em que o produto é necessário na loja. Neste caso se o produto é necessário dia 15/1 na loja, a ordem de compra tem de chegar ao fornecedor dia 5/1, visto que o produto ainda tem de passar por armazéns e entrepostos e tudo isso leva tempo a ser realizado.

Assim esta torna-se a característica mais diferenciadora desta aplicação em relação ao antigo módulo do RMS, que geria o reaprovisionamento. Isto faz com que o AIP seja baseado no tempo ou seja, nos *lead-times* da cadeia de abastecimento (CA).

No entanto este produto *Oracle* não consegue funcionar como um produto independente. Ele necessita que os seus clientes possuam outras aplicações *Oracle Retail* para que este funcione. Isto é estritamente necessário porque o AIP capta toda a informação necessária aos seus cálculos, a partir dos outros sistemas.

Como o AIP é uma aplicação bastante grande seria extremamente complicado analisa-lo em toda a sua dimensão. Assim ficou definido que dentro do AIP este projecto focar-se-ia nas melhorias possíveis à sua interacção com outros módulos *Oracle Retail*. Assim sendo, explicar-se-á esta interacção que existe com esses módulos.

4.3. Interacção do AIP com Outros Módulos Oracle Retail

O módulo AIP para conseguir gerar os resultados a que se propõe necessita de informação para trabalhar. Por isso e como supra-referido é necessário integrar este com certos módulos do *Oracle Retail*. Após algum estudo da documentação da aplicação [16], muito resumidamente, consegue-se perceber que esta necessita de informação como:

1. Lojas que o retalhista possui e suas características;
2. Armazéns que o retalhista possui e suas características;
3. Itens que definem o portfolio de produtos do retalhista e suas características;
4. Stock do portfolio de produtos nas várias lojas e armazéns;
5. Regras de optimização de inventário;
6. Previsões de vendas;

Todos estes elementos já se encontram definidos nos diversos módulos *Oracle Retail*. Portanto é necessário existir ligações apropriadas para que o AIP consiga obter essas informações. Assim resumidamente mostra-se a integração do AIP na Figura 4.5.

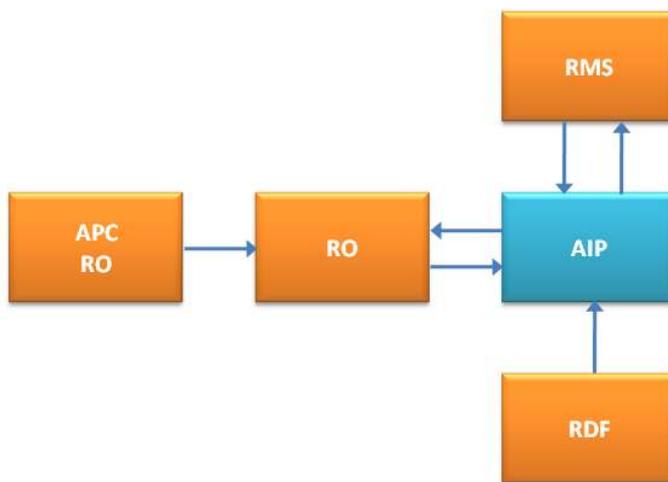


Figura 4.5 - Arquitectura da interacção do AIP com módulos Oracle Retail.

Esta é a arquitectura da interacção do AIP com os módulos *Oracle Retail Merchandising System* (RMS), *Oracle Retail Replenishment Optimization* (RO) e *Oracle Retail Demand Forecasting* (RDF). A explicação para o facto de a interacção ser feita com estes 3 módulos deve-se ao facto de a informação necessária ao AIP estar presentes nestes. Mais especificamente as

informações 1, 2, 3 e 4, definidas anteriormente, estão presentes no RMS. A informação 4 está presente no RO e a 5 está no RDF.

4.3.1. Informação do Módulo RMS

Mais concretamente, a informação que terá de viajar entre o AIP e o RMS tem que conter:

- Ordens de compra fechadas com os fornecedores;
- Itens do portfolio de produtos;
- Lista de fornecedores dos itens;
- Lista de armazéns;
- Lista de lojas;
- Locais das lojas;
- Locais dos armazéns;
- Locais dos fornecedores;
- Inventário corrente das lojas;
- Vida dos produtos das lojas;
- Itens substitutos de outros itens;
- Inventário corrente dos armazéns.

4.3.2. Informação do Módulo RO

A informação enviada do RO para o AIP é composta pelos parâmetros de reaprovisionamento dos vários produtos. Esses parâmetros devem ser definidos pelos retalhistas segundo as características dos produtos, como se pode consultar na Tabela 4.1.

4.3.3. Informação do Módulo RDF

De forma mais específica, o RDF terá de fornecer, ao AIP, a seguinte informação:

- Desvio em relação às previsões diárias standard para uma loja;
- Desvio em relação às previsões semanais standard para uma loja;
- Previsão diária para uma loja;
- Previsão semanal para uma loja.

| Características do produto | Parâmetros |
|--|---------------------------------------|
| Grande ciclo de vida, venda lenta de produtos estáveis (ex: casa, lazer, baterias, itens básicos sazonais) | Min / Max |
| Curto ciclo de vida, produtos frescos do dia | USA, Allocation |
| Vendas espaçadas (ex: óleos especiais, martelos, ...) | Min/Max, Poisson |
| Vendas robustas, grande cariz sazonal e promoções (ex: cereais, itens de natal) | Hybrid, Dynamic |
| Novidades (ex: novos sabores de batatas fritas, novidades de marcas) | Time Supply, Hybrid |
| Itens sem disponibilidade de previsões e produtos novos | Min/Max, Sales Date Range, Factor AWS |
| Itens caros (ex: bebidas espirituosas caras) | Min/Max |

Tabela 4.1 – Parâmetros para o cálculo do reaprovisionamento das lojas dos retalhistas

4.4. Arquitectura do AIP

Anteriormente viu-se como está pensada a interacção do AIP com módulos do *Oracle Retail*. Mas para que se consiga definir bem essa comunicação é necessário estudar bem a arquitectura interna do AIP.

Assim a arquitectura interna desta aplicação, tal como a do *Oracle Retail*, é composta por pequenos módulos onde cada um deles tem um objectivo no cálculo dos resultados da mesma. Esta é composta por 4 módulos distintos. É importante referir que a comunicação referida no capítulo anterior tem uma direcção específica relativamente ao módulo interno do AIP, ao qual se destina. Assim no diagrama seguinte podemos ver mais concretamente a interacção entre os módulos *Oracle Retail* e os módulos que compõe esta aplicação. Portanto, o diagrama da arquitectura interna do AIP pode ser visualizado na Figura 4.6.

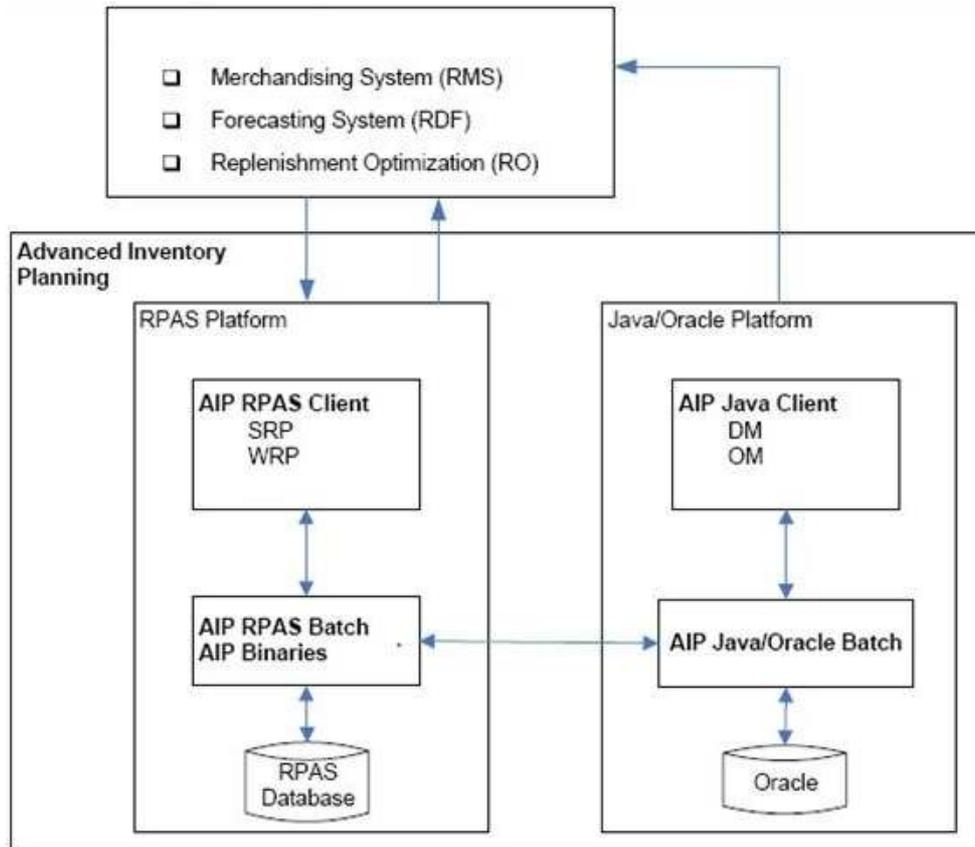


Figura 4.6 - Arquitectura interna do AIP e sua interacção com os módulos Oracle Retail.

Após análise da ilustração, verifica-se que o AIP está dividido em duas partes. Estas distinguem-se segundo a linguagem de programação utilizada para as implementar. Do lado esquerdo, apresenta-se o módulo de planeamento do reaprovisionamento, que por sua vez é composto por 2 módulos, os quais *AIP Store Replenishment Planning* (SRP) e *AIP Warehouse Replenishment Planning* (WRP). Estes encontram-se implementados numa linguagem de programação *Oracle* de nome *Retail Predictive Application Server* (RPAS). Do lado direito, existem dois módulos, o gestor de ordens de compra, *AIP Order Management* (OM), e o gestor da informação vinda dos sistemas externos ao AIP, *AIP Data Management* (DM), ambos implementados em *Java* e *Oracle*. Internamente a comunicação é feita recorrendo ao *Oracle Retail Extract, Transform, and Load* (RETL), protocolo de comunicação que já é comum aos vários módulos *Oracle Retail*. Ambos os lados são compostos por uma base de dados, onde os módulos armazenam informação, e um *Batch* que realiza os cálculos necessários sobre esses dados.

