

UNIVERSIDADE DE SÃO PAULO  
ESCOLA POLITÉCNICA

SERGIO MICHAILOVICI

**SISTEMA FLEXÍVEL DE APOIO A OPERAÇÕES DA  
TESOURARIA DE UM BANCO DE GRANDE PORTE:  
REQUISITOS, MODELAGEM E BENEFÍCIOS**

São Paulo

2012



SERGIO MICHAILOVICI

**SISTEMA FLEXÍVEL DE APOIO A OPERAÇÕES DA  
TESOURARIA DE UM BANCO DE GRANDE PORTE:  
REQUISITOS, MODELAGEM E BENEFÍCIOS**

Trabalho de formatura apresentado à  
Escola Politécnica da Universidade de São  
Paulo para obtenção do diploma de  
Engenheiro de Produção.

São Paulo

2012



SERGIO MICHAILOVICI

**SISTEMA FLEXÍVEL DE APOIO A OPERAÇÕES DA  
TESOURARIA DE UM BANCO DE GRANDE PORTE:  
REQUISITOS, MODELAGEM E BENEFÍCIOS**

Trabalho de formatura apresentado à  
Escola Politécnica da Universidade de São  
Paulo para obtenção do diploma de  
Engenheiro de Produção.

Orientador:  
Prof. Dr. Mauro de Mesquita Spinola

São Paulo

2012

## **FICHA CATALOGRÁFICA**

**Michailovici, Sergio**

**Sistema flexível de apoio a operações da tesouraria de um banco de grande porte: requisitos, modelagem e benefícios /**  
**S. Michailovici. -- São Paulo, 2012.**

**115 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.**

**1.Sistemas de informação 2.Modelagem de dados 3.UML**  
**4.Mercado financeiro I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.**

*À minha família*

*e aos meus amigos*



## **AGRADECIMENTOS**

À minha família, pela educação recebida e pelo apoio dado em momentos tão importantes.

Ao professor Dr. Mauro de Mesquita Spinola, pela confiança depositada em mim, pela preciosa paciência e pelos sábios conselhos durante toda a orientação deste Trabalho de Formatura.

Aos meus amigos, dentre os quais os companheiros de faculdade, no Brasil e na França, pela compreensão, pelo suporte e por todos os momentos compartilhados.

Aos meus colegas de trabalho, pelos ensinamentos e pela constante disposição em colaborar com este projeto.

À Escola Politécnica da Universidade de São Paulo e a todos os seus professores, pela oportunidade e pelo ensino de qualidade.



## EPÍGRAFE

“*Não sou nada.*

*Nunca serei nada.*

*Não posso querer ser nada.*

*À parte isso, tenho em mim todos os sonhos do mundo.”*

*Álvaro de Campos*



## RESUMO

Este trabalho tem como objetivo propor um sistema flexível de apoio a operações do *Trading* da tesouraria de um banco, através da sua modelagem, da especificação de seus requisitos e da análise dos benefícios resultantes. A organização estudada é um grupo bancário de grande porte dentro do qual o autor realiza um programa de *trainee*. Atualmente, registram-se todas as operações realizadas, incluindo todos os detalhes referentes a elas, em um sistema. No entanto, este sistema não é capaz de fornecer relatórios que consolidem a base de operações de que dispõe aos usuários, privando-os de obter informações úteis para as tarefas de seu dia-a-dia. Para solucionar esta questão, conduziram-se diversas entrevistas com esses usuários, que, somadas à própria experiência do autor, que também é um usuário, possibilitaram a melhor compreensão de todo o contexto. A partir disso, pode-se perceber a necessidade e a importância do sistema flexível proposto, que permite diversas modalidades de consulta através de variadas combinações de agrupamento de atributos. Para o desenvolvimento desse sistema, recorre-se, neste estudo, à modelagem orientada a objetos, com a utilização da linguagem UML. Assim, definem-se os requisitos em questão e realiza-se a análise para, finalmente, avaliar os benefícios decorrentes desta melhoria do sistema. O modelo proposto apresenta potenciais vantagens significativas ao usuário em termos de agilidade e reatividade, características importantíssimas no mercado financeiro. Dessa forma, esta flexibilidade do sistema reflete-se também, consequentemente, em ganhos financeiros. Este trabalho visa servir de base para os próximos passos referentes à implantação do sistema proposto no banco.

Palavras-chave: Sistemas de informação. Modelagem de dados. UML. Mercado financeiro.



## ABSTRACT

This paper aims to propose a flexible system for the booking of trades at the Institutional Treasury division of a bank, through the modeling of this system, the specification of its requirements and the analysis of its resulting benefits. The studied organization is a large banking group where the author works as a trainee. Currently, all trades are booked on a system which gathers them, building a complete database. Nevertheless, this system is not able to provide users with reports that consolidate the trades of this database, precluding them from obtaining useful information for their everyday work. In order to solve this issue, several interviews with users were conducted, which, together with the author's own experience as a user, enabled a better understanding of the whole process. From this, we garner the need for a more flexible system, which allows many types of search through different combinations of grouping attributes. In this study, we perform an object-oriented modeling, utilizing the UML language. Then we define all relative requirements and we perform the analysis in order to evaluate the resulting benefits of this improvement on the system. The proposed model presents significant advantages to the user in terms of nimbleness and responsiveness, very important characteristics in financial markets. This system's flexibility can then be translated into financial gains as a consequence. This paper has the goal of providing a solid basis for the next necessary steps to implement the proposed system at the bank.

Keywords: Information system. Data modeling. UML. Financial market.



## LISTA DE FIGURAS

Figura 1.1 – Disposição dos escritórios do Banco Silva por país.....	31
Figura 1.2 – Estrutura da equipe de <i>Trading</i> Internacional da tesouraria do Banco Silva. ....	35
Figura 1.3 – Tela atual de pesquisa do sistema da tesouraria do Banco Silva.....	36
Figura 1.4 – Fluxo das informações, passando por <i>front office</i> , <i>middle office</i> e <i>back office</i> , relacionadas ao registro e processamento de operações no sistema da tesouraria do Banco Silva. ....	36
Figura 2.1 – Eficiência global média de 41 bancos brasileiros entre os anos de 1995 e 1999. ....	45
Figura 2.2 – Tipos de requisitos não funcionais. ....	51
Figura 2.3 – Exemplo de um relacionamento de associação. ....	53
Figura 2.4 – Exemplo de um relacionamento de agregação. ....	53
Figura 2.5 – Exemplo de um relacionamento de composição. ....	54
Figura 2.6 – Exemplo de diagrama de sequência. ....	54
Figura 3.1 – Sequência de ações que se sucedem após a realização de um <i>trade</i> quando não há erros cometidos por nenhuma das partes envolvidas. ....	57
Figura 3.2 – Sequência de ações que se sucedem após a realização de um <i>trade</i> quando o usuário do <i>middle office</i> detecta uma divergência. ....	59
Figura 3.3 – Sequência de ações que se sucedem após a realização de um <i>trade</i> quando o usuário do <i>back office</i> detecta uma divergência. ....	60
Figura 3.4 – Gráfico da evolução do preço da ação PETR4 em 25 de abril de 2012. ....	64
Figura 3.5 – Nota divulgada a respeito de um discurso da presidente da Petrobras, Maria das Graças Foster, em 25 de abril de 2012.....	64
Figura 3.6 – Estrutura dos relacionamentos do sistema com seus usuários, com destaque para a interface a ser analisada e desenvolvida no presente trabalho. ....	66
Figura 3.7 – Base do sistema da tesouraria do Banco Silva contendo operações de RXM2 e TYM2.....	68
Figura 3.8 – Primeiro exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva. ....	69



Figura 3.9 – Segundo exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva. ....	69
Figura 3.10 – Terceiro exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva. ....	69
Figura 4.1 – Diagrama de caso de uso “Realizar o <i>login</i> ”.....	71
Figura 4.2 – Diagrama de sequência referente ao caso de uso “Realizar o <i>login</i> ”....	72
Figura 4.3 – Diagrama de caso de uso “Boletar operação”.....	73
Figura 4.4 – Diagrama de sequência referente ao caso de uso “Boletar operação”..	73
Figura 4.5 – Diagrama de caso de uso “Pesquisar operação”.....	74
Figura 4.6 – Diagrama de sequência referente ao caso de uso “Pesquisar operação”.	75
Figura 4.7 – Diagrama de caso de uso “Consultar operação”.....	76
Figura 4.8 – Diagrama de sequência referente ao caso de uso “Consultar operação”.	76
Figura 4.9 – Diagrama de caso de uso “Alterar operação”.....	77
Figura 4.10 – Diagrama de sequência referente ao caso de uso “Alterar operação”..	77
Figura 4.11 – Diagrama de caso de uso “Cancelar operação”.....	78
Figura 4.12 – Diagrama de sequência referente ao caso de uso “Cancelar operação”.	79
Figura 4.13 – Diagrama de caso de uso “Cadastrar ativo”.....	80
Figura 4.14 – Diagrama de sequência referente ao caso de uso “Cadastrar ativo” ...	80
Figura 4.15 – Diagrama de caso de uso “Aprovar operação”.....	81
Figura 4.16 – Diagrama de sequência referente ao caso de uso “Aprovar operação”.	82
Figura 4.17 – Diagrama de caso de uso “Consultar posição”.....	83
Figura 4.18 – Diagrama de sequência referente ao caso de uso “Consultar posição”.	83
Figura 4.19 – Layout da tela de <i>login</i> do usuário. ....	87
Figura 4.20 – <i>Layout</i> da tela inicial correspondente ao usuário do <i>front office</i> .....	87
Figura 4.21 – <i>Layout</i> da tela inicial correspondente ao usuário do <i>middle office</i> .....	88
Figura 4.22 – <i>Layout</i> da tela inicial correspondente ao usuário do <i>back office</i> .....	88
Figura 4.23 – <i>Layout</i> da tela para boletar uma nova operação. ....	89
Figura 4.24 – <i>Layout</i> da tela para pesquisar operações. ....	90
Figura 4.25 – <i>Layout</i> da tela para consultar a posição. ....	91
Figura 5.1 – Generalização das classes Operação. ....	93
Figura 5.2 – Generalização das classes Produto Financeiro.....	94
Figura 5.3 – Generalização das classes Usuário. ....	94
Figura 5.4 – Classe “Usuário” . ....	95



Figura 5.5 – Classe “Usuário do <i>front office</i> ”.....	96
Figura 5.6 – Classe “Usuário do <i>middle office</i> ”.....	97
Figura 5.7 – Classe “Usuário do <i>back office</i> ”.....	97
Figura 5.8 – Classe “Produto financeiro” .....	98
Figura 5.9 – Classe “Par de moedas”.....	98
Figura 5.10 – Classe “Opção” .....	99
Figura 5.11 – Classe “Contrato futuro” .....	100
Figura 5.12 – Classe “Operação”.....	100
Figura 5.13 – Classe “Operação de moeda”.....	102
Figura 5.14 – Classe “Operação de opção”.....	103
Figura 5.15 – Classe “Operação de contrato futuro” .....	103
Figura 5.16 – Classe “Contraparte”.....	104
Figura 5.17 – Classe “Mesa” .....	104
Figura 5.18 – Classe “Estratégia” .....	105
Figura 5.19 – Diagrama de classes. ....	106



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>29</b>
1.1.	A EMPRESA .....	29
1.2.	A ÁREA DE ATUAÇÃO .....	31
1.3.	ATIVIDADES DA ÁREA DE <i>TRADING</i> .....	33
1.4.	O TRAINEE .....	34
1.5.	O PROBLEMA .....	38
1.6.	OBJETIVO .....	39
1.7.	METODOLOGIA .....	39
1.8.	ESTRUTURA DO TRABALHO .....	40
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>42</b>
2.1.	INFORMAÇÃO .....	42
2.2.	SISTEMAS DE INFORMAÇÃO .....	43
2.3.	ANÁLISE DA RELEVÂNCIA DE SISTEMAS DE INFORMAÇÃO NO SETOR BANCÁRIO .....	44
2.4.	MODELAGEM DE SISTEMAS DE INFORMAÇÃO .....	46
2.5.	MODELAGEM ORIENTADA A OBJETOS .....	47
2.6.	UML – <i>Unified Modeling Language</i> .....	48
2.6.1.	Casos de uso .....	49
2.6.2.	Atores .....	49
2.6.3.	Diagrama de casos de uso .....	50
2.6.4.	Definição de requisitos .....	50
2.6.5.	Diagrama de classes .....	52
2.6.6.	Diagrama de sequência .....	54
<b>3</b>	<b>ANÁLISE DA SITUAÇÃO ATUAL .....</b>	<b>56</b>
3.1.	ESTRUTURA GERAL .....	56



3.2. O SISTEMA .....	62
3.3. O PROBLEMA.....	63
3.4. PROPOSTA DE SOLUÇÃO.....	66
<b>4 DEFINIÇÃO DE REQUISITOS.....</b>	<b>71</b>
4.1. REQUISITOS FUNCIONAIS .....	71
4.1.1. Realizar o login .....	71
4.1.2. Boletar operação .....	72
4.1.3. Pesquisar operação.....	74
4.1.4. Consultar operação.....	75
4.1.5. Alterar operação.....	76
4.1.6. Cancelar operação .....	78
4.1.7. Cadastrar ativo .....	80
4.1.8. Aprovar operação.....	81
4.1.9. Consultar posição.....	83
4.2. REQUISITOS NÃO FUNCIONAIS .....	84
4.2.1. Requisitos de produtos .....	84
4.2.2. Requisitos organizacionais.....	85
4.2.3. Requisitos externos .....	86
4.3. REQUISITOS DE INTERFACE .....	86
<b>5 MODELAGEM DE CLASSES .....</b>	<b>92</b>
5.1. GENERALIZAÇÕES .....	93
5.2. DETALHAMENTO DAS CLASSES .....	95
5.2.1. Usuário .....	95
5.2.2. Usuário do front office .....	96
5.2.3. Usuário do middle office .....	97
5.2.4. Usuário do back office .....	97
5.2.5. Produto financeiro.....	98



5.2.6. Par de moedas .....	98
5.2.7. Opção .....	99
5.2.8. Contrato futuro.....	100
5.2.9. Operação .....	100
5.2.10. Operação de moeda.....	102
5.2.11. Operação de opção .....	103
5.2.12. Operação de contrato futuro.....	103
5.2.13. Contraparte.....	104
5.2.14. Mesa .....	104
5.2.15. Estratégia.....	105
5.3. DIAGRAMA DE CLASSES .....	105
<b>6 RESULTADOS.....</b>	<b>109</b>
<b>7 CONCLUSÃO .....</b>	<b>112</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>114</b>



# 1 INTRODUÇÃO

Para que se entenda melhor todo o contexto no qual o presente trabalho foi realizado, é importante explicar brevemente o histórico da instituição, a área de atuação da empresa, as atividades desempenhadas pela equipe do autor e, mais especificamente, as próprias atividades do autor. Em seguida, pretende-se expor o problema a ser estudado, o objetivo traçado para tratar o referido problema, bem como a metodologia para que se consiga, a partir da situação atual, chegar ao estado proposto como objetivo.

## 1.1. A EMPRESA

Primeiramente, vale observar que, para que este Trabalho de Formatura pudesse ser realizado na instituição em que o autor trabalha, decidiu-se, por questões de privacidade, ocultar o seu nome original, bem como o das empresas que fazem parte de sua história. Assim, ressalta-se a escolha por atribuir nomes fictícios a tais organizações, preservando a desnecessária exposição de dados confidenciais.

Dessa forma, a empresa na qual se realizou toda a análise proposta é chamada, durante todo o desenvolvimento deste trabalho, de Banco Silva. Ele é o braço responsável pelas áreas de banco de atacado, investimentos e tesouraria institucional de um grande grupo financeiro. Seu foco está em investidores institucionais e grandes empresas (cujo faturamento anual seja maior que R\$ 150 milhões), prestando serviços e propondo soluções a esses clientes.

A história do banco começa em 1988, quando dois empresários brasileiros se juntam a um banco europeu para formar o Banco 1, banco de atacado com sede em São Paulo. O crescimento da instituição foi acelerado: nos três anos que sucederam a sua criação, o número de profissionais com os quais contava foi multiplicado por dez e o banco já possuía sucursais no Rio de Janeiro e em Campinas.

Em 1991, início da onda de privatizações das estatais no Brasil, o Banco 1 teve importante participação nesse programa do governo ao coordenar o consórcio de bancos estrangeiros.

Em 1994, já com sucursais em Porto Alegre e Belo Horizonte, o Banco 1 continuou se expandindo internacionalmente, assinando um acordo de cooperação com instituições norte-

americanas, além da instalação de uma unidade no Uruguai e, um pouco mais tarde, na Argentina.

Em 1996, ocorreram mais alianças estratégicas. O banco formou, juntamente com um parceiro estadunidense, uma gestora de fundos que chegava a contar com alguns bilhões de dólares. Também se expandiu no mercado de concessão de crédito ao consumidor, com a aquisição de uma popular financiadora brasileira.

Os passos seguintes do crescimento do banco foram dados em 1999, quando assinou um acordo com uma grande corretora, passando a ter participação muito mais representativa no que diz respeito à distribuição de ações, tanto no Brasil como no exterior. Nesse ano, os ativos do Banco 1 já ultrapassavam a dezena de bilhões de dólares.

No entanto, 2002 foi o ano em que houve a maior transformação até então. Um grande banco de varejo, o Banco 2, comprou o Banco 1 e, então, nascia o Banco Silva, com mais de 95% das ações nas mãos do Banco 2 e o restante nas mãos dos sócios do Banco 1. Os setores de concessão de crédito ao consumidor, corretora e *private banking* foram transferidos para outros setores dentro do Banco 2, enquanto o Banco Silva manteria o seu foco em grandes clientes, na tesouraria e no mercado de capitais. Com a modificação na estrutura, o número de colaboradores da instituição cresceu mais de 30%, e o Banco Silva se firmou como um dos maiores bancos de atacado do Brasil.

Em 2005, o banco inaugurou uma unidade de representação em Xangai, na China. No ano seguinte, com o Banco 2 comprando um importante banco norte-americano, o grupo teve significativa ampliação na sua carteira de clientes e na quantidade e diversidade das suas operações.

No final de 2008, foi anunciada a grande mudança na estrutura do grupo, com a fusão do Banco 2 com o Banco 3. Com um ativo total de algumas centenas de bilhões de reais, o novo grupo financeiro constituído se firmava como um dos maiores do mundo. Como consequência, o Banco Silva passou a contar com novos negócios e produtos, além de diversos novos profissionais, cujo número continuava a crescer.

Hoje, o Banco Silva tem escritórios em 8 cidades brasileiras (Belo Horizonte, Blumenau, Campinas, Curitiba, Porto Alegre, Rio de Janeiro, Salvador e São Paulo) e em 12 países estrangeiros (Alemanha, Argentina, Chile, China, Emirados Árabes Unidos, Espanha,

Estados Unidos, França, Ilhas Cayman, Japão, Portugal e Reino Unido). A visualização dessa distribuição pode ser visualizada na Figura 1.1.



Figura 1.1 – Disposição dos escritórios do Banco Silva por país.  
Fonte: Elaborado pelo autor.

Uma vez apresentado o histórico do Banco Silva, pode-se definir mais precisamente o mercado no qual atua, bem como as principais atividades desempenhadas por ele, com a finalidade de se compreender melhor o ambiente no qual este trabalho foi construído.

## 1.2. A ÁREA DE ATUAÇÃO

O contexto do trabalho desenvolvido é a tesouraria do Banco Silva. Uma das funções da tesouraria de uma instituição financeira é prestar serviços a todas as outras áreas da instituição. Essa prestação de serviços envolve garantir a liquidez do banco, fornecendo preços para as diversas áreas do banco que pedem cotações e, assim, tornando viáveis as operações provenientes de clientes através da absorção e gerenciamento dos riscos que são gerados por elas.

Além disso, a tesouraria também deve procurar gerar lucros através da gestão ativa de riscos de mercado, isto é, investindo o capital próprio da instituição em ativos nos quais se considera que há possibilidades de obtenção de lucros.

Outras atividades desempenhadas pela tesouraria institucional são as de estruturar e conduzir as captações próprias do conglomerado, e fornecer a todas as outras áreas do banco pesquisas sobre a visão macroeconômica que o banco possui.

Para ser capaz de realizar essas atividades, a tesouraria é dividida em 4 áreas: *Banking*, *ALM*, *Pesquisa Macroeconômica* e *Trading*.

A área de *Banking* da tesouraria realiza a gestão e a cotação das operações comerciais que geram descasamentos contábeis. Para tanto, deve-se monitorar a situação não só da tesouraria da sede brasileira do banco, mas também de suas tesourarias no exterior. Assim, são controladas as posições das tesourarias externas, supervisionando seus limites de risco e seus níveis de liquidez.

A *ALM* (*Asset and Liability Management*, ou a gestão dos ativos e passivos) é a área da tesouraria responsável pela gestão da liquidez do conglomerado, pela projeção do caixa e pela gestão estrutural e fiscal do balanço. A área é responsável pela estruturação e negociação de transações e contratos com diversos clientes, podendo estas operações ser de *funding* (substituições de dívidas, em geral dívidas de prazos curtos por dívidas mais longas) ou qualquer outro serviço, como gestão de caixa e recursos, derivativos e diversos outros produtos financeiros. A área também está inteiramente ligada ao relacionamento com bancos internacionais que não possuem subsidiárias no Brasil, e com bancos e organismos de desenvolvimento internacionais, como as ECAs (*Export Credit Agencies*).

Uma outra área é a *Pesquisa Macroeconômica*, que centraliza todos os estudos e todas as pesquisas realizadas pelos economistas do banco. Assim, fornecem tanto a visão macroeconômica internamente, isto é, para as outras áreas do banco, como para outras instituições (portanto fala-se, em jargão do mercado financeiro, que realiza tanto o *buy side* como o *sell side*). O foco é alertar sobre os riscos existentes em diversos contextos econômicos, além de discorrer sobre oportunidades e tendências de diversos aspectos da economia global.

Finalmente, há o *Trading*, divisão na qual o autor deste trabalho realizou o seu programa de *trainee* e que é mais bem detalhada a seguir.

### 1.3. ATIVIDADES DA ÁREA DE *TRADING*

A área de *Trading* está dividida nas seguintes equipes: taxas de juros, moedas, ações e *commodities*, volatilidade, *trading* internacional e crédito. Cada equipe realiza tanto operações de cotação para clientes quanto de gestão de risco por vontade própria. Nas primeiras, as cotações para os clientes (*market making*), deve-se cobrir uma vasta gama de instrumentos financeiros, até os mais exóticos, e conseguir fornecer rapidamente preços competitivos. Nas segundas, incluídas no que se chama de *trading* proprietário, é utilizado somente o capital próprio do banco e, a partir dele, busca-se obter rentabilidades significativas respeitando os limites de risco aceitos pela instituição.

Globalmente, pode-se dizer que a área de *Trading* tem três funções principais: precificar, entender e posicionar-se.

Precificar é o que diz respeito à atividade de *market making*. Diversas áreas do banco procuram o *Trading* da tesouraria para obter cotações das mais diversas operações. Normalmente, essas demandas são provenientes de equipes dentro do banco que estão em constante relacionamento com clientes, que são os interessados finais em realizar tais operações. Assim, a principal competência exigida é a de fornecer preços competitivos e de maneira rápida, já que muitos desses pedidos surgem com relativa urgência.

Um segundo papel crucial é o de entender as tendências e as estruturas conceituais do mercado todo. Com isso, é possível não somente fornecer preços justos e competitivos às áreas que lidam com clientes, mas também alertá-las de eventuais oportunidades e ameaças que uma nova operação pode gerar. Tais atividades são possíveis graças ao profundo conhecimento dos fundamentos do instrumento financeiro e do mercado no qual se está atuando. Vale ressaltar que esse entendimento do mercado exige que os envolvidos estejam em constante atualização de tudo o que está ocorrendo no mercado, todas as notícias, tornando a atividade, como um todo, extremamente dinâmica.

Finalmente, com todo esse conhecimento sobre diversos mercados, é também função do *Trading* buscar oportunidades de negócios, visando obter rentabilidades representativas com a utilização do próprio capital do banco. Esse é o chamado *trading* proprietário, no qual o *trader*, com o seu conhecimento e seu poder analítico, toma a iniciativa de comprar ou vender determinados ativos com a expectativa de se obter um bom retorno financeiro.

#### 1.4. O TRAINEE

O trabalho realizado pelo autor ocorreu dentro do contexto do programa de *trainees* do Banco Silva. O programa tem duração de um ano, no qual o *trainee* pode ou passar por diversas áreas dentro do banco (*trainee* generalista) ou focar-se em uma única área (*trainee* especialista), como foi o caso do autor deste Trabalho de Formatura. A área escolhida para se aprofundar mais durante o programa todo foi o *Trading* da tesouraria do banco, tendo feito parte, mais especificamente, da equipe de *Trading* Internacional.

A equipe de *Trading* Internacional é responsável por atuar no mercado exterior, tanto na área de moedas como na de taxas de juros. Assim, a gama de produtos operados é consideravelmente ampla, podendo-se citar *bonds*, *futuros*, *forwards*, *opções* e *swaps*<sup>1</sup>.

Para ser capaz de acompanhar mais de perto tanto o mercado estadunidense quanto o europeu, a equipe conta com analistas em Nova York e em Londres. A vantagem de tal dispersão geográfica é estar mais perto de clientes e de influentes economistas, o que garante que as informações e as tendências dos mercados cheguem rapidamente aos ouvidos da equipe. Além disso, possibilita que ela esteja ativa por um maior número de horas durante o dia, dados os diferentes fusos horários em que se localizam as três cidades. No entanto, o chefe da equipe encontra-se em São Paulo, na sede do banco, em função da necessidade de participar das principais decisões tomadas pelos altos executivos do banco, incluindo os membros do comitê executivo da instituição. O diagrama representado na Figura 1.2 permite que a estrutura da equipe seja visualizada mais claramente.

---

<sup>1</sup> *Bonds* são títulos de dívida pública ou privada que representam a obrigação do emissor de reembolsar o detentor na data do vencimento e, eventualmente, também em intervalos especificados.

Um contrato futuro representa o compromisso entre duas partes de comprar ou vender certa quantidade de determinada mercadoria ou ativo financeiro, a um preço e em um data futura previamente estabelecidos. Trata-se de um contrato padronizado, cujo valor é reajustado diariamente.

