

Aplicação da Tecnologia Java no Desenvolvimento de Sistema de Controle de Estoque

Dalton Takayuki Kanematsu¹, Sergio Silva Ribeiro¹

dtk230790@gmail.com, professor@sergioribeiro.com.br

Abstract. *The present work aims to develop a system for the company Royale Lanches LTDA, Guarapuava – PR. Due to a deficiency in its current inventory control system for the sector of production and by this accomplish great movements of stocks. The system aims to expand and improve accuracies on inventory information; however to develop the software was used as the programming language, Java to be object-oriented, multi-platform; with Hibernate, JDOM, iReport, and MySQL relational database technologies. And to develop this work it was necessary extensive review of literature on the state of the art in software engineering.*

Keywords: Inventory Control, System, Software, Java.

Resumo. O presente trabalho tem por objetivo desenvolver um sistema para a empresa Royale Lanches LTDA, do Município de Guarapuava - PR. Em função de uma deficiência em seu atual sistema de controle de estoques para o setor de produção e por esta realizar grandes movimentos de estoques. O sistema tem como objetivo ampliar e melhorar precisões sobre informações de estoque; no desenvolvimento do Software foi utilizada a linguagem de programação *Java*, por ser orientado a objetos e multiplataformas; com as tecnologias *Hibernate*, *JDOM*, *iReport*, e o banco de dados relacional *MySQL*. E para o desenvolvimento deste trabalho fez-se necessário ampla revisão de bibliografia referente ao estado da arte em Engenharia de Software.

Palavras-chave: Controle de Estoques, Sistema, Programa, Java.

Introdução

A gestão empresarial como um todo é de extrema importância para o sucesso de uma empresa e esta pode estar relacionada a departamentos específicos. Sendo o de controle de estoque e sua gestão adequada fundamentais para uma empresa, independentemente do porte, pois melhoram a capacidade de prever a quantidade de matérias primas a serem adquiridas. Também traz benefícios como informações sobre

¹ Tecnologia em Análise e Desenvolvimento de Sistemas – Faculdade Guairacá

movimentações de saídas, produto mais vendido, lucro, quantidade mínima existente e necessária (BALLOU, 1993).

A Royale Lanches LTDA é uma empresa do ramo alimentício, localizada no Município de Guarapuava PR, e atua no mercado há 16 anos. A empresa segue uma política de inovação e qualidade para seus produtos e está sempre recebendo técnicos, treinando seus funcionários, assim como participando de feiras e eventos gastronômicos. E uma das suas grandes preocupações é a gestão adequada do seu estoque. A empresa enfrenta problemas como perda de tempo para a contagem de estoques e baixa precisão nas informações sobre o mesmo.

O atual avanço tecnológico oferece um vasto número de recursos que auxiliam no processo de gestão empresarial. Estas soluções podem ser de *hardware* e *software*. E para o auxílio no processo de gestão e tomada de decisão são utilizados softwares específicos. As tecnologias de informações compreendem o *hardware* e *software* destinados a tarefa de processamento de dados e que capturam, manipulam, armazenam, recuperam e transmitem as informações. Salienta que a tecnologia da informação é um componente do sistema de informação e não pode ser confundido como se este elemento fosse o todo. A tecnologia de informação é o ferramental do qual se utiliza a prática de trabalho (ALTER, 1992).

O objetivo deste trabalho mostra o desenvolvimento de um *Software* de gestão de estoque para a empresa Royale, e para a sua realização foi feita uma vasta revisão bibliográfica sobre assuntos relacionados a gestão de estoque, Engenharia de *Software*, Programação Orientada a Objetos, Linguagem de Programação e Banco de Dados.

Fundamentação Teórica

A administração de estoques é de essencial importância para as organizações, principalmente devido a carência existente que tem gerado prejuízos, insatisfação do cliente entre outros problemas. Também é responsável por gerar benefícios, por isso é importante que ele seja bem gerenciado e correto. Precisão nas quantidades, valores, tudo isso a dispor do gestor para que possa administrar a movimentação da empresa e manter o cliente satisfeito (DIAS, 1993).

Todas quantidades de bens físicos que conservados, de forma improdutivo, por algum intervalo de tempo são considerados estoques e constituem tanto os produtos acabados que aguardam venda ou despacho, como matérias-primas e componentes que aguardam utilização na produção. Trabalham como reguladores, garantindo estabilidade nas vendas da empresa tornando-se assim necessidade extrema para ela. O estudo dos estoques é complexo, por envolver diversos setores dentro da organização, por isso, os estoques são considerados causadores de problemas. Por outro lado, são denominados geradores de lucro e de trabalho (MOREIRA, 1996).

Ainda segundo MOREIRA (1996) o estoque de uma empresa é responsável pelo funcionamento dela, podendo proporcionar lucros e recursos para a mesma. É um grande gerador de trabalho e necessita de muito cuidado com sua gestão, por estar vinculado com vários setores dentro da empresa.