Relativamente à comunicação, vê-se então que o módulo de planeamento do reaprovisionamento, composto pelo SRP e WRP, é o que recebe a informação vinda dos módulos *Oracle Retail*. Após estes módulos receberem a informação, ambos vão sincronizá-la com o módulo DM. Este módulo será o que vai comunicar com o utilizador. Aliás é aqui que o utilizador realizará a maior parte das acções sobre o AIP e estas irão afectar a informação do DM.

Posteriormente as alterações feitas no DM serão sincronizadas novamente com os módulos SRP e WRP. Seguidamente serão realizados os cálculos do AIP, nesses módulos e os resultados serão enviados ao OM. No OM o utilizador irá confirmar e alterar os resultados gerados pelo AIP. Só no final de tudo isto é que a informação viajará para o RMS.

Resumidamente é assim o esquema da arquitectura interna do AIP e sua interacção com os módulos *Oracle Retail* [17].

Na arquitectura desta aplicação é também importante explicar o seu batch diário, que basicamente é o que faz esta aplicação funcionar. Assim o diagrama que mostra a organização desse batch está representado na Figura 4.7.

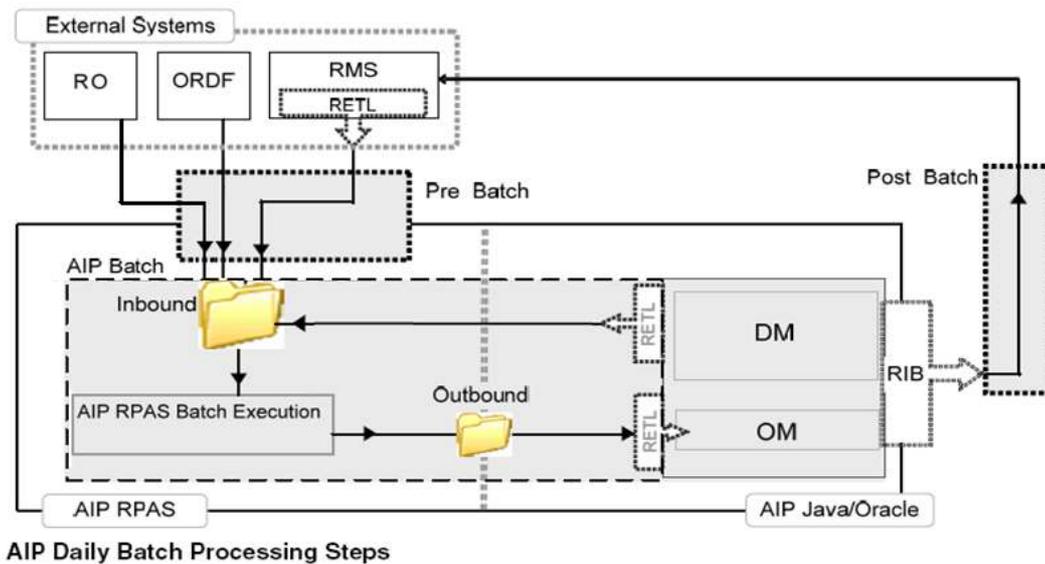


Figura 4.7 - Batch diário executado pelo AIP.

Então podemos ver que o batch se divide em 3 partes:

- Pre Batch;
- AIP Batch;
- Post Batch.

Resumidamente, o pre batch executa a extracção da informação das aplicações que contêm tudo o que o AIP necessita para funcionar e ainda sincroniza os módulos SRP e WRP com o DM. Após extraída essa informação, o AIP batch realiza todos os cálculos sobre esses dados, enviando os resultados ao OM. No final é executado o Post Batch que faz com que os resultados do AIP sejam enviados ao RMS.

No entanto, este projecto foca o seu objectivo num dos fluxos de interacção do AIP, onde se prevê que se possa melhorar o funcionamento global da aplicação. Essa perspectiva de melhoria é apresentada na secção 4.6, mas na próxima secção apresentam-se dois casos em que o AIP já foi colocado em funcionamento.

4.5.Casos Reais de Implementação do AIP

O AIP apesar de ser uma aplicação bastante recente já teve a oportunidade de ser implementado em retalhistas cujo a sua dimensão financeira os permitia arriscar em produtos inovadores. Assim, melhorias possíveis no seu funcionamento só trará mais vantagens para estes clientes e fará com que estes depositem mais confiança na mesma.

Expõe-se então alguns casos onde se justifica o facto de estas empresas arriscarem nesta implementação e quais são as expectativas das mesmas.

4.5.1. Retalhista Britânico

Retalhista Britânico que opera com mais de 700 lojas de venda directa, via telefone e via internet através do Reino Unido e da Irlanda, servindo mais de 130 milhões de pessoas por ano. No último ano financeiro as vendas ascenderam os 3.8 biliões de libras. Todos os anos esta empresa distribui 35 milhões de catálogos que abrangem 70% das casas do Reino Unido.

Com estas características os desafios desta empresa são:

- A sua cadeia de abastecimento não consegue suportar o crescimento do negócio;
- O crescimento do volume de vendas;
- Ter um serviço de recolha ao nível da unidade dos produtos nos armazéns para as vendas da internet;
- Reaprovisionamento rápido é vital para se evitar falta de *stocks*, mantendo assim os altos níveis do serviço aos clientes;
- Necessidade de informação precisa sobre o local e *stocks* dos diversos produtos;
- Aumentar a sincronização da procura dos clientes com a compra dos produtos aos fornecedores.

Com a implementação do AIP nesta empresa, além de se colmatar os desafios expostos espera obter-se os seguintes benefícios:

- Poupar 50 milhões de libras por ano, aumentando a eficiência e eficácia da gestão de *stocks*;
- Cerca de 200 parâmetros afectam as compras dos diversos produtos, para produzir um modelo de reaprovisionamento melhor;
- Reduzir o inventário de maneira a que este se aproxime à procura dos clientes, minimizando assim custos;
- Melhorar o processo de gestão das ordens de compra.

4.5.2. Retalhista do Texas nos EUA

Retalhista que abrange cerca de 1.1 milhões de famílias, durante o ciclo de vida destas no serviço das forças armadas. Opera mais de 3.100 instalações de retalho em 30 países, com mais de 1 milhões de produtos diferentes.

Os desafios desta empresa são:

- Necessidade de uma cadeia de abastecimento global e centralizada que opere de forma eficaz e eficiente;
- Falta de visibilidade que a empresa tem sobre o seu inventário;
- Suportar uma cadeia de abastecimento com menos suporte humano e que seja menos dispendiosa;
- Suportar um modelo de negociação semanal de compras aos fornecedores.

O que se espera que o AIP traga de valor para o cliente, além de resolver os desafios anteriores, é:

- Reduzir em 120 milhões de dólares em custos de inventário;
- Ganhar visibilidade dos seus *stocks* ao longo da cadeia de abastecimento;
- Criar um conjunto de aplicações de retalho que suportam o crescimento dos diversos canais de vendas;
- Aumento da velocidade, da flexibilidade e da resposta da cadeia de abastecimento;
- Abilidade para serem executadas promoções complexas.

Estes foram casos de empresas que já possuem o AIP em execução. Que com esta aplicação os clientes esperam resolver os seus problemas e ainda obter alguns benefícios. Este casos estão expostos em documentos da empresa *Wipro Technologies* [17].

4.6. Perspectiva de Melhoria à Interacção do AIP

Após o estudo realizado ao AIP e com base na experiência dos especialistas da área do AIP, percebeu-se que talvez fosse possível realizar uma melhoria da interface de comunicação entre o AIP e uma das aplicações do *Oracle Retail* e que isso poderia ter um grande impacto ao nível da eficiência do funcionamento da aplicação.

Assim para este projecto decidiu-se fazer uma investigação às ferramentas de integração destes sistemas, de forma a perceber até que ponto os fluxos de informação podem ser melhorados e quais as vantagens e desvantagens de cada ferramenta. Isto é, além de se fazer uma comparação entre ferramentas, que posteriormente servirá de apoio à decisão de implementação de fluxos de informação em aplicações *Oracle Retail*, ainda se provará se é ou não possível melhorar a actual interacção do AIP.

Portanto relativamente à interface que se prevê que possa ser melhorada, esta está representada a vermelho na Figura 4.8.

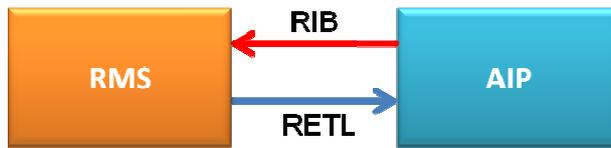


Figura 4.8 - Sugestão de melhoria à interacção do AIP.

Na figura, apresentam-se duas interfaces diferentes. A que envia informação do RMS para o AIP e a que faz o inverso. Neste caso, prevê-se que o fluxo de informação implementado em RIB pode ser melhorado se este fosse implementado noutra ferramenta. Portanto, primeiro será feita a comparação das ferramentas de integração que nos levará a escolher uma delas para mudar a actual interface, no caso de os testes provarem que esta pode ser melhorada.

Capítulo 5

Comparação das Ferramentas de Integração de Aplicações Oracle Retail

A suite *Oracle Retail* ao ser composta por várias aplicações necessita obviamente de ferramentas que as possam integrar, para que a partilha de informação seja possível.

Para tal a *Oracle* tem à disposição 3 ferramentas que podem ser utilizadas para o efeito. Estas ferramentas são o *Oracle Retail Integration Bus* (RIB) e o *Oracle Retail Extract, Transform and Load* (RETL), que são dedicadas ao mundo do retalho e ainda o *Oracle PL/SQL com SQL*Loader* que é de utilização mais genérica. Todas estas ferramentas possuem características muito diferentes e por isso apresentam diferentes comportamentos consoante a situação em que são utilizados. Cada uma delas possui mais vantagens do que as outras em casos específicos. Portanto de seguida será apresentada uma explicação de cada uma delas acompanhada de testes de comparação entre elas.

5.1. Oracle Retail Integration Bus

O RIB é sem sombra de dúvida a ferramenta mais complexa das 3 que vão ser estudadas. Esta é uma ferramenta de integração orientada a mensagens, construída com base na tecnologia Java Message Service (JMS) com sistemas de controlo de erros, de fácil configuração e com um sistema de gestão de mensagens. Em comparação com o JMS o RIB acrescenta-lhe

funcionalidades como: ordem de mensagens, dependência entre mensagens e tratamento e recuperação de erros.