Um *forward* representa o compromisso entre duas partes de comprar ou vender certa quantidade de determinada mercadoria ou ativo financeiro, a um preço e em um data futura previamente estabelecidos. Trata-se de um contrato não padronizado e, portanto, o compromisso é liquidado integralmente na data de vencimento.

Opções são contratos que garantem ao comprador o direito, e não a obrigação, de comprar ou vender determinada mercadoria ou ativo financeiro, a um preço e em um data futura previamente estabelecidos.

*Swaps* são contratos que estabelecem um fluxo de pagamentos entre duas partes em datas previamente estabelecidas

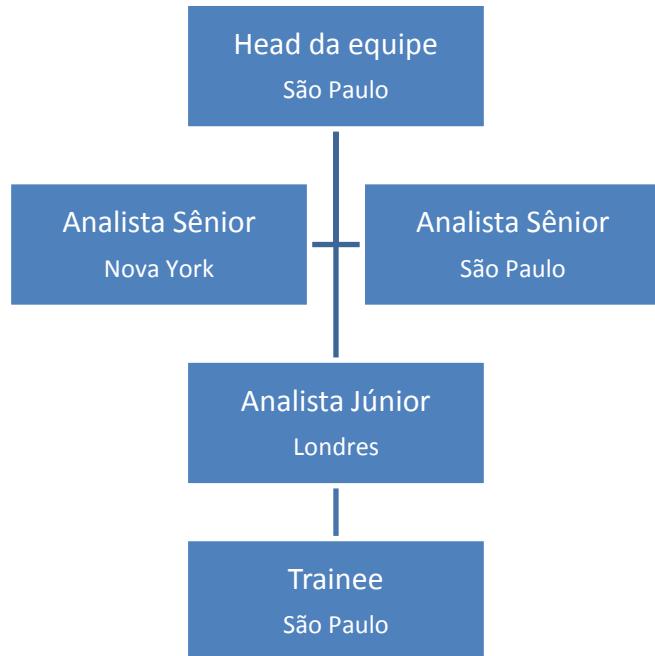


Figura 1.2 – Estrutura da equipe de *Trading* Internacional da tesouraria do Banco Silva.  
Fonte: Elaborado pelo autor.

Dada a configuração da equipe, o *trainee* tem papel fundamental nos processos necessários para o bom funcionamento da estrutura.

Uma das principais atividades do *trainee* é administrar as diversas planilhas de controle da equipe. Elas devem exibir principalmente:

- os ativos operados, mostrando quais já foram zerados e quais ainda possuem posições abertas, separados por contraparte/corretora e por estratégia;
- o risco que esses ativos representam, deixando claro de que maneira estamos expostos frente às diversas oscilações possíveis no mercado;
- o resultado financeiro ao longo do dia para cada um desses ativos, a ser exibido em tempo real em uma tela de comum acesso a todas as equipes do *Trading*.

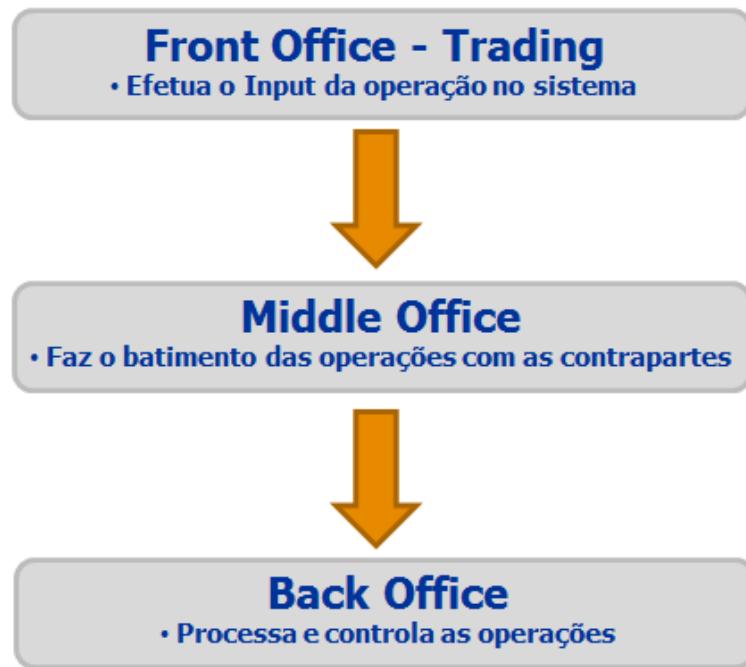
Além dessas atividades gerenciais, o *trainee* também tem a função de introduzir no sistema do Banco Silva todas as operações realizadas. Esse sistema representa, portanto, a base corroborada das operações de toda a tesouraria do banco. Trata-se de um sistema relativamente simples, dispondo de uma tela diferente para o registro de cada produto financeiro e de uma tela para se consultarem as operações registradas. O *layout* atual dessa tela de consulta do sistema pode ser observado na Figura 1.3.

Figura 1.3 – Tela atual de pesquisa do sistema da tesouraria do Banco Silva.

Fonte: Captura de tela do sistema da tesouraria do Banco Silva.

Todo esse processo de inserir as operações no sistema é conhecido como boletagem, e tem também grande importância na medida em que é a forma de o *front office* (no qual se inserem os *traders* e o *trainee*, autor deste trabalho) comunicar ao *middle office* e ao *back office* o que ele reconhece ter feito.

O fluxo das informações entre essas três esferas (*front office*, *middle office* e *back office*) pode ser mais bem compreendido no diagrama da Figura 1.4.

Figura 1.4 – Fluxo das informações, passando por *front office*, *middle office* e *back office*, relacionadas ao registro e processamento de operações no sistema da tesouraria do Banco Silva.

Fonte: Elaborado pelo autor.

Entretanto, não é raro que a contraparte com quem se opera um determinado ativo envie confirmações com dados incorretos ou incompletos. Assim, para que o *middle office* consiga cumprir a sua atividade de conferir todos os dados, comparando-os com os que ele recebe das contrapartes, tem-se a necessidade de se registrarem no sistema todas as operações com todos os detalhes necessários. Somente assim, pode-se ter certeza de que as duas partes envolvidas na compra/venda de um produto concordam com todas as informações referentes à operação.

Outras atividades indispensáveis que dependem desse processo de boletagem são as que dizem respeito ao controle interno do próprio banco. Além de diversas outras áreas do *back office*, há uma responsável apenas por calcular o resultado de cada uma das equipes (que, na prática, é o resultado oficial para o banco), outra responsável por determinar quais os riscos aos quais cada equipe está exposta (para ver se tais riscos se enquadram dentro dos limites pré-estabelecidos) e outra responsável pelo controle do caixa da tesouraria e que irá verificar quais os pagamentos e recebimentos que são devidos em uma determinada data. Citam-se também a equipe responsável pelo controle de produtos (que verificará quais produtos podem ou não ser operados pelo banco), a responsável pela precificação de produtos e o departamento jurídico.

Percebe-se, pois, a necessidade de informar sempre todos os detalhes da operação em uma boleta<sup>2</sup>, como, por exemplo:

- contraparte;
- código do ativo;
- quantidade;
- preço;
- equipe que está operando;
- nome da estratégia dentro da equipe.

E quanto mais complexo for o produto operado, mais detalhes podem ser necessários. Para opções, por exemplo, é necessário também informar a data de vencimento, o preço de exercício e o código do ativo-objeto. Para *swaps*, deve-se acrescentar detalhes como as

---

<sup>2</sup> Boleta é o nome do documento gerado após se registrar uma operação no sistema. Ela contém todos os detalhes da operação que possam, eventualmente, ser necessários.

diversas datas intermediárias de fluxo de caixa, bem como, para cada data, quais as taxas, indexadores e montantes que devem ser considerados para os cálculos.

Uma outra atividade é a de *market making*, viabilizando que a demanda de clientes por cotações seja atendida com precisão e agilidade. Tal exercício envolve especificar o produto em questão, verificar se o produto é aprovado pelo banco para ser operado e verificar com quais contrapartes temos limite de crédito para realizarmos o *hedge* da operação. Além disso, em função desses produtos poderem ter cada um suas particularidades, é sempre fundamental acordar com todos os envolvidos todos os detalhes da operação, evitando divergências futuras.

## 1.5. O PROBLEMA

Uma das competências mais exigidas hoje no mercado financeiro é a agilidade. Em um ambiente tão dinâmico, preços podem mudar em intervalos ínfimos de tempo, o que pode afetar diretamente e representativamente o resultado financeiro de uma operação.

Para que um *trader* seja flexível e consiga reagir rapidamente às oscilações do mercado e a novas ideias que possam surgir, ele deve ter total conhecimento de quais são os riscos aos quais ele está exposto. Assim, é necessário que ele tenha um acesso quase que imediato ao estoque de seus ativos financeiros no exato instante em que desejar.

Essa relação das posições do *trader* poderia e deveria ser gerada pelo sistema no qual tudo que é realizado é boletado. Tendo o registro de todas as operações efetuadas, era de se esperar que o sistema fosse capaz de consolidar tais informações e oferecer várias modalidades de consulta, como por ativo, por data, por corretora, e diversas outras que poderiam ser criadas.

No entanto, o sistema não oferece tais funcionalidades. Essa inflexibilidade do sistema dificulta o dia-a-dia de seus usuários, que não conseguem ter uma base rápida e confiável para consulta em momentos de maior urgência. Como se observa na Figura 1.3, consegue-se pesquisar operações isoladas, de acordo com alguns critérios oferecidos. Contudo, não há como consultar dados relacionados à consolidação dessas operações, o que forneceria, dentre outras informações, o estoque em um dado momento.

Tenta-se contornar tal ausência com o controle que cada *trader* faz por conta própria, através de planilhas em MS Excel®. Entretanto, tal processo estará sempre sujeito a erros de digitação ou qualquer outro tipo de falha humana, o que não garante a confiabilidade necessária que o usuário precisa e que o próprio sistema deveria fornecer.

## 1.6. OBJETIVO

Uma vez posto o problema existente, o próximo passo envolve estabelecer o objetivo do trabalho, norteando todo o seu desenvolvimento.

A questão da rigidez e da falta de flexibilidade do sistema do banco, como foi exposto, dificulta consideravelmente diversas atividades de seus usuários, sobretudo aqueles do *front office*, os chamados *traders*. Dessa forma, o objetivo do trabalho é propor uma solução para esta questão, fazendo com que seja possível que estes usuários disponham de funcionalidades mais completas referentes à consulta das posições de seus ativos financeiros. Com isso, será viável a eles realizar pesquisas a partir de diversas formas de agrupamento de atributos, obtendo relatórios de posição que organizem e propaguem informações importantes de uma maneira simples e satisfatória.

Essa proposta de solução passa pela modelagem de um sistema de informação e, assim, o foco do presente trabalho reside na especificação de requisitos deste sistema e na análise de sua importância em um contexto mais amplo, dentro do ambiente da empresa no qual está inserido.

## 1.7. METODOLOGIA

Já se definiram, neste capítulo, o contexto no qual o presente trabalho se encontra, as atividades do autor dentro deste contexto, o problema levantado e o objetivo a ser atingido. Assim, neste ponto, julga-se conveniente introduzir a metodologia que é utilizada a fim de abordar o problema considerado e alcançar o objetivo traçado.

Para que a análise proposta seja realizada, é preciso investigar todas as características do sistema e de seus usuários, bem como as possíveis relações através das quais estes dois interagem. É igualmente oportuno examinar o fluxo de informações entre todas estas partes

envolvidas, para que se obtenha uma visão mais ampla da importância do sistema de informação dentro do Banco Silva.

A partir destas necessidades, foi desenvolvida a modelagem e, para isso, decidiu-se utilizar a modelagem orientada a objetos e, mais especificamente, a linguagem UML (*Unified Modeling Language*)<sup>3</sup>. A justificativa de tal escolha reside na simplicidade e significativa inteligibilidade oferecidas por ela, considerada ideal em termos de visualização, especificação, construção e documentação de sistemas. Dessa forma, a definição dos requisitos e os diagramas UML, conjuntamente, fornecem, portanto, as ferramentas adequadas para embasar o desenvolvimento e atingir o objetivo deste trabalho.

## 1.8. ESTRUTURA DO TRABALHO

Com o propósito de estimular a fácil compreensão do que se pretende neste Trabalho de Formatura, decidiu-se estruturá-lo em sete capítulos. Ao se realizar esta partição, no entanto, preocupa-se também em garantir que o leitor observe um desenvolvimento lógico e coerente ao longo de todo o trabalho, assegurando a este último o importante aspecto da unidade.

Desta maneira, o Capítulo 1 fornece os elementos essenciais do trabalho, como a definição do contexto, do problema central, do objetivo e da metodologia a ser utilizada.

Em seguida, no Capítulo 2, efetiva-se o levantamento bibliográfico a respeito dos conceitos tratados na análise do problema, apresentando-se definições de noções importantes sob a perspectiva de diversos autores. Além disso, expõem-se também, a partir desta revisão da literatura, análises da relevância de sistemas de informação no contexto da indústria bancária, ambiente no qual o presente trabalho se insere.

Em prosseguimento, no Capítulo 3, são apresentadas a estrutura atual do fluxo de informações estudado e as principais características do sistema em questão. Define-se, então, o problema a ser abordado e, a partir dele, é exposta a proposta de solução, que conduz todo o desenvolvimento dos capítulos seguintes deste trabalho.

---

<sup>3</sup> A *Unified Modeling Language* é uma linguagem de modelagem oriunda do trabalho de Booch, Rumbaugh e Jacobson (1998), e atualmente é considerada um padrão de notação para modelagem de múltiplas perspectivas de sistemas de informação.

No Capítulo 4, definem-se os requisitos para o sistema proposto. Apresentam-se os requisitos funcionais, representados pelos casos de uso e acompanhados de seus diagramas correspondentes, bem como os requisitos não funcionais e os requisitos de interface.

Adiante, no Capítulo 5, desenvolve-se a modelagem do sistema estudado, com a especificação das classes, o detalhamento de cada uma destas classes selecionadas e os casos de generalização que ocorrem entre elas. A modelagem é encerrada com a elaboração do diagrama de classes.

No Capítulo 6, são comentados os resultados deste trabalho, descrevendo a importância da modelagem que foi realizada e expondo a análise dos benefícios que o sistema proposto oferece aos seus usuários.

Finalmente, no Capítulo 7, conclui-se formalmente o trabalho, com a análise crítica do trabalho realizado e de seus resultados, a discussão das vantagens e desvantagens dos métodos utilizados e a recomendação de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar a revisão bibliográfica consultada para o desenvolvimento do trabalho. Inicialmente, apresentam-se conceitos como dado e informação, sem os quais não seria possível prosseguir às definições de sistemas de informação. Em seguida, expõem-se casos que demonstram a relevância do tema em questão no contexto de organizações bancárias. Finalmente, discorre-se sobre modelagem orientada a objetos e, mais especificamente, linguagem UML, acompanhada de suas ferramentas, tão relevantes para o desenvolvimento deste trabalho.

### 2.1. INFORMAÇÃO

Desde sempre uma informação precisa e rápida foi algo valioso, sendo que possuí-la ou não poderia representar a diferença entre uma empresa prosperar ou falir. No entanto, na segunda metade do século XX, os avanços tecnológicos cada vez mais acentuados passaram a dar à informação um valor antes inimaginável.

Um dos primeiros a escrever sobre o assunto foi Touraine (1969), segundo o qual o poder passaria a não estar mais com a burguesia, dona dos meios de produção, mas sim com os possuidores de informação. Bell (1974) sinalizou uma mudança estrutural que estava ocorrendo, com a transição da sociedade industrial para a pós-industrial. Na primeira, a base estava na produção de bens industriais e o poder pertencia aos capitalistas, enquanto na nova sociedade que estava sendo constituída, a base passaria a estar nos serviços e, assim, o poder passa a estar inteiramente ligado à noção de informação. De acordo com Stonier (1983), a informação deixava para trás a terra, o trabalho e o capital e assumia a posição de insumo mais importante naquele momento.

A definição de *informação* passa necessariamente pela de *dado*. Segundo Schmidt (2002), *dado* pode ser definido como um elemento em estado bruto, primário e isolado, que não tem um significado por si só para gerar uma ação. Bio (1996) define o termo de maneira muito próxima, dizendo que dado é um elemento da informação que, tomado isoladamente, não transmite nenhum conhecimento.

Assim, dentro do contexto do presente trabalho, a definição que melhor se adapta à palavra *informação* é dada por Padoveze (2002), que diz que ela é o dado que foi processado

e armazenado de forma compreensível para seu receptor e que apresenta valor real ou percebido para suas decisões correntes ou prospectivas. Para este mesmo autor, o valor da informação está, portanto, relacionado aos seguintes fatores:

- redução da incerteza durante o processo de tomada de decisões;
- relação do benefício gerado pela informação comparado ao custo de produzi-la;
- aumento da qualidade das decisões.

O que se nota, portanto, é que o *dado* é uma espécie de matéria-prima para que se gere uma *informação*, que tem a característica de agregar valor ao seu receptor, fornecendo-lhe algo útil para a sua tomada de decisão. Uma empresa pode possuir uma infinidade de dados, mas, se não souber transformá-los em informação, não conseguirá jamais tomar decisões a partir deles.

## 2.2. SISTEMAS DE INFORMAÇÃO

Uma vez que já se compreenderam o que é um *dado*, o que é uma *informação* e como recentemente esta última ganhou importância a uma velocidade vertiginosa, pode-se apresentar o conceito de sistemas de informação.

Segundo Gil (1999), os sistemas de informação compreendem um conjunto de recursos humanos, materiais, tecnológicos e financeiros agregados de acordo com uma sequência lógica para o processamento dos dados e a correspondente tradução em informações.

O'Brien (2004) os define de maneira semelhante ao afirmar que sistema de informação é um conjunto organizado de pessoas, *hardware*, *software*, redes de comunicações e recursos de dados que coleta, transforma e dissemina informações em uma organização.

De acordo com Laudon e Laudon (2007), um sistema de informação é um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações.

A importância de se ter um desenvolvido sistema de informação dentro de uma empresa está relacionada ao crescente número de informações que circulam diariamente, bem

como ao valor cada vez maior que elas possuem nos dias de hoje. Assim, a empresa deve estar preparada para lidar com eventuais problemas ou mudanças, sejam estas internas ou externas, e somente um sistema de informação muito bem estruturado é capaz de garantir à empresa eficiência e eficácia no processo de tomada de decisões.

Além desse fator relacionado à resolução de problemas, um bom sistema de informação também possibilita à empresa visualizar oportunidades e tomar decisões que a melhorem como um todo, principalmente em termos de versatilidade, organização, redução de custos operacionais e segurança.

Lacombe e Heilborn (2003) afirmam que um sistema de informação é importante em uma empresa por fornecer um conjunto de informações organizadas de forma inteligível. No entanto, argumentam que o seu principal diferencial reside em características como integração, consistência, processamento e comunicação, assim como eficácia e utilidade gerencial para ações e providências administrativas em tempo hábil.

Tendo definido o que é um sistema de informação e como ele é capaz de proporcionar vantagens competitivas a uma empresa, pode-se discorrer sobre os preparativos fundamentais para colocá-lo em prática. Para tanto, é crucial que se estudem previamente quais os usuários deste sistema, qual é a participação de cada um deles e quais os requisitos para a instauração do sistema em questão. Somente após essas etapas torna-se possível que o sistema de informação atenda com sucesso às expectativas dos envolvidos.

A ferramenta utilizada nesses estudos é a modelagem de sistemas de informação, apresentada na seção 2.4. Todavia, antes desta, julga-se relevante analisar a capacidade que sistemas de informação têm em agregar valor a instituições do setor bancário, questão exposta a seguir.

### **2.3. ANÁLISE DA RELEVÂNCIA DE SISTEMAS DE INFORMAÇÃO NO SETOR BANCÁRIO**

Com o recente crescimento da importância de sistemas de informação nas organizações, criou-se consequentemente uma nova questão a ser abordada: como medir os ganhos obtidos em função de melhorias em sistema de informação.

Brynjolfsson e Hitt (1998) expõem as dificuldades que os gerentes de sistemas de informações enfrentam para conseguir justificar investimentos dentro de uma empresa. Estes obstáculos são:

- os erros de medida de *input* e *output*;
- a defasagem entre custos e benefícios;
- a redistribuição e dissipação de lucros;
- as falhas de gerenciamento.

O autor pesquisou análises já realizadas sobre as melhorias de processo no setor bancário, contexto dentro do qual é realizado o presente Trabalho de Formatura. Maçada e Becker (2001) analisaram a eficiência de 41 bancos brasileiros entre os anos de 1995 e 1999. Neste estudo, utilizaram um modelo de efetividade de conversão, proposto por Weill (1989), que diz respeito à capacidade de converter os investimentos em TI em valores que possam trazer retornos para o banco. Desta forma, Maçada e Becker (2001) assumem que os investimentos em TI, juntamente com outros *inputs* (diversas formas de despesas), transformam-se indiretamente em *outputs* (receitas).

A seleção dos *inputs* e *outputs* deve ser muito criteriosa, uma vez que se deve tentar maximizar a relevância e a relação entre eles. No estudo citado, essa escolha envolveu entrevistas com executivos do setor, seguidas por análises estatísticas para justificar a coerência das informações obtidas. Com essas variáveis escolhidas, o próximo passo foi adaptá-las para dados observáveis nas contas de balanço dos bancos, fornecidas pelo Banco Central do Brasil, viabilizando toda a análise.

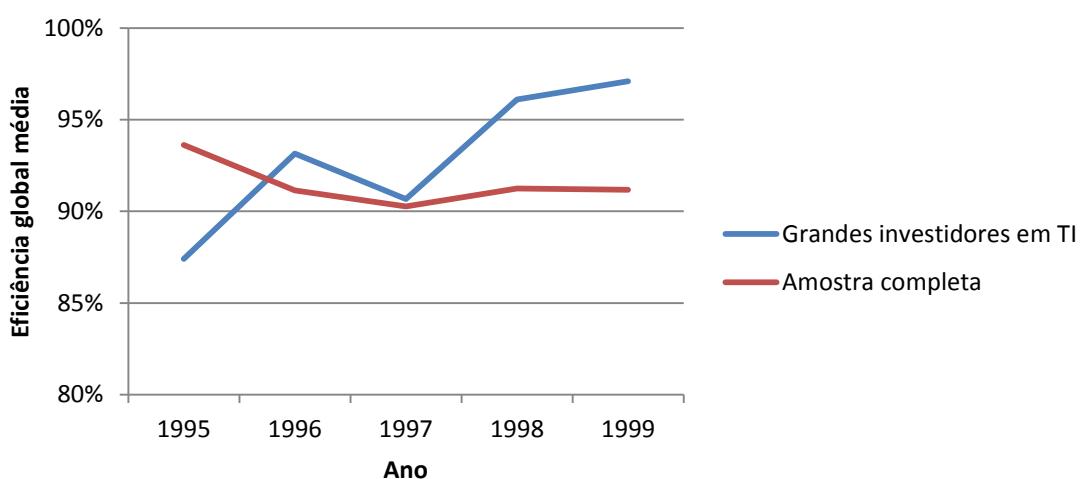


Figura 2.1 – Eficiência global média de 41 bancos brasileiros entre os anos de 1995 e 1999.  
Fonte: Adaptado de Maçada e Becker (2001).

O estudo conclui que os bancos brasileiros que investiram mais em TI ganharam eficiência ao longo do tempo, como se observa na Figura 2.1. Assim, uma vez exposto o resultado do estudo recém-apresentado, considera-se conveniente, no contexto do desenvolvimento do presente trabalho, que se avance à próxima etapa, na qual se definem os conceitos referentes à modelagem de sistemas de informação.

## 2.4. MODELAGEM DE SISTEMAS DE INFORMAÇÃO

Segundo Houaiss (2009), a palavra *modelo* vem do italiano *modello*, que significa “protótipo, imagem que se copia em escultura ou pintura, representação em pequena escala do que se quer executar em tamanho maior (...).” Esse primeiro significado diz respeito, por exemplo, às maquetes, sempre utilizadas para a construção de grandes obras, que têm o objetivo de fornecer uma representação realista e simplificada de um sistema ou objeto para testar e verificar diversos aspectos e comportamentos.

O modelo tratado no presente trabalho, apesar de não corresponder exatamente à definição citada no parágrafo anterior, mantém a ideia principal contida nela: a simplificação da realidade. Esta é a característica comum a todos os modelos, sejam eles modelos matemáticos, modelos econômicos ou modelos de sistemas de informação.

Há limites na competência humana de compreender complexidades. Assim, ao se elaborar um modelo, todas as informações secundárias são descartadas, expondo-se somente os pontos fundamentais, o que facilita a visualização, a compreensão, a identificação de fatores de sucesso ou fracasso, a simulação de comportamentos, a documentação e o debate. Rumbaugh (1994) argumenta também que elementos irrelevantes podem imprecisamente restringir decisões e ter como consequência a perda do foco das questões que realmente interessam.

Vale ressaltar que a definição do que é supérfluo e, logo, deve ser omitido e do que é fundamental para ser considerado no modelo não é uma escolha única. Dependendo do que se quer estudar, pode-se omitir ou não diferentes detalhes. De acordo com Booch, Rumbaugh e Jacobson (1998), modelos podem ser expressos em diferentes níveis de precisão, e a escolha de qual modelo adotar influencia diretamente a abordagem e as soluções vislumbradas para o problema.

Assim percebe-se a enorme importância de se optar pelo modelo que melhor se adapte às necessidades do projeto. No caso deste trabalho, opta-se pela modelagem orientada a objetos, detalhada a seguir.

## 2.5. MODELAGEM ORIENTADA A OBJETOS

A história da modelagem orientada a objetos começa em maio de 1967, quando Ole-Johan Dahl e Kristen Nygaard, ambos noruegueses, propuseram a linguagem de programação Simula 67, a primeira a introduzir os novos conceitos. No entanto, a percepção de seu impacto só viria treze anos mais tarde, com a publicação da linguagem Smalltalk-80. Hoje, linguagens mundialmente difundidas utilizam a programação orientada a objetos, como, para citar algumas, C++, C#, Java, Python e Ruby.

A orientação a objetos é baseada na elaboração de diversas unidades, denominadas objetos, que possuem atributos e comportamentos. Um grupo de objetos com particularidades em comum são agrupados sob a estrutura estática de uma classe, que irá definir quais características (atributos) eles terão, bem como quais tipos de ações (métodos) eles poderão realizar. Já a interação de um objeto com o outro é feita através de mensagens, que são chamadas para que um objeto realize alguma ação descrita em seus métodos.