Engenharia de *Software* é a criação e a utilização de sólidos princípios a fim de obter *Softwares* econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais. Ela abrange um conjunto de três elementos fundamentais métodos, ferramentas e procedimentos – que possibilitam ao gerente o controle do processo de desenvolvimento do *Software* e oferece ao profissional uma base para a construção de *Software* de alta qualidade e produtividade (PRESSMAN, 2005).

Danfe é uma representação gráfica simplificada da Nota Fiscal Eletrônica (NF-e), em papel comum e em via única. Em destaque, contém a chave de acesso com 44 números para consulta da NF-e na Internet, e um código de barras bi-dimensional que facilitará a captura e a confirmação de informações da NF-e pelas unidades fiscais. No site da NF-e é possível gerar o *XML* é necessário a empresa possuir o certificado digital. Posteriormente pode utilizar o documento gerado para facilitar a entrada de estoques em sistemas (SECRETARIA DA FAZENDA SP, 2014).

O ciclo de vida de um projeto de sistema é o modo como o projeto é desenvolvido na empresa, e é uma maneira eficiente para que as pessoas envolvidas no processo de desenvolvimento de sistemas possam entrosar-se com o projeto a ser desenvolvido. Os principais modelos de ciclo de vida são; Modelo Clássico (ou em Cascata); Prototipagem (ou Prototipação); Espiral; Incremental; *RAD* (YORDON,1990).

Prototipação define-se como um sistema que ocorre através da descoberta gradual e evolutiva deste por parte do usuário e do desenvolvedor; obtendo-se um conjunto inicial de necessidades implementando rapidamente e refinando-as de acordo com o aumento do conhecimento do sistema por parte do desenvolvedor e do usuário (YORDON, 1990).

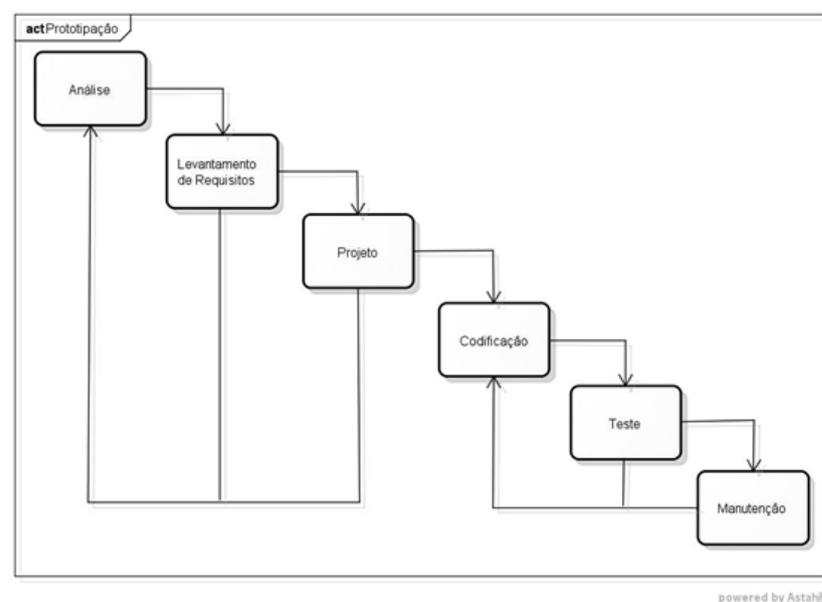


Figura 1. Modelo de Prototipação.

A cada ciclo são gerados os chamados "Protótipos Funcionais" e alguns são mais compreensivos e modelam os sistemas altamente complexos; outros desenvolvem sistemas pequenos e relativamente simples. Cabe ressaltar que uma distinção deve ser feita entre o protótipo e um sistema real, no qual têm a intenção de uso operacional e devem seguir determinados padrões quanto à qualidade, segurança, desempenho, capacidade, robustez e facilidade de manutenção. Os protótipos visam clareza na visualização de determinados aspectos de um sistema sobre os quais há incerteza (YORDON, 1990).

O Diagrama de Casos de Uso é construído para modelar as interações entre um usuário externo do sistema, um objeto ou um conjunto de comunicação do sistema. O Diagrama de Classes é construído para atender alguma necessidade e tem requisitos que precisam ser observados no momento de sua utilização e construção. No Diagrama de sequência, o objetivo é determinar um seguimento de eventos que ocorrem em um

determinado processo, listando os métodos que devem ser disparados entre os atores e objetos envolvidos seguindo uma ordem que devem ocorrer (PRESSMAN, 1995).

O *NetBeans IDE* é um ambiente de desenvolvimento que possui vários utensílios úteis para os programadores, que permite escrever, compilar, depurar e instalar programas. O *IDE* é inteiramente escrito em *Java*, mas permite a utilização de qualquer linguagem de programação. Encontra-se também um grande número de módulos para acrescentar as funcionalidades do *IDE NetBeans*. Também é um produto gratuito, sem restrições à sua forma de usar. Também está acessível a esta plataforma uma base modular e extensível que pode ser utilizada como infra-estrutura para a geração de grandes aplicações de *desktop*. Os Parceiros concedem *plugins* que podem ser facilmente integrados na plataforma, e que permitem ser usados para desenvolver ferramentas e soluções próprias (NETBEANS, 2014).

