



CATARINA SOBREIRA REIS SANTOS

Licenciada em Engenharia Informática pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Relatório de Atividade Profissional

Dissertação para obtenção do Grau de Mestre em Engenharia Informática

Orientador: Professora Doutora Carla Maria Gonçalves Ferreira, Prof. Auxiliar, Faculdade de Ciências e Tecnologia da UNL – Departamento Informática

Júri:

Presidente: Prof. Doutora Susana Maria S. Nascimento M. Almeida
Vogais: Prof. Doutor Artur Miguel Andrade Vieira Dias
Prof. Doutora Carla Maria Gonçalves Ferreira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2014



CATARINA SOBREIRA REIS SANTOS

Licenciada em Engenharia Informática pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Relatório de Atividade Profissional

Dissertação para obtenção do Grau de Mestre em Engenharia Informática

Orientador: Professora Doutora Carla Maria Gonçalves Ferreira, Prof. Auxiliar, Faculdade de Ciências e Tecnologia da UNL – Departamento Informática

Júri:

Presidente: Prof. Doutora Susana Maria S. Nascimento M. Almeida
Vogais: Prof. Doutor Artur Miguel Andrade Vieira Dias
Prof. Doutora Carla Maria Gonçalves Ferreira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2014

Relatório de Atividade Profissional - Dissertação para obtenção do Grau de Mestre em Engenharia Informática

Copyright em nome de Catarina Sobreira Reis Santos, da FCT/UNL e da UNL

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Dedicatória

Aos meus pais, marido, filhos e filha.

Agradecimentos

Os meus sinceros agradecimentos à Professora Doutora Carla Ferreira que me acompanhou neste projeto e me orientou sempre que me deparei com dificuldades. Muito obrigado pela sua total disponibilidade.

Os meus agradecimentos à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa e ao seu Corpo Docente pelo excelente ensino e preparação fornecido no decorrer da Licenciatura, preparação essa que é suporte e alavancagem de todos os meus conhecimentos técnicos.

Os meus agradecimentos ao meu pai que me incentiva em todos os projetos e me auxilia a cada tomada de decisão.

Muito obrigado a todos.

Resumo

O presente Relatório pretende, de acordo com o Despacho n.º 20/2010 do Conselho Diretivo da FCT/UNL relativo ao processo de obtenção do grau de Mestre por licenciados pré-Bolonha, apresentar o meu percurso profissional dedicado desde 1998 aos sistemas de informação numa vasta variedade de tecnologias.

Tendo obtido o meu grau de licenciatura em Engenharia Informática na Faculdade de Ciências e Tecnologia no ano de 1997, tenho estado desde então sempre em funções na área de análise, desenho e implementação de projetos.

Entre 1997 e 1998 estive em funções numa seguradora onde exerci o papel de analista e programadora em sistemas AS/400 com recurso a RPG III e a Synon/400 – ferramenta case em AS/400 geradora de código RPG.

Em 1998 fui contratada por uma entidade financeira para trabalhar também em AS/400 e RPG, seja ele RPG III e RPG IV, complementado por Plex/400, uma ferramenta Case Windows geradora de código RPG III.

Em 2001 iniciei funções na mesma instituição a trabalhar com Oracle, .Net e Java com interações com AS/400 via JDBC e MQSeries. Esta multidisciplinaridade e multi-sistema é sem dúvida a chave do meu curriculum.

Não tendo no decorrer destes anos realizado qualquer formação, para além de um certificado da IBM em RPG IV, considero que a capacidade de aprendizagem em autodidata de toda esta multidisciplinaridade só é possível graças ao tipo de ensino e à abordagem efetuada pela Licenciatura em Engenharia Informática da Faculdade de Ciências e Tecnologia.

No decorrer do meu percurso profissional verifiquei que são poucos os que conseguem entender as vicissitudes de tantos sistemas diferentes e os seus prós e contras. Na implementação de um projeto nenhuma parte deve ser uma caixa negra e por isso os conhecimentos em tantas áreas diferentes tem sido sempre a minha mais-valia.

Para este relatório resolvi abordar em maior detalhe aquele que é o projeto de que mais me orgulho pois proporcionou-me o maior enriquecimento profissional quer na vertente tecnológica, onde aprendi ou aprofundei Oracle, .Net, Objetos COM, MQSeries, Java e mais tarde Z/OS-Cobol, quer na vertente funcional, financeira e matemática. Não tenho dúvida ter sido este o projeto que mais visibilidade me deu e que me continua a servir de cartão de visita.

Continuo a trabalhar na mesma área, agora noutra instituição, por via de aquisição, e sou da opinião que a formação que obtive na Licenciatura em Engenharia Informática na FCT é chave para a capacidade de adaptação e aprendizagem que continuo a utilizar nas minhas funções diariamente.

Palavras chave: MQSeries, AS/400, Z/OS, .Net, Java, Oracle

Abstract

The present report intends, accordingly to the Dispatch number 20/2010 of FCT/UNL's Directing Board relating to the process of achieving the Master Degree for licensees pre-Bologne, to show my professional path dedicated since 1998 to Information Systems in a variety of technologies.

Since I achieved my License Degree from Faculdade de Ciências e Tecnologias in 1997 I have been working on Informational System projects, from analysis to design and implementation.

Between 1997 and 1998 I worked in an insurance company as an analyst and programmer for AS/400 systems based on RPG III and using Synon/400 – a case tool which generates RPG source code.

In 1998 I was hired by a Financial Entity to work also on AS/400 with RPG, either RPG III or RPG IV, and using also PLEX/400, a Windows based case tool generating code for AS/400 RPG.

In 2001, I started to work within the same Institution but with Oracle, .Net and Java and interacting with AS/400 through JDBC or MQSeries. This variety of systems and disciplines are for sure my curriculum key asset.

As I did not participate in any kind of Academic or other type of training, besides an IBM certificate in RPG IV, I consider that my self-training capacity in all of the presented systems was only possible thanks to the teaching process and technics lectured by Informational Engineering License by FCT/UNL.

As my professional path evolved I found out that there are few who can understand the strange aspects of each system and its pros and cons. During a project there should be no black boxes and therefore the variety of knowledge I gathered have been my key asset.

For this report I decided to emphasize the project I am most proud of. This project gave me a professional enrichment either technologically speaking, where I have learnt or mastered Oracle, .Net, Objetos COM, MQSeries, Java and later on Z/OS-Cobol, and functionally speaking, financial and mathematical. I am sure this was the project that gave me more visibility and still is my greeting card.

At the present I am still working in the same area, but in a different institution, due to an acquisition. It is my opinion that the academic training I got on FCT is the key for the adapting and learning abilities that I still use on my everyday tasks.

Keywords: MQSeries, AS/400, Z/OS, .Net, Java, Oracle

Índice de Matérias

1. Introdução.....	1
2. Percurso Profissional - Valores Mobiliários.....	3
2.1. Contexto Histórico.....	3
2.2. A globalização dos Mercados de Capitais.....	6
2.2.1. Introdução Funcional.....	6
2.2.2. Abordagem Técnica.....	9
2.2.2.1 Do Cliente até à Corretora.....	9
2.2.2.2 Da Corretora até à Euronext	14
2.2.2.3 Da Euronext até à Corretora	18
2.2.2.4 Interação entre a Clearnet e a Instituição Membro de Compensação.....	19
2.2.2.5 Do Membro liquidatário até à Casa de Liquidação - Interbolsa.....	24
2.2.2.6 Visão Global Tecnológica.....	26
2.2.3. Migração tecnológica em 2011.....	27
3. A Tecnologia ao serviço dos Mercados de Capitais – O futuro.....	29
4. Conclusões.....	31
5. Referências.....	33
6. Anexos.....	35

Índice de Figuras

Figura 2.1 – Fluxo de Informação entre Euronext, Clearnet e Interbolsa.....	4
Figura 2.2- Circuito de liquidação de um Cliente "Externo".....	8
Figura 2.3 - Cliente com liquidação Externa.....	8
Figura 2.4– Fluxo de validação de saldos.....	10
Figura 2.5 – Fluxo de validação de Ordem num terminal.....	11
Figura 2.6 – Do terminal até ao Mercado e seu retorno.....	12
Figura 2.7 – Do Oracle até ao Mercado e seu retorno.....	13
Figura 2.8 – Estrutura de Queues do circuito interativo.....	21
Figura 2.9 - Estrutura de queues suporte ao protocolo batch.....	22
Figura 2.10 – Estrutura de Contas global.....	24
Figura 2.11 – Visão Global.....	26

Índice de Tabelas

Tabela 2.1 - Alguns dos Campos informados no evento NeworderSingle.....	15
Tabela 2.2 - Campos informados no evento OrderCancelRequest.....	18

Glossário

DVP	Deliver versus Payment
RVP	Receive versus Payment
Interbolsa	Casa de liquidação portuguesa
Global settlers	Entidades que fornecem o serviço de compensação multi-mercado
Clearnet	Entidade que fornece o serviço de compensação para os mercados Euronext.
Banco de Portugal	Banco Central português responsável pelos pagamentos financeiros ao estrangeiro
CCP	Conetividade com a plataforma de negociação Euronext via terminal
CAPI	Conetividade direta e exclusiva com a plataforma de negociação Euronext
MAPI	Conetividade com a plataforma de negociação Euronext via Vendor acreditado utilizando um circuito partilhado por vários Membros
ISV	Independent Software Vendor
Visual Trader	ISV contratado para conetividade à Euronext
CCW	Conetividade com a plataforma de compensação da Clearnet via terminal
CAP	Conetividade direta e exclusiva com a plataforma de compensação da Clearnet
MAP	Conetividade com a plataforma de compensação da Clearnet via Vendor acreditado utilizando um circuito partilhado por vários Membros
Slib	ISV contratado para conetividade à Clearnet
Margem	Valor depositado diariamente como colateral ao risco de falha de liquidação
C21	Sistema de Compensação da Clearnet
Trade leg	Registo do negócio no sistema da Clearnet
Position account	Conta de posição a que se “aloca” cada trade leg
Netting	Valor agregado de compras menos vendas
Posting	Ação de atribuir uma trade leg ou parte dela a uma conta de posição
Give-up	Pedido de transferência de responsabilidade sobre uma trade-leg para outro Membro
Take-up	Aceitação do Pedido de transferência de responsabilidade sobre uma trade-leg efetuado por outro Membro
OIC	Outras Instituições de Crédito
SLRT	Sistema de encontro de Contas entre Membros da casa de liquidação Interbolsa
IIS	Internet Information Services
JSP	Java server Pages

1. Introdução

Nesta dissertação pretendo descrever o percurso profissional efetuado no decorrer dos últimos 15 anos ao serviço de entidades financeiras na área de Mercados de Capitais.

Será abordado mais detalhadamente o projeto efetuado em 2003 de transferência da Bolsa de Valores de Lisboa para Paris.

Neste ano de 2003 efetuou-se uma reestruturação na forma de abordar o circuito de Negociação, de Compensação e de Liquidação de Valores Mobiliários tendo em vista a globalização dos Mercados de Capitais. Pretendeu-se com esta implementação reduzir os riscos de falhas e incumprimentos inerentes à expansão.

O mercado de Lisboa com este projeto conseguiu a sua internacionalização e a equiparação aos mercados Europeus. A integração destes mercados com o Americano ficou a um passo, tendo à data da elaboração deste relatório já sido efetuada e também desmantelada.

Será abordado com detalhe a implementação tecnológica utilizada no projeto e as suas vertentes opcionais e obrigatórias bem como as evoluções que já foram efetuadas desde então.

Para finalizar será apresentado um levantamento das potencialidades da utilização da computação ao serviço de modelos matemáticos algorítmicos que à semelhança das redes neuronais permitem evoluir com o comportamento do mercado e autoajustar-se aos seus resultados. Pretende-se com este tipo de algoritmos, que apesar de complexos não carecem de infraestruturas inacessíveis, maximizar resultados e aprender a compensar as perdas.

2. Percurso Profissional - Valores Mobiliários

2.1. Contexto Histórico

Desde 1998 e ao longo de 5 anos mantive um percurso profissional sempre relacionado com mercados, sejam eles cambiais ou de capitais (Fundos de Investimento e outros Valores Mobiliários como ações).

Em 2003, aquando do lançamento internacional do projeto Euronext, foi-me entregue a responsabilidade da implementação de todo o sistema informático suporte à função de Membro de Negociação e de Membro de Compensação do Mercado de Lisboa da entidade onde trabalhava. Neste projeto crucial ao negócio de trading da instituição, exerci, para além do papel de interpretação e definição de todo o circuito funcional, o de análise e desenho do sistema que o suportaria. Todo o desenho tecnológico e sua montagem foi da minha autoria sendo que apenas subcontratei a uma empresa externa a programação das *gateways* que apresentarei mais à frente. Mesmo neste caso a verificação/validação da correta programação do pacote entregue, quer em termos de funcionamento quer em termos de fiabilidade e tratamento de erros, foi da minha responsabilidade.

A implementação técnica efetuada teve como âmbito os sistemas internos da Instituição e as suas interações e dependências dos *Vendors* contratados e dos seus sistemas.

Este projeto nasceu da aquisição, efetuada em data anterior a 2003, das quatro principais praças de negociação europeias por parte da Euronext. No ano de 2003 foi efetuada a desmobilização do seu funcionamento local para um único espaço central em França.