Para exemplificar os conceitos recém-apresentados, considera-se a classe “Ser Humano”. Um objeto pertencente a essa classe é o José, que possui atributos como altura, peso e idade, e métodos como andar, sorrir e comer. O José pode interagir com a Maria, outro objeto da mesma classe, através, por exemplo, da fala, que é uma mensagem enviada à Maria e faz com que ela sorria, realizando a ação correspondente a um de seus métodos da classe “Ser Humano”.

De acordo com Burleson (2012), optar pela modelagem orientada a objetos proporciona as seguintes vantagens:

- facilidade de manutenção, implicando reduzidos custos relacionados;
- maior facilidade para descrever estruturas e identificar problemas;
- abrange todos os graus de complexidade ou tamanho;
- possibilidade de constante reutilização de objetos e classes.

Vistas as características da modelagem orientada a objetos, especifica-se, adiante, a linguagem que é utilizada ao longo do desenvolvimento deste trabalho, a UML.

## 2.6. UML – *Unified Modeling Language*

A UML (*Unified Modeling Language*) é uma linguagem de modelagem que nasceu de um trabalho conjunto de Grady Booch, James Rumbaugh e Ivar Jacobson. Em 1997, ela passou a ser considerada um padrão de notação para modelagem de múltiplas perspectivas de sistemas de informação pela OMG (*Object Management Group*), órgão internacional que tem como objetivo garantir a interoperabilidade e a portabilidade de aplicações com tecnologia de objetos.

Essas múltiplas perspectivas, que podem ser estruturais (estáticas) ou comportamentais (dinâmicas), são representadas por uma série padronizada de diagramas e notações, que são detalhadas adiante.

Segundo Booch *et al.* (1999), a UML tem como conceitos primordiais quatro atributos, apresentados a seguir:

- Visualização: o modelo visa ser facilmente compreendido por pessoas que jamais tiveram contato com o assunto do qual o sistema trata. Essa simples e rápida visualização permite que se entenda a semântica por trás de cada notação.
- Especificação: busca-se a construção de um modelo preciso, cuja interpretação não dê margem a quaisquer ambiguidades.
- Construção: é possível que os modelos baseados em UML se conectem facilmente a diversas linguagens orientadas a objetos, como C++, Java e Visual Basic.
- Documentação: a linguagem é capaz de documentar toda a arquitetura do sistema, oferecendo também ferramentas para detalhar requisitos e modelar atividades de planejamento e gerenciamento.

### 2.6.1. Casos de uso

Idealizados por Ivar Jacobson em 1986, os casos de uso representam, segundo o OMG (2011), o que o sistema deve fazer. De maneira simples, eles devem ilustrar como as funções pretendidas pelo sistema geram um ou mais benefícios para o cliente ou usuário (que são os atores, cuja explicação mais extensa é apresentada adiante).

Ressalta-se que eles não descrevem como será a construção do sistema. O que é descrito é como o sistema se comportará uma vez construído.

Assim, pode-se dizer que os casos de uso constituem uma base importante para que se consiga definir:

- classes e operações;
- descrições do funcionamento detalhado do produto;
- teste de aceitação;
- roteiro de manual de usuário.

### 2.6.2. Atores

Os atores são peça fundamental na elaboração de casos de uso. São eles que interagem com o sistema, constituindo o ambiente no qual ele funciona. Assim, são considerados atores tanto os que enviam informações ao sistema como os que recebem dele.

De acordo com o OMG (2011), os atores modelam os papéis e não as pessoas dos usuários. Isto é, um mesmo usuário físico pode ser o ator “gerente de vendas” em um caso de uso, e o ator “gestor de estoques” em outro caso de uso.

Uma tarefa nem sempre evidente é a identificação dos atores. De acordo com Pádua (2009), essa distinção pode ser realizada a partir da análise das seguintes condições:

- quem está interessado em certo requisito;
- onde o produto será usado;
- quem se beneficiará do produto;
- quem fornecerá informação ao produto;
- quem usará informação do produto;

- quem removerá informação do produto;
- quem dará suporte e manutenção ao produto;
- quais os recursos externos usados pelo produto;
- quais os papéis desempenhados por cada usuário;
- quais os grupos de usuários que desempenham o mesmo papel;
- quais os sistemas com os quais o produto deve interagir.

#### *2.6.3. Diagrama de casos de uso*

Uma vez apresentadas as definições de casos de uso e de atores, pode-se discorrer sobre o diagrama de casos de uso, outra importante ferramenta. Ele permite que se tenha uma visão externa do sistema, representando graficamente os casos de uso, os atores e a interação entre os dois. Assim, permite-se compreender como o ambiente se relaciona com cada funcionalidade proposta pelo sistema, enriquecendo o entendimento sobre todo o contexto em questão.

De acordo com Pádua (2009), os casos de uso normalmente expressam:

- quais as tarefas de cada ator;
- que informação cada ator cria, armazena, consulta, altera ou remove;
- que informação cada caso de uso cria, armazena, consulta, altera ou remove;
- que mudanças externas súbitas devem ser informadas ao produto pelos atores;
- que ocorrências no produto devem ser informadas a algum ator;
- que casos de uso darão suporte e manutenção ao sistema;
- quais os casos de uso necessários para cobrir todos os requisitos funcionais.

#### *2.6.4. Definição de requisitos*

A análise dos requisitos é importantíssima no que diz respeito à determinação do que o sistema deve fazer para que consiga atender às necessidades dos usuários. Além disso, os requisitos definem as restrições do sistema.

Esse reconhecimento dos requisitos deve ser levantado conjuntamente pelo desenvolvedor e pelo usuário do sistema. Pádua (2009) argumenta que, normalmente, nenhum

desses dois grupos é capaz, individualmente, de especificar os requisitos do sistema. A razão disso é que os usuários não são conhecedores dos processos de desenvolvimento de software, do mesmo modo que os desenvolvedores não têm conhecimento profundo sobre a área de aplicação do sistema.

Costuma-se distinguir os requisitos em três grupos: os funcionais, os não funcionais e os de interface, apresentados a seguir.

De acordo com Sommerville (2007), os requisitos funcionais definem como o sistema deve reagir a certas entradas e se comportar diante de determinadas situações. Podem ser detalhadas todas as entradas, saídas e casos de exceção. Vale ressaltar, entretanto, que eles caracterizam quais funcionalidades espera-se que o sistema deva possuir, mas não *como* o sistema irá desempenhá-las.

Os requisitos não funcionais descrevem propriedades e restrições do sistema, definindo como os requisitos funcionais devem ser alcançados. Também tratam de exigências de qualidade, desempenho e robustez. Sommerville (2007) explica que os requisitos não funcionais podem não estar ligados a uma característica individual do sistema, mas ao sistema como um todo. A consequência disso é que o descumprimento de um desses requisitos não funcionais pode prejudicar todas as funcionalidades do sistema. O mesmo autor explicita todos os tipos de requisitos não funcionais, e estes são expressos na Figura 2.2.

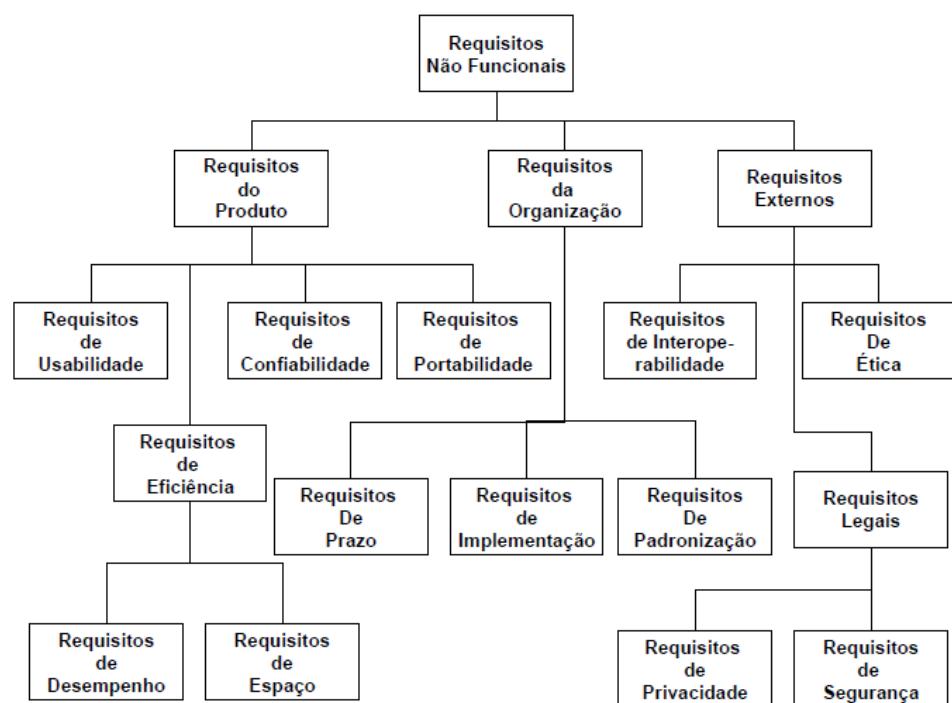


Figura 2.2 – Tipos de requisitos não funcionais.

Fonte: Adaptado de Sommerville (2007).

Finalmente, o terceiro grupo de requisitos são os de interface, que podem ser divididos em duas subcategorias: os de interfaces genéricas e os de interfaces gráficas de usuário. Os primeiros dizem respeito a todas as entradas e saídas do sistema, isto é, arquivos não usados somente pelo sistema e, assim, os seus requisitos também devem ser levados em consideração. Os segundos são requisitos do produto em si, como formato de dados e comando, bem como aspectos de desenho como formatos de telas e janelas.

#### 2.6.5. *Diagrama de classes*

No contexto da modelagem orientada a objetos, o diagrama de classes tem papel fundamental na medida em que expressa o conjunto de classes (com seus respectivos atributos e métodos), interfaces e colaborações e seus relacionamentos. Desta forma, assume papel importantíssimo em termos de visualização, especificação e documentação do modelo em questão, além de servir como base para a elaboração de outros diagramas, como o de comunicação, o de estados e o de sequência.

Uma vez já apresentados os conceitos de objeto e de classe, faz-se necessário descrever como se dá o relacionamento entre as classes. Geralmente, elas colaboram umas com as outras de diversas formas, e essa relação deve ser representada de maneira clara e específica na construção do diagrama de classes.

Para Pádua (2009), os principais relacionamentos são os de associação, que definem relações bidirecionais de dependência semântica entre duas classes estudadas. Neste tipo de relacionamento, um objeto de uma das classes tem conhecimento dos objetos da outra, sendo esse acesso normalmente descrito por um ou mais métodos da primeira classe. Também se pode definir a multiplicidade dessa associação, indicando quantos objetos de uma classe se relacionam com quantos objetos da outra.

Um exemplo de sua representação gráfica é visualizado na Figura 2.3. Pode-se dizer que os objetos da classe “Empresa” têm acesso aos objetos da classe “Mercadoria” através de seus dois métodos “incluirProduto()” e “excluirProduto()”. De maneira análoga, os objetos da classe “Mercadoria” têm conhecimento dos objetos da classe “Empresa”. No que diz respeito à multiplicidade, percebe-se que não há restrições no número de objetos da classe “Empresa”,

nem no número de objetos da classe “Mercadoria”, podendo estes números assumir quaisquer valores inteiros maiores ou iguais a zero.



Figura 2.3 – Exemplo de um relacionamento de associação.

Fonte: Adaptado de Pádua (2009).

Um outro tipo de relacionamento entre classes é o de agregação, que, na verdade, é um caso particular do relacionamento de associação. Na agregação, tem-se a ideia de construção física ou posse lógica, portanto não sendo mais o relacionamento bidirecional. O que se tem são duas classes: a classe parte e a classe todo. Para a sua representação gráfica, a linha apresenta um losango vazio na extremidade mais próxima à parte todo, como ilustra a Figura 2.4.

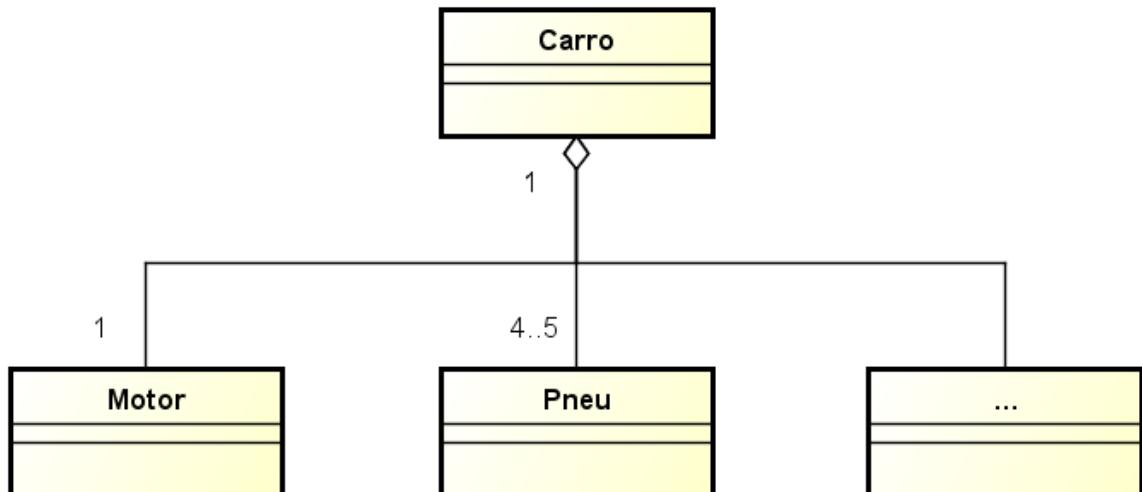


Figura 2.4 – Exemplo de um relacionamento de agregação.

Fonte: Elaborado pelo autor.

Finalmente, um terceiro tipo de relacionamento entre classes é o de composição. Considerado mais forte que os apresentados anteriormente, o relacionamento de composição também envolve duas classes, a parte e a todo. No entanto, diferentemente do que ocorre no relacionamento de agregação, os objetos da classe parte têm a sua existência condicionada à dos objetos da classe todo. Nesta representação, a linha que liga as duas classes apresenta um

losango preenchido na extremidade mais próxima à parte todo, e um exemplo é apresentado na Figura 2.5.

Ressalta-se, portanto, que, no exemplo de associação da Figura 2.4, um motor e um pneu continuam existindo mesmo que um carro deixe de existir, enquanto no exemplo de composição da Figura 2.5, se uma universidade deixa de existir, os seus departamentos também deixarão de existir.

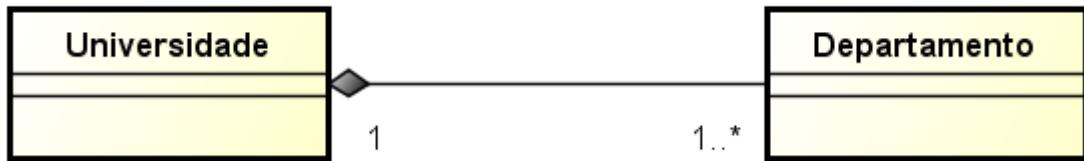


Figura 2.5 – Exemplo de um relacionamento de composição.  
Fonte: Elaborado pelo autor.

#### 2.6.6. Diagrama de sequência

O diagrama de sequência é um modelo dinâmico e, assim, descreve aspectos ligados ao tempo e à sequência de ações que ocorrem, expressando o estado que define o contexto no qual cada evento acontece. O foco é exibir de maneira simples a ordem cronológica das mensagens que são enviadas entre os objetos de um sistema. Um exemplo pode ser observado na Figura 2.6.

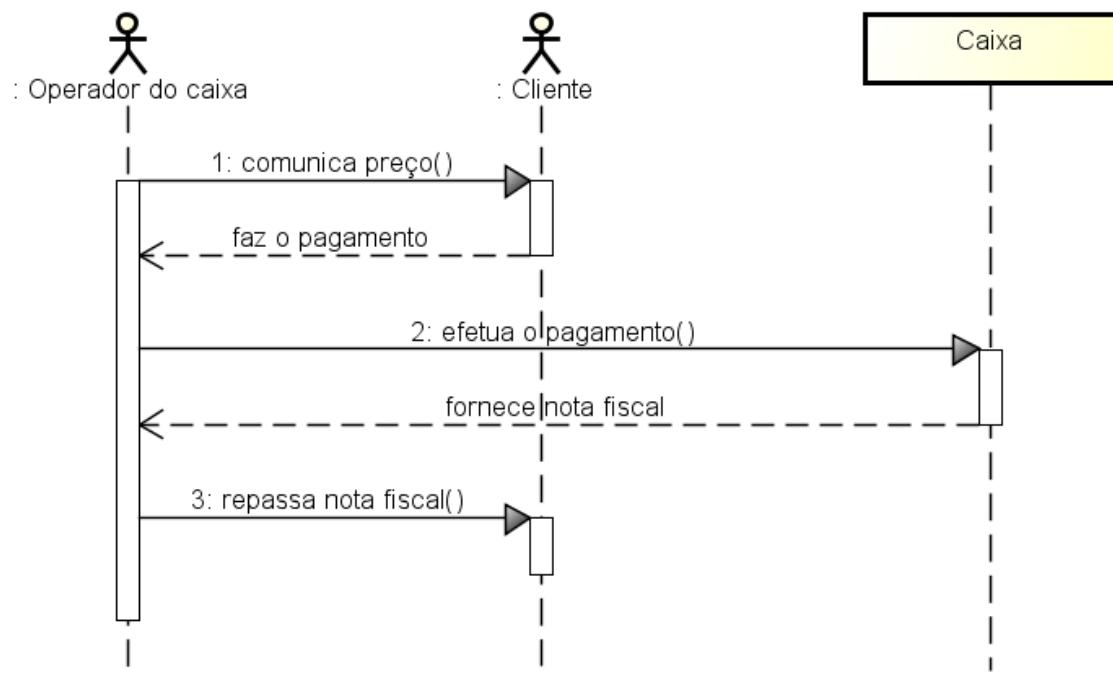


Figura 2.6 – Exemplo de diagrama de sequência.  
Fonte: Elaborado pelo autor.

Assim, após terem sido apresentados os principais conceitos teóricos envolvendo sistemas de informação e modelagem orientada a objetos, já se dispõe de todas as ferramentas necessárias para a apresentação da análise realizada no Banco Silva. Dessa forma, o próximo capítulo versa sobre a situação atual do contexto considerado neste estudo.

### 3 ANÁLISE DA SITUAÇÃO ATUAL

Uma vez realizada a revisão bibliográfica dos tópicos a serem abordados, consegue-se prosseguir à aplicação de tais conceitos no contexto da organização na qual se efetua o presente trabalho. Dessa maneira, este capítulo é iniciado com a apresentação da estrutura atual de todo esse contexto, seguida da definição das principais características do sistema estudado. Posteriormente, é delineado o problema a ser tratado, e é exposta a proposta de solução para esse problema, que envolve um sistema mais flexível, que ofereça ao usuário diversas possibilidades de agrupamento de atributos ao realizar uma consulta.

#### 3.1. ESTRUTURA GERAL

Para o bom funcionamento das atividades do banco, é necessário que toda a série de processos internos de sistemas de informação seja bem sucedida. No caso específico do presente trabalho, decide-se estudar a implementação de novas informações geradas pelo sistema ao *trader*. Entretanto, antes se propõe analisar como o sistema se relaciona com os seus usuários dadas algumas situações, e qual a sequência de ações que são tomadas por estes usuários a partir do momento em que se realiza um *trade*.

Ressalta-se o fato de que o autor deste trabalho é um dos usuários do sistema estudado, o que lhe possibilitou uma análise crítica mais aguçada. Além de sua experiência pessoal, foram realizadas entrevistas informais com os demais usuários do sistema, tornando possível o aprofundamento das principais questões que envolvem a interação desses usuários com o sistema.

Inicia-se, assim, com o caso de um cenário básico, no qual não há erros cometidos por nenhuma das partes envolvidas. Essa sequência pode ser observada na Figura 3.1.

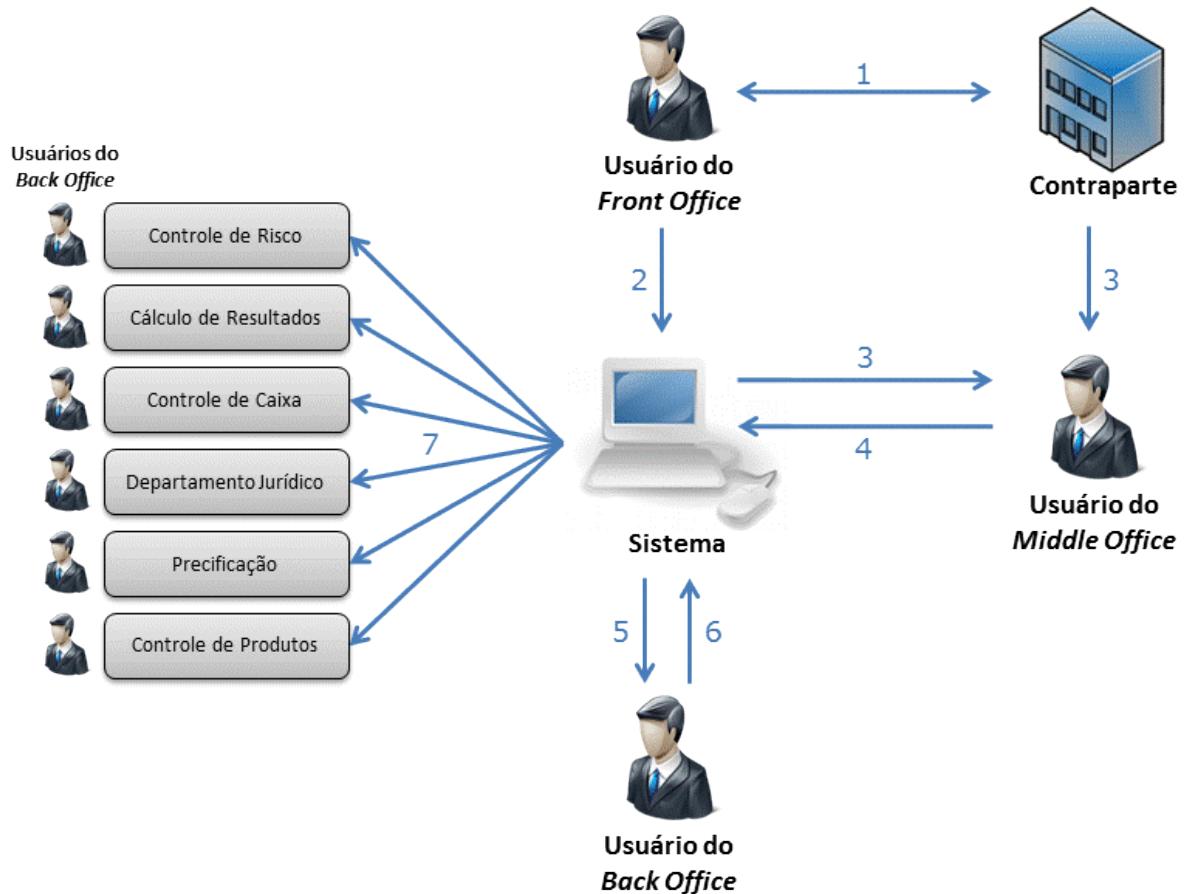


Figura 3.1 – Sequência de ações que se sucedem após a realização de um *trade* quando não há erros cometidos por nenhuma das partes envolvidas.

Fonte: Elaborado pelo autor.

Os números indicados na Figura 3.1 representam as ações descritas a seguir:

**1** – O usuário do *front office* e a contraparte (que pode ser um banco, uma corretora ou uma outra equipe dentro do próprio banco) concordam com a realização de um *trade*, no qual uma das partes comprará e a outra venderá um determinado produto financeiro.

**2** – O usuário do *front office* boleta a operação no sistema, informando todos os detalhes referentes a ela.

**3** – O usuário do *middle office* recebe, com todos os seus detalhes, as informações referentes ao *trade* tanto da contraparte como do sistema, que reflete os dados informados pelo usuário do *front office*.

**4** – O usuário do *middle office* compara as 2 confirmações que recebe, certificando-se de que os dados informados pelo usuário do *front office* correspondem exatamente aos

informados pela contraparte. Após tal verificação, o usuário do *middle office* aprova a operação no sistema.

**5** – Uma vez que a operação foi aprovada no sistema pelo usuário do *middle office*, ela passa a estar visível ao usuário do *back office*, que poderá ter acesso a todos os seus detalhes.

**6** – O usuário do *back office* confere novamente se todos os dados da operação estão coerentes e, caso estejam, aprova a operação no sistema. Essa dupla certificação é um mecanismo encontrado para mitigar riscos decorrentes de falha humana.

**7** – Finalmente, após o usuário do *front office* ter registrado no sistema, o do *middle office* ter conferido e aprovado e o do *back office* ter conferido novamente e aprovado, a operação estará apta a ser processada para todas as bases de controle do banco. Diversas equipes do *back office* usarão essas bases processadas para várias finalidades, como o controle de risco, o cálculo de resultados, o controle de caixa, o controle jurídico, o controle de preços, o controle de produtos etc.

A atual configuração dos relacionamentos do sistema apresenta pontos positivos importantes, como a clara separação entre as tarefas do *front office*, do *middle office* e do *back office*. Isto é, o sistema permite que a cadeia descrita na Figura 1.4 seja cumprida com as condicionalidades necessárias para que se reduzam os problemas operacionais. Em outras palavras, o usuário do *back office* só conseguirá visualizar um *trade* para aprová-lo depois que o usuário do *middle office* o aprovou. Este último, por sua vez, só visualizará o *trade* para aprová-lo depois que o usuário do *front office* o registrou no sistema.

Contudo, como previamente ressaltado, o fluxo descrito na Figura 3.1 é o padrão, mas nem sempre é o que de fato ocorre. Considera-se, por exemplo, o caso no qual o usuário do *middle office* detecta um erro na boleta quando realiza a conferência entre o que está registrado no sistema e o que a contraparte envia. A sequência de ações que descreve o fluxo passa a ser, portanto, a ilustrada na Figura 3.2.

Tal divergência encontrada pelo usuário do *middle office* pode ser em decorrência tanto de um dado incorreto inserido pelo usuário do *front office* (ações 5a e 6a), como de um dado incorreto enviado pela contraparte (ações 5b e 6b).

