**Software desenvolvido para bolsa de valores**

**Allen Hichard Marques dos Santos**

Engenharia de computação – Universidade Estadual de Feira de Santana (UEFS)   
Caixa Postal 252 – 294 – Feira de Santana – BA – Brasil

Terceira solução de caso PBL

[allenhichard@hotmail.com](mailto:allenhichard@hotmail.com)

***Abstract.*** *A consultem firm called TreeStock consulters stock market made ​​a request for a software that assists in their administrative activities. The company works in the stock market managing and providing advice on the amount of shares and values ​​of their customers.*

***Resumo.*** *Uma empresa de consultoria de mercado de ações chamada TreeStock Consulters fez um pedido de um software que auxilie nas suas atividades administrativas. A empresa trabalha no mercado acionário administrando e prestando consultoria na quantidade e valores das ações dos seus clientes.*

* **Introdução**

Um dos mercados mais imprevisíveis é o da bolsa de valores, principalmente por suas instabilidades referentes aos valores de suas ações. A empresa TreeStock Consulters preocupada com seus clientes fez um pedido de um sistema rápido para atender a toda demanda, isso porque precisa agir de forma rápida no gerenciamento dos papeis acionários de seus clientes.

O software tem a finalidade de garantir agilidade de busca, inserção e remoção, tanto de cliente quanto de ações, para garantir uma maior precisão nas atualizações no mercado da bolsa de valores. No programa tem a solução para permitir atualizações instantâneas dos valores das ações e atualizações das quantidades e valores de cada carteira (conjunto de ações do cliente).

* **Metodologia**

O software foi desenvolvido em linguagem de programação “Java”, por ser uma linguagem de fácil interação entre o programador e o usuário. Desde a primeira sessão foram levantados os fatos sobre o funcionamento do programa e estabelecido uma meta para o desenvolvimento do problema, a antecipação da data de entrega.

Os principais fatos padronizados foram:

* Cadastrar o cliente organizando em ordem alfabética;
* Garantir a velocidade de busca, remoção e inserção pedida do problema (log n base 2);
* Os métodos também deverão suportar tratamento de exceções.

Fatos esses que facilitou o melhor entendimento do programa, e garantiu uma maior agilidade no seu processo de construção. No decorrer de cada sessão foi discutida a melhor forma de desenvolvimento para o esclarecimento de cada dúvida. O teste de unidade foi feito para sanar todas as dúvidas existentes até o último decorrer das sessões.

Os testes foram feitos pensando tanto na construção da interface para o usuário, quanto para o completo funcionamento do programa, ou seja, testado tudo que pode ser feito em uma bolsa de valores. Pelo teste não foi detectado erro referente a todos os casos possíveis ao decorrer da execução do sistema, sendo assim é garantido à confiabilidade do funcionamento do programa.

Para melhor desenvolvimento do sistema, a TreeStock Consulters preparou um teste de aceitação (teste esse, feito pela empresa sobre todos os requisitos que serão utilizado na bolsa de valores), sendo assim sanado todas as implementações necessárias para o desenvolvimento do programa, garantindo toda sua funcionalidade.

**Requisitos/funcionalidades para o software**

O programa tem todas as funcionalidades de uma bolsa de valores. Uma ação é um tipo de papel para um determinado cliente, cada cliente pode ter vários papeis, o que caracteriza uma carteira de papeis, que podem pertencer a várias empresas no mercado de ações da bolsa de valores.

Uma ação tem um valor não constante, ou seja pode variar a depender do dia, sendo os valores variáveis, foi implementado de forma que a TreeStock Consulters possa monitorar cada cliente e sua carteira de forma rápida, independente de quantas vezes cada ação mude de valor durante o dia, caso ocorra uma mudança o sistema automaticamente recalcula a quantidade e o valor das carteiras que tiver a determinada ação que sofreu com os altos e baixos da bolsa.

Os clientes foram organizados em ordem alfabética, para manter um maior controle das ações, após a ordenação por ordem alfabética, foi organizado a tabela por melhores carteira, sendo possível a empresa tratá-los de maneira VIP(Very Important People). Contudo a empresa ainda pode utilizar desse meio para tratar os TOP-K de cliente, informando um número, o sistema responderá todos os melhores e retornar até a quantidade informada.

Todas as funcionalidades terão tratamentos de exceções, para que o programa não encerre apenas quando um dado não é encontrado, sendo assim o programa poderá ser mais facilmente manuseado.