As quatro praças adquiridas foram Lisboa, Paris, Bruxelas e Amesterdão.

A desmobilização das quatro praças de negociação levou a uma análise profunda do *modus operandis* de cada uma com o intuito de uniformizar o circuito.

Até esta data cada país dispunha da sua casa de liquidação – em Lisboa a **Interbolsa** – e cada mercado tinha que se relacionar com pelo menos a casa de liquidação do seu país para dar lugar ao pagamento dos negócios efetuados. A **Euronext**, ao integrar quatro mercados, passou a ter que interagir com várias casas de liquidação pelo que este circuito tornava-se altamente complexo e oneroso.

Surge uma nova entidade Central que se responsabiliza pela Compensação de todos os negócios realizados nos mercados Euronext – a Clearnet.

Esta entidade, circuito único para onde a Euronext teria de enviar os negócios realizados, garante aos vários intervenientes o correto pagamento do seu negócio, sejam eles o comprador ou o vendedor. Assim, cada um terá que efetuar um acerto de contas apenas com esta entidade ao invés de ser obrigado a conhecer a contraparte do negócio e ter que se relacionar com todos os intermediários financeiros no mercado.

Em suma estamos perante três entidades relevantes: a praça de negociação (Euronext), a praça de compensação (Clearnet) e a casa de liquidação (Interbolsa no caso de Lisboa).

A Figura 2.1 abaixo apresenta de uma forma simplificada o fluxo de informação entre estas entidades.

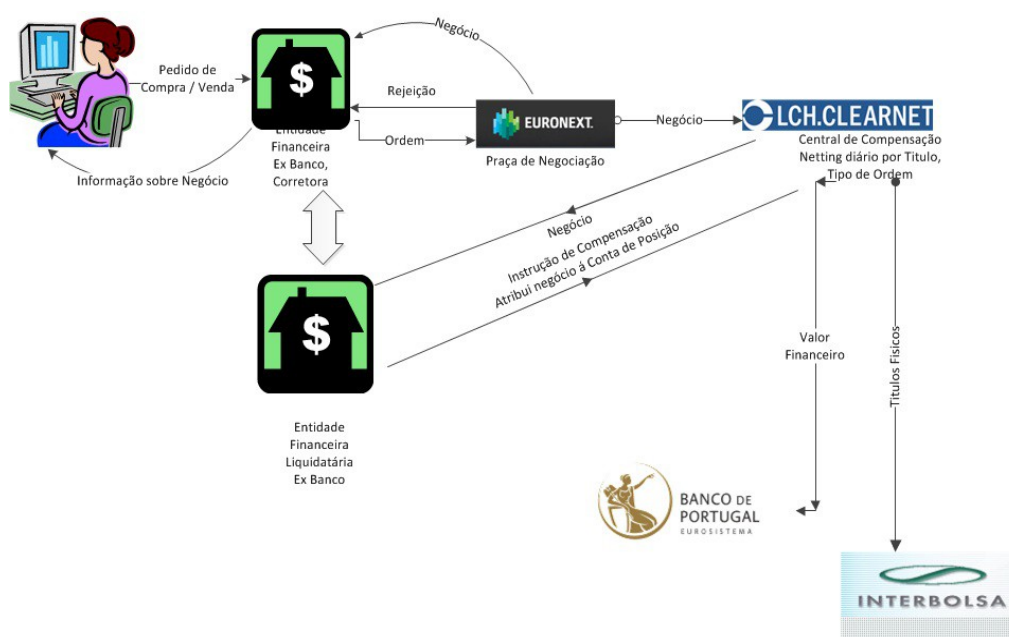


Figura 2.1 – Fluxo de Informação entre Euronext, Clearnet e Interbolsa

O cliente dá uma instrução ao seu Corretor, que sendo membro de Negociação comunica-a à praça de negociação. Ao ser realizado negócio este é comunicado de volta ao Corretor e também à praça de Compensação. Na etapa seguinte o Corretor deve comunicar o negócio à sua Entidade Liquidatária (membro de Compensação) ao mesmo tempo que esta toma conhecimento do mesmo pela praça de Compensação. A Entidade Liquidatária fornece à praça de Compensação as instruções para a Liquidação para que esta posteriormente proceda em

conformidade a nível financeiro (Banco de Portugal) e a nível físico (títulos na casa de liquidação – Interbolsa).

Cada uma destas entidades exige a qualquer Intermediário Financeiro que queira interagir com ela um processo rigoroso de admissão como Membro. A Entidade Financeira onde realizei este projeto optou por ser Membro das três Casas sendo que apenas as duas primeiras admissões são relevantes para este Relatório. Esta autorização conseqüentemente comporta a necessidade de conectividade “física” sobre a qual me debruçarei nos capítulos seguintes .

2.2. A globalização dos Mercados de Capitais

2.2.1. Introdução Funcional

Como referi atrás a Entidade Financeira em questão optou por ser Membro quer da Praça de negociação de Lisboa (Euronext), no papel de Corretora, quer da Praça de Compensação - Clearnet - como Entidade Compensadora.

No processo de admissão a Membro da Euronext são disponibilizadas três modalidades diferentes de conectividade. A primeira modalidade – **CCP** – corresponde à utilização de um terminal isolado com ligação direta à plataforma de negociação e com custos reduzidos. O **CAPI**, a modalidade com custos mais elevados, onde a Euronext oferece um ponto de conectividade exclusivo e direto que permite ao membro receber todas as mensagens comunicadas pelo mercado. Esta modalidade para além dos custos mensais elevados obriga a um desenvolvimento exaustivo dos inúmeros fluxos de mensagens existentes. Por último, a solução **MAPI** onde a Euronext permite a um **ISV** acreditado – **Independent Software Vendor** – receber todo este fluxo de mensagens e distribuí-lo a vários membros. Assim, os custos de ligação sendo partilhados por vários membros são mais reduzidos e o desenvolvimento necessário a essa conectividade é simplificado pelo *Service Vendor*. A opção por nós escolhida foi a solução MAPI sendo que o ISV contratado foi a **Visual Trader**.

Quando se regista a adesão como Membro de Negociação é obrigatória e imprescindível a indicação de qual o Membro de Compensação responsável pelos negócios introduzidos.

No plano da adesão como Membro Compensador a Clearnet também proporcionou aos membros um pacote similar de conectividade com três modalidades **CCW**, **CAP** e **MAP**. Neste caso a acumular ao custo de conectividade a Clearnet optou por cobrar uma taxa conforme o volume de negócios compensado pelo Membro sendo que a negócios por conta de outrem (clientes) é aplicada uma taxa mais elevada que em negócios em nome próprio, considerados de menor risco de falha de liquidação. Esta taxa é denominada “**Margem**” e é aplicada diariamente devendo encontrar-se depositada no Banco de Portugal. Funciona como uma caução.

Neste caso a opção escolhida pela Entidade Financeira em questão foi a solução MAP com a contratação de um Independent Service Vendor, a **Slib**. Esta empresa de origem Francesa, criada a partir do Banco BNP Paribas, disponibiliza um protocolo de comunicação para agilizar a interação com o sistema de Compensação da Clearnet - **C21**.

A Clearnet receberá informação do negócio efetuado pelo Membro de negociação na praça da Euronext e criará um registo – denominado **trade leg** – afeto ao Membro de Compensação que lhe está associado. Desse momento em diante e até ao final do dia deverá esta **trade leg** ser atribuída a um conta de posição (**position account**). No final do dia, é calculado o somatório agregado de cada position account por Título (Valor mobiliário) - **netting** – valor apurado em quantidade de títulos (compras menos vendas) e valor financeiro (produto da quantidade comprada aplicado ao preço do negócio de compra deduzido do produto da quantidade vendida pelo preço do negócio de venda).

Estes valores apurados serão posteriormente utilizados para calcular o valor da **Margem** a depositar no Banco de Portugal bem como serão informados à Casa de Liquidação respetiva – Interbolsa e Banco de Portugal.

A estrutura de contas contratada por nós junto da Clearnet foi simplificada ao ponto de unicamente separar negócios por conta de clientes de negócios da Carteira Própria (negócios detidos pela própria Entidade Financeira), já que os valores elevados de negócios Próprios inflacionariam o valor de Margem a depositar diariamente.

A ação de atribuir um negócio (**trade leg**) a uma **position account**, atrás referida, é denominada de **Post** ou **Posting**.

No entanto para as Corretoras existe uma faixa de clientes que as escolhe como entidade de negociação mas não adere à sua entidade de Liquidação associada. São eles clientes que utilizam a Corretora como meio de chegar ao mercado mas liquidam em outras **OIC** (Outras instituições de Crédito).

Nestes casos, a entidade liquidatária por defeito da corretora não tem interesse em ser responsável pela execução do circuito de compensação e muito menos pela etapa de liquidação. Deve por isso, ao invés de efetuar um **posting** da **trade leg** efetuar um **give-up**. Esta instrução traduz-se num processo de pedido de transferência do negócio para outro membro compensador mas exige que seja feita a aceitação por parte do membro recetor, um **take-up**.

Infelizmente esta aceitação raramente é efetuada atempadamente e por isso o membro mantém-se responsável pelo pagamento do negócio a nível da compensação. Nesta situação o acerto de contas terá que ser feito a nível da casa de liquidação recorrendo a um sistema denominado **SLRT** que recorre a instruções **RVP / DVP** para realizar a troca de títulos e de valor financeiro.

A Figura 2.2 apresentada abaixo representa exatamente o cenário em que o cliente utiliza uma corretora mas não recorre ao seu membro compensador. Neste caso, ao nível da compensação é feito um **posting** e ao nível da liquidação um RVP/DVP.

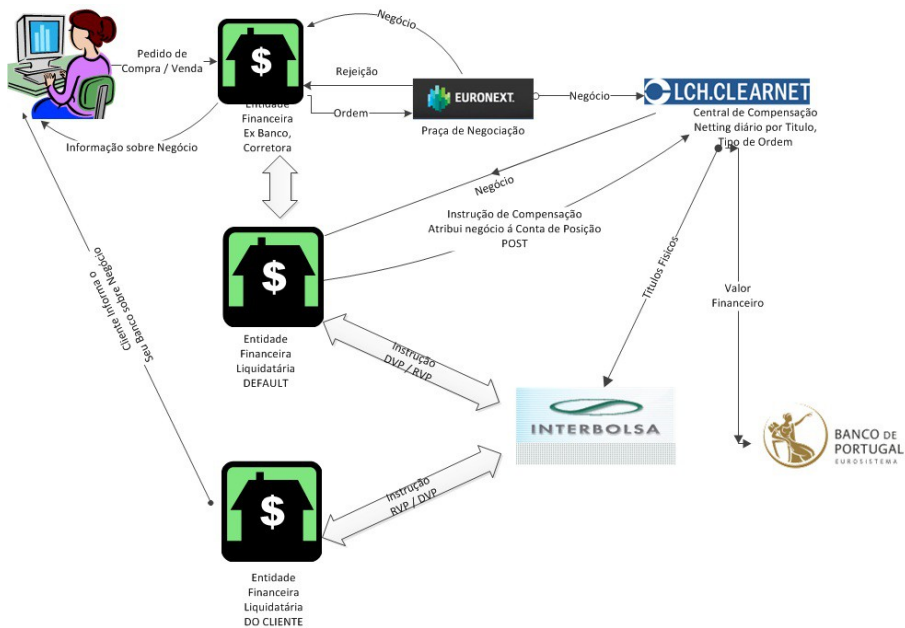


Figura 2.2 - Circuito de liquidação de um Cliente "Externo"

Na Figura 2.3 abaixo o cliente também não liquida no membro default mas este emite uma instrução de *give-up* que é aceite pelo membro destino. Este após *take-up* realiza um *posting* na sua estrutura de contas e torna-se responsável pelo pagamento do negócio logo a nível da compensação (entre membro e Clearnet), não havendo necessidade de acerto de contas a nível da casa de liquidação.

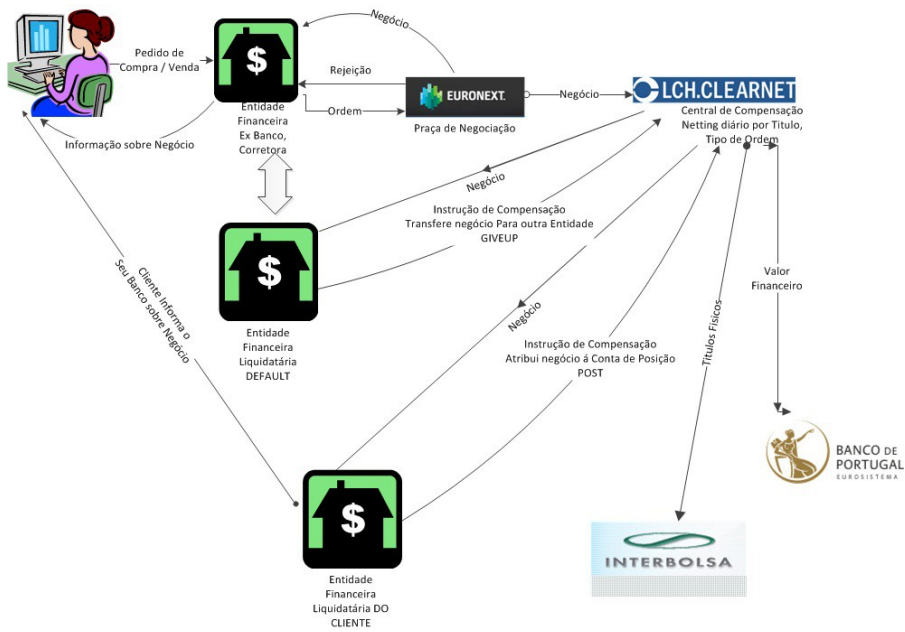


Figura 2.3 - Cliente com liquidação Externa

2.2.2. Abordagem Técnica

Neste capítulo apresentarei as especificações técnicas das peças implementadas para cobrir cada um dos fluxos apresentados nos esquemas anteriores.

