

UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**SISTEMA WEB PARA CONTROLE DE ESTOQUE DO RESTAURANTE
UNIVERSITÁRIO DO CAMPUS SENADOR HELVÍDIO NUNES DE BARROS**

VALDIRENE DE ARAÚJO SILVA

PICOS - PI

2016

VALDIRENE DE ARAÚJO SILVA

SISTEMA WEB PARA CONTROLE DE ESTOQUE DO RESTAURANTE
UNIVERSITÁRIO DO CAMPUS SENADOR HELVÍDIO NUNES DE BARROS

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Esp. Leonardo Pereira de Sousa.

FICHA CATALOGRÁFICA

Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

S586s Silva, Valdirene de Araújo.

Sistema *web* para controle de estoque do restaurante
universitário do campus Senador Helvídio Nunes de Barros /
Valdirene de Araújo Silva. – 2016.

CD-ROM; il.; 4 ¾ pol. (44 f.)

Monografia (Curso Bacharelado em Sistemas de Informação) –
Universidade Federal do Piauí, Picos, 2016.

Orientador(A): Prof. Esp. Leonardo Pereira de Sousa

1. Sistema *Web*. 2. Controle de Estoque. 3. *Ruby on Rails*. I. Título.

CDD 005.1

SISTEMA WEB PARA CONTROLE DE ESTOQUE DO RESTAURANTE
UNIVERSITÁRIO DO CAMPUS SENADOR HELVIDIO NUNES DE BARROS

VALDIRENE DE ARAUJO SILVA

Monografia APROVADA _____ como exigência parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 24 de FEVEREIRO de 2016

Leonardo Pereira de Sousa

Prof.^o Esp. Leonardo Pereira de Sousa
Orientador

Dennis Sávio Martins da Silva

Prof. Esp. Dennis Sávio Martins da Silva
Membro

Márcio Alves de Macêdo

Prof. Me. Márcio Alves de Macêdo
Membro

Dedico esse trabalho aos meus pais, João Pedro e Maria Luzia, e a minha grande família por me apoiarem sempre.

AGRADECIMENTOS

A Deus, pelo dom da vida e pela sua infinita bondade. Por ser minha força em todos os momentos, me permitindo a realização desse trabalho, mesmo com tantas dificuldades no caminho.

Ao meu orientador, Leonardo Sousa, por sempre estar disposto a dividir o seu conhecimento com seus estudantes, me ajudando da melhor forma possível para realização desse trabalho. Agradeço pela paciência e pelo apoio, que foram fundamentais.

A todos os professores do curso de Sistemas de Informação, por sua dedicação em transferir conhecimentos aos alunos, sempre buscando melhorias para o curso. De modo particular agradeço ao meu co-orientador, Ivenilton Alexandre, que muito contribuiu para a realização desse trabalho.

Aos meus pais, João Pedro e Maria Luzia, por todo amor, carinho e paciência. Por me apoiarem na luta por meus objetivos e me presentear com uma grande e maravilhosa família, sempre unida. Amo muito todos vocês!

As minhas amigas Fernanda Luz, Maria Herondina, Elenice e Kleidiana, que dividiram comigo tantos momentos alegres, estavam ao meu lado nas horas difíceis e que muitas vezes também reclamaram da minha ausência por conta dos estudos, mas que sempre me apoiaram na luta por meus objetivos. Sei que posso contar sempre com vocês, obrigada!

Aos meus amigos e também colegas da UFPI, Felipe Luan, Francilene, Laise, Mara Roberta e Viviane. A jornada até aqui não teria o mesmo valor se não fossem vocês que estiveram sempre presentes, me dando força e ânimo na caminhada.

A todos que contribuíram, de forma direta ou indireta, para a realização desse trabalho. Muito obrigada!

“É bom perseguir um objetivo para uma jornada,
mas no fim o que realmente importa é a jornada
em si.”

(Ursula Guin)

RESUMO

Cada vez mais as organizações utilizam sistemas de informação para automatizar suas tarefas, melhorando a maneira de conduzir os negócios. O gerenciamento de itens em estoque é uma atividade essencial em uma organização, sendo necessário utilizar-se de métodos adequados para realizar um controle de estoque eficiente, de modo a equilibrar o fluxo entre o fornecimento (entrada) e a saída. Este projeto tem como objetivo o desenvolvimento de um sistema web para o Restaurante da Universidade Federal do Piauí, Campus Senador Helvídio Nunes de Barros, que realize um controle de estoque adequado e auxilie na elaboração do cardápio, de forma a facilitar o trabalho dos funcionários para que possam atender a demanda de refeições diárias com mais agilidade e eficácia, buscando manter o balanceamento entre estoque e consumo. Para o desenvolvimento do sistema foram utilizados: a linguagem de programação *Ruby* juntamente com o *framework* de desenvolvimento *Ruby on Rails* e o *framework* de *front-end Bootstrap*, executando sobre a plataforma *web*.

Palavras-chave: Sistema *Web*, Controle de Estoque, *Ruby on Rails*.

ABSTRACT

Increasingly organizations use information systems to automate their tasks, improving the way business is conducted. The management of items in stock is an essential activity in an organization, it is necessary to use suitable methods to implementing an efficient inventory control, in order to balance the flow between the supply (input) and output. This project aims to develop a web system for the Restaurant of the Federal University of Piau , Campus Senador Helvidio Nunes de Barros, to conduct a proper inventory control and assist in the preparation of the menu, in order to facilitate the employees work for that can meet the demands of daily meals with more speed and efficiency, seeking to maintain the balance between inventory and consumption. To develop the system were used: the Ruby programming language with the development framework Ruby on Rails and the front-end Framework Bootstrap, running on the web platform.

Keywords: Web System, Inventory Control, Ruby on Rails.

LISTA DE ILUSTRAÇÕES

Figura 01 - Diagrama de Casos de uso	34
Figura 02 - Diagrama de Classe	36
Figura 03 - Tela de <i>Login</i>	37
Figura 04 - Menu para usuário nutricionista	38
Figura 05 - Cadastro de Categorias	38
Figura 06 - Listar Categorias	38
Figura 07 - Cadastro de Fornecedores	38
Figura 08 - Listar Fornecedores	39
Figura 09 - Cadastro de Produtos	39
Figura 10 - Listar Produtos	40
Figura 11 - Nova Entrada no Estoque	40
Figura 12 - Listar Estoques	41
Figura 13 - Baixa no Estoque	41
Figura 14 - Produtos com estoque baixo	42
Figura 15 - Verificar Disponibilidade de Produtos	42

LISTA DE QUADROS

Quadro 01 - Requisitos Funcionais	31
Quadro 02 - Requisitos Não Funcionais	32
Quadro 03 - Regras de Negócio	32

LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i>
DRY	<i>Don't Repeat Yourself</i>
HTML	<i>HiperText Markup Language</i>
HTTP	<i>HiperText Transfer Protocol</i>
MVC	<i>Model-View-Controller</i>
RU	Restaurante Universitário
SAD	Sistema de Apoio a Decisão
SAE	Sistema de Apoio ao Executivo
SCE	Sistema de Controle de Estoque
SE	Sistemas de Escritório
SGBD	Sistema de Gerenciamento de Banco de Dados
SIG	Sistema de Informação Gerencial
SPT	Sistema de Processamento de Transações
SQL	<i>Structured Query Language</i>
STC	Sistema de Trabalho com Conhecimento
TI	Tecnologia da Informação
UFPI	Universidade Federal do Piauí
UML	<i>Unified Modeling Language</i>
XP	<i>Extreme Programming</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivo	15
1.2	Estrutura da monografia	15
2	REFERENCIAL TEÓRICO	17
2.1	Sistemas de Informação	17
2.1.1	Sistemas <i>Web</i>	19
2.1.2	Controle de Estoque.....	20
2.2	Tecnologias	20
2.2.1	Linguagem Ruby	20
2.2.2	<i>Framework Ruby on Rails</i>	21
2.2.3	<i>Framework Front-End Twitter Bootstrap</i>	23
2.2.4	Banco de Dados.....	23
2.3	Engenharia de <i>Software</i>	24
2.3.1	Engenharia de Requisitos.....	26
2.3.2	UML	27
2.3.3	<i>Scrum</i>	28
3	SISTEMA <i>WEB</i> PARA CONTROLE DE ESTOQUE DO RESTAURANTE UNIVERSITÁRIO DA UFPI	29
3.1	Requisitos do Sistema	30
3.2	Diagramas de Casos de Uso	33
3.3	Diagrama de Classes	35
4	FUNCIONAMENTO DO SISTEMA	37
4.1	Funcionalidades	37
5	CONSIDERAÇÕES FINAIS	43
6	REFERÊNCIAS	44

1 INTRODUÇÃO

Com o avanço da Tecnologia da Informação (TI) as empresas passaram a utilizar sistemas informatizados para dar suporte a suas atividades. Um sistema de informação se configura como uma solução organizacional e administrativa baseada na TI para enfrentar um desafio proposto pelo ambiente (LAUDON; LAUDON, 2004).