**Ordem de codificação (Desenvolvimento Teórico)**

A construção do algoritmo foi iniciada criando todos os testes de unidade, montando um teste com uma maior qualidade e confiabilidade, como por exemplo, foi pensando em todas as funcionalidades que o programa teria que ter, após essa análise foram feitos os teste para cada tipo de situação correspondente.

Após a criação de todos os testes das funcionalidades do programa, a implementação foi iniciada, como o teste de aceitação já estava desenvolvido, e os testes de unidades já estavam funcionando, foi começado o desenvolvimento das funcionalidades do programa, baseando-se no teste de aceitação contido no problema

Com todas as classes e métodos completamente criados, foi rodado o teste e verificado os erros de cada situação, com isso todos os erros constados no sistema foram completamente corrigidos.

Desenvolvimento teórico: O grande diferencial foi a utilização das Tabelas Hash, Lista Ordenada, e árvore de busca binária balanceada (AVL), e os tratamentos de exceções que permitem um completo funcionamento do programa.

O software foi desenvolvido em um sistema operacional Windows, foi utilizado o Netbeans que é um interpretador de códigos de linguagem de programação “Java”. O Diagrama de Classes foi feito utilizando o software da Star-UML, que é um programa que constrói UML manualmente ou a partir de um código fonte já desenvolvido gerando-o automático.

* **Resultados e discussões**

O programa garante o completo funcionamento do que foi pedido. O teste de unidade foi feito pensando em múltiplas situações, inclusive situações de erros. O teste de aceitação contido no projeto, foi executado perfeitamente após a implementação das funcionalidades, sendo assim alcançado o pedido da TreeStock Consulters.

Referente ao prazo, foi discutido uma renovação antecipando a entrega do produto, que ficará a espera de uma interface gráfica.

**Saídas do programa**

O programa não tem nenhuma saída referente ao usuário, mas foram implementados funcionalidades que serão saídas para o usuário após a criação da interface. Saídas essas:

* Mostra um determinado número de carteiras dos clientes por ordem dos valores maiores para os menores.
* Lista as ações/quantidades da carteira de um determinado cliente.

Essas saídas irão ficar a critérios da equipe de interface. A equipe da interface que vai fazer essa utilização de forma conveniente, mas em codificação as saídas já estão prontas.

**Testes efetuados e resultados obtidos**

* O teste de aceitação que testa a funcionalidade do problema que já veio contido no mesmo (Foram obtidos 100% dos critérios para o desenvolvimento);
* Ordenação por ordem alfabética;
* Ordenação por TOP-K cliente pelo valor da carteira em ordem decrescente;
* O balanceamento da Árvore, teste esse que imprime a altura das folhas direita e esquerda, sendo assim a diferença ficou sempre entre 0 e 1, diferença aceitável para a árvore AVL.
* O retorno do valor, depois do armazenamento de uma chave na tabela.

**Em que situação o seu programa não funcionaria**

Todos os requisitos estão funcionando por completo. De acordo com os testes de unidade e aceitação (contido no problema) o sistema não ocorrerá falhas.

* **Conclusão**

Após todos os testes realizados, todo o funcionamento que o problema exigiu foi cumprido, os cadastros de clientes e ações, as suas agregações(ações inseridas na carteira de cliente), a forma de listar e modificar ações, e principalmente atualizações ao decorrer da bolsa de valores, todos os critérios atendidos.

Todo o código-fonte está completamente pronto, esperando a equipe que programará a interface gráfica, ficando em consideração que a equipe de interface fará a completa manipulação com a programação já desenvolvida.

As melhorias supostas seriam:

* Desenvolver uma interface linha de comando e gráfica, sendo a gráfica de forma interativa com o usuário;
* Cada Cliente ter uma variedades de carteiras;
* Cliente poder comprar ações diretamente com outros cliente (negociação).
* Cliente poder assumir diferentes tipos de papeis na bolsa de valores.

**Referência**

Apostila Caelum, Java e Orientação a Objetos. http://www.caelum.com.br/apostila-java-estrutura-dados/ e <http://www.caelum.com.br/apostila-java-orientacao-objetos/>. Cada apostila tem uma finalidade: um voltado a programação e outra só estrutura de dados.

Autor Lafore, Robert. Título: Estruturas de dados e algoritmos, segunda edição.