Como já foi explicado existe uma entidade, Visual Trader, entre a corretora e o mercado.

Para se poder compreender os vários circuitos técnicos que contactam com a Visual Trader temos que saber que o seu software (terminais de negociação) é desenvolvido em .Net [1] recorrendo a objetos COM que encapsulam e simplificam o protocolo Fix utilizado internacionalmente para comunicações sobre mercado de capitais [2]. Este software funciona através da subscrição de eventos pertinentes pelo que permite o encaminhamento de mensagens bem como a sua receção.

Os eventos subscritos são:

- StatusEventHandler
- ExecutionReportEventHandler
- OrderCancelRejectEventHandler
- NewOrderEventHandler

2.2.2.1 Do Cliente até à Corretora

Na instituição onde elaborei este projeto existiam quatro canais possíveis de acesso do cliente à plataforma de negociação. O homebanking, a Banca Telefónica e os Balcões tradicionais consistiam em três plataformas assentes na mesma estrutura tecnológica, um ambiente web implementado em Windows assente em IIS (Internet information Services) recorrendo a paginas JSP numa estrutura de três camadas. A camada Web é desenvolvida em JSP e comunica com a camada transacional via Jrun 3.1. A camada transacional desenvolvida em Java por sua vez comunica com bases de dados recorrendo a JDBC [3,12] e com o sistema Central bancário recorrendo a MQSeries [4].

O sistema Central disponibiliza um lote de transações pergunta-resposta passíveis de ser invocadas via MQSeries, um exemplo destas transações é a consulta de saldos financeiros de Depósitos à Ordem referida mais à frente.

No decorrer do período de 2001 a 2003, efetuei variados desenvolvimentos nestas linguagens e ambientes tendo obtido know-how que suportou as necessidades do projeto que descrevo.

A Corretora dispõe de um sistema de suporte à negociação (usualmente denominado front-office) assente na mesma estrutura do sistema de Backoffice, denominado de sistema de Custódia. Este sistema é baseado em bases de dados Oracle com recurso a Stored procedures, front-end em Oracle Forms e serviços complementares em .Net.

O sistema de Custódia de Valores Mobiliários é onde uma entidade liquidatária regista os saldos de títulos depositados após cada operação de compra ou venda. É sobre este sistema que é aplicado todo o tipo de operações que se podem exercer sobre um título, como por exemplo um Rendimento, uma fusão, etc.

É também sobre este sistema que a cada operação de venda se deve garantir a existência de saldos físicos que a cubram. Esta validação é implementada com Stored Procedures.

Por outro lado, no caso do cliente pretender efetuar uma compra de Valores mobiliários cabe ao sistema de custódia garantir a existência de saldo financeiro (\$) que como já referi está registado no sistema Central (AS/400).

O sistema de Bases de Dados Oracle disponibiliza para acesso remoto a possibilidade de invocar pedidos HTTP que utilizando a estrutura tecnológica implementada para o homebanking e para os canais já referidos permite a invocação das transações de consulta de saldos do sistema Central do Banco.

Assim, o mesmo formato de tratamento (stored procedures) permite camuflar a complexidade e disparidade existente entre uma validação física e uma financeira. A Figura 2.4 representa o circuito efetuado por cada pedido de validação.

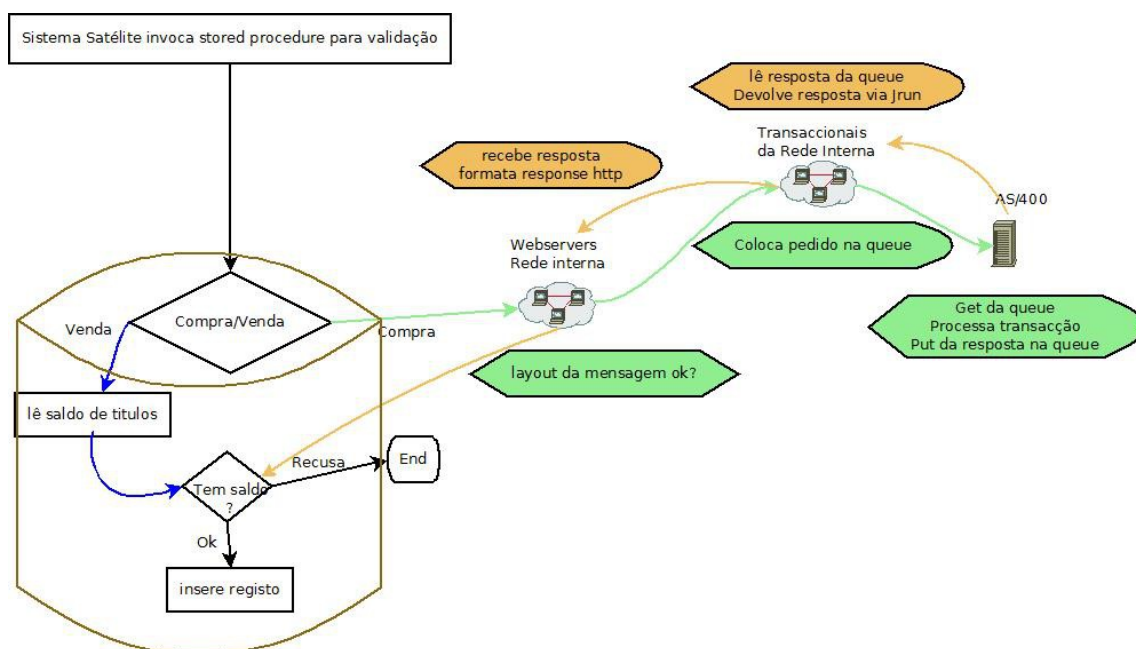


Figura 2.4 – Fluxo de validação de saldos

Assim, qualquer um dos três canais apresentados registra o pedido de Ordem de Bolsa no Front-Office (Oracle), garantindo a necessária validação de saldos (risk management) quer sejam eles físicos ou financeiros. Este registo na tabela de Oracle é feito com um estado específico (valor RT) por forma a que as etapas por que passa sejam auditáveis.

Como já referi a camada transacional é implementada em Java [5], como tal as invocações às funções de MQSeries são feitas recorrendo às funções genéricas de messaging que pela lógica Orientada a Objetos é traduzida numa invocação à função nativa da IBM (as classes da biblioteca da IBM herdam da classe nativa).

Como já foi referido todos estes canais assentam na mesma estrutura tecnológica e inclusivamente partilham-na.

Resta-nos o último canal disponibilizado, a Sala de Mercados. A Sala de Mercados por causa da sua especificidade de funcionamento requer um sistema mais célere de acesso ao mercado. Por esta razão são utilizados terminais de negociação. Estes terminais são workstations de acesso direto ao ISV contratado, neste caso a Visual Trader. A intenção é ter o mínimo de relacionamentos que possam desacelerar o processo.

Mas temos na mesma que acautelar o *Risk Management* (validação de saldos). Para o efeito a Visual Trader disponibilizou juntamente com o seu terminal um circuito de validação/pedido de autorização das Ordens executado antes da sua submissão ao mercado. Este circuito assenta no reconhecimento, por parte do servidor que recebe os pedidos do terminal, da necessidade de invocação de uma DLL (*dynamic linked library*) para pedido de autorização. Esta DLL por sua vez irá invocar a Stored Procedure de validação de Saldos já referida atrás.

A Figura 2.5 explica em maior detalhe como funciona esta invocação.

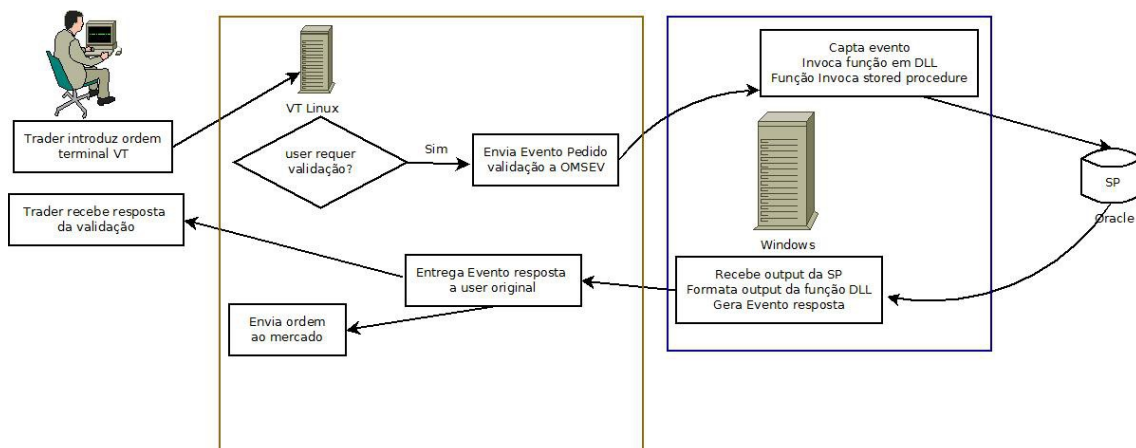


Figura 2.5 – Fluxo de validação de Ordem num terminal

Resumindo, no que diz respeito à introdução de ordens poderemos ter duas vias:

- Canais web da instituição → Os parâmetros da Ordem introduzidos via web são seguidos da validação de saldos físicos e financeiros e registo do pedido no sistema de front-office (Oracle) com Estado RT (recepção terminada)
- Terminal de negociação na Sala de Mercados → Os parâmetros da Ordem indicados pelo cliente ao trader (operador da Instituição) são introduzidos no terminal Visual Trader. Esta introdução é seguida da validação de saldos físicos e financeiros que se aceite autoriza a submissão direta ao mercado via canais Visual Trader.

A Figura 2.6 apresenta de uma forma simples a relação entre o servidor VT, o Oracle e o Mercado quando o pedido é introduzido no terminal de negociação.

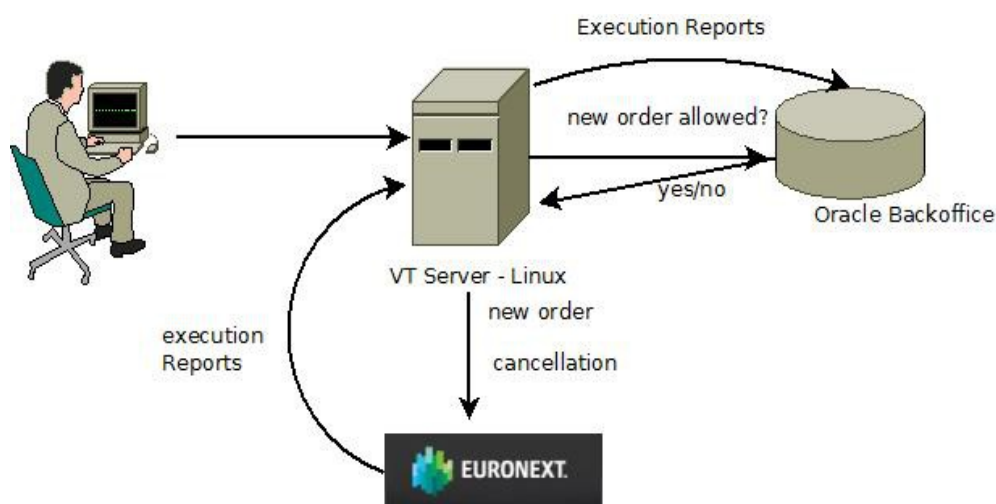


Figura 2.6 – Do terminal até ao Mercado e seu retorno

Com a Figura 2.7 abaixo pretendo explicar o que sucede com cada pedido, independentemente do canal utilizado para a introdução da ordem, e como cada fluxo chega ao mercado e ao Backoffice (Oracle).