O *XAMPP* é um pacote de instalação automática que consente o computador *local* pronto para gerir páginas Web. O *Software* inclui o servidor *Apache*, o *MySQL* e o *PHP*, além de outras linguagens e recursos, como um cliente *FTP*. O objetivo do *XAMPP* é construir uma distribuição fácil de instalar para desenvolvedores entrarem no mundo do *Apache*. Para torná-lo conveniente para os desenvolvedores, o *XAMPP* é utilizado atualmente para servir sites *Web* na *WWW*, e com algumas modificações é geralmente seguro para uso em servidor público. Uma ferramenta especial é incluída para proteger facilmente as partes mais importantes e sensíveis do pacote. O *XAMPP* possui muitos aplicativos, dentre eles o *Apache*, *MySQL*, *phpMyAdmin*, *FileZilla FTP Server*, *OpenSSL*, e sua principal vantagem é a versatilidade, em alguns minutos é instalado todos esses programas (SEDLER, 2014).

Extensible Markup Language (XML) é um conjunto de regras para a demarcação semânticos, que dividem um documento em partes identificáveis. É uma meta linguagem que determina uma sintaxe para ser usada na criação de outras linguagens de marcação para um domínio específico, com estrutura e semânticas próprias. Além disso, *XML* pode também ser definida como uma linguagem de meta-marcação que, como a *HTML*, inclui dados entre marcadores. Entretanto, elas possuem propósitos diferentes: *HTML* é uma linguagem para apresentação e *XML* é uma linguagem para descrição de dados estruturados. Em outras palavras, os marcadores *XML* estão relacionados ao

significado do texto delimitado por eles ao passo que os marcadores *HTML* especificam como os dados serão apresentados (MORALES, 2003).

```
<NFe>
- <infNFe Id="NFe41140378332293000100550030003155451141531518" versao="2.00">
- <ide>
  <cUF>41</cUF>
  <cNF>14153151</cNF>
  <natOp>Venda Mercadoria</natOp>
  <indPag>0</indPag>
  <mod>55</mod>
  <serie>3</serie>
  <nNF>315545</nNF>
  <dEmi>2014-03-01</dEmi>
  <dSaiEnt>2014-03-03</dSaiEnt>
  <tpNF>1</tpNF>
  <cMunFG>4109401</cMunFG>
  <tpImp>1</tpImp>
  <tpEmis>1</tpEmis>
  <cDV>8</cDV>
  <tpAmb>1</tpAmb>
  <finNFe>1</finNFe>
  <procEmi>0</procEmi>
  <verProc>1.0.18</verProc>
</ide>
```

Figura 2 Exemplo de Xml.

A Programação Orientada a Objetos (POO) nos últimos anos tem se tornado a tecnologia de programação dominante, e seus benefícios são vastamente reconhecidos (CHAVEZ, 2004). O mercado de *Software* à procura de qualidade tem levado ao aumento do uso da OO e elevado os níveis de reuso e manutenibilidade dos *Softwares*, aumentando a produtividade do desenvolvimento e suporte a mudanças de requisitos. Os conceitos básicos do paradigma orientado a objetos (modo de fazer as coisas) são relativamente fáceis de apreender e de aplicar. Quando se está tentando solucionar um problema do mundo real pode-se perceber diversas abstrações circundando o problema (O'DOCHERTY, 2005).

Quanto às linguagens de programação para o desenvolvimento, foi utilizado *Java*, com as tecnologias *Hibernate*, *JDOM*, *iReport*, e o Banco de Dados relacional *MySQL*, na plataforma de desenvolvimento foi utilizado o *NetBeans*. Também alguns conceitos e as ferramentas auxiliares que foram utilizadas no decorrer desse trabalho.

A linguagem de programação *Java* foi criada pela Sun Microsystems, em 1991 com, inicialmente, com objetivo de desenvolver *Softwares* para os computadores pessoais. Após a ampliação da *World Wide Web (WWW)*, em 1993, constatou-se a

alternativa de utilizar o Java para acrescentar conteúdo dinâmico às páginas *Web*. Além dessas funcionalidades, essa ferramenta também é empregada no desenvolvimento de Softwares de grande porte, além de prover aplicativos para dispositivos móveis (DEITEL, 2003).

Um programa desenvolvido em *Java* é formado de classes, sendo que cada uma delas possuem métodos responsáveis por concretizar tarefas e retornar dados ao concluí-las. Essas classes podem ser designadas diretamente pelo programador ou agregadas das bibliotecas de classe *Java* (*APIs* do *Java*). Essas bibliotecas são disponibilizadas por fornecedores do compilador ou até mesmo por outros desenvolvedores de *Software*, tornando possível a reutilização de uma classe, sem precisar recriá-la (DEITEL, 2003).