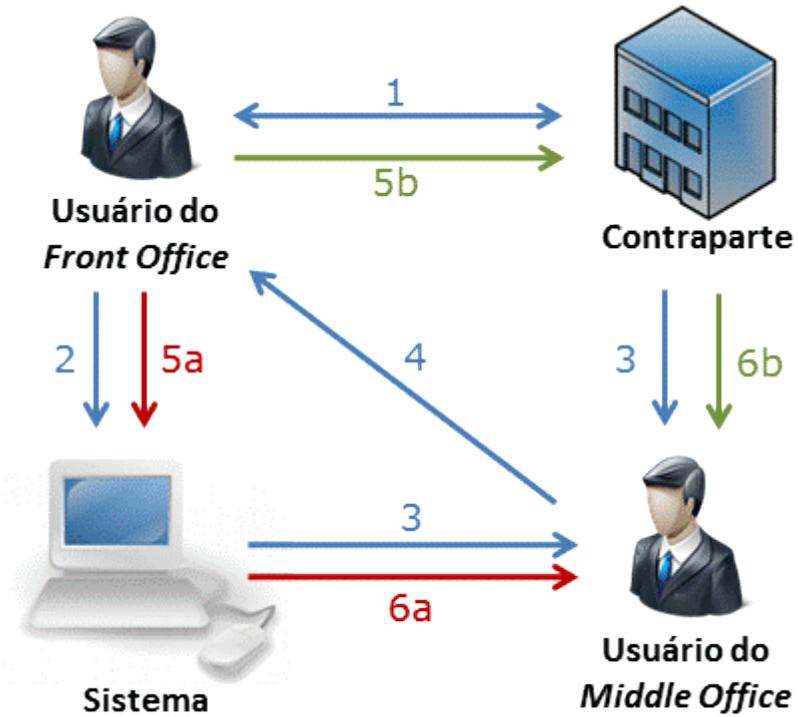


Figura 3.2 – Sequência de ações que se sucedem após a realização de um *trade* quando o usuário do *middle office* detecta uma divergência.

Fonte: Elaborado pelo autor.

Neste novo fluxo, acrescentam-se algumas novas ações, detalhadas a seguir:

**4** – O usuário do *middle office* detecta a divergência entre o que foi registrado no sistema e o que recebeu da contraparte, e comunica tal fato ao usuário do *front office*.

**5 – 5a** – O usuário do *front office* reconhece que na boleta original havia um dado incorreto, e altera esta boleta corrigindo o dado em questão; ou **5b** – O usuário do *front office* não reconhece que na boleta original havia um dado incorreto, e comunica a contraparte a respeito de tal divergência.

**6 – 6a**: O usuário do *middle office* visualiza a operação alterada no sistema e confere esta com o que foi enviado pela contraparte; ou **6b** – A contraparte reconhece que havia enviado algo errado ao usuário do *middle office* e, notificada pelo usuário do *front office*, reenvia-lhe a operação. O usuário do *middle office* confere novamente a operação boletada no sistema com o que foi enviado pela contraparte, agora com os dados alterados.

Outro caso possível, ilustrado na Figura 3.3, é quando o usuário do *back office* detecta alguma incoerência na boleta ou a falta de algum dado que será relevante para as equipes de controle. Mais uma vez, tal fato pode ser em decorrência tanto de um dado incorreto ou

inexistente inserido pelo usuário do *front office* (ações 8a e 9a), como de um dado incorreto ou inexistente enviado pela contraparte e não identificado pelo *middle office* (ações 8b e 9b).

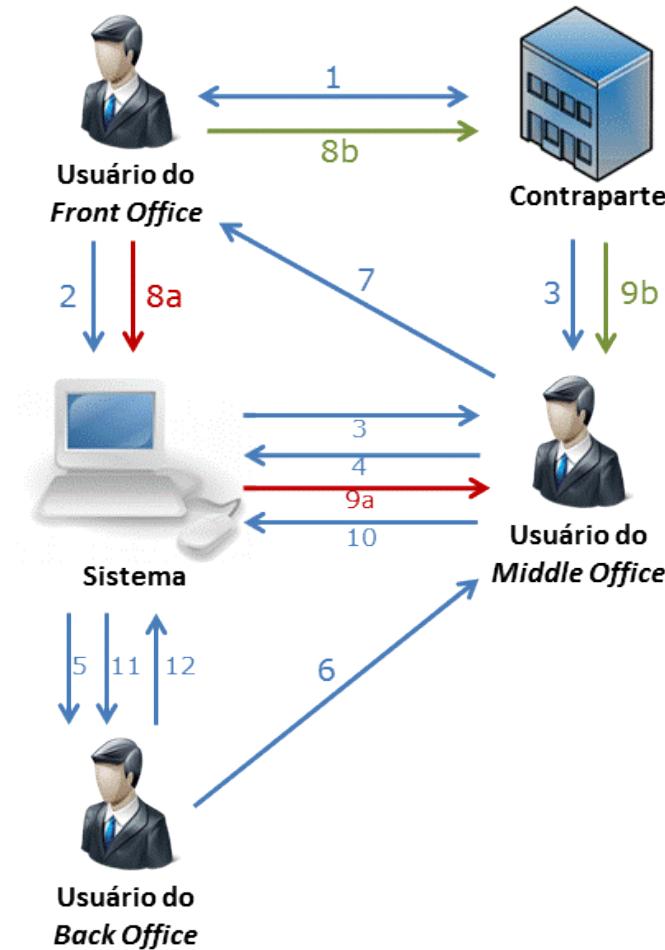


Figura 3.3 – Sequência de ações que se sucedem após a realização de um *trade* quando o usuário do *back office* detecta uma divergência.

Fonte: Elaborado pelo autor.

Neste novo fluxo, as ações que são acrescentadas são as seguintes:

**6** – O usuário do *back office* detecta uma incoerência ou a falta de um dado na boleta aprovada pelo usuário do *middle office*, comunicando-lhe tal fato.

**7** – O usuário do *middle office* comunica essa inconsistência na boleta ao usuário do *front office*.

**8** – 8a – O usuário do *front office* reconhece que na boleta original havia um dado inconsistente ou a falta de algum dado, e altera esta boleta corrigindo o problema em questão; ou 8b – O usuário do *front office* comunica a contraparte a respeito de tal inconsistência e solicita que a contraparte providencie a alteração necessária.

**9 – 9a:** O usuário do *middle office* visualiza a operação alterada no sistema e confere esta com o que foi enviado pela contraparte; ou **9b** – A contraparte, notificada pelo usuário do *front office*, reenvia ao usuário do *middle office* a operação com as devidas correções.

**10** – O usuário do *middle office* compara os dados informados pelo usuário do *front office* com os informados pela contraparte. Após tal verificação, o usuário do *middle office* aprova a operação no sistema.

**11** – Uma vez que a operação foi aprovada no sistema pelo usuário do *middle office*, ela passa a estar visível ao usuário do *back office*, que poderá ter acesso a todos os seus detalhes.

**12** – O usuário do *back office* confere novamente se todos os dados da operação estão coerentes e, caso estejam, aprova a operação no sistema.

Os casos expostos são alguns exemplos de como o fluxo ideal de informação, ao se realizar um *trade*, está sujeito a diversas modificações. Tais mudanças podem ser decorrentes de erros produzidos por qualquer uma das partes envolvidas no processo, que não são poucas. Assim, tenta-se comprovar a exatidão dos dados em diversos pontos de controle no sistema, para identificar eventuais equívocos e poder corrigi-los o mais cedo possível, não permitindo a sua propagação e, portanto, evitando que se gerem trabalhos desnecessários que terão de ser refeitos posteriormente.

Entretanto, esses mecanismos nem sempre conseguem evitar a tempo que erros sejam produzidos. E, uma vez que se detecta um erro em uma operação que não foi registrada no sistema no próprio dia, a alteração da mesma requer a autorização de diversas pessoas. Para a modificação de uma boleta, deve-se obter o consentimento do *trader* responsável pela operação, do chefe da equipe do *trader* responsável pela operação e do chefe de área de suporte ao negócio, setor da tesouraria que tem como finalidade supervisionar essas questões organizacionais e de processos. Consequentemente, todo esse processo pode demorar algum tempo, nem sempre sendo possível a alteração devida no mesmo dia em que se identifica a sua necessidade.

### 3.2. O SISTEMA

O sistema do banco é utilizado como base oficial para todos os controles existentes realizados pelo *back office*, como ilustra a Figura 3.1. Considera-se que as operações que foram inseridas pelo *front office*, aprovadas pelo *middle office* e novamente aprovadas pelo *back office* são aquelas tidas como referência, como o que o banco reconhece ter operado. Cada dado contido em cada uma dessas boletas será o utilizado por todos os responsáveis pelas mais variadas tarefas.

Essas várias equipes do *back office* têm, portanto, uma grande necessidade de interagir eficientemente com o sistema. Se a ligação entre ele e essas equipes que utilizam diversas vezes ao dia os dados contidos nele for simples e prática, haverá a possibilidade de se ganhar tempo consideravelmente, além de reduzir a chance de se cometerem equívocos.

Atualmente, é possível extrair do sistema bases de dados em formato MS Excel® com todos os atributos das operações que as equipes de controle do *back office* necessitam. Assim, elas utilizam essas várias bases para alimentar outros sistemas e outras planilhas nos quais realizam cada uma a sua tarefa específica.

No que diz respeito às tarefas do *middle office*, isto é, a conferência das operações inseridas no sistema pelo *front office* com o que é enviado pelas contrapartes, a interação com o sistema é muito semelhante. Consegue-se extrair em planilhas MS Excel® a base das operações boletadas que ainda estejam pendentes de aprovação. Assim, o *middle office* pode separar tais operações em:

- as que já estão conferidas, possuindo uma operação correspondente idêntica enviada pela contraparte;
- as que provavelmente possuem algum dado incorreto, já que se identificou uma operação correspondente enviada pela contraparte na qual todas as especificações estão idênticas com exceção de um dado que está divergente;
- as que ainda não possuem nenhuma operação candidata a ser a correspondente enviada pela contraparte.

Finalmente, no que se refere à interação do *front office* com o sistema, observa-se que essa ligação poderia ser alvo de melhorias significativas. Ao passo que o usuário registra as operações realizadas no sistema, este último praticamente não fornece ao primeiro informações ou relatórios gerenciais. Toda essa questão é mais amplamente detalhada a seguir.

### 3.3. O PROBLEMA

Como se explicitou neste capítulo até aqui, o *trader* está inserido em um ambiente relativamente desorganizado. De um lado, tem-se toda a estrutura complexa e não ágil da sequência de ações que são tomadas após a realização de um *trade* e, especialmente, do que é necessário fazer para que se consiga alterar boletas no sistema quando for preciso. Ao mesmo tempo, tem-se um sistema que não lhe oferece dados básicos, como o estoque das suas posições em um dado instante. Dado tal contexto de trabalho, a consequência é que o *trader* acaba não conseguindo ter a agilidade e a reatividade ideais.

No mercado financeiro, os preços se atualizam com enorme velocidade, a cada notícia que é divulgada, a cada dado econômico que é publicado, a cada frase dentro de um discurso de uma pessoa influente, isto é, a cada evento que possa alterar a percepção dos *players* do mercado. Logo, não ter uma boa agilidade pode representar ser tarde demais para se comprar ou vender um ativo pelo valor que se gostaria. O que se tem então é um impacto que está diretamente ligado ao resultado financeiro da operação.

Para exemplificar a importância de se ter agilidade no trabalho do *trader*, tomemos o caso das ações preferenciais da Petrobras, cujo código na Bovespa (Bolsa de Valores de São Paulo) é PETR4. No dia 25 de abril de 2012, observou-se uma queda brusca no valor desta ação em um curtíssimo período de tempo, como é mostrado na Figura 3.4.



Figura 3.4 – Gráfico da evolução do preço da ação PETR4 em 25 de abril de 2012.  
Fonte: Bloomberg®.

Constata-se que o preço da ação era de R\$21,30 às 14:08. Apenas cinco minutos depois, o seu preço havia caído para R\$20,94, o que representa uma queda de 1,69% no valor da ação. Quando se tenta entender o que poderia ter motivado tal oscilação significativa em um espaço tão curto de tempo, percebe-se que a presidente da empresa fazia um pronunciamento neste exato instante. Foi noticiado, às 14:09 do dia em questão, que ela julgava que ainda não estaria no momento ideal para aumentar o preço de combustíveis, como ilustra a Figura 3.5.

Apr 25, 2012 02:09:11 PM [BLOOMBERG Brazilian News (in Portuguese)] Page 1 / 1  
**\*GRAÇA: NÃO ESTÁ NA HORA AINDA DE ELEVAR PREÇO DE COMBUSTÍVEIS**  
 Figura 3.5 – Nota divulgada a respeito de um discurso da presidente da Petrobras, Maria das Graças Foster, em 25 de abril de 2012.  
 Fonte: Bloomberg®.

O discurso da presidente da empresa não foi bem recebido pelo mercado, já que o referido aumento de preço de combustíveis seria visto como positivo para os negócios da empresa. Portanto, com o pronunciamento de que tal aumento ainda não ocorreria, os *players* do mercado passaram a precisar um valor inferior ao que, até então, era considerado justo para a ação.

Assim, a partir deste caso selecionado, pode-se tomar como exemplo um *trader* do banco que possuísse ações PETR4. Ao estar ciente das palavras da presidente, às 14:09, é possível que ele quisesse vender todas as suas ações, já antevendo o movimento de queda. Entretanto, com a ausência de um sistema que lhe forneça o estoque de seus ativos com rapidez e precisão, o *trader* poderia demorar um certo tempo até conseguir ter certeza do número de ações de PETR4 que possuía.

Caso o *trader* precisasse de cerca de cinco minutos para obter essa informação, o preço da ação já estaria em R\$20,94, ou seja, 1,69% menor do que o preço das 14:08. Supondo que a quantidade correta de ações em estoque fosse de 100 mil (representando um pouco mais de 2 milhões de reais, um tamanho modesto para os *trades* realizados na tesouraria do Banco Silva), a demora de 5 minutos para conseguir saber quantas ações precisaria vender representaria uma perda financeira de até 36 mil reais. Se o *trader* pudesse rapidamente ter acesso a essa quantidade em estoque, ele poderia agir com maior agilidade e conseguir vender todas as suas ações antes dessa queda brusca no preço.

Contudo, o sistema não disponibiliza essa consulta, o que impede o usuário de acessar, de maneira veloz, uma base confiável de seu estoque. Assim, para ter acesso ao estoque de seus produtos financeiros, atualmente o *trader* tem duas alternativas para tentar contornar a ausência dessa funcionalidade do sistema.

A primeira é utilizar um controle próprio através de planilhas em MS Excel®. No entanto, essa solução apresenta a desvantagem de estar sujeita a falhas humanas, uma vez que todos os *inputs* são colocados a mão e não são conferidos por ninguém (diferentemente das operações registradas no sistema, que são conferidas pelo *middle office* e pelo *back office*).

Alternativamente, pode-se também solicitar às equipes do *back office* que enviem as bases extraídas do sistema. A partir da manipulação dessa base, é possível obter os estoques de cada produto de acordo com os critérios que forem desejados. Entretanto, essa segunda solução também apresenta uma desvantagem significativa: a base extraída considera somente as operações que já foram aprovadas tanto pelo *middle office* quanto pelo *back office* até o fechamento do dia útil anterior à data da extração. Assim, as operações realizadas no próprio dia não são levadas em consideração e o estoque indicado pela base pode já não representar mais o estoque em um dado momento.

Percebe-se, pois, que, por um lado, a primeira opção oferece dados do estoque atualizados até o último momento, mas não inteiramente confiáveis devido à possibilidade de falhas humanas. Ao mesmo tempo, a segunda opção oferece dados confiáveis uma vez que já foram conferidos com os dados enviados pela contraparte, mas não há garantia de que estejam atualizados, visto que não levam em conta as operações realizadas no próprio dia.

O que se observa, portanto, é a existência de um sistema inflexível, incapaz de gerar relatórios a partir da base de dados da qual já dispõe. Esta deficiência representa um obstáculo à agilidade do *trader*, prejudicando o seu dia-a-dia e podendo indiretamente ocasionar perdas financeiras consideráveis, como no exemplo recém-citado.

### 3.4. PROPOSTA DE SOLUÇÃO

Uma vez que o sistema já dispõe de uma base contendo todas as operações, incluindo todos os detalhes referentes a elas, julga-se altamente oportuno que esse sistema tivesse a capacidade de consolidar tais operações e, com isso, ter condições de oferecer relatórios úteis a seus usuários. O que se planeja, por conseguinte, é o desenvolvimento da qualidade do fluxo de informações propagadas do sistema ao *trader*, objetivo que é destacado na Figura 3.6.

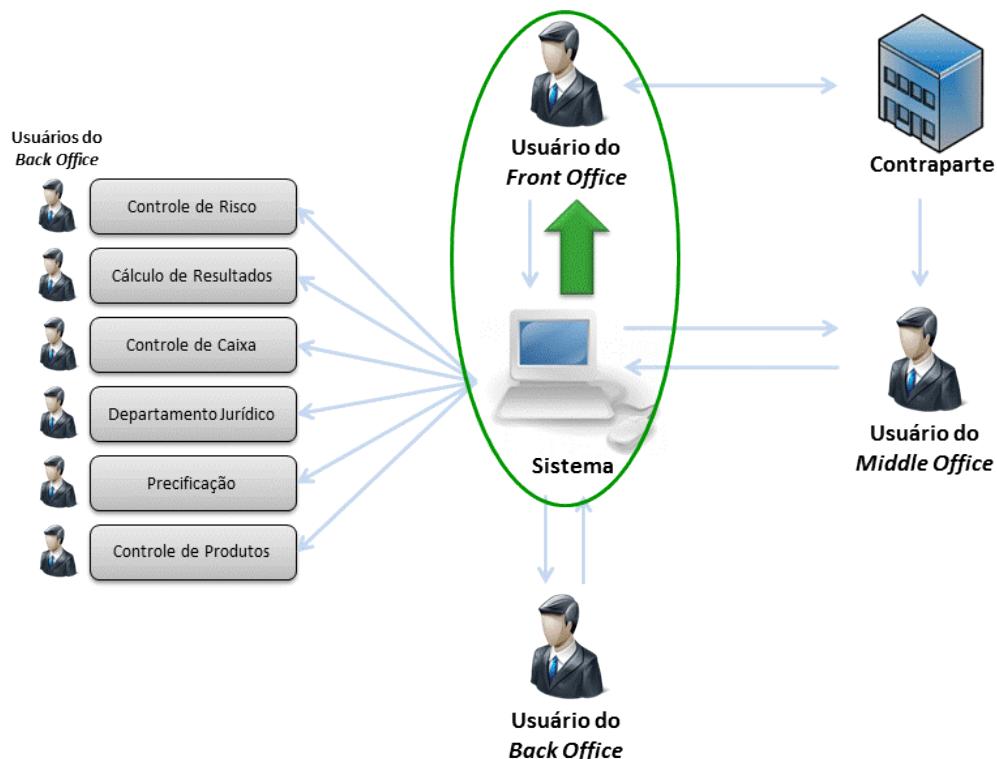


Figura 3.6 – Estrutura dos relacionamentos do sistema com seus usuários, com destaque para a interface a ser analisada e desenvolvida no presente trabalho.

Fonte: Elaborado pelo autor.

Assim, o que se propõe é um sistema versátil, oferecendo um número maior de modalidades de consulta, envolvendo sobretudo a consulta dos estoques. Essa funcionalidade permitirá ao *trader* que ele realize, ao mesmo tempo, uma consulta precisa (vantagem que ele obtinha com as bases extraídas do sistema pelo *back office*) e atualizada (vantagem que ele obtinha com o controle que ele próprio faz via planilha).

Para ilustrar o que se pretende como solução para esta insuficiência do sistema, utilizar-se-á, como exemplo, o controle do estoque dos ativos TYM2 e RXM2. Ressalta-se que não está no escopo deste trabalho aprofundar-se na descrição dos referidos produtos financeiros, em função de a solução proposta ser aplicável para todos os outros ativos. Entretanto, cabe descrevê-los brevemente.

O TYM2 é o contrato futuro que estabelece que, em seu vencimento (29 de junho de 2012), a sua parte compradora comprará da sua parte vendedora um título do governo dos Estados Unidos, com maturidade entre seis anos e meio e dez anos, a um preço acordado na transação do próprio contrato futuro. Estes contratos futuros têm seus preços ajustados de acordo com as expectativas do mercado quanto aos preços dos referidos títulos do governo estadunidense (que é o chamado ativo-objeto do contrato futuro em questão) na data de vencimento do contrato futuro. Já o RXM2 é muito semelhante ao TYM2, com a diferença de que o seu ativo-objeto são títulos do governo alemão com maturidades entre oito anos e meio e 10 anos e meio, e a sua data de vencimento é 11 de junho de 2012.

Um exemplo de base extraída do sistema contendo o conjunto de operações de TYM2 e RXM2 registradas entre 2 de abril de 2012 e 12 de abril de 2012 pode ser visualizado na Figura 3.7. Ressalta-se que somente duas mesas do *Trading* da tesouraria operam tais ativos, a “Trading Internacional” e a “Trading - Credit & Structured Products”, e, portanto, elas são as únicas presentes nesta base apresentada.

DATA_FECHAMENTO	CONTRATO_FUTURO	QTDE_CONTRATOS	PRECO	CONTRAPARTE	NM_MESA_RISCO
02/04/12	RXM2	250	138,04	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
02/04/12	RXM2	-70	138,3	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
02/04/12	RXM2	-180	138,31	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
02/04/12	RXM2	-83	138	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	RXM2	-89	137,97	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	RXM2	-128	137,96	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	RXM2	-200	137,95	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	TYM2	-500	129,375	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	TYM2	-100	129,71875	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
02/04/12	TYM2	-100	129,875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
02/04/12	TYM2	-100	129,90625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
02/04/12	TYM2	200	129,71875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
02/04/12	TYM2	-212	129,703125	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
02/04/12	TYM2	-288	129,71875	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
03/04/12	TYM2	-100	129,90625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	-100	130,03125	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	-100	129,859375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	-50	129,96875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	-50	130,015625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	-200	130	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	300	129,484375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	200	129,890625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
03/04/12	TYM2	100	129,421875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
04/04/12	RXM2	-50	138,62	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	250	138,47	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-50	138,55	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-100	138,5	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-300	138,6	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
04/04/12	RXM2	300	138,79	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
04/04/12	TYM2	250	129,078125	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
04/04/12	TYM2	-200	129,375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
04/04/12	RXM2	100	138,62	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	100	138,65	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-100	138,72	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-100	138,71	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-100	138,65	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-100	138,61	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	RXM2	-50	138,6	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
04/04/12	TYM2	-250	129,265625	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
04/04/12	TYM2	-50	129,453125	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
04/04/12	TYM2	-50	129,40625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
04/04/12	TYM2	-100	129,28125	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	-100	129,90625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	-25	129,875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	-25	129,90625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	RXM2	-100	139,12	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	-100	139,21	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	100	139,19	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	-100	139,13	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	-100	139,12	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	100	139,24	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	100	139,26	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	100	139,28	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	100	139,28	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	99	139,26	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	1	139,26	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
05/04/12	RXM2	135	139,29	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
05/04/12	RXM2	29	139,28	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
05/04/12	RXM2	86	139,27	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
05/04/12	TYM2	500	129,96875	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
05/04/12	TYM2	-25	129,828125	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	50	129,71875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	-100	129,984375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	50	129,9375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
05/04/12	TYM2	350	129,921875	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
06/04/12	TYM2	50	129,90625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
06/04/12	TYM2	50	129,859375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
06/04/12	TYM2	-25	130,96875	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
06/04/12	TYM2	50	129,859375	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
09/04/12	TYM2	-944	131,140625	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
09/04/12	TYM2	-56	131,125	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
09/04/12	TYM2	-25	131,140625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
10/04/12	RXM2	200	139,69	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
10/04/12	RXM2	-200	139,49	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
10/04/12	TYM2	-350	130,96875	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
10/04/12	TYM2	200	131,40625	CITIGROUP GLOBAL MARKETS INC.	TRADING INTERNACIONAL
10/04/12	TYM2	-421	131,609375	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
10/04/12	TYM2	421	131,625	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
12/04/12	TYM2	-423	131,390625	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
12/04/12	TYM2	-577	131,375	JP MORGAN SECURITIES LLC	Trading - Credit & Structured Products
12/04/12	RXM2	60	139,73	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
12/04/12	RXM2	190	139,74	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
12/04/12	RXM2	-250	139,76	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
12/04/12	TYM2	500	131,25	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL
12/04/12	TYM2	-500	131,265625	GOLDMAN SACHS EXECUTION E CLEARING LP	TRADING INTERNACIONAL

Figura 3.7 – Base do sistema da tesouraria do Banco Silva contendo operações de RXM2 e TYM2.

Fonte: Extraído do sistema do Banco Silva.

Todas essas informações fazem parte da base de dados contida no sistema. O passo seguinte, objetivando a consolidação de todas as operações para o fornecimento de relatórios de estoque ao usuário, diz respeito às maneiras de se realizarem tais consolidações. Pode-se decidir por algumas combinações diferentes de agrupamentos de atributos, como por exemplo:

- i) Para o mesmo ativo, para a mesma mesa, fornecer a quantidade e o preço médio para cada contraparte. Este agrupamento é ilustrado na Figura 3.8.

CONTRATO_FUTURO	NM_MESA_RISCO	CONTRAPARTE	QTDE_CONTRATOS	PRECO_MEDIO
RXM2	Trading - Credit & Structured Products	JP MORGAN SECURITIES LLC	50	140,614
RXM2	TRADING INTERNACIONAL	GOLDMAN SACHS EXECUTION E CLEARING LP	-250	137,42528
TYM2	Trading - Credit & Structured Products	JP MORGAN SECURITIES LLC	-2100	131,3586012
TYM2	TRADING INTERNACIONAL	CITIGROUP GLOBAL MARKETS INC.	-275	129,2017045
TYM2	TRADING INTERNACIONAL	GOLDMAN SACHS EXECUTION E CLEARING LP	-500	129,22775

Figura 3.8 – Primeiro exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva.

Fonte: Elaborado pelo autor a partir de base extraída do sistema do Banco Silva.

- ii) Para a mesma contraparte, fornecer a quantidade e o preço médio para cada ativo. Este agrupamento é ilustrado na Figura 3.9.