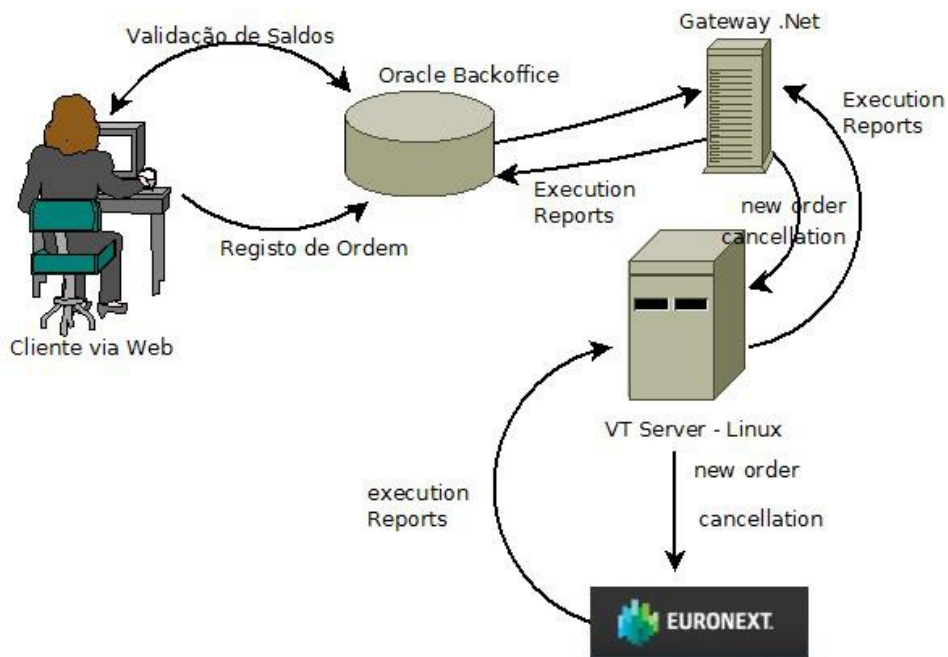


Figura 2.7 – Do Oracle até ao Mercado e seu retorno

Como se pode perceber por este esquema, todo o contacto efetuado com o mercado e vindo deste é feito via Servidor Linux VT. Este servidor ao reconhecer a mensagem do mercado efetua a sua tradução e entrega-a aos utilizadores parametrizados com acesso à sua receção. Cada utilizador tem como parâmetro a lista dos mercados autorizados bem como dos utilizadores cujas ordens podem aceder. Cada utilizador pode ver todas as ordens ou só as suas.

Por esta razão, o utilizador normal do terminal (trader) só recebe as mensagens referentes às suas ordens. O utilizador OMSEV só recebe as mensagens que lhe dizem respeito ou seja, os pedidos de validação. Mais á frente veremos que o utilizador API (registado para utilização na gateway) recebe todas as mensagens de todos os utilizadores exceto OMSEV.

Por outro lado, o pedido do Cliente pode ser um cancelamento de uma ordem. Neste caso e apenas no canal Web, é inserido também um registo na mesma tabela oracle, referindo que o tipo de pedido é “Cancelamento”. Nestes pedidos não se efetua qualquer validação de saldo. Este registo também apresenta o mesmo estado RT, donde também será processado pela gateway .Net para envio ao mercado.

2.2.2.2 Da Corretora até à Euronext

Este capítulo irá debruçar-se sobre as instruções introduzidas via canais WEB e que após execução do circuito apresentado no capítulo anterior estão residentes no sistema de front-office a aguardar tratamento (estado RT).

A Visual Trader disponibilizou uma API .Net que, funcionando nos mesmos moldes do terminal de negociação, encapsula e simplifica o protocolo Fix utilizado internacionalmente para comunicações sobre mercado de capitais. Funcionando como um objeto COM subscreve os eventos pertinentes ao encaminhamento das ordens bem como à receção das mensagens divulgadas pelo mercado . Neste capítulo apresentarei apenas os eventos no sentido de saída da instituição para o mercado.

Esta gateway no arranque estabelece um login ao servidor Visual Trader. Este login valida não só o user especificado (user API) mas também o nome da workstation/server que está a tentar estabelecer a ligação. É mantida uma thread dedicada unicamente a garantir que esta conexão se mantém viva, recuperando-a em caso de quebra.

Foi implementada também uma thread dedicada à transmissão de novas ordens e cancelamentos sendo que a ordem de entrada dos pedidos, por questões legais, é fulcral.

Esta ordenação foi conseguida através da monitorização constante de uma *queued table* [6].

No capítulo anterior verificamos que a ordem/cancelamento tinha sido inserida num tabela do oracle num estado específico. O trigger *after update* desta inserção responsabiliza-se por, neste caso, inserir o mesmo pedido na *queued table* utilizada pela gateway.

Assim, a gateway acede a esta *queued table* e por cada pedido recebido formata um evento “new order” ou “new canc” que entrega á Visual Trader. Por uma questão de monitorização e auditoria é registado no log da ordem o instante em que foi feita esta entrega. A instrução foi “transmitida mas ainda não rececionada pelo mercado”.

Aquando da criação da “new order” é incluída na mensagem ao mercado a indicação do identificador da ordem no sistema do corretor para posterior associação, uma vez que o mercado irá lhe atribuir um número próprio. Assim a qualquer momento é possível manter a coerência entre o estado da ordem no mercado e o estado da instrução originária no corretor.

O evento “new order” corresponde à mensagem FIX NewOrderSingle e contém entre outros os campos referidos na Tabela 2.1 abaixo

Tabela 2.1 - Alguns dos Campos informados no evento NeworderSingle

ClOrdID	Obrigatório	Identificador da ordem no sistema origem
ExecInst	opcional	Qual a instrução a efetuar Diz ao mercado como deve encarar a ordem
MinQty	opcional	Quantidade mínima a ser transacionada
MaxFloor	opcional	Quantidade máxima visível ao motor de negociação
Instrument	Obrigatório	Inclui vários Campos necessários à identificação do título (ex. ISIN e menomónica)
ExDestination	opcional	Praça de negociação ex: LS=Lisboa
Side	Obrigatório	Compra (1) ou Venda (2)
TransactTime	Obrigatório	Timestamp da criação da ordem
OrderQty	Obrigatório	Quantidade da ordem
OrdType	Obrigatório	Tipo de ordem. Exemplos de valores: 1 = Market (Ordem ao melhor) 2 = Limit (ordem com preço limite) 3 = Stop (stop order ao melhor) 4 = Stop limit (stop order com preço limite)
Price	opcional	Preço. Obrigatório se Ordtype for 2 ou 4
StopPx	opcional	Preço Stop. Obrigatório se ordtype for 3 ou 4. A ordem só fica viva no mercado quando é atingido o preço stop
Currency	opcional	Moeda
TimeInForce	opcional	Período de validade da ordem
ExpireTime	opcional	Timestamp de expiração da ordem
OrderCapacity	opcional	Indica o papel pretendido para a ordem. Valores possíveis: A = Agency (ordem para clientes) G = Proprietary (carteira própria) I = Individual P = Principal (Note for CMS purposes, Principal includes Proprietary) R = Riskless Principal

		W = Agent for Other Member Permite informar automaticamente a Clearnet a respeito do enquadramento do negócio
--	--	--

O tratamento do envio da ordem é feito “grosso modo” pelo seguinte bloco de código

```
private void NewOrderEventListener(object sender, Order order)
{
    Order.Status processing = Order.Status.processing;
    try
    {
        _logger.Info("Order sending ( " + order.ToString() + " )");
        this.vtTrader.SendOrder(order.Type, order.Data);
        processing = Order.Status.sent;
        _logger.Info("Order sent");
    }
    catch (InvalidOperationException exception)
    {
        _logger.Error("Order sending Error", exception);
        processing = Order.Status.errorProcessing;
    }
    catch (ArgumentException exception2)
    {
        _logger.Error("Order Fields Error", exception2);
        processing = Order.Status.errorProcessing;
    }
    finally
    {
        try
        {
            _logger.Info("Order Status reporting ( Status: " + processing + " )");
            this.mqseriesTrader.ReportOrderStatus(processing);
            _logger.Info("Order Status reported");
        }
        catch (Exception exception3)
        {
            _logger.Error("Order Status reporting Error", exception3);
        }
    }
}
```

onde this.vttrader.sendOrder invoca

```
[MTAThread]
public void SendOrder(short type, Dictionary<int, string> data)
{
    OMSCOM_Trading_Order order = this.CreateOrder(data);
    try
    {
```

```

        this.SendOrder(order, type);
    }
    finally
    {
        order.Clear();
        for (int i = 1; i > 0; i = Marshal.ReleaseComObject(order))
        {
        }
        order = null;
    }
}

```

onde por sua vez SendOrder é

```

[MTAThread]
private void SendOrder(OMSCOM_Trading_Order order, short type)
{
    int pError = 0;
    _logger.Debug(string.Concat(new object[] { "Order sending - ", order.ToString(), " Type - ", type }));
    if (!((ConnectionStatus) this.vtTrader.Status).Equals(ConnectionStatus.LoggedIn))
    {
        throw new InvalidOperationException("No connection to Service");
    }
    this.vtTrader.SubmitOrder(order, type, out pError);
    if (pError != 0)
    {
        throw new InvalidOperationException(string.Concat(new object[] { "Error (", pError, ") received when trying to Submit Order Type: ", type, "Order: ", order.ToString() }));
    }
    _logger.Debug(string.Concat(new object[] { "Order sent - ", order.ToString(), " Type - ", type, "Error - ", pError }));
}

```

onde SubmitOrder já é uma função da API da Visual Trader.

É importante verificar que a construção efetiva da Mensagem Ordem é feita no método CreateOrder.

```

[MTAThread]
private OMSCOM_Trading_Order CreateOrder(Dictionary<int, string> data)
{
    int pError = 0;
    _logger.Debug("Order creating");
    OMSCOM_Trading_Order order = this.vtTrader.CreateOrder();
    foreach (KeyValuePair<int, string> pair in data)
    {
        _logger.Debug(string.Concat(new object[] { "Field Number: ", pair.Key, "Field Value: ", pair.Value }));
        order.set_Field(pair.Key, pair.Value, pair.Value.Length, out pError);
        if (pError != 0)
        {
            throw new ArgumentException("Error ( " + pError + ") received when trying to set Order Field", string.Concat(new object[] { "Field Number: ", pair.Key, "Field Value: ", pair.Value }));
        }
    }
}

```

```

_logger.Debug("Order created");
return order;
}

```

O evento "new canc" corresponde à mensagem FIX OrderCancelRequest e contém os campos referidos na Tabela 2.2 abaixo

Tabela 2.2 - Campos informados no evento OrderCancelRequest:

CIOrdID	Obrigatório	Identificador da ordem no sistema origem
OrderID	opcional	Número de Ordem atribuído pelo Broker/Mercado
OrderQty	Obrigatório	Quantidade
SecurityID	opcional	ISIN
Side	Obrigatório	1 ou 2 conforme Compra ou Venda
Symbol	Obrigatório	Menomónica do título na negociação
TransactTime	Obrigatório	Timestamp da ordem
SecondaryCIOrdID	opcional	Identificador gerado pela entidade origem

2.2.2.3 Da Euronext até à Corretora

A gateway apresentada no capítulo anterior contém uma thread dedicada à receção de eventos enviados pelo mercado ou pela Visual Trader. Este tipo de mensagens chamam-se *Execution Reports* e representam todos os estados por que passa uma ordem.

Cada um destes eventos inclui a referência indicada aquando do "new order" para ligação com o sistema do corretor (no campo CIOrdID).

Tipicamente cada solicitação efetuada pela corretora será seguida de uma mensagem do tipo "pending ..." que representa a receção do pedido por parte do mercado. Este evento será depois seguido de um outro de confirmação ou rejeição.

Assim, um pedido de "new order" será seguido de um "pending new" que por sua vez poderá ser seguido por um "reject", um "pending confirm" - caso em que é acionado um sistema de controle de mercado que exige confirmação do corretor e responsabilização - , ou um "new" - ordem aceite e viva no mercado.

Estando a ordem viva no mercado, poderá ser seguida de um "partially filled" - quando há negócio mas não perfez a totalidade da ordem – ou um "totally filled" - quando perfaz a totalidade da ordem. É desta forma que o corretor toma conhecimento da existência de negócios.

Estando a ordem viva no mercado ou parcialmente viva, pode ser cancelada por iniciativa do corretor ou por iniciativa do mercado. Caso a iniciativa seja do corretor o estado seguinte será um "pending cancel". Este pedido pode ser aceite ou pode ser rejeitado se já foi entretanto

totalmente efetuado. No primeiro caso terminará o seu ciclo de vida no estado “cancelled”. A iniciativa de pedido de cancelamento por parte do corretor corresponde ao pedido “new canc” já apresentado atrás e pode ser efetuado quer num dos canais web quer no terminal de negociação.

A gateway apresentada subscreve, para aceder a estas mensagens, o evento ExecutionReportEventHandler. Este evento é lançado pela API da Visual Trader a cada receção efetuada pelo seu servidor Linux. O software da gateway capta este evento e interpreta-o inserindo a sua informação numa tabela no Oracle. Este registo é posteriormente processado para afetar a estrutura do Backoffice, ou seja, atualizar a ordem, os saldos físicos e os financeiros. No caso de se tratar de um negócio vai também originar um registo com uma instrução de Post ou Give-up a ser entregue à Clearnet (capítulo seguinte).

O processamento de replicação do efeito do evento na ordem e nas contas é feito por um serviço .Net à parte pois, potencialmente pode ser morosa e não deve atrasar a receção dos restantes eventos do mercado.

A atualização referida consiste numa série de updates e inserts a tabelas Oracle.

Sempre que o execution report refere o evento “new” relativo a uma ordem criada no terminal de negociação, ou seja com identificador clordid desconhecido, o seu processamento corresponde à criação da mesma a nível do backoffice. Assim temos registo em Backoffice de todas as ordens independentemente do canal de entrada.