O gerenciamento de itens em estoque é uma tarefa fundamental a uma organização. Segundo Pozo (2010), “é notório que todas as organizações de serviço devem preocupar-se com o controle de estoques, visto que desempenham e afetam de maneira bem definida o resultado da empresa”.

Percebe-se a importância de se administrar corretamente os produtos estocados quando não há a quantidade suficiente disponível no momento exato para atender as necessidades da organização.

O equilíbrio dos níveis de estoque é um desafio enfrentado atualmente no Restaurante Universitário (RU) da UFPI (Campus Senador Helvídio Nunes de Barros – Picos), que tem a finalidade de fornecer refeições balanceadas e de baixo custo à Comunidade Universitária (estudantes, professores, servidores técnico-administrativos e terceirizados), de forma a conseguir atender a clientela com o máximo de efetividade e o mínimo de desperdício no seu sistema de produção, garantindo que os usuários tenham uma alimentação de qualidade e no tempo determinado.

Atualmente o controle de estoque no RU do Campus de Picos é feito através de planilhas do Excel¹, baseando-se no pregão de gêneros e notas de entrega dos fornecedores (entrada), utilizando uma planilha de cardápio para registrar a saída (tudo o que é utilizado para preparar as refeições). Dois funcionários são responsáveis pela parte de controle de estoque, além da nutricionista, que também elabora o cardápio.

O problema nessa forma de organização é que não se pode saber com exatidão o balanço entre o que foi entregue e o que já foi utilizado, sendo preciso sempre fazer a conferência no estoque físico uma vez por dia. Além disso, também não há uma sincronia nas alterações das planilhas da nutricionista responsável pelo setor e os demais funcionários, o que gera erros na hora de elaborar o cardápio por causa da inconsistência nos dados.

O cardápio é feito baseando-se no padrão da UFPI e considerando todas as características da clientela (semestral, bimestral, mensal, semanal e diário). A refeição,

¹ Microsoft Excel – <https://products.office.com/pt-br/excel>. O Excel é um software para criação de planilhas eletrônicas, produzido pela Microsoft.

normalmente, é composta por arroz, feijão, salada crua ou cozida, carnes bovina, suína, peixe ou ave, uma fruta de sobremesa e suco nas terças-feiras; evitando repetições entre os dias da semana e proporcionando uma dieta rica em proteínas, fibras, carboidratos, minerais e vitaminas.

Ao preparar o cardápio, a nutricionista baseia-se no valor da quantidade de alimentos presente nas planilhas, porém muitas vezes é necessário refazer tudo novamente, porque quando se verifica o estoque não tem algum dos componentes escolhidos, devido à falta de uniformidade entre as planilhas e às vezes por não haver uma atualização na quantidade após a retirada do ingrediente. Isto torna a atividade muito custosa por conta do tempo que é gasto para realizá-la.

Quando a informação de estoque no sistema de controle, seja ele informatizado ou manual, não confere com o saldo real, diz-se que este inventário não é confiável ou não tem acuracidade. O controle de estoque informatizado proporciona maior acurácia e contribui para a previsão de compras de produtos, através de relatórios que permitem analisar o fluxo de saída, gerando uma projeção de compra para que não haja falta de produtos em estoque.

1.1 Objetivo

O objetivo desse trabalho é desenvolver um sistema *web*² para o Restaurante Universitário da UFPI, Campus Senador Helvídio Nunes de Barros, que realize o controle de estoque de forma a conseguir suprir as necessidades organizacionais com relação a estocagem dos produtos, buscando uma maior eficiência nesse processo e facilitando a elaboração do cardápio.

1.2 Estrutura da monografia

O trabalho está organizado da seguinte maneira:

- Capítulo 2 – Referencial Teórico: Fornece o embasamento teórico para o trabalho. São mostrados conceitos relacionados a sistemas de informação, engenharia de *software* e tecnologias utilizadas para o desenvolvimento do mesmo.

² Sistema *web* é um sistema de informação desenvolvido para funcionar utilizando tecnologia *web*, podendo estar acessível através da *Internet*.

- Capítulo 3 – Especificação do Sistema: Será mostrada a análise de requisitos, os diagramas produzidos para o entendimento do problema e o desenvolvimento do sistema.
- Capítulo 4 – Funcionamento do Sistema: São mostrados os usuários do sistema e as principais funcionalidades que o sistema apresenta.
- Capítulo 5 – Considerações Finais: Apresenta-se a conclusão do trabalho e indicações de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Esse capítulo descreve a fundamentação teórica necessária para o desenvolvimento deste trabalho, abordando conceitos relacionados a Sistemas de Informação, Engenharia de *Software* e as tecnologias utilizadas.

2.1 Sistemas de Informação

Um sistema é um conjunto de elementos ou componentes que interagem para atingir objetivos (STAIR; REYNOLDS, 2006). Um sistema possui três funções básicas: entrada, processamento e saída. Durante o processo de entrada o sistema recebe dados para serem processados, produzindo uma informação (processamento) e apresentando um resultado (saída).

Os dados podem ser considerados os fatos brutos, que não possuem significado, isoladamente. Informação é o conjunto de fatos organizados de modo a terem valor adicional, além do valor dos fatos propriamente ditos (STAIR; REYNOLDS, 2006). O valor da informação está diretamente ligado a como ela auxilia os tomadores de decisões a atingir seus objetivos organizacionais.

Em consequência do crescimento da importância da informação, surgiu a necessidade de gerenciá-las de uma forma adequada e eficiente e, dessa necessidade, surgiram os sistemas de informação (BEZERRA, 2007).

Segundo O'Brien (2013), “Um sistema de informação usa recursos de pessoas, *hardware*, *software*, dados e redes para realizar as atividades de entrada, processamento, saída, armazenamento e controle que convertem os recursos de dados em produtos de informação”. Os sistemas de informação transformam a informação em algo significativo e útil, de modo a facilitar o planejamento, o controle e o processo decisório nas organizações. A informação auxilia a coordenar esforços e produzir efeitos, orientando a tomada de decisão.

Presentes em todos os níveis da organização, desde o operacional até o estratégico, os sistemas de informação trazem uma série de benefícios, tais como: redução do tempo para realizar uma tarefa, a facilidade e rapidez no acesso à informação, aumento da segurança e a diminuição de erros.

De acordo com Laudon e Laudon (2004), podemos classificar os sistemas de informação de acordo com o nível organizacional que o sistema atende e também apresentam

os tipos de sistemas de informação associados a cada nível. São quatro tipos principais de níveis de sistemas de informação:

- Sistemas do nível operacional: acompanham atividades e transações elementares da organização;
- Sistemas do nível de conhecimento: suportam os trabalhadores do conhecimento e de dados da organização;
- Sistemas do nível gerencial: atendem às atividades de monitoração, controle, tomada de decisões e procedimentos administrativos dos gerentes médios;
- Sistemas do nível estratégico: são sistemas destinados as decisões de alto nível.

Há seis tipos de sistemas de informação, sendo que para cada nível podem estar associados um ou mais tipos de sistemas:

- Sistemas de processamento de transações (SPTs): atendem ao nível operacional da organização. Realizam e registram transações rotineiras necessárias ao funcionamento da empresa;
- Sistemas de trabalho com conhecimento (STCs): encontrados no nível de conhecimento da organização, utilizam projetos, especificações e bases de conhecimento em geral para produzir modelos e gráficos;
- Sistemas de escritório (SE): encontrados no nível de conhecimento da organização. Aumentam a produtividade na manipulação de dados em um escritório;
- Sistemas de informação gerencial (SIG): atendem ao nível gerencial da organização. Fornecem relatórios aos gerentes. Apoiam as funções de planejamento, controle e decisão no nível gerencial;
- Sistemas de apoio a decisão (SAD): atendem ao nível de gerencia da organização. Ajudam os gerentes a tomar decisões não-usuais, que se alteram com rapidez e que não são facilmente especificadas com antecedência;
- Sistemas de apoio ao executivo (SAEs): atendem ao nível estratégico, fornecendo amplo ambiente de computação e telecomunicações para dar assistência ao processo de decisão da alta administração.

Os sistemas de informação, independentemente de seu nível ou classificação, objetivam auxiliar os processos de tomada de decisões na organização. Segundo Stair e Reynolds (2006), as organizações usam os sistemas de informação para apoio aos objetivos organizacionais.

Os sistemas de informação vêm substituindo progressivamente procedimentos manuais por procedimentos automatizados de trabalho, reduzindo o custo de operações em muitas empresas, porque dispensam as rotinas manuais e em papel que envolvem. Segundo Laudon e Laudon (2004), os sistemas de informações são parte integrante de uma organização e é um produto de três componentes: tecnologia, organização e pessoas.