CONTRAPARTE	CONTRATO_FUTURO	QTDE_CONTRATOS	PRECO_MEDIO
CITIGROUP GLOBAL MARKETS INC.	TYM2	-275	129,2017045
GOLDMAN SACHS EXECUTION E CLEARING LP	RXM2	-250	137,42528
GOLDMAN SACHS EXECUTION E CLEARING LP	TYM2	-500	129,22775
JP MORGAN SECURITIES LLC	RXM2	50	140,614
JP MORGAN SECURITIES LLC	TYM2	-2100	131,3586012

Figura 3.9 – Segundo exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva.

Fonte: Elaborado pelo autor a partir de base extraída do sistema do Banco Silva.

- iii) Para determinado ativo, para determinada mesa, fornecer a quantidade e o preço médio para cada data. Este agrupamento é ilustrado na Figura 3.10.

CONTRATO_FUTURO	NM_MESA_RISCO	DATA_FECHAMENTO	QTDE_CONTRATOS	PRECO_MEDIO
TYM2	Trading - Credit & Structured Products	02/04/12	-100	129,71875
TYM2	Trading - Credit & Structured Products	05/04/12	350	129,921875
TYM2	Trading - Credit & Structured Products	09/04/12	-1000	131,13975
TYM2	Trading - Credit & Structured Products	10/04/12	-350	130,9499554
TYM2	Trading - Credit & Structured Products	12/04/12	-1000	131,3816094

Figura 3.10 – Terceiro exemplo de agrupamento de atributos a partir de base extraída do sistema da tesouraria do Banco Silva.

Fonte: Elaborado pelo autor a partir de base extraída do sistema do Banco Silva.

É evidente que diversas outras combinações de agrupamento são possíveis e estas podem variar de acordo com o interesse do usuário. Cada agrupamento fornecerá a ele uma informação que lhe pode ser útil para determinadas finalidades.

O agrupamento considerado em i) permite que o *trader* saiba com quais contrapartes ele ainda tem posições abertas (não zeradas) caso queira fechá-las. É possível que, considerando-se todas as contrapartes, ele não tenha mais nenhuma exposição em determinado ativo, mas possua posições contrárias em duas contrapartes diferentes (estando comprado em uma e vendido na mesma quantidade em outra). Assim, tal agrupamento garantirá que ele possa zerar as suas posições com cada uma das contrapartes.

O caso do agrupamento exposto em ii) pode ser interessante ao usuário na medida em que possibilita que se comparem os preços médios históricos dos *trades* com cada contraparte. Assim, oferece-lhe a oportunidade de se analisar a qualidade do serviço prestado por cada uma, podendo servir de base para decisões futuras no momento de optar por uma dentre elas.

Finalmente, o modelo de agrupamento exemplificado em iii) agrega valor pelo fato de possibilitar ao *trader* que ele visualize o histórico, dia a dia, das suas operações de determinado produto financeiro. Dessa forma, ele é capaz de analisar com mais embasamento a evolução do mercado, bem como as suas próprias visões e as suas atitudes ao longo do tempo, ajudando-o a refletir e a buscar aperfeiçoamentos.

Percebe-se, pois, toda a importância da viabilização de um sistema que permita a consulta de estoques e ainda ofereça diversas possibilidades de agrupamento dos atributos, de acordo com a preferência do usuário. Essa base de consulta rápida e confiável ajudará muito o *trader* em termos de agilidade, fator crucial para o sucesso da profissão em questão.

Dessa forma, uma vez que o problema está bem delineado e uma proposta de solução está definida, já se possuem as bases necessárias para que se prossiga à definição dos requisitos do sistema sugerido e à modelagem desta solução.

## 4 DEFINIÇÃO DE REQUISITOS

Após a análise da situação atual e tendo-se definido o objetivo deste trabalho, é apresentada, neste capítulo, a definição dos requisitos do sistema proposto. Dessa forma, são expostos os requisitos funcionais, através dos casos de uso, com seus respectivos diagramas, assim como os requisitos não funcionais e os requisitos de interface. Cada um destes três tipos de requisito é detalhado a seguir.

### 4.1. REQUISITOS FUNCIONAIS

Apresentam-se, a seguir, os requisitos funcionais do sistema proposto, que correspondem aos casos de uso referentes ao modelo. Eles foram identificados após entrevistas com os seus usuários e a partir da experiência própria do autor deste trabalho, que também é um usuário.

Ressalta-se que, visando à inteligibilidade dos casos de uso expostos, definem-se os seus atores apenas como usuários genéricos. A distinção das atividades de cada tipo de usuário existente é realizada no capítulo seguinte, no qual detalha-se cada uma das classes.

#### 4.1.1. *Realizar o login*

Este caso de uso representa a ação do usuário do *front office*, *middle office* ou *back office* realizar o seu *login*, identificando-se com suas credenciais para obter autorização de acesso ao sistema. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.1 e na Figura 4.2.

#### Diagrama de caso de uso:

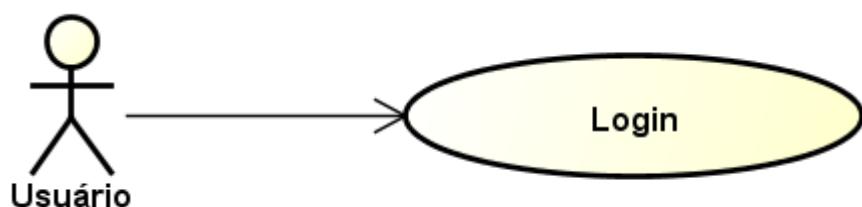


Figura 4.1 – Diagrama de caso de uso “Realizar o *login*”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

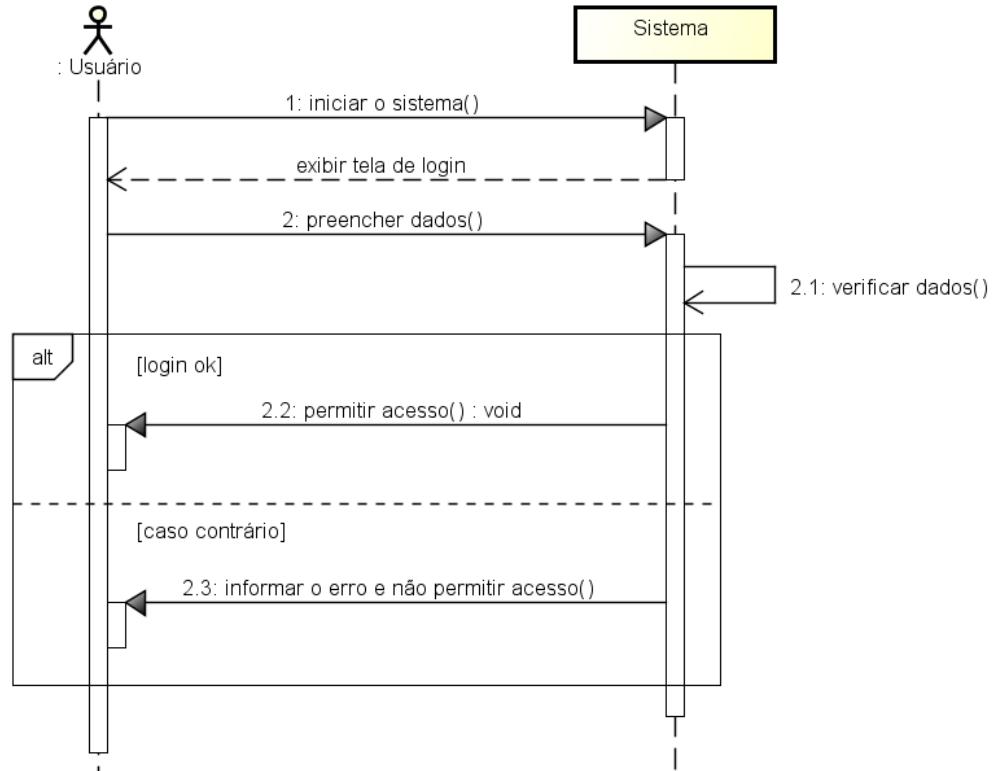


Figura 4.2 – Diagrama de sequência referente ao caso de uso “Realizar o *login*”.  
Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário inicia o sistema.
- 2) O sistema exibe a tela de *login*.
- 3) O usuário preenche os dados de *login* (nome de usuário e senha).
- 4) O sistema verifica se o nome de usuário e a senha estão corretos.
- 5) Caso os dados estejam corretos, o sistema garante acesso ao usuário. Caso contrário, exibe uma mensagem de erro para que o usuário tente o acesso novamente ou desista de realizar o *login*.
- 6) O caso de uso é encerrado.

#### 4.1.2. Boletar operação

Este caso de uso representa a ação do usuário do *front office* boletar uma operação, ou seja, inserir um novo *trade* no sistema. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.3 e na Figura 4.4.

### Diagrama de caso de uso:

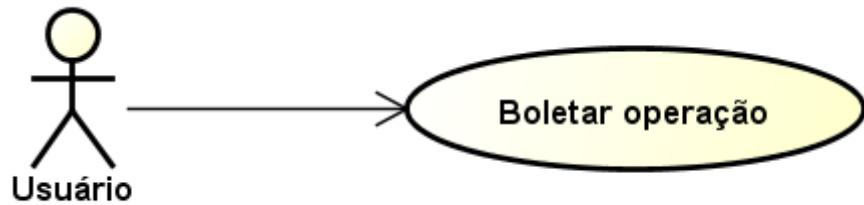


Figura 4.3 – Diagrama de caso de uso “Boletar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

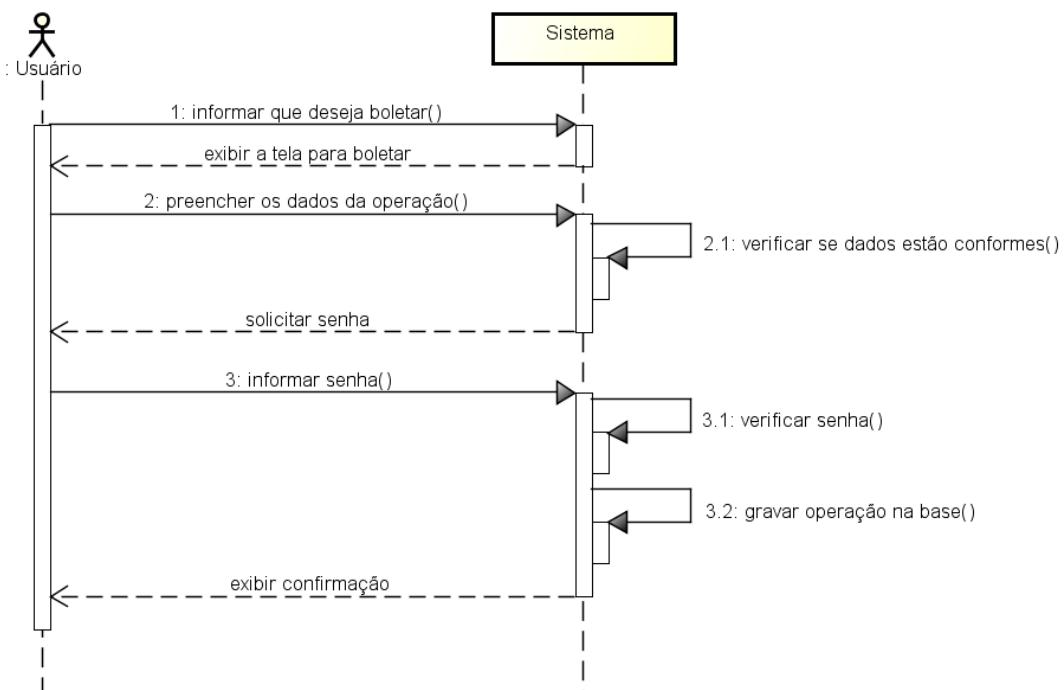


Figura 4.4 – Diagrama de sequência referente ao caso de uso “Boletar operação”.  
Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário seleciona no sistema a opção de incluir uma nova operação.
- 2) O sistema exibe ao usuário a tela para a inserção de novas operações.
- 3) O usuário preenche todos os dados necessários referentes à operação que quer boletar no sistema.
- 4) O usuário indica ao sistema que deseja registrar a operação.
- 5) O sistema verifica se todos os campos obrigatórios foram preenchidos, e se estes contêm informações cujos formatos são os desejáveis.

- 6) O sistema solicita ao usuário que confirme a sua senha de acesso.
- 7) O usuário informa a sua senha de acesso ao sistema.
- 8) O sistema verifica se a senha do usuário está correta.
- 9) Caso a senha esteja correta, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro ao usuário para que ele tente o acesso novamente ou cancele o processo.
- 10) O sistema salva a operação em sua base.
- 11) O sistema exibe ao usuário a confirmação e o número da boleta referente à operação boletada.
- 12) O caso de uso é encerrado.

#### 4.1.3. Pesquisar operação

Este caso de uso representa a ação do usuário do *front office*, *middle office* ou *back office* pesquisar uma operação, através de determinados critérios, na base de *trades* que já existem no sistema. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.5 e na Figura 4.6.

##### Diagrama de caso de uso:

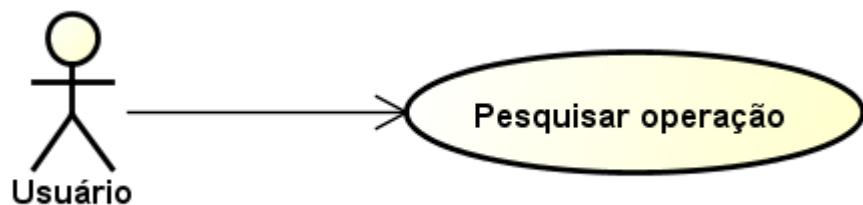


Figura 4.5 – Diagrama de caso de uso “Pesquisar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

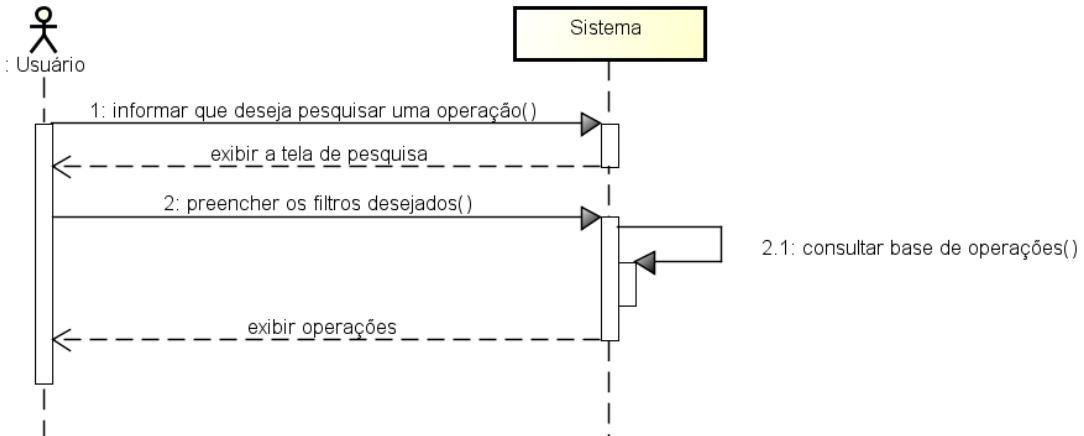


Figura 4.6 – Diagrama de sequência referente ao caso de uso “Pesquisar operação”.

Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- O usuário seleciona no sistema a opção de pesquisar uma operação.
- O sistema oferece ao usuário todos os filtros que podem ser utilizados na pesquisa, como data da operação, equipe, estratégia dentro da equipe, número da boleta, contraparte e status dela (aprovada pelo usuário do *middle office* ou pendente de aprovação).
- O usuário define os filtros de sua escolha.
- O sistema exibe ao usuário a lista de operações que respeitem as condições impostas nos filtros selecionados.
- O caso de uso é encerrado.

#### 4.1.4. Consultar operação

Este caso de uso representa a ação do usuário do *front office*, *middle office* ou *back office* consultar uma operação, ou seja, visualizar os detalhes de um *trade* que já existe no sistema. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.7 e na Figura 4.8.

### Diagrama de caso de uso:



Figura 4.7 – Diagrama de caso de uso “Consultar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

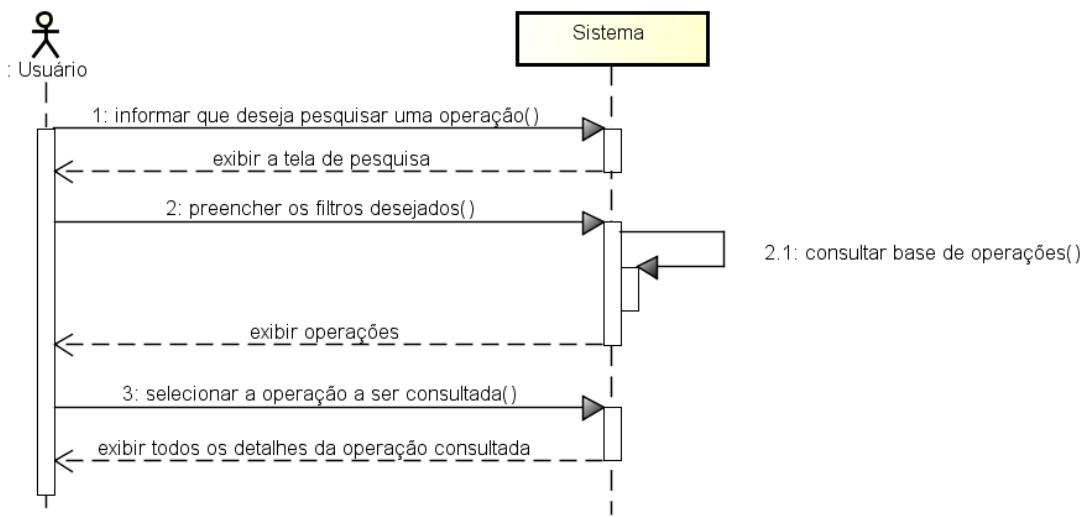


Figura 4.8 – Diagrama de sequência referente ao caso de uso “Consultar operação”.  
Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário realiza a pesquisa de operações no sistema.
- 2) O usuário seleciona no sistema a operação que deseja consultar.
- 3) O sistema exibe ao usuário todas as informações disponíveis referentes à operação escolhida.
- 4) O caso de uso é encerrado.

#### 4.1.5. Alterar operação

Este caso de uso representa a ação do usuário do *front office* alterar uma operação, ou seja, modificar um *trade* que já existe no sistema. O diagrama de caso de uso e o diagrama de

sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.9 e na Figura 4.10.

### Diagrama de caso de uso:



Figura 4.9 – Diagrama de caso de uso “Alterar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

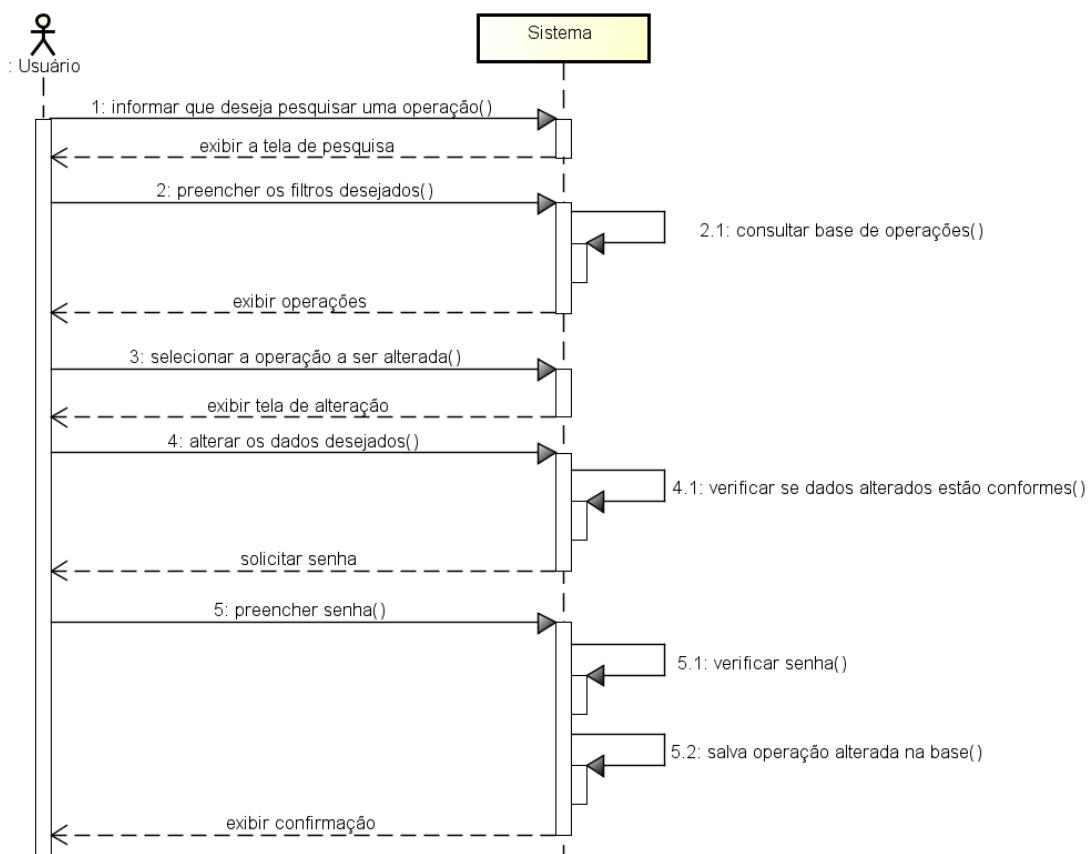


Figura 4.10 – Diagrama de sequência referente ao caso de uso “Alterar operação”.  
Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário realiza a pesquisa de operações no sistema.
- 2) O usuário seleciona no sistema a operação que deseja alterar.
- 3) O sistema exibe ao usuário a tela para alteração de operações.

- 4) O usuário altera todos os dados que desejar referentes à operação escolhida.
- 5) O usuário indica ao sistema que deseja registrar as alterações da operação.
- 6) O sistema verifica se todos os campos obrigatórios foram preenchidos, e se estes contêm informações cujos formatos são os desejáveis.
- 7) O sistema solicita ao usuário que confirme a sua senha de acesso.
- 8) O usuário informa a sua senha de acesso ao sistema.
- 9) O sistema verifica se a senha do usuário está correta.
- 10) Caso a senha esteja correta, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro ao usuário para que ele tente o acesso novamente ou cancele o processo.
- 11) O sistema salva a operação, já com os dados que foram alterados, em sua base.
- 12) O sistema exibe ao usuário o número da boleta referente à operação alterada.
- 13) O caso de uso é encerrado.

#### 4.1.6. Cancelar operação

Este caso de uso representa a ação do usuário do *front office* cancelar uma operação, ou seja, anular um *trade* que já existe no sistema. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.11 e na Figura 4.12.

##### Diagrama de caso de uso:



Figura 4.11 – Diagrama de caso de uso “Cancelar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

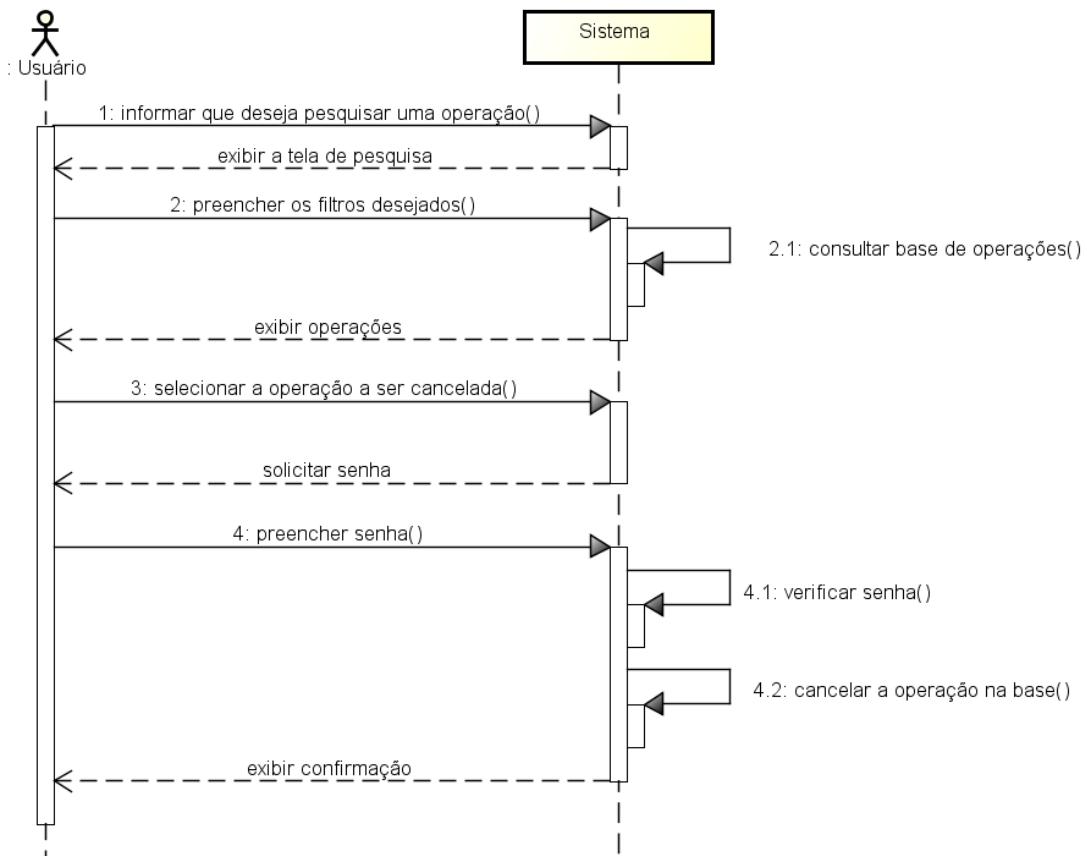


Figura 4.12 – Diagrama de sequência referente ao caso de uso “Cancelar operação”.

Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário realiza a pesquisa de operações no sistema.
- 2) O usuário seleciona no sistema a operação que deseja cancelar.
- 3) O usuário indica ao sistema que deseja cancelar a operação.
- 4) O sistema solicita ao usuário que confirme a sua senha de acesso.
- 5) O usuário informa a sua senha de acesso ao sistema.
- 6) O sistema verifica se a senha do usuário está correta.
- 7) Caso a senha esteja correta, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro ao usuário para que ele tente o acesso novamente ou cancele o processo.
- 8) O sistema cancela a operação e exibe ao usuário a confirmação de tal cancelamento.
- 9) O caso de uso é encerrado.

#### 4.1.7. Cadastrar ativo

Este caso de uso representa a ação do usuário do *middle office* cadastrar um novo ativo no sistema, permitindo que *trades* envolvendo tal ativo possam ser boletados no sistema e reconhecidos por ele. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.13 e na Figura 4.14.