2.2.2.4 Interação entre a Clearnet e a Instituição Membro de Compensação

Aquando da introdução de uma ordem no corretor existe um campo onde se pode especificar se a ordem diz respeito a um Cliente ou se diz respeito à Carteira Própria do Compensador (OrderCapacity).

Para além desta informação também a referência da ordem no sistema do corretor é transmitida pela Euronext ao sistema da Clearnet.

Quando é efetuado um negócio no mercado Português a Euronext transmite imediatamente esse negócio para a Clearnet para que seja compensado. Este negócio entra no sistema C21 num estado “to process” (não atribuído). Cabe ao Membro da Clearnet a responsabilidade de efetuar a atribuição à conta de posição (position account). Caso esta atribuição não seja feita no decorrer do dia a Clearnet afeta-o automaticamente à conta da Carteira Própria (“House”).

O Membro pode, no entanto, registar regras de atribuição automáticas (Systematic Postings[7]). Ou seja, formas de identificar a tomada de decisão da conta destino para que o posting seja feito imediatamente pela Clearnet.

A Entidade em questão optou por não efetuar nenhuma regra sistemática mas deixar ao sistema interno a responsabilidade de fazer a atribuição.

Assim, o reconhecimento de um negócio por via da receção do execution report, já apresentado atrás, - já que estamos a falar de Entidade Corretora e Compensadora em simultâneo - permite a imediata tomada de decisão de qual ação se pretende efetuar. Esta decisão prende-se unicamente com o fator cliente da ordem.

Caso se trate de um cliente de retalho normal a decisão será um posting na position account de clientes.

Mas caso o Cliente seja a Carteira Própria o posting já será feito na position account House.

No entanto, ainda existe outro universo de clientes, aqueles que não pretendem liquidar nessa mesma Entidade mas noutra Banco da praça. Para este universo o cadastro de cliente contém o identificador do membro compensador e liquidador. Deve-se então fazer um give-up para o Compensador registado no cliente para que ele possa fazer o take-up e assumir a responsabilidade do negócio. Infelizmente o protocolo interentidades obriga a que o outro Banco da praça seja notificado pelo cliente da existência e aceitação do negócio e este passo normalmente não é feito a tempo. Caso o take-up não seja atempado a Clearnet assume o negócio como "to process" provocando uma atribuição à conta House.

Toda esta lógica é implementada quando é aplicado o execution report como já descrevi no parágrafo anterior.

A comunicação entre o Membro Compensador e a Clearnet é feita via SLIB (ISV acreditado para ligações MAP).

A SLIB dispõe de um terminal (CMS) onde se pode consultar o estado dos negócios e atuar sobre eles em caso de necessidade.

Como sistema automático foi implementado um protocolo de comunicações recorrendo a MQSeries.

Nas instalações do Banco foi criado um Queue Manager [4] com uma ligação server-server com o Queue Manager da Slib (em França).

Esta ligação dispõe de duas server connections. Uma para as denominadas comunicações batch e outra para as interativas. Pretende-se unicamente que duas áreas diferentes de negócio não compitam por largura de banda.

Para comunicações interativas temos uma queue remota no nosso server (local no server da Slib) e uma queue remota no server deles (local no nosso). A cada afetação feita no C21, tipicamente os postings, a Slib coloca uma mensagem na sua queue remota, nossa queue local para sabermos onde foi feito o post e onde vai liquidar. Após leitura dessa mensagem escrevemos o acknowledge na nossa queue remota para que eles o recebam e marquem a

mensagem como entregue. Uma falha neste acknowledge é interpretada como notificação não entregue e é acionada a sua retransmissão. Assim é registado no sistema de Backoffice cada afetação a conta de posição que servirá de guia à liquidação. No Anexo A incluo o layout detalhado desta mensagem .

A estrutura de queues, detalhada na figura 2.8, é a seguinte :

Na Slib: MQS9997.QL.RFINRQ (local) corresponde a FBA205007.QR.EFINRQ (remota) via channel FBA205007.XYLOS.01 e serve o envio dos acknowledges

Na Slib: MQS9997.QR.EFINRQ (remota) corresponde a FBA205007.QL.RFINRQ (local) via channel XYLOS.FBA205007.01 e serve o envio dos reports de posting

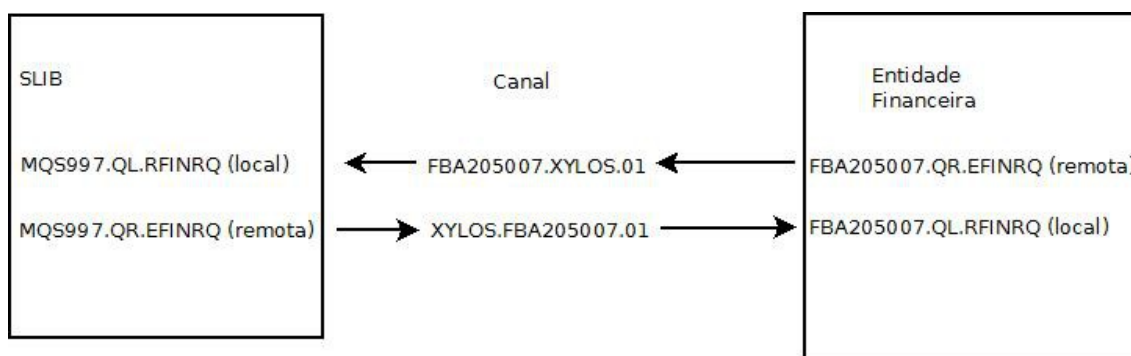


Figura 2.8 – Estrutura de Queues do circuito interativo

Como comunicações Batch temos um conjunto de duas queues para suporte a transmissão de ficheiros em cada sentido, ou seja quatro no total.

Duas das queues são utilizadas para transmitir em bloco as instruções de posting e give-up.

As restantes são utilizadas para transmitir ficheiros variados enviados pela Clearnet ao membro como o relatório das margens a ser depositadas no Banco de Portugal para garantia.

O princípio do Protocolo inclui uma queue de dados, em que cada mensagem tem uma parte comum que identifica o ficheiro e um sequenciador de mensagem seguido de uma área variável. Após escrita a ultima mensagem de dados regista-se uma mensagem de controle na queue de controle. Esta mensagem contem a chave comum aos registos de dados transmitidos. Esta é a indicação que o ficheiro está completo e pode ser recebido. No Anexo B apresento a documentação destas mensagens sob o nome MQSeries batch data transfers.

A estrutura de queues é a seguinte:

Na Slib: MQS9997.QL.RFINBA.DTA (local) corresponde a FBA205007.QR.EFINBA.DTA via channel FBA205007.XYLOS.02 e serve o envio dos dados (instruções de posting e give-up).

Na Slib: MQS9997.QL.RFINBA.CTL (remota) corresponde a FBA205007.QR.EFINBA.CTL via channel FBA205007.XYLOS.02 e serve o envio do registo controle para acionar a leitura dos vários registos na queue DTA.

Na Slib: MQS9997.QR.EFINBA.CTL (remota) corresponde a FBA205007.QL.RFINBA.CTL via channel XYLOS.FBA205007.02 e serve para escrever a mensagem de acknowledge (sentido Slib → Entidade). Tráfego de carácter puramente técnico.

Na Slib: MQS9997.QR.EFINBA.DTA (remota) corresponde a FBA205007.QL.RFINBA.DTA via channel XYLOS.FBA205007.02 e serve o envio dos ficheiros de texto e dados entregues pela Clearnet (sentido Slib → Entidade)

Na Slib: MQS9997.QR.EFINBA.CTL (remota) corresponde a FBA205007.QL.RFINBA.CTL via channel XYLOS.FBA205007.02 e serve para escrever a mensagem a indicar que já se podem ler os registos de DTA com o ficheiro entregue pela Clearnet (sentido Slib → Entidade)

O acknowledge é escrito na queue FBA205007.QR.EFINBA.CTL que corresponde a MQS9997.QL.RFINBA.CTL via channel FBA205007.XYLOS.02. A diferença entre estas mensagens e as apresentadas atrás está na informação contida no campo Message Type existente no header da mensagem.

A Figura 2.9 traduz a estrutura de queues referida.

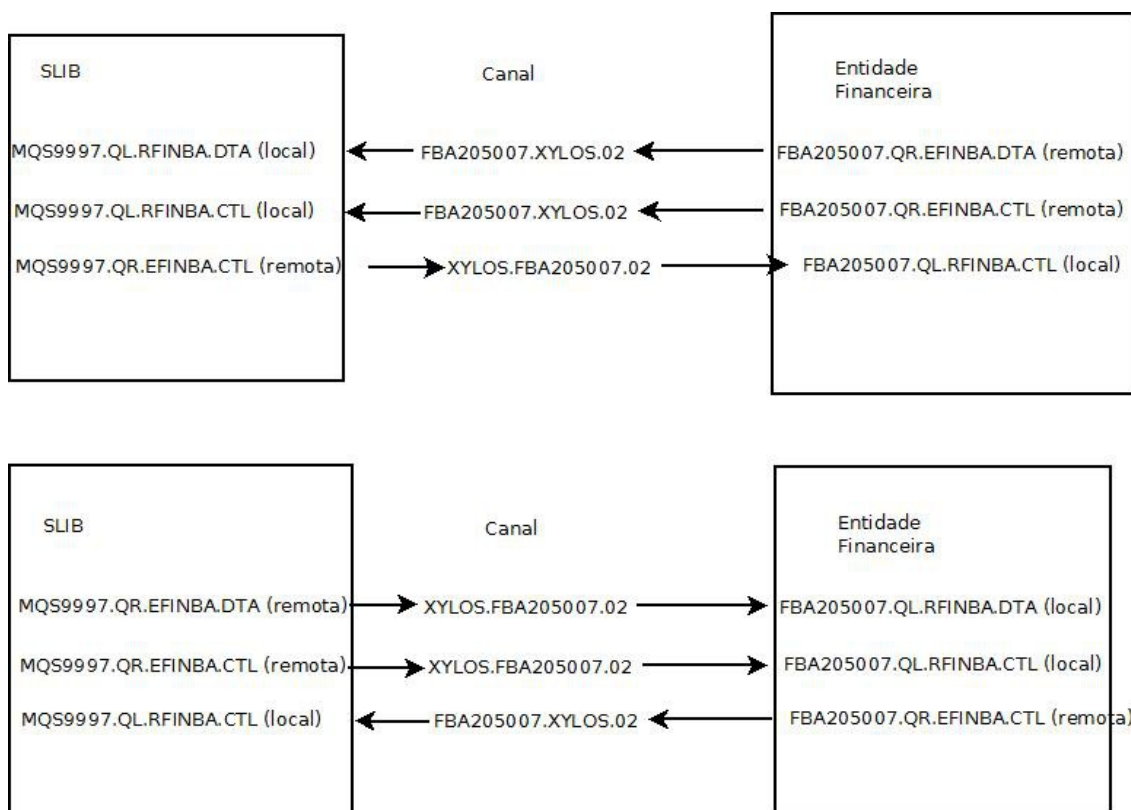


Figura 2.9 - Estrutura de queues suporte ao protocolo batch

Toda a lógica de acesso ao Mqseries, seu carregamento e interpretação foi implementado numa gateway Java como um serviço autónomo recorrendo às bibliotecas fornecidas pela IBM no pacote de software Websphere MQ. Também aqui se garantiu o correto uso de metodologias orientadas a objetos (recorrendo a heranças) para que não houvesse demasiada dependência do provider de messaging já que a qualquer momento pode haver a necessidade de alteração por questões contratuais ou políticas.

Este serviço está desdobrado em várias threads conforme a queue a monitorizar.

- CFRLListener - Monitorização da tabela de Oracle onde é escrita a instrução a enviar. Gera o número seguinte de transmissão. Lê o registo da instrução por enviar, formata a mensagem de MQ e escreve (put) na queue FBA205007.QR.EFINBA.DTA. Guarda em tabela que essa instrução foi transmitida com o seu timestamp e com o número de transmissão que lhe foi associado. Após lidas todas as instruções pendentes é inserida a mensagem de ConTroLe em FBA205007.QR.EFINBA.CTL para acionar a sua receção por parte da Slib.
- SlibRTLListener – Thread que está indefinidamente a fazer get da queue FBA205007.QL.RFINRQ para receber as mensagens a acusar a realização da instrução. A cada receção escreve (put) o seu acknowledge na queue FBA205007.QR.EFINRQ É verificada a sua correta formatação e se a mensagem recebida for válida é escrita na tabela CFR_SLIB_RT_MSG
- SlibRTMsgsToCFRLListener – Aplica no sistema de Backoffice as mensagens escritas pela thread anterior e ainda não aplicadas. Mais uma vez pretende-se celeridade e não congestionamento. Esta aplicação corresponde unicamente a um update da informação de contas de posição no negócios que já foi inserido pela gateway Visual Trader.
- SlibBALListener – Thread que monitoriza a queue FBA205007.QL.RFINBA.CTL e que a cada mensagem recebida vai processar os registos correspondentes na queue FBA205007.QL.RFINBA.DTA. Estes são escritos num ficheiro de texto em file system. Após descarregar todo o ficheiro escreve o acknowledge na queue FBA205007.QR.EFINBA.CTL

Em fim de dia a Clearnet calcula o netting de cada conta de posição por título. Este valor é transmitido à casa de liquidação para que liquide em D+3.