O *MySQL* é um Banco de Dados que pode armazenar e gerenciar os dados de um sistema; por ser *opensource* e seu bom desempenho popularizou-se entre os desenvolvedores, tanto para internet como para sistemas *off-line*. Este gerenciador pode ser organizado em tabelas, que armazenam informações em forma de linhas e colunas de dados relacionados. O servidor do Banco de Dados geralmente, roda junto a um servidor *Web* na mesma máquina, no qual os dois trabalharão juntos na leitura e escrita de dados e na entrega de páginas *Web* (MORRISON e BEIGHLEY, 2010).

Um *Framework* é um conjunto de componentes que facilitam o desenvolvimento de certas tarefas (MELO, 2007). Neste trabalho utilizou-se o *iReport / JasperReports* para geração de modo dinâmico de relatórios, compatível com formatos *XML*, *PDF* e *HTML*, *Hibernate* para persistência com Banco de Dados, além do mapeamento objeto-relacional e *JDOM* para a leitura de *XML*.

JasperReports é um poderoso *framework* *opensource* para geração de relatórios. Essa biblioteca foi desenvolvida utilizando a linguagem *Java* e possui grande capacidade na organização e apresentação de conteúdo, permitindo a geração dinâmica de relatórios em diversos formatos, como *PDF*, *HTML*, *XLS*, *CSV* e *XML*, podendo ainda ser utilizada em qualquer aplicação *Java*, incluindo aplicações *desktop*, *Web* e distribuídas. Escrever o layout utilizando o formato *XML* significa uma tarefa longa, dolorosa e pouco compensadora. Por isso surgiram algumas alternativas disponíveis que são muito mais fáceis de usar. A melhor de todas, é o uso de um editor visual para

projetar, compilar e testar os relatórios. Um dos editores visuais mais úteis que podem usar é o *iReport*. Ele é um aplicativo gráfico, que permite o *design* de relatórios, utilizando uma palheta, e arrastando e soltando componentes, de forma bem parecida com a criação de interfaces e janelas para programas (JASPER, 2014).

O *Hibernate* pode ser definido como um serviço de mapeamento objeto/relacional que proporciona aos desenvolvedores que utilizam a linguagem *Java* uma maneira de trabalhar as informações de um banco de dados relacional de forma eficiente e com facilidade. Com o uso do *Hibernate* no desenvolvimento em *Java* tem-se vários benefícios. O primeiro desses benefícios está no aumento da produtividade, pois o desenvolvedor não se preocupa com grande parte do trabalho pesado, focando no problema do negócio. A manutenção do sistema também é facilitada pela diminuição do código-fonte, tornando o sistema mais compreensível (ELLIOT, O'BRIEN E FOWLER, 2009).

JDOM é um modelo *opensource* de uma biblioteca de manipulação de documentos *XML* projetado especificamente para a plataforma *Java*. *JDOM* é utilizado em ambientes de desenvolvimento mais integrados para desenvolvimento de *Software Java*. *JDOM* foi desenvolvido no início de 2000, e traz um meio consistente para criar, escrever, ler e manipular dados *XML*, podendo construir uma nova árvore na memória. Os dados para se construir uma árvore podem vir de uma fonte que não seja *XML* como um banco de dados. Toda a árvore pode ser lida e escrita, dessa maneira não existe seção na árvore das quais não se pode modificar. Por ter sido criado em *Java* e ser usado em *Java* o *JDOM* provê um estilo de representar documentos *XML* para uma fácil e eficiente leitura, manipulação e escrita. Ele possui uma *API* leve, veloz e otimizada para programadores *Java*. É uma opção para *DOM* e *SAX*, embora consiga se integrar bem com as mesmas (SEDLER, 2014).

Materiais e Métodos

No desenvolvimento deste projeto realizou-se uma pesquisa bibliográfica de literatura relacionada para a construção e o desenvolvimento de *Software*, linguagens de programação e uso de requisitos funcionais para organização do sistema. Utilizou-se o *NetBeans* com a plataforma *Java* para o desenvolvimento, e a confecção dos diagramas de foi feita com o auxílio do programa *Astah*. Quanto ao banco de dados usou-se

MySQL o qual tem o *XAMPP* um programa que gerencia o servidor do Banco de Dados. E para a modelagem foi utilizado o programa *MySQL Workbench*.

Na implementação do sistema foi utilizado o *Framework JASPER REPORTS* para a construção dos relatórios, *Hibernate* para a persistência de dados e *JDOM* para a leitura de *XML*, proporcionando mais produtividade no ambiente de desenvolvimento. As etapas de desenvolvimento foram de acordo com o ciclo de vida de Prototipação para promover um melhor desempenho nos requisitos do sistema, no qual o levantamento de requisitos foi realizado com uma entrevista junto a administradora da empresa, onde levantou-se os requisitos e necessidades do *Software* para atender os processos repassados pela mesma, para que fossem possíveis o prosseguimento e o desenvolvimento do programa, Apêndice 1.

Em sequência ao embasamento teórico e estudos já realizados neste trabalho, foi desenvolvida o sistema, com um trabalho de ação com a construção de uma tela de acesso a empresa e aos seus produtos para gerenciamento de estoques.