#### Diagrama de caso de uso:

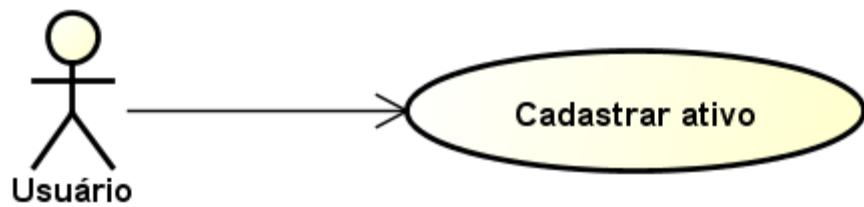


Figura 4.13 – Diagrama de caso de uso “Cadastrar ativo”.  
Fonte: Elaborado pelo autor.

#### Diagrama de sequência:

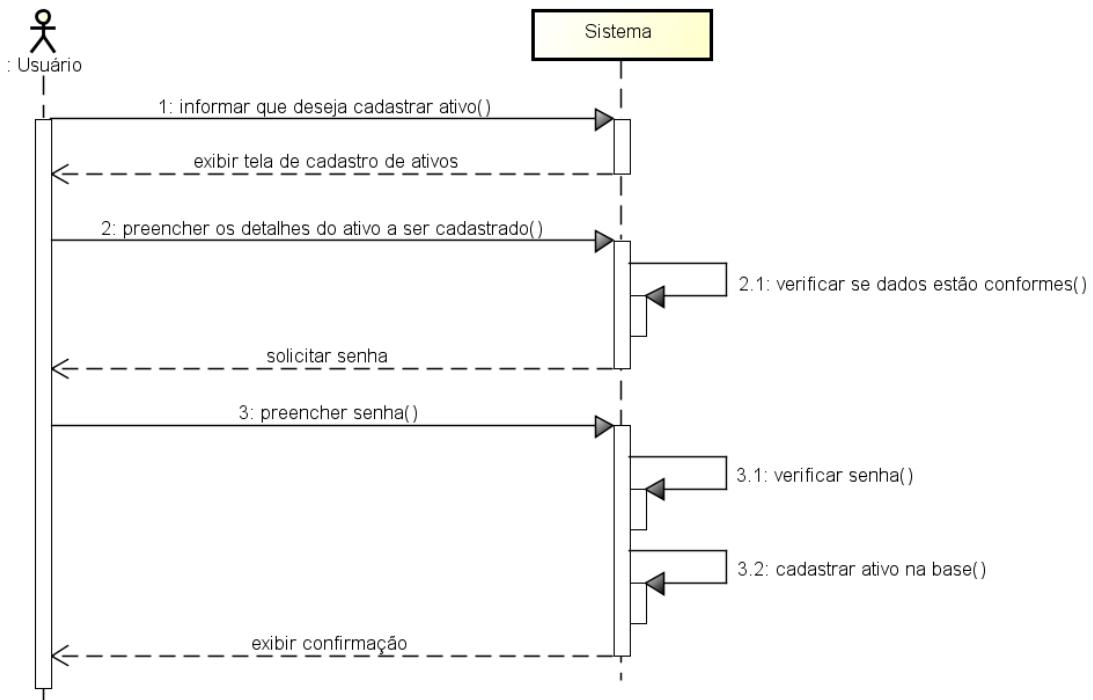


Figura 4.14 – Diagrama de sequência referente ao caso de uso “Cadastrar ativo”.  
Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário indica ao sistema que deseja cadastrar um novo ativo.
- 2) O sistema exibe ao usuário a tela de cadastro de ativos a ser preenchida pelo usuário.
- 3) O usuário preenche todos os detalhes referentes ao ativo que deseja cadastrar e confirma que quer concluir o procedimento.
- 4) O sistema verifica que não há dados incorretos ou incompletos.
- 5) Caso os dados estejam conformes, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro, indicando ao usuário onde está o problema, para que ele tente o cadastro novamente ou cancele o processo.
- 6) O sistema solicita ao usuário que confirme a sua senha de acesso.
- 7) O usuário informa a sua senha de acesso ao sistema.
- 8) O sistema verifica se a senha do usuário está correta.
- 9) Caso a senha esteja correta, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro ao usuário para que ele tente o acesso novamente ou cancele o processo.
- 10) O sistema comunica ao usuário que o novo ativo foi cadastrado com sucesso.
- 11) O caso de uso é encerrado.

#### 4.1.8. Aprovar operação

Este caso de uso representa a ação do usuário do *middle office* aprovar um *trade* boletado no sistema, o que ocorre se é bem sucedida a conferência dos detalhes da operação com os enviados pela contraparte. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.15 e na Figura 4.16.

### Diagrama de caso de uso:



Figura 4.15 – Diagrama de caso de uso “Aprovar operação”.  
Fonte: Elaborado pelo autor.

### Diagrama de sequência:

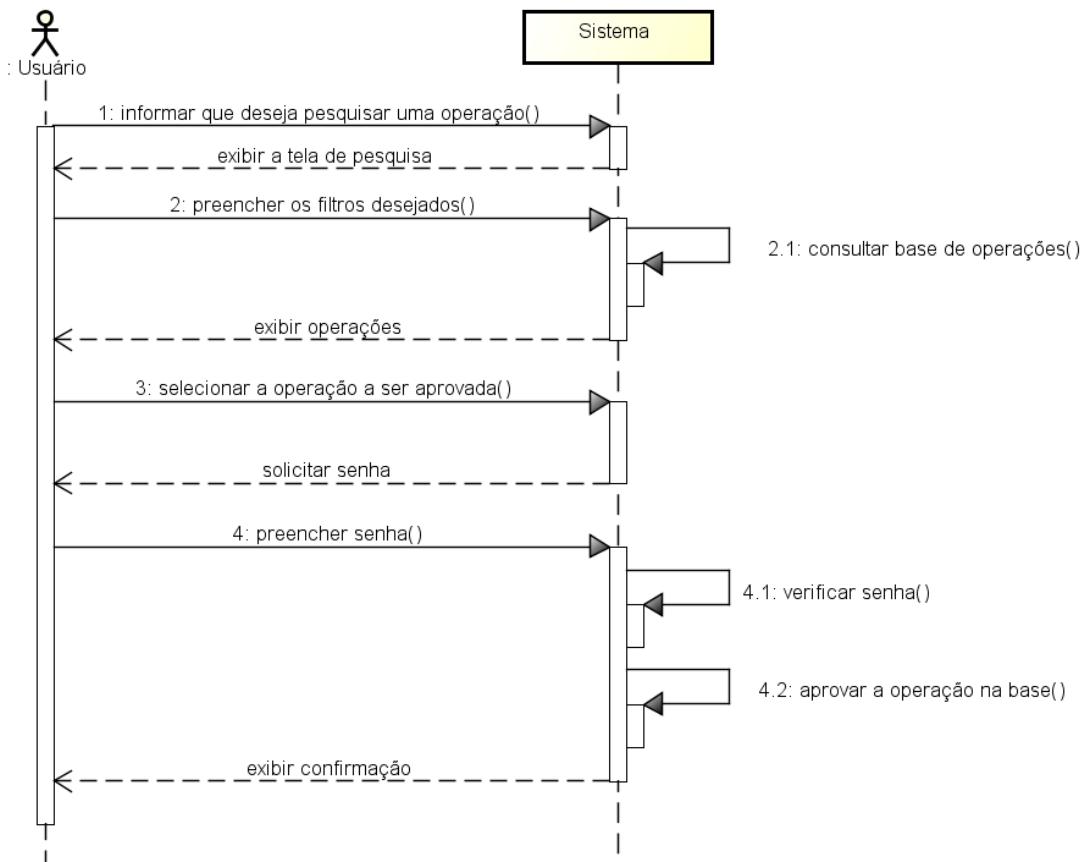


Figura 4.16 – Diagrama de sequência referente ao caso de uso “Aprovar operação”.

Fonte: Elaborado pelo autor.

### Fluxo de eventos:

- 1) O usuário realiza a pesquisa de operações no sistema.
- 2) O usuário seleciona no sistema a operação que deseja aprovar.
- 3) O usuário indica ao sistema que deseja aprovar a operação.
- 4) O sistema solicita ao usuário que confirme a sua senha de acesso.
- 5) O usuário informa a sua senha de acesso ao sistema.
- 6) O sistema verifica se a senha do usuário está correta.
- 7) Caso a senha esteja correta, o caso de uso prossegue. Caso contrário, o sistema exibe uma mensagem de erro ao usuário para que ele tente o acesso novamente ou cancele o processo.
- 8) O sistema altera o status da operação para aprovada e exibe ao usuário a confirmação de tal aprovação.
- 9) O caso de uso é encerrado.

#### 4.1.9. Consultar posição

Este caso de uso representa a ação do usuário do *front office* consultar a sua correspondente posição a partir de determinados critérios. O sistema é capaz de fornecer essa informação a partir da consolidação de todas as operações que estejam relacionadas com os filtros estabelecidos na consulta. O diagrama de caso de uso e o diagrama de sequência correspondentes podem ser visualizados, respectivamente, na Figura 4.17 e na Figura 4.18.

#### Diagrama de caso de uso:

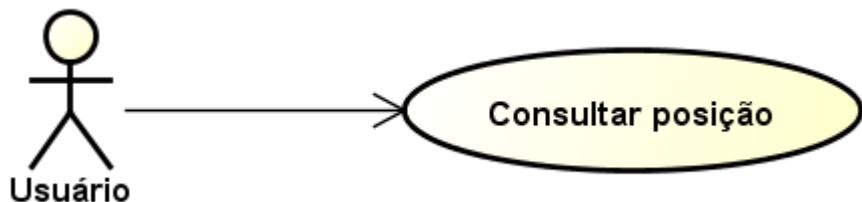


Figura 4.17 – Diagrama de caso de uso “Consultar posição”.  
Fonte: Elaborado pelo autor.

#### Diagrama de sequência:

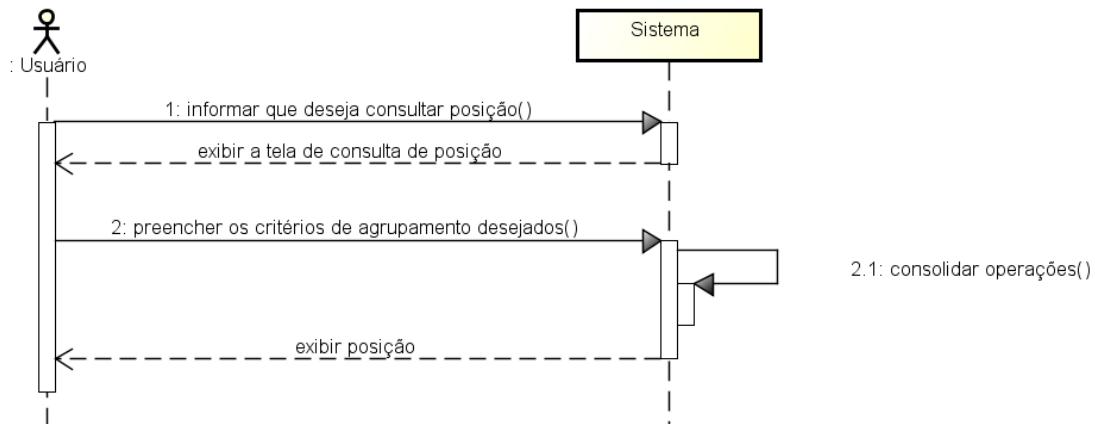


Figura 4.18 – Diagrama de sequência referente ao caso de uso “Consultar posição”.  
Fonte: Elaborado pelo autor.

#### Fluxo de eventos:

- 1) O usuário seleciona no sistema a opção de consultar a sua posição.
- 2) O sistema exibe ao usuário a tela de consulta de posição, que lhe oferece as diversas modalidades de agrupamento existentes.

- 3) O usuário define os seus critérios de agrupamento para a consulta de posição.
- 4) O sistema exibe ao usuário a sua posição consolidada de acordo com os critérios escolhidos.
- 5) O caso de uso é encerrado.

## 4.2. REQUISITOS NÃO FUNCIONAIS

Uma vez que, através da apresentação dos diversos casos de uso, já se definem os requisitos funcionais, são apresentados a seguir os requisitos não funcionais. Como se detalha no Capítulo 2 e se ilustra na Figura 2.2, estes são divididos em três tipos de requisitos: de produtos, organizacionais e externos.

### 4.2.1. *Requisitos de produtos*

#### Facilidade de uso

É fundamental que o sistema seja acessado e utilizado de maneira simples e intuitiva ao usuário. Com isso, reduzem-se as dúvidas e, consequentemente, os erros. Além disso, o sistema é utilizado por muitos estagiários e *trainees*, cargos para os quais observa-se grande rotação de funcionários e, assim, há a constante necessidade de instruir novos usuários cujas tarefas envolvam o sistema. Portanto, garantir a sua facilidade de uso implica usuários familiarizando-se mais rápida e confortavelmente com o sistema. Podem ser citadas como medidas para quantificar tal requisito o número de telas de ajuda e o tempo necessário de aprendizagem.

#### Rapidez de desempenho

Característica sempre ressaltada ao longo do presente trabalho, a agilidade é crucial no desempenho do *trader*. Como se mostrou, em intervalos mínimos de tempo resultados financeiros podem se alterar representativamente no universo do mercado financeiro. Assim, um sistema veloz é capaz de agregar valor ao usuário na medida em que lhe permitirá ser mais reativo em todas as suas outras atividades. Considera-se que um tempo aceitável que se leva para, por exemplo, realizar a consulta a um estoque seja de três segundos.

### Confiabilidade

Da mesma forma que uma grande demora pode ocasionar danos financeiros ao *trader*, informações erradas causam o mesmo efeito. Desta forma, é primordial que o sistema tenha a capacidade de organizar os dados com qualidade para que possa exibir ao usuário informações precisas. Um sistema em que não se pode confiar plenamente tem o seu papel seriamente prejudicado, já que sempre haverá a necessidade de novas consultas por outros meios para que se tente confirmar as informações obtidas através dos sistema.

#### *4.2.2. Requisitos organizacionais*

##### Prazo

Recomenda-se que o sistema esteja funcional o mais breve possível, uma vez que a sua existência facilitará diversas atividades do dia-a-dia do *trader*. No entanto, há de se ter cautela para não se precipitar e colocá-lo em funcionamento antes que todas as simulações necessárias tenham sido realizadas.

##### Implementação

Todos os testes de funcionamento devem preceder a implantação do sistema. Após certificar-se de que o sistema cumpre os demais requisitos descritos, este poderá ser implementado. Ressalta-se a importância de não permitir que os dois sistemas, o anterior e o mais recente, sejam acessíveis concomitantemente, uma vez que isto poderia gerar ambiguidades e/ou dados que sejam cadastrados em somente um deles.

##### Padronização

Julga-se importante que o sistema forneça relatórios em formatos padronizados. Com isso, o usuário poderá comparar mais facilmente diversas consultas que ele realizar. Além disso, a padronização também permite maior clareza na exposição de seus dados, reduzindo as possibilidades de haver interpretações errôneas por parte do usuário.

#### 4.2.3. Requisitos externos

##### Interoperabilidade

Deve-se haver a possibilidade de exportar os diversos relatórios gerados pelo sistema para o MS Excel®. Esta interoperabilidade é útil pois permite ao usuário que ele trabalhe os dados e as informações extraídos do sistema através de manipulações via planilha Excel, cada vez da maneira que melhor lhe convier.

##### Privacidade

Cada usuário do sistema deve possuir um nome de usuário e uma senha próprios, que não devem ser compartilhados. Estes dados pessoais devem estar ligados a um tipo de perfil que seja particular a cada usuário, estando relacionado à equipe na qual trabalha e às suas funções exercidas.

##### Segurança

O sistema não deve permitir o acesso de pessoas não autorizadas, uma vez que há inúmeras informações confidenciais contidas nele. Além disso, é importante eliminar as possibilidades de que usuários inapropriados realizem alterações indevidas no sistema, colocando em risco o seu bom funcionamento.

### 4.3. REQUISITOS DE INTERFACE

Como explica Pádua (2009), na interface gráfica com o usuário há questões que representam requisitos do produto, como o formato de dados e comandos. No entanto, outros detalhes, como formatos de telas e janelas, são aspectos que se enquadram na categoria de requisitos de interface, terceiro e último dentre os tipos de requisito apresentados neste trabalho.

Um requisito de interface que deve ser considerado é a existência de um *layout* simples e amigável. Uma tela muito carregada, com muitas informações, pode confundir o usuário ou até mesmo desestimulá-lo a usar o sistema. Deste modo, é necessário que a disposição dos campos e dos botões seja muito bem planejada de modo a garantir que o *layout* seja amigável e que, ao mesmo tempo, todas as funcionalidades continuem sendo oferecidas

ao usuário. Para cumprir tal objetivo, conduziram-se entrevistas com os usuários, além da experiência própria do autor, que também é um usuário (com perfil de *front office*) do sistema.

O que se propõe, pois, são três *layouts* diferentes de tela inicial, um para cada tipo de perfil de usuário: usuário do *back office*, usuário do *middle office* e usuário do *front office*. Tal distinção é útil uma vez que cada um desses tipos de usuário realiza tarefas diferentes no sistema.

Primeiramente, logo ao iniciar o sistema, é solicitado identicamente aos três usuários que eles realizem o *login*, como é ilustrado na Figura 4.19.

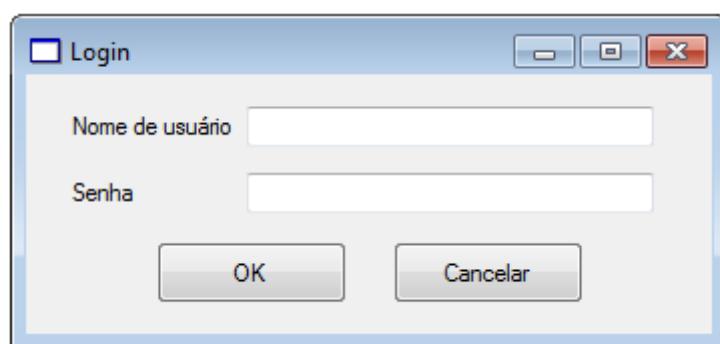


Figura 4.19 – Layout da tela de *login* do usuário.  
Fonte: Elaborado pelo autor.

Em seguida, o sistema identificará a qual tipo de perfil está relacionado o usuário que está tentando o acesso, para que este possa ser direcionado à tela inicial que contenha as tarefas que lhe correspondam. Para os usuários do *front office*, do *middle office* e do *back office*, o *layout* da tela inicial pode ser visualizado, respectivamente, na Figura 4.20, Figura 4.21 e Figura 4.22.

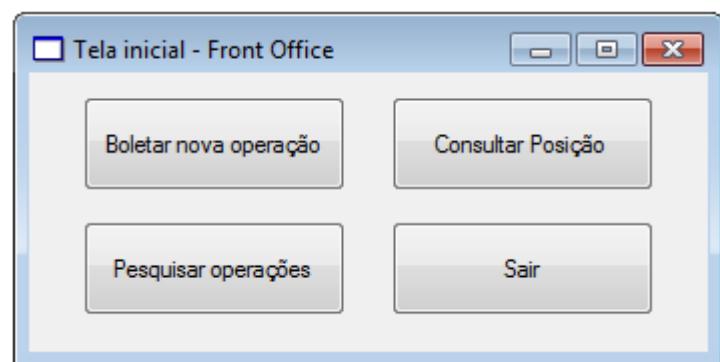


Figura 4.20 – Layout da tela inicial correspondente ao usuário do *front office*.  
Fonte: Elaborado pelo autor.

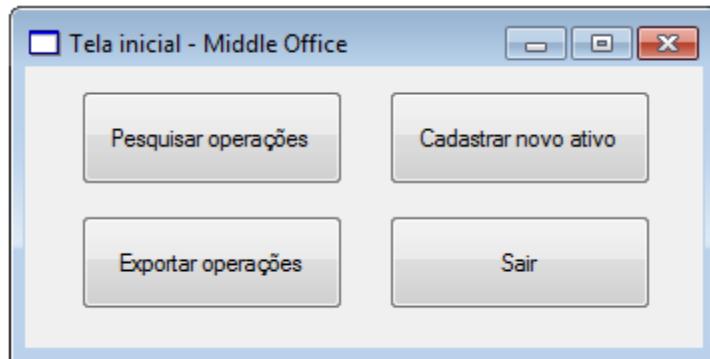


Figura 4.21 – *Layout* da tela inicial correspondente ao usuário do *middle office*.

Fonte: Elaborado pelo autor.

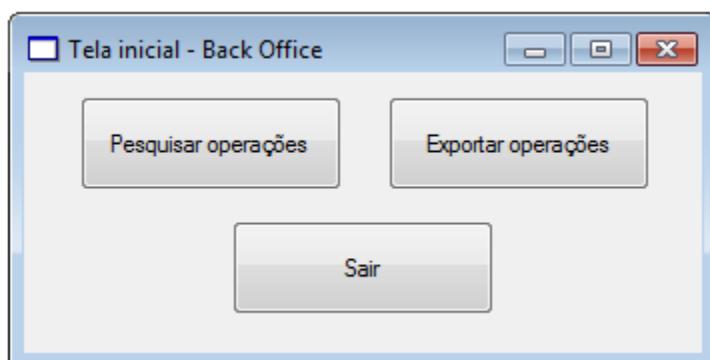


Figura 4.22 – *Layout* da tela inicial correspondente ao usuário do *back office*.

Fonte: Elaborado pelo autor.

Percebe-se que, nesta tela inicial, cada tipo de usuário já visualizará clara e facilmente quais são as atividades que ele pode realizar no sistema. Após isso, uma vez apresentadas as primeiras telas, são apresentados os esboços julgados relevantes dos *layouts* das telas seguintes às telas iniciais.

- i) Pode-se visualizar a boletagem de uma nova operação na Figura 4.23. Nesta tela, primeiramente, o usuário escolhe a que tipo de ativo se refere a operação que ele deseja boletar no sistema. Uma vez realizado tal procedimento, os campos obrigatórios para o tipo de ativo escolhido aparecerão para que o usuário os preencha. No caso de um contrato futuro, que é o caso exemplificado, o usuário fornece o nome deste ativo e clica em “Validar” para que o sistema o reconheça. Em seguida, preenche os dados básicos da operação, como quantidade, preço, contraparte, data da operação, mesa e estratégia correspondente dentro da mesa.

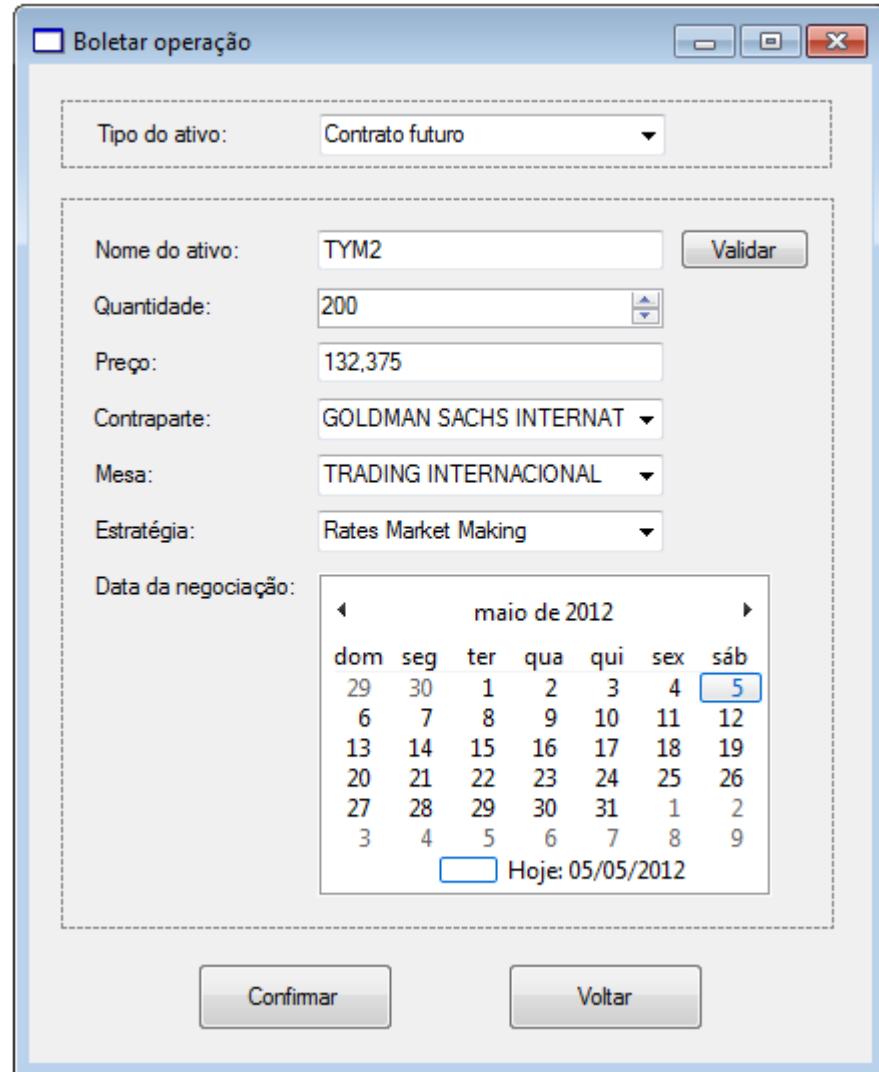


Figura 4.23 – *Layout* da tela para boletar uma nova operação.

Fonte: Elaborado pelo autor.

- ii) A tela para se realizar a pesquisa de operações é visualizada na Figura 4.24. Neste *layout*, o usuário preenche os critérios que deseja estabelecer para a sua consulta. Ele opta também por um dos dois operadores lógicos, E ou OU, para definir se a operação procurada deve atender, respectivamente, a todos os critérios impostos ou somente a um deles.