2.1.1 Sistemas *Web*

Segundo Laudon (1999) a *Internet* está inspirando novas utilizações para os sistemas de informação e novos modelos de negócios, por eliminar muitas barreiras técnicas, geográficas e de custo que obstruem o fluxo global de informações. O autor ainda acrescenta que a *Internet* está mudando a forma como as pessoas acessam informações, conduzem negócios, se comunicam, colaboram e mesmo passam seu tempo livre.

A *Internet* é uma rede de computadores que conecta uma grande variedade de dispositivos de todas as partes do mundo. A *web* é um conjunto de páginas, que são acessadas através do protocolo HTTP (*HyperText Transfer Protocol*, Protocolo de Transferência de Hipertexto). Essas páginas contêm hipermídia, ou seja, textos, gráficos, imagens, vídeo e todos os tipos de outros tipos de multimídia, junto com links para outras páginas. As páginas *web* evoluíram de um formato estático, constituído por páginas HTML³ simples para um formato dinâmico, em que são disponibilizadas as informações e também são construídas aplicações sobre as quais são realizados negócios no mundo todo.

Uma aplicação *web* é acessada através da *Internet*, usando um navegador como cliente e consiste de uma coleção de *scripts* cliente e servidor, páginas HTML e outros recursos que podem estar dispostos em múltiplos servidores. Na arquitetura Cliente-Servidor, o cliente conecta com o servidor, requisita serviços e/ou recursos. O servidor “escuta” as requisições e disponibiliza os recursos e/ou serviços.

Software com base na *web* tem as seguintes vantagens (GAMBLE et al., 2013): maior facilidade de distribuição, implantação e manutenção, independente de plataforma e podem ser acessados de qualquer lugar.

³ HTML – Linguagem de Marcação de Hipertexto. É uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores.

2.1.2 Controle de Estoque

Estoques são o conjunto de bens armazenados, que estão disponíveis em um determinado momento, que podem ser utilizados. O gerenciamento de itens em estoque é uma atividade essencial em uma organização, sendo necessário utilizar-se de formas adequadas para realizar um controle eficiente, de forma a equilibrar o fluxo entre o fornecimento (entrada) e a saída.

Segundo Pozo (2010), todas as organizações de serviço devem preocupar-se com o controle de estoques, visto que desempenham e afetam de maneira bem definida o resultado da empresa. O autor ainda acrescenta que a importância da correta administração de materiais pode ser mais facilmente percebida quando os bens necessários não estão disponíveis no momento exato.

Depreende-se daí a importância de manter estoques: ter o produto certo, na hora certa e em quantidade suficiente para atender as necessidades da organização. O controle é necessário para que os níveis de estoques existentes estejam de acordo com as necessidades da demanda. É muito importante, pois é através dele que será possível prever o quanto que será necessário comprar no próximo pedido ao fornecedor.

É praticamente impossível conhecer a demanda futura; tornando-se necessário manter determinado nível de estoque, para assegurar disponibilidade de produtos às demandas, bem como minimizar os custos de produção, movimentação e estoques (POZO, 2010).

Os estoques ajudam a maximizar o atendimento aos clientes, protegendo a empresa da incerteza (ARNOLD, 2012). O acompanhamento dos níveis de estoques através de sistemas informatizados é essencial. Segundo Ballou (1993) esse controle informatizado proporciona maior acurácia, pois gera relatórios mais eficientes.

2.2 Tecnologias

As principais tecnologias utilizadas durante esse projeto são abordadas a seguir.

2.2.1 Linguagem Ruby

*Ruby*⁴ é uma linguagem de programação orientada a objetos, interpretada, com tipagem forte e dinâmica, que tem como foco a simplicidade e produtividade. Desenvolvida por Yukihiro Matsumoto e lançada em 1995. Foi inspirada em outras linguagens como *Perl*, *Smalltalk* e *Lisp*, e hoje está entre as linguagens mais usadas, muito em função da disseminação do seu principal *framework*⁵, o *Ruby on Rails*⁶ (SOUZA, 2012).

As características da linguagem *Ruby* incluem sintaxe simples, multiparadigma (orientado a objetos, funcional, reflexivo e imperativo) e multiplataforma (os códigos *Ruby* podem ser executados em vários sistemas operacionais).

O *Ruby* possui um gerenciador de pacotes e dependências muito eficiente chamado *RubyGems*. As *gems* podem ser vistas como bibliotecas reutilizáveis de código, que auxiliam e aceleram o desenvolvimento. Existem diversos tipos de *gems*, para todos os tipos de funções tais como: autenticação de usuários, *upload* de arquivos, entre outros, que podem ser encontradas e instaladas facilmente através do *RubyGems*⁷.

2.2.2 Framework Ruby on Rails

O *Ruby on Rails* é um *framework* de código aberto escrito na linguagem de programação *Ruby* para desenvolvimento de aplicações *web*. Foi criado por David Heinemeier Hansson e lançado em 2004. Tem como foco o desenvolvimento ágil, de forma a aumentar a produtividade com uma quantidade menor de linhas de código.

O *Rails* foi projetado para ser uma solução de desenvolvimento completa, fornecendo todas as ferramentas necessárias para a criação de aplicações *web*. Baseia-se em dois conceitos fundamentais:

- **Convenção sobre configuração:** escrever menos código para implementar uma determinada funcionalidade na aplicação, desde que se respeite alguns padrões de nome e localização de arquivos, nome de classes e métodos, entre outras regras simples.
- **DRY (*Don't Repeat Yourself*, “Não Se Repita”):** incentiva a fazer uso da reutilização de código, que é uma das principais vantagens da orientação a objetos, e adotar padrões de projeto mais adequados para essa finalidade.

⁴ Linguagem de Programação *Ruby* – <https://www.ruby-lang.org>

⁵ *Framework* é um conjunto de ferramentas usadas para auxiliar no desenvolvimento, eliminando algumas dificuldades associadas com atividades comuns.

⁶ *Framework Ruby on Rails* – <http://rubyonrails.org/>

⁷ *RubyGems*: Gerenciador de pacotes do *Ruby* – <https://rubygems.org/>

O *Rails* incentiva as boas práticas de codificação e organização das aplicações. Seguindo esse princípio, utiliza o padrão de projeto MVC (Model, View, Controller), que divide a aplicação em 3 camadas:

- **Modelo (*Model*)** é onde ficam as regras de negócio e tudo que estiver relacionado a acesso ao banco de dados e arquivos, validações em geral, cálculos e requisitos específicos do projeto.
- **Visão (*View*)** tem como função mostrar as informações. É a parte da aplicação que é apresentada ao usuário.
- **Controlador (*Controller*)** é o responsável por retirar as informações do banco de dados e tornar essas informações disponíveis para as exibições. Um controlador pode ser considerado como um intermediário entre *models e views*. Faz com que os dados do modelo estejam disponíveis para a visão para que ele possa exibir os dados para o usuário, e ele salva ou atualiza os dados do usuário para o modelo.

O modelo MVC foi desenvolvido por Trygve Reenskaug, em 1979. As principais vantagens desse modelo incluem reaproveitamento de código e regras, facilidade de manutenção, permite integração de equipes e/ou divisão de tarefas, facilidade na atualização da interface da aplicação. Os projetos desenvolvidos em *Rails* estão bem organizados nas três camadas, de forma a tornar o processo de desenvolvimento mais rápido e facilitar as futuras manutenções necessárias.

Todas as aplicações *Rails* utilizam a mesma estrutura de pastas e os mesmos padrões em nomes, resultando em aplicações fáceis de entender (GRIFFITHS, 2010). Cada diretório tem uma função específica e bem clara na aplicação:

- `app` - A maioria dos arquivos específicos da aplicação ficam aqui (inclusive todo o MVC, dividido em diretórios);
- `bin` - Executáveis do *Rails* e das *gems* instaladas;
- `config` - Configurações da aplicação;
- `db` - Migrações, esquema e outros arquivos relacionados ao banco de dados;
- `doc` - Documentação do sistema;
- `lib` - Bibliotecas auxiliares;
- `log` - Informações de *log*;
- `public` - Arquivos estáticos que serão servidos pela *web*;
- `test` - Testes da aplicação;
- `tmp` - Arquivos temporários como cache e informações de sessões;

- venvor - Dependências e bibliotecas de terceiros.

Ao escrever uma aplicação *Rails*, espera-se que você siga as convenções que foram apresentadas, devendo, portanto, definir apenas configurações que não sejam convencionais. Em vez de configuração, o *Rails* depende de estruturas comuns e convenções de nomenclatura.

2.2.3 Framework Front-End Twitter Bootstrap

Desenvolvido pela equipe do Twitter⁸, o *Bootstrap* é um *framework front end*⁹ de código aberto que torna o desenvolvimento *web* mais rápido e fácil e possibilita a criação de *layouts* responsivos¹⁰. Algumas vantagens:

- Possui documentação detalhada e de fácil entendimento;
- É otimizado para o desenvolvimento de *layouts* responsivos;
- Possui componentes suficientes para o desenvolvimento de qualquer *site* ou sistema *web* com interface simples;
- Facilita a criação e edição de *layouts* por manter padrões;
- Funciona nos principais navegadores atuais (Chrome, Safari, Firefox, IE, Opera).