Resultados

No Apêndice 2 temos o diagrama de caso de uso que foi gerado com base na análise de requisitos, que pode ser visto no Apêndice 1. Após o desenvolvimento do diagrama de caso de uso foi desenvolvido a modelagem lógica do banco de dados, que pode ser vista pelo diagrama que se encontra no Apêndice 3.

A tabela 1 apresenta um quadro comparativo do antes e depois da implantação do sistema. Por meio dela podemos compreender melhor os resultados obtidos com o desenvolvimento e implantação da aplicação.

Tabela 1 - Comparação Após a Implementação do Controle de Estoques

| Antes | Depois |
|---|---|
| Contagem de estoques eram feitas manualmente obtendo-se a quantidade sem a precisão, além de demorar muito tempo no processo. | Com os relatórios do sistema ficou mais fácil e rápido saber a quantidade existente no estoque. |
| Ocorriam a falta de matérias primas ou sobravam muitos, vencendo a data de | Com a funcionalidade quantidade mínima do sistema foi possível saber quais |

| | |
|--|--|
| validade. | produtos estavam em estado de alerta e precisavam ser adquiridos. |
| Sem controle de gastos e previsão de vendas. | Histórico completo de entradas e saídas de estoque, possibilitando a previsão de vendas. |

Na tabela 2 podem ser visualizados uma comparação entres os problemas apresentados e fixados ao longo das versões.

Tabela 2 - Comparação da Evolução do Sistema nas Versões

| | Problemas Apresentados | Fixação |
|-----------------|--|---|
| Versão 1 | No cálculo do valor total do produto a partir da multiplicação da quantidade por valor unitário estava tendo como resultado números quebrados. | Criado um método para permitir apenas duas casas decimais. |
| Versão 2 | Sistema de consultas muito simples, problemas na geração de relatórios. | Implementado mais opções de consulta, geração de relatório aprimorado e corrigidos problemas. |
| Versão 3 | Falta de uma funcionalidade para a identificação de produtos abaixo da quantidade mínima. | Implementado opção para determinar uma quantidade mínima do produto, e criado uma opção para demonstrar no relatório os produtos com quantidades abaixo do mínimo. |
| Versão 4 | Falta de sistema de <i>login</i> , e necessidade de correções de <i>design</i> . | Implementado sistema de <i>login</i> para melhor segurança, e design aperfeiçoado como a inserção de ícones em botões, e alinhamento de formulários e textos, e padronização das telas. |

Na tabela 3 uma síntese de aderência da versão final do sistema aos requisitos do cliente, conforme poderá ser verificado no Apêndice 4.

Tabela 3 - Quadro de Aprovação das Funcionalidades

| Funcionalidades | Aprovação da Administradora |
|------------------------|------------------------------------|
| Sistema de Cadastros | Atende |
| Entrada de Estoques | Atende |
| Saída de Estoques | Atende |
| Relatórios | Atende |

Considerações Finais

Este projeto teve como objetivo desenvolver Sistema para empresa Royale Lanches Ltda, do município de Guarapuava - Pr; pois a empresa ainda não possui um programa específico para o setor de produção. Considerando este problema produziu-se um *Software* para melhorar, a gestão de seus estoques; e a empresa como usuário administrador terá um controle adequado e acompanhamento da evolução dos consumos para a produção, que são críticos para a lucratividade e competitividade empresarial.

A construção deste sistema proporcionou um crescimento pessoal e intelectual na área do curso de Análise e Desenvolvimento de Sistemas, sendo um grande aprendizado para demais trabalhos futuros, e pesquisas de linguagens de programação.

Futuramente será modificado o sistema, incluindo novas telas e funcionalidades tal como; Receitas para que possam ser realizados um orçamento do custo de uma produção; Pedidos, nela poderá selecionar a receita e ser emitida em uma nota para ser utilizado na produção como referência dos serviços diário a serem realizados.

Referências

ALTER, Steven. **Information System: A Management Perspective**. United States of America: Addison – Wesley Publishing Company, 1992.

BALLOU, R. H. **Logística Empresarial**. São Paulo: Atlas, 1993.

BEIGHLEY, L. e MORRISON, M. **Use a cabeça! PHP & MySQL**. Rio de Janeiro: ALTA BOOKS Editora, 2010.

CHAVEZ, C. V. **Um Enfoque Baseado em Modelos para Design Orientado a Aspectos**. Pontifca Universidade Católica do Rio de Janeiro. Rio de Janeiro. Tese (Doutorado), 2004.

DEITEL, H. M. **Java, como programar**. Porto Alegre: Bookman, 2003.

ELLIOT, James; O'BRIEN, Timothy M.; FOWLER, Ryan. **Dominando Hibernate**. Rio de Janeiro: AltaBooks, 2009.

SECRETARIA DA FAZENDA, **Nota Fiscal Eletrônica, Perguntas Frequentes**. Disponível em https://www.fazenda.sp.gov.br/nfe/perguntas_frequentes/respostas_vi.asp. Acesso em 25 de Abril de 2014.

SMART, John Ferguson, **Java Reporting com JasperReports e iReport Open Source - Parte I**. Disponível em <http://www.devmedia.com.br/java-reporting-com-jasperreports-e-ireport-open-source-parte-i/1686>. Acesso em 08 de Abril de 2014.