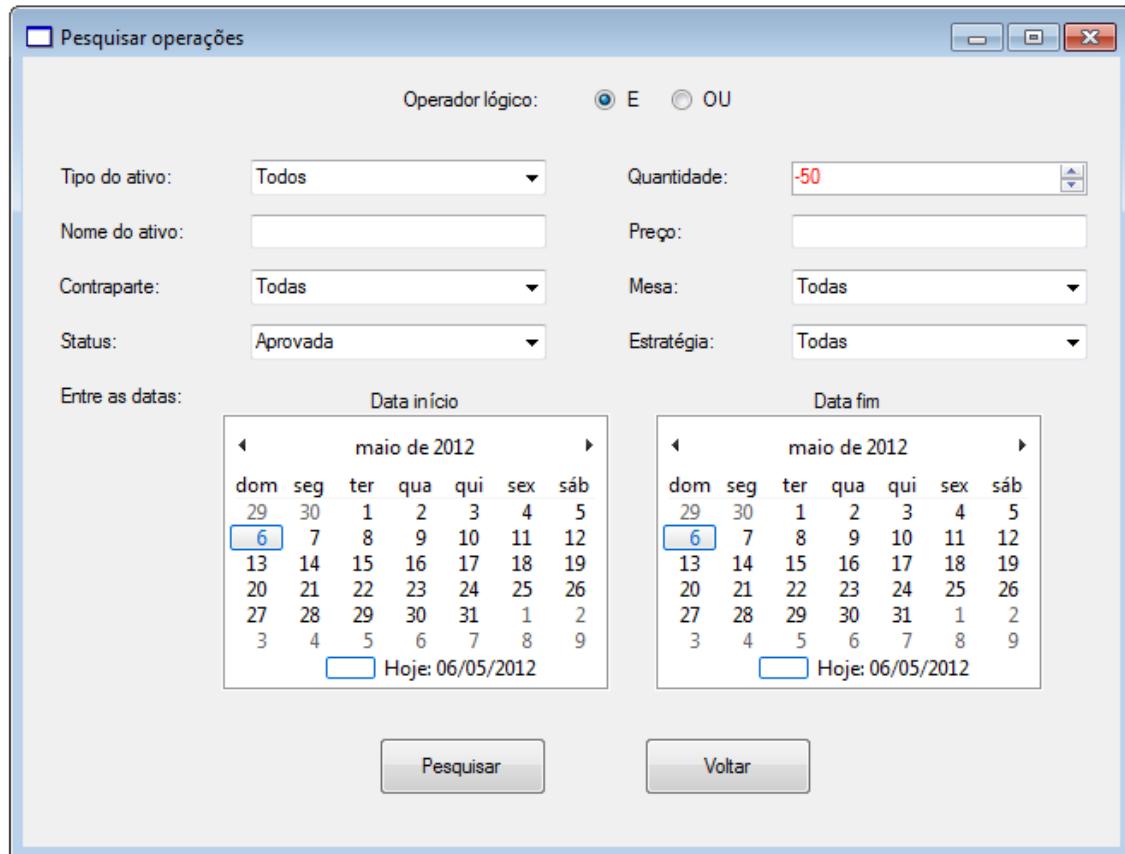


Figura 4.24 – *Layout* da tela para pesquisar operações.

Fonte: Elaborado pelo autor.

- iii) Na Figura 4.25, é mostrada a tela de consulta do estoque a partir de diversos critérios de agrupamento (como alguns apresentados na seção 3.4 do presente trabalho). O usuário escolhe quais são os critérios que serão agrupados, e em qual ordem, e quais são os dados que devem ser fornecidos como *outputs* para os critérios definidos. O usuário deve fornecer também os valores para os critérios que foram definidos, dentre os quais a data de referência. Esta, que pode não necessariamente corresponder ao dia no qual se realiza a pesquisa, é importante uma vez que o relatório gerado representará o estoque consultado para esta data em questão.

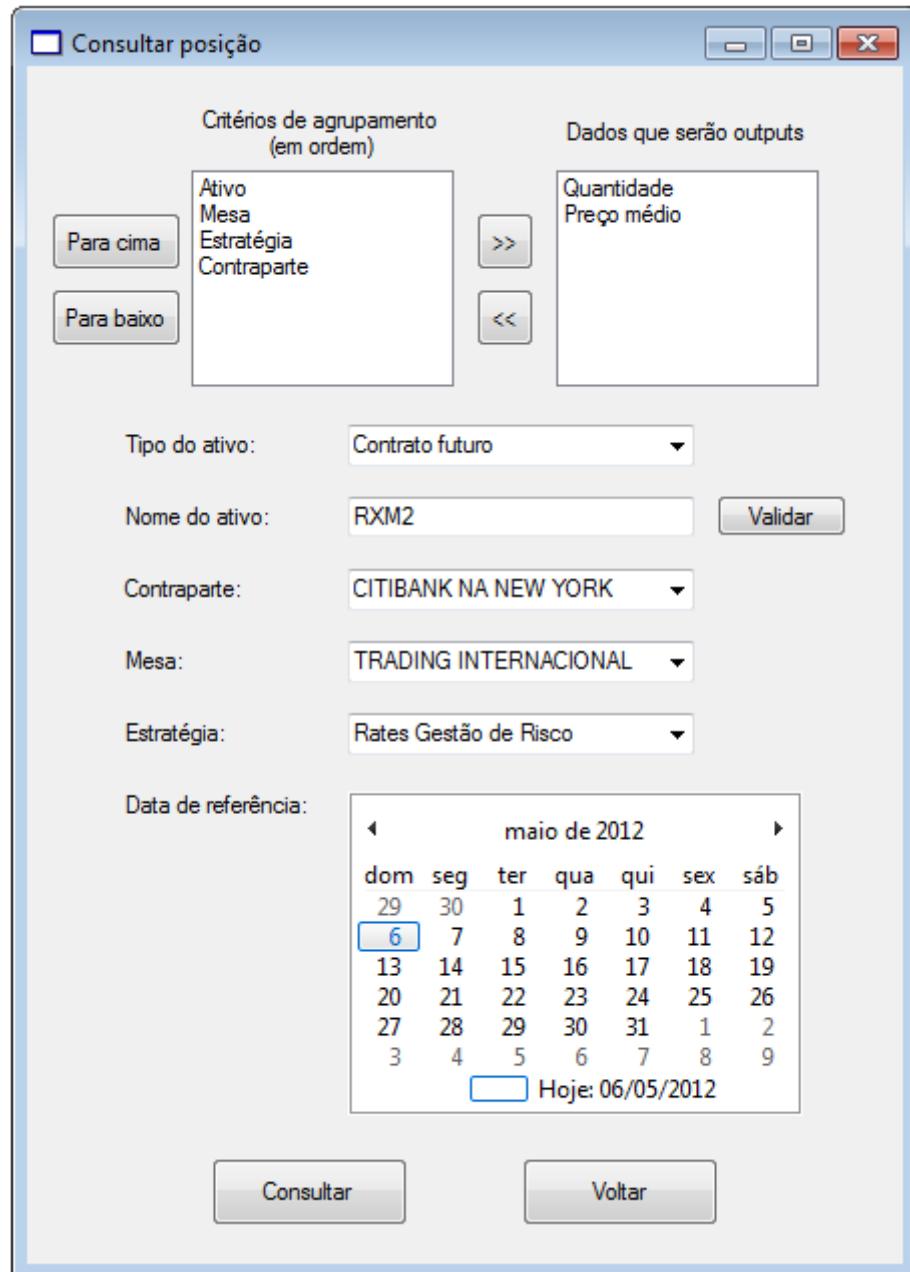


Figura 4.25 – Layout da tela para consultar a posição.

Fonte: Elaborado pelo autor.

Ressalta-se que os *layouts* expostos devem ser considerados apenas esboços, não tendo o objetivo de restringir quaisquer outras formas de se construir a interface do sistema.

## 5 MODELAGEM DE CLASSES

Após a definição dos requisitos para o sistema proposto, este capítulo visa dar continuidade à aplicação das ferramentas da UML ao contexto estudado do sistema do Banco Silva.

A modelagem orientada a objetos traz diversas vantagens no que diz respeito à visualização do contexto estudado. No entanto, quem realiza a análise através deste modelo encontra uma dificuldade significativa: a definição das classes. Identificá-las corretamente e completamente, bem como os seus atributos e métodos, representa um processo trabalhoso e que, se mal realizado, pode comprometer todo o andamento futuro do projeto.

Neste trabalho, optou-se pela técnica da identificação dirigida por casos de uso, segundo a qual o conjunto de casos de uso representa as responsabilidades dos diversos objetos que devem colaborar para produzir os resultados especificados. Assim, através da análise individual de cada caso de uso, identificam-se os substantivos para definir quais são os objetos participantes e, logo, candidatos a serem classes. Então, ao final deste processo, pode-se conhecer quais as classes que são consideradas no modelo referente à análise realizada.

Após tal processo, as classes definidas, que são detalhadas mais adiante<sup>4</sup>, foram as seguintes:

- Usuário;
- Usuário do *front office*;
- Usuário do *middle office*;
- Usuário do *back office*;
- Produto financeiro;
- Par de moedas;
- Opção;
- Contrato futuro;
- Operação;
- Operação de moeda;
- Operação de opção;
- Operação de contrato futuro;

---

<sup>4</sup> É oportuno ressaltar que se opta, nos diagramas UML deste trabalho, por não utilizar nenhuma acentuação gráfica no nome das classes, bem como no de seus atributos e métodos.

- Contraparte;
- Mesa;
- Estratégia.

Percebe-se, entre as classes recém-indicadas, a existência de algumas generalizações, que são detalhadas a seguir.

### 5.1. GENERALIZAÇÕES

Este tipo de relação entre classes envolve a identificação de uma superclasse que encapsule a estrutura e o comportamento comuns a mais de uma subclasse. Estas subclasses herdam, portanto, os atributos e os métodos da superclasse.

Assim, a partir das classes recém-apresentadas, os seguintes casos de generalização são observados:

#### Operação

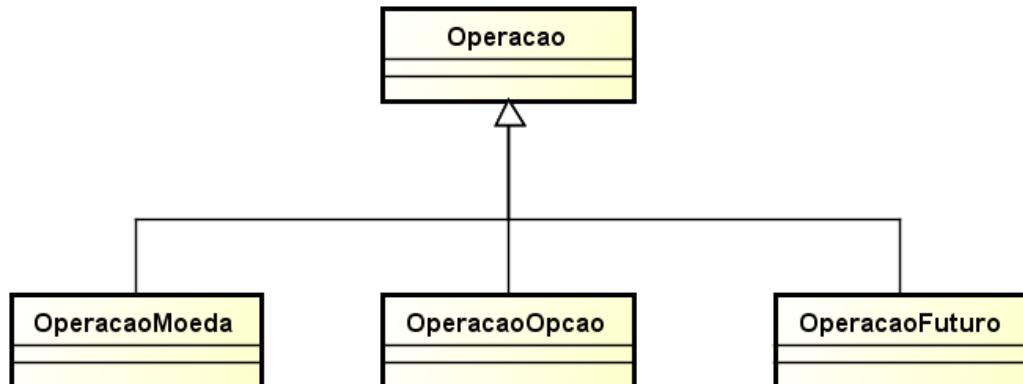


Figura 5.1 – Generalização das classes Operação.  
Fonte: Elaborado pelo autor.

As três subclasses OperacaoMoeda, OperacaoOpcão e OperacaoFuturo possuem diversos atributos e métodos comuns e, portanto, podem representar um caso de herança de uma superclasse, como representado na Figura 5.1. As únicas diferenças são as relacionadas ao produto financeiro que é objeto de cada uma destas operações. Os atributos e métodos de cada uma das classes em questão são apresentados a seguir, na seção 5.2.

### Produto financeiro

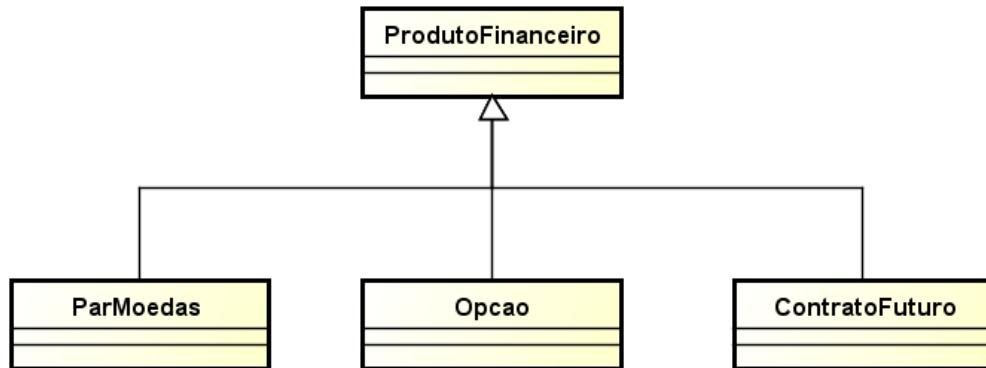


Figura 5.2 – Generalização das classes Produto Financeiro.  
Fonte: Elaborado pelo autor.

As três subclasses ParMoedas, Opcão e ContratoFuturo possuem atributos comuns e, portanto, podem representar um caso de herança de uma superclasse, como representado na Figura 5.2. As diferenças são, evidentemente, as especificidades de cada um destes três produtos financeiros. Os atributos e métodos de cada uma destas classes são apresentados a seguir, na seção 5.2.

### Usuário

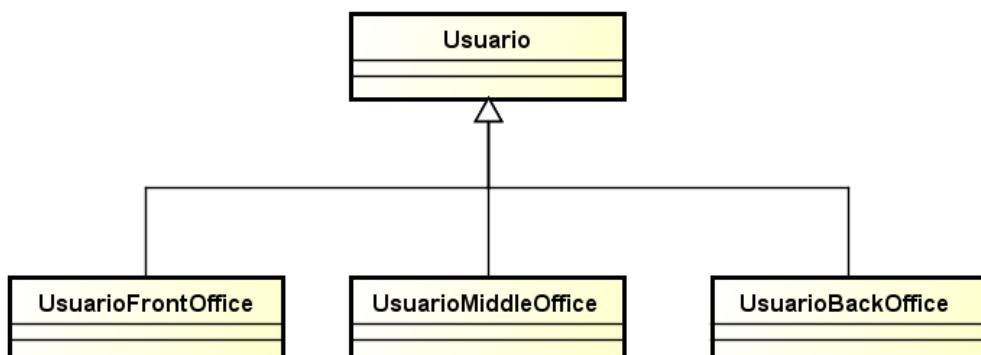


Figura 5.3 – Generalização das classes Usuário.  
Fonte: Elaborado pelo autor.

As três subclasses UsuarioFrontOffice, UsuarioMiddleOffice e UsuarioBackOffice possuem diversos atributos e métodos comuns e, portanto, podem representar um caso de herança de uma superclasse, como representado na Figura 5.3. As principais diferenças referem-se às atividades de cada uma dessas três áreas (*front office*, *middle office* e *back*

*office*), que se refletem diretamente no tipo de acesso permitido a cada uma no sistema. Os atributos e métodos destas classes são apresentados a seguir.

## 5.2. DETALHAMENTO DAS CLASSES

Esta seção pretende apresentar e explicar os atributos e métodos de cada uma das classes selecionadas para a análise do problema deste trabalho. Este detalhamento é extremamente importante no prosseguimento da modelagem do sistema.

### 5.2.1. *Usuário*

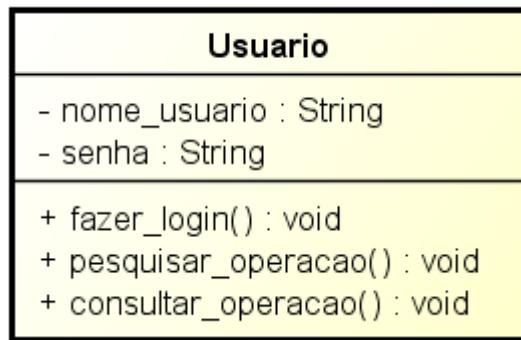


Figura 5.4 – Classe “Usuário”.  
Fonte: Elaborado pelo autor.

#### Atributos

- nome\_usuario: o nome de usuário, único e intransferível, referente à pessoa autorizada a utilizar o sistema.
- senha: a senha escolhida pelo usuário para obter o seu acesso ao sistema.

#### Métodos

- fazer\_login: ação de o usuário efetuar o seu acesso ao sistema.
- pesquisar\_operacao: ação de o usuário realizar, a partir dos critérios desejados, uma pesquisa na base de operações do sistema.
- consultar\_operacao: ação de o usuário acessar todos os detalhes referentes a uma determinada operação.

### 5.2.2. Usuário do *front office*

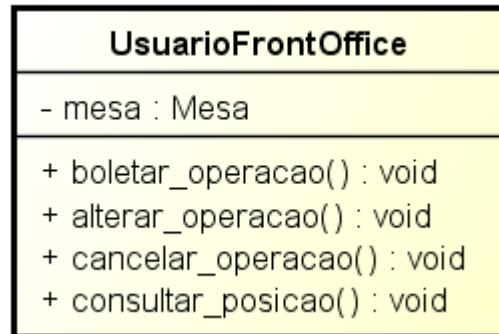


Figura 5.5 – Classe “Usuário do *front office*”.  
Fonte: Elaborado pelo autor.

#### Atributos

- mesa: equipe à qual o usuário do *front office* pertence.

#### Métodos

- boletar\_operacao: ação de o usuário do *front office* inserir uma nova operação no sistema.
- alterar\_operacao: ação de o usuário do *front office* realizar a alteração de uma operação registrada no sistema.
- cancelar\_operacao: ação de o usuário do *front office* realizar a anulação de uma operação registrada no sistema.
- consultar\_posicao: ação de o usuário do *front office* ter acesso, a partir dos critérios de agrupamento que desejar, a dados consolidados referentes às operações registradas no sistema.

### 5.2.3. Usuário do *middle office*

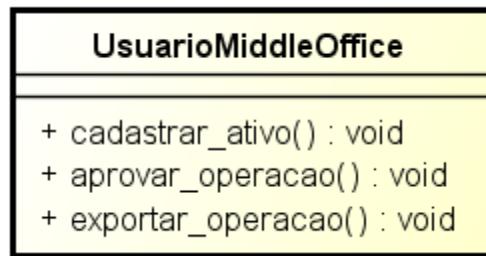


Figura 5.6 – Classe “Usuário do *middle office*”.

Fonte: Elaborado pelo autor.

### Métodos

- *cadastrar\_ativo*: ação de o usuário do *middle office* inserir um novo produto financeiro na base de produtos reconhecidos pelo sistema.
- *aprovar\_operacao*: ação de o usuário do *middle office* efetuar a aprovação de uma operação boletada pelo usuário do *front office* no sistema.
- *exportar\_operacao*: ação de o usuário do *middle office* exportar as operações registradas no sistema para diversos propósitos.

### 5.2.4. Usuário do *back office*

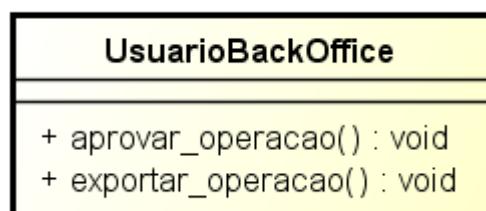


Figura 5.7 – Classe “Usuário do *back office*”.

Fonte: Elaborado pelo autor.

### Métodos

- *aprovar\_operacao*: ação de o usuário do *back office* efetuar a aprovação de uma operação já aprovada pelo usuário do *middle office* no sistema.
- *exportar\_operacao*: ação de o usuário do *back office* exportar as operações registradas no sistema para diversos propósitos.

### 5.2.5. Produto financeiro

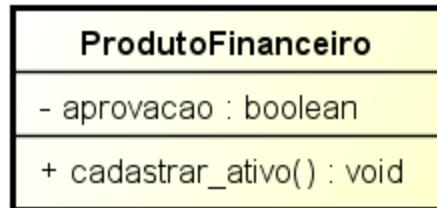


Figura 5.8 – Classe “Produto financeiro”.

Fonte: Elaborado pelo autor.

#### Atributos

- aprovação: atributo que assume apenas dois valores, caracterizando o fato de o objeto desta classe, um produto financeiro qualquer, ser aprovado ou não para ser operado pelo banco.

#### Métodos

- cadastrar\_ativo: ação de um novo produto financeiro ser inserido pelo usuário do *middle office* na base de produtos reconhecidos pelo sistema.

### 5.2.6. Par de moedas

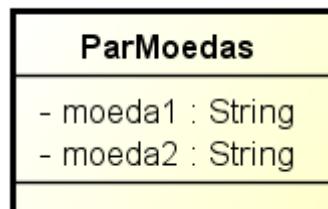


Figura 5.9 – Classe “Par de moedas”.

Fonte: Elaborado pelo autor.

## Atributos

- moeda1: símbolo da unidade monetária da moeda que será operada ao par com a moeda2.<sup>5</sup>
- moeda2: símbolo da unidade monetária da moeda que será operada ao par com a moeda1.<sup>5</sup>

### 5.2.7. *Opção*

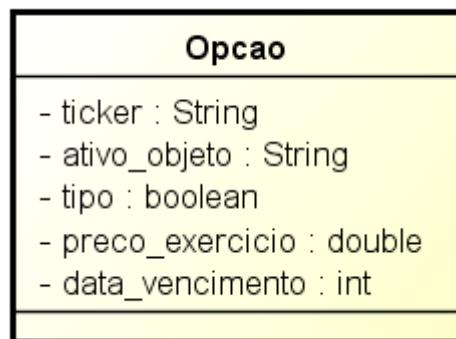


Figura 5.10 – Classe “Opção”.  
Fonte: Elaborado pelo autor.

## Atributos

- ticker: símbolo através do qual é conhecida uma determinada opção.
- ativo\_objeto: ativo que o comprador de uma determinada opção tem o direito de comprar ou vender.
- tipo: atributo que assume apenas dois valores, indicando se a opção é de compra ou de venda do ativo-objeto. A opção de compra é chamada de *call* e a de venda, *put*.
- preco\_exercicio: preço pré-estabelecido que o comprador de uma determinada opção tem direito de pagar ou receber pelo ativo-objeto.
- data\_vencimento: data pré-estabelecida na qual o comprador de uma determinada opção tem o direito de comprar ou vender o ativo-objeto.

---

<sup>5</sup> As operações no mercado de moedas sempre são realizadas envolvendo duas moedas. Para se comprar uma certa quantia de uma primeira moeda, deve-se vender uma outra quantia de uma segunda moeda, sendo esta segunda quantia definida pela taxa cambial entre as duas moedas. Por exemplo, seja a taxa cambial do dólar americano em relação ao real brasileiro igual a 1,95. Assim, utilizando tal taxa, a operação de compra de 100 dólares implica necessariamente a venda de 195 reais.

### 5.2.8. *Contrato futuro*

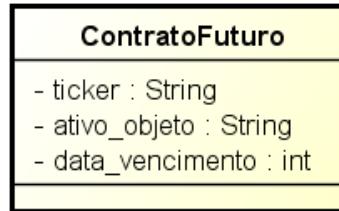


Figura 5.11 – Classe “Contrato futuro”.

Fonte: Elaborado pelo autor.

### Atributos

- ticker: símbolo através do qual é conhecido um determinado contrato futuro.
- ativo\_objeto: ativo que será operado, na data de vencimento, pelo comprador e pelo vendedor de um determinado contrato futuro.
- data\_vencimento: data na qual haverá a operação do ativo-objeto entre o comprador e o vendedor de um determinado contrato futuro.

### 5.2.9. *Operação*

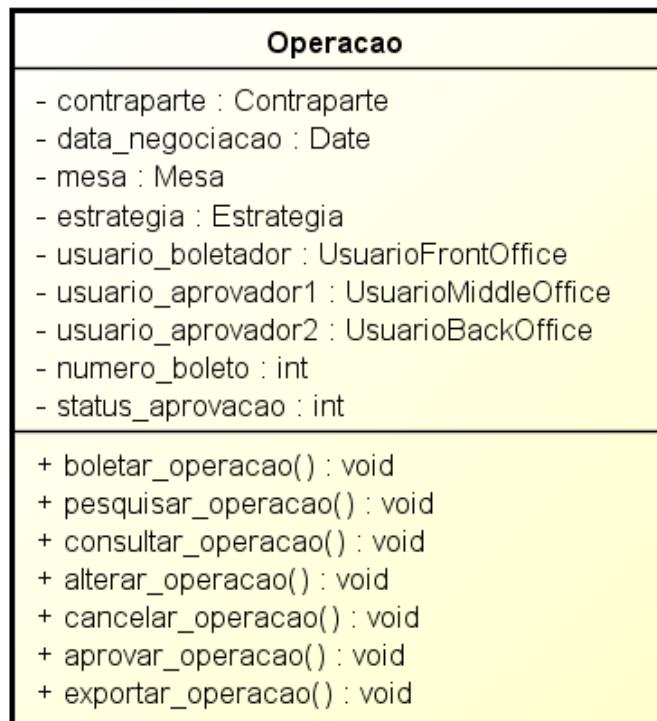


Figura 5.12 – Classe “Operação”.

Fonte: Elaborado pelo autor.

### Atributos

- contraparte: refere-se a um objeto da classe Contraparte, definindo quem é a outra parte, além do Banco Silva, envolvida na operação.
- data\_negociacao: data na qual foi realizada a operação.
- mesa: refere-se a um objeto da classe Mesa, definindo para qual mesa a operação deve ser alocada.
- estrategia: refere-se a um objeto da classe Estrategia, indicando para qual estratégia, dentro da definida mesa, a operação deve ser alocada.
- usuario\_boletador: refere-se a um objeto da classe UsuarioFrontOffice, definindo qual usuário do *front office* inseriu a operação no sistema.
- usuario\_aprovador1: refere-se a um objeto da classe UsuarioMiddleOffice, definindo qual usuário do *middle office* aprovou a operação no sistema.
- usuario\_aprovador2: refere-se a um objeto da classe UsuarioBackOffice, definindo qual usuário do *back office* aprovou a operação no sistema.
- numero\_boleto: número único referente à operação para a sua identificação.
- status\_aprovacao: indica qual o estado em que se encontra a operação registrada no sistema, como, por exemplo, aprovada, cancelada, pendente de aprovação do *middle office* ou pendente de aprovação do *back office*.

### Métodos

- boletar\_operacao: ação de uma nova operação ser inserida no sistema pelo usuário do *front office*.
- pesquisar\_operacao: ação de uma pesquisa ser realizada pelo usuário, na base de operações do sistema, a partir dos critérios desejados.
- consultar\_operacao: ação de uma operação ter todos os seus detalhes visualizados pelo usuário.
- alterar\_operacao: ação de uma operação registrada no sistema ser alterada pelo usuário do *front office*.
- cancelar\_operacao: ação de uma operação registrada no sistema ser anulada pelo usuário do *front office*.

- aprovar\_operacao: ação de uma operação registrada no sistema ser aprovada pelo usuário do *back office* ou do *middle office*.
- exportar\_operacao: ação de um conjunto de operações registradas no sistema ser exportado pelo usuário do *back office* ou do *middle office*.

