

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ESCOLA DE ENGENHARIA

DEPARTAMENTO DE ELETRÔNICA

Software Financeiro com Suporte à Tomada de Decisões

Autor: _____

Lissandro Ferreira da Rocha

Orientador: _____

Prof. Antônio Cláudio Gómez de Sousa

Examinador: _____

Prof. Flávio Luis de Mello

Examinador: _____

Prof. Aloysio de Castro Pinto Pedroza

DEL

Novembro de 2008

Dedicatória

Dedico este trabalho aos que me apoiaram ao longo desta trajetória e a tornaram mais interessante e proveitosa. Ao meu pai, que me incentivou a dar o melhor diante de cada situação. À minha mãe, que me acalmou nos momentos difíceis, colocando as coisas no lugar sempre que elas teimavam em sair. Ao meu irmão, que mesmo à distância contribuiu de forma decisiva na minha carreira. À minha namorada, que suportou os fins de semana perdidos e a separação durante boa parte deste trajeto. Aos meus avós, que sempre estiveram muito presentes nesta caminhada. Por fim, à Leny, minha segunda mãe, que me ajudou sempre em tudo o que precisei.

Agradecimentos

Agradeço primeiramente à minha família por ter me apoiado e me incentivado sempre a ser um bom profissional.

Agradeço também aos meus amigos da faculdade que enriqueceram esta experiência e compartilharam comigo minhas alegrias e frustrações.

Agradeço ao professor Antônio Cláudio Gómez de Sousa pela orientação desse projeto e aos professores Flávio Luis de Mello e Aloysio de Castro Pinto Pedroza.

Resumo

O trabalho aqui apresentado é um projeto de final de curso na graduação Engenharia Eletrônica e de Computação da Universidade Federal do Rio de Janeiro.

As áreas de trabalho escolhidas para o desenvolvimento do projeto foram Computação e Finanças.

O sistema consiste em um software em linguagem Delphi que servirá de base para a tomada de decisão frente a vários problemas ligados a finanças. O armazenamento de dados é feito através do formato XML (eXtensible Markup Language). As entradas do sistema são via XML ou através de interface gráfica, gerada através do Delphi. As saídas também podem ser no formato XML, pela interface gráfica ou ainda por meio da geração de relatórios em HTML, o que facilita a visualização e torna a utilização do software mais prática.

Quanto aos problemas tratados pelo software, temos a capitalização do dinheiro ao longo do tempo pelos métodos simples, composto e contínuo. Para a capitalização composta, podem ser utilizados dados de mercado, como por exemplo cotações das ações listadas na Bovespa.

Além disso, há uma ferramenta de análise de investimentos que aplica os métodos de Rentabilidade Simples, Período de Retorno de Investimento, Valor Presente Líquido e Taxa Interna de Retorno e gera relatórios comparativos sobre diversos projetos de investimento, visando auxiliar o gestor na escolha do projeto mais adequado a sua empresa.

Também há o suporte para análise de opções de ações, sobre as quais se pode calcular o preço teórico pelo modelo de Black & Scholes, assim como as diversas “gregas” que representam a sensibilidade do preço destas opções aos seus vários parâmetros de entrada. São 18 analíticos que permitem ao usuário determinar um bom momento para compra e venda destas opções, bem como montar estratégias visando maximizar o lucro.

Entre as funcionalidades do projeto também está a análise de risco. Através do cálculo de séries de retornos e medidas básicas de estatística (média, variância, desvio-

padrão), o usuário pode determinar qual o melhor investimento entre um máximo de 4 disponíveis para análise simultânea. É fornecido um percentual de sucesso para a simulação tomando como premissa que as séries de retorno analisadas seguem o padrão de uma distribuição normal de probabilidade.

Palavras-Chave

Finanças, Análise de Investimentos, Opções, Avaliação de Risco, Valor Presente, Capitalização.

Glossário

- **Ativo Subjacente:** em uma opção, refere-se ao ativo sobre o qual a opção foi negociada.
- **Fator de Carrego:** fator utilizado para levar um fluxo de caixa no presente até uma data futura.
- **Fator de Desconto:** fator utilizado para trazer um fluxo de caixa no futuro até a data presente.
- **Fluxo de Caixa:** representa uma entrada ou saída de investimento em um determinado projeto de investimento.
- **Mercado de Derivativos:** mercados em que operações com liquidações futuras são implementadas, tornando possível a gestão do risco de preço de diversos ativos.
- **Opção:** são instrumentos financeiros utilizados no mercado de capitais. Uma opção confere, ao titular, o direito (e não obrigação) de comprar ou de vender um determinado ativo (ação, título ou bem qualquer) por um valor determinado.
- **Renda Fixa:** Alternativa de investimento praticamente sem riscos e que permite ao investidor determinar seus lucros no futuro.
- **Renda Variável:** Alternativa de investimento que envolve riscos e não permite ao investidor determinar o lucro que terá no futuro.
- **Spread:** diferença entre o preço de compra(procura) e venda(oferta) da mesma ação, título ou transação monetária.
- **Strike:** preço de exercício de uma opção, ou seja, o preço que irá vigorar na data de vencimento do contrato.
- **Sub-prime:** uma linha de crédito considerada de maior risco e que acabou atingindo um nível preocupante de inadimplência nos Estados Unidos.
- **TMA (taxa mínima de atratividade):** taxa mínima a partir da qual um projeto passa a ser viável economicamente.
- **Volatilidade:** é uma medida da velocidade do mercado: mercados que se movem lentamente são mercados de baixa volatilidade e os que se movem rapidamente são mercados de alta volatilidade.
- **XML (*eXtensible Markup Language*):** é uma linguagem capaz de descrever diversos tipos de dados de forma estruturada.

Índice

| | |
|--|-----------|
| 1. Introdução | 1 |
| 1.1 O Problema | 1 |
| 1.2 Motivação e Localização | 2 |
| 1.3 Tecnologias | 3 |
| 1.4 Sumário | 3 |
| 2. Metodologia..... | 5 |
| 2.1 A Metodologia Estruturada na construção de Softwares | 5 |
| 2.2 Ferramentas e ambiente..... | 6 |
| 2.2.1 Delphi..... | 6 |
| 2.2.2 Notepad++ | 6 |
| 2.2.3 Xml Notepad 2007 | 7 |
| 2.2.4 Calc | 7 |
| 3. Análise do Problema..... | 8 |
| 3.1 Descrição..... | 8 |
| 3.2 Especificação de Requisitos | 8 |
| 3.2.1 Eventos..... | 8 |
| 3.2.2 Diagramas de fluxo de dados..... | 9 |
| 3.2.2.1 Alterar as configurações do Sistema..... | 9 |
| 3.2.2.2 Realizar cálculos de Capitalização..... | 10 |
| 3.2.2.3 Realizar análise de fluxos de caixa..... | 10 |
| 3.2.2.4 Realizar Análise de Investimentos | 11 |
| 3.2.2.5 Realizar Análise de Risco de Investimentos em Renda Variável | 12 |
| 3.2.2.6 Precificação e cálculo de sensibilidades de opções de ações | 13 |
| 3.2.2.7 Solicitar a ajuda do sistema | 14 |
| 3.2.2.8 Verificar a versão do sistema..... | 15 |
| 3.3 Características do sistema | 15 |
| 3.3.1 Amigabilidade | 15 |
| 3.3.2 Confiabilidade..... | 15 |
| 3.3.3 Manutenibilidade..... | 15 |
| 3.3.4 Flexibilidade..... | 16 |
| 4. Estrutura do Projeto | 17 |
| 4.1 Visão Geral..... | 17 |

| | | |
|------------|---|-----------|
| 4.2 | Detalhamento | 19 |
| 4.2.1 | O Módulo de Mercado | 19 |
| 4.2.1.1 | Basis : Contagem de Dias | 19 |
| 4.2.1.2 | <i>Benchmark</i> | 20 |
| 4.2.1.3 | Série Histórica | 21 |
| 4.2.1.4 | Curva | 22 |
| 4.2.1.5 | Calendários | 22 |
| 4.2.1.6 | Superfícies de volatilidade | 22 |
| 4.2.2 | Resumo | 23 |
| 4.2.3 | Adicionais | 24 |
| 4.2.4 | Análise de Investimentos..... | 24 |
| 4.2.5 | Avaliação de Risco | 24 |
| 4.2.6 | Capitalização..... | 25 |
| 4.2.6 | Opções..... | 25 |
| 4.2.7 | Projetos de Investimentos | 25 |
| 4.2.8 | Valor Presente..... | 25 |
| 4.3 | Dados | 26 |
| 4.4 | Interfaces externas..... | 27 |
| 5. | Desenvolvimento do Projeto | 28 |
| 5.1 | Histórico | 28 |
| 5.2 | Dificuldades Encontradas | 28 |
| 5.2.1 | Armazenamento..... | 29 |
| 5.2.2 | Componente | 29 |
| 5.2.3 | Salvamento de Arquivo..... | 29 |
| 5.2.4 | Caminhos Relativos | 29 |
| 5.2.5 | Comunicação com o Excel e Calc..... | 30 |
| 5.2.6 | Cores diferentes para os resultados | 30 |
| 5.3 | Dependência entre os Módulos..... | 30 |
| 6. | Resultados | 32 |
| 6.1 | Capitalização..... | 32 |
| 6.2 | Valor Presente..... | 32 |
| 6.3 | Análise de Investimentos..... | 32 |
| 6.4 | Avaliação de Risco | 32 |
| 6.5 | Opções..... | 33 |

| | |
|--|-----------|
| 7. Estrutura dos Testes | 34 |
| 7.1 Testes de Interface | 34 |
| 7.2 Testes na exibição das mensagens de erro | 34 |
| 7.3 Testes de precisão nos cálculos | 34 |
| 7.3.1 Capitalização | 35 |
| 7.3.2 Valor Presente | 36 |
| 7.3.3 Opções | 37 |
| 8. Conclusão | 39 |
| 8.1 Avaliação das Tecnologias | 39 |
| 8.2. Avaliação da metodologia | 39 |
| 8.3. Avaliação das ferramentas | 40 |
| 8.4 Aprendizado | 40 |
| 8.5 Propostas Futuras | 41 |
| 9. Bibliografia | 43 |
| Apêndice 1: O Modelo de Black & Scholes | 44 |
| Apêndice 2: Estruturas do Sistema | 48 |
| Apêndice 3: Instalação e Atualização de Dados | 62 |
| Manual do Usuário | 63 |

1. Introdução

Nesta parte serão tratadas as considerações iniciais sobre o projeto, o surgimento da idéia, a motivação e a justificativa das tecnologias utilizadas.

Para isso, a introdução se dividirá em 3 partes: Problema, Motivação e Localização, e Tecnologias, nas quais será apresentado o problema, discutidas as motivações e justificadas as tecnologias utilizadas no projeto.

1.1 O Problema

A cada dia cresce mais a participação dos brasileiros no mercado de ações. Nos Estados Unidos já são aproximadamente 17% da população com dinheiro investido em bolsas, mas nós ainda não alcançamos os 2%. Com o advento do *Home Broker* no Brasil, em 1999, a participação de pessoas físicas no mercado financeiro foi grandemente facilitada e uma multidão de pequenos investidores já soma boa parte do total de investimentos do país.

Tendo em vista este cenário, se tornam cada vez mais necessários softwares que auxiliem os investidores na hora de realizar uma operação financeira, já que a cada ano aumenta o número de investidores que, sem experiência e seduzidos por lucros fáceis, afundam em dívidas por desconhecimento de princípios básicos de investimento.

O software procura suprir parte desta lacuna auxiliando o investidor na tomada de decisões. Entre as várias aplicações do programa, sobressaem a análise de investimentos e o suporte à tomada de decisões na compra e venda de opções, bem como a avaliação de risco de investimentos em renda variável.

Especialmente em momentos de crise, como este que vivemos hoje, com os problemas devido ao *sub-prime* americano, é de fundamental importância poder enxergar o mercado de forma realista e é neste ponto que o software mostra sua utilidade.

1.2 Motivação e Localização

O consumidor comum não dispõe, na maioria das vezes, de métodos de cálculo que lhe permitam analisar situações corriqueiras que dizem respeito ao valor do dinheiro no tempo. Saber o valor à vista daquele eletrodoméstico comprado à prestação ou saber a que taxa efetiva foi feito um determinado financiamento deveria estar ao alcance de todos. Deveria. Mas o que ocorre no Brasil é justamente o oposto: poucos brasileiros sabem o quanto pagam pelo dinheiro que tomam emprestado ou financiam. Neste caso, o software visa fornecer soluções simples que auxiliem o consumidor comum na hora de assumir compromissos no mercado.

Já a parte que trata da análise de investimentos, através de uma matemática um pouco mais refinada, é de grande utilidade para tomada de decisão em empresas que buscam alternativas de maior lucratividade. O mercado brasileiro é carente de softwares que sirvam para este tipo de análise, dado o fato de a nossa cultura de investimento ainda estar engatinhando. Sendo assim, os softwares que realizam este tipo de análise são importados e trazem consigo toda uma dificuldade no que diz respeito à adaptação ao nosso mercado.

Quanto à tomada de decisões na compra/venda de opções, a demanda é enorme e vem crescendo muito nos últimos anos. O mercado de derivativos (contratos que tem seu preço dependente do preço de um outro ativo) registra enorme crescimento nos últimos cinco anos e cada vez mais os profissionais do mercado precisam estar bem informados a seu respeito. A Teoria de Opções evoluiu muito nos últimos anos, especialmente com o nascimento do modelo Black & Scholes, que rendeu um Prêmio Nobel a seus descobridores em 1997. A análise estatística dos dados do mercado tem sido fundamental na saúde de toda grande e média empresa. Assim, pretende-se desenvolver um método que alcance um maior grau de confiança para operações com estes tipos de contrato.

Quanto à avaliação de risco de investimentos em renda variável, o software fornece a avaliação do cenário atual do mercado e projeta os rendimentos para o futuro, através de medidas estatísticas como média, variância e desvio-padrão. Com isso, podem ser tomadas decisões mais acertadas a respeito da alocação do investimento em carteiras mais lucrativas.

1.3 Tecnologias

No que diz respeito às tecnologias, o sistema foi todo construído na linguagem Delphi, por ser uma linguagem que permite a montagem de uma interface gráfica muito funcional de uma forma simples e intuitiva.

Para o armazenamento de dados foi escolhido o formato XML (eXtensible Markup Language), por ser uma forma estruturada e bem difundida de armazenamento. Não foi criado um banco de dados para o projeto porque o foco estava na criação de funcionalidades e na utilidade no projeto, e não em parâmetros como por exemplo o desempenho.

As saídas do sistema são em sua maioria exibidas via interface gráfica, mas em alguns casos elas são salvas no formato XML ou até mesmo HTML (HyperText Markup Language), como é o caso dos relatórios gerados pelo módulo de Análise de Investimentos.

Para a validação da base de dados, que é toda em formato XML, foi utilizado o software XML Notepad 2007 e o próprio Internet Explorer. Para a validação dos cálculos, foram utilizados a calculadora financeira HP 12C, a ferramenta Calc do OpenOffice e o Matlab. No diretório *C:\Projeto\Planilhas* podem ser encontradas planilhas com exemplos de cada caso possível de cálculo do sistema. Assim, os cálculos puderam ser demonstrados e validados.

Para registrar as DLLs utilizadas pelo sistema, deve-se acessar o diretório *C:\Projeto\Build* e rodar o arquivo “Registra.bat”.

Para a formação do arquivo HTML de saída dos relatórios foi utilizado um modelo que pode ser observado em *C:\Projeto\RelatoriosHTML*.

1.4 Sumário

O relatório será dividido nas seguintes partes: Metodologia, Análise do Projeto, Estrutura do Sistema, Desenvolvimento, Resultados, Estrutura de Testes e Conclusão.

Na parte de Metodologia, é especificada a metodologia utilizada no desenvolvimento do projeto e é dado um apanhado geral sobre as tecnologias utilizadas. No que diz respeito à parte de Análise do Problema, serão dados os eventos possíveis no sistema, além de seus requisitos básicos. Também serão definidas as características do produto final.

Na parte de estrutura do sistema será discutida a estrutura tanto de codificação quanto de armazenamento de dados, fazendo o *link* entre as duas partes sempre que possível. Também serão definidas as interfaces externas, que representam a comunicação com o usuário. Além disso, é dada uma visão geral do sistema, a fim de facilitar o entendimento dos capítulos subsequentes.

No capítulo sobre Desenvolvimento traçamos o histórico do projeto, desde o surgimento da idéia até os vários problemas encontrados na implementação e como se conseguiu chegar a soluções para cada um deles.

Na parte de Resultados, são comentados os resultados obtidos em cada módulo do sistema e é feita uma avaliação de cada um separadamente.

Na parte de estrutura de Testes foi mostrado como foram feitos os vários testes no sistema, desde os testes de interface, passando pelos testes do sistema de *log* até os cálculos de precisão dos cálculos financeiros, estes últimos validados por planilhas confeccionadas no OpenOffice Calc.

Em seguida, são feitas conclusões sobre o aspecto geral do projeto. Foram avaliadas a metodologia de projeto, as tecnologias escolhidas e as ferramentas utilizadas. Também foi mostrado o aprendizado proporcionado pelo desenvolvimento do projeto e apontadas propostas de melhorias futuras para o projeto.

Ao final, foram incluídos anexos que dão uma base teórica sobre o modelo de Black & Scholes, utilizado pelo sistema no módulo de opções. Além disso, foi incluído um anexo com as estruturas das principais classes do sistema, com seus atributos e métodos. Ao final, encontra-se um Manual do Usuário bastante detalhado focado em orientar o usuário a resolver possíveis problemas que venha a ter na utilização do software.

2. Metodologia

Neste capítulo será discutida a metodologia usada no projeto. Essa metodologia foi absorvida durante a disciplina de Engenharia de Software e, posteriormente, utilizada neste projeto final.

O ciclo de vida do projeto segue o esquema de cascata, e no item 3 deste relatório pode ser encontrada a parte referente à análise.

Para isso, será discutida a metodologia estruturada para desenvolvimento de software, que foi a escolhida para esse projeto. Também abordaremos o ambiente e as ferramentas usadas no mesmo, sendo estas: os ambientes Delphi e Calc; a linguagem de programação Delphi 2005 e, por fim, os formatos de dados XML e HTML.

2.1 A Metodologia Estruturada na construção de Softwares

A metodologia escolhida para o projeto foi a estruturada. Essa metodologia propõe uma distinção clara entre dados e funções. Desse modo, o sistema pode ser dividido em duas partes: projeto de dados e projeto de funções.

Um aspecto fundamental da fase de projeto de dados consiste em estabelecer de que forma serão armazenados os mesmos no sistema. A plataforma de implementação para armazenamento de dados utilizada no projeto foi o armazenamento via arquivos XML.

O projeto estruturado de programas envolve duas grandes etapas: o projeto da arquitetura do sistema e o projeto dos módulos. Paralelamente, devem ser feitos o projeto de dados e o projeto da interface com o usuário. Este projeto foi planejado de forma a permitir a maior comunicação possível entre os módulos, para que os cálculos ficassem mais completos.

Para isso a estrutura será dividida em dois capítulos: um de análise e outro de projeto. No primeiro, descreveremos o produto e suas funções, assim como as características do usuário. Além disso, abordaremos requisitos específicos do sistema através de eventos, diagrama de fluxo de dados e diagramas de transição de estado. Concluindo essa primeira etapa, vamos abordar as interfaces externas com usuários e os meios de comunicação. Serão feitas referências aos apêndices quando necessário.

No capítulo de projeto abordaremos a arquitetura do sistema, incluindo sua decomposição em módulos e a dependência entre os mesmos.

2.2 Ferramentas e ambiente

Esta seção mostra um pouco melhor o ambiente de desenvolvimento utilizado no projeto. Para isso, serão mostrados os ambientes Delphi, do Notepad++, XML Notepad 2007, e Calc.

2.2.1 Delphi

O ambiente de desenvolvimento escolhido foi o Delphi e as variáveis que contribuíram para esta escolha foram a grande familiaridade com a linguagem, já que esta é ferramenta de trabalho do autor há quase 2 anos e também pelo prático ambiente de desenvolvimento proporcionado pelo Borland Delphi 2005.

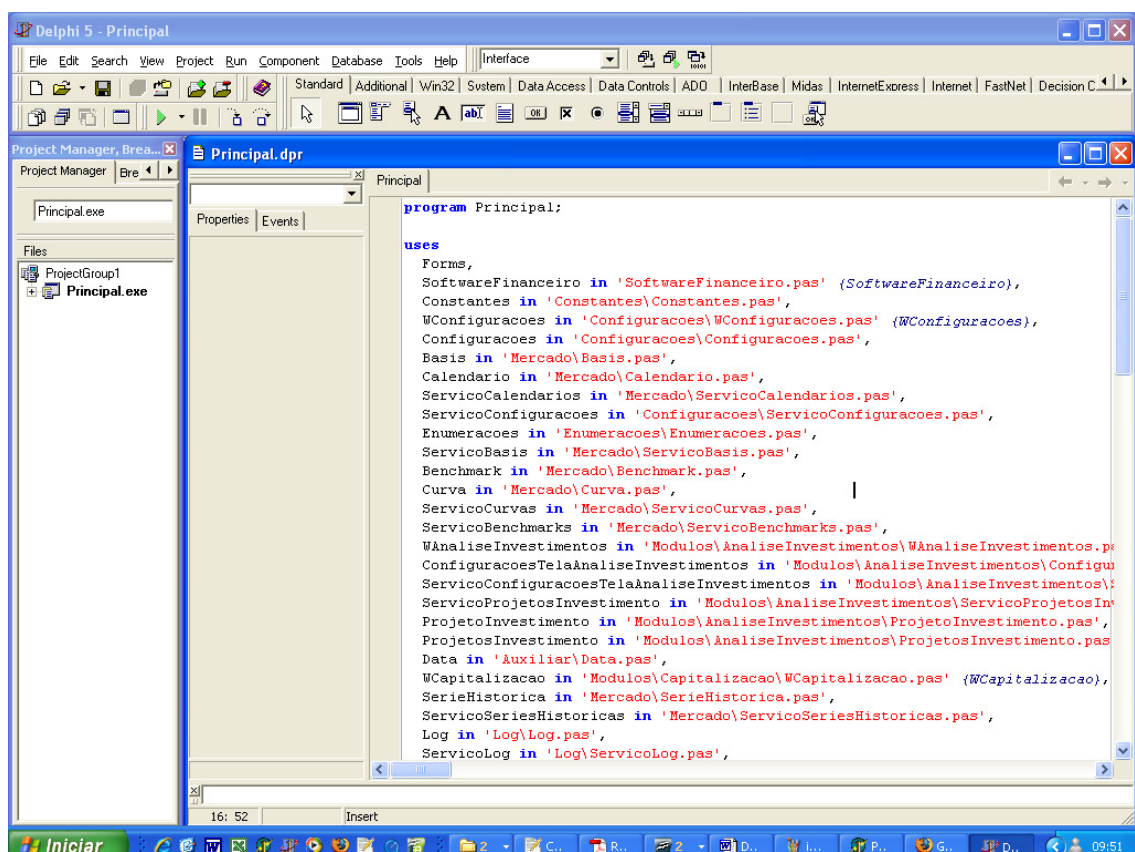


Figura 1 - Ambiente de desenvolvimento Borland Delphi 2005

2.2.2 Notepad++

Este compacto mas poderoso editor foi de grande ajuda na validação dos arquivos XML e HTML, já que possui um sistema de cores que facilita a visualização. Ele foi utilizado para edição dos arquivos e para validação de dados, em conjunto com o XML Notepad 2007.