MELO, Alexandre Altair de. **PHP Profissional: Aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto**. São Paulo: Novatec Editora, 2007.

MORALES, L. **XML e Java: uma Perspectiva do Programador**. Monografia de Especialização. Instituto de Informática – UFG, 2003.

NETBEANS. **O que é o NetBeans**. Disponível em https://netbeans.org/index_pt_PT.html. Acesso em 09 de Abril de 2014.

O'DOCHERTY, MIKE. **Object-oriented analysis and design: understanding system development with UML 2.0**. John Wiley & Sons Ltd : 2005.

PRESSMAN, Roger S. **Engenharia de Software**. Trad. José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

SEDLER, Munhoz Kai'Oswald. **Sobre**. Disponível em https://www.apachefriends.org/pt_br/about.html. Acesso em 10 de Abril de 2014.

VOLNEI. **Utilizando JDOM para ler e gerar XML**. Disponível em <http://javafree.uol.com.br/topic-847694-Utilizando-JDOM-para-ler-e-gerar-XML.html>. Acesso em 05 de Abril de 2014.

YORDON, Edward. **Análise Estruturada Moderna**. 3ª Ed. Trad. Dalton C. de Alencar. Rio de Janeiro: Campus, 1990.

Apêndice 1: Menu Funcionalidades

Em continuidade a descrição de cada funcionalidade criada para o sistema.

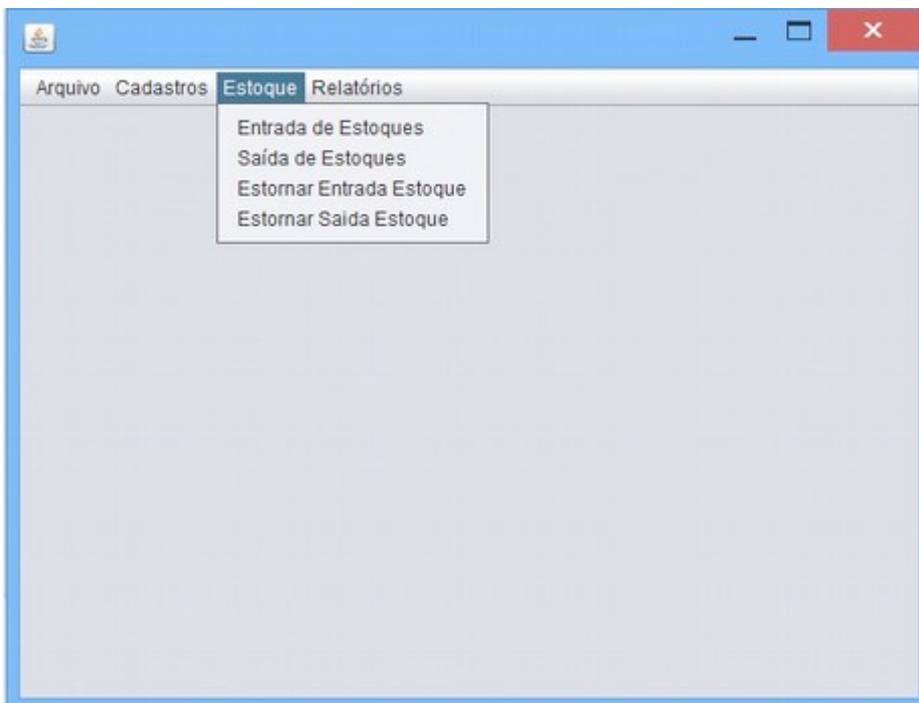


Figura 3 Tela de funções do Menu.

CADASTRAR PRODUTOS:

O sistema deverá permitir cadastrar novos produtos com todos os seus atributos (descrição, código alternativo, lv, ncm, etc.), relacionando de acordo com as suas características (grupo, LV, Fornecedor). No qual há a função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado. E a função CONSULTA de um produto já existente no banco de dados, utilizando como critérios a descrição, código alternativo, grupo e fornecedor.

CADASTRAR FORNECEDOR:

O sistema deverá permitir cadastrar novos fornecedores, podendo escolher o tipo da pessoa, física ou jurídica. No qual há a função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado. E a função

CONSULTA de um fornecedor já existente no banco de dados, utilizando como critérios a descrição.

CADASTRAR GRUPOS:

O sistema deverá permitir cadastrar novos GRUPOS. No qual há a função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado. E a função CONSULTA de um GRUPO já existente no banco de dados, utilizando como critérios a descrição.

ENTRADA E SAÍDA DE ESTOQUES:

O sistema permitirá a ENTRADA E SAÍDA DE ESTOQUES, selecionando os itens e preenchendo os formulários. No qual há a função ADICIONAR que adicionará um novo produto na tabela; RETIRAR que excluirá o produto selecionado da tabela; a função IMPORTAR XML, exclusivo da entrada de estoques para facilitar a tarefa; a função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro.