Pelas facilidades que apresenta em reutilização de componentes prontos o *Bootstrap* vem sendo utilizado cada vez mais pelos desenvolvedores, pois permite uma maior agilidade na hora de montar as interfaces dos seus sistemas *web*.

2.2.4 Banco de Dados

Conforme Laudon e Laudon (1999), as vantagens de se utilizar uma abordagem de banco de dados para organizar informações incluem independência entre dados e programas aplicativos, redução da redundância e da inconsistência de dados, eliminação da confusão dos dados, consolidação do gerenciamento de dados e facilidade de acesso e uso das informações.

Um modelo de banco de dados relacional possibilita melhor facilidade de acesso e organização dos dados, auxiliando assim os cadastros, consultas e acesso ao banco. Esse modelo representa um banco de dados como uma coleção de tabelas ou relações, usa o conceito de uma relação matemática — algo como uma tabela de valores — como seu bloco

⁸ Twitter – <https://twitter.com/>

⁹ Framework Front-End é um *framework* relacionado ao desenvolvimento de *layouts* de sistemas *web*.

¹⁰ *Layout* Responsivo é quando um *site* se adapta automaticamente a largura de tela do dispositivo no qual está sendo visualizado.

de construção básica, e tem sua base teórica na teoria dos conjuntos e na lógica de predicados de primeira ordem. A maioria dos bancos de dados relacionais usa uma linguagem de consulta de alto nível chamada SQL (*Structured Query Language* – Linguagem de Consulta Estruturada).

A criação de um banco de dados é o processo de armazenar os dados em alguma mídia apropriada controlada pelo Sistema Gerenciador de Banco de Dados (SGBD). Na concepção dos autores Elmasri e Navathe (2005):

Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de *software* de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações. A definição de um banco de dados implica especificar os tipos de dados, as estruturas e as restrições para os dados a serem armazenados em um banco de dados. (ELMASRI, NAVATHE, 2005, p.10)

O *MySQL*¹¹ é um SGBD que utiliza a linguagem SQL como *interface*. Suas características incluem robustez, segurança, portabilidade, confiabilidade, armazenamento e gerenciamento de dados mais rápido, flexível e seguro.

Segundo Milani (2007), o *MySQL* é indicado para uso em aplicações em todas as áreas de negócio, independentemente do tamanho de sua aplicação, por ser extremamente rápido e altamente confiável. O autor afirma que atualmente, o *MySQL* tem sido o banco de dados *open-source* mais utilizado em aplicações para a *Internet* pelo fato de essas aplicações demandarem rápido acesso para a geração de páginas HTML, qualquer que seja a linguagem de programação que faça a conexão com o banco.

2.3 Engenharia de *Software*

A Engenharia de *Software* surgiu com o objetivo de aperfeiçoar a qualidade dos produtos de *software* e aumentar a produtividade no processo de desenvolvimento. Engloba processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de *software*. É uma disciplina da engenharia que se ocupa de todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE, 2005).

¹¹ Sistema Gerenciador de Banco de Dados MySQL – <https://www.mysql.com/>

Um processo de *software* visa melhorar a produção de sistemas de alta qualidade, de acordo com as necessidades dos usuários, dentro do prazo e do orçamento previsto. Segundo Paula Filho (2005), um processo de *software* pode ser visto como um conjunto de atividades, métodos, práticas e transformações, usado para desenvolver e manter produtos de *software*. O *software*, assim como todo produto industrial, tem um ciclo de vida (PAULA FILHO, 2005):

- Ele é concebido a partir da percepção de uma necessidade;
- Desenvolvido, transformando-se em um conjunto de itens entregue a um cliente;
- Entra em operação, sendo usado dentro de um algum processo de negócio, e sujeito a atividades de manutenção, quando necessário;
- É retirado de operação, ao final de sua vida útil.

O uso de modelos de processo procura organizar e estruturar o desenvolvimento de *software*. De acordo com Pressman (2011), cada um desses modelos sugere um fluxo de processo ligeiramente diferente, mas todos realizam o mesmo conjunto de atividades metodológicas genéricas: comunicação, planejamento, modelagem, construção e implantação.

Um modelo prescritivo de processo de *software* é um conjunto de elementos que inclui ações de engenharia de *software*, produtos de trabalho e mecanismos que garantam a qualidade e controle de modificações em cada projeto necessárias para o desenvolvimento de um sistema de *software* (PRESSMAN, 2011). Alguns exemplos de modelos de processos prescritivos:

- Cascata: É conhecido também como ciclo de vida clássico. Divide o projeto com base nas atividades, sendo que estas são realizadas de forma sequencial. Uma fase só pode ser iniciada após o término e aprovação da fase anterior a ela.
- Espiral: Divide um projeto em conjuntos de funcionalidades, sendo que a cada iteração faz o ciclo de vida completo, gerando *software* com base no conjunto de funcionalidades.

Os modelos de processos prescritivos citados anteriormente gastam muito tempo no planejamento, projeto, documentação, e não apresentam meios de se adaptar rapidamente às mudanças nos requisitos. Como uma forma de resolver esses problemas e também a alta frequência com que os projetos de software deixavam de cumprir seus cronogramas e extrapolavam seus orçamentos, surgiram as metodologias ágeis.

As metodologias ágeis são adaptativas, trabalham com constante *feedback*¹², o que permite adaptar-se rapidamente a eventuais mudanças nos requisitos e novos fatores durante o desenvolvimento do projeto, ao invés de tentar analisar de forma prévia tudo o que pode ou não acontecer durante o desenvolvimento.

Segundo Pressman (2011), a abordagem mais amplamente utilizada para desenvolvimento de *software* ágil é a *Extreme Programming – XP* (Programação Extrema). A metodologia XP apresenta quatro atividades: planejamento, projeto, codificação e testes. Em todas elas há um acompanhamento contínuo e a realização de pequenos ajustes para adequar o desenvolvimento.

2.3.1 Engenharia de Requisitos

O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto forma a Engenharia de Requisitos (PAULA FILHO, 2005). O levantamento dos requisitos consiste em entender os requisitos do sistema, obter e definir as exigências e solicitações do usuário.

A engenharia de requisitos é indispensável para desenvolver sistemas de qualidade, que satisfaçam o cliente, cumpram os prazos e se mantenham dentro do orçamento. Para isso, deve-se documentar os requisitos do cliente da forma mais completa possível, de modo que se possa identificar o quanto antes se há problemas nos requisitos.

Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos (PFLEEGER, 2004). A compreensão completa dos requisitos de um sistema de informação é fundamental para um desenvolvimento bem sucedido. O levantamento de requisitos é útil para:

- Identificar as necessidades dos usuários;
- Verificar a viabilidade de implementar estas necessidades;
- Criar um modelo do sistema que será utilizado nas fases de desenvolvimento seguintes.

Os requisitos de sistema de *software* são, frequentemente, classificados em requisitos funcionais, não funcionais ou requisitos de domínio (SOMMERVILLE, 2005):

¹² *Feedback* é um retorno, uma análise que permite avaliar os resultados do sistema através do processo de retroalimentação.

- Requisitos funcionais: são as declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações.
- Requisitos não funcionais: são restrições sobre os serviços ou as funções oferecidas pelo sistema.
- Requisitos de domínio: são provenientes do domínio da aplicação do sistema e que refletem as características e as restrições desse domínio. Podem ser requisitos funcionais ou não funcionais.

2.3.2 UML

A UML¹³ (Unified Modeling Language, Linguagem de Modelagem Unificada) é uma linguagem gráfica para especificar, visualizar, construir e documentar os artefatos de sistemas complexos de software (BOOCH; RUMBAUGH; JACOBSON, 2005). A UML surgiu da necessidade de padronização da modelagem orientada a objetos, de modo que refletisse as melhores práticas, estimulando os desenvolvedores a modelar os sistemas antes de construí-los. Utiliza notações gráficas, que permitem uma melhor comunicação e entendimento, ajudando a obter uma visão geral do sistema.

Com a ajuda da modelagem, pode-se compreender melhor o sistema a ser desenvolvido. Os modelos proporcionam um guia para a construção do sistema, reduzindo a probabilidade de ocorrência de erros. Quanto maior e mais complexo for o sistema, maior será a importância da modelagem, uma vez que não é possível compreendê-lo em sua totalidade.

Segundo Booch, Rumbaugh e Jacobson (2005), a modelagem de um sistema envolve a identificação de itens considerados importantes de acordo com uma determinada visão. Os diagramas permitem a visualização de um sistema sob diferentes perspectivas. A UML possui vários diagramas e cada um desses apresenta o sistema de uma visão diferente. Alguns dos diagramas oferecidos pela UML são (BOOCH; RUMBAUGH; JACOBSON, 2005):

- Diagrama de Classe: exibe um conjunto de classes, *interfaces* e colaborações, bem como seus relacionamentos. Esses diagramas são encontrados com maior frequência em sistemas de modelagem orientados a objeto e abrangem uma visão estática da estrutura do sistema.