2.2.3 Xml Notepad 2007

Este software foi utilizado unicamente para fins de validação dos arquivos no formato XML que servem de entrada ao sistema. Sua ferramenta de visualização em forma de árvore permitiu uma análise mais acurada dos dados, facilitando assim a correção de problemas nos dados. No caso deste projeto, a consistência dos dados é fundamental, já que temos arquivos grandes de entrada (por exemplo, o de cotações de séries históricas tem entre 30000 e 40000 linhas). Abaixo, um exemplo da estrutura em árvore mostrada pelo XML Notepad 2007:

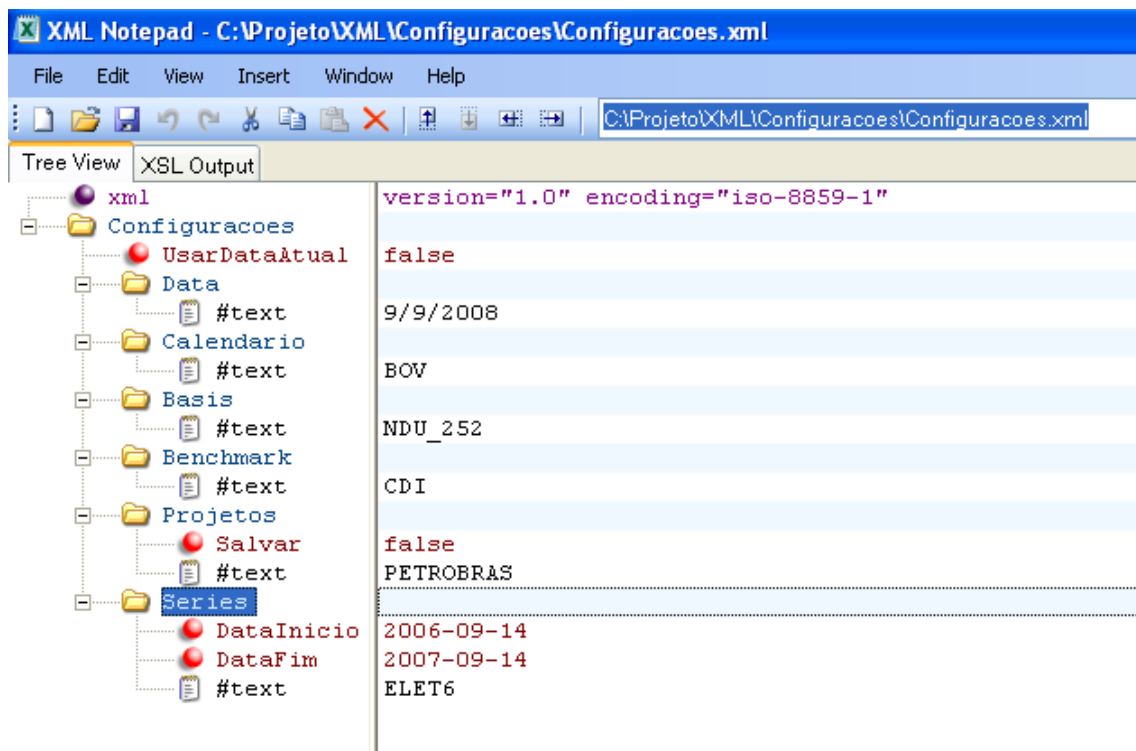


Figura 2 - Estrutura em árvore visualizada no XML Notepad 2007

2.2.4 Calc

Esta ferramenta pertencente ao pacote OpenOffice foi de fundamental importância para o sucesso do projeto. Com ela foi possível testar o sistema de uma forma rápida e

de fácil visualização. As planilhas com os testes de cada funcionalidade do sistema se encontram em *C:\Projeto\Planilhas*.

3. Análise do Problema

Aqui será feita uma análise do problema e serão descritos os requisitos básicos do sistema.

3.1 Descrição

O projeto é um software que tem como objetivo fornecer algumas funcionalidades úteis no dia-a-dia do mercado financeiro. Assim, serão abordados problemas de Capitalização, Análise de Investimentos, Opções e Avaliação de Risco.

3.2 Especificação de Requisitos

Neste tópico serão especificados os requisitos que o sistema deve atender. Ele será dividido em 3 partes: Eventos, Diagramas de Fluxo de Dados e Diagramas de Transição de Estados.

3.2.1 Eventos

Os eventos possíveis para o sistema serão:

- Alterar as configurações do sistema (Data, Calendário, Basis e *Benchmark*);
- Realizar cálculos de capitalização simples;
- Realizar cálculos de capitalização composta;
- Realizar cálculos de capitalização contínua;
- Analisar fluxos de caixa pelo método de Rentabilidade Simples;
- Analisar fluxos de caixa pelo método de *Pay-back Period*;
- Analisar fluxos de caixa pelo método de Valor Presente Líquido;
- Analisar fluxos de caixa pelo método da Taxa Interna de Retorno;
- Gerar e/ou exibir relatórios HTML contendo as análises acima mencionadas e uma avaliação comparativa entre os projetos;
- Realizar análise de risco tomando como base um cenário passado;
- Precificar e calcular “gragas” de opções de ações;

- Exibir a ajuda do sistema e/ou tutoriais.

Os eventos acima foram colocados de forma bem geral, pois um nível alto de detalhes dificultaria a compreensão das idéias básicas do sistema. Para uma idéia mais detalhada sobre os eventos acima mencionados, consultar o Manual do Usuário e os Tutoriais disponíveis.

3.2.2 Diagramas de fluxo de dados

Nesta parte trataremos dos diagramas de fluxos de dados do sistema, com o objetivo de dar melhor entendimento sobre as suas funcionalidades. Para isso, será montado um diagrama de fluxo de dados para cada tela do sistema, mostrando o processo de cálculo e a exibição de resultados.

3.2.2.1 Alterar as configurações do Sistema

O processo de alterar as configurações do sistema pode ser acessado através da tela de configurações, o primeiro item no menu do programa principal. A tela é exibida com as informações atuais do sistema. O usuário fornece o dia, mês, ano, basis, *benchmark* e calendário. Após clicar no botão “Salvar” as configurações serão salvas no sistema (Veja tópico Configurações, no Manual do Usuário).

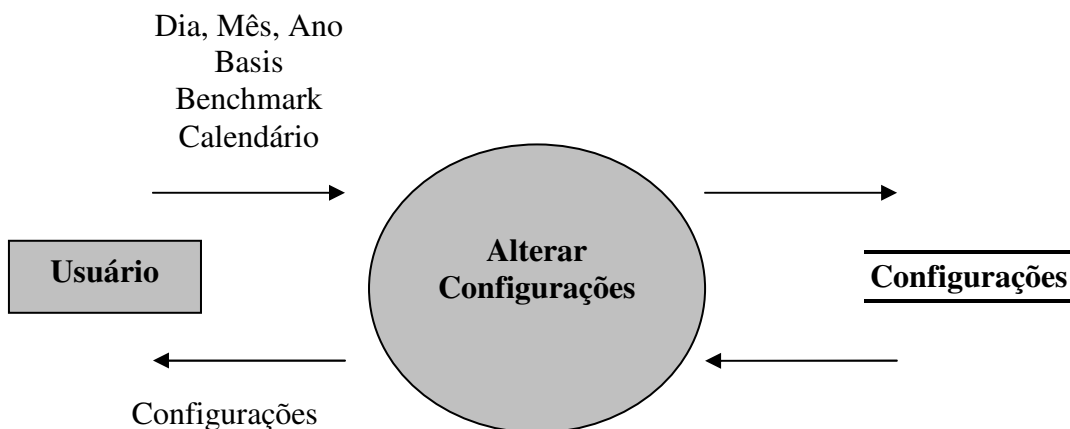


Figura 3 - DFD “Alterar as configurações do Sistema”

3.2.2.2 Realizar cálculos de Capitalização

Para realizar um cálculo de capitalização basta ir à tela de capitalização, o segundo item da tela principal. Nela, o usuário escolhe o tipo de capitalização a ser utilizado (simples, composta ou contínua), a data de início de capitalização, a data de término da capitalização, o *benchmark* (no caso de capitalização composta; para um maior grau de detalhe, veja o Anexo *Benchmark*), e dois valores entre principal, taxa fixa e montante. O sistema detecta automaticamente a variável não preenchida e a calcula, fornecendo o resultado na cor azul.

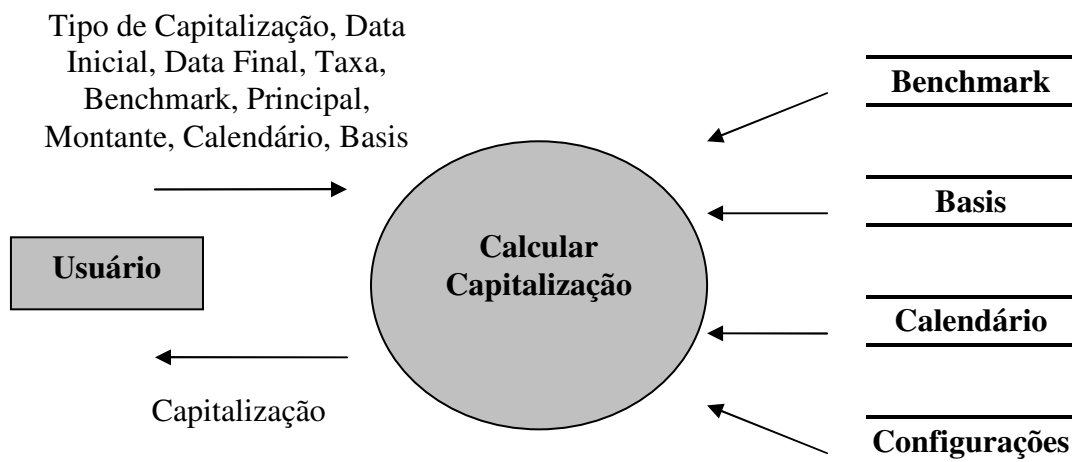


Figura 4 - DFD “Realizar cálculos de Capitalização”

3.2.2.3 Realizar análise de fluxos de caixa

Para realizar uma análise dos fluxos de caixa de um determinado projeto, o usuário deve acessar a tela de Valor Presente, o terceiro item do menu principal. Nesta tela ele deverá escolher a lista de fluxos, o método de análise dos fluxos de caixa (Rentabilidade simples, *Pay-Back Period*, Valor Presente ou Taxa Interna de Retorno) e a Data de Análise (data na qual os cálculos serão realizados), a Data de Mercado (o sistema irá olhar o mercado a partir desta data). Como resposta, o usuário receberá o valor do método escolhido para aquele projeto, além de uma lista de fluxos de caixa com seus respectivos valores presentes líquidos, caso seja escolhida a opção Valor Presente.

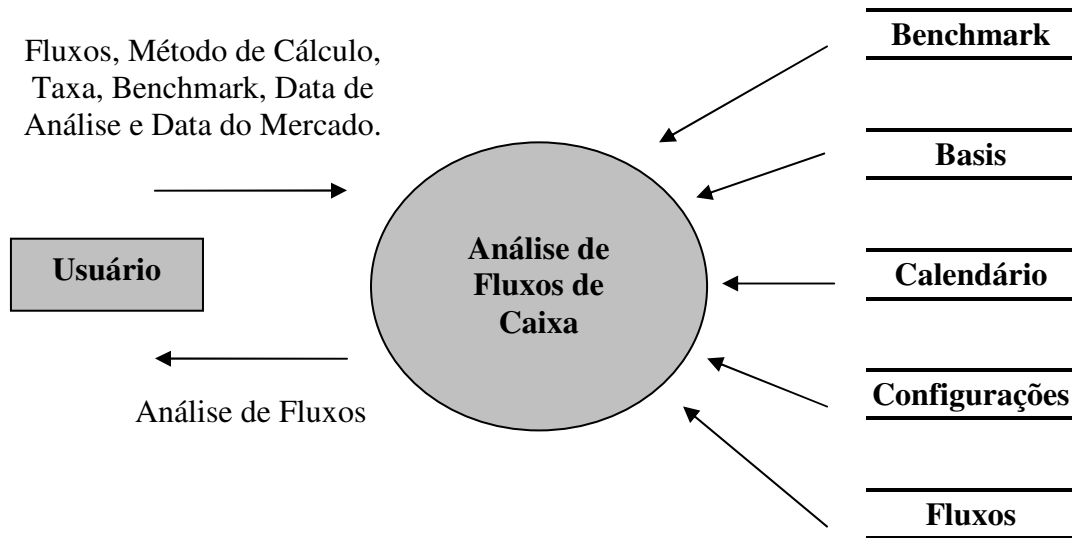


Figura 5 - DFD “Realizar análise de fluxos de caixa”

3.2.2.4 Realizar Análise de Investimentos

Para realizar uma análise comparativa de projetos de investimento o usuário deve acessar a tela de análise de investimentos. Nela, ele irá escolher os projetos de investimentos (uma lista com no mínimo um projeto), a data de mercado a ser utilizada, um *flag* que indica se os vários projetos serão ou não comparados, um *flag* que indica se o relatório deverá ser exibido ao finalizar o cálculo, um *flag* indicando se o relatório deverá ser salvo ou não em disco, o caminho no qual o relatório deverá ser salvo, a Data de Análise, as configurações (basis, benchmark, calendário, taxa fixa, os *flags* de usar configurações do sistema e manter configurações dos projetos) e as informações referentes aos métodos de análise (*flags* indicando que métodos serão calculados – rentabilidade simples, *pay-back period*, valor presente e taxa interna de retorno – a taxa mínima de atratividade da empresa, o período médio de retorno do investimento e os percentuais de confiança dos métodos de rentabilidade simples e valor presente). Como saída deste módulo temos o relatório em formato HTML com a análise comparativa dos projetos de investimento.

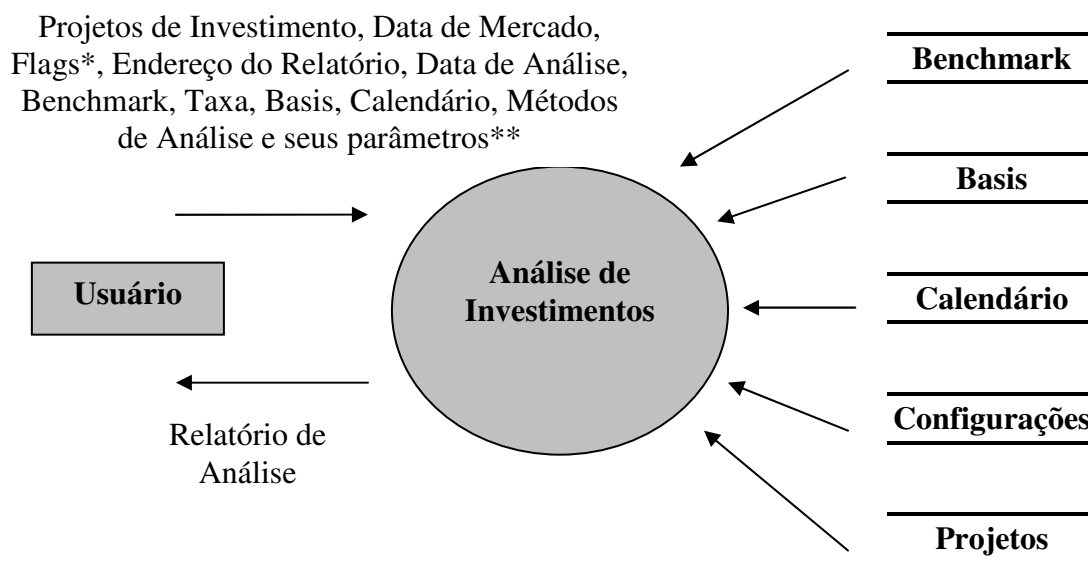


Figura 6 - DFD “Realizar Análise de Investimentos”

**Flags*: Comparação de projetos, Exibição e Salvamento de Relatório, usar data atual, usar configurações do sistema, manter configurações dos projetos.

***Métodos de Análise e seus parâmetros*: flags de Rentabilidade Simples, *Pay-back Period*, Valor Presente e Taxa Interna de Retorno, taxa mínima de atratividade da empresa, período médio de retorno do investimento, percentual de confiança do método de Rentabilidade Simples, percentual de confiança do método de Valor Presente.

3.2.2.5 Realizar Análise de Risco de Investimentos em Renda Variável

Para realizar a análise de risco de investimentos em renda variável no sistema, o usuário deve acessar a tela de análise de risco no menu principal. Nela, ele deve entrar com um mínimo de 1 e um máximo de 4 ativos para comparação. Além disso, deve fornecer as datas de início e fim do período que será tomado como base para o comportamento estatístico dos ativos selecionados: estas datas estão representadas no sistema como início e fim do histórico. O usuário deve delimitar ainda o período que pretende investir seu dinheiro e que rentabilidade ele espera obter neste período. Como saída ele terá na interface a média, variância, desvio-padrão, a série de retornos diários para cada série escolhida (para isso basta clicar na série correspondente ao ativo que se deseja visualizar), o período do investimento em dias, o período do histórico em dias, a

rentabilidade média do período e o percentual de sucesso em obter a rentabilidade fornecida no período escolhido, caso o mercado se comporte como especificado no histórico. Segue abaixo o diagrama de fluxo de dados:

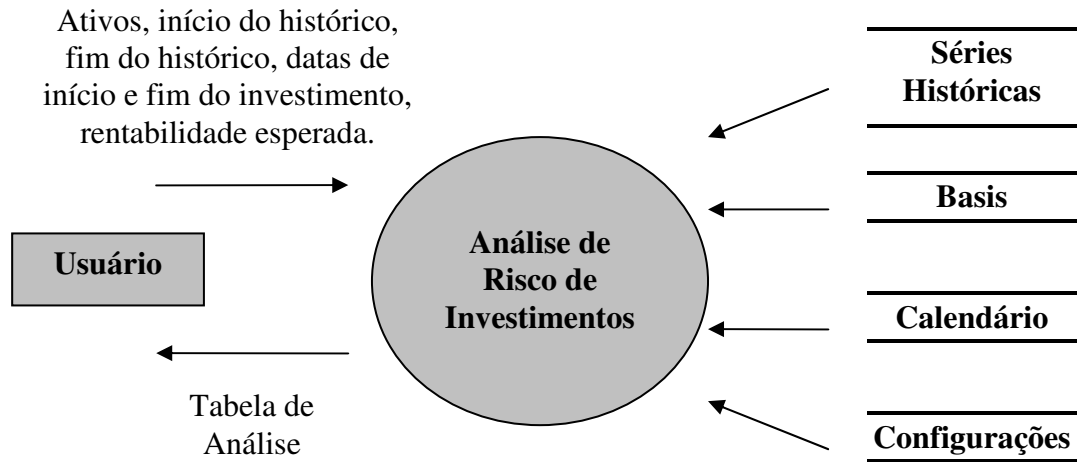


Figura 7 - DFD “Realizar Análise de Risco de Investimentos em Renda Variável”

3.2.2.6 Precificação e cálculo de sensibilidades de opções de ações

Para trabalhar com opções o usuário deve acessar a tela de opções, no menu principal do sistema. Nela ele poderá escolher uma das opções cadastradas no sistema para análise. O diagrama de fluxo de dados é como mostrado abaixo:

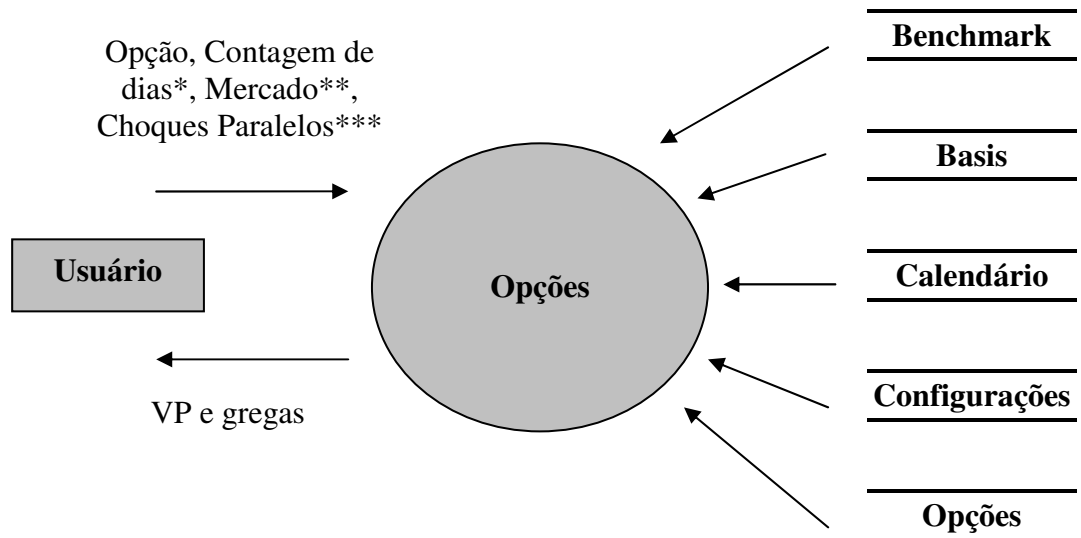


Figura 8 - DFD “Realizar Análise de Risco de Investimentos em Renda Variável”

***Contagem de dias:** envolve os parâmetros data de emissão, data de vencimento, data de fixing, calendário, basis e tipo da opção (*call* ou *put*).

****Mercado:** engloba os parâmetros *strike* (preço de exercício), curva de preço do ativo subjacente, principal do contrato, curva de taxa *forward*, curva de taxa *backward*, superfície de volatilidade e volatilidade implícita.