#### 5.2.10. Operação de moeda

OperacaoMoeda	
- par_moedas : ParMoedas	
- moeda_comprada : String	
- qtd_moeda_comprada : double	
- moeda_vendida : String	
- qtd_moeda_vendida : double	
- taxa : double	
- data_vencimento : Date	

Figura 5.13 – Classe “Operação de moeda”.

Fonte: Elaborado pelo autor.

#### Atributos

- par\_moedas: refere-se a um objeto da classe ParMoedas, definindo quais são as moedas que fazem parte da operação.
- moeda\_comprada: símbolo da moeda que está sendo comprada na operação.
- qtd\_moeda\_comprada: quantidade da moeda que está sendo comprada na operação.
- moeda\_vendida: símbolo da moeda que está sendo vendida na operação.
- qtd\_moeda\_vendida: quantidade da moeda que está sendo vendida na operação.
- taxa: taxa cambial referente ao par de moedas que faz parte da operação.
- data\_vencimento: data na qual há a efetiva compra da moeda que está sendo comprada e da venda da moeda que está sendo vendida.

### 5.2.11. Operação de opção

OperacaoOpcão
<ul style="list-style-type: none"> <li>- opcao : Opcão</li> <li>- quantidade : int</li> <li>- preco : double</li> </ul>

Figura 5.14 – Classe “Operação de opção”.  
Fonte: Elaborado pelo autor.

### Atributos

- opcao: refere-se a um objeto da classe Opcão, definindo qual é a opção que está sendo comprada/vendida na operação.
- quantidade: quantidade de opções que estão sendo compradas/vendidas na operação. Números negativos indicam venda.
- preco: preço pago/recebido pela opção na operação.

### 5.2.12. Operação de contrato futuro

OperacaoFuturo
<ul style="list-style-type: none"> <li>- contrato_futuro : ContratoFuturo</li> <li>- quantidade : int</li> <li>- preco : double</li> </ul>

Figura 5.15 – Classe “Operação de contrato futuro”.  
Fonte: Elaborado pelo autor.

### Atributos

- contrato\_futuro: refere-se a um objeto da classe ContratoFuturo, definindo qual é o contrato futuro que está sendo comprado/vendido na operação.
- quantidade: quantidade de contratos futuros que estão sendo comprados/vendidos na operação. Números negativos indicam venda.
- preco: preço pago/recebido pelo contrato futuro na operação.

### 5.2.13. Contraparte

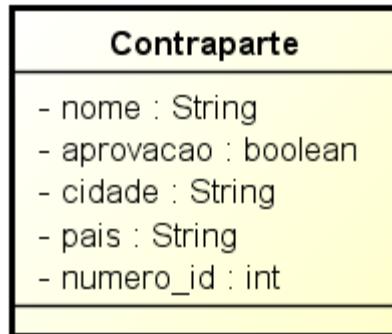


Figura 5.16 – Classe “Contraparte”.  
Fonte: Elaborado pelo autor.

#### Atributos

- nome: nome da contraparte.
- aprovacao: pode assumir dois valores, referentes ao fato de a contraparte estar ou não aprovada para participar de operações com o Banco Silva.
- cidade: nome da cidade em que se encontra a sede da contraparte.
- pais: nome do país em que se encontra a sede da contraparte.
- numero\_id: número único referente à contraparte para a sua identificação.

### 5.2.14. Mesa



Figura 5.17 – Classe “Mesa”.  
Fonte: Elaborado pelo autor.

#### Atributos

- nome: nome da equipe.

- **usuario:** refere-se a objetos da classe `UsuarioFrontOffice`, indicando quais são os usuários do *front office* que fazem parte da mesa.

### 5.2.15. Estratégia



Figura 5.18 – Classe “Estratégia”.

Fonte: Elaborado pelo autor.

### Atributos

- **nome:** nome da estratégia.
- **mesa:** refere-se a um objeto da classe `Mesa`, definindo a qual mesa a estratégia pertence.

## 5.3. DIAGRAMA DE CLASSES

Uma vez que todas as classes consideradas na modelagem foram definidas e seus atributos e métodos foram especificados e detalhados, pode-se, finalmente, construir o diagrama de classes.

Como exposto na fundamentação teórica, no Capítulo 2, o diagrama de classes é de grande importância na medida em que possibilita a ampla compreensão da situação estudada através da observação dos relacionamentos entre as classes. O diagrama construído é ilustrado na Figura 5.19 e, em seguida, apresentam-se informações detalhadas referentes a ele.

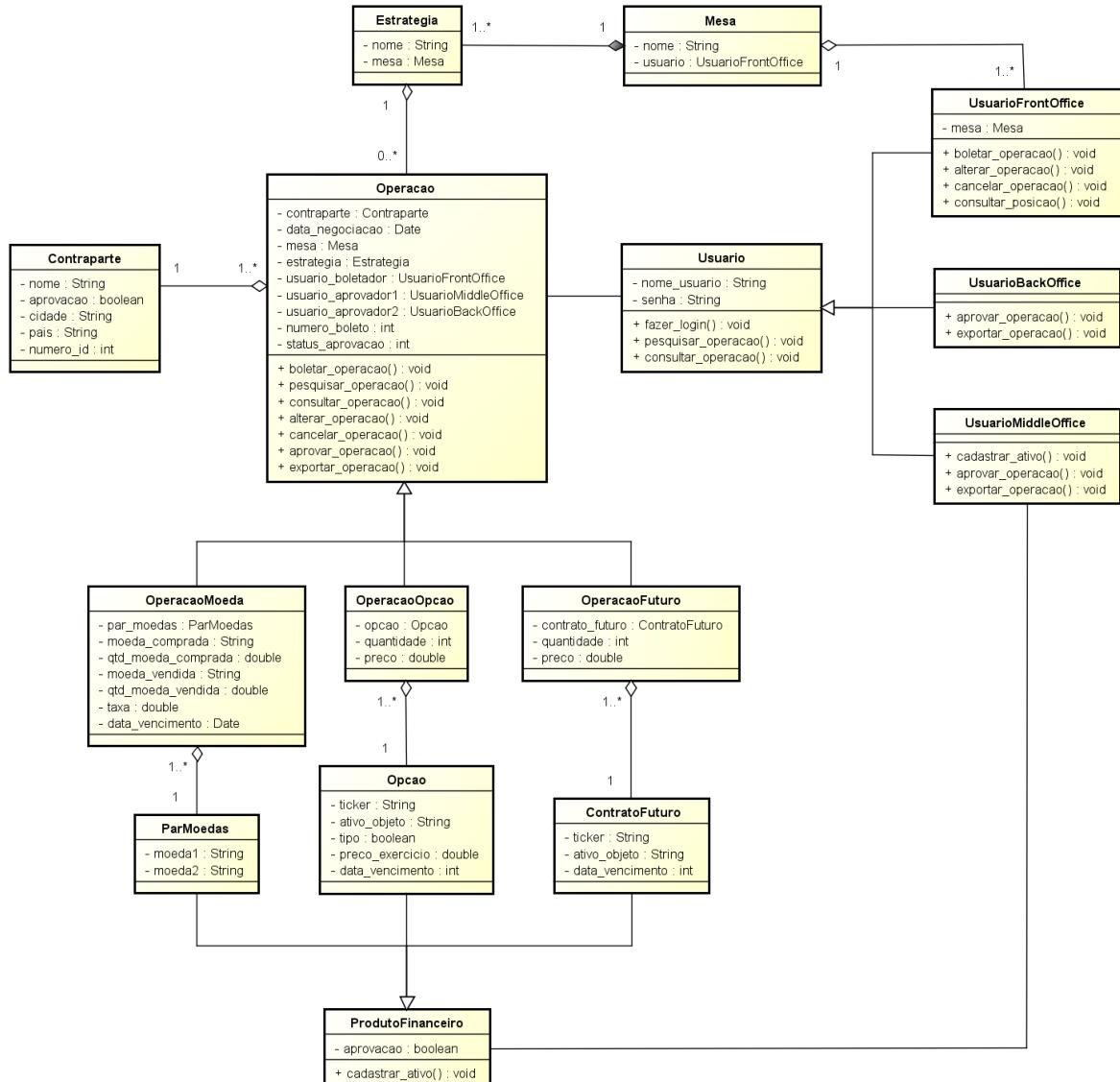


Figura 5.19 – Diagrama de classes.

Fonte: Elaborado pelo autor.

Como discorrido na seção 5.1, observa-se a ocorrência de três casos de generalização, nos quais há, em cada um deles, três subclasses relacionadas a uma superclasse. A seguir, descrevem-se os demais relacionamentos entre as classes presentes no diagrama.

Constata-se a existência de um relacionamento de agregação entre as classes Opcao e OperacaoOpcao. Isso se dá em função de o relacionamento não ser bidirecional e de existir a ideia de constituição: uma opção, objeto da classe Opcao, faz parte de uma operação de opção, objeto da classe OperacaoOpcao. Além disso, não há a noção de existência condicionada, o que caracterizaria tal relacionamento como uma composição. O que se tem é que uma opção continua existindo independentemente da existência da operação de opção. Ressalta-se também a multiplicidade que se emprega no relacionamento em questão: uma

operação de opção não pode ser constituída por mais de uma única opção, enquanto uma opção pode fazer parte de diversas operações de opção.

Dois casos exatamente análogos são observados nos relacionamentos entre as classes ParMoedas e OperacaoMoeda, e ContratoFuturo e OperacaoFuturo, tanto no que diz respeito ao tipo de relacionamento quanto à multiplicidade considerada. Este mesmo relacionamento de agregação pode ser observado em outros três casos no diagrama de classes.

Primeiramente, ele ocorre entre as classes Contraparte e Operacao. Há a noção de que uma contraparte faz parte de uma operação, mas sem ter a sua existência condicionada à desta última. Também se percebe que uma operação é constituída de somente uma contraparte, ao passo que uma contraparte pode constituir inúmeras operações.

Em seguida, há este relacionamento também entre as classes Operacao e Estrategia, isto é, uma operação constitui uma estratégia, porém sem a ideia de existência condicionada. Além disso, uma operação pode constituir somente uma estratégia, enquanto uma estratégia é constituída por diversas operações.

Finalmente, as classes UsuarioFrontOffice e Mesa interagem igualmente através deste mesmo tipo de relacionamento, a agregação. Ela ocorre pois um usuário do *front office* faz parte de uma mesa, e continuará existindo mesmo que tal mesa acabe. Quanto à multiplicidade, uma mesa conta com diversos usuários do *front office*, mas um usuário do *front office* não pode fazer parte de mais de uma mesa.

Observa-se também a existência de um relacionamento de composição, que se dá entre as classes Mesa e Estrategia. Uma mesa é composta por diversas estratégias, e uma estratégia não pode fazer parte de mais de uma mesa. Entretanto, diferentemente dos relacionamentos explicitados anteriormente, neste caso há um vínculo de existência condicionada. Ou seja, caso uma mesa não exista mais, as estratégias que faziam parte dela passam a não existir tampouco, caracterizando a dependência da existência de um objeto da classe Estrategia em relação a um objeto da classe Mesa. Trata-se, pois, de um relacionamento de composição.

No diagrama de classes apresentado, também há relacionamentos entre classes configurados como associações, que são bidirecionais. Um primeiro a ser abordado é o relacionamento entre as classes Usuario e Operacao. Um usuário, objeto da classe Usuario, tem, através de seus métodos, acesso aos atributos de uma operação, objeto da classe Operacao. Por exemplo, o seu método consultar\_operacao lhe permite conhecer todos os

atributos referentes a uma operação. Um usuário também é o responsável por criar novos objetos da classe Operacao, bem como exclui-los ou alterar os seus atributos.

Um segundo relacionamento de associação presente no diagrama de classes é observado entre as classes UsuarioMiddleOffice e ProdutoFinanceiro. Um usuário do *middle office*, objeto da classe UsuarioMiddleOffice, é o responsável por cadastrar novos ativos no sistema, criando, assim, novos objetos da classe ProdutoFinanceiro. Dessa forma, através de seus métodos, o usuário tem acesso aos atributos de um determinado produto financeiro.

## 6 RESULTADOS

Uma vez realizado o delineamento do problema, pôde-se desenvolver a proposta de solução e a sua modelagem, que culminou na construção do diagrama de classes da Figura 5.19. A partir desse estudo, resultados foram obtidos, e estes constituem o tema do qual o presente capítulo trata.

O diagrama de classes permite uma visualização clara e objetiva de como as classes se relacionam entre si, possibilitando que se comprehenda o funcionamento do sistema proposto como um todo.

Neste diagrama, esperava-se e pôde constatar-se que a classe central, em torno da qual todo o sistema funciona, é a *Operacao*. Todos os usuários que têm acesso ao sistema têm métodos relacionados a uma operação, objeto da classe *Operacao*. Tais métodos são diversos, podendo estes representar ações que criem, excluam, alterem, aprovem, pesquisem ou consultem as operações, como especificado nos requisitos funcionais.

Com o Banco Silva adotando o sistema proposto, que conta com a funcionalidade sugerida de gerar relatórios inteligentes a partir de formas distintas de agrupamento de atributos, identificam-se resultados benéficos ao *trader*, que são expostos a seguir.

Observa-se que a inflexibilidade do sistema atual impede que o *trader* tenha a agilidade necessária para reagir diante do dinamismo do mercado financeiro. Assim, com a proposta de permitir que se consultem de maneira simples os estoques no sistema, verifica-se a possibilidade de impacto direto nos resultados financeiros da atividade do *trader*.

Primeiramente, retoma-se o caso relativo ao preço das ações da Petrobras no dia 25 de abril de 2012, ilustrado na seção 3.3. Neste exemplo, após a presidente da empresa realizar um discurso que, teoricamente, não era bom para os negócios da Petrobras, viu-se uma queda de 1,69% no preço da ação em apenas cinco minutos. No momento em que o *trader* tomou conhecimento desse pronunciamento, era possível que ele quisesse se desfazer de todas as ações da Petrobras que possuísse, vendendo-as ao prever a sua desvalorização iminente. Dessa forma, precisaria saber quantas ações necessitaria vender naquele momento.

Possuindo o sistema flexível proposto no presente trabalho, o *trader* poderia rapidamente acessá-lo e entrar na tela de consulta de posição, cujo *layout* é ilustrado na Figura 4.25. Dessa maneira, selecionaria o produto financeiro que deseja e os dados referentes à

equipe dele para saber, de forma confiável, a quantidade exata de ações que precisaria vender, diminuindo o seu prejuízo financeiro frente a essa desvalorização.

O exemplo acima retrata uma queda brusca no preço de um ativo em um espaço curto de tempo, e a necessidade do *trader* de obter uma informação precisa rapidamente. No entanto, pode haver também um caso análogo no qual ocorra a repentina subida de preço do ativo. Neste caso, o *trader* pode querer comprar rapidamente ações da Petrobras para se aproveitar dessa valorização iminente. Contudo, o Banco Silva, assim como todos os bancos, possui limites de exposição, ou seja, um número máximo de ações da Petrobras que se pode possuir, com o objetivo de controlar o risco ao qual está exposto. Dessa forma, o *trader* precisa saber até quantas ações ele pode comprar sem que a soma das que já possua em estoque com as que está prestes a comprar ultrapasse o limite pré-estabelecido. Assim, mais uma vez, o *trader* utilizaria o sistema flexível proposto, obteria rapidamente o número de ações em estoque, saberia até quantas mais poderia comprar e, então, poderia aproveitar-se dessa súbita alta do preço da ação para obter lucros.

Portanto, percebe-se que é de extrema importância o *trader* conseguir ter acesso rapidamente a essas informações do estoque fornecidas pelo sistema proposto. Enquanto no caso da queda brusca do preço da ação, a flexibilidade do sistema possibilita que o *trader* reduza as suas perdas, no caso da subida brusca de preço, essa flexibilidade permite que o *trader* aumente os seus ganhos. Verifica-se, pois, a relação direta existente entre essa flexibilidade e o resultado financeiro das operações realizadas pelo *trader*.

Vale ressaltar que, para as quantias normalmente investidas pelos *traders* da tesouraria do Banco Silva, essas pequenas oscilações percentuais nos preços dos ativos representam quantias muito expressivas. Além disso, considerando-se o fato de que tais comportamentos repentinos ocorrem com os preços de diversos ativos, diversas vezes ao dia, são ainda mais relevantes os benefícios financeiros proporcionados ao *trader* pelo sistema proposto.

Além da consulta do estoque geral de um determinado ativo, que pode impactar os ganhos financeiros do *trader*, como recém-apresentado, o sistema flexível proposto também oferece outras modalidades de consulta. Ao se fazer variar a forma de agrupar os atributos das operações, outras informações muito relevantes podem também ser obtidas.

Na seção 3.4, apresentam-se três exemplos de modalidades possíveis de agrupamento e verifica-se que cada uma delas fornece informações relevantes ao *trader* de acordo com o

que ele queira analisar. Esses estudos são relevantes no dia-a-dia do *trader* uma vez que podem proporcionar-lhe percepções e opiniões capazes de influenciar as suas tomadas de decisão.

Pode-se, por exemplo, analisar qual contraparte (na maioria dos casos, uma corretora) ofereceu, historicamente, preços mais atrativos em operações envolvendo um determinado ativo, significando menores taxas de corretagem cobradas. Sendo a base de operações analisadas suficientemente extensa para englobar um número grande de operações, espera-se que o preço médio destas seja semelhante para todas as contrapartes. Caso o *trader* utilize o sistema flexível proposto e constate que há uma contraparte que pratica melhores preços, ele pode passar a realizar mais operações com essa determinada contraparte. Por outro lado, de maneira análoga, é possível que se note que a média dos preços de operações com uma certa contraparte é comparativamente pior e, portanto, o *trader* pode decidir realizar menos operações com ela.

Um outro estudo que o *trader* pode realizar com o sistema flexível proposto envolve a consulta de suas operações feitas a cada dia, para que analise suas qualidades e suas deficiências a partir dos *trades* que realizou. Ele pode notar comportamentos seus que se repitam constantemente, como, por exemplo, tender sempre a vender um determinado ativo às segundas-feiras e comprá-lo às sextas-feiras. Este padrão, se verificado, pode sugerir que o *trader* tem sido excessivamente pessimista com este ativo às segundas-feiras e otimista às sextas-feiras. Ciente desta possibilidade, o *trader* pode passar a refletir com mais calma, tentando não se deixar influenciar por aspectos emocionais, e rever racionalmente se deseja realmente comprar ou vender o ativo em questão.

Indubitavelmente, várias outras combinações de agrupamento são possíveis. O *trader* pode realizar os estudos que pretender, conforme o seu interesse. Cada agrupamento fornecerá a ele uma informação que lhe pode ser útil para uma determinada análise.

No entanto, vale relembrar que, geralmente, os *trades* que ocorrem na tesouraria do Banco Silva movimentam quantias muito grandes. Portanto, qualquer oportunidade de se conseguir reduzir custos percentuais, por menores que sejam, ou melhorar práticas gerais nas atividades do *Trading*, teria influência direta e representativa nos resultados financeiros da área.

## 7 CONCLUSÃO

Em linhas gerais, este Trabalho de Formatura envolveu a análise do sistema de registro das operações realizadas na tesouraria do Banco Silva, identificando nele pontos a serem melhorados. Propuseram-se funcionalidades que tornassem o sistema mais flexível, agregando valor ao *trader*, usuário deste sistema, em termos de agilidade e reatividade. Estas características têm impacto direto nos seus ganhos financeiros, uma vez que são fatores críticos de sucesso em um ambiente tão dinâmico como o mercado no qual atua a organização estudada.

Para iniciar a análise, realizou-se uma revisão bibliográfica, apresentando os principais conceitos envolvidos e salientando outros estudos que trataram de ganhos de eficiência no setor bancário como consequência de investimentos em sistemas de informação. Em seguida, decidiu-se conduzir uma modelagem orientada a objetos, com auxílio da linguagem UML, útil na definição dos requisitos do sistema, na identificação das classes e na elaboração do diagrama de classes. Também expuseram-se e discutiram-se os benefícios resultantes da utilização do sistema flexível proposto, podendo-se afirmar, pois, que os objetivos definidos no início do trabalho foram plenamente atingidos.

A utilização da linguagem UML oferece fácil compreensão, além de independe de qual linguagem de programação será usada posteriormente. Como o sistema não será implementado pelo autor, é de extrema importância a correta modelagem do problema, uma vez que o futuro desenvolvedor do software provavelmente não estará familiarizado com o funcionamento dos atuais sistemas e com o mercado no qual o estudo está inserido.

Ressalta-se também o grande valor proveniente da comunicação com aqueles que estão envolvidos com o sistema proposto. Cada usuário pode perceber deficiências diferentes no sistema e possuir necessidades distintas em relação a ele. Assim, percebe-se que cada um desses usuários não poderia fornecer, individualmente, as informações requeridas para que se abrangessem todos os requisitos, tendo cabido ao autor a tarefa de consolidá-las.

Dessa forma, percebe-se que a modelagem orientada a objetos teve papel fundamental ao permitir que o autor compreendesse todo o contexto do estudo, fornecendo uma visão ampla e possibilitando que se unissem os comentários de cada um dos usuários. Assim, pôde-se chegar apropriadamente à definição dos requisitos do sistema sugerido.

Julga-se relevante também salientar a importância das etapas posteriores, que não se enquadram no escopo do trabalho realizado pelo autor, como a criação da arquitetura, a condução dos testes de validação e a implantação *de facto* do sistema na tesouraria do Banco Silva. Este Trabalho de Formatura, através dos requisitos levantados, da definição dos casos de uso, das classes, dos relacionamentos entre as classes e da avaliação dos benefícios resultantes, pretende servir de base para que a equipe responsável por projetos de tecnologia do banco possa implementar esses próximos passos.

Ressalta-se igualmente que, apesar de o presente trabalho ter sido realizado no ambiente de uma instituição bancária, a modelagem UML e toda a definição de requisitos podem ser extremamente úteis quando aplicadas aos mais diversos contextos. Dessa forma, o autor considera que estes instrumentos da modelagem orientada a objetos são ferramentas poderosas para se analisarem e se resolverem problemas de sistemas de informação.

O autor julga que este método oferece diversas vantagens, como a grande facilidade para descrever estruturas e para abranger problemas grandes e complexos, além de garantir que se reutilizem classes e seus métodos, facilitando a sua manutenção e a implementação posterior de novas funcionalidades. No entanto, ressalta-se o fato de que, para sistemas muito simples, este método pode trazer um grau de complexidade indesejável e desnecessário. Portanto, primeiramente, deve-se refletir se o método é realmente adequado de acordo com as características e complexidade do sistema que será estudado. Para o sistema analisado neste trabalho, como já mencionado, o método verificou-se extremamente vantajoso.

Caso as novas funcionalidades propostas sejam incorporadas pelo sistema do Banco Silva, poder-se-á confirmar as consequentes vantagens trazidas por ele. Isso pode representar um estímulo para que sejam adotados mais projetos de melhoria nos sistemas de informação do banco, cuja eficiência foi tão prejudicada em função da recente fusão pela qual passou.

Por fim, pode-se concluir que este Trabalho de Formatura utilizou diversos conceitos adquiridos durante a formação de Engenharia de Produção. Foi possível aplicar, no contexto de um grupo bancário de grande porte, no qual o autor trabalha, muitos dos ensinamentos obtidos referentes a sistemas de informação, modelagem e gestão de projetos. A utilização de tais ferramentas possibilitou que se analisasse um problema real e se propusessem soluções capazes de ser revertidas em vantagens competitivas para o banco.

## REFERÊNCIAS BIBLIOGRÁFICAS

BELL, D. **O Advento da Sociedade Pós-Industrial**. São Paulo: Cultrix, 1974.

BIO, B. F. **Sistema de Informação: um enfoque gerencial**. São Paulo: Atlas, 1996.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language Reference Manual**. Reading: Addison-Wesley, 1998.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. Reading: Addison-Wesley, 1999.

BRYNJOLFSSON, E.; HITT, L. M. Beyond the Productivity Paradox. **Communications of the ACM**, August, 1998.

BURLESON, D. **Advantages and Disadvantages of Object-Oriented Approach**. Disponível em: < [http://www.dba-oracle.com/t\\_object\\_oriented\\_approach.htm](http://www.dba-oracle.com/t_object_oriented_approach.htm) >. Acesso em: 22 abr. 2012.

GIL, A. L. **Sistema de Informações Contábil/Financeiros**. 3<sup>a</sup> ed. São Paulo: Atlas, 1999.

HOUAISS, A. **Dicionário Eletrônico Houaiss de Língua Portuguesa**. Versão 3.0. Rio de Janeiro: Objetiva, 2009.

LACOMBE, F. J. M.; HEILBORN, G. L. J. **Administração: princípios e tendências**. 1<sup>a</sup> ed. São Paulo: Saraiva, 2003.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de informações gerenciais**. 7<sup>a</sup> ed. São Paulo: Pearson Prentice Hall, 2007.

MAÇADA, A.; BECKER, J. **Análise da Eficiência Relativa dos Investimentos em TI nos Bancos Brasileiros**. Anais do XXV ENANPAD, ANPAD: Campinas, 2001.

O'BRIEN, J. A. **Sistemas de Informação e as decisões gerenciais na era da internet**. 2<sup>a</sup> ed. São Paulo: Saraiva, 2004.

OMG – Object Management Group. **UML Superstructure Specification**. Versão 2.4.1. Agosto de 2011.

- PÁDUA, W. **Engenharia de Software: fundamentos, métodos e padrões.** 3<sup>a</sup> ed. Rio de Janeiro: LTC, 2009.
- PADOVEZE, C. L. **Sistema de informações contábeis: fundamentos e análise.** 3<sup>a</sup> ed. São Paulo: Atlas, 2002.
- RUMBAUGH, J. et al. **Modelagem e projetos baseados em objetos.** Rio de Janeiro: Campus, 1994.
- SCHMIDT, P. **Controladaria: agregando valor para a empresa.** Porto Alegre: Bookman, 2002.
- SOMMERVILLE, I. **Engenharia de Software.** 8<sup>a</sup> ed. São Paulo: Addison-Wesley, 2007.
- STONIER, T. **The Wealth of Information: A Profile of the Post-Industrial Economy.** London: Methuen, 1983.
- TOURAINE, A. **La société post-industrielle.** Paris: Denoël/Gonthier, 1969.
- WEILL, P. **The relationship between investment in Information Technology and firm performance in the manufacturing sector.** PhD. Dissertation, Stern School of Business, New York University, 1989.