2.2.2.5 Do Membro liquidatário até à Casa de Liquidação - Interbolsa

Como já descrevi atrás a Clearnet diariamente apura o netting das transações efetuadas por cada valor mobiliário para informar a casa de liquidação local o valor que terá que ser liquidado três dias úteis mais tarde.

É calculado o netting pois entende-se esta etapa por si como uma liquidação, ou seja, o Membro Compensador terá que pagar à Clearnet. Por essa razão se o Membro Compensador for responsável por compras no total de uma quantidade Q de certo título que terá que receber, e simultaneamente for responsável por uma quantidade Q2 vendida que terá que entregar, bastará acertar contas pela diferença entre Q e Q2. Sendo a quantidade a receber superior à quantidade a entregar então a Entidade deverá receber Q-Q2. Caso contrário deverá entregar Q-Q2.

O mesmo raciocínio é efetuado em termos financeiros mas neste caso o montante é depositado ou levantado da conta da Entidade no Banco de Portugal. No caso em que haja lugar a receção de dinheiro é efetuado um depósito pela Clearnet na conta da Entidade no Banco de Portugal.

Quando é feito o registo do membro na Clearnet é indicada a correspondência a ser feita entre a position account e a chamada conta de liquidação na Interbolsa.

A nível da Interbolsa existe uma estrutura de contas como descrito abaixo.

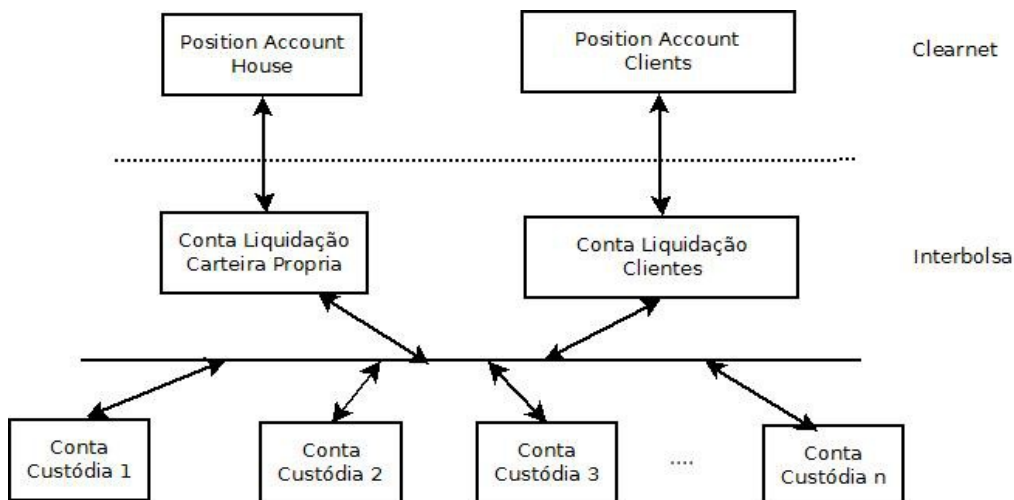


Figura 2.10 – estrutura de Contas global

Assim, como se entende no esquema a Clearnet movimenta unicamente a camada “Conta de Liquidação” sendo que os valores nela existente são provisionados diariamente de acordo com as necessidades de liquidação. Não ficarão nestas contas saldos excedentes.

Resumindo, caso para certo título haja lugar a uma obrigação de entrega de títulos então faz-se uma transferência dos títulos da camada de “Conta de Custódia” para a camada “Conta Liquidação” e assim no momento da liquidação já existe na “Conta de Liquidação” saldo suficiente para cobrir as necessidades.

No caso de a obrigação ser unicamente de recetor de títulos comprados haverá lugar a um depósito por parte da Clearnet para que posteriormente a Entidade os possa transferir para a camada “Conta de Custódia”.

Por questões legais é obrigatória a separação dos títulos em Contas de Custódia de acordo com a fiscalidade dos clientes. Assim o sistema de Custódia inclui dados suficientes a nível dos clientes para permitir aferir a conta de custódia que lhes corresponde.

Por esta razão um movimento na Conta de Liquidação pode corresponder a vários movimentos de transferência de contas de liquidação para Contas de Custódia e vice-versa.

No Anexo C, incluí o layout do ficheiro a ser transmitido à Interbolsa para realizar estas transferências. A Interbolsa permite aos seus Membros o upload de ficheiros, no seu sistema STD, para introdução das instruções de transferência. O ficheiro em questão, denominado TRF, é gerado , em Oracle Forms, uma vez que a implementação do protocolo com a Slib dotou o sistema de todos os dados necessários à decisão da transferência.

Por outro lado o mesmo sistema STD da Interbolsa fornece um ficheiro denominado OTC que informa o estado da liquidação permitindo verificações de alguma falha. O carregamento deste ficheiro para Oracle, via Oracle Forms, proporciona a validação de coerências inter-sistemas e de falhas na liquidação.

Desta forma ficou coberto o circuito de liquidação da responsabilidade do Membro.

Introduzi em capítulos anteriores que por vezes torna-se necessário realizar a nível da Interbolsa um acerto de contas com outra Entidade. Este facto sucede sempre que o cliente solicitou liquidação em outra entidade mas o take-up não foi efetuado atempadamente.

Neste caso, o sistema SLRT é acionado e por carregamento do ficheiro SLRTmsg submetesse o pedido de liquidação. Neste sistema o membro destino também terá que submeter uma instrução similar para que o casamento corresponda ao acerto de contas. A geração do SLRTmsg foi implementada nos mesmos moldes do TRF (Oracle Forms).

Não detalharei mais este sistema pois, sempre vigorou no mercado de valores mobiliários nacional.

2.2.2.6 Visão Global Tecnológica

A Figura 2.11 pretende de uma forma consolidada apresentar a globalidade dos intervenientes referidos até agora.

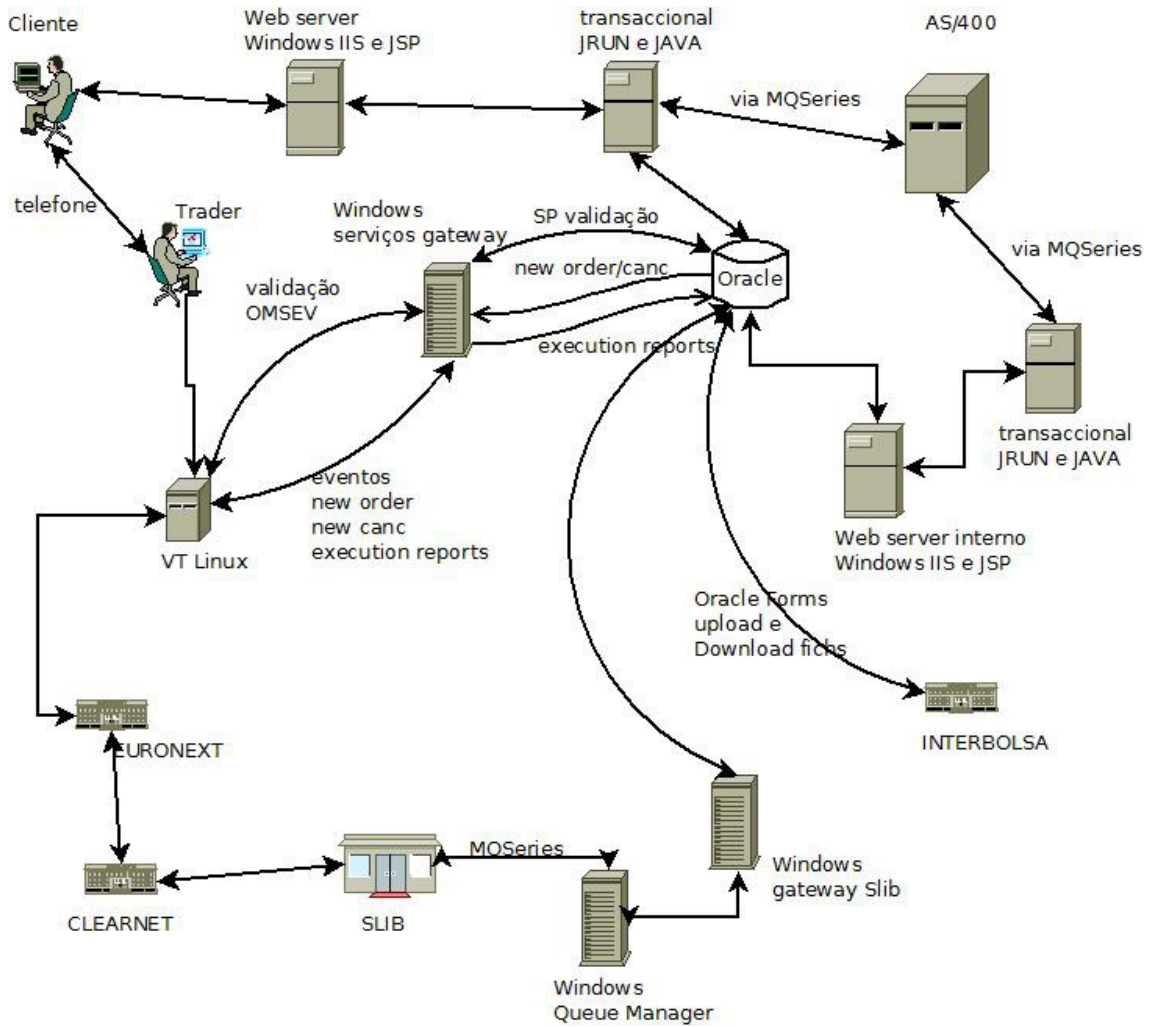


Figura 2.11 – Visão Global

2.2.3. Migração tecnológica em 2011

Em 2011, a Entidade Financeira em questão foi comprada por outra Entidade de maior dimensão.

Por esta razão, o sistema de Backoffice de Títulos não se manteve e os seus dados foram migrados para outro sistema. No entanto o sistema de negociação, terminais de negociação e relacionamentos via Visual Trader foram mantidos.

Foi também mantido o relacionamento com a Clearnet via Slib.

O sistema de Backoffice desta nova Entidade estava suportado em Z/OS [8] com bases de dados DB2 [9] e manuseamento unicamente via Cobol [10].

Mais uma vez, o trabalho de levantamento e validação das adaptações a efetuar esteve a meu cargo sendo que implementei as adaptações de código à gateway da Slib mas entreguei a outra equipa as adaptações da gateway com a VT bem como o suporte necessário em Z/OS-Cobol. Esta distribuição de trabalho teve a ver unicamente com questões de tempo e número de recursos e não com capacidades ou conhecimentos técnicos.

Nesta migração todas as Bases de dados suporte às gateways existentes no Oracle foram criadas no DB2. Como o DB2 não suporta queued tables alterou-se o relacionamento com a gateway VT para a utilização de MQSeries.

O sistema de canais desta entidade efetua a validação de saldos físicos e financeiros e regista o pedido de ordem ou cancelamento no DB2. Este registo foi adaptado para que dentro do horário de aceitação de ordens de bolsa fosse escrita uma mensagem com a mesma estrutura da anterior queued table. Assim a gateway VT passou a monitorizar (get) uma queue de MQ ao invés de uma queued table. Do mesmo modo a alteração do estado desse pedido, que era feito por update da tabela de Oracle, passou a ser feito por escrita de uma mensagem numa queue específica.

No sentido inverso, não só preparou-se uma queue para constante reporte do estado da conexão com a VT como criou-se uma queue para escrita de cada execution report. As regras da mensagem de execution report mantiveram-se.. Por seu lado, no Z/OS foi associada uma transação CICS [11] a cada uma das queues de entrada (comunicações, estado de ordem e execution report) tendo sido parametrizada para que processasse apenas uma mensagem de cada vez por forma a garantir a correta execução dos eventos.

Manteve-se a validação desencadeada a partir do terminal de negociação sendo que a alteração efetuada foi a nível do provider (Oracle para Z/OS) e a criação da stored procedure no DB2. No entanto, a entidade decidiu utilizar esta invocação para, em caso de aceitação na validação, efetuar imediatamente o registo da ordem. No cenário anterior a ordem só era

registada aquando da receção do execution report correspondente. Como o serviço “criação de uma ordem” era já uma transação cics de alguma complexidade e a execução de uma procedure DB2 não permite invocações a transações CICS foi necessário definir um subterfúgio. Tratou-se de fazer com que a stored procedure escrevesse uma mensagem numa queue de MQ com os dados necessários à criação da ordem e parametrizar o MQSeries para desencadear a execução da transação pretendida. Bastou de seguida que a stored procedure aguardasse pelo resultado da transação.

Por definição interna nesta Entidade é obrigatória a implementação das stored procedures em Cobol, não sendo permitido o uso de SQL. O uso de SQL é considerado pelos DBA de DB2 um risco pela existência de biding em tempo de execução.

Estando a recorrer a Cobol nesta implementação e por forma a simplificar o acesso a MQSeries utilizou-se a vertente MQ via DB2. Assim, o DB2 fornece um conjunto de funções (send, receive, browse, etc) para aceder ao MQ nos mesmos moldes que se acede a uma tabela DB2 [10].

No que diz respeito ao relacionamento com a Slib, toda a definição de Queue Manager existente anteriormente foi recriada dentro do Z/OS. Assim, o server local deixou de ser Windows para passar a ser Z/OS. Foi no entanto uma alteração relativamente transparente.