*****Choques Paralelos:** engloba os choques a serem dados nos parâmetros de cálculo das gregas. O usuário poderá especificar os choques nas gregas delta, rho, theta e vega. O choque para a grega de gamma será o mesmo que para a grega de delta, devido à relação entre estas duas medidas de sensibilidade.

Todos os parâmetros da opção são carregados da base de dados ao se clicar sobre a opção, mas o usuário tem a liberdade de alterar quaisquer destes parâmetros, seja para fins de correção ou para uma avaliação do efeito de determinada alteração.

3.2.2.7 Solicitar a ajuda do sistema

Para solicitar a ajuda do sistema o usuário deve clicar no menu ajuda na tela principal do sistema. Com isso, a tela de ajuda será exibida com duas opções para o usuário: Manual do Usuário e Tutoriais. Clicando em uma destas opções o sistema abre o documento correspondente para a leitura e esclarecimento de dúvidas. Segue abaixo o diagrama de fluxo de dados:

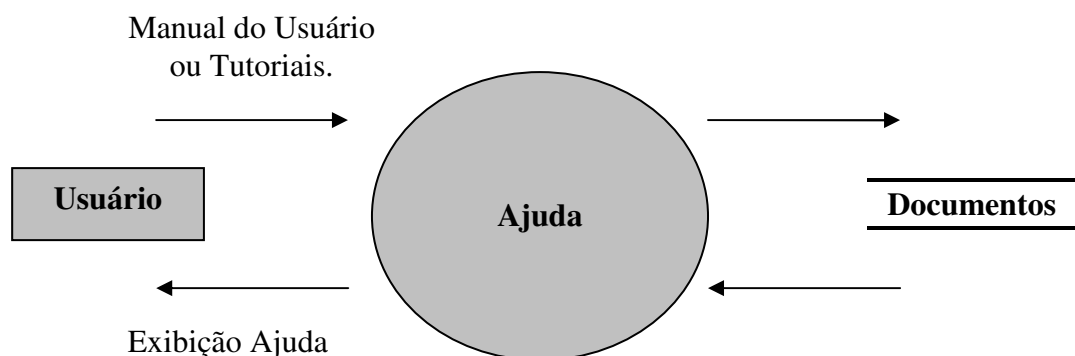


Figura 9 - DFD “Solicitar a ajuda do sistema”

3.2.2.8 Verificar a versão do sistema

Para verificar a versão do sistema, assim como a data de sua liberação, clique no último menu da tela principal, o menu “Sobre”. Será exibida uma caixa de diálogo contendo estas informações.

3.3 Características do sistema

O sistema possui algumas características fundamentais a um software para cálculos em ambientes financeiros:

3.3.1 Amigabilidade

Um sistema de cálculo financeiro deve ser de utilização simples e amigável para que o usuário sinta-se à vontade enquanto estiver utilizando-o. Como as análises são demoradas e detalhadas, foi priorizado o visual e a eficiência para que o usuário consiga operar o sistema sem maiores problemas. O sistema também foi estruturado visando dar o maior retorno possível ao usuário, mostrando às vezes cálculos intermediários para se chegar às soluções, de forma a ganhar a confiança de quem opera o sistema. Exemplos dessa estratégia podem ser vistos na tela de opções, onde as taxas e a volatilidade são mostradas e na tela de análise de risco, na qual o usuário pode visualizar toda a sua série de retornos diários relativos ao ativo escolhido.

3.3.2 Confiabilidade

Esta é uma característica fundamental para um software que faz cálculos financeiros, afinal ele servirá de apoio à tomada de decisões em um ambiente bastante hostil. Por isso, foram feitos testes de cada funcionalidade do sistema através da ferramenta Calc, fornecida no pacote OpenOffice.

3.3.3 Manutenibilidade

O sistema tem essa característica devido a estar sendo bem especificado, e a codificação será feita com comentários para facilitar o entendimento, o que facilitará a correção de erros e a expansão. Além disso, a atualização dos dados históricos de

mercado pode ser feita de uma maneira bem simples (para maiores informações, consulte o Apêndice “Instalação e Atualização de Dados”) e os arquivos dos repositórios XML podem ser facilmente testados através de softwares como o XML Notepad 2007.

3.3.4 Flexibilidade

Como em finanças são muito utilizados o Microsoft Excel e o Calc, em várias telas foi inserido o botão de copiar, para que os *grids* de valores gerados pelo sistema pudessem ser levados até estes softwares e facilitar o processo de testes. Esta estratégia torna o software mais flexível, permitindo uma comunicação com outros softwares.

4. Estrutura do Projeto

Neste capítulo será feita a análise da estrutura do Projeto. Para isso, será mostrada a divisão em módulos que está feita tanto na parte de codificação quanto na parte de dados. Os módulos estão fornecidos de forma resumida. Qualquer dúvida em relação a algum método de cálculo ou em relação à estrutura de dados de algum elemento, consulte o Manual do Usuário que é parte integrante deste documento.

4.1 Visão Geral

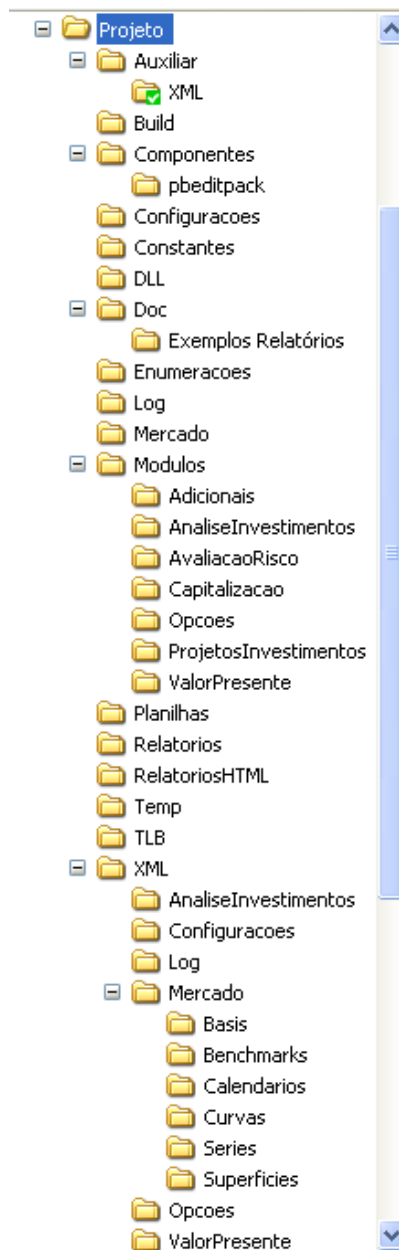


Figura 10 – Estrutura do Projeto.

O projeto está estruturado como mostrado acima. Em seguida serão mostrados os conteúdos de cada parte do sistema, com a finalidade de fornecer uma visão global da estrutura:

- **Auxiliar:** temos as estruturas auxiliares do sistema, que incluem tratamento de datas e de arquivos no formato XML.
- **Build:** Responsável pelo registro das bibliotecas necessárias ao correto funcionamento do sistema. Quando o projeto for rodado pela primeira vez em um computador, antes se deve rodar o arquivo de registro de DLLs para o correto funcionamento do sistema.
- **Componentes:** Armazena os componentes Delphi utilizados pelo sistema.
- **Configurações:** Módulo responsável por tratar das configurações do sistema.
- **Constantes:** Arquivo de constantes do sistema com os *paths* para os repositórios de dados, as *tags* dos elementos XML, entre outras constantes do sistema.
- **DLL:** Pasta com as DLLs do sistema.
- **Doc:** Pasta contendo toda a documentação do Projeto. Aqui se encontram o documento final, o Manual do Usuário e os Tutoriais. Além disso, possui um exemplo de relatório gerado pelo sistema.
- **Enumerações:** Contém as enumerações utilizadas pelo sistema.
- **Log:** Módulo responsável por todo o trabalho de *log* do sistema.
- **Mercado:** Módulo com todo o mecanismo de cálculo da parte de mercado do sistema. Este módulo foi melhor detalhado abaixo.
- **Módulos:** São os módulos com as principais funcionalidades do sistema: Capitalização, Valor Presente, Análise de Investimentos, etc. Este módulo também foi mais bem detalhado abaixo.
- **Planilhas:** Contém as planilhas elaboradas no Calc para validação dos cálculos do sistema. Para maiores informações, consulte o capítulo sobre a estrutura dos testes.
- **Relatórios:** Contém toda a parte responsável pela geração de relatórios HTML.
- **Relatórios HTML:** Armazena os modelos de relatório que o sistema faz uso durante a geração dos relatórios.
- **Temp:** No caso de o usuário querer apenas exibir um relatório e não salvar, o relatório ficará armazenado nesta pasta temporária enquanto durar a exibição.

- **TLB:** Armazena alguns arquivos no formato *.TLB (type library)*. Estes arquivos são representações de interfaces de DLLs utilizados pelo Delphi.
- **XML:** Esta pasta é a raiz do repositório de dados do sistema e é mais bem detalhada adiante.

4.2 Detalhamento

Será feito um detalhamento do que há em cada módulo e como eles se relacionam. Porém, antes de iniciar este detalhamento, será feita uma análise do módulo de Mercado, que está em uma camada mais baixa e serve de base para quase todos os módulos do sistema.

4.2.1 O Módulo de Mercado

O módulo de mercado serve de base para os cálculos do sistema e possui os seguintes elementos: basis, calendários, benchmarks, curvas, séries históricas e superfícies de volatilidade. Para cada um desses elementos há um serviço associado, que é responsável pela carga dos repositórios XML e por manusear estes repositórios a fim de retornar um determinado item gravado no sistema. Agora será feito um detalhamento de cada um desses módulos e suas principais funções.

Obs.: Como os vários serviços associados a cada um dos elementos do sistema possuem finalidades muito semelhantes, será mostrado apenas o *ServicoBasis*, para fins de exemplificação. Os outros serviços recebem, em geral, as mesmas atribuições de carga e busca de dados nos repositórios XML. Também serão omitidos métodos que não sejam essenciais ao entendimento do sistema. Um exemplo seria o método *CarregaElemento*, que existe em todos os elementos do sistema e que carrega um elemento em memória a partir de sua representação XML.

4.2.1.1 Basis : Contagem de Dias

Os basis no sistema servem exclusivamente para cálculo de período, e têm sua estrutura intimamente ligada aos Calendários.

4.2.1.1.1 ServiçoBasis

Este serviço é responsável pela carga e manipulação dos basis no sistema. Suas principais funções são:

1. **CarregaTodosBasis** - Carrega todos os basis do sistema para a memória.
2. **PegaBasisNome (Nome : String) : Pointer** – Esta função retorna um basis dado o seu nome.
3. **PegaNumeroBasis : Integer** – Retorna o número de basis cadastrados no sistema.
4. **PegaNomeBasisIndice (Indice : Integer) : String** – Retorna o nome do basis dado o seu índice no sistema.

4.2.1.1.2 Basis

Os basis são responsáveis pela contagem de dias no sistema. Suas principais funções são:

1. **CalculaPeriodoAnual (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TdateTime** – Calcula o período anual entre as duas datas recebidas. É importante salientar que para o cálculo do período o basis precisa utilizar um outro elemento do módulo de mercado, o calendário, que será detalhado mais à frente.
2. **CalculaPeriodoMensal (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TdateTime** – Calcula o período mensal entre as duas datas recebidas.
3. **CalculaPeriodoSemanal (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TdateTime** - Calcula o período semanal entre as duas datas recebidas.
4. **CalculaDiasUteisPeriodo (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TdateTime** – Calcula o número de dias úteis para o período recebido.

4.2.1.2 Benchmark

Este elemento é central para todo o sistema, sendo responsável pelo carregamento de valores de uma data à outra. O Benchmark engloba outros dois elementos do sistema, a Série Histórica e a Curva. Isto porque é necessário diferenciar dados passados de projeções futuras nas análises. Sua principal função é:

- **CalculaFatorIntervalo (DataMercado, DataInicio, DataFim : TDateTime) : Double** – Esta função calcula o fator para carregamento no intervalo fornecido, utilizando o mercado da DataMercado. Para isso, o elemento benchmark fará uso dos elementos Curva e Série Histórica, que serão especificados adiante.

4.2.1.3 Série Histórica

Este elemento serve como repositório de informações passadas do mercado. As séries podem ser de taxa ou de preço. Suas principais funções são:

1. **CalculaTaxaPeriodo (DataInicio, DataFim : TDateTime) : Double** – Retorna a taxa efetiva para um determinado período.
2. **CalculaFatorPeriodo (DataInicio, DataFim : TDateTime) : Double** – Retorna o fator de carregamento para um determinado período.
3. **GeraListaRetornosDiarios (DataInicio, DataFim : TdateTime)** – Gera uma lista de retornos diários entre as duas datas recebidas. Esta lista servirá como base para a análise estatística desenvolvida na parte de Análise de Risco de Investimentos.
4. **CalculaMediaRetornos (DataInicio, DataFim : TDateTime) : Double** – Calcula a média dos retornos diários calculados entre as duas datas recebidas.
5. **CalculaVarianciaRetornos (DataInicio, DataFim : TDateTime) : Double** – Calcula a variância dos retornos diários entre as duas datas recebidas.
6. **CalculaDesvioPadraoRetornos (DataInicio, DataFim : TDateTime) : Double** – Calcula o desvio-padrão dos retornos diários entre as duas datas recebidas.
7. **CalculaProbabilidadeSucesso (DataInicio, DataFim : TDateTime; PercentualRentabilidade : Double; Media, Variancia : Double) : Double** – Calcula a probabilidade de sucesso (atingir uma rentabilidade maior ou igual à PercentualRentabilidade) de um investimento feito entre as datas recebidas, usando a média e a variância fornecidas.

4.2.1.4 Curva

Uma curva de mercado, assim como a série, pode conter valores de taxas ou de preços. A diferença principal entre curvas e séries históricas é que as curvas representam dados projetados para o futuro. A estrutura se encontra detalhada no Manual do Usuário. Suas funções mais importantes são:

1. **CalculaFatorPeriodo (DataCurva, DataInicio, DataFim : TDateTime) : Double** – Calcula o fator de carregamento para o período recebido. Como a curva varia dia a dia, é necessário especificar a data da curva de mercado.
2. **CalculaTaxaPeriodo (DataCurva, DataInicio, DataFim : TDateTime) : Double** – Calcula a taxa efetiva para o período recebido.
3. **PegaVerticeInterpoladoCurva (DataCurva, DataCotacao : TDateTime) : Double** – Interpola de forma linear um vértice para datas nas quais não exista cotação.

4.2.1.5 Calendários

Os calendários no sistema são simples repositórios nos quais são adicionados os feriados. Entre suas principais funções, encontram-se:

1. **VerificaDiaUtil(Data : TDateTime) : Boolean** – Verifica se a data recebida é dia útil ou não. Para os dias de fim de semana o basis será levado em conta para decidir se o dia é útil ou não..
2. **VerificaFeriado(Data : TDateTime) : Boolean** – Verifica se uma determinada data está cadastrada como feriado para o calendário.
3. **PegaDiaUtilCorrigido(Data : TDateTime) : TdateTime** – Ajusta a data caso ela caia em um feriado, retornando o próximo dia útil a partir da data recebida.

4.2.1.6 Superfícies de volatilidade

Para as opções de ações, existem volatilidades associadas a cada *strike* (Preço de exercício) e a cada data disponível. Logo, tem-se 3 eixos: data, volatilidade e strike. Esta configuração forma uma superfície de volatilidade, utilizada no módulo de opções. Suas principais funções estão descritas abaixo:

1. **PegaVolatilidade (DataSuperficie, DataVolatilidade : TDateTime; Strike : Double) : Double** – Retorna uma volatilidade a partir dos parâmetros recebidos. A data da superfície é a data do mercado e a data da volatilidade é a data para a qual se deseja saber a volatilidade de determinada ação. O strike representa o preço de exercício da opção.

4.2.2 Resumo

Com a estrutura anterior foi possível compor um módulo de mercado que representasse de forma satisfatória vários eventos do mercado financeiro real. As principais atribuições desse módulo, de forma resumida, seriam:

- Fornecer flexibilidade à contagem de dias, permitindo fazer cálculos tanto em ambiente nacional, que usa uma base de 252 dias úteis/ano, como em ambiente internacional, no qual é utilizado o sistema de dias corridos e é considerado um ano de 360 dias.
- Permitir a inclusão dos feriados específicos para os pregões de negociação em cada bolsa existente no sistema.
- Fornecer dados passados de mercado, principalmente para análises estatísticas.
- Permitir que o usuário possa inserir suas projeções e perspectivas de mercado para o futuro.
- Fornecer suporte para carregamento e desconto de valores ao longo do tempo, tanto para datas passadas quanto futuras.
- Fornecer flexibilidade para que se possa cadastrar dados de taxas de juros e de preços.
- Permitir a inserção de superfícies de volatilidades de opções.
- Realizar interpolação linear no caso de valores futuros, para facilitar o trabalho de projeção de dados, tanto no caso de taxas de juros quanto de preços.

Com este módulo de mercado foram criadas várias aplicações para o mercado financeiro, as quais serão descritas a seguir.

Na parte de codificação, o sistema ficou dividido da seguinte forma:

| Nome | Tamanho | Tipo | Data de modificação |
|-----------------------|---------|-------------------|---------------------|
| Adicionais | | Pasta de arquivos | 22/9/2008 23:43 |
| AnaliseInvestimentos | | Pasta de arquivos | 23/9/2008 09:23 |
| AvaliacaoRisco | | Pasta de arquivos | 22/9/2008 23:39 |
| Capitalizacao | | Pasta de arquivos | 17/9/2008 13:00 |
| Opcoes | | Pasta de arquivos | 22/9/2008 22:47 |
| ProjetosInvestimentos | | Pasta de arquivos | 24/5/2008 09:02 |
| ValorPresente | | Pasta de arquivos | 22/9/2008 11:04 |

Figura 11 – Estrutura de organização do código do sistema.

Estes módulos utilizam as funções do módulo de mercado para construir as aplicações do sistema. Será dada uma explicação resumida das atribuições de cada módulo, pois são muito extensos e isso acabaria prejudicando a visão geral do módulo. Para maiores detalhes, consulte o Manual do Usuário.

4.2.3 Adicionais

Este módulo é responsável por fazer a interface de comunicação com o usuário. Ele é responsável pela exibição da ajuda (Manual do Usuário e Tutorias), e também por disponibilizar informações sobre a versão do sistema, bem como sua data de liberação.

4.2.4 Análise de Investimentos

Este módulo é responsável pela análise comparativa de projetos de investimentos. A explicação para os cálculos da tela podem ser obtidas no Manual do Usuário, que é parte integrante deste relatório.

De forma resumida, o que o módulo faz é calcular alguns indicadores (Rentabilidade Simples, Valor Presente entre outros) para cada projeto de uma lista e, através de parâmetros-base inseridos pelo usuário, realiza uma análise comparativa na tentativa de auxiliar o gerente financeiro na escolha do melhor projeto de investimento. Ao final do processo, é gerado um relatório em formato HTML.

4.2.5 Avaliação de Risco

Este módulo tem por objetivo avaliar o risco associado a um determinado investimento em um ativo. O usuário pode escolher até 4 ativos para serem comparados simultaneamente. A análise gera como saída algumas medidas estatísticas e o percentual

de sucesso que se poderia obter com cada ativo escolhido. É um módulo muito útil no que diz respeito à alocação de recursos.

4.2.6 Capitalização

Este módulo é um dos mais básicos do sistema e pode ser utilizado para carregar valores entre datas. Para isso, podem ser utilizados 3 tipos de capitalização (simples, composta e contínua) e pode ser utilizado também um benchmark de mercado.

4.2.6 Opções

Este módulo faz cálculos relativos à precificação e sensibilidades das opções de ações cadastradas no sistema. Para um maior nível de detalhe, consulte o Manual do Usuário e o anexo relativo ao Modelo de Black & Scholes, utilizado para precificar as opções do sistema.

4.2.7 Projetos de Investimentos

Este módulo é responsável pela manipulação dos projetos de investimento, e dá suporte aos cálculos efetuados na tela de análise de investimentos. Para isso, faz uso de vários outros módulos do sistema, entre eles o módulo de Mercado e o de Valor Presente.

4.2.8 Valor Presente

Engloba todo o mecanismo de cálculo básico para se chegar aos indicadores de Rentabilidade Simples, *Pay-back Period*, Valor Presente e Taxa Interna de Retorno. É utilizado como sub-módulo pelo módulo de Projetos de Investimentos.

4.3 Dados

A estrutura de dados do sistema está disposta da seguinte maneira:

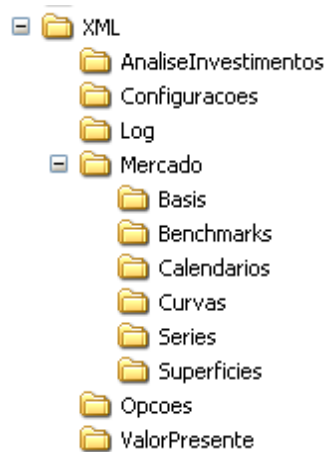


Figura 12 – Estrutura de organização de dados do sistema

A pasta XML representa o repositório raiz do sistema. Todos os arquivos XML estão localizados dentro desta pasta. Nela, os dados estão divididos da seguinte forma:

Análise de Investimento: Nesta pasta encontram-se os arquivos que guardam as estruturas de projetos de investimentos analisados na tela de análise de investimentos.

Configurações: Aqui estão armazenadas as configurações do sistema.

Log: Nesta parte se localiza o arquivo com os *logs* de erro do sistema.

Mercado: Aqui se encontram todos os dados de mercado do sistema, através dos elementos *Basis*, *Benchmarks*, *Calendários*, *Curvas*, *Séries* e *Superfícies*.

Opções: Neste repositório estão localizadas as opções do sistema.

Valor Presente: Aqui se encontram os projetos utilizados pela tela de valor presente.

4.4 Interfaces externas

As interfaces com o usuários serão realizadas em 3 padrões diferentes: arquivos em formato XML, arquivos em formato HTML (no caso da geração de relatórios) e pela própria interface gráfica gerada no Delphi.

A tela inicial apresenta menus em cascata, e as funcionalidades estão bem divididas entre as telas, de forma que o usuário não manuseie telas com poucas informações nem com informações em demasia. Foi buscado um balanço para que o sistema ficasse fácil de se utilizar e, ao mesmo tempo, contento várias funcionalidades matematicamente refinadas.

5. Desenvolvimento do Projeto

Neste capítulo será discutida a inspiração para o projeto bem como os problemas encontrados durante a sua construção e as respectivas soluções encontradas.

