

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

GLAUCO FERNANDO LEITE MOMOLLI

SISTEMA WEB PARA DISTRIBUIÇÃO DE ALIMENTOS

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

GLAUCO FERNANDO LEITE MOMOLLI

SISTEMA WEB PARA DISTRIBUIÇÃO DE ALIMENTOS

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Edilson Pontarolo

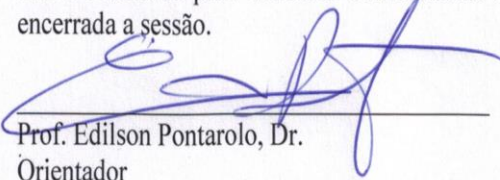
PATO BRANCO

2015

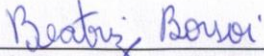
ATA Nº: 274

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO GLAUCO FERNANDO LEITE MOMOLLI.

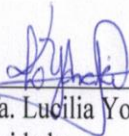
Às 15:15 hrs do dia 26 de novembro de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Edilson Pontarolo (Orientador), Beatriz Terezinha Borsoi (Convidada) e Lucilia Yoshie Araki (Convidada), para avaliar o Trabalho de Diplomação do aluno Glauco Fernando Leite Momolli, matrícula 116690, sob o título **Sistema web para distribuidora de alimentos**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:00 hrs foi encerrada a sessão.



Prof. Edilson Pontarolo, Dr.
Orientador



Profa. Beatriz Terezinha Borsoi, Dr.
Convidada



Profa. Lucilia Yoshie Araki, M.Sc.
Convidada

RESUMO

MOMOLLI, Glauco Fernando Leite. Sistema web para distribuição de alimentos. Monografia. 53f – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

A Prefeitura Municipal de Pato Branco (PMPB) participa do Programa de Aquisição de Alimentos (PAA) do governo federal, que incentiva a compra de alimentos orgânicos de pequenos produtores rurais e sua distribuição para consumo em escolas e outras entidades. Dada a variedade, sazonalidade e caráter perecível desses alimentos, bem como a quantidade e distribuição geográfica dos produtores rurais e entidades consumidoras, um controle manual eficiente do processo de distribuição não se faz viável. Este trabalho apresenta um sistema web para gestão do PAA pela prefeitura. O sistema foi implementado utilizando-se o framework Phalcon e a linguagem de programação PHP. O sistema permite o cadastramento das entidades e dos produtores rurais e respectivos alimentos e estoques potenciais a serem fornecidos ao longo do ano. Por meio da interface web do sistema, os usuários em cada entidade podem registrar a demanda semanal de alimentos. Com base nos estoques contínuos previstos e na junção das demandas indicadas, o sistema gera uma lista de produtores, produtos e quantidades para aquisição semanal, permite a emissão dos recibos para conferência dos alimentos recebidos e distribuídos, bem como sua respectiva baixa na base de dados do sistema. O sistema ainda não foi testado com usuários em relação a aspectos de usabilidade e atendimento dos requisitos funcionais.

Palavras-chave: Linguagem de programação PHP. Phalcon framework. Sistema web. Programa de aquisição de alimentos.

ABSTRACT

MOMOLLI, Glauco Fernando Leite. Web system for food distribution. Monograph - Course of Technology in Systems Analysis and Development, 53p. Federal Technological University of Paraná, Campus Pato Branco. Pato Branco, 2015.

The Pato Branco City Hall (PMPB) participates in the Food Acquisition Program (PAA) from the federal government, which encourages the purchase of organic foods from small farmers and their distribution to consumption in schools and other entities. Given the variety, seasonal and perishable nature of these foods as well as the number and geographic distribution of farmers and consumer organizations, an efficient manual control of the distribution process is not feasible. This paper presents a web system for PAA management by the city. The system was implemented using the Phalcon framework and PHP programming language. The system allows the registration of organizations and farmers and their feed and potential inventories to be supplied throughout the year. Through the system's web interface, users in each entity can register the weekly demand for food. Building on the expected continuous inventory and at the junction of the specified demands, the system generates a list of producers, products and quantities for weekly acquisition, allows the issuance of receipts to the conferencing of the received and distributed food as well as their respective low in the system database, as well as their respective withdraw of the system database. The system has not been tested with users regarding aspects of usability and attendance of functional requirements.

Keywords: PHP. Phalcon Framework. Web System. PAA.

LISTA DE FIGURAS

Figura 1 - Diagrama de caso de uso	24
Figura 2 - Diagrama de atividades.	25
Figura 3 - Diagrama de classes.....	26
Figura 4 - Diagrama de Entidade Relacionamento (a)	31
Figura 5 - Diagrama de Entidade Relacionamento (b)	31
Figura 6 - Tela de login do usuário	32
Figura 7 - Tela de cadastro de usuário	33
Figura 8 - Tela de lista de usuários.....	33
Figura 9 - Tela de edição de cadastro do produtor	34
Figura 10 - Tela de pesquisa de requisição	34
Figura 11 - Tela de cadastro de requisição de compra	35
Figura 12 - Tela de listagem das requisições semanais	35
Figura 13 - Tela de produtos por requisição semanal	36
Figura 14 - Tela de listagem de produtores por produto	36
Figura 15 - Tela de pedido de compra.....	37
Figura 16 - Tela de compra	37
Figura 17 - Tela de requisição semanal	38
Figura 18 – Relatório de quantia entregue para instituição.	38
Figura 19 – Relatório de usuários.	39
Figura 20 – Relatório cadastro de produtos.....	39
Figura 21 - Pastas e arquivos do projeto.	40

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas	15
Quadro 2 - Requisito Funcional Cadastrar Produtores	20
Quadro 3 - Requisito Funcional Cadastrar Produtores	20
Quadro 4 - Requisito Funcional Cadastrar Usuários	21
Quadro 5 - Requisito Funcional Emitir Recibo	21
Quadro 6 - Requisito Funcional Consultar Produtores de um Produto	22
Quadro 7 - Requisito Funcional Requisição de Compra	22
Quadro 8 - Requisito Funcional Pedido de Compra.....	22
Quadro 9 - Requisito Funcional Registrar Requisição	22
Quadro 10 - Requisito Funcional Registrar Compra	23
Quadro 11 – Requisito Funcional Registrar Ordem de Entrega dos Produtos	23
Quadro 12 - Requisitos Suplementares	24
Quadro 13 - Descrição da classe usuário	27
Quadro 14 - Descrição da classe instituição	27
Quadro 15 - Descrição da classe requisição de compra.....	27
Quadro 16 - Descrição da classe itens requisição	28
Quadro 17 - Descrição da classe produto.....	28
Quadro 18 - Descrição da classe itens compra	28
Quadro 19 - Descrição da classe itens pedido	29
Quadro 20 - Descrição da classe compra	29
Quadro 21 - Descrição da classe pedido compra	29
Quadro 22 - Descrição da classe produtor	30
Quadro 23 - Descrição da classe ordem de entrega	30
Quadro 24 - Descrição da classe requisição geral	30

LISTAGEM DE CÓDIGO

Listagem 1 - Procedimento de busca de usuário	41
Listagem 2 - Edição de Usuário	42
Listagem 3 - Criação do usuário.....	43
Listagem 4 - Função save	44
Listagem 5 - Função para excluir usuário.....	45
Listagem 6 - Função utilizada para listar produtos	47
Listagem 7 - Listagem de produtor.....	48
Listagem 8 - Iniciando sessão do usuário	48
Listagem 9 - View requisição	49
Listagem 10 - Código relatório de entrega	51

LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i>
DDL	<i>Data Definition Language</i>
DER	Diagrama Entidade Relacionamento
DML	<i>Data Manipulation Language</i>
HTML	<i>HyperText Markup Language</i>
MVC	<i>Model View Controller</i>
PAA	Programa de Aquisição de Alimentos
PHP	<i>Hypertext Preprocessor (Personal Home Page)</i>
SGDB	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1 INTRODUÇÃO	10
1.1 CONSIDERAÇÕES INICIAIS	10
1.2 OBJETIVOS	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 JUSTIFICATIVA	11
1.4 ESTRUTURA DO TRABALHO	12
2 REFERENCIAL TEÓRICO	13
2.1 <i>MODEL VIEW CONTROLLER</i>	13
3 MATERIAIS E MÉTODO	15
3.1 MATERIAIS	15
3.2 MÉTODO.....	18
4 RESULTADOS	19
4.1 ESCOPO DO SISTEMA	19
4.2 MODELAGEM DO SISTEMA	19
4.3 APRESENTAÇÃO DO SISTEMA	32
4.4 IMPLEMENTAÇÃO DO SISTEMA	40
5 CONCLUSÃO	52
REFERÊNCIAS	53

1 INTRODUÇÃO

Neste capítulo são apresentadas as considerações iniciais, que contém a visão geral do sistema. Também são apresentados os objetivos e a justificativa do trabalho. Por fim está a estrutura do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

O Programa de Aquisição de Alimentos (PAA) propicia a aquisição de alimentos de agricultores familiares, com isenção de licitação, a preços compatíveis aos praticados nos mercados regionais. Os produtos comprados desses agricultores são destinados a instituições públicas e filantrópicas (hospitais, escolas, universidades, creches e presídios) e estabelecimentos privados, como supermercados (BRASIL, 2014). Além disso, esses alimentos também contribuem para a formação de cestas de alimentos distribuídas a grupos populacionais específicos. O projeto contribui ainda para a formação de estoques estratégicos e para o abastecimento de mercado institucional de alimentos, que compreende as compras governamentais de gêneros alimentícios para fins diversos. Permitindo, também, aos agricultores familiares que estoquem seus produtos para possam comercializá-los a preços mais justos.

Como a cota de venda desses produtores é baixa e também pela grande diversidade de alimentos solicitados por essas entidades, torna o processo trabalhoso, pois é necessário fazer a gestão da demanda de alimentos que as instituições necessitam. Como esse processo é semanal e o controle é feito através de planilhas, verificou-se a necessidade de desenvolvimento de um sistema para realizar esse controle.

O sistema foi implementado em *PHP* juntamente com *Phalcon Framework* que é utilizado para criar projetos na estrutura *Model View Controller (MVC)*. Além dos controles de aquisição distribuição e do pedido de compra o sistema também vai gerar relatórios gerencias para conferência de produtos adquiridos, produtos entregues, entre outros.