¹³ Linguagem de Modelagem Unificada – www.uml.org/

- Diagrama de casos de uso: exibe um conjunto de casos de uso, atores e seus relacionamentos. Abrangem a visão estática de casos de uso do sistema e são importantes principalmente para a organização e a modelagem de comportamentos do sistema.
- Diagrama de atividade: exibe a estrutura de um processo. Abrange a visão dinâmica do sistema e é importante principalmente para a modelagem da função de um sistema e dá ênfase ao fluxo de controle entre objetos.

A UML é independente de linguagem de programação e de processo de desenvolvimento, e sua correta utilização auxilia na boa execução do projeto, possibilitando uma melhor comunicação entre as pessoas interessadas no projeto. A existência de vários diagramas permite uma visualização do sistema de acordo com a necessidade de cada um.

2.3.3 *Scrum*

Segundo Pressman (2011) *Scrum* é um método de desenvolvimento ágil de *software*, seus princípios estão de acordo com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades: requisitos, análise, projeto, evolução e entrega.

Com o *Scrum* ocorrem entregas frequentes, cada parte implementada do projeto é levada ao cliente uma versão funcional para ser testada e validada. Com isso o cliente pode dar um *feedback* do sistema e indicar mudanças ou detalhes adicionais. Assim, o cliente acompanha todo o desenvolvimento diminuindo a quantidade de erros. As vantagens de utilizar o *Scrum*, incluem (SABBAGH,2013):

- Entregas frequentes de retorno ao investimento dos clientes;
- Redução dos riscos do projeto;
- Maior qualidade no produto gerado;
- Mudanças utilizadas como vantagem competitiva;
- Visibilidade do progresso do projeto;
- Redução do desperdício;
- Aumento de produtividade.

3 SISTEMA *WEB* PARA CONTROLE DE ESTOQUE DO RESTAURANTE UNIVERSITÁRIO DA UFPI

Esse capítulo descreve o Sistema de Controle de Estoque (SCE) desenvolvido para Restaurante Universitário da Universidade Federal do Piauí, Campus Senador Helvídio Nunes de Barros. Os detalhes do sistema são retratados englobando desde o levantamento de requisitos, diagramas de casos de uso e até mesmo o diagrama de classes.

O SCE é um sistema de auxílio no gerenciamento dos produtos utilizados no RU, desenvolvido para a plataforma *web*, de modo a facilitar o acesso às informações e eliminar a inconsistência dos dados gerada pelas planilhas da nutricionista e dos funcionários estoquistas, podendo ser utilizado de qualquer máquina, desde que tenha acesso a rede e seja autorizado previamente.

No sistema serão cadastrados os alimentos fornecidos realizando um controle de estoque eficiente, evitando a ocorrência de erros e facilitando as tarefas de seus usuários, para que a demanda de refeições possa ser atendida prontamente. Foi escolhido o estilo de desenvolvimento através da *web*, visto que cada funcionário deverá ter acesso ao sistema para manter sua base de informações sobre o estoque e dar acesso à nutricionista responsável por realizar o cardápio para que a mesma utilize as informações cadastradas. Sendo assim, as utilizações de padrões, linguagens e metodologias de desenvolvimento *web* foram as que mais se enquadraram ao contexto.

A metodologia de desenvolvimento utilizada foi o *Scrum*, devido às características de acompanhamento do desenvolvimento diariamente, revisando e analisando o que foi projetado no dia anterior, determinando metas para os momentos seguintes.

Para o desenvolvimento do protótipo funcional, foi utilizada a linguagem de programação *Ruby* e o *framework Ruby on Rails*. O *Ruby* é uma linguagem orientada a objetos, de sintaxe simples e extremamente dinâmica, focada na produtividade. O *Ruby on Rails* é um *framework* escrito em *Ruby* que facilita a construção de aplicações *web*, eliminando algumas dificuldades associadas com atividades comuns no desenvolvimento *web*, por exemplo, disponibilizando recursos para facilitar o acesso ao banco de dados e promover a reutilização de código na aplicação. A grande vantagem de utilizar um *framework*, como o *Rails*, é que toda a infra-estrutura da aplicação é criada automaticamente, devendo o programador preocupar-se apenas com as regras de negócio.

Foi utilizado o *framework front-end Bootstrap* para o desenvolvimento da *interface*, melhorando a apresentação do sistema, para que o usuário consiga utilizar as funcionalidades implementadas.

3.1 Requisitos do Sistema

Nesta parte do trabalho, são apresentados os resultados do levantamento e análise de requisitos, que são de fundamental importância para a construção do sistema.

Serão tratadas as especificações do SCE, que foi proposto e desenvolvido neste trabalho, dando ênfase nas principais tarefas que o sistema deverá efetuar para alcançar os objetivos dos usuários e da organização como um todo.

Como base para o desenvolvimento do sistema, foram realizadas reuniões com os funcionários do RU que são responsáveis pela manutenção do estoque, verificando as deficiências nesse processo e identificando as necessidades que precisavam ser atendidas. Esses funcionários citados são conhecidos no processo de análise de requisitos como *Stakeholders*, que são pessoas ou organizações que influenciam os requisitos do sistema.

O gerenciamento do estoque, que até então era feito exclusivamente através de planilhas eletrônicas, gerava problemas como baixo grau de acuracidade, divergência entre as planilhas utilizadas pela nutricionista e os estoquistas, dificuldade para verificar itens que estão em baixa quantidade, demora para consultar os ingredientes do cardápio.

Foi feita uma análise organizacional a fim de verificar a viabilidade de construção do sistema. Em seguida, planejou-se o levantamento de requisitos, como uma forma de identificar as necessidades e como as mesmas poderiam ser atendidas.

Com as informações obtidas, foram formulados os requisitos funcionais, as regras de negócio e os requisitos não funcionais do SCE. O Quadro 01 contém detalhes sobre os requisitos funcionais identificados por um código único, com suas respectivas descrições e suas dependências em casos específicos.

No sistema podem ser cadastrados os produtos utilizados, que entram para formação do estoque, o qual deve ser controlado, bem como as devidas saídas a serem efetuadas à medida que os produtos forem retirados do estoque físico. O sistema emite relatório para os produtos abaixo do estoque mínimo.

É feito um controle de usuários para melhor manuseio e segurança das informações inseridas e atualizadas no sistema. Existem dois tipos de usuário: a nutricionista, que possui o privilégio de administrador, e os funcionários estoquistas, com funções mais limitadas.

Identificador	Descrição	Depende de
RF01	O sistema deverá possuir cadastro de usuários.	
RF02	O sistema deverá possuir a autenticação do usuário através do <i>login</i> .	RF01
RF03	Os níveis de acesso aos recursos do sistema serão de acordo com os tipos.	RF02
RF04	O sistema permitirá cadastro de todos os produtos utilizados no RU, bem como os fornecedores e categorias.	RF03
RF05	O sistema permitirá entrada de produtos para formação de estoque.	RF04
RF06	O sistema permitirá realizar baixa no estoque, à medida que os produtos forem retirados do estoque físico.	RF05
RF07	O sistema emitirá relatório dos produtos que estão com quantidade de estoque abaixo do valor mínimo cadastrado.	RF03, RF04
RF08	O sistema permitirá o cadastro de um modelo de cardápio, onde serão registrados os itens e a quantidade de cada um, e o sistema verifica se os mesmos estão disponíveis no estoque.	RF03, RF04

Quadro 01 - Requisitos Funcionais

A nutricionista pode:

- Cadastrar novos usuários;
- Cadastrar, alterar e atualizar os produtos estocados;
- Fazer movimentações no estoque, cadastrando, alterando ou efetuando baixa;
- Analisar os relatórios com as informações sobre produtos que estão abaixo do estoque mínimo;
- Verificar a disponibilidade dos itens necessários para o cardápio.

Os funcionários estoquistas podem fazer as funções descritas anteriormente, exceto a de cadastrar novos usuários e verificar os produtos disponíveis para formação de cardápio.

No Quadro 02, são apresentados os requisitos não funcionais, com uma pequena descrição seguida por pelas categorias a qual pertencem e suas respectivas dependências.

O Quadro 03 mostra as regras de negócios com seus respectivos identificadores, uma descrição e suas dependências.