O RIB é baseado na ideia de publicador e subscritor. Ou seja, existe uma aplicação que publica a mensagem no Bus de integração e esta mensagem permanecerá lá até que todos os subscritores dessa mensagem a leiam. Assim sendo, esta aplicação apresenta as seguintes características [18]:

- Comunicação assíncrona de mensagens;
- Permite que os sistemas sejam fracamente acoplados;
- Não há dependências físicas entre aplicações;
- As aplicações comunicam através de um Bus de Comunicação;
- Consistência de Dados;
- Escalabilidade;
- Confiabilidade;
- Recuperação e tratamento de erros – Hospital;
- Sequência de Mensagens.

A utilização de mensagens assíncronas é uma vantagem na medida em que as aplicações emissor e receptor tornam-se ainda mais independentes, ou seja, assim não é necessário estabelecer uma sincronia entre ambas.

Permite-se que os sistemas sejam fracamente acoplados devido ao facto de uma aplicação não precisar de saber informações de implementação de outra para que a comunicação funcione. Mais uma vez, independência entre aplicações.

Pelas características referidas anteriormente, a independência entre aplicações é grande o que permite também que não haja dependência física entre elas.

O RIB possui o Bus de integração, supra referido, que faz a difusão das mensagens.

A consistência de dados e a confiabilidade são também uma grande mais-valia para o RIB. Este garante sempre a consistência de dados. Aconteça o que acontecer, os dados nunca são perdidos numa comunicação RIB e portanto a consistência e a confiabilidade estão garantidos.

O hospital de erros, que é o método de tratamento de erros do RIB, tem como função:

- Manipular erros sem que bloqueie ou pare o canal de comunicação;
- Preserva a ordem das mensagens mesmo quando acontecem erros;
- Tentativas automáticas de envio de mensagens com erro, com um determinado espaço de tempo;

Mas é importante referir que o hospital de erros foi desenhado para receber mensagens de erro ocasionais, devido ao facto de que quanto mais erros são gerados no RIB menos eficiente este se torna.

5.1.1. Descrição do Funcionamento

O funcionamento da ferramenta RIB consegue ser representado pelo seguinte diagrama apresentado na Figura 5.1 [17].

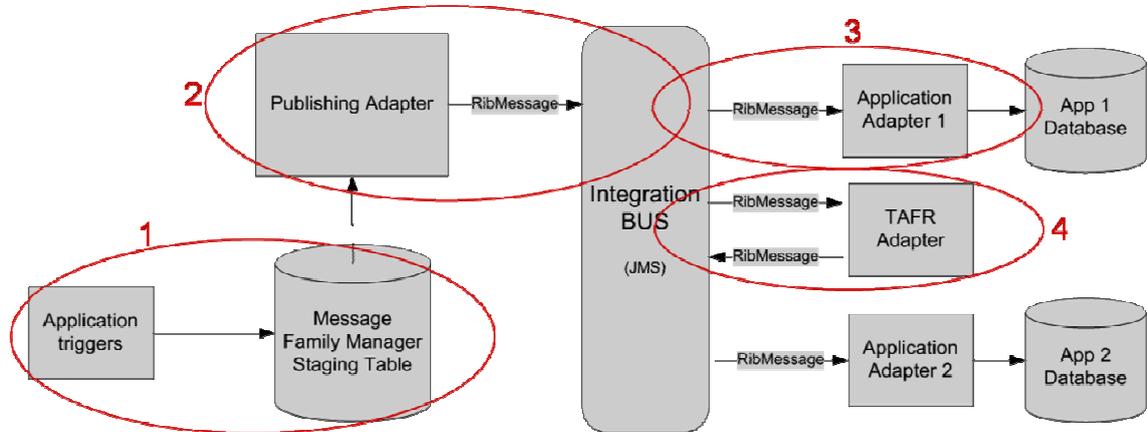


Figura 5.1 - Diagrama do funcionamento do RIB.

Para que os vários elementos presentes na figura acima possam ser melhor esclarecidos apresentam-se algumas definições na Tabela 5.1.

Após esta explicação de alguns dos termos da Figura 5.1 descrever-se-á cada um dos passos do funcionamento do RIB.

O passo 1 é o processo de desencadeamento da subscrição de uma mensagem RIB é iniciado num trigger da base da dados que pretende enviar dados a uma determinada aplicação. Esse *trigger* após construir a mensagem (no formato de registo de base de dados) a enviar escreve-a numa tabela onde o RIB frequentemente verifica se existe novas mensagens.

No passo 2, o Adaptador de Publicação (SSB) analisa a tabela de mensagens a enviar. Se esta possuir informação converte essa informação de um registo de base de dados para XML e publica a mensagem RIB.

No passo 3, o Adaptador de Subscrição (MDB) recolhe a mensagem RIB do Bus de integração e dinamicamente converte-a num registo de base de dados. Após conversão a mensagem é validada e inserida numa tabela de base de dados que aguarda por mensagens novas. Esta tabela possui um trigger que permite avisar a base de dados de destino da mensagem que chegou uma nova.

No passo 4, o Adaptador TAFR analisa todas as mensagens RIB para verificar se estas apresentam erros. Se for o caso, estas serão enviadas para o hospital de erros do RIB.

| Elemento da Figura | Descrição |
|--------------------------------------|--|
| Application triggers | São <i>triggers</i> ³ accionados quando alguma informação é introduzida editada e/ou eliminada de uma base de dados. |
| Message Family Manager Staging Table | Tabela onde os <i>Application triggers</i> colocam a informação a ser enviada pelo RIB. Esta tabela também é o local ao qual o RIB está sempre à escuta, esperando nova informação a ser enviada |
| Publishing Adapter | Adaptador que publica no <i>Integration BUS</i> as mensagens que estão nas <i>Message Family Manager Staging Table</i> . |
| Integration BUS | Sistema em Java que recebe as mensagens e faz o seu devido reencaminhamento, segundo as características dessas mensagens. |
| Application Adapter | Adaptador que recebe as mensagens enviadas pelo <i>Integration BUS</i> e que as irá publicar correctamente na base de dados de destino. |
| TAFR Adapter | Adaptador que recebe mensagens que precisem de algum tipo de transformação aos seus dados. Este recebe uma mensagem do <i>Integration BUS</i> e após transformação da mesma devolve-a ao mesmo para que possa ser entregue a um <i>Application Adapter</i> . |
| RIB Messagee | Mensagens RIB que contém a informação a ser transmitida e que circulam por todo o sistema. |

Tabela 5.1 – Definições dos elementos da Figura 5.1.

5.2. Oracle Retail Extract, Transform and Load

O RETL é uma ferramenta de movimentação de informação de alta performance entre bases de dados, de processamento paralelo, escalável, independente de qualquer plataforma e independente das bases de dados. Esta é uma ferramenta que possui a característica de permitir o desenvolvimento rápido de integrações. O desenvolvimento nesta ferramenta torna-se então mais simples e menos propenso a erros comparando com outros métodos como C ou C++.

O RETL além das vantagens referidas tem a característica de ser um bom manipulador da informação recolhida tanto de ficheiro como de uma base de dados. Este permite um enorme conjunto de operações sobre esses dados (ordenação de dados, junção de dados, filtragem, etc.) [19].

³ *Trigger* é um procedimento de código que é automaticamente executado em resposta a determinados eventos de uma base de dados. Normalmente inserções, alteração e remoções de registos de tabelas de uma base de dados [23].

5.2.1. Descrição do Funcionamento

O RETL além de ser como o RIB no facto de transmitir informação entre bases de dados, ainda possui outro tipo de operações como se pode ver na Figura 5.2.

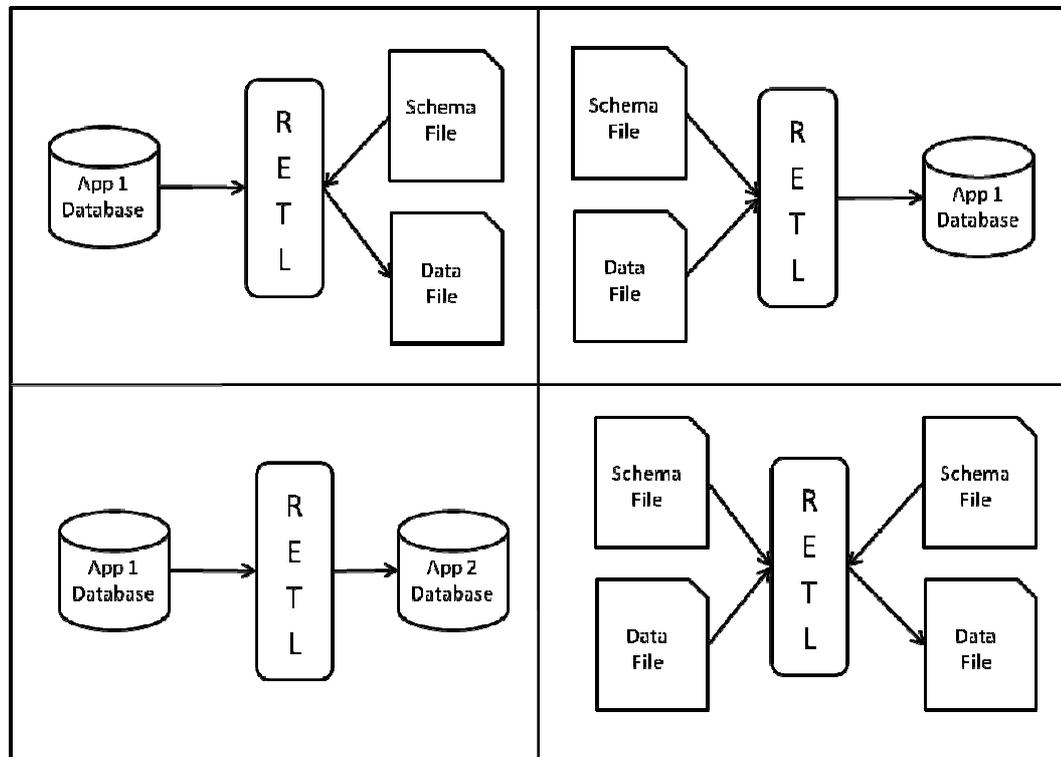


Figura 5.2 - Diagrama de funcionamento do RETL.

A ferramenta RETL pode realizar 4 tipos de operações, as 4 que estão representadas no diagrama. Ou seja, com esta podemos ir a uma base de dados buscar informação e colocá-la num ficheiro, o mesmo mas no sentido inverso, de base de dados para base de dados e de ficheiro para ficheiro.

Assim, a informação é lida directamente tanto de ficheiro como da base de dados e transformada da forma necessária no interior da aplicação RETL. Normalmente esta transformação é a necessária para que a informação encaixe de ambos os lados. No caso de bases de dados é necessário formatar os dados lidos para que possam ser correctamente carregados no destino, no caso de ficheiros estes podem ter que ter um formato especial pretendido pelo programador.