5.1 Histórico

Tudo começou quando ingressei como estagiário em uma empresa de software que vende produtos para o mercado financeiro. Nela eu aprendi muito sobre finanças e me mantive envolvido no ambiente financeiro por mais de um ano. Com isso, comecei a tomar gosto por esta área do conhecimento que ainda tem muito campo para se expandir aqui no Brasil.

Depois disso vieram alguns cursos e muita leitura.

A idéia para o projeto final surgiu então naturalmente. Como trabalho na empresa com derivativos e, em especial, com opções, achei que seria um ótimo incentivo para aprender mais sobre o assunto. A partir daí surgiram idéias para incrementar o projeto: a primeira, a de incluir um módulo que fizesse a análise de projetos de investimentos, surgiu durante a leitura do livro “Cálculo Financeiro das Tesourarias”, de José Roberto Securato. Esta idéia, por depender de uma ampla estrutura de mercado, serviu para compor 3 módulos: Capitalização, Valor Presente e Análise de Investimentos.

Criados estes módulos e somando-se a eles um módulo que tratasse das opções de ações, tema que me interessa muito por representar grandes possibilidades de lucros no mercado, já seria suficiente para montar a estrutura de um projeto final. Resolvi adicionar o módulo de Análise de Risco por achar que ele complementa as idéias do projeto e porque boa parte da estrutura a ser utilizada já havia sido desenvolvida para suprir as necessidades dos outros módulos. Assim, o projeto tomou a forma atual, a qual foi desenvolvida com o critério e cuidado que as análises financeiras merecem.

No decorrer do projeto, vários problemas surgiram e significaram importantes momentos de superação e aprendizado. Em seguida, estão listados os principais problemas e a forma com que foram solucionados.

5.2 Dificuldades Encontradas

No decorrer do projeto, foram encontradas dificuldades devido ao uso de uma tecnologia não totalmente conhecida e de pressupostos teóricos ainda não muito bem sedimentados. Abaixo, os principais problemas no decorrer do desenvolvimento do projeto e as soluções encontradas.

5.2.1 Armazenamento

Uma das dúvidas iniciais do projeto seria a forma de armazenar os dados.

Solução:

Como o projeto já é extenso o bastante, foi decidido montar uma estrutura em XML e utilizar uma biblioteca em Delphi para fazer as operações de consulta e inserção no banco de dados. Esta forma de armazenamento facilitou a atualização de dados no sistema, que deve ser feita periodicamente.

5.2.2 Componente

Inexistência de um componente do Delphi específico para exibir números.

Solução: Para contornar o problema, inicialmente tentou-se conseguir uma biblioteca com um componente gráfico específico para exibição de números. Como não há muito desenvolvimento em Delphi gratuito na internet, esta tentativa acabou por falhar. Como solução, foram acrescentadas funções que otimizassem o componente padrão do Delphi para exibição de textos para que ele pudesse ser utilizado na exibição de números. O resultado foi bom e a adaptação do componente se mostrou bastante eficaz.

5.2.3 Salvamento de Arquivo

Conseguir uma janela de diálogo para salvamento de arquivos. Isto seria necessário para que a interface de salvamento de relatórios, na tela de Análise de Investimentos, fosse amigável ao usuário. Este componente seria de fundamental importância, já que esta tela é uma das mais complexas em quantidade de informações e a intenção era reduzir o trabalho do usuário ao operar a tela.

Solução: Após muitas pesquisas na internet descobri que há um componente padrão do Delphi que implementa esta caixa de diálogo.

5.2.4 Caminhos Relativos

A utilização de caminhos relativos para os *paths* dos repositórios ocasionou um problema bastante sutil. Ao utilizar componentes do Delphi com caixas de diálogo de

salvamento e abertura de arquivo, o Delphi alterava o diretório atual fazendo com que os caminhos relativos perdessem o sentido.

Solução: Substituição dos caminhos relativos por caminhos absolutos.

5.2.5 Comunicação com o Excel e Calc

Como muitos dos cálculos em finanças são feitos em Excel, seria de grande importância poder transferir os dados para o Excel para que o usuário alternasse entre os ambientes, aumentando sua produtividade.

Solução: Havia uma biblioteca em Delphi que copiava texto para o *Clipboard*. Depois disso, o problema passou a ser que tipo de texto deveria ser gravado para que a distribuição das colunas e linhas ficasse correta no Excel. Depois de encontrado o caracter que demarca as linhas e colunas, foi necessário procurar sua representação em Delphi para que a transferência entre as aplicações funcionasse.

5.2.6 Cores diferentes para os resultados

Seria interessante exibir os resultados dos cálculos em uma cor diferente para chamar a atenção do usuário.

Solução: No próprio componente Delphi havia uma opção para se alterar a cor; bastou pesquisar um pouco mais sobre o componente.

5.3 Dependência entre os Módulos

Abaixo é mostrada a dependência entre os módulos do sistema. Esta dependência é a nível de dados e de código, ou seja, deve ser satisfeita tanto no repositório XML quanto nos mecanismos de cálculo que integram o sistema.

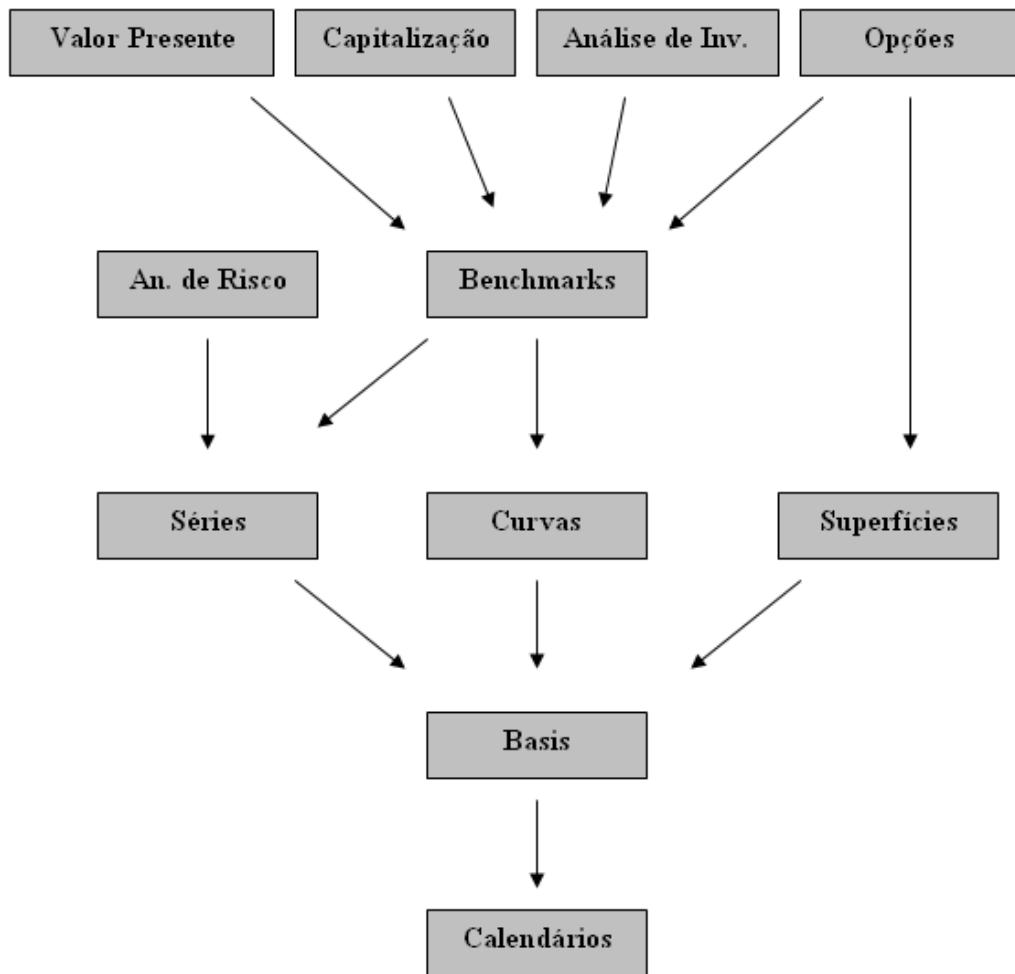


Figura 13 – Dependências entre os principais elementos do sistema.

Pelo diagrama notamos que os calendários são os elementos mais básicos do sistema. Um Basis, para realizar seus cálculos de período, precisa de um calendário com sua lista de feriados. As séries, curvas e superfícies precisam do basis pois utilizam períodos para o cálculo de fatores e taxas entre duas datas. Os benchmarks são compostos por uma curva e uma série, e permitem uma grande flexibilidade no cálculo de fatores de carregamento e desconto, tanto para datas passadas quanto futuras.

O módulo de Análise de Risco faz uso das séries históricas para geração de suas estatísticas. Os módulos de Capitalização, Valor Presente e Análise de Investimentos utilizam a estrutura do benchmark para carregar e descontar valores pelo tempo. Já o módulo de Opções utiliza, além do benchmark, as superfícies de volatilidade para buscar a volatilidade referente ao ativo subjacente de uma opção em uma determinada data.

6. Resultados

Tendo especificado o sistema, passamos à análise dos resultados.

Seguem abaixo os resultados obtidos com a utilização do sistema, separados por módulo.

6.1 Capitalização

O sistema apresentou uma boa representação dos processos de capitalização simples, composta e contínua. No caso da capitalização composta, foram testados os casos utilizando um benchmark do mercado, composto ou não com uma taxa de *spread* fixa e os resultados estiveram dentro do esperado. A tela de capitalização foi testada em cada situação: cálculo de principal, de taxa e de montante, e foi capaz de calcular sempre uma das variáveis a partir das outras. Com isso, o objetivo da tela foi atingido.

6.2 Valor Presente

O sistema é capaz de calcular corretamente os indicadores de Rentabilidade Simples, *Pay-back Period*, Valor Presente e Taxa Interna de Retorno. Além disso, é capaz de gerar uma tabela com os fluxos de caixa do projeto e seus valores presentes na data de análise, o que facilita muito a análise. Com isso, o objetivo do módulo foi satisfeito permitindo uma análise eficiente de um conjunto de fluxos de caixa.

6.3 Análise de Investimentos

Este é o módulo mais complexo do sistema. É capaz de receber uma lista de projetos de investimento, calcular os indicadores citados no item anterior e elaborar tabelas de fluxos de caixa e comparativas dos vários projetos recebidos. Além disso, fornece um relatório em HTML muito útil para fins de validação dos cálculos utilizados. Com isso, este módulo cobre o que se propõe a fazer, que é a análise comparativa de investimentos.

6.4 Avaliação de Risco

Este módulo utiliza dados passados para estimar o percentual de sucesso de um dado investimento no futuro. Para isso, ele deve cobrir toda uma estrutura de cálculo estatístico que foi desenvolvida com base em livros de estatística financeira. Os cálculos

se mostraram dentro do esperado e os cálculos do sistema acompanham o resultado de funções pré-definidas do Calc. Sendo assim, o modelo foi validado e o objetivo alcançado. Para maiores detalhes sobre os testes, ver o capítulo dos testes modulares.

6.5 Opções

Este capítulo é responsável pela parte matemática mais sofisticada do sistema. Os cálculos das gregas de opções devem possuir grande precisão, caso contrário seus valores perdem o sentido. Com isso, os testes foram rigorosos e os resultados podem ser observados no capítulo referente aos testes. O objetivo foi satisfeito.

7. Estrutura dos Testes

Os testes feitos no sistema seguem três linhas gerais: testes de interface, testes na exibição das mensagens de erro e testes de precisão nos cálculos.

7.1 Testes de Interface

Os testes de interface foram conduzidos da seguinte maneira: nas várias telas do sistema foram testadas situações extremas para ver como o sistema se comportaria. Este tipo de teste permitiu a descoberta de vários erros no que diz respeito à proteção contra entradas inválidas.

Também foram testados os casos em que partes da tela devem ser habilitadas ou desabilitadas dependendo de algum comando do usuário. Um exemplo disso pode ser visto na tela de capitalização: quando o usuário seleciona o método de Rentabilidade Simples, as combos de datas e do benchmark são desabilitadas, já que não fariam sentido para o cálculo. Problemas como estes foram exaustivamente testados para que o sistema permitisse a edição apenas dos valores necessários para cada cálculo, evitando assim confusões desnecessárias.

7.2 Testes na exibição das mensagens de erro

Nas situações onde o sistema deveria mostrar mensagens de erro para alertar o usuário foram realizados testes e verificado o comportamento do sistema. Basicamente esta etapa constituiu no teste do sistema de *log* do projeto. Este tipo de teste é de fundamental importância porque é ele que informa ao usuário sobre o que ocorreu de errado durante algum cálculo. Assim, os testes permitiram descobrir algumas falhas na comunicação com o usuário. Algumas mensagens foram melhoradas, outras agrupadas para passar uma idéia mais clara do problema encontrado.

7.3 Testes de precisão nos cálculos

Nesta etapa dos testes foi utilizado o software OpenOffice Calc para fazer as validações necessárias. Daremos uma maior atenção a este módulo porque é ele o principal responsável pela credibilidade do sistema, já que, em se tratando de finanças, a precisão é um quesito fundamental.

A estrutura destes testes será dividida de acordo com os módulos do sistema. As planilhas mencionadas no decorrer dos testes podem ser encontradas em *C:\Projeto\Planilhas*.

7.3.1 Capitalização

Nesta tela foram feitos testes para cada situação possível de capitalização. A replicação dos testes encontra-se na planilha Capitalização.

Capitalização simples: Foram testados os cálculos de taxa, de principal e de montante para diferentes bases e calendários. Com isso, foi possível verificar o correto funcionamento do mecanismo de basis do sistema, assim como o efeito de um feriado no calendário (para isso foram testados dois calendários diferentes com o mesmo basis para mostrar que o resultado da capitalização é diferente). Os cálculos planejados encontram-se na aba “Capitalização Simples”.

Capitalização Composta sem *Benchmark*: O análogo ao item anterior para capitalização composta também foi testado e pode ser encontrado na aba “Cap Comp sem Bench” da planilha de capitalização.

Capitalização Contínua: O teste também foi realizado para capitalização contínua e pode ser acessado na aba “Capitalização Contínua” da planilha.

Capitalização composta com *Benchmark* (datas passadas): O teste com a capitalização composta utilizando um benchmark de mercado foi feito para datas passadas utilizando séries de taxa e preço. No caso da série de taxa, utilizou-se o benchmark de CDI, cujo *accrual* de juros encontra-se calculado na aba “Benchmark CDF”. No caso da série de preço, foi utilizada a série histórica de GGBR4 (Gerdau PN) para carregamento do valor. Os cálculos encontram-se na aba “Cap Comp com Bench”.

Capitalização Composta (Composição entre curva e série histórica): Neste teste foi utilizado um intervalo de tempo que abrange valores passados e futuros, em relação à data de mercado. Com isso, o *benchmark* deverá ser desmembrado em dois fatores: um proveniente da série histórica e outro proveniente da curva de mercado. Os cálculos encontram-se na aba “Cap Comp Composição”.

Capitalização Composta (Composição entre *benchmark* e taxa fixa): Neste caso, teremos o efeito conjunto de um benchmark de mercado e uma taxa fixa, conhecida como *spread*. Os valores encontram-se na aba “Cap Comp Composição”.

Capitalização Composta em datas futuras: Este teste utilizará apenas as curvas de mercado com valores futuros projetados. Foi testada também a interpolação linear nas curvas. O cálculo foi realizado para curvas de preço e taxa e encontra-se descrito na aba “Capitalização Composta Curvas”.

Ainda nesta planilha há uma aba contendo os calendários do sistema para o cálculo dos períodos.

Com isso, fica validado o carregamento e desconto de valores ao longo do tempo, que será amplamente utilizado em outros módulos.

7.3.2 Valor Presente

Foram feitos testes de carregamento a valor presente e os resultados estão mostrados na planilha “Valor Presente”, localizada no diretório de planilhas do sistema.

Valor Presente e Futuro a partir de taxa fixa: Neste teste foram realizados carregamentos de valores tanto para o passado quanto para o futuro através de uma taxa fixa. O resultado usado para comparação foi copiado do grid da tela de Valor Presente e levado ao Calc para comparação com os valores calculados. Na planilha há uma coluna de erros que representa a diferença entre os valores do sistema e os valores calculados. O sistema consegue zerar o erro em todos os dois casos: tanto no carregamento como no desconto de valores.

Valor Presente e Futuro a partir de um benchmark: Neste teste foram realizados carregamentos de valores tanto para o passado quanto para o futuro através de um benchmark. O resultado usado para comparação foi copiado do grid da tela de Valor Presente e levado ao Calc para comparação com os valores calculados. Na planilha há uma coluna de erros que representa a diferença entre os valores do sistema e os valores calculados. O sistema consegue zerar o erro em todos os dois casos: tanto no carregamento como no desconto de valores. Cabe ressaltar aqui que, como temos carregamento e desconto de fluxos, o sistema utiliza tanto a série histórica quanto a curva de preço.

7.3.3 Opções

Para o módulo de opções, foram feitos testes de precificação segundo o modelo de Black & Scholes e também de todas as gregas do sistema.

Preço: O teste do mecanismo de cálculo de preço da opção se encontra na aba “Opção Precificação”, e mostra como é feito o cálculo do preço de uma opção de ação segundo o modelo de Black & Scholes. Para executar o cálculo segundo este modelo, é necessária uma função que calcule o valor da integração sob uma gaussiana. O valor desta integral pode ser computado de forma numérica, como mostrado na função na linguagem VBA abaixo:

```
Public Function CDF(x As Double)
Dim P, COEF1, COEF2, COEF3, COEF4, COEF5, Z, Serie As Double
P = 0.2316419
COEF1 = 0.31938153
COEF2 = -0.356563782
COEF3 = 1.781477937
COEF4 = -1.821255978
COEF5 = 1.330274429
Z = 1 / (1 + P * Abs(x))

Serie = COEF1 * Z + COEF2 * Z * Z + COEF3 * Z * Z * Z + COEF4 * Z * Z * Z * Z + COEF5 *
Z * Z * Z * Z * Z
Normal = 1 - (1 / (2 * 3.14159265358979) ^ (0.5)) * Exp(-x * x / 2) * Serie

If (x < 0) Then
Normal = 1 - Normal
End If
CDF = Normal
End Function
```

Esta função foi testada para diferentes entradas e seus resultados comparados com uma tabela de resultados para a integral da gaussiana. Os valores possuem excelente precisão, em torno da sexta casa decimal.

O prêmio da opção gerado pelo sistema foi corroborado pelos cálculos da planilha com precisão até a oitava casa decimal, que é a precisão máxima permitida no sistema.

Deltas: No caso dos deltas, foi feito o teste dando o choque paralelo na taxa e subtraindo os preços com e sem choque. Os valores da planilha são iguais aos do sistema. Este teste pode ser visto na aba “Deltas”.

Vegas: Neste caso, subtraímos o valor do prêmio da opção calculado com a volatilidade chocada do prêmio sem nenhum choque. O resultado está de acordo com o mostrado no sistema. O teste se encontra na aba “Vegas”.

Gammas: Para os gammas, que representam a segunda derivada do preço da opção em relação ao ativo subjacente, subtrai-se o preço com dois choques no ativo subjacente do preço com um choque apenas. O resultado está de acordo com o apresentado pelo sistema e pode ser visto na aba “Gammas”.

Rhos: Para os Rhos o cálculo é análogo ao delta, mas com o choque dado na taxa forward continuamente composta. O resultado está idêntico ao do sistema, o que pode ser confirmado na aba “Rhos”.

Thetas: Por fim, cuidamos do choque no período até o vencimento da opção. Para isso, o usuário informa a quantidade de dias que quer chocar e o sistema converte este valor para a base anual, segundo o basis da opção. Os resultados validam o mecanismo de cálculo do sistema e podem ser vistos na aba “Thetas”.

8. Conclusão

Neste capítulo será concluído o relatório deste projeto final. Serão discutidas as tecnologias utilizadas, a metodologia empregada na construção do projeto, os desafios encontrados e será feita uma avaliação das ferramentas utilizadas.

Também será feito um balanço do que foi aprendido e mencionadas propostas de futuras melhorias no projeto.

8.1 Avaliação das Tecnologias

As tecnologias escolhidas para o projeto já eram do uso do autor há algum tempo, já que é seu instrumento de trabalho. Por isso foram escolhidas, para que o foco do trabalho não ficasse em cima de detalhes sobre as tecnologias utilizadas e sim sobre os cálculos financeiros e a organização do projeto. Assim, não foi demandado muito tempo no estudo de novas tecnologias, o que permitiu que o projeto pudesse ficar rico em detalhes de cálculos financeiros. Esta era a intenção desde o início do projeto, e foi uma estratégia muito bem sucedida.

8.2. Avaliação da metodologia

Como o sistema é um software, iremos avaliar a metodologia de projeto usada para o desenvolvimento do mesmo, que foi a estruturada.

Por se tratar de uma metodologia facilmente compreensível e capaz de atender aos objetivos desse sistema, ela foi adequada e proporcionou o desenvolvimento do projeto de uma forma mais objetiva e organizada.

Através da mesma foi possível prever os pontos mais críticos do projeto, assim como planejar as melhores soluções. Foi possível também identificar alguns problemas que puderam ser novamente analisados e corrigidos, como os já tratados no capítulo anterior.

8.3. Avaliação das ferramentas

O ferramental utilizado se mostrou muito útil e indispensável ao correto andamento do projeto. Na parte de codificação, o ambiente do Delphi se mostrou muito prático, especialmente no que diz respeito às interfaces. Na parte de gerenciamento de alterações no projeto, foi de fundamental importância a utilização do software WinMerge, que compara arquivos. Este software permitiu detectar alguns problemas e a fazer a junção de duas versões de datas diferentes.

Na parte de cálculo, foi de imensa importância o uso do software OpenOffice Calc, sem o qual provavelmente este projeto não seria tão confiável. Sua contribuição na parte de testes foi essencial para agilizar o processo.

Na parte de dados, os softwares Notepad ++ e XML Notepad 2007 foram muito importantes na edição e validação dos repositórios XML do sistema.

Enfim, o aparato de ferramentas utilizado permitiu um maior controle sobre o desenvolvimento, agilizando-o, e facilitou a validação de dados e dos mecanismos de cálculo do sistema.

8.4 Aprendizado

O aprendizado adquirido ao se construir um sistema como este é enorme. Para isso, foi preciso dominar uma série de ferramentas, já descritas acima. Foi preciso também dominar os formatos de arquivos XML e HTML e a linguagem de programação Delphi. Apesar de tudo isso, o principal desafio do projeto foi teórico.

