

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

VALCIR BALBINOTTI JUNIOR

**SISTEMA WEB PARA CONTROLE DE ESTABELECIMENTOS DE COMÉRCIO DE
REFEIÇÕES**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2017**

VALCIR BALBINOTTI JUNIOR

SISTEMA WEB PARA CONTROLE DE ESTABELECIMENTOS DE COMÉRCIO DE REFEIÇÕES

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2017**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento
de Sistemas



TERMO DE APROVAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO
SISTEMA WEB PARA CONTROLE DE ESTABELECIMENTOS DE
COMÉRCIO DE REFEIÇÕES

POR

VALCIR BALBINOTTI JUNIOR

Este trabalho de conclusão de curso foi apresentado no dia 05 de dezembro de 2017, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Profª Drª Beatriz Terezinha Borsoi
Orientadora

Profª MSc. Andreia Scariot Beulke

Prof Esp. João Guilherme Brasil Pichetti

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Profª Drª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

BALBINOTTI JUNIOR, Valcir. Sistema web para controle de estabelecimentos de comércio de refeições. 2017. 64f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Ambientes que servem refeições podem utilizar sistemas e aplicativos computacionais para auxiliar em diversas atividades que são realizadas nesse tipo de estabelecimento, como: controle de pedidos (comandas), contas a pagar e a receber, pagamento de funcionários, controle de estoque e de cardápio. Esses sistemas podem ser bastante abrangentes em termos de funcionalidades e possuir interfaces distintas, como: interface em dispositivo móvel para que garçons possam fazer pedidos; interface *touch screen* que para cozinheiros possam indicar o *status* de pedidos; acesso *web* para gerenciamento e visualização dos dados de negócio para proprietários e gerentes; interface para controle de pagamento; e interface com balança e impressora. Considerando esse contexto de abrangência de funcionalidades para sistemas para estabelecimentos que servem refeições, seja na modalidade *self service* ou *a la carte*, neste trabalho é apresentado o desenvolvimento de um sistema *web* e com acesso por meio de dispositivos móveis para comércio de refeições. PHP foi a linguagem de programação utilizada na implementação do sistema, com as tecnologias Adobe Illustrator, Bootstrap e Cascade Style Sheet para a composição da interface.

Palavras-chave: Sistema web. Aplicativos móveis. PHP. Sistemas para estabelecimentos de comércio de refeições.

ABSTRACT

BALBINOTTI JUNIOR, Valcir. Meal trade establishments control web system. 2017. 64f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Meal-serving environments can use computer systems and applications to assist in various activities that are performed in this type of establishment, such as: order control, payables and receivables, employee payment, inventory and menu control. These systems can be quite comprehensive in terms of functionalities and have different interfaces, such as: mobile interface for the waiter to order, touch screen interface for cooks to indicate order status, web access for data management and visualization business interface, payment control interface and weighing scale interface. Considering this context of range of functionalities that can be performed in establishments that serve meals, either in the self service or a la carte mode, this work presents the development of a web system and with access through mobile devices to trade meals. PHP was the programming language used in the implementation of the system, with the technologies Adobe Illustrator, Bootstrap and Cascade Style Sheet for the composition of the interface.

Keywords: Web system. Mobile applications. PHP. Web system for restaurants.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	25
Figura 2 – Diagrama de classes de análise do sistema	44
Figura 3 – Tela de acesso ao sistema.....	47
Figura 4 – Tela inicial do módulo mensalistas.....	48
Figura 5 – Cadastro mensalistas.....	48
Figura 6 – Cadastro mensalistas: detalhe de <i>tag</i> de mensagem.....	49
Figura 7 – Cadastro mensalista: verificação de campos	49
Figura 8 – Tela para lançamento de pagamento de mensalistas.....	50
Figura 9 – Tela de pagamento de débitos de mensalistas	50
Figura 10 – Estrutura de pastas para organização do projeto.....	51
Figura 11 – Tela inicial de acesso ao sistema.....	54
Figura 12 – Tela de recebimento de clientes: não há valores a receber	61
Figura 13 – Tela de recebimento de clientes: há valores a receber	61