1.2 OBJETIVOS

O objetivo geral corresponde ao resultado principal deste trabalho, uma aplicação *Web* para o setor de distribuição de alimentos da prefeitura. Os objetivos específicos complementam o objetivo geral, com detalhes técnicos sobre suas funcionalidades.

1.2.1 Objetivo Geral

Desenvolver um sistema *Web* que realizará o controle na solicitação e distribuição de alimentos, destinados às instituições públicas e filantrópicas ligadas ao programa PAA para a prefeitura de pato branco.

1.2.2 Objetivos Específicos

- Facilitar a requisição de alimentos por parte das escolas e entidades, agilizando o processo como um todo.
- Proporcionar maior eficiência para identificar os produtores rurais que possuem limite e cota disponível, e que possuam o produto disponível para entrega.
- Efetuar registro de compra dos produtores rurais, validando se o produtor possui cota para venda.

1.3 JUSTIFICATIVA

O processo realizado na distribuição de alimentos pela prefeitura de Pato Branco é controlado atualmente por meio de planilhas e apresenta grande volume de demanda semanal. Esses fatores acarretam falta de confiabilidade nos dados e impossibilitam a geração de relatórios, tornando necessário um sistema *Web* tanto para facilitar a comunicação entre a entidade e a distribuidora no processo de solicitar alimento, quanto no próprio controle da distribuidora, que disporá relatórios com informações relevantes para realizar o gerenciamento do processo.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos, no primeiro é apresentada a introdução com a visão geral do sistema resultante deste trabalho. No Capítulo 2 está o referencial teórico, sobre os padrões de projetos *Model View Controller*. O Capítulo 3 consta os materiais e métodos utilizados na análise e desenvolvimento do sistema. No Capítulo 4 possui a modelagem do banco de dados, o escopo do sistema juntamente com as principais janelas do sistema e suas codificações. No Capítulo 5 é apresentada a conclusão e as perspectivas futuras para continuidade do trabalho.

2 REFERENCIAL TEÓRICO

Nesse capítulo é apresentado o referencial teórico que fundamenta a construção desse trabalho.

2.1 MODEL VIEW CONTROLLER

A arquitetura Modelo *Model View Controller* (MVC) tem como finalidade organizar o projeto e agilizar o processo de desenvolvimento fracionando o projeto em três camadas que serão descritas a seguir.

A primeira camada é o Modelo (*Model*), responsável por armazenar os modelos de classe utilizadas em um projeto, dessa forma a camada de *Controller* acessa a camada de *View* para obter as classes. O modelo (*model*) consiste nos dados da aplicação, lá estarão todas as lógicas necessárias para acessar e atualizar os dados da aplicação. O modelo fornece ao controlador acesso para que o mesmo possa utilizar as funcionalidades do *model* em sua própria lógica. No modelo podemos encontrar as definições de dados, como número de colunas, tipos de dados, relação com outras classes, pesquisas e tudo que está relacionado à dados (RABELLO, 2015).

A segunda camada é a *View*, essa é responsável por conter a interface gráfica com o usuário e armazenar o código HTML das páginas (RABELLO, 2015). A visualização (*view*) define como os dados serão mostrados para o usuário, toda a estilização da tela, bem como cada dado enviado pelo *controller* é definido nas visualizações. Com isso podemos concluir que caso desejar alterar o modo como os dados são mostrados, não é necessário alterar todo o sistema, podendo modificar apenas a camada de visualização sem afetar o sistema como um todo, Estes aspectos fazem do MVC uma arquitetura que facilita a manutenção.

A última camada é a Controladora (*Controller*) que tem como objetivo acessar o banco de dados e realizar operações como cadastros, acesso a informações e geração de relatórios (RABELLO, 2015). O controlador guarda todo o comportamento da aplicação, nele estão todas as funções que irão interpretar o que o usuário deseja fazer na aplicação, feito isso o mesmo utiliza-se de funções do *model* para acessar os dados necessários e então completar a requisição. Por exemplo, em aplicações *web* estas ações são cliques em botões, seleção de um

item no menu, salvar dados entre outras coisas. Com base nestas escolhas do usuário, o controlador também determina qual visualização dar para aquela ação e então chama a *view*..

2.2 VANTAGENS DA ARQUITETURA MVC.

Com o aumento da complexidade dos códigos e a demanda de organização em grandes projetos, a arquitetura MVC tem se mostrado de grande utilidade. Uma vez que através dela é possível setorizar um projeto cujo pessoal responsável pela interface gráfica trata somente da camada de visualização, deixando toda a regra de negócio segura de alterações que poderiam ocasionar erros ou falhas no sistema. Além de ser amplamente utilizado pelos novos *frameworks* do mercado, o Phalcon que foi utilizado no projeto, faz uso da estrutura *MVC* como padrão para a criação de projetos.

A característica dessa arquitetura é dividir um problema em vários problemas menores e de menor complexidade. Dessa forma, qualquer tipo de alteração feita em uma das camadas não interfere nas demais, facilitando a atualização de layouts, alteração nas regras de negócio e adição de novos recursos. Em caso de grandes projetos, o MVC facilita muito a divisão de tarefas entre a equipe (BASTOS, 2015)

3 MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar como foi o processo para alcançar o objetivo do trabalho, listando seus materiais e método utilizado no desenvolvimento do trabalho.

3.1 MATERIAIS

O Quadro 1 contém uma relação das ferramentas utilizadas na elaboração deste trabalho.

Ferramenta Tecnologia	Versão	Referência	Finalidade
Phalcon	2.0.4	https://phalconphp.com/pt/	Framework php que auxilia no Desenvolvimento.
PHP	5.5	https://www.php.net/	Linguagem de programação.
MySQL	5.0.11	https://www.mysql.com/	Sistema Gerenciador de Banco de Dados.
Bootstrap	3	http://getbootstrap.com/	Framework de desenvolvimento ágil HTML, CSS e Java Script.
XAMPP	3.2.1	https://www.apachefriends.org/pt_br/	Ambiente de Desenvolvimento.
Visual Paradigm	10.0	www.visual-paradigm.com/	Modelagem do Sistema.
ERwin	7.5	erwin.com/	Modelagem do Banco de Dados.
PHPMyAdmin		https://www.phpmyadmin.net/	Interface Gráfica para manutenção do banco.

Quadro 1 – Tecnologias e ferramentas utilizadas

a) Phalcon

Phalcon é projeto *OpenSource*, um *framework* completo para PHP escrito como uma extensão-C, otimizado para alto desempenho. Não necessitando o conhecimento da linguagem C, uma vez que a funcionalidade é exposta como classes PHP, prontos para usar. Phalcon também é fracamente acoplada, permitindo o uso de seus objetos como componentes de cola com base nas necessidades de sua aplicação (PHALCON, 2015). Esse *framework* auxiliou no processo de criação dos cadastros e edições de tabelas, tornando rápida a criação

do projeto com estrutura *MVC*. Porém, essa tecnologia possui alguns problemas, como a falta de documentação em português, além disso, outro problema é o tempo de aprendizagem e domínio da ferramenta.

b) PHP

O *Hypertext Preprocessor PHP* (PHP) é uma linguagem de script código aberto (*open source*) de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML (PHP, 2015).

c) MySQL

O MySQL é um sistema de gerenciamento de banco de dados. Um banco de dados é uma coleção de dados estruturados. Para adicionar, acessar e processar dados armazenados em um banco de dados, é necessário um sistema de gerenciamento de banco de dados, como MySQL Server (MYSQL, 2015).

d) Bootstrap

Desenvolvido pela equipe do Twitter, o Bootstrap é um *framework front-end* de código aberto (*opensource*). Em palavras simples, é um conjunto de ferramentas criadas para facilitar o desenvolvimento de sites e sistemas *web* (NASCIMENTO, 2015)

Compatível com HTML5 e CSS3, o *framework* possibilita a criação de layouts responsivos e o uso de grids, permitindo que seu conteúdo seja organizado em até 12 colunas e que se comporte de maneira diferente para cada resolução. Pelo fato de possuir vários componentes atuais para programação para internet agilizou o processo de estilização das páginas.

A escolha desse *framework* foi feita devido ao fato de o processo de criação de componentes para interface impele muito tempo, agilizando o processo de desenvolvimento.

e) XAMPP

O objetivo do XAMPP é construir uma distribuição fácil de instalar para desenvolvedores que utilizam o Apache. Atualmente distribuições para Windows, Linux e OS X (APACHE, 2015).

O Xampp foi utilizado por trazer uma ambiente web para desenvolvimento. Anexo a um pacote de ferramentas como o MYSQL, o PHPMYADMIN, disponibilizando para o desenvolvimento um ambiente já estruturado.

f) Visual Paradigm

A Visual Paradigm é uma ferramenta *Computer Aided Software Engineering* (CASE) voltada para *Unified Modeling Language* (UML), que disponibiliza, diagramas como caso de uso, classes e sequência, dentre outros. Além disto, a Visual Paradigm, nas versões Professional e Enterprise, possui ferramentas de auxílio ao mapeamento do modelo de classes para o modelo de dados relacional sendo, inclusive, capaz de gerar tabelas (ARAÚJO, 2015).

g) Erwin

O CA ERwin® Data Modeling oferece um ambiente colaborativo de modelagem de dados para o gerenciamento de dados corporativos por meio de uma interface gráfica intuitiva. Tanto os participantes administrativos quanto os técnicos podem ter uma visão comum das informações em contexto, através de um portal com base na Web e de ferramentas de design para desktop, com suporte de um repositório de modelos de nível corporativo. Com uma visão centralizada nas principais definições de dados, pode-se ter uma melhor compreensão dos dados corporativos, gerenciados de maneira mais eficiente e econômica (ERWIN, 2015). Utilizado para a modelagem do banco de dados e geração de código *SQL* para formatação do banco de dados.

h) PhpMyAdmin

O PhpMyAdmin pode gerenciar um servidor MySQL inteiro assim como um simples banco de dados. Para conseguir realizar as tarefas é necessário configurar um usuário de MySQL que possa apenas ler/escrever apenas no banco de dados desejado. (PHPMYADMIN, 2010) Foi utilizado para a manutenção do banco de dados, operações como: executar script gerado pelo Erwin e também para realizar edições *Data Definition Language DDL* e *Data Manipulation Language DML* diretamente no banco.