Identificador	Descrição	Categoria	Depende de
RNF01	O acesso as funcionalidades será realizado através das permissões que cada usuário possuirá, sendo estabelecido no cadastro.	Segurança de Acesso	RF02
RNF02	Persistência das informações deve ser implementada em um Sistema Gerenciador de Banco de Dados (SGBD) MySQL	Manutenibilidade	
RNF03	O sistema deverá apresentar telas intuitivas de fácil aprendizagem	Facilidade de uso	
RNF04	O sistema será utilizado na rede interna, a partir dos principais navegadores disponíveis no mercado.	Portabilidade	
RNF05	O sistema não pode demorar mais do que 5 minutos para responder a uma solicitação.	Desempenho	

Quadro 02 - Requisitos Não Funcionais

Identificador	Descrição	Depende de
RN01	Antes de cadastrar os produtos, devem ser inseridas as categorias e informado os fornecedores.	RF04
RN02	Todos os produtos deverão conter uma quantidade mínima, sendo inserida na hora do cadastro.	RF04
RN03	As saídas deverão obedecer e respeitar o estoque mínimo de cada produto.	RF04, RF06
RN04	O usuário deverá estar devidamente autenticado no sistema para fazer uso de suas funcionalidades.	RF02, RF03

Quadro 03 - Regras de Negócio

Após o levantamento dos requisitos foram modelados os diagramas de caso de uso e de classe, que demonstram o que se pretende desenvolver de acordo com as atividades que os usuários terão disponíveis para suprir suas necessidades.

3.2 Diagramas de Casos de Uso

O diagrama de caso de uso representa os requisitos do sistema, sendo utilizado para documentar as funcionalidades. Envolve a interação dos atores com o sistema, sendo que um ator representa um conjunto de papéis exercido por um usuário do sistema ao interagir com um determinado caso de uso.

Guedes (2011, p.52) afirma que:

O diagrama de casos de uso é de grande auxílio para a identificação e compreensão dos requisitos do sistema, ajudando a especificar, visualizar e documentar as características, funções, serviços do sistema desejados pelo usuário. O diagrama de casos de uso tenta identificar os tipos de usuários que irão interagir com o sistema, quais papéis esses usuários irão assumir e quais funções um usuário específico poderá requisitar.

Um caso de uso deve permitir uma leitura e entendimento das funcionalidades que o sistema deverá possuir, tanto para a equipe de desenvolvimento quanto para os clientes, para que possam participar efetivamente do projeto, sem dúvidas e ou eventuais problemas por não terem um conhecimento fundamental sobre o domínio do sistema.

A Figura 01 apresenta o caso de uso, onde são representadas as funcionalidades do sistema, bem como os atores envolvidos. Os atores que serão os usuários do sistema estão representados por bonecos, enquanto as funcionalidades, ou seja, os casos de uso estão representados por elipses e as retas demonstram as relações entre eles.

No diagrama mostrado na Figura 02, podemos notar que o sistema possuirá dois usuários: estoquista, que poderá desempenhar quase todas as funcionalidades do sistema, enquanto a nutricionista fará uso de todas as funções, mais a verificação do cardápio e o cadastro de usuários, como forma de restringir quem poderá utilizar as funcionalidades do sistema, ou seja, apenas pessoas autorizadas (cadastradas) por esse é que poderão realizar as demais funcionalidades restantes do sistema.

A seguir, especificações detalhando o diagrama de casos de usos, com seus atores envolvidos, requisitos e uma breve descrição, que auxiliará no entendimento de cada caso de uso.

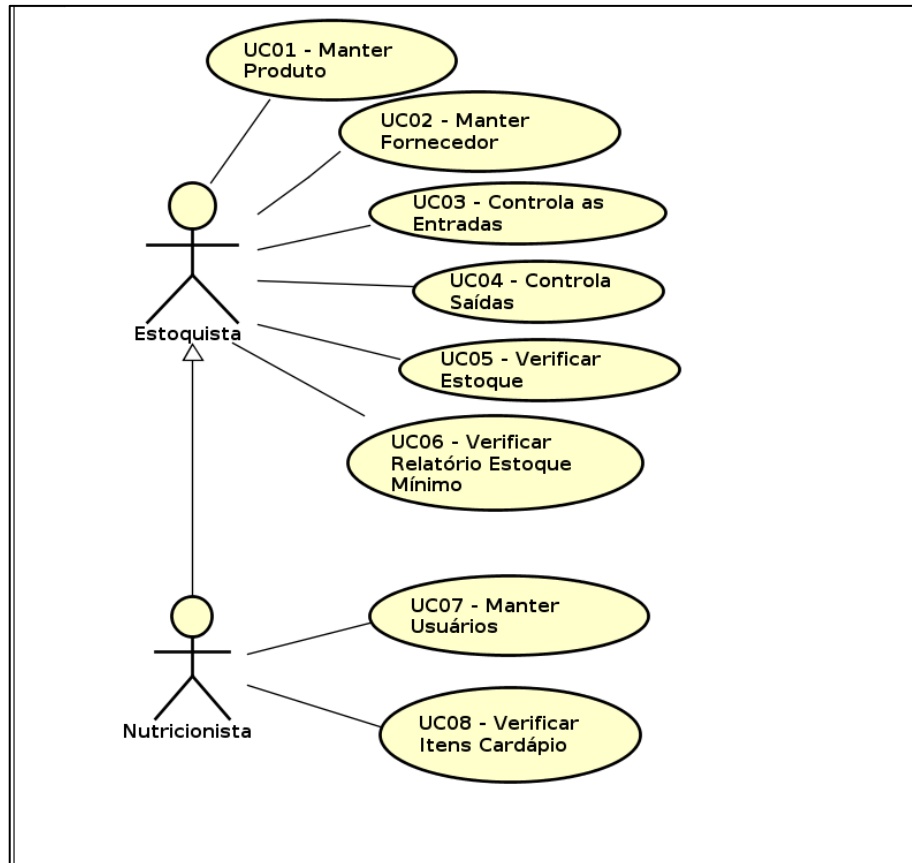


Figura 01 - Diagrama de Casos de uso

- **Caso de Uso 01 – Manter produto**
 - **Ator:** Estoquista e Nutricionista;
 - **Requisitos:** RF04;
 - **Descrição:** Realizar cadastro, edição, exclusão e listagem dos produtos.
- **Caso de Uso 02 – Manter fornecedor**
 - **Ator:** Estoquista e Nutricionista;
 - **Requisitos:** RF04;
 - **Descrição:** Realizar cadastro, edição, exclusão e listagem dos fornecedores.
- **Caso de Uso 03 – Controlar entradas**
 - **Ator:** Estoquista e Nutricionista;
 - **Requisitos:** RF05;
 - **Descrição:** Controlar todas as entradas do sistema é sem dúvida umas das principais atividades a serem realizadas pelo sistema, para que haja um controle rigoroso do estoque sendo formado, facilitando a obtenção das informações de determinado produto.
- **Caso de Uso 04 – Controlar saídas**

- **Ator:** Estoquista e Nutricionista;
- **Requisitos:** RF06;
- **Descrição:** Realizar o controle de todas as saídas de produtos do estoque, levando em conta o estoque mínimo e mantendo as informações referentes ao produto sempre atualizadas e seguras.
- **Caso de Uso 05 – Verificar estoque**
 - **Ator:** Estoquista e Nutricionista;
 - **Requisitos:** RF05 e RF06;
 - **Descrição:** Realizar o controle de todos os produtos do estoque.
- **Caso de Uso 06 – Verificar relatório estoque mínimo**
 - **Ator:** Nutricionista;
 - **Requisitos:** RF07;
 - **Descrição:** Verificar os produtos que estão em quantidade abaixo do estoque mínimo cadastrado.
- **Caso de Uso 07 – Manter usuário**
 - **Ator:** Nutricionista;
 - **Requisitos:** RF01;
 - **Descrição:** Realizar o cadastro, edição, exclusão e listagem dos usuários do sistema. Os usuários estoquistas, somente poderão ser cadastrados pela Nutricionista, para garantir o controle e segurança no acesso as informações.
- **Caso de Uso 08 – Verificar itens cardápio**
 - **Ator:** Nutricionista;
 - **Requisitos:** RF08;
 - **Descrição:** Verifica a disponibilidade dos itens para formação de cardápio.

3.3 Diagrama de Classes

Uma classe é representada como um retângulo subdividido em três compartimentos, separados por linhas horizontais que armazenam o nome da classe, seus atributos e operações. O Diagrama de Classes exibe as classes que compõem o sistema, juntamente com seus atributos, métodos e relacionamentos. Trata-se de um dos mais importantes diagramas da UML, servindo como base para o desenvolvimento de outros diagramas.

Na visualização do diagrama de classes do sistema, podemos ter uma visão de como deverá ser as tabelas do banco de dados e seus relacionamentos. Ele permite que as operações a serem executadas pelo sistema sejam visualizadas facilmente (LOBO, 2009).

O diagrama de classes da Figura 02 representa as classes, com seus atributos e métodos, assim como suas interligações.