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas	18
Quadro 2 – Requisitos funcionais.....	23
Quadro 3 – Requisitos não funcionais.....	24
Quadro 4 – Caso de uso manter estabelecimento: operação cadastrar	26
Quadro 5 – Caso de uso manter estabelecimento: operação remover	27
Quadro 6 – Caso de uso manter estabelecimento: operação editar	27
Quadro 7 – Caso de uso manter funcionário: operação incluir.....	28
Quadro 8 – Caso de uso manter funcionário: operação remover.....	29
Quadro 9 – Caso de uso manter funcionário: operação editar	29
Quadro 10 – Caso de uso manter produto: operação incluir	30
Quadro 11 – Caso de uso manter produto: operação remover	30
Quadro 12 – Caso de uso manter produto: operação editar	31
Quadro 13 – Caso de uso manter ingrediente: operação incluir	31
Quadro 14 – Caso de uso manter ingrediente: operação excluir	32
Quadro 15 – Caso de uso manter ingrediente: operação editar.....	32
Quadro 16 – Caso de uso registrar entrada de estoque	33
Quadro 17 – Caso de uso corrigir entrada de estoque.....	33
Quadro 18 – Caso de uso manter cliente: operação incluir.....	34
Quadro 19 – Caso de uso manter cliente: operação excluir.....	34
Quadro 20 – Caso de uso manter cliente: operação editar	35
Quadro 21 – Caso de uso manter pagamento de salário: operação registrar pagamento de salário	36
Quadro 22 – Caso de uso manter pagamento de salário: operação corrigir registro de pagamento de salário	36
Quadro 23 – Caso de uso manter conta: operação incluir	37
Quadro 24 – Caso de uso manter conta monetária: operação excluir	37
Quadro 25 – Caso de uso manter conta: operação editar.....	38
Quadro 26 – Caso de uso manter conta: operação registrar entrada/saída de caixa	38
Quadro 27 – Caso de uso manter conta: operação corrigir entrada/saída em conta monetária	39
Quadro 28 – Caso de uso abertura de caixa.....	39
Quadro 29 – Caso de uso iniciar venda	40
Quadro 30 – Caso de uso incluir produto a venda	41
Quadro 31 – Caso de uso remover produto de venda	41
Quadro 32 – Caso de uso fechar venda.....	42
Quadro 33 – Caso de uso manter cardápio	42
Quadro 34 – Caso de uso compor produto	43
Quadro 35 – Caso de uso registrar saída de ingredientes	43
Quadro 36 – Descrição da classe Usuario	44
Quadro 37 – Descrição da classe estabelecimento	45
Quadro 38 – Descrição da classe conta.....	45
Quadro 39 – Descrição da classe lançamento	45
Quadro 40 – Descrição da classe cliente	45
Quadro 41 – Descrição da classe venda.....	46
Quadro 42 – Descrição da classe funcionário	46
Quadro 43 – Descrição da classe produto	46

Quadro 44 – Descrição da classe ingrediente.....	46
Quadro 45 – Descrição da classe econtrole.....	47

LISTAGENS DE CÓDIGO

Listagem 1 – Arquivo de conexão	52
Listagem 2 – Função para buscar as permissões de acesso de usuários	53
Listagem 3 – Conexão com o banco de dados	53
Listagem 4 – Para seleção de vias de clientes em atraso.....	56
Listagem 5 – Definição de elementos de menu.....	60

LISTA DE SIGLAS

CSS	<i>Cascading Style Sheets</i>
FTP	<i>File Transfer Protocol</i>
HTML	HyperText Markup Language
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model-View-Controller</i>
<i>PDO</i>	<i>PHP Data Objects</i>
RIA	<i>Rich Internet Applications</i>
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 CONSIDERAÇÕES INICIAIS	11
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral	12
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO	14
2 REFERENCIAL TEÓRICO	15
2.1 APLICAÇÕES INTERNET RICAS	15
3 MATERIAIS E MÉTODO	18
3.1 MATERIAIS	18
3.2 MÉTODO	19
4 RESULTADO	20
4.1 ESCOPO DO SISTEMA	20
4.2 MODELAGEM DO SISTEMA	22
4.4 APRESENTAÇÃO DO SISTEMA	47
4.5 IMPLEMENTAÇÃO DO SISTEMA	51
5 CONSIDERAÇÕES FINAIS	62
REFERÊNCIAS	63

