



DESENVOLVIMENTO DE UM SISTEMA CONTROLADOR DE ESTOQUE COM CONCEITOS DE SISTEMAS DISTRIBUÍDOS

Rodrigo Mota Lacerda, Lucas Araújo Borges, João Paulo Pereira Nery, Luanna Ferreira Neves, Joana Gabriela Ribeiro de Souza, Amanda Gabriely Santos Alves, Jhonatan Moraes Dias

Introdução

O controlador de estoque proposto neste trabalho é um sistema distribuído que tem como objetivo principal otimizar as operações de cadastro, atualização, exclusão e consulta de produtos em estoque de unidades de uma empresa. O projeto aqui proposto não contempla todas as funções de um controlador de estoque que é comercializado, mas se apresenta como um modelo simplificado de um terminal de manipulação de dados referentes a um estoque.

A concepção do mesmo se deu especialmente para diminuir o tempo de resposta gasto nas operações que anteriormente eram realizadas via telefone. Esta aplicação seria destinada a lojas com mais de uma filial, onde o vendedor poderá identificar de forma rápida em quais filiais um determinado item se encontra.

Material e métodos

As aplicações distribuídas podem consultar uma mesma base de dados independentemente da localização geográfica do servidor. Para explanação do desenvolvimento, torna-se preciso a conceituação de alguns tópicos:

A. Sistemas distribuídos

Coulouris, Dollimore e Kindberg [1] expõem o sistema distribuído sendo aquele no qual os componentes de hardware e software, localizados em uma coleção de computadores interligados em rede, se comunicam e coordenam suas ações apenas enviando mensagens entre si. Utilizando-se deste princípio, o sistema distribuído desenvolvido se comunica e mantém-se sempre atualizado quando alguma ação for realizada.

B. Cliente - Servidor

O funcionamento de aplicações web consiste no modelo cliente-servidor. Nas palavras de Kurose e Ross [3], “Um programa cliente é um programa que funciona em um sistema final, que solicita e recebe um serviço de um programa servidor, que funciona em outro sistema final”. No entendimento desses autores, as interações entre os programas clientes e os programas servidores ocorrem por envio mensagens através da Internet. Destaca-se que os papéis de cliente e servidor geralmente não são fixos.

C. Sockets

“A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to”, Oracle [4]. Por definição dos sockets adotam o paradigma cliente-servidor. O computador servidor é o responsável por abrir os sockets e fica na escuta, a espera de mensagens e pedidos das aplicações clientes.

D. Threads

“Threads share the process's resources, including memory and open files. This makes for efficient, but potentially problematic, communication”, Oracle [5]. As threads são importantes componentes utilizados no sistema, pois possibilitam a multiprogramação, fator imprescindível ao tipo de aplicação desenvolvida.

E. Replicação

A replicação tem como objetivo a sincronização de dois ou mais servidores de banco de dados com o objetivo de garantir que os dados estejam disponíveis em dois ou mais servidores, entendem Elmasri e Navathe [2]. Essa técnica possui vantagens como a disponibilidade, pois caso um servidor pare o outro continua funcionando e uma menor sobrecarga, visto que as alterações e consultas aos dados podem ser divididas entre os servidores.

F. Controlador de estoque



Controladores de estoque geralmente são sistemas que fazem acompanhamentos sobre situações do estoque de uma determinada empresa. Há vários sistemas de controle de estoque comercializados atualmente, cada um com diferenciais, como: permitir acompanhamento sobre pedidos de compra, emitir alertas sobre quantidade mínima de estoque, fazer cotações de preços com fornecedores, tudo isso geralmente integrado com outros sistemas das empresas.

O sistema desenvolvido neste trabalho, representa uma forma simplificada apenas para consulta de informações sobre determinados produtos do estoque. Entretanto, deve-se destacar que os fundamentos utilizados para realização do mesmo servem como base para o desenvolvimento de novas funcionalidades para esta aplicação, e alterações futuras podem ser facilmente realizadas.

Desenvolvimento

O Gerenciador de Estoque foi desenvolvido por meio da IDE NET BEANS com a linguagem de programação JAVA, utilizando-se 5 aplicações, sendo dois servidores e três clientes. Para fundamentar a base de dados, foi usado como Sistema Gerenciador de Banco de Dados MySQL, através do WAMPSEVER. O processo de replicação foi feito pelo phpMyAdmin, sendo do tipo Master-Master, que permite a atualização dos dados vinda do primeiro servidor para o segundo servidor e vice-versa. A arquitetura do sistema é representada pela figura 1. Nela é exposto os servidores e os clientes (lojas), sendo possível cada máquina encontrar-se em locais geográficos diferentes.

O banco de dados contém somente uma tabela com quatro campos, conforme a figura 3, desenvolvida pelo software Astah. No campo id é guardado o id produto. Em produto, o nome do produto. Já em quantidade, a quantidade do produto. Em loja, o nome da loja. A interface do sistema tem o objetivo de interagir com o usuário e fornecer dados necessários para fazer uso das funções do sistema. Esta interface é representada pela figura 2.