Na parte teórica, o projeto demandou relativamente muito mais tempo de estudo. Foi preciso descobrir os detalhes do mercado na realização dos cálculos financeiros e os detalhes da Matemática Financeira para compor a parte de Capitalização. Na parte de valor presente, foi preciso estudar a fundo a análise de investimentos e o cálculo dos indicadores mais utilizados. Para a parte de análise de investimentos, basicamente foi utilizado o ferramental desenvolvido nas duas primeiras etapas, mas o projeto foi grande e detalhado. No que diz respeito a análise de risco, foi preciso conhecer os conceitos de média, variância e desvio-padrão. Também foi preciso saber como o mercado calcula os retornos diários para uma série de cotações. Para chegar a uma análise estatística do

risco, foi necessário revisar os conceitos da distribuição normal e pesquisar um algoritmo que fizesse numericamente a integral sobre esta curva. Um outro problema foi normalizar os valores estatísticos obtidos, já que a função que retorna a área sob a normal trabalha com uma função normalizada. A parte mais desafiadora, entretanto, foi construir um modelo de opções que calculasse um preço justo e valores das gregas compatíveis com o mundo real. Para isso, foi estudada a teoria de precificação de opções segundo Black & Scholes, assim como calcular as sensibilidades (gregas) segundo este modelo. Este cálculo também envolveu a integração da curva normal, que pôde ser aproveitada do outro módulo.

Depois de tanta teoria sobre o mercado financeiro, surge um resultado sólido e bem estruturado, fruto da junção da escolha correta das ferramentas, do estudo detalhado da teoria, da montagem de um sistema de testes rigoroso e de muito tempo dedicado ao projeto.

8.5 Propostas Futuras

Como o projeto é rico em idéias e bastante diversificado, surgem muitas idéias para futuras melhorias. Entre elas, pode-se citar:

1. **Basis:** Adicionar outros basis ao sistema, como o que conta um ano de 365 dias (e 366 dias, caso bissexto), entre outros. O sistema possui apenas o básico para se trabalhar com os mercados brasileiro e americano, mas tem estrutura fácil de ser ampliada com novos basis.
2. **Dados:** Uma ferramenta interessante para o sistema seria a atualização automática de dados. Um sistema auxiliar se conectaria à internet e coletaria dados diariamente, eliminando a necessidade de atualização manual do sistema. Esta ferramenta seria muito importante, já que eliminaria a interferência do usuário na base de dados (evitando assim erros acidentais).
3. **Opções:** Ampliar o leque dos tipos de opções, incluindo opções de DI, IDI e commodities negociadas na BM&F. Para isso, seria necessário desenvolver um modelo à parte para cada tipo de opção já que cada uma possui a sua particularidade e seu modelo de cálculo.

4. **Estatística:** Acrescentar mais ferramentas estatísticas para análise de risco, com suporte à regressão linear, cálculo de matrizes de correlação e variância, entre outras ferramentas utilizadas no cotidiano do mercado.
5. **Testes:** Implementar um sistema automático de testes que detectasse problemas o mais cedo possível. Seria interessante um teste que gerasse uma versão do sistema diariamente e testasse seus cálculos. Caso houvesse alguma divergência em relação aos valores esperados, um alerta seria enviado ao desenvolvedor para que ele se encarregasse de resolver o problema.

São estas as principais modificações propostas. Com estas alterações o sistema ficaria mais completo e confiável, além de mais robusto.

Foi uma experiência muito prazerosa desenvolver este projeto e o resultado mostra que todo o esforço valeu a pena.

9. Bibliografia

- Assaf Neto, Alexandre, “Mercado Financeiro”, 8ª Edição, Editora Atlas, 2008.
- Ross, Stephen A., “Princípios de Administração Financeira”, Stephen A. Ross, Randolph W. Westerfield, Bradford D. Jordan, 2 ed. São Paulo, Atlas, 2008.
- Securato, José Roberto, “Cálculo Financeiro das Tesourarias – Bancos e Empresas”, São Paulo, Saint Paul, 2005.
- Silva Neto, Lauro de Araújo, “Opções: do tradicional ao exótico”, 2 ed., São Paulo, Atlas, 2008.
- Bodie, Zvi: Fundamentos de Investimentos / Zvi Bodie, Alex Kane e Alan J. Marcus; trad. Robert Brian Taylor. – 3.ed. – Porto Alegre: Bookman, 2000.
- Mercado Financeiro: conceitos, cálculo e análise de investimentos / coordenação José Roberto Securato, José Cláudio Securato. – 2. ed. Ver. E ampl. – São Paulo : Saint Paul Editora, 2007.
- Options, Futures & Other Derivatives / John Hull – 6 SOL Ed. – Ed. Prentice Hall
- Aprenda a Investir em Ações e a Operar na Bolsa Via Internet Carlos A. H. Brum – 1ª Ed. – Editora Ciência Moderna.

Apêndice 1: O Modelo de Black & Scholes

O desenvolvimento das teorias de precificação de ativos tem uma história longa e ilustre. Já em 1900, Bachelier ensaiava o uso da estatística clássica na área. Seu trabalho foi retomado por Sprenkle e Boness entre os anos de 1960 e 1963 e aperfeiçoado por Samuelson em 1965.

Foi somente em 1973, no entanto, que Fischer Black e Myron Scholes publicaram a primeira solução para a fórmula de equilíbrio geral na avaliação do prêmio de opções. Este trabalho pioneiro deu início a uma cooperação estreita entre traders e acadêmicos, que gerou as teorias modernas de administração de ativos. Black e Scholes usaram um instrumental matemático sofisticado para a construção de sua fórmula, cuja exposição em detalhes excede o propósito deste texto.

Função Normal

A função considera intervalos infinitesimais de tempo (ao contrário do modelo binomial, que utiliza valores discretos) para a formação de preços. Para tanto, o modelo utiliza a distribuição normal (em vez da binomial) para calcular a probabilidade de o ativo objeto atingir determinado nível de preço.

De fato, a distribuição binomial aproxima-se da normal para n tendendo a infinito, e o modelo binomial converge para o de Black & Scholes.

A fórmula para a distribuição normal é:

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).$$

onde:

- μ = Média
- σ = Desvio-padrão
- $\pi = 3.14159265\dots$
- $e = 2,718228183$

Para efeito de cálculo de probabilidades, usa-se a normal reduzida, que se obtém admitindo-se uma função normal de desvio-padrão 1 e média 0, conforme a seguinte fórmula:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

Podemos interpretar o significado do desvio-padrão com o auxílio do gráfico a seguir:

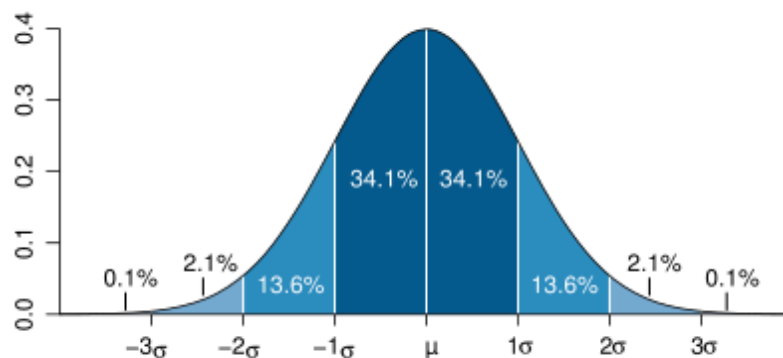


Figura 14 – A distribuição normal e o cálculo de probabilidades.

Afastando-nos da média mais ou menos um desvio-padrão (um desvio para a direita e um desvio para a esquerda), englobaremos cerca de 68,3 das ocorrências. Afastando-nos da média mais ou menos dois desvios-padrão, cobriremos cerca de 95,4% e assim por diante, como se pode deduzir do gráfico.

Convém ainda, antes de explicitarmos as fórmulas do Modelo de Black & Scholes, falar da distribuição *log* normal, que é a usada no modelo, a fim de restringir as oscilações negativas de preço.

A *log* normal é dada pela função:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right]$$

As formas gráficas que esta função pode assumir encontra-se no gráfico abaixo:

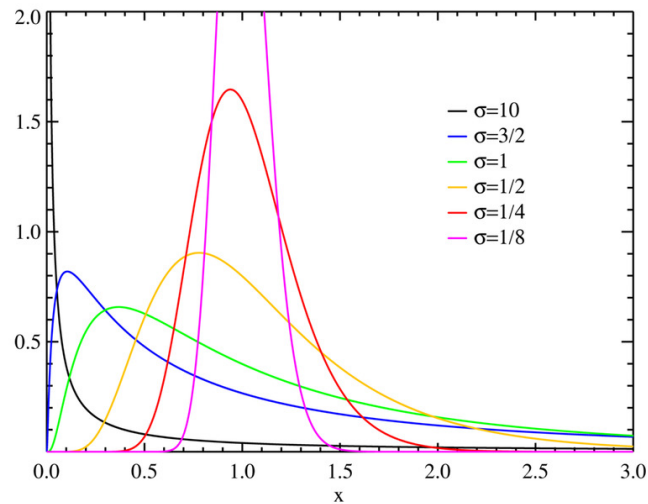


Figura 15 – A distribuição *log normal*.

A fórmula de Black & Scholes para a avaliação do prêmio de opções do tipo europeu sobre ativos que não distribuam dividendos é:

Para a *call*:

$$C = SN(h) - Ee^{-rt} N(h - v\sqrt{t})$$

Para a *put*:

$$P = SN(-h) + Ee^{-rt} N(v\sqrt{t} - h)$$

onde:

$$h = \frac{\ln\left(\frac{S}{Ee^{-rt}}\right)}{v\sqrt{t}} + \frac{v\sqrt{t}}{2}$$

C = Valor teórico de uma *call*
P = Valor teórico de uma *put*
S = Preço da mercadoria, ou ativo financeiro, a vista
E = Preço de exercício da opção
t = Tempo até o vencimento
v = Volatilidade expressa na forma decimal
r = Taxa de retorno "sem risco" (capitalização contínua)
e = Base dos logaritmos naturais = 2,718282.....
ln = Logaritmo natural
N(x) = Função cumulativa normal

Figura 16 - Retirada do livro “Opções: do tradicional ao exótico”, de Lauro de Araújo Silva Neto, mostra o cálculo do preço de opções sobre ações.

Cálculo da função cumulativa normal, $N(x)$

Como descrito, um dos fatores chave para a fórmula de Black & Scholes é a probabilidade associada à distribuição normal. A normal de x poderá ser encontrada em tabelas estatísticas ou calculada em máquinas e planilhas que possuam a função pré-programada. Geralmente, o valor da normal de um número é calculado com base no seguinte desenvolvimento polinomial:

$$N(x) \approx 1 - (1/\sqrt{2\pi}) e^{-\frac{x^2}{2}} (b_1K + b_2K^2 + b_3K^3 + b_4K^4 + b_5K^5)$$

onde:

$$a = 0,2316419$$

$$b_1 = 0,31938153$$

$$b_2 = -0,356563782$$

$$b_3 = 1,781477937$$

$$b_4 = -1,821255978$$

$$b_5 = 1,330274429$$

$$k = 1/(1+ ax)$$

O cálculo acima é válido para x maior do que zero. Quando x for menor que zero, calcula-se a normal como se x fosse positivo e do resultado subtrai-se 1. Para $x = 0$, $N(x) = 50\%$.

O polinômio acima, considerando até b_5 , dará uma aproximação de $N(x)$ de até seis casas decimais.

Cabe ressaltar que os valores de r e v têm que ser congruentes com a unidade de t . Assim, se estivermos trabalhando com volatilidade anual, a taxa de juros usada no modelo também deverá ser anual e o tempo para o vencimento expresso em frações de ano.

Apêndice 2: Estruturas do Sistema

Para dar uma idéia mais ampla sobre o sistema e tornar este documento mais completo, seguem abaixo os atributos e funções dos principais elementos do sistema. Não estão incluídas as especificações das telas do sistema nem os serviços que manuseiam os dados nos repositórios XML. Para ter acesso a este conteúdo pode-se consultar a documentação eletrônica do projeto.

- **Basis**

type

```
TBasis = class (TObject)
```

private

```
Codigo : String;
```

```
Nome : String;
```

```
NDUAno : Integer;
```

```
NDUMes : Integer;
```

```
NDUSemana : Integer;
```

```
procedure InicializaLog;
```

public

```
constructor Create (oNome : String; oNDUAno, oNDUMes, oNDUSemana : Integer);
```

```
function PegaNome : String;
```

```
procedure CarregaElemento (Elemento : IXMLElement);
```

```
function CalculaPeriodoAnual (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TDateTime;
```

```
function CalculaPeriodoMensal (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TDateTime;
```

```
function CalculaPeriodoSemanal (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TDateTime;
```

```
function CalculaDiasUteisPeriodo (const Calendario : TCalendario; DataInicio, DataFim : TDateTime) : TDateTime;
```

```
function PegaNDUAno : Integer;
```

```
function PegaNDUMes : Integer;
```

```
function PegaNDUSemana : Integer;
```

end;

- **Benchmark**

type

```

    TBenchmark = class (TObject)
private
    Codigo    : String;
    Nome      : String;
    CodigoSerie : String;
    CodigoCurva : String;

    procedure InicializaServicosAuxiliares;
    function  PegaCurvaBenchmark : TCurva;
    function  PegaSerieBenchmark : TSerieHistorica;
protected
    public
    constructor Create (oNome : String);
    function  PegaNome : String;
    function  PegaCodigoSerie : String;
    function  PegaCodigoCurva : String;
    procedure CarregaElemento(Elemento : IXMLElement);
    function  CalculaFatorIntervalo (DataMercado, DataInicio, DataFim : TDateTime) : Double;
end;

```

- **Calendário**

```

type
    TCalendario = class (TObject)
private
    Codigo : String;
    Nome   : String;
    Pais   : String;
    Bolsa  : String;
    { Lista de Datas em formato string }
    ListaFeriados : TStringList;
    function PegaNumeroFeriados : Integer;
    procedure InicializaLog;
protected
    public
    constructor Create;
    function  PegaNome : String;
    function  PegaPais : String;
    function  PegaBolsa : String;
    procedure CarregaElemento(Elemento : IXMLElement);

```

```

function VerificaDiaUtil(Data : TDateTime) : Boolean;
function VerificaFeriado(Data : TDateTime) : Boolean;
function PegaDiaUtilCorrigido(Data : TDateTime) : TDateTime;
end;

```

- **Configurações**

```

type
TConfiguracoes = class (TObject)
private
    Documento    : IXMLDocument;
public
    DataSistema  : TDateTime;
    DataAtual    : TDateTime;
    NomeCalendario : String;
    NomeBasis    : String;
    NomeBenchmark : String;
    UsarDataAtual : Boolean;
    CodigoSerie  : String;
    DataInicio   : TDateTime;
    DataFim      : TDateTime;
    { Configurações da Tela de Análise de Investimentos }
    NomeProjetos : String;
    procedure CarregaDadosSistema;
    procedure SalvaDadosSistema;
    function PegaDataSistema : TDateTime;
    function PegaDataAtual : TDateTime;
    function PegaCalendarioSistema : String;
    function PegaBasisSistema : String;
    function PegaBenchmarkSistema : String;
    function PegaNomeProjetos : String;
    constructor Create;
end;

```

- **Curva**

```

(*
* Record que representa uma cotação da curva
*)

```

```

type

```

```

TCotacao = record
    Data : TDateTime;
    Valor : Double;
end;

(*
* Classe que representa uma lista de cotações em uma determinada data.
*)

type
TListaCotacoes = class (TObject)
public
    ListaValores : Array of TCotacao;
    Data      : Double;
    Constructor Create;
end;

(*
* Classe que representa uma curva com valores de data e cotacao.
*)

type
TCurva = class (TObject)
private
    Codigo      : String;
    Nome        : String;
    NomeCalendario : String;
    NomeBasis   : String;
    NaturezaCurva : Integer;
    { Lista de Valores }
    ListasCotacoes : TList;
    function PegaNumeroCotacoesLista(ListaCotacoes : TListaCotacoes) : Integer;
private
    procedure CarregaListaCotacoes (Elemento : IXMLElement);
    procedure InsereCotacao (ListaCotacoes : TListaCotacoes; Cotacao : Double; Data :
TDateTime);
    procedure InsereListaCotacoes (ListaCotacoes : TListaCotacoes);
    function VerificaExistenciaCotacao(ListaCotacoes: TListaCotacoes; DataCotacao: TDateTime):
Boolean;
    function PegaDataCorrigida (Data : TDateTime) : TDateTime;

```

```

    procedure InicializaServicosAuxiliares;
    function PegaCotacaoData (ListaCotacoes : TListaCotacoes; DataCotacao : TDateTime) :
Double;
    function PegaProximaCotacaoData(DataCurva, DataDesejada : TDateTime; out DataCotacao :
TDateTime; out ValorCotacao : Double) : Boolean;
    function PegaCotacaoAnteriorData(DataCurva, DataDesejada : TDateTime; out DataCotacao :
TDateTime; out ValorCotacao : Double) : Boolean;
    function PegaBasis : TBasis;
    function PegaCalendario : TCalendario;
    function PegaListaCotacoesData(DataCurva : TDateTime) : TListaCotacoes;
protected
public
    constructor Create;
    function PegaCodigo: String;
    function PegaNome : String;
    function PegaNaturezaCurva : Integer;
    procedure CarregaElemento(Elemento : IXMLElement);
    function CalculaFatorPeriodo (DataCurva, DataInicio, DataFim : TDateTime) : Double;
    function CalculaTaxaPeriodo (DataCurva, DataInicio, DataFim : TDateTime) : Double;
    function CalculaFatorPeriodoCurvaTaxa (DataCurva, DataInicio, DataFim : TDateTime) :
Double;
    function CalculaFatorPeriodoCurvaPreco (DataCurva, DataInicio, DataFim : TDateTime) :
Double;
    function CalculaTaxaPeriodoCurvaTaxa (DataCurva, DataInicio, DataFim : TDateTime) :
Double;
    function CalculaTaxaPeriodoCurvaPreco (DataCurva, DataInicio, DataFim : TDateTime) :
Double;
    function PegaVerticeInterpoladoCurva (DataCurva, DataCotacao : TDateTime) : Double;
    function PegaPeriodoEntreDatas (DataInicio, DataFim: TDateTime) : Double;
end;

```

- **Fluxos de Caixa**

```

type
    TFluxo = record
        Data : TDateTime;
        Valor : Double;
    end;

```

```

type

```

```

TFluxosCaixa = class (TObject)
private
    Nome : String;
    LimiteSuperior : Double;
    LimiteInferior : Double;
private
    procedure InsereFluxoCaixa (Valor : Double; Data : TDateTime);
    procedure InicializaServicos;
    function PegaBasis : TBasis;
    function PegaCalendario : TCalendario;
protected
public
    NomeBenchmark : String;
    NomeCalendario : String;
    NomeBasis : String;
    DataAnalise : TDateTime;
    TaxaFixa : Double;
    { Lista de fluxos de caixa }
    ListaFluxos : Array of TFluxo;
    constructor Create;
    function PegaNome : String;
    procedure CarregaElemento(Elemento : IXMLElement);
    function PegaNumeroFluxos : Integer;
    function CalculaValorPresenteTaxaFixa (TaxaFixa : Double) : Double;
    function CalculaValorPresente : Double;
    function CalculaTIR : Double;
    function CalculaRentabilidadeSimples : Double;
    function CalculaPayBackPeriod : Double;
    function CalculaFatorFixo(DataFluxo : TDateTime; Taxa : Double) : Double;
    function GeraNovaTaxa(TaxaAtual : Double; TaxaProcurada : Integer) : Double;
    function PegaDataPrimeiroFluxo : TDateTime;
    function CalculaValorPresenteFluxoIndice (Indice : Integer) : Double;
    function PegaDataFluxoIndice (Indice : Integer) : TDateTime;
    function PegaValorFluxoIndice (Indice : Integer) : Double;
end;

```

- **Gerador de Relatórios HTML**

```

type
TGeradorRelatorioHTML = class (TObject)

```

```

private
    DataGeracao : TDateTime;
    Arquivo : TStringList;
    procedure InicializaServicos;
    procedure CarregaArquivoModelo (PathModelo : String);
    function CriaTabelaDadosFluxos (Projetos : TProjetosInvestimento; FluxosCaixa :
TFluxosCaixa) : String;
    function GeraLinhaTituloTabela (NomeProjeto : String) : String;
    function GeraLinhaCabecalhoTabela : String;
    function GeraCorpoResultadosFluxos(FluxosCaixa : TFluxosCaixa) : String;
    function GeraLinhaResultadoFluxo(Data : TDateTime; Valor, VP, Acumulado, AcumuladoVP :
Double) : String;
    function PegaStringHTMLModelo : String;
    function PegaStringFinalTabela : String;
    function GeraLabelCalendario(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelBasis(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelBenchmark(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelMetodosAnalise(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelPercentualRS(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelPercentualVP(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelTMA(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelPBP(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLabelNomeProjeto(Projetos : TProjetosInvestimento; FluxosCaixa :
TFluxosCaixa) : String;
    function GeraTabelasFluxosCaixa(ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraTabelaComparativa (ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLinhaTabelaComparativa (ProjetosInvestimento : TProjetosInvestimento;
IndiceProjeto : Integer) : String;
    function GeraTabelaComparativaFinal (ProjetosInvestimento : TProjetosInvestimento) : String;
    function GeraLinhaTabelaComparativaFinal (ProjetosInvestimento : TProjetosInvestimento;
IndiceProjeto : Integer) : String;
protected
public
    constructor Create;
    function GeraRelatorioHTML(ProjetosInvestimento : TProjetosInvestimento) : Boolean;
    function ConverteStringParaPChar (Const Str : String) : PChar;
end;

```

- **Log**

(*

* Record que representa uma mensagem do *log*

*)

type

```
TMensagem = record
    Modulo : String;
    Mensagem : String;
```

end;

type

```
TLog = class (TObject)
```

private

```
Data : TDateTime;
{ Lista de Mensagens }
ListaMensagens : Array of TMensagem;
```

protected

public

```
constructor Create (oData : TDateTime);
procedure CarregaElemento(Elemento : IXMLElement);
function CriaElemento : IXMLElement;
procedure AdicionaMensagem (Modulo, Mensagem : String);
function PegaData : TDateTime;
function PegaNumeroMensagens : Integer;
procedure PegaMensagemIndice (Indice : Integer; out Modulo, Mensagem : String);
```

end;