CADASTRAR CÓDIGO FORNECEDOR:

O sistema deverá permitir cadastrar novos CÓDIGOS DE FORNECEDORES, para associar com os produtos que será utilizado para a identificação dos itens do xml na importação. A função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado. E a função CONSULTA de um CÓDIGO FORNECEDOR já existente no banco de dados, utilizando como critérios o código.

CADASTRAR USUÁRIOS

O sistema permitirá cadastrar novos USUÁRIOS apenas para Administrador. A função SALVAR que salvará um novo registro do sistema no banco de dados; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado. E a função CONSULTA de um USUÁRIO já existente no

banco de dados, utilizando como critérios o código.

ESTORNO DE ENTRADA E SAÍDA DE ESTOQUES:

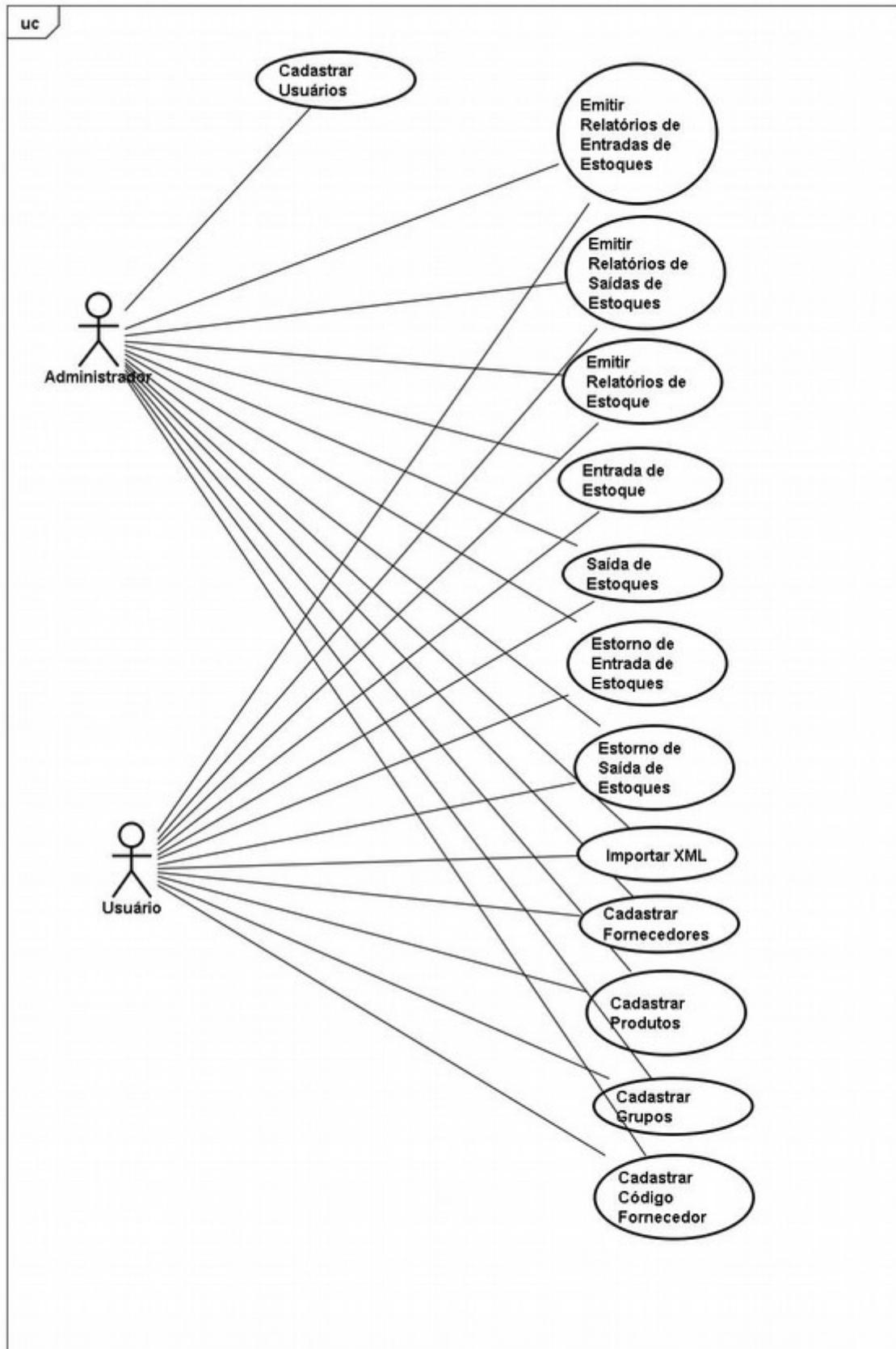
O sistema permitirá o estorno ENTRADA E SAÍDA DE ESTOQUES, localizando a partir da função CONSULTA onde poderá utilizar filtros como um período de data, número da nota ou fornecedor; a função DELETAR que eliminará o registro e a função ALTERAR que dará a possibilidade de mudar seu dado cadastrado.