3.2 MÉTODO

As principais etapas relacionadas ao desenvolvimento deste trabalho estão segmentadas em:

a) Levantamento de Requisito

Foi realizada uma entrevista com o gestor da área de distribuição de alimentos da prefeitura, em que o mesmo informou os pontos críticos da sua regra de negócio e as principais funcionalidades do sistema.

b) Análise

Na análise foi realizada a modelagem dos requisitos do sistema identificados na etapa de levantamento de requisitos. Esses requisitos auxiliaram a definir o funcionamento do sistema e as tecnologias a serem utilizadas para atender melhor os requisitos de negócio identificados. A ferramenta *Visual Paradigm* foi utilizada para realizar a modelagem do sistema por meio dos diagramas de casos de uso, classes e atividades.

c) Implementação

Terminada a análise do sistema foi montando o ambiente de desenvolvimento a fim de escrever a regra de negócio em nível de código de máquina. Foi utilizada a linguagem PHP com o Framework Phalcon para a codificação e o SGDB MySQL para a persistência dos dados.

d) Teste

Foram realizados testes funcionais e de unidade para verificar se o software atenderá aos requisitos básicos de usabilidade, além de validar a existências de falhas no software.

4 RESULTADOS

Neste capítulo apresentam-se o escopo, a modelagem e a apresentação do sistema obtido na fase de análise.

4.1 ESCOPO DO SISTEMA

O sistema visa automatizar o controle na distribuição dos alimentos adquiridos por produtores rurais, para as entidades públicas do município de Pato Branco, dentro do programa PAA. Os produtores rurais que participam desse programa devem ter seus dados cadastrados no sistema e os produtos fornecidos para que seja possível saber qual o contato do produtor e qual o limite de produção ele pode fornecer.

Visto que várias instituições fazem requisição de alimentos, o sistema será web, na qual cada instituição vai solicitar os alimentos que estão em falta. Cada usuário estará vinculado a uma instituição, exceto o usuário que faz o controle de distribuição. Os usuários vinculados à instituição possuem acesso somente a requisição de alimentos.

O sistema faz o controle de estoque por instituição e também do fornecedor e do depósito da distribuidora. Os produtos são divididos em perecíveis e comuns, dessa forma é evitado o desperdício de alimentos perecíveis.

Também são gerados alguns relatórios para melhorar o controle/Auditoria da distribuição dos alimentos. São gerados relatórios de alimentos distribuídos por entidade e também produtos por fornecedor.

4.2 MODELAGEM DO SISTEMA

A seguir são apresentados os requisitos identificados na fase de levantamento de requisitos para o sistema, serão apresentados os diagramas de casos de uso, atividades, classes e de entidades e relacionamentos gerados.

4.2.1 Requisitos funcionais e Não funcionais

Os requisitos funcionais identificados para o sistema estão listados nessa seção com abreviação F que significa Requisito Funcional. Os requisitos Não-Funcionais estão relacionados com a regra de negócio, os requisitos Não funcionais estão listados NF significa Requisito Não-Funcional. Os Requisitos suplementares pertencem a categorias de segurança, integridade e interface, e são representados com S. Os requisitos estão listados do Quadro 2 ao 12.

O Quadro 2 apresenta o requisito funcional Cadastrar Produtores e os requisitos não funcionais relacionados.

F1 Cadastrar Produtores		Oculto ()			
Descrição: O sistema deve oferecer ao usuário atendente a possibilidade de cadastrar os produtores rurais, contendo os dados pessoais. Além disso, deve ser permitido editar e excluir cadastros.					
Requisitos Não – Funcionais					
Nome	Restrição	Categoria	Desejável	Permanente	
NF Inclusão de foto	1.1 O formulário de cadastro de produtores deve possibilitar a inserção de foto do Produtor.	Interface	()	(x)	
NF Cadastro do Sistema	1.2 O Produtor rural deve estar vinculado a um programa e apresentar uma documentação comprovando.	Regra de Negócio	()	(X)	
NF Observação	1.3 O sistema deve conter um meio de vincular ao cadastro do usuário o seu histórico de produtos	Regra de Negócio	()	(X)	

Quadro 2 - Requisito Funcional Cadastrar Produtores

O Quadro 3 apresenta o requisito funcional Cadastrar Instituição e os requisitos não funcionais relacionados.

F2 Cadastrar Instituição		Oculto ()			
Descrição: O sistema deve oferecer ao usuário administrador a possibilidade de cadastrar as Instituições que dependem da distribuidora de alimentos, contendo os dados de registro de cada Instituição. Além disso, deve ser permitido editar e excluir cadastros.					
Requisitos Não - Funcionais					
Nome	Restrição	Categoria	Desejável	Permanente	
NF Vínculo	2.1 As instituições que estiverem cadastradas no sistema devem possuir algum vínculo com a prefeitura. A verificação de vínculo não é realizada por meio do sistema.	Regra de Negócio	()	(X)	

Quadro 3 - Requisito Funcional Cadastrar Produtores

Quadro 4 apresenta o requisito funcional Cadastrar Usuários e os requisitos não funcionais relacionados.

F3 Cadastrar Usuários		Oculto ()		
Descrição: O sistema deve oferecer ao usuário a possibilidade de cadastrar os usuários que fazem parte do processo do sistema, contendo os dados de <i>login</i> e senha e nível de acesso. Além disso, deve ser permitido editar e excluir cadastros.				
Requisitos Não – Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Tipo de Usuários	O sistema basicamente possui 2 tipos de usuários: - Usuário da instituição, que terá a finalidade de solicitar uma requisição de compra para a instituição. - Usuário Administrador que terá acesso a todo sistema e também vai realizar a parte operacional de conferir os pedidos e gerar compras	Regra de Negócio	()	(x)
NF3.2 Nível de acesso de Usuário	O usuário instituição tem acesso somente a ao menu de requisição. Já o usuário administrador terá acesso aos cadastros, ao menu de movimentação, aos relatórios e ao menu de requisição.	Regra de Negócio	()	(x)

Quadro 4 - Requisito Funcional Cadastrar Usuários

O Quadro 5 apresenta o requisito funcional Emitir Recibo e os requisitos não funcionais relacionados.

F4 Emitir recibo para instituição		Oculto ()		
Descrição: O sistema deve emitir uma lista com os itens entregues em uma instituição para que a mesma confirme o recebimento dos itens.				

Quadro 5 - Requisito Funcional Emitir Recibo

O Quadro 6 apresenta o requisito funcional Consultar produtores de um produto e os requisitos não funcionais relacionados.

F5 Consultar Produtores de um Produto		Oculto ()		
Descrição: O sistema deve oferecer ao usuário a função de Consultar os Produtores que fornecem determinado alimento. Essa consulta será utilizada quando uma instituição realizar a solicitação de alimentos				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 5.1 Pré-Requisito	Serão listados os Produtores que tem o produto e ainda tem saldo	Interface	()	(x)

	de venda no mês.			
--	------------------	--	--	--

Quadro 6 - Requisito Funcional Consultar Produtores de um Produto

O Quadro 7 apresenta o requisito funcional Requisição de Compra e os requisitos não funcionais relacionados.

F6 Requisição de Compra		Oculto ()		
Descrição: O sistema deve oferecer ao usuário que representa a instituição a possibilidade de cadastrar o pedido de compra semanal de alimentos, em que o usuário deve informar quais os itens estão em falta quais e a são necessários repor no estoque.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 6.1 Restrição	Somente usuários que representem uma instituição terão acesso a essa funcionalidade.	Regra de Negócio	()	(x)

Quadro 7 - Requisito Funcional Requisição de Compra

O Quadro 8 apresenta o requisito funcional Pedido de Compra e os requisitos não funcionais relacionados.

F7 Pedido de Compra		Oculto ()		
Descrição: O sistema deve oferecer a possibilidade de realizar o controle de agendamento do sistema, em que cada compra em aberto possui uma data, para que o produto seja entregue.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 7.1 Não Entregue	Caso o produtor não entregue o produto ou entregue parte da quantidade solicitada, será selecionado outro produtor para entregar a quantidade pendente, se por acaso não existir outro produtor, será feita uma compra direto do mercado.	Regra de Negócio	()	(x)

Quadro 8 - Requisito Funcional Pedido de Compra

O Quadro 9 apresenta o requisito funcional Registrar Requisição de compra Semanal e os requisitos não funcionais relacionados.

F8 Registrar Requisição Semanal Compra		Oculto ()		
Descrição: O sistema deve oferecer ao usuário a possibilidade de listar os pedidos de compra das instituições, e a partir deles gerar a requisição semanal de compra, que contém os itens a serem comprados de todos os pedidos de instituições.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente

Quadro 9 - Requisito Funcional Registrar Requisição

O Quadro 10 apresenta o requisito funcional Registrar Compra e os requisitos não funcionais relacionados.

F9 Registrar Compra		Oculto ()		
Descrição: O sistema deve oferecer ao usuário a função de realizar a compra de produtores a partir da demanda de compra semanal, na qual a cada semana será realizada compra de produtores.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 9.1 Bloqueio de Compra	O Produtor terá um limite de venda de produtos em Reais, ou seja, o produtor terá um limite por programa e caso esse limite seja excedido o sistema vai bloquear a venda e vai listar outros fornecedores.	Regra de Negócio	()	(x)
NF 9.2 Produto não entregue	Caso o produtor não entregue a quantidade determinada ou total, será aberto uma nova compra de fornecedor para suprir essa demanda. Caso não haja mais fornecedores, o será realizada uma compra do mercado.	Regra de Negócio	()	(x)

Quadro 10 - Requisito Funcional Registrar Compra

O Quadro 11 apresenta o requisito funcional Registrar Ordem de Entrega dos Produtos e os requisitos não funcionais relacionados.

F10 Registrar Ordem de Entrega dos Produtos		Oculto ()		
Descrição: Após realizar as compras e receber os alimentos dos produtores, o sistema deve fornecer ao usuário a função de gerar a distribuição dos produtos entre as entidades, para que todas possam receber os alimentos solicitados.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 10.1 Distribuição da Instituição	A distribuição para as entidades será realizada de forma manual, selecionando o item e quantidade desejados.	Regra de Negócio	()	(x)

Quadro 11 – Requisito Funcional Registrar Ordem de Entrega dos Produtos

O Quadro 12 apresenta os Requisitos Suplementares relacionados com o comportamento do sistema.