- **Opções**

type

```
TParametrosAnaliticos = Class (TObject)
```

private

```
PrecoSubjacente : Double;
TaxaForwardCC : Double;
TaxaBackwardCC : Double;
PeriodoVencimento : Double;
PeriodoFixing : Double;
Volatilidade : Double;
```



```

end;

(*
* Classe que representa uma Opção no sistema.
*)

type
    TOPcao = class (TObject)
    private
        ParametrosAnaliticos : TParametrosAnaliticos;
        ParametrosGerados : Boolean;
        procedure InicializaServicosAuxiliares;
        function PegaBasis : TBasis;
        function PegaCalendario : TCalendario;
        function Normal(X: Double): Double;
        function PegaCurvaForward : TCurva;
        function PegaCurvaBackward : TCurva;
        function PegaCurvaPrecoSubjacente : TCurva;
        function PegaSuperficieVolatilidade : TSuperficieVolatilidade;
        procedure GeraParametrosAnaliticos (DataCalculo : TDateTime);
    protected
    public
       Codigo: String;
        Nome : String;
        NomeCalendario: String;
        NomeBasis : String;
        DataEmissao: TDateTime;
        DataVencimento: TDateTime;
        DataFixing: TDateTime;
        Principal: Double;
        Strike: Double;
        CodigoCurvaForward: String;
        CodigoCurvaBackward: String;
        CodigoCurvaPrecoSubjacente : String;
        CodigoSuperficieVolatilidade : String;
        VolatilidadeImplicita: Double;
        ChoqueDeltaEGamma: Double;
        ChoqueVega: Double;
        ChoqueRho: Double;
        ChoqueTheta : Double;

```

```

IsCall : Boolean;
constructor Create;
function  PegaCodigo : String;
function  PegaNome  : String;
procedure CarregaElemento(Elemento : IXMLElement);
function  ConverteChoqueAnual (Choque : Double) : Double;

{ Analíticos de opções }
function  CalculaPreco(DataCalculo : TDateTime) : Double;
function  CalculaPrecoUnitario(DataCalculo : TDateTime) : Double;
function  CalculaPrecoBlackScholes (Parametros : TParametrosAnaliticos) : Double;
function  CalculaVolatilidade (DataCalculo : TDateTime) : Double;
function  CalculaDeltaNominal (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaGammaNominal (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaThetaNominal (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaVegaNominal (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaRhoNominal (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaDelta (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaGamma (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaTheta (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaVega (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaRho (DataCalculo : TDateTime; Choque : Double) : Double;
function  CalculaPrecoAtivoSubjacente(DataCurva, DataPreco : TDateTime) : Double;
function  CalculaTaxaForwardCC(DataCurva, DataCotacao : TDateTime) : Double;
function  CalculaTaxaBackwardCC(DataCurva, DataCotacao : TDateTime) : Double;
function  CalculaPeriodoVencimento (DataAnalise : TDateTime) : Double;
function  CalculaPeriodoFixing (DataAnalise : TDateTime) : Double;
end;

```

- **Projetos de Investimento**

```

type
  TProjetosInvestimento = class (TObject)
  private
    procedure CarregaElementoMetodosAnalise(Elemento : IXMLElement);
    procedure CarregaElementoConfiguracoes(Elemento : IXMLElement);
    procedure CarregaElementoRegressao(Elemento : IXMLElement);
    procedure CarregaElementoRelatorio(Elemento : IXMLElement);
  protected
  public

```

```

Nome : String;
RS : Boolean;
PBP : Boolean;
VP : Boolean;
TIR : Boolean;
Comparacao : Boolean;
PercentualRS : Double;
PercentualVP : Double;
UsarConfiguracoesSistema : Boolean;
UsarDataSistema : Boolean;
DataAnalise : TDateTime;
Calendario : String;
Basis : String;
Benchmark : String;
TaxaFixa : Double;
PercentualRegressao : Double;
PeriodoRegressao : Double;
SalvarRelatorio : Boolean;
ExibirRelatorio : Boolean;
PathRelatorio : String;
ValorTMA : Double;
ValorPBP : Double;
ManterConfiguracoesProjetos : Boolean;
{ Lista de projetos }
ListaFluxosCaixa : TList;
constructor Create;
procedure CarregaElemento(Elemento : IXMLElement);
function PegaNumeroProjetos : Integer;
procedure PreparaProjetos;
end;

```

- **Série Histórica**

```

(*)
* Record que representa uma cotação de série histórica
*)

```

```

type
TCotacao = record
    Data : TDateTime;

```

```

        Valor : Double;
    end;

    (*
    * Classe que representa uma série histórica no sistema
    *)

    type
        TSerieHistorica = class (TObject)
    private
        Codigo : String;
        Nome: String;
        NomeCalendario : String;
        NomeBasis    : String;
        Natureza     : Integer;
        { Lista de Valores }
        ListaCotacoes : Array of TCotacao;
        function PegaNumeroCotacoes : Integer;
    private
        procedure InsereCotacao (Cotacao : Double; Data : TDateTime);
        procedure InsereRetornoDiario (Cotacao : Double; Data : TDateTime);
        function  VerificaExistenciaCotacao(Data : TDateTime) : Boolean;
        function  VerificaValoresIntervaloDatas(DataInicio, DataFim : TDateTime) : Boolean;
        function  PegaCotacaoData(Data : TDateTime) : Double;
        procedure InicializaServicosAuxiliares;
        function  AcumulaTaxa (DataInicio, DataFim : TDateTime) : Double;
        function  PegaDataCorrigida(Data : TDateTime) : TDateTime;
        function  PegaNDUBasis : Integer;
        function  CalculaPeriodoAnualEntreDatas (DataInicio, DataFim : TDateTime) : TDateTime;
        function  PegaCalendario : TCalendario;
        function  PegaBasis : TBasis;
        function  Normal(X: Double): Double;
    protected
    public
        { Série de retornos percentuais contínuos }
        RetornosDiarios : Array of TCotacao;
        constructor Create;
        function  PegaNome : String;
        function  PegaCodigo: String;
        procedure CarregaElemento(Elemento : IXMLElement);

```

```

function CalculaTaxaPeriodo (DataInicio, DataFim : TDateTime) : Double;
function CalculaTaxaPeriodoSeriePreco (DataInicio, DataFim : TDateTime) : Double;
function CalculaTaxaPeriodoSerieTaxa (DataInicio, DataFim : TDateTime) : Double;
function CalculaFatorPeriodo (DataInicio, DataFim : TDateTime) : Double;
function CalculaFatorPeriodoSeriePreco(DataInicio, DataFim: TDateTime): Double;
function CalculaFatorPeriodoSerieTaxa(DataInicio, DataFim: TDateTime): Double;
{ Funções estatísticas }
procedure GeraListaRetornosDiarios (DataInicio, DataFim : TDateTime);
function CalculaMediaRetornos (DataInicio, DataFim : TDateTime) : Double;
function CalculaVarianciaRetornos (DataInicio, DataFim : TDateTime) : Double;
function CalculaDesvioPadraoRetornos (DataInicio, DataFim : TDateTime) : Double;
function CalculaProbabilidadeSucesso (DataInicio, DataFim : TDateTime;
PercentualRentabilidade : Double; Media, Variancia : Double) : Double;
function PegaNumeroDiasEntreDatas(DataInicio, DataFim : TDateTime) : TDateTime;
function PegaNumeroRetornos : Integer;
end;

```

- **Superfície de Volatilidade**

```

(*
* Record que representa uma volatilidade
*)

type
TCotacao = record
    Data      : TDateTime;
    Strike    : Double;
    Volatilidade : Double;
end;

(*
* Classe que representa uma lista de volatilidades.
*)

type
TListaVolatilidades = class (TObject)
public
    ListaValores : Array of TCotacao;
    Data         : Double;
    Constructor Create;
end;

```

```

(*)
* Classe que representa uma superfície com valores de data, strike e volatilidade.
*)

type
    TSuperficieVolatilidade = class (TObject)
private
    Codigo      : String;
    Nome        : String;
    { Lista de Valores }
    ListasVolatilidades : TList;
private
    procedure InicializaServicosAuxiliares;
    procedure CarregaListaVolatilidades (Elemento : IXMLElement);
    function  PegaListaVolatilidadesData(Data : TDateTime) : TListaVolatilidades;
    function  PegaNumeroListasVolatilidade : Integer;
    procedure InsereListaVolatilidades (ListaVolatilidades : TListaVolatilidades);
    procedure InsereVolatilidade (ListaVolatilidades : TListaVolatilidades; Data : TDateTime;
Strike, Volatilidade : Double);
    function  PegaVolatilidadeData (ListaVolatilidades : TListaVolatilidades; DataVolatilidade :
TDateTime; Strike : Double) : Double;
protected
public
    constructor Create;
    function  PegaCodigo: String;
    function  PegaNome : String;
    procedure CarregaElemento(Elemento : IXMLElement);
    function  PegaVolatilidade (DataSuperficie, DataVolatilidade : TDateTime; Strike : Double) :
Double;
end;

```

Apêndice 3: Instalação e Atualização de Dados

O projeto é de simples instalação e atualização. Para instalar, basta descompactar o arquivo em formato .rar do projeto no diretório C:\ do seu computador. Após fazer isso, será necessário registrar as DLLs para que o sistema funcione corretamente. Para isso, vá ao diretório *C:\Projeto\Build* e execute o arquivo *Registra.bat* que se encarregará de registrar as DLLs do sistema. Após fazer isso, o sistema encontra-se pronto para ser executado: vá até o arquivo *C:\Projeto\Principal* e inicie-o!

Para atualizar os dados o usuário deverá basear-se nos arquivos em formato XML apresentados ao longo desta documentação. A maior parte dos dados provém de séries históricas de preços, como as de cotações da bovespa. Estes dados podem ser encontrados no *site* oficial da bolsa em formato Excel. Uma dica interessante é inserir o formato XML já nas próprias colunas do Excel. A partir daí, é só copiar e colar direto do Excel no arquivo repositório. Os outros dados, como dependem de projeções futuras, são deixados a cargo do usuário escolher a melhor fonte para alimentar os repositórios.



Introdução:

Este manual tem o objetivo de esclarecer as dúvidas sobre a utilização do sistema e fornecer exemplos práticos e de uso cotidiano.

Serão abordadas cada uma das funcionalidades do sistema e suas especificidades, na intenção de dar uma visão completa da ferramenta para cálculos financeiros.

Cabe ressaltar que, em casos de funcionamento irregular - devido a falta de dados ou entradas incorretas - o usuário será alertado através de um sistema de *log*, também especificado adiante.

Componentes Básicos do Sistema

Os componentes mais básicos do sistema são aqueles que tratam dos dados de mercado. Todas as outras funcionalidades do sistema dependem, em última instância, do aparato de cálculo dos dados do mercado. Sendo assim, este manual se inicia dando uma completa visão dos blocos básicos do sistema, para posteriormente avançar no sentido de detalhar melhor as funcionalidades de cada tela do sistema.

O Módulo de Mercado

O módulo de mercado comporta as definições de calendários, basis, benchmarks, séries históricas, curvas e superfícies de volatilidade. Segue abaixo uma descrição de cada um destes módulos, que servem de base aos outros cálculos do sistema.

Calendário

Um calendário no sistema serve apenas para marcar um feriado. Como o sistema oferece a opção de se levar em conta ou não os dias de finais de semana nos cálculos, os feriados devem estar representados no calendário mesmo que ocorram em finais de semana. Segue abaixo o modelo de entrada de um calendário no sistema, em formato XML:

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <Calendarios>
    <Calendario Nome="BMF" Pais="BRA" Bolsa="BMF">
      <Feriado> 12/10/1987 </Feriado>
      <Feriado> 2/11/1987 </Feriado>
      <Feriado> 25/12/1987 </Feriado>
      <Feriado> 1/1/1988 </Feriado>
      <Feriado> 15/2/1988 </Feriado>
    </Calendario>
  </Calendarios>
```

```
</Calendario>
</Calendarios>
```

O usuário poderá inserir quantos elementos “Calendário” quiser, com uma lista de feriados de tamanho indeterminado. Um calendário também possui um nome (seu identificador único), o país ao qual ele se aplica e a bolsa que o rege. Ao inserir um novo elemento “Calendario” no arquivo de XMLs, o usuário terá acesso a ele em todas as telas do sistema automaticamente.

Os feriados utilizados pelo sistema foram retirados dos calendários oficiais das bolsas: Bovespa, BM&F, Over the Counter (Brasil) e NYSE.

Basis

O Basis representa um objeto que auxilia na contagem de dias e no cálculo dos períodos no sistema. Este objeto deve estar sempre acompanhado do objeto de calendário, porque o basis “sabe contar” os dias, ao passo que o Calendário lhe indica os feriados.

No Brasil, é mais comumente utilizado o basis conhecido como “252 dias úteis”, o que no sistema é representado pelo nome NDU_252. Este basis não utiliza os dias de sábado e domingo como entradas válidas para os seus cálculos. Para cálculo do período, o basis conta os dias entre duas datas e divide por um divisor fixo, que pode ser visto abaixo no arquivo de basis do sistema:

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <ListaBasis>
    <Basis nome="NDU_252" NDUAno="252" NDUMes="21" NDUSemana="5"/>
    <Basis nome="ACTUAL_360" NDUAno="360" NDUMes="30"
      NDUSemana="7"/>
  </ListaBasis>
```

Nele podemos ver os dois basis que estarão a disposição do usuário: um é o já mencionado NDU_252, e o outro, mais comum nos Estados Unidos, chama-se ACTUAL_360 e leva em conta inclusive os finais de semana.

Os atributos que acompanham os basis são:

- NDUAno: Representa o divisor utilizado no cálculo do período anual;
- NDUMes: Representa o divisor utilizado no cálculo do período mensal;
- NDUSemana: Representa o divisor utilizado no cálculo do período semanal.

Para tornar os conceitos mais claros, vamos a um exemplo:

Suponhamos querer calcular o período semanal, utilizando o basis NDU_252 e o calendário oficial da Bovespa, entre os dias 01/01/2008 (terça-feira) e 10/01/2008 (quinta-feira). O dia 01/01/2008 é feriado na bolsa, logo não deverá ser contado. 05 e 06/01 são sábado e domingo, respectivamente, e não serão contados por este basis, que só leva em conta os dias úteis. Logo, teremos entre as datas 6 dias úteis (não é incluído o último dia, por ser prática corrente do mercado). O período será então $6/NDUSemana = 6/5 = 1,2$ semanas. Para o Basis ACTUAL_360 o raciocínio é análogo, com as diferenças de contar dias nos finais de semana e também dos divisores.

Dados de Mercado: Séries Históricas, Curvas e Benchmarks.

Os dados de mercado consistem em indicadores econômicos (como por exemplo o IPCA – Índice de Preços ao Consumidor Amplo), taxas de juros (taxa over do mercado brasileiro, CDI, SELIC), índice de ações (Ibovespa, Ibr-X, etc) ou até mesmo as próprias cotações de uma determinada ação ou opção ao longo do tempo. Para representar estes dados, temos as Séries Históricas, as Curvas e os Benchmarks.

Séries Históricas

As Séries Históricas representam valores passados do mercado. Os valores serão buscados em séries históricas em datas anteriores à data de mercado corrente. O análogo à série histórica para valores futuros é chamado no sistema de Curva.

No sistema estão cadastradas cotações de várias ações negociadas na Bolsa de Valores de São Paulo (Petrobrás, Vale do Rio Doce, Gerdau, Banco do Brasil, Bradesco, Embraer, Companhia Siderúrgica Nacional, Eletrobrás, Sadia, Telemar) desde 01/01/2000, assim como os valores do Índice Bovespa, que representa as 50 ações mais líquidas da bolsa brasileiras, e as taxas do CDI e Selic. Estes dados foram obtidos através da própria bolsa de valores e do site do Banco Central do Brasil.

Para facilitar a compreensão deste importante objeto do sistema, iniciaremos com o formato XML responsável pelo seu armazenamento:

```
<SeriesHistoricas>
  <SerieHistorica Codigo="BBAS3" Nome="Banco do Brasil SA"
  Calendario="BOV" Basis="NDU_252" Natureza="Preco">
    <Cotacao Data="2000-01-03 ">1,75 </Cotacao>
    <Cotacao Data="2000-01-04 ">1,66 </Cotacao>
    <Cotacao Data="2000-01-05 ">1,68 </Cotacao>
    <Cotacao Data="2000-01-06 ">1,7 </Cotacao>
  </SerieHistorica>
</SeriesHistoricas>
```

O arquivo inicia-se com uma tag obrigatória “SeriesHistoricas”. A partir dela, o usuário poderá inserir qualquer número de séries históricas com seus respectivos dados de mercado. Uma Série Histórica tem um código que a identifica unicamente no sistema, um nome, um calendário e basis associados a contagem de dias e o atributo “Natureza”, que indica se a Série está carregada com valores de preço ou de taxa. O valor do Atributo “Natureza” influenciará de forma decisiva no cálculo dos fatores de carregamento, como será visto adiante.

Cada série tem uma lista de datas e suas respectivas cotações, que o usuário pode escolher como cotações médias, de abertura, de fechamento, etc, de acordo com sua preferência. A precisão máxima para os valores de cotações é de 8 casas decimais.

O que denomina-se fator de carregamento entre datas em finanças é o multiplicador que leva um valor de uma determinada data a outra. As séries dividem-se em séries de taxa e de preço, e o mecanismo de cálculo é diferente em cada uma delas:

Séries de Taxa: Servem para representar, por exemplo, as taxas do CDI (Certificado de Depósito Interbancário), que representa a taxa “livre de risco” da economia brasileira. O Banco Central do Brasil (Bacen) divulga esta taxa em formato anual (ano-base de 252 dias úteis) diariamente. Assim, para sabermos a taxa efetiva entre dois períodos precisamos “carregar” a taxa dia-a-dia. Como exemplo, podemos calcular o fator do CDI para 1 dia em que a taxa divulgada pelo Bacen foi de 12,42% a.a. Temos:

$$\text{Fator_CDI_1dia} = (1 + 12,42\%)^{(1/252)}$$

O cálculo para períodos maiores é feito utilizando-se o produtório dos fatores diários mostrados acima.

Esta é a forma de se calcular o fator de carregamento para séries de taxa no sistema.

Séries de Preço: Estas séries representam, por exemplo, cotações de ações ou um índice de ações (por exemplo o Ibovespa). Para se calcular o fator de carregamento entre duas datas basta dividir a cotação da data final pela data inicial:

$$\text{Fator_IBOV} = \text{CotacaoFinal}/\text{CotacaoInicial}$$

No caso de um dos dias (ou até mesmo os dois) serem feriados, o sistema automaticamente ajusta as datas para o dia útil anterior.

Desta forma o sistema permite que qualquer valor seja transferido entre duas datas quaisquer no passado. Seguimos agora para a representação de valores futuros.

Curvas

As Curvas se encarregam da representação de valores futuros no sistema. Segue um exemplo de XML de curva:

```

<?xml version="1.0" encoding="iso-8859-1"?>
  <Curvas>
    <Curva Codigo="BBAS3" Nome="Ações Banco do Brasil S.A."
    Natureza="Preco" Calendario="BOV" Basis="NDU_252">
      <ListaCotacoes DataCurva="2008-09-09">
        <Cotacao Data="2008-11-03">14</Cotacao>
        <Cotacao Data="2008-12-23">17</Cotacao>
      </ListaCotacoes>
    </Curva>
  </Curvas>

```

Como pode-se notar, as curvas possuem estrutura muito semelhantes às séries. Possuem um código como identificador único, um “Nome” e uma “Natureza”, que representará se a curva possui valores de taxa ou de preço em suas cotações. Uma das diferenças é que, como temos ali representadas previsões, a curva varia a cada dia. O atributo “DataCurva” representa o dia presente de onde se observa o mercado. Em seguida temos uma lista de cotações e suas respectivas datas. Se a curva for de taxa, os valores representam a taxa anual esperada entre a DataCurva e a data indicada na cotação. Esta taxa será carregada pelo período entre as duas datas para fornecer o fator de carregamento. Caso a curva seja de preço, o valor da cotação é uma estimativa de quanto estaria o preço em uma determinada data, para um observador que analise o mercado na DataCurva. O fator é calculado dividindo-se a cotação final pela cotação inicial. O objeto de curvas é capaz de interpolar os valores dados para encontrar pontos não cobertos diretamente pelos dados. A interpolação é linear e leva em conta o basis e o calendário fornecidos na definição da curva. Assim como o objeto de séries, caso o cálculo seja chamado em um dia não útil, a data é automaticamente ajustada para o dia útil anterior.

Benchmark

O Benchmark nada mais é que a junção dos conceitos de série e de curva. Com ele podemos transferir valores de uma data a outra, utilizando as séries e curvas. O arquivo de Benchmarks do sistema tem o seguinte formato:

```

<?xml version="1.0" encoding="iso-8859-1" ?>
    <Benchmarks>
    <Benchmark Nome="BBAS3" CodigoSerie="BBAS3" CodigoCurva="BBAS3" />
    <Benchmark Nome="BBDC4" CodigoSerie="BBDC4" CodigoCurva="BBDC4" />
    <Benchmark Nome="CDI" CodigoSerie="CDI" CodigoCurva="CDI" />
    </Benchmarks>

```

Caso o usuário queria inserir um novo Benchmark no sistema, deverá cadastrar também sua série e curva, ou utilizar uma das fornecidas no sistema. Se o código da série ou curva forem valores inválidos, o sistema retornará fator 1 para qualquer período.

Para carregamento e desconto de valores entre uma data passada e outra futura, o Benchmark calcula o fator da série para datas passadas e o fator da curva para datas futuras.

Superfícies de Volatilidade

As superfícies de volatilidade servem para armazenar dados das volatilidades de ativos subjacentes de opções a partir de seu preço de exercício. Uma opção de ação, por exemplo, tem especificado em seu contrato um preço de exercício (*strike*), que é o preço combinado para a ação na data de vencimento da opção. Com isso, a representação da volatilidade deve ser feita em superfícies, pois para cada *strike* e para cada data teremos um valor de volatilidade diferente. Segue o formato do arquivo de superfícies de volatilidades:

```

<?xml version="1.0" encoding="iso-8859-1" ?>
    <SuperficiesVolatilidade>
    <SuperficieVolatilidade Codigo="PETR4" Nome="PETR4">
    <ListaVolatilidades DataSuperficie="2008-09-09">
    <Volatilidade Data="2008-11-03" Strike="39">25</Volatilidade>
    <Volatilidade Data="2008-11-23" Strike="58">25,18</Volatilidade>
    </ListaVolatilidades>
    <ListaVolatilidades DataSuperficie="2008-12-02">
    <Volatilidade Data="2008-11-23" Strike="55">23,37</Volatilidade>
    <Volatilidade Data="2008-12-23" Strike="58">25,21</Volatilidade>

```



```
</ListaVolatilidades>  
</SuperficieVolatilidade>  
</SuperficiesVolatilidade>
```

A superfície possui um código, que é seu identificador único no sistema, e um nome. Em seguida temos listas de volatilidade separadas pelos objetos “ListaVolatilidades”, que representam as várias volatilidades para os diferentes strikes e datas, numa determinada data do mercado definida por DataSuperfície. O módulo de superfícies, a exemplo dos outros módulos mencionados, faz o ajuste da data caso esta seja um dia não útil, levando-a para o dia útil anterior.