No sistema proposto foram desenvolvidos métodos que realizam as funções: cadastro de produtos em uma loja específica, atualização de informações, exclusão de produtos, consulta sobre determinado produto cadastrado em uma só loja ou em todas as lojas. O sistema é composto pelas seguintes classes: Cliente, ConnectionFactory, ProdutoDAO, TCPServidor, TCPCliente e TCPServidorThread.

A classe Cliente contém a interface gráfica do sistema. Nela se encontram os métodos responsáveis pela estruturação da interface, bem como de captar os dados preenchidos nela e exibir as respostas vindas do servidor.

Na classe ConnectionFactory se encontra o método responsável por declarar o driver de conexão do banco de dados e retornar uma conexão para a classe ProdutoDAO utilizar.

Na classe ProdutoDAO são realizadas as operações com o banco de dados. O objeto ProdutoDAO identifica qual operação deve ser realizada e retorna as mensagens. Nesta classe, foram implementados métodos de controle de concorrência do banco de dados.

Na classe TCPCliente é criado o socket cliente para a conexão com o servidor principal, caso falhe, será criado um novo socket para o servidor secundário. Ainda nesta classe, se encontra a estrutura que permite a leitura e envio de uma stream para o servidor, além de implementar envio de uma mensagem de retorno.

Na classe TCPServidor é criado o Socket Servidor e inicializada a Thread cujas operações se encontram na classe TCPServidorThread, além de criar um novo socket para implementar as estruturas que fazem leitura de streams vindas do cliente e fazem o retornar as informações. Nesta classe é criado o objeto ProdutoDAO,

Resultados

As consultas entre cliente e servidor se mostraram rápidas e eficazes, garantindo total confiabilidade, alta disponibilidade e integridade das informações armazenadas no banco de dados. Além disso, foi garantido a alta disponibilidade visto que os servidores são replicados. Foram feitos testes com o servidor principal parando de funcionar e também outros testes com servidor secundário parando de funcionar.

Também foram realizados testes quanto aos tratamentos de exceções e das operações do banco de dados. Por fim, a aplicação cumpre com objetivo para o qual foi planejada, de tal modo que se torna recomendada para as empresas que implementarão um software desta natureza visando apenas um terminal básico de consultas.

Conclusões



A aplicação desenvolvida proporciona uma concentração de informações em duas únicas fontes de dados, o que possibilita uma fácil busca e manipulação além de possibilitar e a redundância dos dados. Através de seu uso torna-se desnecessário o uso de várias bases de dados e, mesmo assim, possibilita o uso simultâneo por vários clientes.

Referências

- [1] COULOURIS G., DOLLIMORE J., KINDBERG T. **Sistemas Distribuídos: Conceitos e Projeto**. Bookman. Quarta edição. 2007.
- [2] ELMASRI R., NAVATHE Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.
- [3] KUROSE, James F., ROSS Keith W. **Redes de computadores e internet: uma abordagem top-down**. 5. ed. São Paulo: Addison Wesley, 2010.
- [4] ORACLE. **Lesson: All About Sockets**. Disponível em <<http://java.sun.com/docs/books/tutorial/networking/sockets>>. Acesso em: 02 julho 2015.
- [5] ORACLE. **Lesson: Concurrency**. Disponível em <<https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>>. Acesso em: 02 julho 2015.

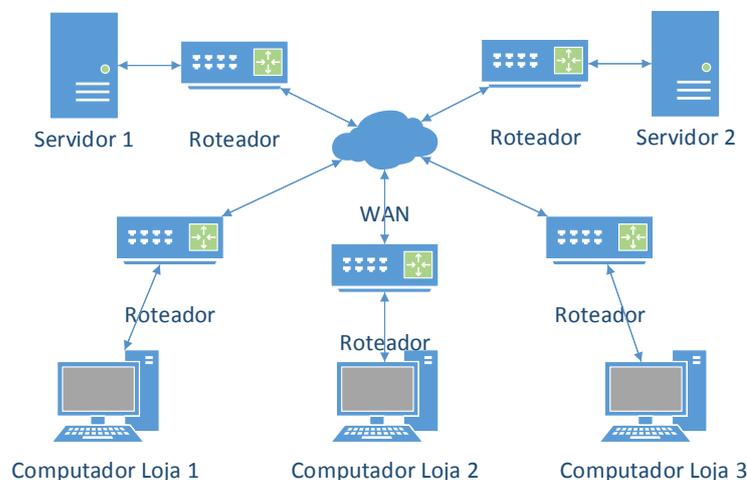
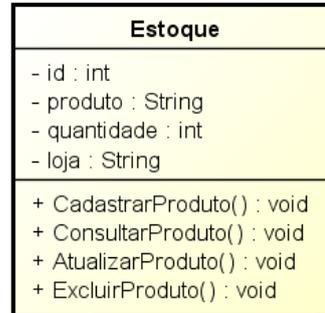


Figura 1. Arquitetura do sistema desenvolvido. Fonte própria, 2015.

Figura 2. Interface do sistema desenvolvido. Fonte própria, 2015.



powered by Astah

Figura 3. Diagrama de Classe do sistema desenvolvido. Fonte própria, 2015.