No caso dos ficheiros é importante acrescentar que o RETL necessita sempre de um ficheiro de esquema a acompanhar o ficheiro. Esse esquema transmite ao RETL de que forma a informação está organizada no ficheiro ou como se pretende organizar a informação no ficheiro.

5.3. PL/SQL com Oracle SQL*Loader

PL/SQL com Oracle SQL*Loader são duas plataformas que juntas fazem com que seja possível criar interfaces de integração. O PL/SQL tem o objectivo de ser utilizado para extrair informação da base de dados e o SQL*Loader para carregar.

O PL/SQL foi desenhado para ser um processador de comandos SQL. Assim pode-se utilizar todos os comandos SQL necessários para se aceder à informação da base de dados. No entanto o PL/SQL fornece mais algumas funcionalidades como passagem de informação lida com o SQL para ficheiro.

O SQL*Loader é uma plataforma primária para popular rapidamente tabelas *Oracle* com informação contida em ficheiros.

5.3.1. Descrição do Funcionamento

O funcionamento do PL/SQL com o SQL*Loader pode ser descrito pela Figura 5.3.

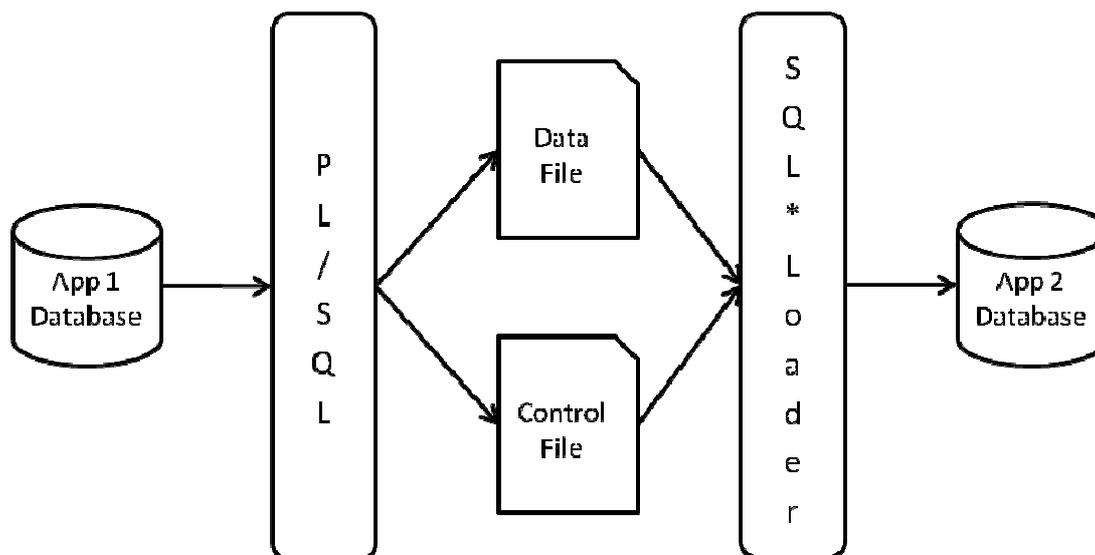


Figura 5.3 - Diagrama do funcionamento do PL/SQL com SQL*Loader.

O funcionamento de PL/SQL é simples no sentido em que é utilizado como uma linguagem de programação. Portanto, é necessário desenvolver o código pretendido, com o devido tratamento de erros, de forma a obter a informação da base de dados e a escreve-la num ficheiro de dados, escrevendo também um ficheiro de controlo onde se indica a organização do ficheiro de dados e parâmetros essenciais ao SQL*Loader.

Para funcionar o SQL*Loader precisa de 2 ficheiros (controlo e de informação). Com o ficheiro de controlo é possível saber a organização dos dados no ficheiro de informação e quais são determinados parâmetros, como a (s) tabela (s) em que se pretende inserir os dados.

5.4. Comparação das 3 Ferramentas

A comparação das 3 ferramentas é então uma das partes importantes deste projecto. É após a análise do desempenho de cada uma em determinadas situações, que se irá verificar se é possível ou não melhorar o fluxo que envia os resultados do AIP para o RMS.

Inicialmente, explicar-se-á os métodos utilizados para a avaliação de desempenho de cada uma das ferramentas e posteriormente apresentar-se-á os resultados obtidos.

5.4.1. Métodos Utilizados

Após o estudo intensivo das 3 ferramentas, verifica-se que todas elas têm apenas a função de passar informação entre bases de dados.

Decidiu-se então afastar esta avaliação dos produtos *Oracle Retail* e utilizar apenas bases de dados normais. Obviamente que para todas as avaliações pretende-se manter o mesmo ambiente de teste.

Para esta avaliação foram definidas métricas para que se possa avaliar várias características das interfaces de comunicação. Essas métricas são:

- Custo de Desenvolvimento de uma interface de comunicação;
- Custo de Alteração de uma interface de comunicação;
- Tempo que a interface demora a passar determinada quantidade de informação;
- Método de tratamento de erros;
- Facilidade de recuperação de erros.

Na análise do custo de desenvolvimento de uma interface, o objectivo é definir uma interface de comunicação e posteriormente desenvolver essa mesma interface com cada uma das ferramentas. Durante este processo será medido o número de horas de recursos humanos que foram dispendidos para tal tarefa. Já que o custo de desenvolvimento só envolve recursos humanos e nenhum recurso financeiro, então a avaliação torna-se assim válida.

O custo de alteração de uma interface será avaliado da mesma forma que o anterior, mas neste caso será definida uma alteração à interface anterior e posteriormente será implementada essa alteração em cada uma das ferramentas.

O tempo que a interface demora a passar determinada quantidade de informação entre bases de dados vai ser avaliada colocando dados numa das base de dados e depois fazendo-os passar para a outra base de dados com cada uma das implementações da interface. Neste caso vai ser medido o tempo que essa informação leva a ser colocada na segunda base de dados. Esta será também a métrica a que será dada maior relevância, visto que é a partir dos seus resultados que se conclui se se pode melhorar ou não o tempo de integração do AIP com o RMS.

Nas duas últimas métricas a avaliação será ligeiramente diferente. Ou seja, neste caso a avaliação não vai ser tão objectiva como as anteriores, visto que não existe essa possibilidade. Neste caso vamos colocar um utilizador razoavelmente experiente na área da informática a fazer o tratamento de erros, avaliando assim qual delas tem o melhor método e qual é a facilidade de execução dessa tarefa.

É importante referir aqui também que a comparação realizada podia ter obtido melhores resultados se esta tivesse os melhores recursos de *hardware* possíveis. Os testes realizados utilizaram apenas como *hardware* um computador pessoal. As bases de dados e as 3 ferramentas correram num sistema virtual Linux, utilizando a ferramenta VMware® no sistema operativo Microsoft Windows®. Como se pode imaginar um computador pessoal a executar dois sistemas operativos, uma máquina virtual e ainda todas as aplicações necessárias aos testes terá alguns problemas ao nível do desempenho de todos os sistemas. Toda esta configuração dos elementos utilizados pode ter criado alguns problemas de performance nas várias ferramentas, que segundo a documentação destas a mais afectada pode ter sido o RIB. No caso desta última documentação, os seus autores referiam que o RIB deve ter utilizado num servidor dedicado a si [18]. Esta é uma aplicação que necessita de muitos recursos como memória e capacidade de processamento, visto que é uma aplicação que está a ser executada todo o tempo. Assim é importante ter em atenção os resultados obtidos, segundo este factor, principalmente na ferramenta RIB. Nas conclusões apresentadas na secção 5.4.3 já será tido em conta este problema.

5.4.2. Resultado Obtidos

Após execução dos métodos anteriormente descritos apresentam-se os resultados obtidos nas 3 ferramentas de integração.

Relativamente ao custo de desenvolvimento de uma interface de comunicação os resultados obtidos podem ser consultados na Tabela 5.2.

| RIB | RETL | PL/SQL com SQL*Loader |
|------|------|-----------------------|
| 600% | 100% | 300% |

Tabela 5.2 – Resultados obtidos na métrica de custo de desenvolvimento de uma interface de comunicação.

As percentagens da tabela mostram quanto custa desenvolver uma interface de comunicação em relação à que custa menos. Ou seja, neste caso a ferramenta que apresenta melhor tempo de desenvolvimento é o RETL, sendo que o PL/SQL com SQL*Loader custa 3 vezes mais e o RIB 6 vezes. Este resultado foi obtido, com um recurso humano sem experiência com as ferramentas, desenvolvendo uma interface de comunicação em cada uma delas para que desempenhassem exactamente a mesma função. É importante referir também que o tempo do PL/SQL com SQL*Loader pode não ser muito real, porque o tempo apresentado não conta com a

Comparação das Ferramentas de Integração de Aplicações Oracle Retail

implementação de tratamento de erros. Assim, qualquer tratamento de erros que se pretenda tem de ser pensado e implementado pelo programador. Portanto, o tempo de desenvolvimento desta ferramenta pode duplicar ou triplicar neste caso, levando a que seja pior que o RIB em termos de custos de desenvolvimento.

Em relação ao custo de alteração de uma interface de comunicação já implementada, os resultados podem ser consultados na Tabela 5.3.

| RIB | RETL | PL/SQL com SQL*Loader |
|------|------|-----------------------|
| 300% | 100% | 100% |

Tabela 5.3 - Resultados obtidos na métrica de custo de alteração de uma interface de comunicação.

Desta tabela concluímos que as alterações às interfaces em RETL e o PL/SQL com SQL*Loader levam o mesmo tempo a ser realizadas. Importante referir aqui, que este resultado não conta com alterações ao tratamento de erros. Se for preciso fazer alterações a esse tratamento de erros o tempo de alteração a uma interface em PL/SQL com SQL*Loader pode aumentar bastante.

As interfaces implementadas em RIB como se pode ver pela Tabela 5.3 levam até três vezes mais tempo a realizar uma alteração do que as outras duas. Isto é compreensível devido à complexidade desta ferramenta. A complexidade é tanta que para se realizar uma alteração pode ter que se alterar código em mais do que cinco ficheiros, ao contrário das outras aplicações que basta alterar no máximo dois.