Nome	Restrição	Categoria	Desejável	Permanente
S1	O sistema deve possuir uma interface simples, de forma a facilitar o seu uso efetivo por parte dos usuários.	Usabilidade	()	(x)
S2	O sistema deve ser web, pois deve ser acessado por várias instituições para fazer pedidos de compra.	Usabilidade	()	(x)

Quadro 12 - Requisitos Suplementares

4.2.2 Diagrama de casos de uso

Na fase de levantamento de requisitos foram identificadas as principais atividades do sistema, ilustradas na Figura 1 como caso de uso. No qual o usuário administrador realiza operações críticas do sistema, como cadastros e movimentação, enquanto o usuário ligado a uma instituição possui acesso somente ao lançamento de requisição.

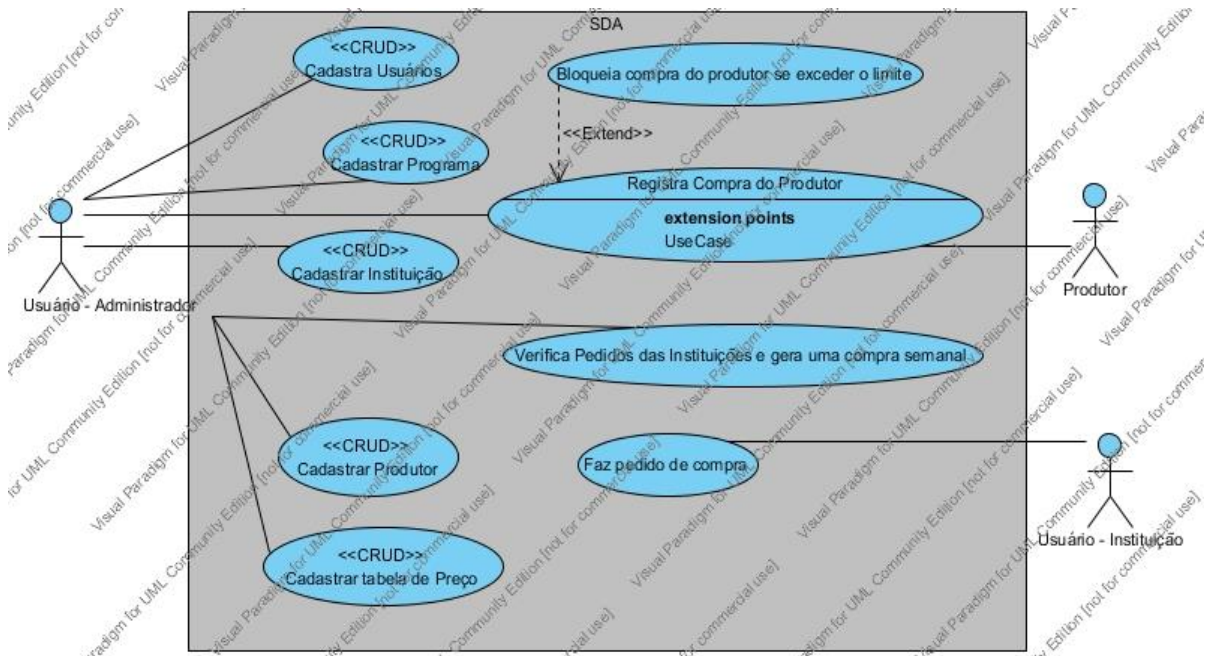


Figura 1 - Diagrama de caso de uso

4.2.3 Diagrama de atividades

A Figura 2 apresenta o diagrama de atividades do sistema, que exemplifica o processo de levantamento de fornecedor e agendamento a entrega, por meio do pedido de venda.

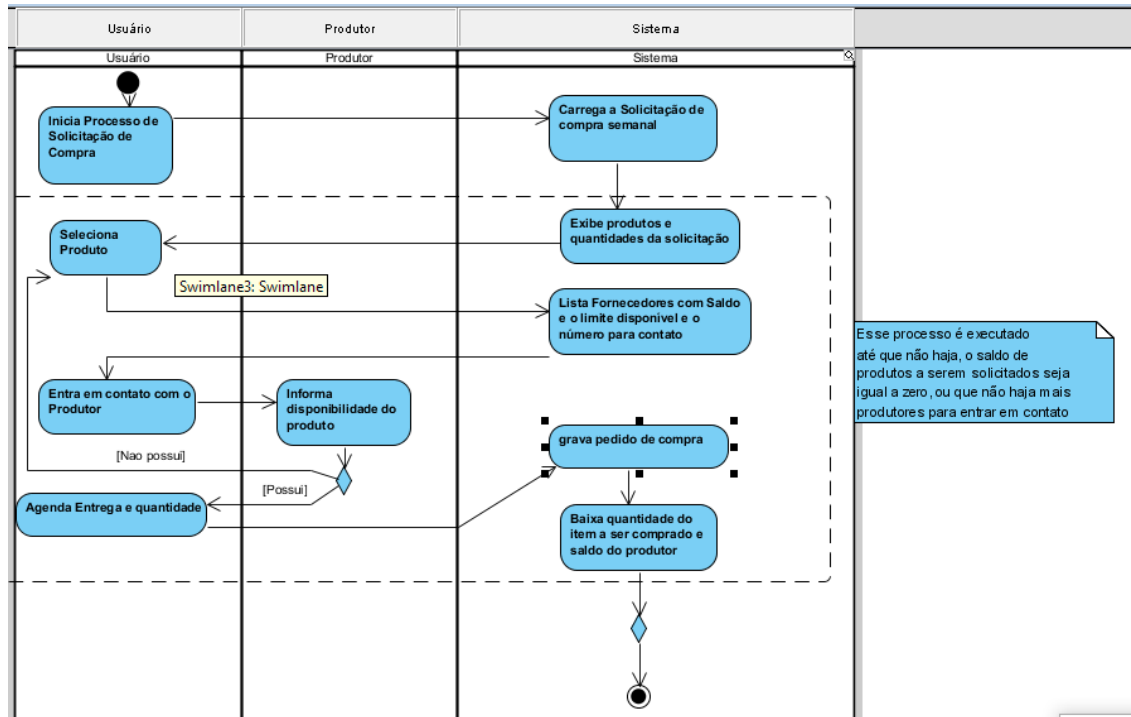


Figura 2 - Diagrama de atividades.

4.2.4 Diagrama de classe

A Figura 3 apresenta o diagrama de classe definido para o sistema e a função de cada classe.

Os Quadros 13 a 24 apresentam detalhes das classes apresentadas na Figura 3. No Quadro 13 é apresentada a descrição da classe funcionário.

<p>Identificação: Usuário</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados de usuários.</p> <p>Atributos: Nome (String): Nome do usuário; Login (String): login do usuário; Senha (String): senha do usuário; Instituição (Instituicao): Instituição do usuário; Tipo (String): Tipo do usuário.</p> <p>Métodos: Cadastrar Alterar Consultar Excluir</p>

Quadro 13 - Descrição da classe usuário

No Quadro 14 é apresentada a descrição da classe instituição.

<p>Identificação: Instituição</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados da instituição..</p> <p>Atributos: Nome (String): Nome da instituição; CNPJ (String): CNPJ da instituição; Endereço (String): Endereço da instituição.</p> <p>Métodos: Cadastrar Alterar Consultar Excluir</p>
--

Quadro 14 - Descrição da classe instituição

No Quadro 15 é apresentada a descrição da classe requisição compra.

<p>Identificação: Requisição de Compra</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados da requisição de compra.</p> <p>Atributos: Data (Date): Data da requisição de compra; Instituição (instituicao): objeto Instituição da requisição de compra; Item (ItensPedido[]): Vetor de Itens da requisição de compra.</p> <p>Métodos: Cadastrar Alterar Consultar Excluir</p>

Quadro 15 - Descrição da classe requisição de compra

No Quadro 16 é apresentada a descrição da classe itens requisição.

<p>Identificação: Itens Requisição</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados dos itens da requisição.</p> <p>Atributos:</p> <ul style="list-style-type: none"> Quantidade (Double): Quantidade de itens na requisição; Requisição (requisição de compra): Requisição de compra do item; Item (Produto): Produto da classe itens requisição. <p>Métodos:</p> <ul style="list-style-type: none"> Cadastrar Alterar Consultar Excluir
--

Quadro 16 - Descrição da classe itens requisição

No Quadro 17 é apresentada a descrição da classe produto.

<p>Identificação: Produto</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados de produto.</p> <p>Atributos:</p> <ul style="list-style-type: none"> Nome (String): Nome do produto; Unidade (String): CNPJ do produto; Preço (double): Endereço do produto. <p>Métodos:</p> <ul style="list-style-type: none"> Cadastrar Alterar Consultar Excluir

Quadro 17 - Descrição da classe produto

No Quadro 18 é apresentada a descrição da classe itens compra.

<p>Identificação: Itens Compra</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados dos itens da compra.</p> <p>Atributos:</p> <ul style="list-style-type: none"> Quantidade (Double): Quantidade de itens na compra; Compra (Compra): objeto compra de itens compra; Valor (Double): valor do item da compra; Item (Produto): Produto da classe de itens compra. <p>Métodos:</p> <ul style="list-style-type: none"> Cadastrar Alterar Consultar Excluir
--

Quadro 18 - Descrição da classe itens compra

No Quadro 19 é apresentada a descrição da classe itens pedido.

<p>Identificação: Itens Pedido</p> <p>Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados dos itens do pedido.</p>
--

Atributos:

Quantidade (Double): Quantidade de itens na compra;
 Pedido (Pedido Compra): objeto compra de itens compra;
 Valor (Double): valor do item da compra;
 Item (Produto): Produto da classe de itens compra.

Métodos:

Cadastrar
 Alterar
 Consultar
 Excluir

Quadro 19 - Descrição da classe itens pedido

No Quadro 20 é apresentada a descrição da classe compra.

Identificação: Compra

Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados da compra.