RELATÓRIOS:

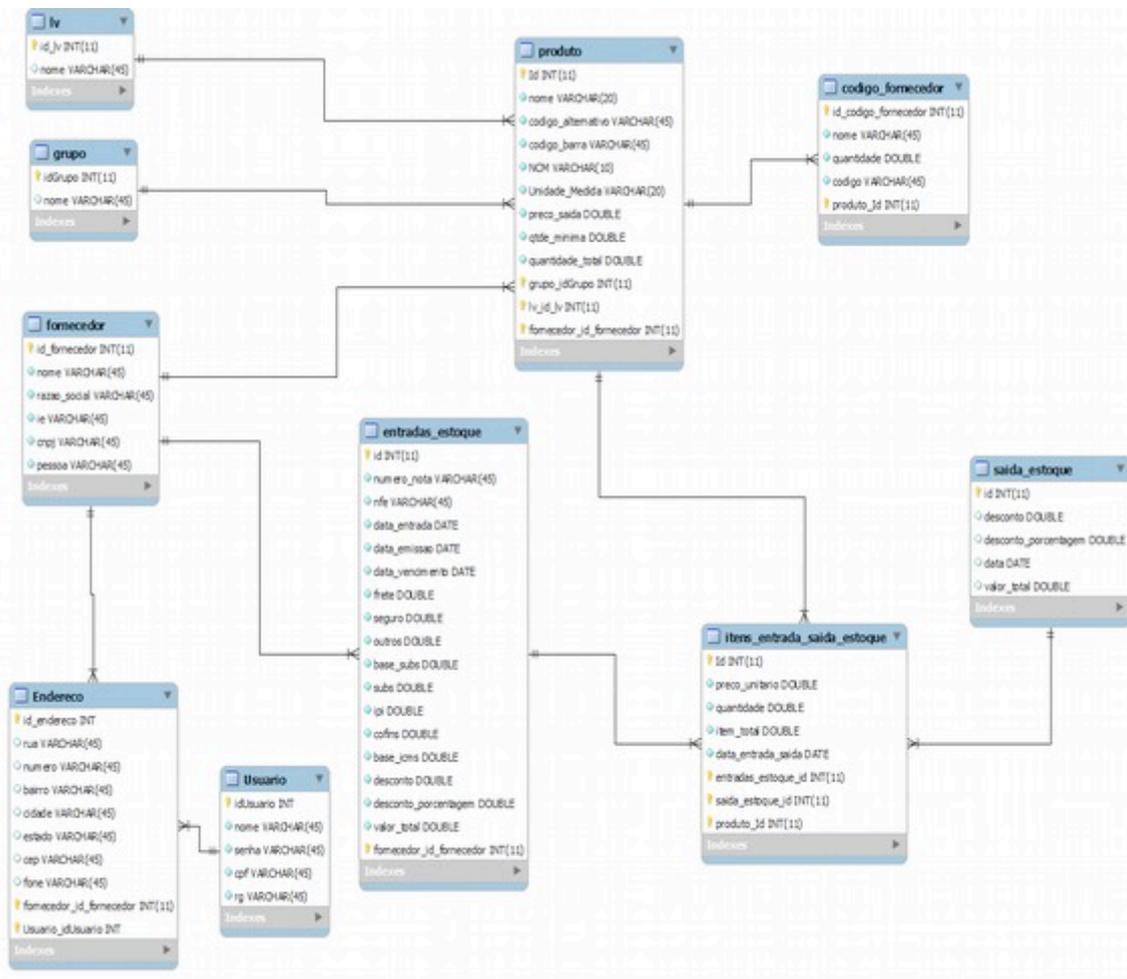
O sistema permitirá a emissão de relatórios de entrada, saída e estoque. No qual há a função SEPARAR, onde será realiza uma distinção dos produtos por grupo, fornecedor ou número da nota; A função ORDENAR, para classificação, podendo ser por nome ou data.

De acordo com os diagramas de UML que representa no sistema estão o apêndice 2 que representa o diagrama de Caso de Uso, no apêndice 3, 4, 5, 6, 7, 8 Diagrama de classe, já o apêndice 9, 10 e 11 mostram o Diagramas de Sequência, o apêndice 12 é a modelagem de banco de dados *Workbench*, e por último no apêndice 13 o relatório de avaliação do sistema pela administradora da Royale.

Apêndice 2: Caso de uso



Apêndice 3: Modelagem do banco no WORKBENCH



Apêndice 4: Relatório de Avaliação do Sistema

Relatório de Avaliação do Sistema

Este documento é o Relatório de avaliação do Sistema de Controle de Estoques para a empresa Royale Lanches Ltda. A avaliação foi feita junto com a administradora da empresa para saber se os requisitos funcionais atenderam às expectativas do usuário.

| Funcionalidade | Implementado | Atende | Não Atende |
|----------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| Login de Usuários | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Cadastro de Produto | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Cadastro de Fornecedor | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Entrada de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Importação de Xml | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Saída de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Estorno de Entrada de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Estorno de Saída de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Relatório de Entrada de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Relatório de Saída de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Relatório de Estoques | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |


YAMAGUTI E YAMAGUTI LTDA
RAZÃO SOCIAL
Assinatura da Administradora da empresa Royale