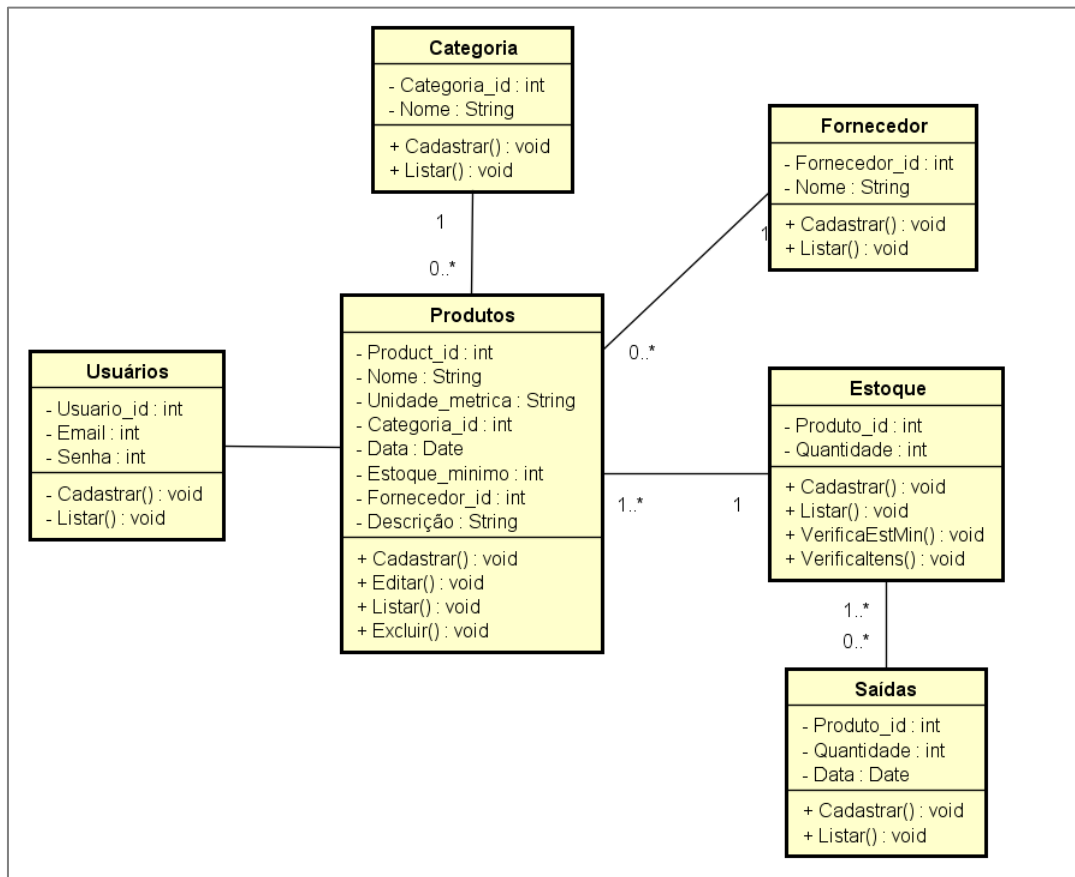


Figura 02 - Diagrama de Classe

4 FUNCIONAMENTO DO SISTEMA

Esse capítulo explica as principais funcionalidades do sistema SCE, onde são mostradas as telas do protótipo desenvolvido, detalhando as funções as quais os usuários possuem acesso.

4.1 Funcionalidades

Para utilizar-se das funcionalidades do sistema, é necessário que o usuário esteja previamente cadastrado. Esse fato garante que apenas as pessoas autorizadas poderão realizar as tarefas disponíveis, de acordo com o tipo de usuário. A tela de *login* pode ser visualizada na Figura 03, sendo necessário o preenchimento do email e senha.

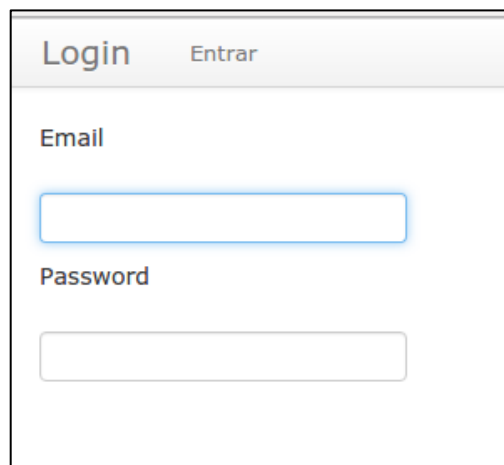
A imagem mostra a interface de login do sistema. No topo, há uma barra de título com o texto "Login" em azul e "Entrar" em cinza. Abaixo, há um campo de entrada rotulado "Email" com um retângulo branco e uma borda azul. Logo abaixo, há um campo de entrada rotulado "Password" com um retângulo branco e uma borda cinza.

Figura 03 - Tela de *Login*

Os usuários podem ser do tipo estoquista ou nutricionista. A nutricionista, que detém privilégios de administrador, é responsável por cadastrar os demais usuários estoquistas e possui mais uma funcionalidade disponível, a de cadastrar modelo de cardápio. O funcionário estoquista pode utilizar as funções que dão acesso ao cadastro de produtos, categorias, fornecedores, entrada/saída do estoque, visualização do relatório dos produtos com estoque abaixo do mínimo.

Na Figura 04 é mostrado o menu para o usuário nutricionista, com as funções disponíveis para esse tipo. Ao fazer *login*, será exibido o email do usuário na parte superior direita, seguida da opção de sair do sistema e registrar novo usuário.

Como uma forma de facilitar a organização das informações, os produtos utilizados estão divididos em categorias. Na Figura 05 pode-se observar a tela onde são inseridas as categorias, sendo que o cadastro das categorias deve ser feito antes dos produtos.

Na Figura 06 são listadas as categorias, com opções para editá-las ou excluí-las e um menu no qual é possível criar novas categorias.



Figura 04 - Menu para usuário nutricionista

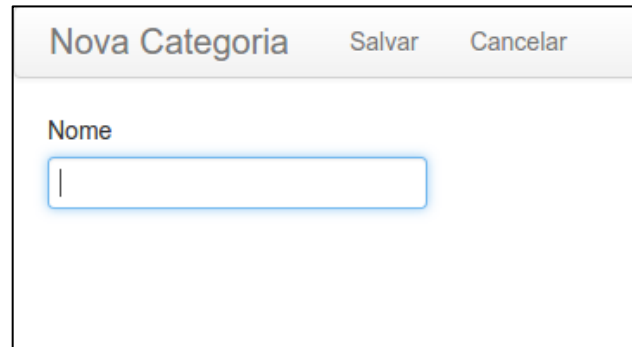
A screenshot of a form titled 'Nova Categoria'. The form has a header bar with the title 'Nova Categoria' and two buttons: 'Salvar' and 'Cancelar'. Below the header, there is a label 'Nome' followed by a text input field with a blue border and a vertical cursor.

Figura 05 - Cadastro de Categorias



Categorias		Criar nova
Nome	Ações	
Frutas	Editar Excluir	
Gêneros Secos	Editar Excluir	
Verduras	Editar Excluir	

Figura 06 - Listar Categorias

Cada produto utilizado deve armazenar o nome do fornecedor ao qual aquele produto foi adquirido. Na Figura 07 tem-se a tela de cadastro de fornecedores e na Figura 08 são mostrados todos que estão cadastrados, com opções para editar, excluir e criar novo.

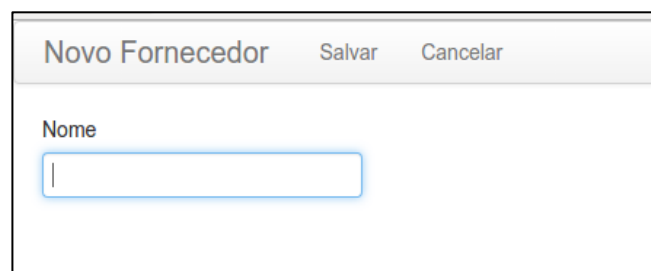
A screenshot of a form titled 'Novo Fornecedor'. The form has a header bar with the title 'Novo Fornecedor' and two buttons: 'Salvar' and 'Cancelar'. Below the header, there is a label 'Nome' followed by a text input field with a blue border and a vertical cursor.

Figura 07 - Cadastro de Fornecedores

Na Figura 09, temos o usuário cadastrando um produto, com os seguintes campos: nome do produto, descrição, quantidade mínima que deve estar disponível no estoque, unidade métrica, data de validade, categoria e fornecedor a qual pertence. Essas informações serão utilizadas para cada entrada ou saída que ocorrer no estoque.

Fornecedores Criar novo	
Nome	Ações
Joaquim	Editar Excluir
José	Editar Excluir

Figura 08 - Listar Fornecedores

Novo Produto Salvar Cancelar	
Nome	<input type="text" value="Arroz"/>
Descrição	<input type="text" value="Pregão 10080"/>
Quantidade estoque minimo	<input type="text" value="20"/>
Unidade	<input type="text" value="KG"/>
Data	<input type="text" value="29/01/2016"/>
Categoria	<input type="text" value="Gêneros Secos"/>
Fornecedor	<input type="text" value="Joaquim"/>

Figura 09 - Cadastro de Produtos

No campo descrição podem ser registradas informações referentes ao produto, como o número do pregão de gêneros. Para o campo categoria e fornecedor, serão listados todos as opções cadastradas previamente, devendo o usuário, portanto, selecionar a categoria a que se

refere o produto e escolher qual o fornecedor dentre as opções disponíveis. Dessa forma, facilita-se a inserção das informações, uma vez que não é necessário digitar a categoria toda vez que cadastrar um novo produto, tendo em vista que irão existir vários produtos para uma mesma categoria. O mesmo aplica-se quanto aos fornecedores e até mesmo para as unidades de medida (a escolher entre quilo KG e unidade UN).