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa de realização do trabalho. O capítulo é finalizado com a apresentação dos capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

Em 2013, dados indicaram que 32,9% do consumo alimentício da população brasileira foi realizado fora do domicílio familiar (PORTAL BRASIL, 2014). Essas refeições são realizadas em bares, restaurantes, lanchonetes e outras denominações ou categorizações utilizadas para designar ambientes que oferecem serviços de alimentação ou refeições.

Nesse tipo de estabelecimento, a existência de um controle de estoque é fundamental para que não haja desabastecimento de produtos essenciais e, ao mesmo tempo, que produtos não percam o prazo de validade e em decorrência disso tenham que ser descartados ou sejam indevidamente utilizados. Para estabelecimentos que preparam refeições, prazo de validade de produtos é um aspecto crítico do negócio. Controlar esses prazos é essencial para a qualidade do que é servido e para evitar desperdício e a necessidade de descartar alimentos. Os alimentos, especialmente os altamente perecíveis, precisam ser adquiridos de forma que não haja falta para o preparo das refeições, mas também que não sobrem de maneira a vencer o seu prazo de validade.

Outro fator essencial nesse tipo de negócio é o gerenciamento dos pedidos, quando de refeições do tipo *a la carte*, por exemplo. É importante que os clientes recebam os produtos conforme solicitado e o mais rapidamente possível e que o controle de gastos por mesa, por comanda ou por cliente seja correto e efetivo. Comanda é o termo utilizado para a anotação de pedidos realizados pelos clientes em estabelecimentos como bares e restaurantes. Nesse contexto, o valor decorrente da pesagem de um prato em um restaurante *self service* é visto como um pedido, além dos pedidos tradicionais como bebidas, porções e refeições *à la carte*.

Os recursos tecnológicos - sejam os que permitem o desenvolvimento de aplicações *web* mais responsivas, com redução de tráfego de rede e com

possibilidades de interação mais efetivas dos usuários, sejam os utilizados para o desenvolvimento de aplicações para dispositivos móveis - têm possibilitado facilidades para os responsáveis pelo gerenciamento do negócio e para os usuários finais dos sistemas computacionais. Uma aplicação *web* que pode ser acessada por um dispositivo móvel ou pela própria Internet permite que o gerente ou proprietário de um estabelecimento acompanhe à distância e em tempo real dados do seu negócio. Para proprietários, a possibilidade de acesso à distância pode ser bastante útil no caso de estabelecimentos distintos, por exemplo, ou de eles não poderem estar presentes no estabelecimento. Os clientes, por sua vez, podem usar *smartphones* e outros dispositivos móveis para fazer reservas e pedidos. A entrega de comida na forma denominada *delivery* é uma das beneficiárias da possibilidade de realização de pedidos via *web* que pode ocorrer pelo uso de dispositivos móveis. Os dispositivos móveis também podem ser utilizados para realizar reservas e pedidos nos próprios estabelecimentos.

Considerando as possibilidades das tecnologias existentes que propiciam o desenvolvimento de aplicativos *web* e que podem ser implementados com interface responsiva, por meio da realização deste trabalho foi desenvolvido um sistema para controle de estabelecimentos que possuem como serviço ou negócio o oferecimento de refeições. O sistema é *web* responsivo e visa atender as duas maneiras mais utilizadas no comércio de refeições: *self-service (buffet)* e *a la carte*.

1.2 OBJETIVOS

Os objetivos deste trabalho representado pelo desenvolvimento de um sistema visam auxiliar na realização dos processos de trabalho tanto de restaurantes *self-service* quanto *à la carte*. E, ainda, instrumentar proprietários de múltiplos estabelecimentos com um sistema *web* para realizar o gerenciamento dos seus negócios.

1.2.1 Objetivo Geral

Desenvolver um sistema *web* para estabelecimentos de comércio de refeições.