Com isto fica especificado o módulo de mercado e podemos prosseguir com a apresentação das funcionalidades do sistema.

Tela Principal

É nesta tela que o usuário tem as opções das várias análises realizadas pelo sistema. Para facilitar a explicação, segue abaixo uma imagem da tela:

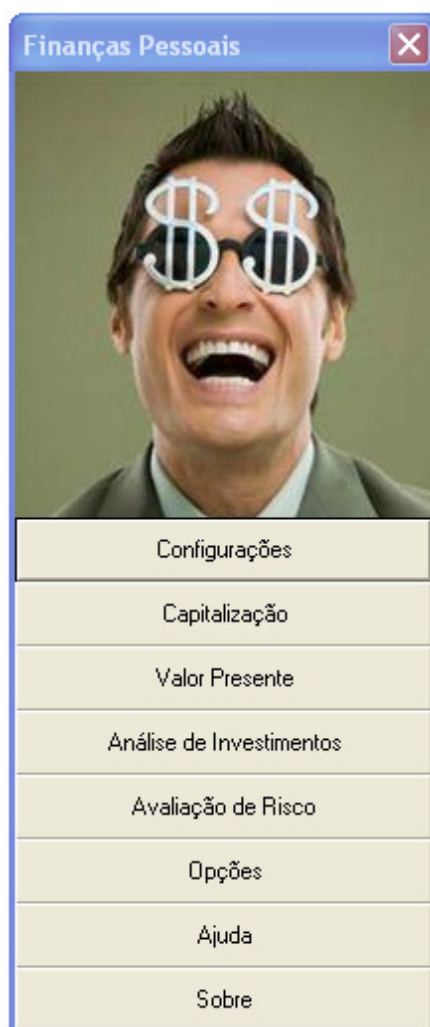


Figura 17 – Tela Principal do sistema

A tela mostra as seguintes opções, que serão detalhadas a seguir:

1. **Configurações:** Permite ao usuário selecionar as propriedades básicas no sistema, que serão utilizadas nas outras telas.
2. **Capitalização:** Permite a análise de capitalização simples, composta e contínua.
3. **Valor Presente:** Fornece a análise de um projeto de investimento através dos métodos de Rentabilidade Simples, Pay-back Period, Valor Presente e Taxa Interna de Retorno.

4. **Análise de Investimentos:** Fornece análise comparativa de vários projetos de investimentos, com a opção de se gerar, ao final da análise, um relatório em formato HTML com os resultados da análise.
5. **Avaliação de Risco:** Através de medidas de médias de retorno, desvio-padrão e variância, permite avaliar o risco de se fazer um investimento em um determinado ativo. Permite comparação de até 4 ativos simultâneos.
6. **Opções:** Engloba os cálculos de opções de ações (*Stock Option*), assim como vários de seus analíticos (delta, gamma, vega, theta, rho, volatilidade, entre outros).
7. **Ajuda:** Fornece ao usuário ajuda sobre como utilizar o software para tirar o máximo proveito dele e esclarece dúvidas sobre os vários erros reportados pelo sistema de *Log*.
8. **Versão:** Exibe a versão do sistema, assim como sua data de liberação, para que auxilie no processo de detecção de problemas por versão.

Tela de Configurações Iniciais

É na tela de configurações iniciais que o usuário escolhe seus parâmetros que serão utilizados como *default* nos cálculos das outras telas do sistema. Em alguns casos algumas dessas propriedades gerais do sistema podem ser alteradas em outras telas, por motivos de praticidade e facilidade de visualização. Estes casos específicos serão mencionados nos capítulos próprios de cada tela. Segue abaixo a representação XML das configurações do sistema, que ficam armazenadas em *C:\Projeto\XML\Configuracoes\Configuracoes.xml*.

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <Configuracoes UsarDataAtual="false">
    <Data>18/9/2008</Data>
    <Calendario>BOV</Calendario>
    <Basis>NDU_252</Basis>
    <Benchmark>CDI</Benchmark>
    <Projetos Salvar="false">PETROBRAS</Projetos>
    <Series DataInicio="2006-09-14" DataFim="2007-09-14">EMBR3</Series>
  </Configuracoes>
```

Os elementos “Projetos” e “Series” guardam informações de outras telas, que serão comentadas mais à frente. O atributo “UsarDataAtual” representa a opção de iniciar o sistema utilizando a data do sistema operacional. Além disso, são armazenados o basis, calendário e benchmark que serão utilizados nas outras telas. Cabe lembrar que em alguns casos estas configurações poderão ser alteradas em outras telas, e as alterações serão gravadas neste mesmo repositório XML.

A tela de configurações iniciais é mostrada abaixo:

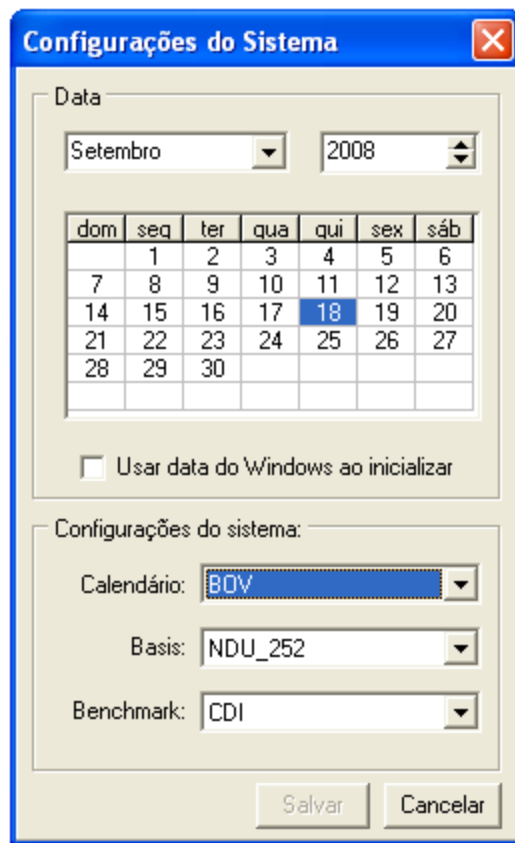


Figura 18 – Tela de Configurações do sistema

A primeira opção da tela é a escolha da data do sistema. O usuário pode escolher o mês, ano e dia. Um calendário é mostrado para que o usuário saiba que dia da semana está escolhendo. Cabe ressaltar aqui que a data escolhida representa a **data de mercado**, ou seja, a data de acordo com a qual serão buscados os dados de mercado para os cálculos financeiros. Este ponto será melhor detalhado adiante.

Como é muito comum a situação de o usuário querer realizar a análise no próprio dia atual, foi incluída a opção “Usar data do windows ao iniciar”. Com esta opção marcada, o sistema inicia na mesma data do sistema operacional, eliminando a necessidade de ajuste diário da data.

Na parte de Configurações do Sistema, são expostas 3 opções: Calendário, que representa o calendário de feriados utilizados para a análise (Ver tópico Calendários para mais detalhes), Basis, que representa a base de contagem de dias do sistema, se 252 dias úteis ou 360 dias corridos (Para mais detalhes, ver tópico Basis) e Benchmark, este uma estrutura um pouco mais complexa, representando o custo de oportunidade para um

determinado investimento (Informação mais detalhada pode ser encontrada no tópico Benchmark).

Por fim, o botão Salvar salva as alterações no repositório XML, enquanto o botão Cancelar descarta as alterações feitas pelo usuário.

Tela de Capitalização

Nesta tela são realizados cálculos de capitalização simples, composta e contínua. O formato da tela é como mostrado abaixo:

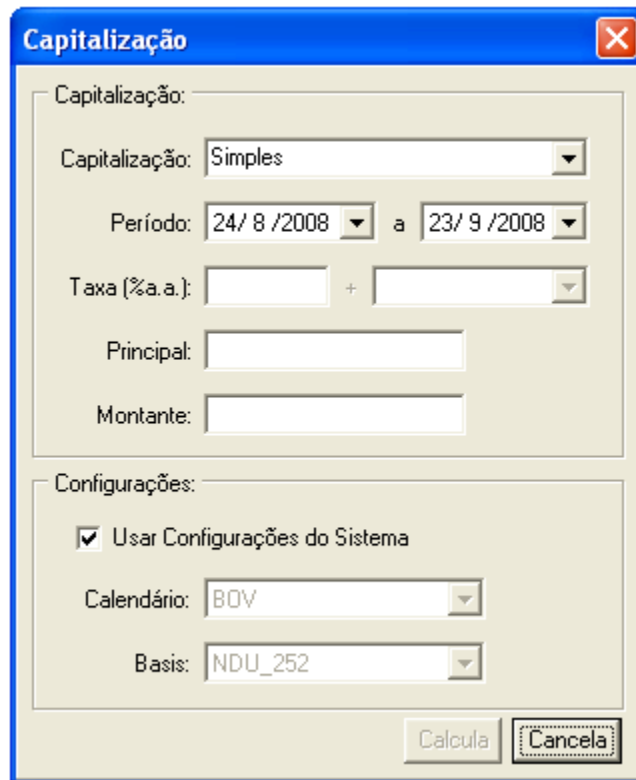


Figura 19 – Tela de Capitalização.

A primeira opção define o tipo de capitalização a ser utilizada: simples, composta ou contínua. A segunda define o período de análise, onde são informadas as datas inicial e final.

Na terceira opção temos a informação da taxa e do Benchmark. O sistema só está preparado para utilizar os dados de mercado (representados pelo benchmark) através da capitalização composta. Sendo assim, ao selecionar na combo de capitalização algum valor diferente de “Composta”, a combo de Benchmark é imediatamente desabilitada e limpa.

Nos casos de capitalização simples e contínua, que não permitem o uso de Benchmark, o usuário terá 3 opções de preenchimento: taxa, principal e montante. A intenção aqui é que se deixe 1(um) e apenas um destes campos sem preenchimento. O sistema automaticamente detecta o campo vazio e calcula-o. Seu valor aparece na cor azul para destacá-lo dos outros. Por fins de praticidade, foi incluído na parte de baixo da tela um acesso a duas das configurações do sistema anteriormente mencionadas: Calendário e Basis. Também com o intuito de tornar o software mais amigável para o usuário, há uma opção de se usar as configurações atuais do sistema (escolhidas na tela de configurações iniciais), que, desmarcada, permite um ajuste rápido do calendário e basis pretendidos.

Já no caso da capitalização composta, as opções são ainda mais variadas. Para o caso da taxa, podemos ter agora uma taxa fixa somada a um custo de oportunidade derivado de dados de mercado (por exemplo, o de ter investido nas ações da Petrobrás). Estes dados de mercado são calculados através de séries históricas e curvas de mercado, já detalhadas em seus respectivos tópicos. A diferença entre estes dois objetos do sistema é muito simples: enquanto a série apresenta dados passados, a curva de mercado representa as projeções para o futuro. Aqui se torna importante mencionar a data de mercado, selecionada na tela de configurações iniciais. Ela aparece na barra de título desta tela de capitalização. Para datas anteriores à data de mercado, os dados correspondentes serão buscados nas respectivas séries históricas, ao passo que para o dia atual e datas futuras, as consultas serão feitas nas curvas de mercado. A estrutura de cálculo de fatores e taxas de carregamento pode ser melhor compreendida observando o tópico Mercado, já desenvolvido anteriormente. Para o caso de capitalização composta, também vale a regra de deixar uma variável em branco para que o sistema detecte e calcule a variável faltante. Cabe mencionar apenas que, para o caso da taxa, o Benchmark ou a Taxa já são suficientes para definir esta variável como preenchida. Caso haja os dois, seus efeitos serão somados no cômputo do resultado final.

Tela de Valor Presente

Nesta tela pode-se realizar o carregamento e desconto de fluxos de caixa através de taxas fixas e/ou de benchmarks de mercado. Segue o esquema da tela:

Valor Presente 18/9/2008

Valor Presente:

Fluxos:

Método:

Benchmark: + %

Data Análise: 1/9/2008

Data Mercado: 1/9/2008

Resultado:

Fluxos de Caixa:

| Data | Fluxo | VP |
|------|-------|----|
| | | |
| | | |
| | | |
| | | |

Calcula Fechar

Figura 20 – Tela de Valor Presente.

Na primeira combo pode-se escolher o projeto de investimento, ou seja, a sequência de fluxos de caixa de um dado planejamento de investimento. O repositório para estes arquivos está localizado em C:\Projeto\XML\ValorPresente\FluxosDeCaixa.xml. O formato do arquivo é mostrado abaixo:

```

<?xml version="1.0" encoding="iso-8859-1"?>
  <FluxosDeCaixa>
    <Fluxos Nome="Fluxos - Lissandro" Calendario="BOV" Basis="ACTUAL_360"
      Benchmark="CDI" TaxaFixa="10" DataAnalise="2008-10-01">
      <Fluxo Data="2008-03-23">-1000</Fluxo>
      <Fluxo Data="2008-04-23">200</Fluxo>
      <Fluxo Data="2008-05-23">300</Fluxo>
      <Fluxo Data="2008-06-23">400</Fluxo>
      <Fluxo Data="2008-07-23">200</Fluxo>
    </Fluxos>
  </FluxosDeCaixa>

```

O arquivo começa com a tag “FluxosDeCaixa”, e a partir daí pode ter quantos projetos de investimento o usuário desejar. Para adicionar um projeto de investimento, o usuário deverá inserir um elemento “Fluxos”, que possui um nome como identificador único e um calendário e basis. Caso o projeto não possua calendário e basis definidos, serão utilizadas as configurações do sistema. O usuário também pode escolher o Benchmark, a taxa fixa (constante) e a data de análise que deverá ser considerada para o carregamento dos fluxos. Em seguida, há a lista de fluxos representada por pares data-valor. Uma entrada de dinheiro deve ser representada como um fluxo positivo e uma saída com um valor negativo. Para maiores esclarecimentos de como são feitos os carregamentos e descontos de valores segundo um benchmark pode-se recorrer à parte que trata dos benchmarks, neste mesmo tutorial.

A segunda opção permite alternar entre os diferentes métodos de análise de investimento, a saber: Rentabilidade Simples, Período de Retorno de Investimento (Pay-back Period), Valor Presente e Taxa Interna de Retorno (TIR). No caso dos métodos que não consideram o valor do dinheiro no tempo (Rentabilidade simples e Pay-back Period), as opções de datas, taxa e benchmark serão desabilitadas por não fazer sentido para o cálculo em questão.

Seguem os cálculos feitos pelo sistema para cada método de análise:

• **Rentabilidade Simples (h):** é calculada dividindo-se o fluxo de caixa anual médio do projeto (L) pelo total do investimento (G):

$$h = L/G$$

Como pode-se notar, o índice revela o retorno anual por unidade de capital investido no projeto. O projeto de investimento será tanto melhor quanto maior for a quantidade de recuperação de recursos por unidade de investimento.

A grande vantagem deste índice é a facilidade de cálculo, obtido a partir das receitas e custos orçados para o projeto. O índice apresenta, entretanto, a desvantagem básica de não considerar os efeitos do tempo sobre o valor do dinheiro.

Seria um erro calcular a rentabilidade simples utilizando lucro. É importante ter em mente que quando se utilizam técnicas de análise de investimento deve-se empregar o fluxo de caixa.

Para ter acesso a este indicador no sistema, selecione Rentabilidade Simples na combo de Método. O valor de h aparecerá em azul na parte de resultado.

• **Período de Retorno do Investimento(p):** também conhecido com *pay-back period*, o método é um dos mais difundidos entre os administradores de empresas. O método consiste, basicamente, na determinação do número de períodos necessários para recuperar o capital investido (no sistema, este valor é exibido numa base anual).

A partir deste dado, a empresa decide sobre a implementação do projeto, comparando-o com os seus referenciais no tempo para a recuperação do investimento.

O período de retorno do investimento (p) é a relação entre o total do investimento e o fluxo de caixa médio provável gerado pelo projeto:

$$p = G/L = 1/h$$

Pela expressão nota-se que o período de retorno do investimento equivale ao inverso da rentabilidade simples, apresentando, portanto, os mesmos méritos e deméritos desse índice. Como o método do período de retorno do investimento não leva em conta o valor do dinheiro no tempo, poderão ocorrer situações em que projetos

desiguais do ponto de vista dos fluxos de caixa apresentem o mesmo *pay-back period*, tornando-os indiferentes à luz desse indicador. Essas considerações tornam desaconselhável o uso deste método como critério de avaliação de investimento: deve ser utilizado apenas como informação complementar sobre a recuperação do investimento realizado.

Para ter acesso a este indicador no sistema, selecione Pay-back Period na combo de Método. O valor de p aparecerá em azul na parte de resultado.

Com a finalidade de contornar parcialmente este problema, o sistema oferece os métodos do Valor Presente Líquido (VPL) e Taxa Interna de Retorno (TIR).

• **Valor Presente Líquido (VPL):** O método do Valor Presente Líquido, também conhecido como método do valor atual líquido, é caracterizado, basicamente, pela transferência para a data zero das entradas e saídas dos fluxos de caixa associados ao projeto, tendo como base de cálculo a sua taxa mínima de atratividade.

No sistema, o cálculo do VPL foi flexibilizado para poder ser realizado em uma data qualquer, e não apenas na data zero. Além disso, a taxa mínima de atratividade (TMA), pode ser representada por um benchmark com os dados de mercado, por exemplo as cotações de uma ação qualquer listada na Bovespa.

Considerando as entradas e saídas de caixa, representadas ao longo do tempo por $R_0, R_1, R_2, \dots, R_n$:

$$VPL = R_0 + \frac{R_1}{(1+i)} + \frac{R_2}{(1+i)^2} + \dots + \frac{R_n}{(1+i)^n}$$

e a taxa mínima de atratividade do projeto (supondo que seja constante apenas para fins de facilitar a demonstração), representada por i , o valor presente líquido do projeto, indicado abaixo por VPL, é obtido pela expressão:

$$VPL = \sum_{j=0}^n \frac{R_j}{(1+i)^j}$$

Quanto ao sinal do VPL, podemos ter:

1. $VPL > 0$: O resultado positivo significa que o valor presente das entradas supera o valor presente das saídas de caixa. Nesse caso, o projeto será economicamente interessante à taxa de juros considerada. Quanto maior o VPL, mais interessante será o projeto de investimento.

2. $VPL = 0$: Significa que o valor presente das entradas é igual ao valor presente das saídas de caixa. Nesse caso, o projeto ainda não é desinteressante do ponto de vista econômico, pois as entradas futuras são equivalentes aos desembolsos realizados com o projeto. Isso implica dizer que o projeto produz retorno igual à taxa mínima de atratividade da empresa.

3. $VPL < 0$: Significa que o valor presente das entradas é menor que o valor presente das saídas de caixa do projeto. Nesse caso, o projeto não será economicamente interessante à taxa de juros considerada, pois não ocorre sequer a recuperação do investimento realizado.

É importante deixar claro que o VPL de um projeto depende da taxa mínima de atratividade usada em seu cálculo. Não há uma medida única de VPL: ela é função da taxa de juros utilizada.

• **Taxa Interna de Retorno (TIR)**: Por definição, a taxa interna de retorno de um projeto é a taxa de juros i para a qual seu VPL é nulo:

$$R_0 + \frac{R_1}{(1+i)} + \frac{R_2}{(1+i)^2} + \dots + \frac{R_n}{(1+i)^n} = 0$$

A solução da equação anterior é complexa e exige métodos matemáticos sofisticados. O sistema utiliza um algoritmo de busca de raízes por bissecção.

Do que foi dito anteriormente, podemos concluir que a TIR de um projeto é a taxa de juros para a qual o valor presente das entradas iguala, em valores absolutos, o valor presente das saídas de seus fluxos de caixa. Caracteriza, dessa forma, a taxa de remuneração do capital investido.

Após determinar a TIR do projeto, devemos compará-la com sua TMA. O projeto será considerado rentável e, portanto, atraente do ponto de vista econômico, se sua TIR for, no mínimo, igual à TMA. Ou seja, a TIR deve ser maior ou igual a TMA.

Obs.: O método ora apresentado pressupõe, implicitamente, que os fluxos intermediários de caixa do projeto sejam reinvestidos à sua taxa interna de retorno. Essa premissa é de fundamental importância na utilização prática do método.

Para os métodos Rentabilidade Simples, *Pay-back Period* e Taxa Interna de Retorno, as entradas benchmark, taxa fixa e as datas de análise e mercado são desabilitadas, já que não fazem sentido para estes indicadores. Ao realizar estes cálculos, o sistema exibe o valor calculado na variável resultado, exibida em azul.

No caso do cálculo do Valor Presente, o sistema fornece ao usuário uma lista dos fluxos do projeto contendo 3 colunas: Data (representando a data de cada fluxo), Fluxo (representando o valor do fluxo) e VP (Valor Presente do fluxo na data de análise, usando para carregamento e desconto o mercado da data de mercado). Esta parte da tela possui ainda, à sua direita, um botão de copiar que permite levar estes fluxos para uma planilha eletrônica no Microsoft Excel ou OpenOffice Calc.

Para acionar o cálculo, depois de escolhidos os parâmetros, basta clicar em “Calcula”. Para desistir da operação, basta clicar no botão “Cancelar”.

Tela de Análise de Investimentos

A tela de Análise de Investimentos pode ser acessada a partir da tela principal do sistema.

Nesta tela o usuário tem à sua disposição os mecanismos de cálculo da tela anterior, organizados de uma maneira mais interessante e sofisticada. Agora as entradas podem ser grupos de projetos. O usuário poderá gerar um relatório comparativo destes projetos, atribuindo percentuais de confiança aos métodos de análise. Para facilitar a visualização, segue o formato da tela de análise de investimentos:

Análise de Investimentos 1/9/2008

Projetos e Relatórios:

Projetos: Data do Mercado:

Comparar Projetos

Exibir Relatório HTML

Salvar relatório HTML em:

Data de Análise:

Usar Data do Sistema

| dom | seg | ter | qua | qui | sex | sáb |
|-----|-----|-----|-----|-----|-----|-----|
| | | | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 31 | |

Configurações Gerais:

Usar Configurações do Sistema

Calendário:

Benchmark: +

Basis:

Manter configurações dos projetos

Métodos de Análise:

Rentabilidade Simples Pay-back Period Valor Presente TIR

Taxa mínima de Atratividade (TMA): % Percentual RS: %

Período Médio de Retorno do Investimento: (anos) Percentual VP: %

Figura 21 – Tela de Análise de Investimentos

Projetos e Relatórios: Nesta parte o usuário escolhe um grupo de projetos, cujo repositório pode ser encontrado em *C:\Projeto\XML\AnaliseInvestimentos*. O formato do arquivo pode ser visto no exemplo abaixo:

```
<AnaliseInvestimentos>
  <Projetos Nome="PETROBRAS">
    <MetodosAnalise RS="true" PBP="true" VP="true" TIR="true"
Comparacao="true" TaxaMinimaAtratividade="8" ValorPBP="3" PercentualRS="10"
    PercentualVP="90"/>
    <Configuracoes UsarConfiguracoesSistema="true" UsarDataSistema="false"
Data="23/03/2008" Calendario="BOV" Basis="NDU_252" Benchmark="CDI"/>
    <Relatorio Salvar="true" Exibir="true" PathRelatorio=""/>
    <Projeto>
      <Fluxos Nome="PETRO-TI" Calendario="BOV" Basis="NDU_252"
Benchmark="CDI" TaxaFixa="0" DataAnalise="2006-01-05">
        <Fluxo Data="2006-01-05">-18000</Fluxo>
        <Fluxo Data="2009-07-05">1875</Fluxo>
        <Fluxo Data="2009-10-05">1875</Fluxo>
        <Fluxo Data="2010-10-05">1875</Fluxo>
      </Fluxos>
    </Projeto>
    <Projeto>
      <Fluxos Nome="PETRO-Plataforma" Calendario="BOV" Basis="NDU_252"
Benchmark="CDI" TaxaFixa="10" DataAnalise="2008-10-01">
        <Fluxo Data="2006-01-05">-18000</Fluxo>
        <Fluxo Data="2006-04-05">1875</Fluxo>
        <Fluxo Data="2009-07-05">1875</Fluxo>
        <Fluxo Data="2009-10-05">1875</Fluxo>
        <Fluxo Data="2010-10-05">1875</Fluxo>
      </Fluxos>
    </Projeto>
  </Projetos>
</AnaliseInvestimentos>
```


O elemento “Projetos” abre o grupo de projetos e é ele que aparece na primeira combo da tela. Em seguida temos as configurações que serão utilizadas para se gerar o relatório, que podem ser divididas em partes:

Métodos de Análise: informam que métodos estarão presentes no relatório: Rentabilidade Simples, *Pay-back Period*, Valor Presente e Taxa Interna de Retorno. Neste elemento também serão definidos os valores do período médio de retorno do investimento e da taxa mínima de atratividade da empresa, para fins de comparação e avaliação dos projetos. Ainda neste elemento serão definidos os percentuais de confiança que o gerente financeiro dá aos métodos do Período de Retorno do Investimento e de Valor Presente. Estes valores serão utilizados para comparar os dois projetos e decidir sobre qual seria o mais viável de ser implementado.

Configurações: Neste elemento o usuário decide se irá utilizar as configurações e a data do sistema ou se irá escolher atributos diferentes. Também poderá escolher basis, calendário, benchmark e taxa fixa para serem aplicados no processo de carregamento e desconto.

Relatório: Neste elemento será informado se o usuário deseja salvar o relatório e também se deseja exibi-lo ao fim do cálculo.

Projeto: Este elemento representa um projeto de investimento e é equivalente ao elemento FluxosDeCaixa, já descrito anteriormente. Esta estrutura facilita o mecanismo de testes do sistema, por apresentar estruturas compatíveis.

O relatório HTML gerado pelo sistema possui um cabeçalho com informações sobre o projeto. Em seguida, surge uma estrutura em tabelas que trazem as mais variadas informações. Inicialmente é exibida uma tabela com as configurações gerais escolhidas pelo usuário, como basis, calendário e benchmark. Logo em seguida vem uma tabela chamada “Configurações do Analista”, que mostra as escolhas dos parâmetros descritos acima em Métodos de Análise.

Em seguida temos a análise dos fluxos de caixa, na qual cada projeto cadastrado gera uma tabela com a data do fluxo, seu valor, seu valor presente, o valor acumulado dos fluxos sem levar em conta o valor dos fluxos no tempo e o VP Acumulado, para

mostrar ao analista o volume de dinheiro que ele possui dia-a-dia. Cada tabela possui o nome do projeto que lhe deu origem.

Em seguida vem um comparativo das análises, no qual os indicadores são calculados e é emitido um parecer sobre o projeto, além de exibir um índice de desempenho relativo às informações do analista para a sua taxa mínima de atratividade e seu período médio de retorno do investimento.

Para ver um modelo de relatório gerado pelo sistema, basta acessar *C:\Projeto\Doc\Exemplos Relatórios*.

Tela de Avaliação de Riscos de Investimento

A tela de Avaliação de Riscos pode ser acessada a partir da tela principal do sistema. Nesta tela, o usuário irá escolher um conjunto de no máximo 4 ativos e um período a partir do qual será traçado um histórico dos retornos desses ativos. Antes de prosseguir, definimos o que entende-se por retorno de um ativo:

$$\text{Retorno} = \text{LN}(\text{Cotação}(t)/\text{Cotação}(t-1))$$

Esta forma de calcular retorno é muito utilizada em finanças porque parte do pressuposto de que a evolução das cotações se dá de forma contínua.

Para facilitar o entendimento, segue o modelo da tela sendo utilizada para comparar 4 ativos que compõem a carteira do Ibovespa:

The screenshot shows a software window titled "Avaliação de Risco". It contains several sections:

- Dados Históricos:** Four dropdown menus for "Valores Históricos" (ELET6, GGBR4, EMBR3, SDIA4) and two for "Início Histórico" (14/9/2006) and "Fim Histórico" (14/9/2007).
- Investimento:** Two dropdown menus for "Data Início" (14/9/2007) and "Data Fim" (14/9/2008), and a text input for "Rentabilidade Esperada" (30) followed by "% a.p."
- Resultados:** A table with 5 columns (headers: ELET6, GGBR4, EMBR3, SDIA4) and 10 rows of statistical data.
- Buttons:** "Executar" and "Cancelar" at the bottom right.

| | ELET6 | GGBR4 | EMBR3 | SDIA4 |
|-----------------|-------------|-------------|-------------|-------------|
| Média | 0,0009089% | 0,0016540% | -0,0003649% | 0,0017945% |
| Variância | 0,0004424% | 0,0004236% | 0,0002997% | 0,0006696% |
| Desvio-Padrão | 0,2103429% | 0,2058155% | 0,1731297% | 0,2587590% |
| Série Retornos | Clique Aqui | Clique Aqui | Clique Aqui | Clique Aqui |
| Período Invest. | 247 dias | 247 dias | 247 dias | 247 dias |
| Período Hist. | 246 dias | 246 dias | 246 dias | 246 dias |
| Média Período | 25,1565078% | 50,4109558% | -8,6199848% | 55,7138794% |
| % Sucesso | 36,4001338 | 91,8603058 | 0,0000955 | 86,2778069 |

Figura 22 – Tela de Avaliação de Risco de Investimentos

As datas de início e fim de histórico delimitam o universo de dados que serão utilizados para gerar as estatísticas. A partir desses dados das séries históricas, será montada uma outra série, agora de retornos, para que possamos realizar os cálculos estatísticos.

Em seguida, informamos as datas de início e fim do investimento, e a rentabilidade que se espera conseguir no período. O sistema avaliará qual das alternativas será a melhor, se o comportamento fosse semelhante ao período que foi observado no histórico.

Ao clicar no botão “Executar”, o sistema realiza os cálculos e exibe os resultados, explicados a seguir:

Média: Representa a rentabilidade média diária de cada um dos ativos.

Variância: Representa a variância da série de retorno de cada ativo.

Desvio-padrão: Representa o desvio-padrão de cada série de retornos dos ativos escolhidos.

Série Retornos: Ao clicar neste *link*, o usuário será levado a uma outra tela, na qual poderá observar os retornos calculados para cada dia no período considerado. Segue abaixo o modelo da tela:

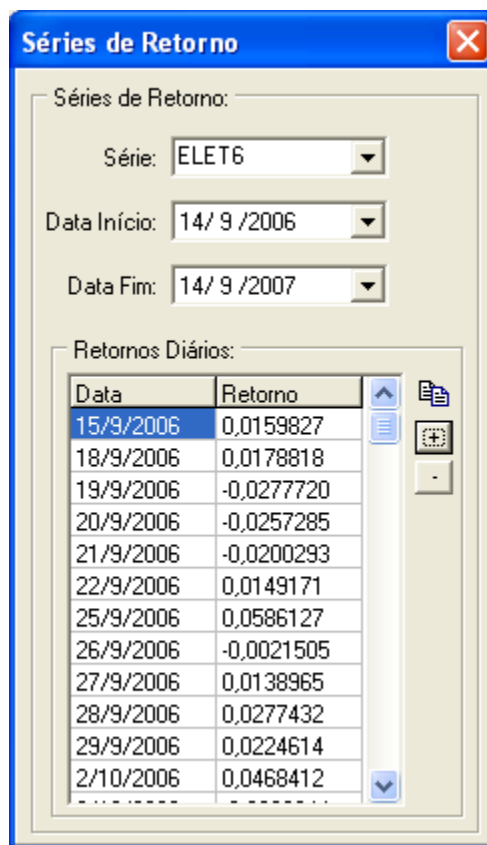


Figura 23 – Tela de Retornos de Séries Históricas

Nesta tela, o usuário poderá inclusive escolher o período em que deseja analisar os retornos, assim como a série que deseja observar. Pode ainda alterar o número de casas decimais da visualização e copiar os dados para uma planilha eletrônica. Esta ferramenta é muito útil para fins de validação dos cálculos deste módulo, e também para trazer mais confiança ao usuário.

Período de Investimento: Mostra o período em dias (segundo o basis e calendário da própria série) em dias.

Período Histórico: Análogo ao anterior, mas para o período do histórico.

Média Período: Representa a média de rentabilidade do ativo naquele período. É calculada compondo-se de forma composta a taxa média de retorno diário para cada dia do período de investimento.

% Sucesso: Representa o percentual de um dado investimento ter sucesso (ou seja, a rentabilidade do investimento ser maior ou igual á rentabilidade esperada). Este

cálculo é realizado da seguinte maneira: a partir da série de retorno, obtemos a média e a variância. Com estes valores, o sistema monta internamente uma distribuição gaussiana. Através de um algoritmo numérico, o sistema realiza a integral da distribuição de probabilidade gaussiana encontrada para encontrar a probabilidade de o rendimento ser maior ou igual à rentabilidade esperada para o investimento.

Para uma melhor compreensão da distribuição gaussiana e do mecanismo de cálculo acima, veja o Anexo “O Modelo de Black & Scholes”.

Nesta tela de Avaliação de risco, o usuário pode variar o número de casas decimais do resultado, além de poder copiar o conteúdo da análise e colar em sua planilha eletrônica.

Tela de Análise de Opções

Esta pode ser acessada a partir da tela principal, e é a mais complexa do sistema. Ela se destina a usuários um pouco mais familiarizados com o mercado financeiro, em especial com o mercado de opções. Para ilustrar a explicação, segue o esquema da tela:

| Opções Cadastradas | |
|---------------------|--|
| PETR4_55_2008-12-12 | |
| PETR4 39 2008-11-03 | |

| Contagem de Dias | |
|------------------|-----------|
| Emissão: | 3/9/2008 |
| Vencimento: | 3/11/2008 |
| Fixing: | 3/11/2008 |
| Calendário: | BOV |
| Basis: | NDU_252 |
| Tipo Opção: | Call |

| Mercado | |
|-----------------|------------|
| Strike: | 39,00 |
| Subjacente: | PETR4 |
| Volatilidade: | PETR4 |
| Principal: | 1.000,0000 |
| Curva Forward: | CDI |
| Curva Backward: | DDI |
| Vol. Implícita: | 0,0000 |

| Choques Paralelos | | | |
|-------------------|--------|--------|--------|
| Delta: | 0,0001 | Theta: | 1,0000 |
| Vega: | 0,0100 | Rho: | 0,0001 |

| Analíticos | |
|--------------------|--------------|
| Valor Presente | 5,3304834597 |
| PrecoUnitário | 0,0053304835 |
| Taxa Forward CC | 0,1135072408 |
| Taxa Backward CC | 0,0518331486 |
| Período Vencimento | 0,1547619048 |
| Período Fixing | 0,1547619048 |

Figura 24 – Tela de Análise de Opções

Os dados que suprem esta tela estão localizados no repositório *C:\Projeto\XML\Opcoes*. O formato do repositório é mostrado abaixo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```

<Opcoes>
  <Opcao Codigo="PETR4_55_2008-12-12" Nome="PETR4_55_2008-12-12"
    Call="True">
    <ContagemDias Calendario="BOV" Basis="NDU_252" DataEmissao="2008-05-
      05" DataVencimento="2008-12-12" DataFixing="2008-12-12"/>
      <Mercado Principal="1000" Strike="60" CurvaForward="CDI"
        CurvaBackward="" CurvaPrecoSubjacente="CDI" Superficie=""
          VolatilidadeImplicita="15"/>
    </Opcao>
  <Opcao Codigo="PETR4_39_2008-11-03" Nome="PETR4 39 2008-11-03"
    Tipo="Call">
    <ContagemDias Calendario="BOV" Basis="NDU_252" DataEmissao="2008-09-
      03" DataVencimento="2008-11-03" DataFixing="2008-11-03"/>
      <Mercado Principal="1000" Strike="39" CurvaForward="CDI"
        CurvaBackward="DDI" CurvaPrecoSubjacente="PETR4"
          SuperficieVolatilidade="PETR4" VolatilidadeImplicita=""/>
    </Opcao>
  </Opcoes>

```

Neste arquivo, o usuário pode inserir quantas opções quiser segundo o modelo acima. Uma opção tem um identificador único que é seu código. O nome é útil para exibição na tela (note que na parte da direita da tela temos uma lista de opções representadas pelo nome). Tudo na tela é automático, por isso não há o botão “Calcular” nem o botão “Cancelar”. Qualquer alteração efetuada pelo usuário fará efeito imediatamente.

Ainda na descrição dos atributos da opção, há o seu tipo, que pode ser de compra ou de venda (representados no sistema respectivamente por *call* e *put*).

A seguir, encontramos os seguintes elementos:

Contagem de Dias: Este elemento informa as datas necessárias à precificação da opção. Nele estão as datas de emissão do contrato, a data de vencimento do contrato e a data de fixing, inserida para representar o *lag* de liquidação do ativo subjacente à opção. Além disso, nesse elemento são definidos o basis e o calendário.

Mercado: Neste elemento estão representados os dados de mercado. Entre eles encontramos o principal da opção (também conhecido como *notional* do contrato), o *Strike* (que representa o preço de exercício da opção), as curvas forward e backward, que são responsáveis pelo carregamento e desconto dos valores, a curva de preço do ativo subjacente à opção, a superfície de volatilidade (onde será buscada a volatilidade do ativo subjacente na data de mercado) e a volatilidade implícita, caso o analista decida utilizar um valor fixo para a volatilidade.

Na tela também poderão ser escolhidos os choques paralelos para se calcular as “gregas” da opção, como são conhecidas as medidas de sensibilidade.

Com estes dados fornecidos, o sistema calculará uma série de analíticos, listados e descritos abaixo:

- **Valor Presente:** Calcula o valor presente de uma opção segundo o modelo de Black & Scholes. Para maiores informações, vide Anexo “O Modelo de Black & Scholes”.
- **Preço Unitário:** Representa o preço ou prêmio da opção sem levar em conta o principal.
- **Taxa Forward CC:** Representa a taxa forward continuamente composta da opção. O sistema pega na curva forward o valor discreto da taxa e o converte para contínuo para ser utilizado no cálculo do preço.
- **Taxa Backward CC:** Análoga ao explicado para a taxa forward, só que para a curva backward.
- **Período Vencimento:** Representa o período anual calculado da data de análise até a data do vencimento da opção, segundo o basis e o calendário da opção.
- **Período Fixing:** Representa o período anual que vai da data de análise até a data fixing da opção, levando em conta seu basis e seu calendário.
- **Volatilidade:** Exibe a volatilidade do ativo subjacente à opção.
- **Black & Scholes Delta:** Representa a sensibilidade relativa do preço da opção em relação ao ativo subjacente.
- **Black & Scholes Delta Nominal:** Representa a sensibilidade em termos absolutos do preço da opção em relação ao ativo subjacente.
- **Black & Scholes Vega:** Representa a sensibilidade relativa do preço da opção em relação à volatilidade.

- **Black & Scholes Vega Nominal:** Representa a sensibilidade em termos absolutos do preço da opção em relação à volatilidade.
- **Black & Scholes Gamma:** Representa a sensibilidade relativa do preço da opção em relação ao delta.
- **Black & Scholes Gamma Nominal:** Representa a sensibilidade em termos absolutos do preço da opção em relação ao delta.
- **Black & Scholes Theta:** Representa a sensibilidade relativa do preço da opção em relação ao tempo.
- **Black & Scholes Theta Nominal:** Representa a sensibilidade em termos absolutos do preço da opção em relação ao tempo.
- **Black & Scholes Rho:** Representa a sensibilidade relativa do preço da opção em relação à taxa de juros.
- **Black & Scholes Rho Nominal:** Representa a sensibilidade em termos absolutos do preço da opção em relação à taxa de juros.

A partir dos resultados acima, o usuário poderá montar de forma mais eficiente sua carteira de opções.

A tela conta com botões para variar o número de casas decimais do resultado, além de um botão copiar para transportar seus resultados para uma planilha eletrônica.

Tela de Ajuda

Nesta tela, o usuário poderá ter acessos aos documentos explicativos do sistema. O modelo da tela é:

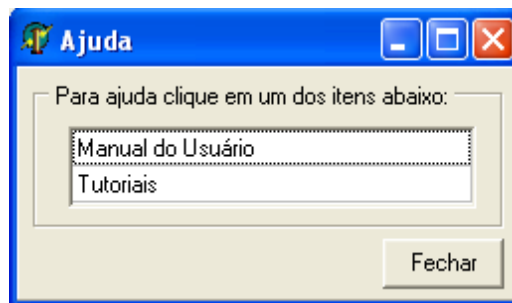


Figura 25 – Tela de Ajuda

Nesta tela, o usuário pode acessar este Manual do Usuário e o arquivo de Tutoriais, que exemplifica por meio de planilhas confeccionadas no Calc as várias funcionalidades do sistema.

Tela “Sobre”

Esta tela é bastante simples e apenas informa ao usuário que versão do sistema ele está utilizando e em que data esta versão foi liberada. A tela tem o *layout* abaixo:



Figura 26 – Tela de Exibição de Versão.

Tela de *Logs* de Erro

Em caso de geração de erro em alguma das telas, o sistema exibirá mensagens de *log* para facilitar a vida do usuário. Segue o formato do arquivo de *log*:

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <LOGSISTEMA>
    <LOG Data="2008-09-16"/>
    <LOG Data="2009-09-01">
      <Mensagem Unit="SerieHistorica">Erro : Cotação Inválida para Gerdau
        PN</Mensagem>
      <Mensagem Unit="Curva">Erro : Cotação Inválida para Ações
        Gerdau</Mensagem>
    </LOG>
  </LOGSISTEMA>
```

O arquivo se inicia com um elemento LOGSISTEMA, que delimita o início e fim do arquivo de *log*. Em seguida, temos os vários *logs* separados por datas. Esta data é a data do sistema de quando o *log* foi gerado. O elemento *log* é composto por uma série de mensagens, que informam ao usuário em qual módulo o *log* foi gerado e uma mensagem de erro explicativa.

Este sistema de *log* facilita muito o processo de depuração de erros ao mesmo tempo que ajuda o usuário na utilização do software, pontuando erros com mensagens claras e informando o módulo onde o erro ocorreu.

Segue abaixo a tela de exibição de *log* do sistema:

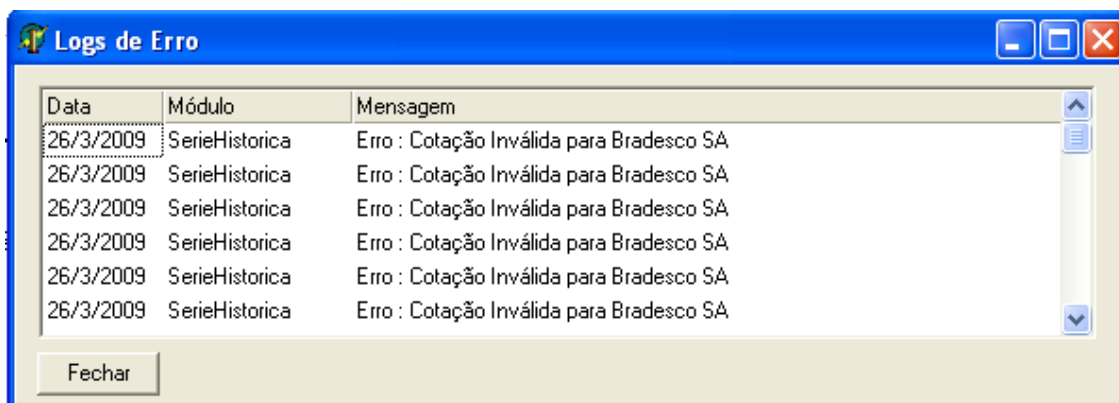


Figura 27 – Tela de Exibição de *Logs* de Erro.

Este *log* foi gerado devido à inserção de cotações inválidas em uma das séries do sistema. Cabe ressaltar que a mensagem de erro é gerada em tempo de execução. No caso acima, por exemplo, podemos observar o nome da série que gerou o erro. Isso facilita muito o trabalho de correção de um eventual problema e alerta o usuário sobre possíveis problemas nos dados.

O formato de exibição do *log* é sempre como o mostrado acima: dividido em 3 colunas com data, módulo e mensagem. Na mensagem, sempre que possível foi incluído o nome do elemento do sistema que gerou o erro.

Enfim, chegamos ao final deste manual na esperança de ter conseguido esclarecer as dúvidas quanto à utilização do software. Caso reste alguma dúvida sobre alguma parte da teoria ou sobre a estrutura do sistema, favor consultar a bibliografia ou escrever para lissandro@gmail.com.

Um bom proveito!!!

Lissandro Rocha