De resto, todas as tabelas suporte a esta gateway foram recriadas no DB2 pelo que a adaptação a efetuar na gateway reduziu-se à mudança de provider e de nomes de tabelas e campos.

Esta migração foi um caso claro em que a correta utilização dos mecanismos de herança permitiu uma célere adaptação à nova realidade de provider de bases de dados.

3. A Tecnologia ao serviço dos Mercados de Capitais – O futuro

A nível mundial já existem correntes de utilização de sistemas informáticos, alguns bastante rudimentares, para a automatização da tomada de decisão no que diz respeito a investimentos financeiras.

O objetivo destes sistemas, denominados de trading sistemático ou trading intensivo, é maioritariamente a aplicação de raciocínios matemáticos a valores recebidos por feeds de bolsa online ou em diferido. Por se tratarem de algoritmos matemáticos a maioria destes sistemas é implementado em Matlab por financeiros com poucos conhecimentos técnicos.

O funcionamento destes sistemas prende-se com o estudo prévio por parte do financeiro de um histórico de valores de transações de títulos tipificados. Assim, essa pessoa por análise estatística muitas vezes apurado num simples EXCEL estabelece o que considera ser uma regra comportamental do título. A regra apurada é introduzida num sistema que a aplica por prazo mais ou menos alargado ao mercado analisando o feed recebido diariamente. Este sistema tem a capacidade de dar instruções ao mercado em nome do financeiro e reagir ao resultado por forma a obedecer à regra que lhe foi indicada.

Qualquer pessoa que implementa este tipo de sistema sabe que o valor aplicado, o prazo de autonomia do sistema e a cadência de ação do mesmo está estreitamente relacionado com o resultado obtido. Mas para que seja possível o sucesso deste tipo de funcionamento o financeiro deve alhear os seus sentimentos por mais difícil que seja. Todo o investimento tem altos e baixos e o seu sucesso só pode ser medido depois de atingido o prazo determinado.

Este tipo de sistemas não são utilizados em Portugal mas são utilizados em países como o Luxemburgo onde há um nível de conhecimentos financeiros mais elevado. Luxemburgo é um país de referência no que diz respeito a mercados de capitais e disponibilização de plataformas ao Cliente.

Sou da opinião que este tipo de sistemas poderia ir mais além.

Penso que ainda não se aprofundou a tecnologia ao serviço do mercado financeiro.

A utilização de inteligência artificial para apurar o algoritmo mais adequado ao cabaz de valores mobiliários em análise bem como a sua autoadaptação ao decorrer dos resultados dos investimentos por forma a maximizar os ganhos e reduzir as perdas será certamente possível.

Por natureza um financeiro trabalha isolado e raramente em equipa com um engenheiro informático. Mas esta combinação abriria portas à utilização de inteligência artificial. Quem sabe se a aplicação de lógica difusa num mundo que normalmente é só preto ou branco não

seria produtiva. Certamente que este sistema conseguiria chegar onde chega o homem e provavelmente mais além.

4. Conclusões

O projeto aqui apresentado teve uma duração de pouco mais de seis meses no decorrer do ano de 2003. A etapa de migração efetuada em 2011 teve uma duração similar derivado das dificuldades de definição política inerentes a um processo de integração após uma OPA (Oferta pública de Aquisição).

Foi na minha opinião um projeto essencial ao meu percurso profissional quer na vertente técnica quer na vertente funcional.

Sou defensora do reduzir da dificuldade de comunicação entre o negócio e a equipa técnica.

Penso que dificilmente será possível uma equipa técnica implementar um sistema com sucesso sem perceber realmente o seu objetivo funcional. Deveria existir sempre mais que uma pessoa a compreender o sistema como um todo pois este tipo de sistemas, sua criticidade e dimensão, não devem estar na mão de uma pessoa só. Na maioria das vezes isso não acontece.

Quando perdura a dificuldade da equipa técnica entender o seu cliente (negócio) deteriora-se a comunicação, pois o cliente cada vez vai explicar menos por pressupor que já deviam perceber do que falam, e o resultado final vai surgir cada vez mais desenquadrado da realidade e com menos aplicabilidade prática.

Neste projeto consegui ganhar esse entendimento, interpretar mesmo os pedidos que o cliente não conseguia explicar ou pedir e por essa razão culminou num cliente claramente satisfeito.

Em termos técnicos, este projeto foi um desafio. A existência de novas entidades com que nunca me tinha relacionado, com sistemas que me eram total ou parcialmente desconhecidos foi aliciante.

Já tendo no percurso anterior largo conhecimento de MQSeries e Java, nomeadamente pelas implementações da estrutura de homebanking, foi aliciante desenhar um sistema correto e duradouro que já foi apresentado a várias entidades sem que nenhuma apontasse qualquer falha de desenho.

Utilizei já esse mesmo conhecimento para ensinar vários técnicos que pretendem utilizar sistemas de messaging e têm dificuldade em o fazer.

Foi inovador para mim neste projeto a utilização de .Net e de Objetos COM com subscrição de eventos. Infelizmente trata-se de uma tecnologia totalmente *Windows dependent* aspeto a seu desfavor. Apesar de tecnologias independentes de sistema operativo serem mais do meu agrado não posso deixar de afirmar que esta utilização foi para mim novidade e aliciante.

Certamente enriqueci os meus conhecimentos técnicos neste projeto mas também dei a enriquecer conhecimentos a outros.

Infelizmente não são muitos os projetos empresariais que correspondem a este nível de enriquecimento de conhecimentos.

5. Referências

1. [http://msdn.microsoft.com/en-us/library/vstudio/a4t23kkt\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/a4t23kkt(v=vs.90).aspx)
2. http://www.visual-trader.com/docs/Tecnica/2014/07/FIXBRIGDE_VT_4.4_Specifications.pdf
3. JDBC¿ API Tutorial and Reference , by Maydene Fisher, ,Jon Ellis,Jonathan Bruce
4. <https://publib.boulder.ibm.com/iserics/v5r2/ic2924/books/csqzal05.pdf>
5. Java(TM) 2 SDK, Standard Edition Documentation 1.3.1
<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-13docs-2045553.html>
6. http://docs.oracle.com/cd/B10501_01/appdev.920/a96587.pdf
7. [https://www.lchclearnet.com/Images/How_it_works_Cash_markets\[1\]_tcm6-46823.pdf](https://www.lchclearnet.com/Images/How_it_works_Cash_markets[1]_tcm6-46823.pdf)
8. <https://www.informatik.uni-leipzig.de/cs/Literature/Textbooks/jcl/MvsJclUserGuide.pdf>
9. <http://publib.boulder.ibm.com/epubs/pdf/dsnapm0a.pdf>
10. IBM COBOL for VSE/ESA, IBM, Language Reference, Release 1, SC26
11. <ftp://ftp.software.ibm.com/software/cics/pdf/dfhp300.pdf>
12. Oracle9i, JDBC Developer's Guide and Reference, Release 2 (9.2), March 2002, Part No. A96654-01

6. Anexos

ANEXO A

	Fields	Obligatory	Declaration	Length	Start	End	Comments
1	Message code	y	Char(4)	4	1	4	'POST' = Posting 'GIUP' = Give-up 'GUPR' = Rejected Give-ups
2	Message ID	y	Char(15)	15	5	19	Unique reference for a message ccymmdd + sequential number
3	Old message ID	n	Char(15)	15	20	34	Previous Message ID (only for modify or cancel)
4	Message sent before	y	Char(1)	1	35	35	'Y' = Yes 'N' = No
5	Type of message	y	Char(3)	3	36	38	'NEW' = New 'MOD' = Modify 'CAN' = Cancel
6	Clearinghouse Code	y	Char(3)	3	39	41	C21
7	Clearing organization	y	Char(5)	5	42	46	'SBF' 'SNM' 'MATIF' 'MONEP'
8	Exchange	y	Char(5)	5	47	51	'SBF' 'SNM' 'MATIF' 'MONEP'
9	Financial Market	y	Char(5)	5	52	56	'025' = Marché SBF - Bourse de Paris '270' = Nouveau Marché '271' = EuroCAC '272' = Marché libre OTC '273' = Marché des reports, débat des valeurs '274' = Monep '275' = Marché des reports, débat de l'argent '276' = Matif '277' = Marché centralisé de prêt / emprunt de titres '278' = Brussels '279' = Amsterdam '280' = Brussels Derivatives '281' = Amsterdam Derivatives '290' = Lisbon Cash
10	Clearing member (CMF)	y	Char(10)	10	57	66	<i>NSC</i>
11	Trading member (TMF)	y	Char(10)	10	67	76	<i>00010015</i>
12	Trading Date *	y	Date	10	77	86	Trading Date assigned by NSC. format: ccyymm-dd <i>2003-09-26</i>
13	Business Date of the Trade	y	Date	10	87	96	Business Date assigned by C21. format: ccyymm-dd <i>2003-09-26</i>
14	Settlement Date *	y	Date	10	97	106	format: ccyymm-dd
15	External trade id	y	Num(10)	10	107	116	Trade external id given by the matching engine. This id is unique per business date, matching engine, side (buy/sell). This id is not used by C21 but conveyed to the member.
16	C21 trade leg id	y	Num(10)	10	117	126	Trade id given by C21. This id is unique per business date.

17	Trading order number	n	Char(10)	10	127	136	ID assigned by the TMF, conveyed through NSC and C21
18	Guaranteed Indicator	y	Char(1)	1	137	137	Trade guarantee or non-guaranteed by the Clearing Organisation. The Clearing Organisation Guarantee is a characteristic of the traded security. 'Y' = Yes 'N' = No
19	Type of trade	y	Char(2)	2	138	139	Origin of the trade: 'AS' = Trade generated following exercise / assignment 'BK' = Back-up trade 'CE' = Trade generated following an external correction 'CO' = Corrective trade 'GU' = Trade generated by a Give-up 'MK' = Market Trade 'LT' = Linked Trade generated following a corporate event
20	Corporate event indicator	y	Char(1)	1	140	140	Generated by a Corporate event 'Y' = Yes 'N' = No
21	Corporate event ID	n	Char(6)	6	141	146	Corporate event ID.
22	C21 parent trade reference (trade leg ID)	n	Num(10)	10	147	156	C21 parent trade leg ID for a trade generated by a Corporate event
23	Date of clearing of the parent trade	n	Date	10	157	166	Business date of the C21 parent trade leg ID for a trade generated by a Corporate event. format: ccpy-mm-dd
24	Matching engine	y	Char(10)	10	167	176	System which has generated the trade: 'NSCVE' 'NSCVF' 'NSCVO' 'NSCVL' 'ACT' 'C21'
25	Contango Code	n	Char(3)	3	177	179	Specific code for a trade issued by NSCVL: 'MEP' = Mise en place pour SBF 'MNM' = Mise en place pour SNM 'REM' = Remboursement pour SBF 'RNM' = Remboursement pour SNM
26	ISIN code	y	Char(12)	12	180	191	ISIN Code of the security.
27	Trading code	y	Char(40)	40	192	231	Trading Code of the security
28	Trade Origin	y	Char(1)	1	232	232	'C' = Client 'H' = House 'T' = Teneur de Marché / Market Maker / Liquidity Provider
29	Code memo / Mailing comment	n	Char(30)	30	233	262	Free text assigned by the TMF, conveyed through NSC to C21
30	Quotation code	y	Char(1)	1	263	263	'Q' = Quantity 'N' = Nominal
31	Buy or sell indicator	y	Char(1)	1	264	264	'A' = Buy 'V' = Sell

C/H/L

32	Trade Quantity	y	Num(15,2)	15	265	279	Trade quantity
33	Trade Currency	y	Char(3)	3	280	282	Quotation Currency
34	Trade unit price	y	Num(15,6)	15	283	297	Unit matching price of the trade.
35	Amount of settlement	y	Num(15,2)	15	298	312	Trade Quantity * Valued price (Trade unit price accrued interest included)
36	Settlement Currency	y	Char(3)	3	313	315	Currency of trade except for some Dutch products
37	Exchange rate	y	Num(15,6)	15	316	330	Currently = 1
38	Timestamp reception in SLIB	y	Timestamp	26	331	356	format: ccyymmddhhmmssnnnnn
39	Matching Timestamp	y	Timestamp	26	357	382	Matching Timestamp in NSC format: ccyymmddhhmmssnnnnn
40	Trade C21 Timestamp	y	Timestamp	26	383	408	Integration Timestamp in C21 format: ccyymmddhhmmssnnnnn
41	Date of clearing of the posting or give-up	y	Date	10	409	418	format: ccyymmdd
42	Posting or give-up ID	y	Num(10)	10	419	428	C21 ID of the posting or the give-up.
43	Clearing operation mode	y	Char(1)	1	429	429	'S' = systematic (rules implemented in C21) 'A' = automatic (clearing data filled in the order) 'M' = manual
44	Internal Correction Type	n	Char(1)	1	430	430	'A' = Position Account and O/C indicator modified 'O' = O/C Indicator modified 'P' = Position Account modified
45	Quantity of posting or give-up	y	Num(15,2)	15	431	445	Quantity posted or given-up <= Trade Quantity
46	Amount of settlement	y	Num(15,2)	15	446	460	Quantity Posted or given-up * Valued price (Trade unit price accrued interest included) <i>ven a zero5</i>
47	O/C indicator	n	Char(1)	1	461	461	Derivatives: either the posting will open a new position or close an existing one. 'O' = Open 'C' = Close
48	Position account	n	Char(15)	15	462	476	For a posting
49	Destination member	n	Char(10)	10	477	486	For a give-up
50	Posting or give-up C21 Timestamp	y	Timestamp	26	487	512	Clearing action Timestamp assigned by C21 format: ccyymmddhhmmssnnnnn
51	Previous Date of clearing of the posting or give-up	n	Date	10	513	522	Clearing Date of the previous posting or give-up format: ccyymmdd
52	Previous posting or give-up C21 ID	n	Num(10)	10	523	532	C21 ID of the previous posting or give-up
53	Previous quantity of posting or give-up	n	Num(15,2)	15	533	547	Quantity of the previous posting or give-up
54	Previous O/C indicator	n	Char(1)	1	548	548	Derivatives: O/C indicator of the previous posting 'O' = Open 'C' = Close
55	Previous position account	n	Char(15)	15	549	563	Position account of the previous posting