1.2.2 Objetivos Específicos

- Prover um controle do pagamento de refeições realizadas, permitindo pagamentos parciais e mensais, entre outras formas.
- Possibilitar que proprietários tenham acesso aos dados dos seus estabelecimentos em tempo real.
- Agilizar o registro da pesagem e o controle de pagamento em estabelecimentos com refeições tipo *self-service*.
- Facilitar o controle de pedidos realizados por clientes em estabelecimentos do tipo *a la carte*.
- Implementar a interface do sistema atendendo padrões de usabilidade, visando reduzir o número de telas e de operações realizadas pelo usuário e tornar o uso do sistema mais intuitivo.

1.3 JUSTIFICATIVA

O desenvolvimento de uma solução como um sistema computacional para bares e restaurantes não se justifica pela inexistência de aplicativos na área. A Wesco (2016), por exemplo, indica 10 aplicativos somente para dispositivos móveis para o segmento de bares e restaurantes. Muitos outros aplicativos *web*, *mobile* e *desktop* são encontrados para o controle desses estabelecimentos.

A solução proposta não visa ser inovadora, mas possuir funcionalidades relacionadas ao gerenciamento de pedidos, controle de estoque e de notas fiscais, integração de balança com *smartphone* para a geração de comanda, uso de dispositivo móvel pelos garçons, cadastro de cardápio e demais funcionalidades inerentes a esse tipo de serviço e de comércio.

O desenvolvimento da interface seguirá padrões de usabilidade com o objetivo que o aplicativo seja fácil de usar e agilize o trabalho dos profissionais envolvidos na atividade. É importante que os profissionais dessa área possam realizar as suas atividades com rapidez para que os clientes sejam atendidos o mais rapidamente possível. E com segurança para que os pedidos sejam entregues como solicitado e sejam cobrados os valores corretos.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos. Este é o primeiro e apresenta as considerações iniciais com o contexto do sistema que foi desenvolvido, os objetivos e a justificativa. O Capítulo 2 apresenta o referencial teórico sobre aplicações *web*, caracterizadas como ricas. No Capítulo 3 estão as ferramentas e as tecnologias utilizadas na modelagem e na implementação do sistema. No Capítulo 4 é apresentada a modelagem do sistema e parte do seu desenvolvimento. Devido à quantidade de funcionalidades o sistema está em implementação. Por fim estão as considerações finais seguidas pelas referências utilizadas no texto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico trazendo conceitos relacionados às aplicações Internet denominadas *Rich Internet Application*. Esse tipo de aplicação caracteriza o sistema que foi desenvolvido como resultado deste trabalho.

2.1 APLICAÇÕES INTERNET RICAS

A rápida evolução da *web*, tecnologias emergentes como *HyperText Markup Language* (HTML) na versão 5 e os conceitos que caracterizam aplicações *web* como ricas têm fortemente suportado novos tipos de aplicações para a Internet. Diferentemente do desenvolvimento *web* tradicional que considera *tags* HTML e código *script* como os elementos básicos de uma aplicação *web*, as *Rich Internet Applications* (RIA) definem um novo tipo de aplicações para a *web* que integra tecnologias visando alcançar um alto grau de interatividade (AUGUSTIN, 2015). Roubi, Erramdani e Mbarki (2015) ressaltam que nos últimos anos tem sido presenciado um rápido crescimento de aplicações *web* com comportamento de interface com o usuário sofisticado. Essas aplicações, chamadas RIA, trazem a riqueza de interface das aplicações *desktop* para o mundo dos navegadores *web*.

O termo *Rich Internet Application* foi primeiramente apresentado em um *white paper* da Micromedia no ano de 2002 (JEREMY, 2002). O advento de Flash MX e de tecnologias similares ganhou força pelo crescente uso da *web* na sociedade e com isso surgiram as demandas de negócio e usuários finais para uma experiência *web* mais rica e qualitativa (CASTELEYN; GARRIGÓS; MAZÓN, 2014).