Em relação ao tempo que a interface demora a transmitir determinada quantidade de informação os resultados obtidos estão documentados na Tabela 5.4.

| Número de Registos Transmitidos | RIB | RETL | PL/SQL com SQL*Loader |
|---------------------------------|---------------|-----------|-----------------------|
| 1.000.000 | 8 h 42 m 20 s | 14 m 14 s | 30 m 38 s |
| 500.000 | 4 h 3 m 47 s | 6 m 24 s | 16 m 40 s |
| 250.000 | 2 h 15 m 22s | 3 m 51 s | 9 m 3 s |
| 100.000 | 50 m 14 s | 1 m 22 s | 3 m 20 s |
| 50.000 | 21 m 36 s | 1 m 9 s | 1 m 52 s |
| 25.000 | 12 m 32 s | 1 m 4 s | 1 m 7 s |
| 10.000 | 4 m 53 s | 59 s | 58 s |

Tabela 5.4 – Resultados obtidos na métrica do tempo que a interface demora a passar determinada quantidade de informação.

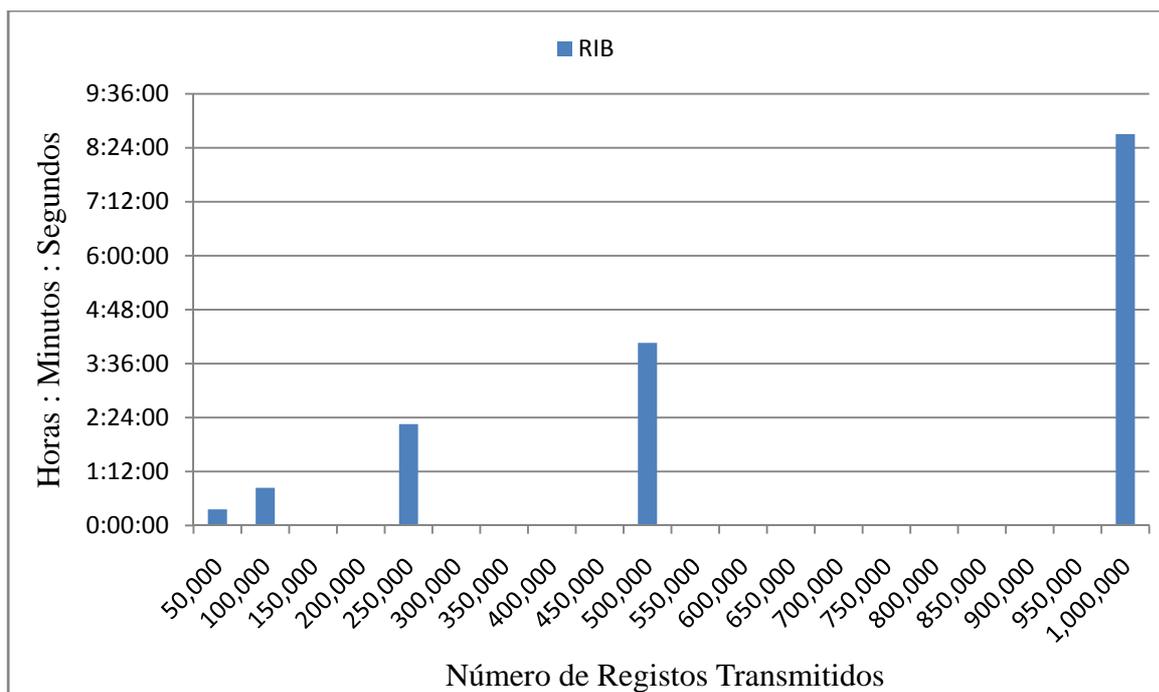


Figura 5.4 – Resultados obtidos na métrica do tempo que a interface, implementada em RIB, demora a passar grandes quantidades de informação.

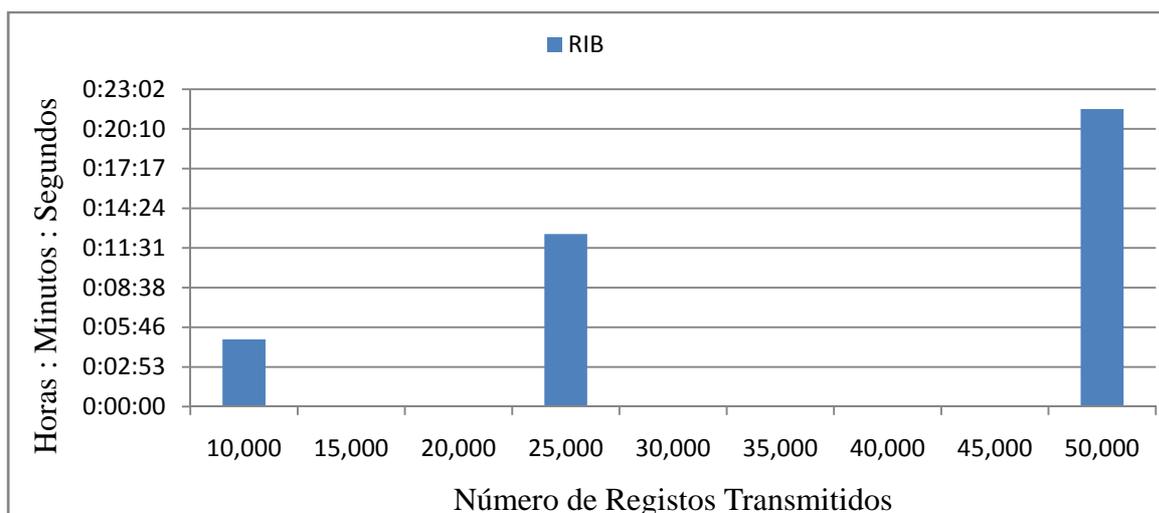


Figura 5.5 - Resultados obtidos na métrica do tempo que a interface, implementada em RIB, demora a passar pequenas quantidades de informação.

Para que se possam tirar várias conclusões aos resultados da Tabela 5.4, apresentam-se os mesmos em gráficos. Portanto serão apresentadas 4 Figuras (Figura 5.4, Figura 5.5, Figura 5.6 e Figura 5.7). As duas primeiras apresentam resultados sobre a ferramenta RIB e as duas seguintes apresentam os resultados das outras duas ferramentas comparadas. Não se juntou os resultados das 3 ferramentas devido ao facto de o RIB apresentar resultados muito maiores do que as outras duas, levando a que não se consiga perceber os 3 resultados juntos. Depois a diferença entre as duas primeiras Figuras e as duas últimas é o facto de estas estarem divididas por quantidade de informação transmitida. O RETL e o PL/SQL com SQL*Loader apresentam diferentes

Comparação das Ferramentas de Integração de Aplicações Oracle Retail

características nos seus resultados para quantidades grandes de informação e para pequenas quantidades.

Na Figura 5.4 e Figura 5.5, vê-se o desempenho da ferramenta RIB onde se apresenta na primeira figura valores grandes e a seguinte valores mais pequenos da quantidade de informação transmitida.

Com estas duas Figuras podemos concluir que o RIB apresenta um resultado linear no que toca ao tempo de transmissão. Ou seja, a quantidade de dados aumenta proporcionalmente ao tempo que demora a transmissão e isto mantêm-se tanto para valores grandes como valores pequenos no número de registos transmitidos.

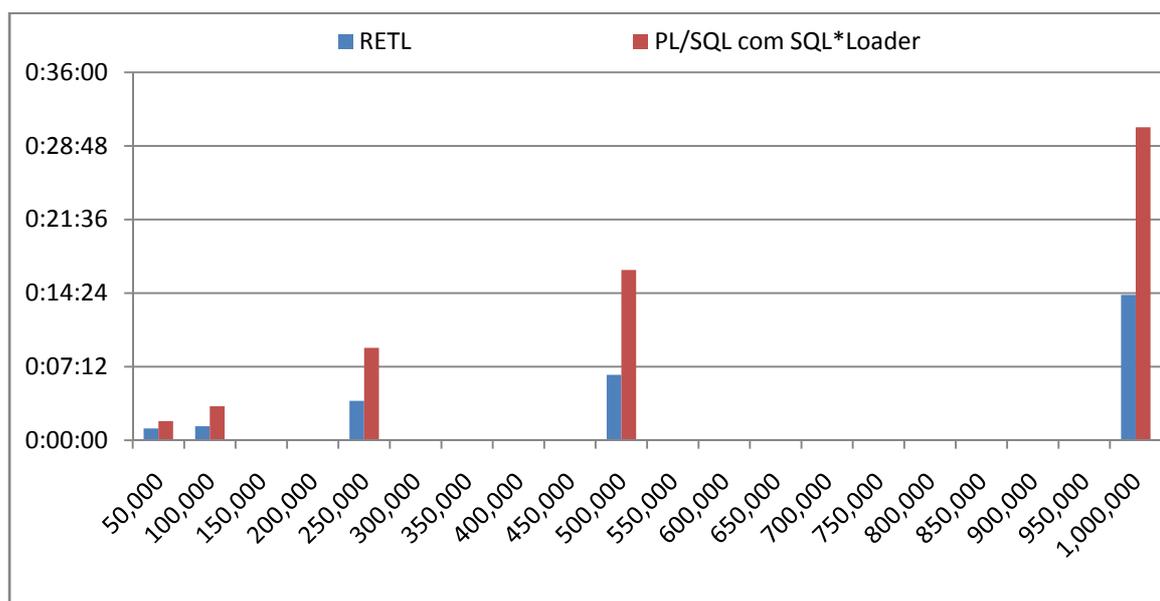


Figura 5.6 - Resultados obtidos na métrica do tempo que as interfaces, implementadas em RETL e PL/SQL com SQL*Loader, demoram a passar grandes quantidades de informação.

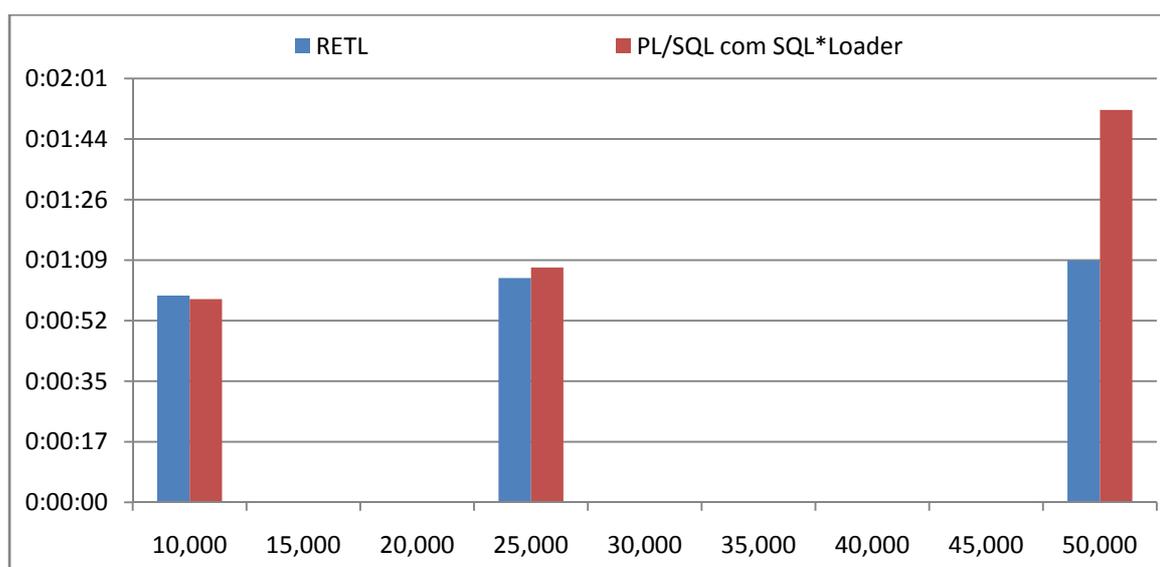


Figura 5.7 - Resultados obtidos na métrica do tempo que as interfaces, implementadas em RETL e PL/SQL com SQL*Loader, demoram a passar pequenas quantidades de informação.