Atributos:

Produtor (produtor): Produtor da compra;
 Solicitação (SolicitacaoGeral): Solicitação Geral da compra;
 ValorTotal (doble): Valor total da compra.

Métodos:

Consulta limite do produtor
 Cadastrar
 Alterar
 Consultar
 Excluir

Quadro 20 - Descrição da classe compra

No Quadro 21 é apresentada a descrição da classe pedido de compra.

Identificação: Pedido de Compra

Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados de pedido de compra.

Atributos:

Produtor (produtor): Produtor da compra;
 Solicitacao (SolicitacaoGeral): Solicitação Geral da compra;
 DataEntrega(date): Data para entrega dos produtos;
 ValorTotal (doble): Valor total da compra.

Métodos:

Consulta limite do produtor
 Cadastrar
 Alterar
 Consultar
 Excluir

Quadro 21 - Descrição da classe pedido compra

No Quadro 22 é apresentada a descrição da classe produtor.

Identificação: Produtor

Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados do produtor.

Atributos:

Nome (String): Nome do produtor;

<p>Cota (double): cota do produtor; Endereço (String): Endereço do produtor; Telefone (String): Telefone do produtor; CPF (String): CPF do produtor; RG (String): RG do produtor.</p> <p>Métodos: Cadastrar Alterar Consultar Excluir</p>
--

Quadro 22 - Descrição da classe produtor

No Quadro 23 é apresentada a descrição da classe ordem de entrega.

<p>Identificação: Ordem de Entrega Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados de ordem de entrega. Atributos: Instituição (instituicao): Instituição que será realizada da ordem de entrega; Requisição (RequisicaoGeral): Código da requisição geral da ordem de entrega; Produtos (produto[]): Vetor de produtos da ordem de entrega. Métodos: Cadastrar Alterar Consultar Excluir</p>

Quadro 23 - Descrição da classe ordem de entrega

No Quadro 24 é apresentada a descrição da classe requisição geral.

<p>Identificação: Requisição Geral Descrição: Realizar as operações de inclusão, exclusão, consulta e alteração de dados de requisição geral. Atributos: Requisições (RequisicoesCompra[]): Vetor com requisições de compra; Data (Date): Data da movimentação; DataEntrega (date): Data para entrega da requisição geral. Métodos: Cadastrar Alterar Consultar Excluir</p>

Quadro 24 - Descrição da classe requisição geral

4.2.4 Diagrama de entidade e relacionamento

Nesta sessão é apresentado o digrama de Entidade Relacionamento gerado pela ferramenta ERwin o que que possibilitou a geração do script SQL para criação

do banco de dados. O digrama foi dividido em suas imagens para facilitar sua visualização e é apresentado nas Figuras 4 e 5.

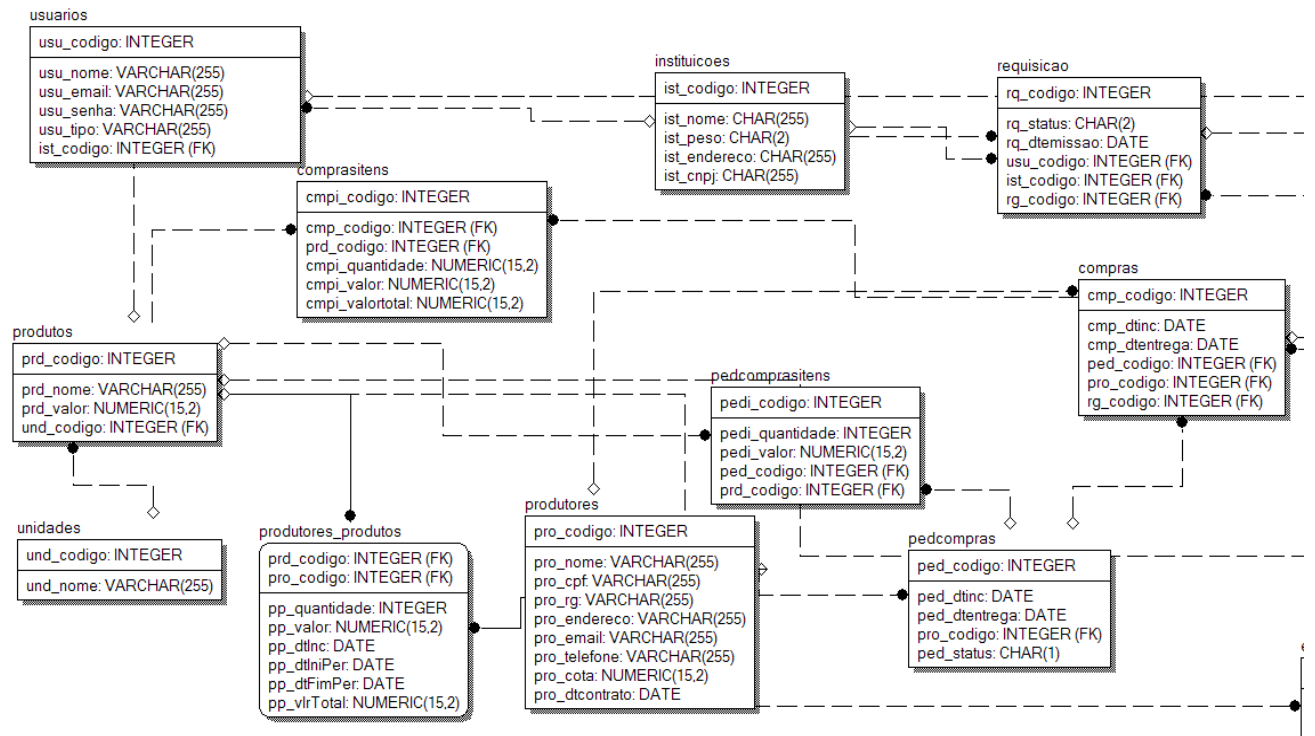


Figura 4 - Diagrama de Entidade Relacionamento (a)

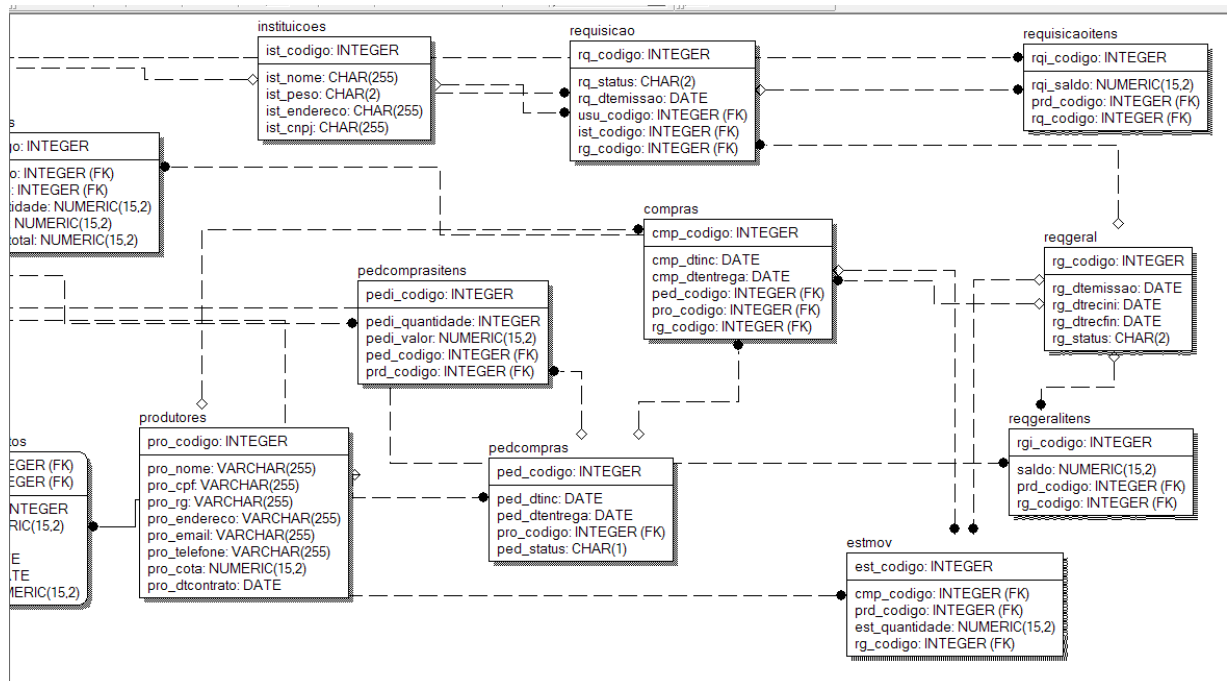
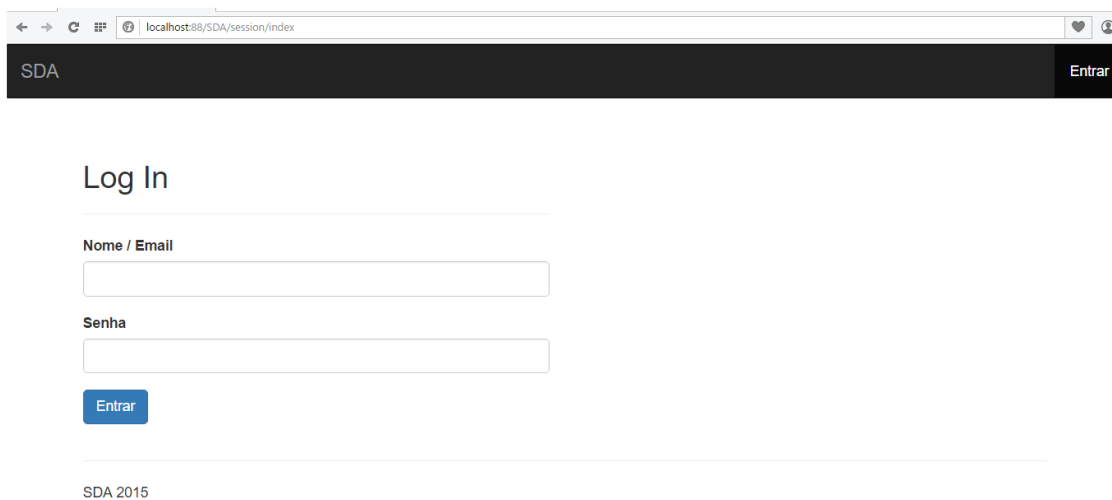


Figura 5 - Diagrama de Entidade Relacionamento (b)

4.3 APRESENTAÇÃO DO SISTEMA

Nesta sessão serão apresentadas as principais telas do sistema para distribuidora trabalho juntamente com uma explicação do funcionamento de cada tela. A Figura 6 apresenta a tela de *login* no sistema.