RIAs primam por uma interface com o usuário melhorada, com melhor responsividade e mais recursos e funcionalidades de interação. Elas tipicamente dependem fortemente de tecnologias no lado cliente, combinadas com comunicação assíncrona para atender a esses requisitos (CASTELEYN; GARRIGÓS; MAZÓN, 2014). Esses autores fizeram um apanhado de algumas publicações para definir uma RIA, construindo o conceito. As publicações que serviram de base são: Bozzon et al. 2006; Amalfitano et al. 2008; Brambilla et al. 2008; Farrell and Nezelek 2007; Meliá et al. 2010; Preciado et al. 2005; Fraternali et al. 2010; Lawton 2008.

O conceito definido com base nessas referências é que as RIAs são aplicações *web* com o objetivo de prover características e funcionalidades das aplicações *desktop*, oferecendo, assim, uma experiência do usuário mais rica e satisfatória se comparada com as aplicações *web* tradicionais. Essas aplicações podem prover interação alternativa para o uso comum do clique do mouse, como os comandos de teclado similares às aplicações *desktop* que possibilitam uma navegação mais rápida e interativa, desencadeando, eventualmente, mudança dinâmica de conteúdo (FERNANDES et al., 2012).

Em termos de interação, as RIAs devem: visar melhor responsividade, melhorar as funcionalidades de interação e prover uma interface com o usuário rica (CASTELEYN; GARRIGÓS; MAZÓN, 2014).

As RIAs são parametrizadas por três tecnologias ou conceitos fundamentais (CASTELEYN; GARRIGÓS; MAZÓN, 2014):

- a) Comunicação assíncrona entre cliente e servidor: a comunicação assíncrona é a chave para evitar transmissão desnecessária de dados, mantendo o tráfego de rede tão baixo quanto possível e prevenindo a recarga (*refresh*) de partes da página que não sofreram alteração com a operação realizada (BENJAMIN et al., 2010).
- b) Distribuição de dados e operações entre cliente e servidor: operações no lado cliente sincronizadas com o servidor realizadas somente quando necessário reduzem a comunicação de dados e aumentam a responsividade. Técnicas para reduzir a carga de dados trafegados são o aspecto central das RIAs (MESBAH; VAN DEURSEN, 2008).
- c) Bibliotecas ricas para interface gráfica com o usuário no lado cliente: tecnologias no lado cliente como HTML5, JavaScript e *Cascading Style Sheets* (CSS) têm sido empregadas para suportar essas funcionalidades.

Além do uso dessas tecnologias, a interface com o usuário das RIAs deve assegurar que a experiência de navegação é rica o suficiente para ser comparada a uma aplicação *desktop*. Para satisfazer essa necessidade, várias bibliotecas de interface gráfica com o usuário foram desenvolvidas. Elas representam o paradigma guiado por evento (*event-driven*), permitindo aos desenvolvedores reagir adequadamente às mudanças de interface (AMALFITANO et al., 2008). Além disso, a natureza de *event-driven* da implementação das RIAs, combinado com suporte *push* do servidor (em uma única requisição podem ser enviadas diversas respostas),

permite a múltiplos clientes, RIAs colaborativas que reagem a eventos locais e remotos (CARUGHI et al., 2007).

As RIAs, além de apresentar uma melhor usabilidade, permitem melhorar o desempenho da aplicação porque elas reduzem significativamente o número de requisições ao servidor pelo processamento no cliente de operações e a manipulação de dados solicitados pelo usuário, reduzindo a atualização da página e o tráfego na Internet (BERNARDI; DI LUCCA; DISTANTE, 2014).

3 MATERIAIS E MÉTODO

A seguir estão os materiais e o método utilizados para a modelagem e a implementação do sistema.

3.1 MATERIAIS

O Quadro 1 apresenta as tecnologias e as ferramentas utilizadas na modelagem e na implementação do sistema *web* obtido como resultado da realização deste trabalho.

Ferramenta / Tecnologia	Versão	Finalidade
Astah Community	6.21	Modelagem dos casos de uso.
Sublime	3	Ferramenta para desenvolvimento da aplicação.
phpMyAdmin	1.22.1	Administrador de banco de dados.
MySql	8.00.01	Gerenciador de banco de dados.
PHP	7	Linguagem de programação.
CSS	3	Linguagem de estilização para HTML ou <i>eXtensible Markup Language (XML)</i> .
HTML	5	Linguagem de marcação utilizada para construção de páginas web.
Bootstrap	3.3.7	Framework <i>front-end</i> .
IIS	10.1	Interpretador de código PHP que funciona ao lado do Servidor.
Adobe Illustrator	CC	Criação da iconografia.
FileZilla	3.14	Cliente <i>File Transfer Protocol (FTP)</i> para controle de arquivos no servidor.
CPanel		Controle do servidor contratado.