56	Previous Posting or give-up C21 Timestamp	n	Timestamp	26	564	589	C21 Timestamp of the previous posting or give-up format: ccyy-mm-dd-hh.mm.ss.nnnnnn
57	C21 Contract ID	y	Char(40)	40	590	629	C21 Code of the product.
58	Destination member's Position Account	n	Char(15)	15	630	644	Position account sent for information to the destination member of a give-up
59	Mailing Comment	n	Char(30)	30	645	674	Free text sent for information to the destination member of a give-up
60	Key for fee distribution	n	Num(1)	1	675	675	Fee distribution for an external correction: '1' = 0% sending mb - 100% destination mb '2' = 50% sending mb - 50% destination mb '3' = 100% sending mb - 0% destination mb '4' = 0% sending mb - 0% destination mb
61	Type of brokerage fee	n	Char(1)	1	676	676	Type of brokerage fee: 'L' = Unit amount per product 'P' = Percentage of the trade value 'T' = Amount for the trade
62	Amount of brokerage fee	n	Num(15,2)	15	677	691	Brokerage fee: linked with the previous field.
63	Sending member	n	Char(10)	10	692	701	Sending member for a trade received after acceptance of a take-up request.
64	Posting Order Number	n	Char(16)	16	702	717	Posting Order Number of the trade.
65	Counter Party	n	Char(15)	15	718	732	For a non-guaranteed trade
66	CMF of the Counter Party	n	Char(10)	10	733	742	For a non-guaranteed trade
67	Valued price	y	Num(15,6)	15	743	757	Trade unit price + accrued interest
68	Filler			243	758	1000	

Data can be describes as follows :

A) Trade Data

➤ Trade data

B) Clearing Action Data

- Posting data for :
 - A Posting creation,
- New posting data for :
 - A Posting modification,
- Cancelled posting data for :
 - A Posting cancellation,
- Give-up data for :
 - A Give-up creation,
- New give-up data for :
 - A Give-up modification,
- Cancelled give-up data for :
 - A Give-up cancellation,

C) Previous Clearing Action Data

- Previous posting data for :
 - A posting modification,
- Previous give-up data for :
 - A give-up modification

1 Document purpose

This document describes the batch exchanges through MQSeries between two systems. The business content of exchanged data is described elsewhere in other analyses and documents, we just define here the technical protocol used to transfer data through MQSeries middleware.

2 Philosophy overview

2.1 General architecture

- Batch exchanges are carried out using Control and Data Queues in both sending and receiving data.
- Control queue allows :
 - To trigger the processing of a transaction (transaction=set of data messages making up an overall transfer) once all data of this transaction have been put in the associated Data queue
 - To communicate between network partners on the fulfillment of the exchange; allowing then to know what is the status of a transaction data transfer (acknowledged, rejected); data sent through the Control queue is strictly technical.
 - To avoid processing collisions (eg : several processes tackling the same transaction), while limiting necessary queues
- Within a same transaction, miscellaneous messages will bear same Transaction Id (Application Header data field), but also same MSGID (MQSeries System Header data field), and are sent through a queue called Data Queue. When all messages of a transaction are put in the Data Queue, a message, called control message, is put in the associated Control Queue with the same MSGID, Transaction Id and Transaction Code than the "data" messages of the transaction. The receiver then processes the data messages, and sends back to the sender a control message which is an acknowledgement if transaction data has been received with success, or a rejection message in any other case. In both cases these control messages have the same Transaction Id and Transaction Code (the MSGID is not concerned) than the data and control messages previously received from the sender of the transaction.

2.2 Volumes

As exchanged data volumes might be important, and to avoid Queues saturation, it is possible to put several records in a same MQSeries message, bearing in mind the following restrictions:

- There is only one Application Header for all records included in a same message, which implies that records put in one message have same structure and length
- Number of records included in the message as well as the length of each record are given in the Application Header (see below usage of 20 characters filler zone)
- Even though an MQSeries message may have a larger length, messages will have a maximal length of 32766 characters.

NB : it is possible to have only one record per message. This is recorded for the partner (sender or receiver), on partner demand.

2.3 Communication rules

Each MQSeries message is made up of the MQSeries System Header, an Application Header implementing the communication protocol between the partners and business data called Application Data

3 Data description

3.1 MQSeries System Header

The hereafter table sums up the use of the system header data fields.

Data field	Usage
CorrelID	Not used
Msg ID	Feeding left to MQSeries. However it will bear the same value in all data messages of a given transaction, as well as in the control message indicating the end of the sending of the data messages of this transaction. NB : the MSGID of the control messages generated by the receiver (to notify the sender of an acknowledgement or a rejection of a transaction) are independent from the MSGID of the messages of the transactions acknowledged or rejected.
Reply-To Queue	Not used
Other...	Tuning (performances, etc...).

3.2 Application Header

3.2.1 Structure

Hereafter tables sum up Application Header structure in case of control messages (messages sent in the Control Queue) and of data messages (messages sent in the Data Queue).

Column "Sent back ?" says whether the data field is carried over from the data messages to the related control messages; if yes, data and control messages will have the same value in the data field.

3.2.1.1 Control messages

Data field	Format	Description	Sent back ?
Transaction ID	A16	Transaction identifier	Yes
Transaction Step	N7	1	
Transaction Code	A6	Transaction or request type. <i>C2C1</i>	Yes
Timestamp	N20	Messages sending timestamp. Format ccyymmddhhmmssuuuuuu	
User ID	A10	Caller (requester) identifier <i>005TUK271</i>	Yes
Adherent code	A3	Internal SLIB adherent code of the partner (sender / receiver) <i>107</i>	Yes
Question/response code	A1	For information only. <i>Q/R</i>	
Message Type	A3	Type of the message. Possible values are: <ul style="list-style-type: none"> ➤ "CTL", all data messages of the transaction have been sent, this message is sent by the sender of the transaction to notify the receiver ➤ "ACQ", acknowledgement ; the receiver was able to get all transaction data and notifies the sender ➤ "REJ", rejection ; the receiver was not able to get the data messages of the transaction and notifies the sender 	
Filler	A20	Filler zone, left blank.	

3.2.1.2 Data messages

Data field	Format	Description	Sent back ?
Transaction ID	A16	Transaction identifier	Yes
Transaction Step	N7	Numbering of the messages of a same transaction.	
Transaction Code	A6	Transaction or request type.	Yes
Timestamp	N20	Messages sending timestamp. Format ccyymmddhhmmssuuuuuu	

Data field	Format	Description	Sent back ?
User ID	A10	Caller (requester) identifier	Yes
Adherent code	A3	Internal SLIB adherent code of the partner (sender / receiver)	Yes
Question/response code	A1	For information only.	
Message Type	A3	Type of the message. Possible values are: > " ", normal case of message containing business data and which is not the last message of the transaction > "FIN", last data message of the transaction	
Filler	A20	Filler zone. This data field is structured according to the following : > Characters from position 1 to 5 (included) are used to give the number of records actually included in the message ; if not filled in or if not a valid number, this number is assumed to be equal to 1 (i.e. one record only in the message) > Characters from position 6 to 10 (included) are used to give the length of all records included in the message If filled in, these numbers must be valid numbers.	

3.3 Application Data

3.4 Overview

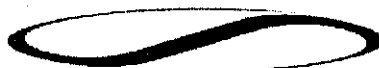
Application Data structure depends on exchanged data and underlying business processes taking care of them (i.e. depends on the Transaction Code).

However Control Messages have an "Application Data" structure which is the same, whatever the type of the transaction.

3.4.1 Control message structure

Column "Sent back ?" says whether the data field has the same value in every control message of the transaction (i.e. message sent by the sender to notify receiver of the end of the sending of data messages, and message sent back by the receiver to notify the sender of the acknowledgement or the rejection of the transaction).

Data field	Format	Description	Sent back ?
Batch number	6N	Number which is controlled against the expected batch number. This number must be without gap and depends on the Transaction Code.	
Records number	15N	The total number of sent records.	
Sender zone	90A	A filler reserved to the sender. For example SLIB uses this filler to manage the acknowledgements and rejections of its partners. The receiver of a transaction is just requested to send back the part of data in its control message. The content of this data field is sender responsibility.	Yes
Message Code <i>'POST'</i>	7A	Message code giving more details on the acknowledgement/rejection of the receiver. Filled in by the receiver.	
Message label	80A	Above message label. Filled in by the receiver.	



INTERBOLSA

Mnemónica =	TRF
Denominação =	TRANSFERÊNCIA IMEDIATAS COM INTERVENÇÃO DO BANCO DE PORTUGAL
Menu =	Movimento de contas
Descrição =	Esta mensagem fornece informação sobre as transferências com efeitos imediatos entre Intermediários Financeiros e transferências realizadas entre o BP e os Intermediários Financeiros no âmbito de operações de cedência de fundos, garantias prestadas, FGD...
Tipo de grelha =	Formatada

Posição	Comp.	Tipo	Designação STD	Descrição
01	12	N	Numero	Número sequencial de entrada da mensagem no Sistema
13	08	A	Hora	Hora, minuto e segundo de entrada da mensagem no Sistema - no formato HH:MM:SS
01	03	N	Motivo	Motivo da transferência - 115, 151, 153, 154, 180, 181, 301, 310, 312, 320, 321, 366 e 375.
04	35	A	Descricao	Descrição da mensagem (*)
39	05	N	Num-Doc	Número do documento
44	09	A	Val-Mob	Código do valor mobiliário - no formato EEEYTTDIS
53	14	N	Quantidade	Quantidade - no formato 99.999.999.999.999
67	03	N	Sit-Orig	Situação origem
70	03	N	Sit-Dest	Situação destino
73	10	N	Conta-Orig	Conta origem
83	10	N	Conta-Dest	Conta destino

(*) Descrição da mensagem

Motivo (Posição 01)	Mensagem (Posição 05)
115	EXECUCAO DE PENHOR BP/FGD
151	BL. REPORTE AGUARDA CFR BP (para IF destino) BL. REPORTE CONFIRMADO PELO BP (para IF's origem e destino) BL. REPORTE RECUSADO PELO BP (para IF's origem e destino) BL. REPORTE EXCLUÍDO PELO IF (para IF destino)
153	CONSTITUICAO DE PENHOR A FAVOR BP
154	EXTINCAO DE PENHOR BP
180	REPORTE EM D AGUARDA CFR BP (para IF destino) REPORTE EM D CONFIRMADO PELO BP (para IF's origem e destino) REPORTE EM D RECUSADO PELO BP (para IF's origem e destino) REPORTE EM D EXCLUÍDO PELO IF (para IF destino)
181	FIM REPORTE EFECTUADO PELO BP
301	TRF. (=TIT) AGUARDA CFR (para IF destino) TRF. (=TIT) CONFIRMADA (para IF's origem e destino) TRF. (=TIT) RECUSADA (para IF's origem e destino) TRF. (=TIT) EXCLUÍDA (para IF destino)



INTERBOLSA

Motivo (Posição 01)	Mensagem (Posição 05)
310	TRF. P/EXERCICIO WARRANTS AGUARDA CF (para IF destino) TRF. P/EXERCICIO WARRANTS CONFIRMADA (para IF's origem e destino) TRF. P/EXERCICIO WARRANTS RECUSADA (para IF's origem e destino) TRF. P/EXERCICIO WARRANTS EXCLUÍDA (para IF destino)
312	TRF. WARRANTS-EXERCICIO AGUARDA CF (para IF destino) TRF. WARRANTS-EXERCICIO CONFIRMADA (para IF's origem e destino) TRF. WARRANTS-EXERCICIO RECUSADA (para IF's origem e destino) TRF. WARRANTS-EXERCICIO EXCLUÍDA (para IF destino)
320	BLOQUEIO P/LIQ.BOLSA CONFIRMADO
321	DESBLOQUEIO P/LIQ.BOLSA CONFIRMADO
366	TRANSF. OTC AGUARDA CFR (para IF destino) TRANSF. OTC CONFIRMADA (para IF's origem e destino) TRANSF. OTC RECUSADA (para IF's origem e destino) TRANSF. OTC EXCLUÍDA (para IF destino)
375	REGUL. INTERNA CONFIRMADA (IF origem = IF destino)