SDA

Entrar

Log In

Nome / Email

Senha

Entrar

SDA 2015

Figura 6 - Tela de login do usuário

Na Figura 6 é realizado o *login* do usuário, após o usuário realizar o *login* será armazenado em sessão o nome, código, tipo de usuário e a entidade relacionada. Os tipos de usuários estão categorizados em Administrador e Instituição, cujo usuário instituição faz somente a requisição de compra e o administrador tem acesso geral dos processos.

SDA Cadastros Movimentação Relatórios Sair

Produtos **Usuários** Produtores Instituições Unidades

Voltar Salvar

Inserir Usuário

Nome:

Email:

Senha:

Tipo:

Código Instituição:

Salvar

SDA 2015

Figura 7 - Tela de cadastro de usuário

Na Figura 7 consta a tela de cadastro do usuário, nela é possível informar a senha do usuário que será criptografada e identificar o tipo de usuário e vinculá-lo a uma instituição.

SDA Cadastros Movimentação Relatórios Sair

Produtos **Usuários** Produtores Instituições Unidades

Voltar Inserir

Lista de Usuários

Código	Nome	Email	Tipo		
1	Glauco	glauco@rpinfo.com.br	Administrador	Editar	Excluir
2	Larissa	larissa@hotmail.com	Instituição	Editar	Excluir
3	Paulo	paulo@hotmail.com	Instituição	Editar	Excluir

Primeiro Anterior Próximo Último 1/1

SDA 2015

Figura 8 - Tela de lista de usuários

Na Figura 8 o usuário administrador pode visualizar os usuários cadastrados no sistema e o seu tipo de usuário.

Editar Produtor

Código:

Nome:

CPF:

RG:

Cota:

Validade de Contrato:

Endereço:

Email:

Telefone:

Produto	Quantidade	Data Inclusão	Periodo		
Batata doce	2.00	0000-00-00	0000-00-00 até 0000-00-00	2.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
Batata doce	1.00	0000-00-00	0000-00-00 até 0000-00-00	0.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
Batata doce	22.00		0000-00-00 até 0000-00-00	0.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
Tomate	3.00		0000-00-00 até 0000-00-00	0.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
Tomate	3.00		0000-00-00 até 0000-00-00	0.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 9 - Tela de edição de cadastro do produtor

A tela apresentada na Figura 9 é utilizada para fazer a edição do cadastro de produtor, mas principalmente realizar a inclusão de itens para aquele produtor, provisionando os produtos que ele fornecerá durante o ano. Essa lista é utilizada para fazer o pedido de compra.

Pesquisar Requisição

Código:

Código de Usuário:

Status:

SDA Requisição

SDA 2015

Figura 10 - Tela de pesquisa de requisição

Na Figura 10 é possível realizar a pesquisa por parte das instituições das requisições de alimentos para a distribuidora.

Figura 11 - Tela de cadastro de requisição de compra

Na Figura 11 está a tela de cadastro de requisição no qual o usuário pode fazer o pedido dos itens que estão em falta.

Código	Data Geração	Período Inicial	Período Final	Status	Baixar	Editar	Excluir
1	2015-11-14	2015-11-06	2015-11-13	Pendente	Baixar	Editar	Excluir
2	0000-00-00	0000-00-00	0000-00-00	Baixado		Editar	Excluir
6	2015-11-16	0000-00-00	0000-00-00	Pendente	Baixar	Editar	Excluir
7	2015-11-16	0000-00-00	0000-00-00	Pendente	Baixar	Editar	Excluir
8	2015-11-16	0000-00-00	0000-00-00	Pendente	Baixar	Editar	Excluir
9	2015-11-16	0000-00-00	0000-00-00	Pendente	Baixar	Editar	Excluir
10	2015-11-16	2015-10-10	2016-10-10	Pendente	Baixar	Editar	Excluir
11	2015-11-16	2015-10-10	2016-10-10	Pendente	Baixar	Editar	Excluir
12	2015-11-16	2015-10-10	2016-10-10	Pendente	Baixar	Editar	Excluir

Figura 12 - Tela de listagem das requisições semanais

Na Figura 12 é ilustrada a tela na qual são listadas as requisições semanais, agrupando a requisição de todas as unidades em uma requisição unificada.

Cód. Req. Geral	Cód. Produto	Produto	Saldo
1	2	Batata doce	100.00

Primeiro Anterior Próximo Último 1/1

SDA 2015

Figura 13 - Tela de produtos por requisição semanal

Na Figura 13 é possível identificar quais os itens referentes à requisição semanal que tenha sido selecionada na tela da Figura 12.

Cód.	Nome	Cota	Telefone	Nome Produto	Quantidade Prevista	Periodo
1	Joao	2000.00	(46)9979-5346	Batata doce	2.00	0000-00-00 Até 0000-00-00
1	Joao	2000.00	(46)9979-5346	Batata doce	1.00	0000-00-00 Até 0000-00-00
1	Joao	2000.00	(46)9979-5346	Batata doce	22.00	0000-00-00 Até 0000-00-00

Primeiro Anterior Próximo Último 1/1

SDA 2015

Figura 14 - Tela de listagem de produtores por produto

Na Figura 14 podemos listar os produtores de determinado produto selecionado na Figura 13. Nessa tela é exibida a cota do produtor juntamente com o período previsto, nessa, após o administrador entrar em contato com o produtor e verificar se ele possui o item para a entrega, posteriormente será realizado o pedido.

SDA Cadastros Movimentação Relatórios Requisição Sair

Requisição Entidades Gerar Requisição Pedidos Compras Compras Itens

Voltar Salvar

Inserir Pedido

Data de Entrega: 11/13/2015
Código do Produtor:

Itens

Código Produto	Quantidade	Valor	Ações
<input type="text"/>	<input type="text"/>	<input type="text"/>	Remove
Adicionar			

Figura 15 - Tela de pedido de compra

Na Figura 15 observa-se a tela por meio da qual se pode fazer o pedido de compra ao produtor, onde o usuário pode identificar o produto e quantidade desejados, juntamente com a data de entrega do produtor.

SDA Cadastros Movimentação Relatórios Requisição Sair

Requisição Entidades Gerar Requisição Pedidos Compras Compras Itens

Voltar Salvar

Inserir compras

Código do Produtor:
Código Requisição Geral:

Salvar

Itens

Código Produto	Quantidade	Valor	Ações
<input type="text"/>	<input type="text"/>	<input type="text"/>	Remove
Adicionar			

SDA 2015

Figura 16 - Tela de compra

Na Figura 16 está ilustrado como efetivar o pedido levantado na Figura 16, porém não é possível efetuar a compra através do pedido, pois, até o momento, o pedido é somente para controle.



Figura 17 - Tela de requisição semanal

A Figura 17 contém a tela responsável por englobar os pedidos de todas as instituições em uma requisição semanal, o que auxilia no controle e auditoria dos pedidos.

Código	Nome	Unidade	Quantidade
1	Repolho	KG	1350.00
7	Maçã	KG	250.00
8	Abacate	KG	250.00
6	Teste	Caixa	3530.00
2	Batala doce	SACO	3950.00
3	Tomate	SACO	800.00
5	Beterraba	SACO	350.00
5	Beterraba	SACO	350.00

Assinatura

Figura 18 – Relatório de quantia entregue para instituição.

Na Figura 18, consta o relatório que será entregue em cada entidade, em que o responsável por receber a mercadoria na instituição, vai identificar no relatório a quantia entregue e conferir com o que realmente foi entregue. Logo em seguida o responsável vai assinar a lista confirmando a quantia entregue.

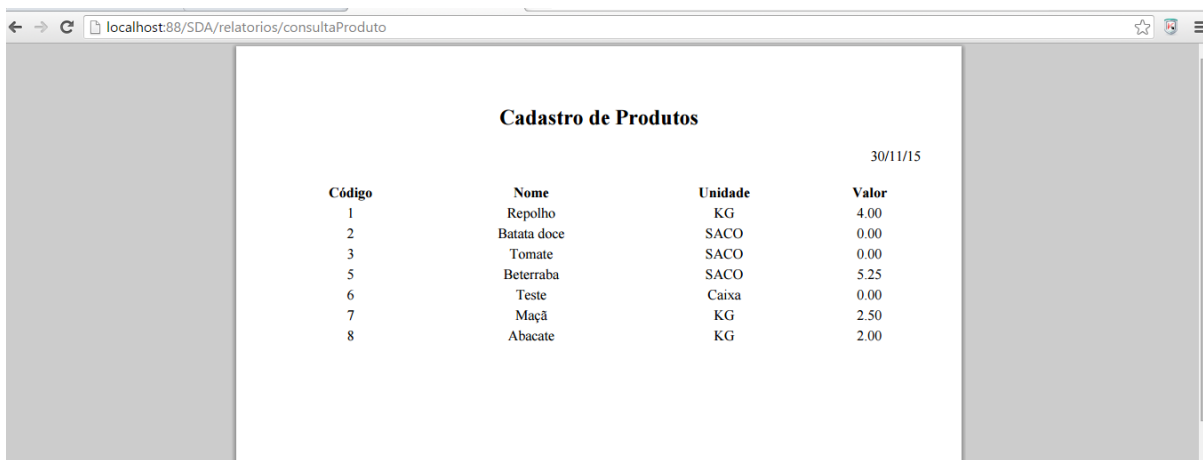


The screenshot shows a web browser window with the address bar displaying 'localhost:88/SDA/relatorios/consultausuario'. The main content area is titled 'Cadastro de Usuários' and includes a date '30/11/15'. Below the title is a table with five columns: 'Código', 'Nome', 'Email', 'Cód.', and 'Nome Instituição'. The table contains four rows of user data.