A quantidade mínima é utilizada para verificar se os itens estocados estão em quantidade suficiente para atender a demanda de refeições do RU. O usuário poderá visualizar uma lista com todos os produtos que estão com estoque abaixo da quantidade mínima cadastrada, através de uma função específica. Sendo assim, pode-se prevenir para que nunca falte determinado produto. Na Figura 10 são listados todos os produtos cadastrados.

Produtos		Criar novo					
Nome	Descrição	Quantidade	Unidade	Data	Categoria	Fornecedor	Ações
Arroz	Pregão 10080	20	KG	2016-03-24	Gêneros Secos	Joaquim	Editar Excluir
Feijão	Pregão 10080	20	KG	2016-03-25	Gêneros Secos	Joaquim	Editar Excluir
Óleo	Pregão 10090	10	UN	2016-04-20	Gêneros Secos	Joaquim	Editar Excluir
Abacaxi	Pregão 10090	40	UN	2016-02-10	Frutas	José	Editar Excluir
Melão	Pregão 10090	35	UN	2016-02-10	Frutas	José	Editar Excluir
Melancia	Pregão 10090	10	UN	2016-03-09	Frutas	José	Editar Excluir
Tomate	Pregão 10090	30	KG	2016-03-04	Verduras	José	Editar Excluir

Figura 10 - Listar Produtos

Para cadastrar entradas no estoque, deve-se selecionar um dos produtos registrados anteriormente, informando sua quantidade e a data da entrada. A Figura 11 demonstra a tela do sistema responsável por essa atividade.

Nova Entrada
Salvar
Cancelar

Produto

Arroz
▼

Quantidade

20

Data

05/02/2016

Figura 11 - Nova Entrada no Estoque

A cada nova entrada para um mesmo produto, soma-se a quantidade e atualiza o novo valor no estoque para refletir o valor do estoque físico dos produtos. Como parte do Caso de Uso Verificar Estoque, a listagem dos produtos armazenados é parte essencial no sistema, porque o usuário encontra as quantidades que ficam disponíveis para realização das operações de saída. Na Figura 12 pode-se observar os itens armazenados no estoque.

Estoque		Cadastrar Nova Entrada
Produto	Quantidade	
Abacaxi	40	
Arroz	70	
Melão	20	
Feijão	45	
Melancia	8	
Tomate	20	

Figura 12 - Listar Estoques

À medida que os produtos são utilizados, deve ser registrada uma baixa no estoque, que contém o produto com sua respectiva quantidade a ser retirada. A Figura 13 mostra essa funcionalidade.

Nova Baixa Estoque		Salvar	Cancelar
Produto	<input type="text" value="Arroz"/>		
Quantidade	<input type="text" value="2"/>		

Figura 13 - Baixa no Estoque

Ao cadastrar o produto deve-se informar a quantidade mínima em estoque necessária para atender a demanda de refeições do RU. No sistema existe uma funcionalidade para listar todos os produtos que se encontram em quantidade inferior à mínima exigida, ilustrada na Figura 14.

Na Figura 15 é mostrada a tela para cadastrar um modelo de cardápio, na qual a nutricionista poderá inserir os produtos necessários e a quantidade de cada um. Em seguida poderá acessar a função que lista quais destes produtos inseridos estão disponíveis. Essa tarefa facilita e agiliza o trabalho da nutricionista, que antes tinha que verificar item a item em várias planilhas, para poder saber quais produtos poderiam ser utilizados no cardápio do RU.

Produtos Com Estoque Mínimo		Cadastrar Nova Entrada
Produto	Quantidade	
Melão	20	
Melancia	8	
Tomate	20	

Figura 14 - Produtos com estoque baixo

Nesse capítulo foram demonstradas as principais telas do sistema desenvolvido, apresentando suas funcionalidades e como as mesmas contribuem para atender aos requisitos propostos.

Novo Modelo de Cardápio

Name

Items

Quantity

Quantity

Back

Figura 15 - Verificar Disponibilidade de Produtos

5 CONSIDERAÇÕES FINAIS

Esse projeto foi desenvolvido com a finalidade de apresentar um sistema para auxiliar na gestão de estoques no Restaurante Universitário da Universidade Federal do Piauí – Campus Senador Helvídio Nunes de Barros. A implementação do sistema foi proposta para a plataforma web, permitindo aos usuários utilizar o sistema em qualquer dispositivo que tenha acesso a internet.

Foram descritas as etapas para a construção do sistema, começando pela necessidade identificada, pela qual foi constatada através de entrevistas com funcionários do RU. Sucessivamente, foi descrito o embasamento teórico abordando as principais tecnologias e ferramentas utilizadas durante o desenvolvimento do projeto.

Para melhor compreensão do planejamento e modelagem do sistema, foram explicados os diagramas que tratam dos casos de uso do sistema, além dos requisitos do sistema e os diagramas de classes.

Os dados obtidos como resultados são as descrições das funcionalidades que foram desenvolvidas e testadas para atendimento aos requisitos, as quais foram mostradas através da captura de imagens que tratam das principais telas do sistema.

Através do sistema, os funcionários podem desempenhar suas tarefas com mais facilidade, precisão e rapidez. O processo de controle de estoque torna-se mais eficaz, devido o levantamento das informações que fazem parte desse processo estarem disponíveis de forma organizada.

Para trabalhos futuros, aconselha-se desenvolver funções que gerem o cardápio do RU, realizando combinações com os produtos disponíveis em estoque.

6 REFERÊNCIAS

ARNOLD, JR. TONY. **Administração De Materiais: Uma Introdução**. 3 ed. Rio de Janeiro: Atlas, 2011.

BALLOU, Ronald H. **Logística empresarial: transporte, administração de materiais e distribuição física**. São Paulo: Atlas, 1993.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2ª ed. Rio de Janeiro: Elsevier, 2007.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. 2 ed. Rio de Janeiro: Elsevier, 2005.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 4a ed. São Paulo: Pearson Addison Wesley, 2005.

GAMBLE, Adam; CARNEIRO JR., Cloves; AL BARAZI, Rida. **Beginning Rails 4**. 3 ed. Nova York: Apress, 2013.

GRIFFITHS, David. **Use a cabeça: Rails**. 2 ed. Rio de Janeiro: Alta Books, 2010.

GUEDES, Gilleanes T. A. **UML 2: Uma abordagem prática**. 2 ed. São Paulo: Novatec, 2011.

LAUDON, KENNETH C. ; LAUDON, JANE P. **Sistemas de Informação: com Internet**. 4 ed. Rio de Janeiro: LTC, 1999.

LAUDON, Kenneth C; LAUDON, Jane P. **Sistemas de Informação Gerenciais: Administrando a empresa digital**. 5 ed. São Paulo: Pearson Prentice Hall, 2004.

LOBO, Edson Junio Rodrigues. **Guia Prático de Engenharia de Software**. São Paulo: Digerati Books, 2009.

MILANI, André. **MySQL: Guia do Programador**. São Paulo: Novatec Editora, 2007.

O'BRIEN, JAMES A. **Administração de Sistemas de Informação**. 15 ed. Porto Alegre: AMGH, 2013.

FILHO, Wilson de Pádua Paula. **Engenharia de Software: Fundamentos, Métodos e Padrões**. 3 ed. Rio de Janeiro: LTC, 2005.

PFLEEGER, Shari Lawrence. **Engenharia de Software: Teoria e Prática**. 2 ed. São Paulo: Prentice Hall, 2004.

POZO, Hamilton. **Administração de recursos materiais e patrimoniais: uma abordagem logística**. 6ª ed. São Paulo: Atlas, 2010.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7 ed. Porto Alegre: AMGH, 2011.

RODRIGUES, Andréa. **Desenvolvimento para Internet**. Curitiba: LTC, 2010.

SABBAGH, Rafael. **SCRUM: Gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.

SOUZA, Lucas. **Ruby: Aprenda a programar na linguagem mais divertida**. São Paulo: Casa do Código, 2012.

STAIR, RALPH M.; REYNOLDS, GEORGE W. **Princípios de Sistemas de Informação: uma abordagem gerencial**. Tradução da 6ª edição. São Paulo: Pioneira. Thompson Learning, 2006.



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
- () Dissertação
- (X) Monografia
- () Artigo

Eu, **Valdirene de Araújo Silva**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Sistema Web para Controle de Estoque do Restaurante Universitário do Campus Senador Helvídio Nunes de Barros** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 08 de março de 2016.

Valdirene de Araújo Silva

Assinatura