Comparação das Ferramentas de Integração de Aplicações Oracle Retail

Segundo a Figura 5.6 e a Figura 5.7 verifica-se que o PL/SQL com SQL/Loader possui quase sempre um tempo de transmissão maior do que o RETL para a mesma quantidade de informação. Ainda destas Figuras pode-se concluir que as ferramentas para valores elevados de informação apresentam um crescimento linear, mas para valores inferiores começa-se a perder a linearidade. Este facto deve-se ao tempo que ambas as ferramentas levam a iniciar o processo de transmissão. Este tempo é sempre igual, independentemente da quantidade de informação a transmitir. Por isso quanto maior for a quantidade de informação menor se nota a influência deste tempo no resultado final.

Mas para facilitar a leitura dos números da Tabela 5.4, apresentam-se os mesmos resultados mas em unidades diferentes na Tabela 5.6.

| Número de Registos Transmitidos | RIB | RETL | PL/SQL com SQL*Loader |
|---------------------------------|-----|-------|-----------------------|
| 1.000.000 | 32 | 1,171 | 544 |
| 500.000 | 34 | 1,302 | 500 |
| 250.000 | 31 | 1,082 | 460 |
| 100.000 | 33 | 1,220 | 500 |
| 50.000 | 39 | 725 | 446 |
| 25.000 | 33 | 391 | 373 |
| 10.000 | 34 | 169 | 172 |

Tabela 5.5 - Interpretação diferente dos resultados obtidos na Tabela 5.4 (unidades em registos/segundo).

A partir da Tabela 5.5 criou-se a Tabela 5.6 que representa a média de registos transmitidos por cada uma das ferramentas. É importante referir que no RETL e no PL/SQL com SQL*Loader a média inclui apenas os resultados entre 100.000 e 1.000.000 de registos, visto que são os valores onde o tempo inicial de transmissão afecta menos o resultado.

| RIB | RETL | PL/SQL com SQL*Loader |
|-----------------------|------------------------|------------------------|
| 34 registos / segundo | 1200 registos /segundo | 500 registos / segundo |

Tabela 5.6 – Média dos resultados obtidos na Tabela 5.5.

Com estes resultados percebe-se que a ferramenta RIB é bastante pior que as outras no que toca à quantidade de informação que consegue transmitir por unidade de tempo. Após todo o estudo realizado a estas ferramentas percebe-se o porquê disto acontecer. Ou seja, o objectivo do RIB, nas aplicações *Oracle Retail*, é bem diferente do RETL. Conseguiu-se perceber que o primeiro tem a função de transmitir dados *online* enquanto o RETL só o consegue fazer *offline*.

Isto é, o RIB consegue passar informação sem que para isso seja necessário correr o *batch* diário da aplicação. Portanto a aplicação pode estar disponível ao seu cliente e estar em tempo real a passar informação a outra aplicação. Enquanto no RETL, a informação só será transmitida quando a aplicação estiver *offline* e o *batch* for executado. O PL/SQL com SQL*Loader não acrescenta vantagens em relação ao RETL, relativamente a esta métrica. O seu resultado é pior e não consegue ter nenhuma vantagem em relação ao RETL. Isto é compreensível, devido ao facto de não ser uma aplicação dedicada ao mundo do retalho *Oracle*.

Em suma, apesar de ambas as aplicações (RIB e RETL) apresentarem tempos de transmissão extremamente diferentes, ambas têm objectivos diferentes para a integração de aplicações *Oracle Retail*.

Em relação ao método de tratamento de erros e como referido anteriormente, este serão apresentados de forma não objectiva e baseados na experiência de utilização de cada uma das ferramentas. Começando pela ferramenta que não integra a suite *Oracle Retail*, ou seja o PL/SQL com o SQL*Loader, esta apresenta uma forma de resolução de erros quase nula. O único método que existe de tratamento de erros é na parte do SQL*Loader, que guarda todos os registos com erro num file “bad”, para que no final o utilizador possa ir verificar a causa daqueles registos terem falhado e depois de os corrigir voltar a enviá-los à base de dados. No lado que extrai a informação da base de dados (PL/SQL) não existe qualquer tratamento de erros. Todos os erros têm de ser previstos pelo utilizador e posteriormente terá de ser implementado esse tratamento de erros. Assim, considera-se que o método de tratamento de erros é bastante mau nesta ferramenta.

No RIB e no RETL o tratamento de erros já é muito mais adequado, visto que são ferramentas dedicadas ao mundo do retalho da *Oracle*. Como neste mundo as aplicações lidam com milhões de registos de informação, prevê-se que possam surgir erros nesses registos. Portanto, ambas as ferramentas possuem tratamento de erros. Considera-se que o RIB tem um método de tratamento de erros muito melhor que o RETL. O RIB apresenta métodos como hospitais de erros onde é bastante fácil fazer o tratamento de erros. Os hospitais contêm a informação com erros, até que seja corrigida pelo utilizador para que na próxima tentativa de envio do RIB esta seja transmitida. Ou seja, após corrigida a informação o utilizador não precisa de se preocupar em enviar novamente a informação, visto que o RIB o faz automaticamente. Para corrigir uma mensagem basta o utilizador ir a uma tabela própria do hospital verificar a condição que falhou e corrigir essa informação na mensagem. Portanto torna-se bastante simples o método de tratamento de erros. No RIB a quantidade de erros gerados são uma grande preocupação. Quanto mais erros são gerados, mais afectada fica a performance do mesmo, visto que o RIB tenta de tempos a tempos fazer a retransmissão de todos os registos com erros. Ou seja, quanto maior o número de erros maior a utilização de recursos para a retransmissão.

No RETL o tratamento de erros é baseado em mensagens para o *stdout*⁴. Assim, e sempre que o programador o desejar pode definir o RETL para que este faça o *debug*⁵ dos erros que vão sendo gerados na transmissão de informação. Esta ferramenta possui uma propriedade em que se define o número de registos com erro que podem acontecer sem que esta aborte. Isto é útil porque no retalho é crítico serem gerados grandes quantidades de erros. Assim se os erros forem mais do que uma determinada quantidade será necessária intervenção humana para que se corrija o problema. Se a quantidade de erros for razoavelmente insignificante então estes podem ser vistos com menos preocupação porque não irão afectar grandemente o negócio do cliente. Se os erros fizerem abortar o RETL então será necessário voltar a executar a transmissão da informação novamente, ao contrário do RIB. Senão basta verificar os que possuem erros e voltar a envia-los.

5.4.3. Conclusões da Comparação das Ferramentas

Após a comparação destas 3 ferramentas de integração das aplicações *Oracle Retail*, pode-se tirar diversas conclusões.

Conclui-se que a ferramenta PL/SQL com SQL*Loader não é uma solução para os problemas de integração das aplicações. Ou melhor a utilização desta não trás vantagens em comparação com as outras duas. Esta ferramenta assemelha-se bastante à ferramenta RETL só que com piores resultados nos testes efectuados. O tempo de desenvolvimento e alteração das interfaces é maior, o tempo de transmissão de informação é maior e o tratamento de erros tem de ser implementado manualmente. Assim verifica-se que o facto de a ferramenta não estar associada à suite *Oracle Retail* tem o seu significado. Apesar de que esta é uma ferramenta genérica de utilização em bases de dados ou seja, já existia antes de aparecerem as aplicações dedicadas ao retalho.

Então excluindo esta ferramenta tirar-se-á conclusões sobre as outras duas. Nos testes realizados ambas as ferramentas apresentam resultados bem diferentes e sugestivos ao facto de que o RIB é bem pior que o RETL, excepto no tratamento de erros. Mas como já referido anteriormente, mesmo assim o RIB tem algumas vantagens no seu funcionamento que não se traduz nos resultados.

Conclui-se então que as duas ferramentas têm funções bem diferentes na utilização para integração das aplicações de retalho. O RIB apresenta-se como a ferramenta de integração para

⁴ É o *standard output*, que é um fluxo de informação entre uma aplicação computacional e o seu ambiente (sistema operativo), que normalmente é no formato de texto. Neste caso, o fluxo de informação é no sentido da aplicação para o ambiente [24].

⁵ É o processo de encontrar e diminuir o número de erros ou defeitos numa aplicação de software [25].

transmissão de dados *online* enquanto o RETL é *offline*, como explicado na secção 5.4.2. O método de funcionamento de RIB tem também muitas vantagens em relação à integração da mesma informação com diversas aplicações. Ou seja, quando é necessário sincronizar a informação de uma aplicação com várias outras, o facto de o RIB ser baseado no método publicador subscritor, é uma mais-valia. Isto porque o publicador só se preocupa em publicar no RIB e depois este fica encarregue de entregar a mensagem a cada um dos seus subscritores. O RETL consegue-se diferenciar do RIB no facto de que é extremamente mais rápido a transmitir informação entre bases de dados.

Portanto conclui-se que não existe uma ferramenta melhor do que a outra para todos os casos mas que por outro lado as ferramentas têm funções diferentes no que toca à transmissão de informação entre bases de dados.

Capítulo 6

Melhoria à Interação do AIP

Após o trabalho realizado com as 3 ferramentas de integração de aplicações *Oracle Retail*, procedeu-se a um trabalho de melhoria da interface proposta na secção 4.6 (Perspectiva de Melhoria à Interação do AIP). Os resultados obtidos na comparação do capítulo anterior conseguiram ser suficientemente entusiasmantes para que se procedesse à tentativa de melhoria.