Código	Nome	Email	Cód.	Nome Instituição
1	Glauco	glauco@rpinfo.com.br	1	Creche
2	Larissa	larissa@hotmail.com	2	Cólegio
3	Paulo	paulo@hotmail.com	1	Creche
4	Leonardo	leonardo@hotmail.com	3	Apae

Figura 19 – Relatório de usuários.

Na figura 19 consta o relatório de usuários do sistema, em que cada usuário possui uma instituição, para que o mesmo possa realizar requisições para essa instituição.



The screenshot shows a web browser window with the address bar displaying 'localhost:88/SDA/relatorios/consultaProduto'. The main content area is titled 'Cadastro de Produtos' and includes a date '30/11/15'. Below the title is a table with four columns: 'Código', 'Nome', 'Unidade', and 'Valor'. The table contains eight rows of product data.

Código	Nome	Unidade	Valor
1	Repolho	KG	4.00
2	Batata doce	SACO	0.00
3	Tomate	SACO	0.00
5	Beterraba	SACO	5.25
6	Teste	Caixa	0.00
7	Maçã	KG	2.50
8	Abacate	KG	2.00

Figura 20 – Relatório cadastro de produtos.

Na figura 20, está o relatório de produtos, juntamente com o preço e a unidade de medida do mesmo. Esse relatório é utilizado para conferência no momento do lançamento de requisição.

4.4 IMPLEMENTAÇÃO DO SISTEMA

Na figura 21, é exibida toda a estrutura do projeto, como a arquitetura MVC que está categorizada em *models*, *views* e *controllers*. O *models* contém toda a regra de negócio definidos no modelo de cada classe, o *view* é responsável por armazenar a interface gráfica do sistema, já o *controllers* é responsável por fazer acesso ao banco de dados, como recuperar, inserir, atualizar e deletar informações do banco.

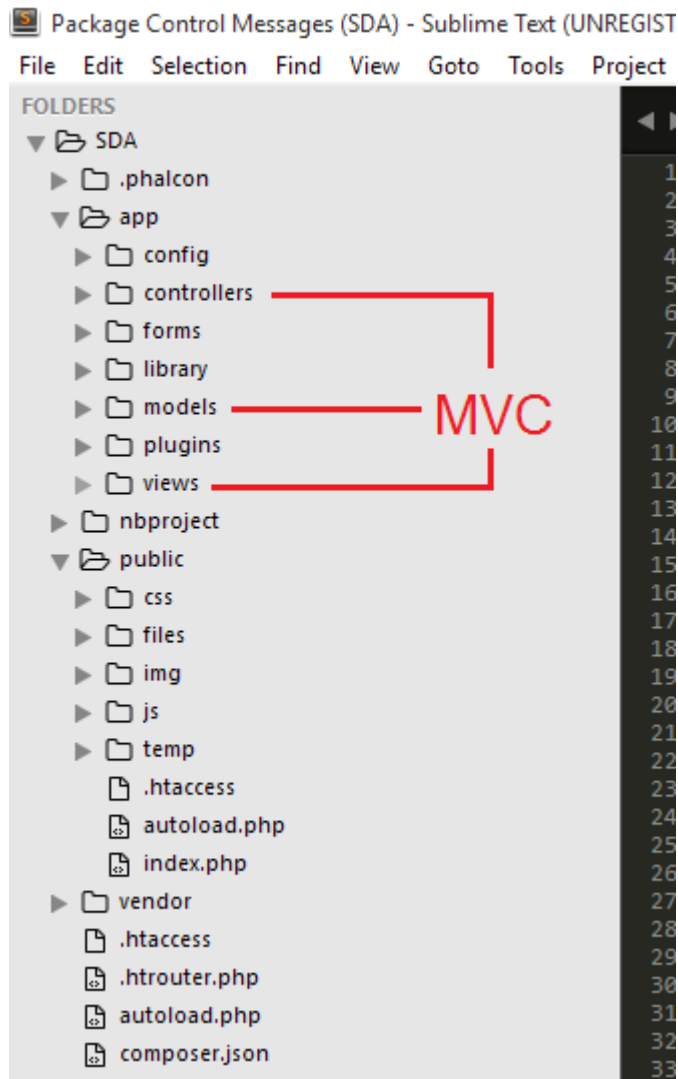


Figura 21 - Pastas e arquivos do projeto.

Nesta sessão são apresentados os principais trechos de códigos explicando sua funcionalidade no sistema elaborado.

```
public function searchAction()
{
    $numberPage = 1;
    if ($this->request->isPost()) {
```

```

$query = Criteria::fromInput($this->di, "Usuarios", $_POST);
$this->persistent->parameters = $query->getParams();
} else {
    $numberPage = $this->request->getQuery("page", "int");
}

$parameters = $this->persistent->parameters;
if (!is_array($parameters)) {
    $parameters = array();
}
$parameters["order"] = "usu_codigo";

$usuarios = Usuarios::find($parameters);
if (count($usuarios) == 0) {
    $this->flash->notice("The search did not find any usuarios");

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "index"
    ));
}

$paginator = new Paginator(array(
    "data" => $usuarios,
    "limit" => 10,
    "page" => $numberPage
));
$this->view->page = $paginator->getPaginate();}

```

Listagem 1 - Procedimento de busca de usuário

A Listagem 1 e 2 apresentam o código que faz parte da controladora de usuários. Sua função é alimentar a listagem de usuários, recuperando os parâmetros para filtro, realizando a pesquisa no banco de dados e posteriormente passando o objeto usuário para a tabela.

```

public function editAction($usu_codigo)
{
    if (!$this->request->isPost()) {

```

```

$usuario = Usuarios::findFirstByusu_codigo($usu_codigo);
if (!$usuario) {
    $this->flash->error("usuario was not found");

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "index"
    ));
}

$this->view->usu_codigo = $usuario->usu_codigo;

$this->tag->setDefault("usu_codigo", $usuario->usu_codigo);
$this->tag->setDefault("usu_nome", $usuario->usu_nome);
$this->tag->setDefault("usu_email", $usuario->usu_email);

$this->tag->setDefault("usu_tipo", $usuario->usu_tipo);
$this->tag->setDefault("ist_codigo", $usuario->ist_codigo);

}
}

```

Listagem 2 - Edição de Usuário

```

public function createAction()
{
    if (!$this->request->isPost()) {
        return $this->dispatcher->forward(array(
            "controller" => "usuarios",
            "action" => "index"
        ));
    }

    $usuario = new Usuarios();

    $usuario->usu_codigo = $this->request->getPost("usu_codigo");
    $usuario->usu_nome = $this->request->getPost("usu_nome");
    $usuario->usu_email = $this->request->getPost("usu_email");
    $usuario->usu_senha = sha1($this->request->getPost("usu_senha"));
    $usuario->usu_tipo = $this->request->getPost("usu_tipo");
}

```

```

$usuario->ist_codigo = $this->request->getPost("ist_codigo");

if (!$usuario->save()) {
    foreach ($usuario->getMessages() as $message) {
        $this->flash->error($message);
    }

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "new"
    ));
}

$this->flash->success("usuario cadastrado");

return $this->dispatcher->forward(array(
    "controller" => "usuarios",
    "action" => "index"
));
}

```

Listagem 3 - Criação do usuário

As Listagens 3 e 4 ilustram a codificação que criar e salva o objeto usuário para cadastro do usuário, ao inserir os dados o *controller* recebe os dados e armazena no objeto usuário com a senha com criptografia sha1.

```

public function saveAction()
{
    if (!$this->request->isPost()) {
        return $this->dispatcher->forward(array(
            "controller" => "usuarios",
            "action" => "index"
        ));
    }

    $usu_codigo = $this->request->getPost("usu_codigo");

    $usuario = Usuarios::findFirstByusu_codigo($usu_codigo);

```

```
if (!$usuario) {
    $this->flash->error("usuario does not exist " . $usu_codigo);

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "index"
    ));
}

$usuario->usu_codigo = $this->request->getPost("usu_codigo");
$usuario->usu_nome = $this->request->getPost("usu_nome");
$usuario->usu_email = $this->request->getPost("usu_email");
$usuario->usu_senha = sha1($this->request->getPost("usu_senha"));
$usuario->usu_tipo = $this->request->getPost("usu_tipo");
$usuario->ist_codigo = $this->request->getPost("ist_codigo");

if (!$usuario->save()) {

    foreach ($usuario->getMessages() as $message) {
        $this->flash->error($message);
    }

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "edit",
        "params" => array($usuario->usu_codigo)
    ));
}

$this->flash->success("usuário atualizado");

return $this->dispatcher->forward(array(
    "controller" => "usuarios",
    "action" => "index"
));
}
```

Listagem 4 - Função save

Na Listagem 5 é utilizada para excluir o usuário do banco de dados.

```
public function deleteAction($usu_codigo)
{

    $usuario = Usuarios::findFirstByusu_codigo($usu_codigo);
    if (!$usuario) {
        $this->flash->error("usuario was not found");

        return $this->dispatcher->forward(array(
            "controller" => "usuarios",
            "action" => "index"
        ));
    }

    if (!$usuario->delete()) {

        foreach ($usuario->getMessages() as $message) {
            $this->flash->error($message);
        }

        return $this->dispatcher->forward(array(
            "controller" => "usuarios",
            "action" => "search"
        ));
    }

    $this->flash->success("usuario was deleted successfully");

    return $this->dispatcher->forward(array(
        "controller" => "usuarios",
        "action" => "index"
    ));
}
```

Listagem 5 - Função para excluir usuário

```
public function localizaAction($rg_codigo)
{

    $numberPage = 1;
```

```

if ($this->request->isPost()) {
    $query = Criteria::fromInput($this->di, "Reqgeralitens", $_POST);
    $this->persistent->parameters = $query->getParams();
} else {
    $numberPage = $this->request->getQuery("page", "int");
}

$parameters = $this->persistent->parameters;
if (!is_array($parameters)) {
    $parameters = array();
}
$parameters["order"] = "rgi_codigo";

$phql = $this->modelsManager->createQuery("SELECT rg.*, rgi.*, p.* FROM reqgeral rg,
reqgeralitens rgi, produtos p
        WHERE rg.rg_codigo = rgi.rgi_codigo and
        p.prd_codigo = rgi.prd_codigo and
        rg.rg_codigo = ".$rg_codigo
        );

$reqgeralitens = $phql->execute();

if (count($reqgeralitens) == 0) {
    $this->flash->notice("Requisição Geral sem produtos");

    return $this->dispatcher->forward(array(
        "controller" => "reqgeralitens",
        "action" => "index"
    ));
}

$paginator = new Paginator(array(
    "data" => $reqgeralitens,
    "limit" => 10,
    "page" => $numberPage
));

$this->view->page = $paginator->getPaginate();
}