Nesta tentativa de melhoria e segundo os resultados obtidos, decidiu-se descartar o PL/SQL com o SQL*Loader. A explicação para o facto de se descartar esta ferramenta está no capítulo 5.4.3(Conclusões da Comparação das Ferramentas). Como já foi dito esta é uma ferramenta que não acrescenta vantagens quando comparada com as outras duas. Além disso não é uma ferramenta de integração dedicada a aplicações *Oracle Retail*. Portanto a partir deste momento irá considerar-se apenas o RIB e RETL.

Assim, começou-se a desenvolver este trabalho, no qual se fez um estudo inicial no sentido de perceber qual seria a melhor forma de criar mais valor de mercado no AIP. Durante esse estudo foram surgindo algumas dúvidas e questões. Questões essas que incidiam no facto de o RIB e o RETL terem funcionamentos completamente diferentes, que nos levaram a não olhar exclusivamente, de forma objectiva, para os resultados obtidos. Assim colocaram-se questões como:

- O RETL é mais rápido que o RIB, mas não passa dados online, será que compensa?
- Será uma mais valia substituir completamente o RIB pelo RETL?

Após alguma reflexão e estudo, decidiu-se que o melhor mecanismo seria criar uma interface híbrida e dinâmica, permitindo usufruir do melhor destas duas ferramentas de integração. Com

isto pretende-se que o cliente desta aplicação possa escolher a solução que melhor se adequa ao seu negócio, tirando assim o melhor partido da mesma.

Para explicar um pouco melhor o porquê desta solução, tem que se explicar um pouco uma parte do funcionamento do AIP. Lembrando alguns conceitos anteriormente explicados, o AIP é internamente composto por 4 módulos. Os módulos são: *Store Replenishment Planning* (SRP), *Warehouse Replenishment Planning* (WRP), *Data Management* (DM) e o *Order Management* (OM). Então é importante referir que o OM é o módulo que após o AIP ter realizado os cálculos das PO e TSF necessárias, as envia para ele de forma a serem validadas pelo cliente (para mais detalhe consultar a secção 4.4). Mas este módulo pode não ser utilizado pelo AIP, tudo depende das configurações que o cliente pretender. No caso de o cliente definir que o AIP pode enviar directamente as PO e as TSF para o RMS sem necessidade de aceitação por mão humana, então estas nem passam pelo OM. Caso contrário irão para o OM para que sejam validadas por recursos humanos. Com o estudo realizado sobre o RIB, o RETL e o AIP consideramos que esta característica do funcionamento do último é muito importante na resposta às perguntas anteriores.

Por isso, em vez de se substituir a interface que envia as PO e TSF do AIP para o RMS, que actualmente está implementada em RIB, acrescenta-se uma nova implementação dessa mesma interface mas implementada em RETL. Assim, considera-se que esta interface será híbrida, devido ao facto de existirem duas implementações diferentes, que podem ser escolhidas pelo cliente, consoante as necessidades que este tenha num determinado momento.

O motivo para o facto de ser uma melhoria para o AIP deixar as implementações da interface, que faz a comunicação do AIP para o RMS, em RIB e em RETL é o facto de cada uma delas ter vantagens em casos de utilização específicos do AIP. De seguida dá-se o exemplo de dois casos de utilização específicos.

Mas antes de se explicar cada um dos casos de utilização é importante introduzir alguns dos valores do negócio dos retalhistas. Na Tabela 6.1 podemos consultar os volumes de registos que normalmente são enviados por dia e por hora de PO e de TSF. Estes valores são meramente ilustrativos mas representam bem a realidade do mundo do retalho [20].

| Informação a Enviar | Registos por Dia | Registos por Hora |
|-----------------------|------------------|-------------------|
| Ordens de Compra (PO) | 355 000 | 355 000 |
| Transferências (TSF) | 1 000 000 | 250 000 |

Tabela 6.1 – Volume de registos de ordens de compra e de transferência a serem transmitidos.

O primeiro caso de utilização e que é favorável ao RETL é o facto de o cliente definir as propriedades do AIP para que as PO e TSF, geradas pela aplicação, sejam aceites automaticamente pelo sistema, sem que tenham de ser validadas por recursos humanos. Isto fará com que todas as PO e TSF estejam prontas para serem enviadas no final da execução do *Batch*

do AIP. Logo, isto fará com que a carga de envio destes elementos, no final do *Batch*, seja bastante grande. Portanto neste caso a utilização do RIB faria com que o envio de todas as PO e TSF demorassem bastante tempo, visto que a eficiência do RIB em transmissão de informação não é muito elevada. Logo faria diminuir a eficiência do AIP. A utilização do RETL neste caso torna-se vantajosa. Como já visto anteriormente, o RETL pode enviar cerca de 1.200 registos por segundo, assegurando que a eficiência no envio de todas as ordens de compra e transferências não seja grandemente afectada. Basicamente, o RETL enviaria todas as PO e TSF diárias em menos de 18 minutos (tempo que levaria se o RETL tivesse a correr nas condições de performance explicadas no último parágrafo da secção 5.4.1), isto segundo os valores da Tabela 6.1. Além disso, o RETL só pode ser utilizado durante a execução do *Batch*, visto que o código para o executar tem de ser colocado no mesmo. Este último factor adequa-se perfeitamente a este caso de utilização, visto que se pode acrescentar o código necessário para correr a interface em RETL no final do código do *Batch*, ou seja após a criação de todas as PO e TSF.

Mas neste caso de utilização pode surgir a questão de pensarmos o porquê de acelerar o envio das PO e TSF no final do *Batch*, se estas eram lentamente enviadas se tivessem de ser validadas por mão humana. A vantagem de se acelerar o envio quando estas não precisam de validação é o facto de se dar mais tempo ao retalhista para que possa prever as compras necessárias para a sua cadeia de abastecimento (CA). Isto é, imaginemos que o retalhista no dia 10 precisa de ter determinado produto a chegar às suas lojas e o seu fornecedor desses produtos diz que a encomenda tem de chegar dia 5 à sua empresa para que os produtos estejam dia 10 no retalhista. Então a encomenda tem de ser gerada até ao dia 5. Se as encomendas fossem enviadas lentamente, o AIP tinha de fazer previsões no dia 4 para que durante o dia 5 a encomenda chegasse ao fornecedor, enquanto que se estas forem enviadas mais rapidamente a encomenda podia ser gerada com as previsões do dia 5 para serem entregues ao fornecedor no mesmo dia. Prever uma encomenda para dia 10, no dia 4 ou no dia 5 pode fazer muita diferença num retalhista. O objectivo destes é sempre aproximar o mais possível a procura com as compras aos fornecedores e quanto menor for esse tempo mais rentabilidade têm os retalhistas.

Por isso, o facto de o AIP permitir ao cliente escolher entre validar as encomendas ou não é só para manter mais a confiabilidade do mesmo na aplicação. Normalmente, os retalhistas querem validar sempre as encomendas no início de utilização da aplicação. Assim que percebem que a aplicação gera bons resultados e que a validação já não vale a pena, então abandonam a mesma para que possam ganhar mais tempo na previsão das encomendas.

O segundo caso de utilização, e que é mais favorável à utilização do RIB, é quando o envio das PO e TSF não é feito automaticamente para o RMS. Então, estas são colocadas no módulo OM para que sejam validadas pelo utilizador, para posteriormente poderem viajar para o RMS. Neste caso é vantajoso utilizar o RIB, porque as PO e TSF vão sendo validadas ao longo de um determinado espaço de tempo. Ou seja, dá tempo suficiente ao RIB para entregar as PO e TSF,

assim que vão sendo validadas. Com a utilização de RIB neste caso, o processo de entrega das PO e TSF para o RMS é feito em tempo real. Ou seja, ao contrário do RETL o RIB não entrega estas informações só durante a execução do *Batch* (*offline*). Este consegue transmitir informação *online* ou seja durante o funcionamento da aplicação. Isto pode tornar-se vantajoso visto que faz chegar as ordens de compras e as transferências ao RMS, mal estas sejam validadas. Segundo, os resultados obtidos na ferramenta RIB, esta conseguiria enviar todas as PO e TSF em cerca de 11 horas, que distribuídas pelo período em que a aplicação está *online* (cerca de 14 horas) daria perfeitamente, apesar de que acredita-se que o RIB a funcionar com um servidor dedicado a si (ou seja, como costuma estar instalado nos clientes) podia reduzir esse tempo de 11 horas para menos de metade. Isto permite com segurança garantir que as PO e TSF são entregues em tempo real ou quase à medida que vão sendo validadas. O caso mais irrealista que podia fazer com que estas informações não fossem entregues em tempo real era o facto de serem validadas mais de 2000 PO ou TSF ao mesmo tempo, fazendo com que levassem cerca de um minuto a serem entregues.

É importante referir que o tempo acima indicado, para o envio das PO e TSF diárias, através das interfaces RIB e RETL é meramente ilustrativo. No entanto o tamanho dos registos no qual foram efectuados os testes às diferentes ferramentas têm sensivelmente o mesmo tamanho que as PO e TSF, logo permite-nos fazer este cálculo de tempo aproximado, traduzindo-se assim praticamente nos valores reais.

Em suma, as vantagens de manter as duas ferramentas são:

- No caso em que o RETL é melhor (as PO e TSF são entregues automaticamente, sem aprovação prévia):
 - Envio mais rápido de todas as PO e TSF geradas para o RMS.
 - Se neste caso se usasse RIB, as PO e TSF demorariam muito mais a chegar ao RMS, fazendo com que se pudessem atrasar a chegar aos fornecedores.
- No caso em que o RIB é melhor (o contrário da anterior):
 - Envio das PO e TSF em tempo real, mal são aprovadas;
 - Se fosse o RETL, as PO e TSF só seriam enviadas na próxima execução do Batch. Assim, chegam mais cedo aos fornecedores;

Assim na Figura 6.1 vê-se a nova arquitectura da interface que envia a informação do AIP para o RMS.

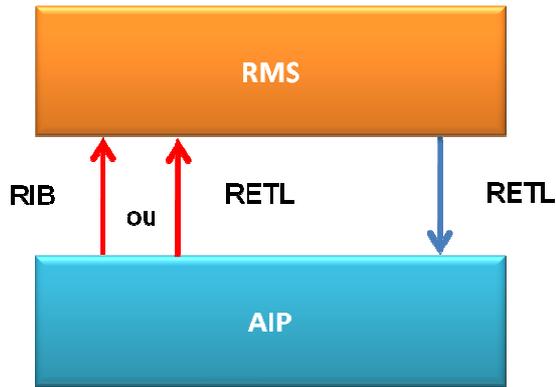


Figura 6.1 – Arquitectura final da integração do AIP melhorada.