```

Listagem 6 - Função utilizada para listar produtos

Na Listagem 6 encontra-se o código que filtra e exibe os produtos de uma requisição semanal em uma tabela.

```

public function listaprodutoresAction($prd_codigo)
{
    $numberPage = 1;
    if ($this->request->isPost()) {
        $query = Criteria::fromInput($this->di, "ProdutoresProdutos", $_POST);
        $this->persistent->parameters = $query->getParams();
    } else {
        $numberPage = $this->request->getQuery("page", "int");
    }

    $parameters = $this->persistent->parameters;
    if (!is_array($parameters)) {
        $parameters = array();
    }
    $parameters["order"] = "pp_codigo";

    $produtores_produtos = ProdutoresProdutos::find($parameters);

    $phql = $this->modelsManager->createQuery("SELECT pr.*, pri.*, p.* FROM produtores
pr, ProdutoresProdutos pri, produtos p
        WHERE pr.pro_codigo = pri.pro_codigo and
        p.prd_codigo = pri.prd_codigo and
        p.prd_codigo = ".$prd_codigo
    );

    $produtores_produtos = $phql->execute();

    if (count($produtores_produtos) == 0) {
        $this->flash->notice("The search did not find any produtores_produtos");
    }

    return $this->dispatcher->forward(array(
        "controller" => "produtores_produtos",
        "action" => "index"
    ));
}

```



```

    $paginator = new Paginator(array(
        "data" => $produtores_produtos,
        "limit"=> 10,
        "page" => $numberPage
    ));

    $this->view->page = $paginator->getPagate();
}

```

Listagem 7 - Listagem de produtor

Na Listagem 7 consta o código que exibe em uma tabela os produtores de um alimento, utilizado pelo botão listar produtores na tela de listagem de produto ilustrada na Figura 13.

```

private function _registerSession(Usuarios $user)
{
    $this->session->set('auth', array(
        'ID' => $user->usu_codigo,
        'NOME' => $user->usu_nome,
        'INS' => $user->ist_codigo,
        'TIPO_USUARIO' => $user->usu_tipo
    ));
}

```

Listagem 8 - Iniciando sessão do usuário

Na Listagem 8 o código é utilizado no momento do *login* para armazenar na sessão o código do usuário, nome, instituição e o tipo de usuário, para posteriormente o sistema montar o menu e definir o acesso. O código da instituição é utilizado no momento da requisição quando o tipo de usuário for instituição.

```

<table>
  <tr>
    <td align="left">
      <?php // seta hora
        date_default_timezone_set('America/Sao_Paulo');
        $date = date('Y-m-d');
        echo $this->tag->hiddenField(array("rq_dtemissao", "type" => "date","value" =>
$date)) ?>
    </td>

```

```

<tr>
  <td align="right">
    <label for="ist_codigo">Código Instituição:</label>
  </td>
  <td align="left">
    <?php
      $auth = $this->session->get('auth');
      echo $this->tag->textField(array("ist_codigo", "type" => "number", "readonly" =>
"readonly", "value"=>$auth['INS'])) ?>

      <?php // Armazena Usuário do pedido

      echo $this->tag->hiddenField(array("usu_codigo", "type" => "date", "value" =>
$auth['ID'])) ?>
    </td>
  </tr>

  <tr>
    <td></td>
    <td><?php echo $this->tag->submitButton("Salvar") ?></td>
  </tr>
</table>

```

Listagem 9 - View requisição

Na Listagem 9 o código identifica a instituição pelo usuário que está na sessão, atribui também a data de emissão à requisição.

```

public function entregaAction(){
  $c =0;
  $phql = $this->modelsManager->createQuery("SELECT e.*, i.*, p.*, un.* FROM
entregue e, instituicoes i, produtos p, unidades un WHERE
      e.ist_codigo = i.ist_codigo and
      p.und_codigo = un.und_codigo and
      e.prd_codigo = p.prd_codigo and
      e.ist_codigo = 1");
  $rows = $phql->execute();
  $html='<head><div align="center" style = "Strong">
      <h2>Relatório de Entrega</h2> </div><div align="right">'.
date("d")."/". date("m")."/". date("y")
      .'</div><head></head>';
  $emp = 0;

```

```

foreach ($rows as $row) {

    if ($temp != $row->i->ist_codigo and $temp == 0){
        $html = $html.'<thead>
<tr>
<th>Código:'. $row->i->ist_codigo.'</th>
<th>Instituição: '. $row->i->ist_nome.'</th>
<tr>
</thead>
        $html = $html.'<table align="center" width = "750">

                <thead><tr>
                    <th>Código</th>
                    <th>Nome</th>
                    <th>Unidade</th>
                    <th>Quantidade</th></tr>
                </thead>
                <tbody>';
    } elseif ($temp != $row->i->ist_codigo and $temp != 0) {
        $html = $html.'</tbody></table></html> <table align =bottom>
<tr><td>-----</td></tr>
        <tr align =center><td>Assinatura</td></tr></table>';
        $html = $html.'<thead>
                <tr>
                    <th>Código:'. $row->i->ist_codigo.'</th>
                    <th>Instituição: '. $row->i->ist_nome.'</th>
                </tr>
                </thead>        ';        $html = $html.'<table
align="center" width = "750">

                <thead>
                <tr>
                    <th>Código</th>
                    <th>Nome</th>
                    <th>Unidade</th>
                    <th>Quantidade</th>
                </tr>
                </thead>
                <tbody>';}

        $temp = $row->i->ist_codigo;
//detalhe

```

```

        $html = $html. '<td align="center">'.$row->e->prd_codigo.'</td>'
                . '<td align="left">'.$row->p->prd_nome.' </td>'
                . '<td align="left">'.$row->un->und_nome.'</td>'
                . '<td align="center">'.$row->e->e_quantidade.' </td>'
</tr>';}

        $html = $html.'</tbody></table></html> <table align =botton>

                <tr><td>-----</td></tr>

                <tr align =center><td>Assinatura</td></tr></table>';

print_r($html);
        $dompdf = new domPdf();
}

```

Listagem 10 - Código relatório de entrega

Na Listagem 10 é apresentado o código para a geração do relatório da figura 18, a estrutura *HTML* foi utilizada para formatar o relatório.

5 CONCLUSÃO

O objetivo do trabalho foi concluído parcialmente, visto que os relatórios de produtores com cota para produção e outros pontos de controle ainda não estão bem adequados para a utilização final do software. Entre as principais ferramentas utilizadas como foco de estudo está o Phalcon Framework, porém o mesmo se mostrou pouco flexível para algumas operações, como por exemplo, a execução de uma *procedure* via banco de dados.

Quanto à modelagem do banco de dados com a ferramenta Erwin se mostrou satisfatória, visto que após concluir a modelagem do banco de dados, o mesmo possibilitou gerar o *script* SQL. Phalcon foi outra ferramenta com papel fundamental para a realização desse projeto, organizando a estrutura do projeto como MVC, auxiliando na segurança do projeto e agilizando o procedimento de cadastros CRUD, porém sua documentação é toda em inglês e sua curva de aprendizagem até o domínio do *framework* é muito alta.

O projeto ficou bem estruturado em relação ao banco de dados e cadastros, porém o mesmo ainda está pendente de relatórios para auxiliar a administração em sua gestão de estoque. Operacionalmente o sistema também pode melhorar bastante a sua usabilidade.

Visto que o sistema ainda precisa de alguns ajustes e melhorias de processo, a perspectiva é continuar melhorando o sistema. Fazendo, assim, com que seja de grande ajuda no processo de distribuição de alimentos. Será gratificante possuir um sistema sendo utilizado em um processo crítico para a PMPB.

REFERÊNCIAS

APACHE. **Apache**. Disponível em: <<https://www.apachefriends.org/about.html>>. Acessado em: 14 nov.2015.

ARAÚJO, M. A. **Modelagem de dados com a Visual Paradigm - do modelo de classes à criação do banco de dados**. Disponível em: <<http://www.devmedia.com.br/artigo-sql-magazine-42-modelagem-de-dados-com-a-visual-paradigm-do-modelo-de-classes-a-criacao-do-banco-de-dados/7019>>. Acessado em: 05 nov.2015.

BASTOS, D. F. **O que é Model-View-Controller (MVC)?** Disponível em: <https://www.oficinadanet.com.br/artigo/desenvolvimento/o_que_e_model-view-controller_mvc>. Acessado em: 12 nov.2015.

BRASIL. Ministério do Desenvolvimento Social e Combate à Fome. **PAA: 10 anos de aquisição de alimentos**. Brasília, DF: MDS; Secretaria Nacional de Segurança Alimentar e Nutricional; Secretaria de Avaliação e Gestão da Informação, 2014. 280 p.

ERWIN. **CA ERwin data modeler**. Disponível em: <<http://erwin.com/worldwide/portuguese-brazil>>. Acessado em: 13 nov.2015.

MYSQL. **Mysql 5.0 reference manual**, Revisão 45396 2015

NASCIMENTO, T. **Desenvolvendo com Bootstrap 3: um framework front-end que vale a pena!** Disponível em: <http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena>>. Acessado em: 10 nov. 2015.

PHALCON. **Phalcon - framework PHP**. Disponível em: <<https://docs.phalconphp.com/en/latest/index.html>>. Acessado em: 09 nov.2015.

PHPMYADMIN. **PhpMyAdmin** Disponível em: <<http://docs.phpmyadmin.net/en/latest/intro.html>>. Acessado em: 10 nov.2015.

PHP NET. **Php Net**. Disponível em: <http://php.net/manual/pt_BR/intro-what-is.php>. Acessado em: 08 nov. 2015.

RABELLO, J. **PHP orientado a objetos: MVC em PHP**. Disponível em: <<http://www.escolacriatividade.com/php-orientado-a-objetos-mvc-em-php>>. Acessado em: 08 nov. 2015.