É importante explicar que o cliente da aplicação AIP para alterar de uma interface para outra basta ir a uma tabela de propriedades do sistema e alterar um dos campos que indica qual das duas utilizar. Basicamente cada uma das ferramentas é sempre executada, mas ambas têm uma condição inicial que verifica qual das duas está activa. Ou seja, no fim do *Batch* executa-se o código RETL para que transmita a informação mas este vai à base de dados e se detectar que é o RIB que está activo aborta, acabando por não executar nada. No RIB é exactamente a mesma coisa, mas como esta aplicação está sempre a ser executada de tempos a tempos ela vai à tabela verificar qual das ferramentas é para ser utilizada, se for o RETL não faz nada e volta a ser executada passado um determinado tempo.

Capítulo 7

Conclusões

Segundo os objectivos do trabalho pode concluir-se que foram cumpridos e que desse trabalho surgiram melhorias à integração do AIP com outras aplicação *Oracle Retail*, que era o objectivo principal do trabalho, tal como o título do projecto indica.

Para a obtenção das melhorias pretendidas foi necessário fazer um estudo e uma comparação das ferramentas de integração de aplicações do mundo do retalho. Pode dizer-se que os resultados obtidos desse estudo foram muito importantes para a realização do restante trabalho. A partir deste estudo obteve-se uma visão geral, muito boa, das características e funcionalidades das diversas ferramentas. Com esta parte do projecto documentou-se essas funcionalidades segundo casos práticos de aplicação, sendo agora informação útil no processo de desenvolvimento de software para a suite *Oracle Retail*.

Portanto, as conclusões, que se tiram deste estudo inicial, são que o PL/SQL com SQL*Loader não é uma solução para integração dos diferentes sistemas da *Oracle Retail* e que o RIB e o RETL têm objectivos muito diferentes no que toca a executar essa mesma tarefa . Então a primeira ferramenta não é solução, visto que não acrescenta vantagens em relação às outras ferramentas e visto que apresenta piores resultados nas diferentes métricas de avaliação das mesmas. Excluindo então esta da comparação, resta a comparação entre o RETL e o RIB. Destas, conclui-se que ambas têm funções diferentes na integração e que por isso uma não pode ser considerada melhor do que a outra na generalidade. Portanto, para se considerar uma melhor do que a outra temos de as enquadrar com diferentes cenários de utilização. Assim, considera-se que ambas as ferramentas podem ser bastante úteis na integração de aplicações *Oracle Retail*.

Então, a partir do estudo inicialmente realizado o objectivo seria estudar a possibilidade de melhorar a integração do AIP com outras aplicações *Oracle Retail*, do qual obtivemos a conclusão de que seria possível. Inicialmente pensava-se que a melhoria possível seria trocar a implementação da interface, que faz a integração entre o AIP e o RMS, de RIB para RETL. Mas com os resultados obtidos no estudo anterior podemos concluir que seria muito mais vantajoso outro tipo de melhoria. Então decidiu-se que seria extremamente benéfico criar um novo conceito de interfaces, que se encaixaria perfeitamente com o funcionamento do AIP. Assim, o objectivo passou a ser criar uma interface híbrida e dinâmica para integrar o AIP e o RMS. Basicamente, uniu-se numa interface de integração as vantagens das ferramentas RIB e RETL. A interface híbrida criada permite agora aos clientes do AIP escolher qual das duas implementações utilizar, consoante as suas necessidades, tirando assim o maior partido das vantagens de cada uma.

Portanto, a conclusão dos resultados dos objectivos do projecto é o facto de se ter melhorado uma aplicação recente no mundo do retalho, dando-lhe mais um avanço nas vantagens competitivas que esta apresenta para os seus clientes. São pequenas melhorias como esta que vão tornando o AIP mais sólido e mais interessante no mercado. Esta melhoria também veio ajudar a minimizar um dos maiores problemas apontados pelos retalhistas a esta aplicação. Ou seja, o tempo de entrega das ordens de compra (PO) e transferências (TSF) ao RMS, por parte do AIP. Em causa estava o tempo de integração, num dos cenários de utilização descritos no Capítulo 6.

Em suma, com este projecto conseguiu-se contribuir para a investigação científica de três formas. Em primeiro lugar, conseguiu-se realizar uma comparação entre ferramentas de integração de aplicações *Oracle Retail*, o que será um grande apoio quando os programadores e consultores destes sistemas tiverem de escolher a melhor forma de integrar um sistema.

Em segundo lugar, conseguiu-se desenvolver um novo conceito de integração híbrida e dinâmica de aplicações *Oracle Retail*, ou seja, utilizar duas ferramentas de integração para a mesma interface e permitir ao cliente alternar entre ambas, dependendo das actividades e do estado do seu negócio.

Por fim, com o conceito criado conseguiu-se obter melhorias ao funcionamento do AIP, elevando assim este a um novo patamar de interesse perante os clientes.

7.1. Dificuldades Encontradas

Durante este projecto foram encontradas bastantes dificuldades. Algumas delas impediram que se obtivessem muito melhores resultados no projecto.

A primeira dificuldade encontrada foi o facto de não existir hardware disponível para que se realizassem testes mais rigorosos, obtendo assim melhores resultados. Ou seja, quando se realizou a comparação das 3 ferramentas de integração de aplicações *Oracle Retail* tinha sido mais

rigoroso realizar testes utilizando servidores dedicados às aplicações. Este problema reflecte-se mais nos resultados do RIB, visto que esta é uma aplicação que precisa de bastante memória e capacidade de processamento. Como os testes foram realizados num ambiente virtual os resultados desta aplicação podem ter ficado um pouco condicionados.

Outro dos problemas encontrados foi o facto de as aplicações *Oracle Retail* terem muitas dependências entre dados. Isso condiciona o facto de se conseguir criar informação aleatória, de forma a popular os sistemas com grandes quantidades de registos. Segundo pessoas especializadas na área das aplicações de retalho criar grandes quantidades de informação dinamicamente e inseri-la nos sistemas é extremamente difícil. Isto levou a que fosse sugerido arranjar-se informação real de um retalhista para que se pudesse ter muitos dados, mas isso também não foi possível, porque este tipo de informação não está disponível. Este problema condicionou o facto de se testar a melhoria realizada directamente no AIP. Assim, não se conseguiu testar concretamente que melhorias foram atingidas nas diversas métricas, com a nova implementação da interface melhorada do AIP.

7.2. Melhorias Possíveis no Trabalho Realizado

Uma das melhorias a este projecto seria melhorar a dinâmica de alteração entre as duas implementações da interface que envia a informação do AIP para o RMS. Com a melhoria implementada neste projecto, o utilizador tem de decidir que interface utilizar e para isso tem de alterar um parâmetro na base de dados, indicando qual utilizar. Seria bastante útil criar um mecanismo que automaticamente alternasse entre as duas interfaces consoante as actividades que se desenrolam no AIP.

Conclusões

Referências

- [1] (2000, Fevereiro) FAS. [Online].
<http://www.fas.usda.gov/info/factsheets/China/distribution.html>
- [2] Marshall L. Fisher, Ananth Raman, and Anna Sheen McClelland, "Rocket Science Retailing Is Almost Here - Are You Ready?", Harvard, 2000.
- [3] Retail Indicators Branch, U.S. Census Bureau. (2010, Abril) Economic Indicators.gov. [Online]. <http://www.economicindicators.gov/>
- [4] Hau L. Lee, *Retail Supply Chain Management: Quantitative Models and Empirical Studies*, Narendra Agrawal and Stephen A. Smith, Eds. Santa Clara, USA: Springer, 2009.
- [5] D. F. Ross, "Competing through supply chain management: creating market-winning strategies through supply chain partnerships", New York, 1997.
- [6] L.W. Stern, A.I. El-Ansary, A.T. Coughlan, and E. Anderson, "Marketing Channels", New Jersey, 2001.
- [7] J.R. Stock and D.M. Lambert, "Strategic Logistics Management", Singapore, 2001.
- [8] M. Christopher, "Logistics and Supply Chain Management", London, 1998.
- [9] Kate Vitasek. (2010, Fevereiro) Council of Supply Chain Management Professionals. [Online]. <http://cscmp.org/digital/glossary/glossary.asp>
- [10] Stock. J.R. and D.M. Lambert, "Strategic Logistics Management", Singapore, 2001.
- [11] Martin Gullberg and Peter Lundvall, "Retail Supply Chain Management", 2003.

REFERÊNCIAS

- [12] Oracle. (2008) Oracle Retail Realease 13: Enabling Transformation, from Integration to Insight. [Online]. <http://www.oracle.com/ocom/groups/public/@ocompublic/documents/webcontent/024018.pdf>
- [13] Oracle. Oracle Retail Products. [Online]. <http://www.oracle.com/industries/retail/productmap.html>
- [14] Wikipédia. (2010, Maio) Interface. [Online]. <http://pt.wikipedia.org/wiki/Interface>
- [15] Retailing Terms. [Online]. http://retail.about.com/od/glossary/g/purchase_order.htm
- [16] Oracle. (2009) Oracle Retail Advanced Inventory Planning Documentation - Implementation Guide. [Online]. http://download.oracle.com/docs/cd/E12546_01/aip/pdf/1311/aip-1311-imp.pdf
- [17] Wipro Technologies, "Documentação Avulsa", Maia,.
- [18] Oracle. (2009, June) Oracle Retail Integration Bus Documentation - Operations Guide. [Online]. http://download.oracle.com/docs/cd/E12461_01/131/operations_guide/rib-131-og.pdf
- [19] Oracle. (2009, June) Extract Transform and Load Documentation - Programmer's Guide. [Online]. http://download.oracle.com/docs/cd/B31321_01/retl/pdf/120/retl-120-pg.pdf
- [20] Susan McKibbon, "Oracle® Retail Integration Bus: Operations Guide", Oracle, Redwood, Application Documentation 2009.
- [21] Oracle. (2009) Oracle Retail Advanced Inventory Planning. [Online]. <http://www.oracle.com/applications/retail/library/data-sheets/advanced-inv.pdf>
- [22] Frederic P. Miller, Agnes F. Vandome, and John McBrewster, *Enterprise Application Integration: Enterprise application integration, Computer software, Business semantics management, Comparison of business integration software, Data integration.*: Alphascript Publishing, 2009.
- [23] Wikipédia. (2010, May) Database Trigger. [Online]. http://en.wikipedia.org/wiki/Database_trigger
- [24] Wikipédia. (2010, April) Standard streams. [Online]. http://en.wikipedia.org/wiki/Standard_streams
- [25] Wikipédia. (2010, June) Debugging. [Online]. <http://en.wikipedia.org/wiki/Debugging>