Quadro 1 – Tecnologias e ferramentas utilizadas

3.2 MÉTODO

O levantamento dos requisitos iniciou pelo acompanhamento dos processos em um restaurante *self-service* (Restaurante Estação Gourmet) e um estabelecimento que serve *a la carte* (Blues&Burgers Chopperia). Paralelamente a essa etapa foram realizadas entrevistas informais (no estilo conversa) com os trabalhadores desses locais. Essas atividades permitiram levantar os requisitos do sistema para fundamentar a análise.

Baseando-se na análise das tarefas realizadas nesses ambientes, mas especialmente no restaurante por ser mais complexo e amplo em termos de processo de negócio, foi realizada a modelagem das telas do sistema e a definição do banco de dados. Partindo dessa análise foi e está sendo realizada a implementação e produção gradativa do sistema. À medida que as funcionalidades foram desenvolvidas (para o que já foi implementado), elas eram colocadas em produção e testadas. Isso foi realizado por usuários do restaurante visando avaliar a interface de interação do sistema, atendimento à regra de negócio e identificação de possíveis erros no sistema.

Em ciclos iterativos de desenvolvimento por grupos de funcionalidades, como preconizado pelo processo unificado (SCOTT, 2003), o sistema foi (e está sendo) implementado. Ao final do processo espera-se que ele esteja em efetiva produção nos dois ambientes utilizados como base para o levantamento de requisitos e a implantação de teste. Esse processo está, ainda, ativo, em decorrência das muitas funcionalidades levantadas, como pode ser observado pela modelagem desenvolvida e apresentada na Seção 4.2.

4 RESULTADO

Este capítulo apresenta o resultado deste trabalho que é a modelagem e o desenvolvimento de um sistema *web* para estabelecimentos que realizam comércio de refeições.

4.1 ESCOPO DO SISTEMA

O sistema possui funcionalidades para controlar os dados e os processos gerenciais e funcionais de estabelecimentos como restaurantes, bares, choperias e comércios de alimentos em geral.

As funcionalidades gerenciais visam fornecer recursos para acompanhamento de múltiplos estabelecimentos simultaneamente. Entre essas funcionalidades estão o controle de funcionários, de estoques, de entrada e saída de caixa, movimento de clientes nos estabelecimentos e do recebimento de clientes mensalistas. Esses são os clientes que usam o estabelecimento com frequência - as vezes diária, pelo menos durante a semana - e realizam o pagamento mensalmente.

O cadastro de funcionários permite a hierarquização dos usuários do sistema. Cada função designada pelo administrador terá apenas acesso ao que lhe remete. Para lidar com as necessidades de acompanhamento do sistema são necessários os seguintes níveis:

- Administrador: terá acesso a todas as funcionalidades do sistema e exclusivamente ao cadastro de estabelecimentos;
- Gerente: terá acesso a todas as telas do sistema, exceto as de acesso exclusivo do administrador. E acesso às entradas e saídas do estoque e ao controle de clientes.
- Caixa: fará o controle das vendas, com o recebimento dos pagamentos dos clientes.
- Garçom: fará apenas a saída de estoque que estará relacionada às vendas e ao pedido, quando o atendimento for *a la carte*.
- Cozinheiro: fará apenas a saída de estoque que estará relacionada à produção.

- Cliente: pode iniciar uma venda, quando o processo dele dentro do restaurante inicia com a pesagem do prato que ao ser colocado sobre a balança gera uma comanda. Essa comanda passa a ser, também, a identificação do cliente e pode ser anexada a uma comanda, quando o cliente posteriormente consumirá outros produtos, como bebidas. O cliente não solicita o fechamento da conta que é realizado pelo caixa e solicita a inclusão ou exclusão de produtos na comanda. Essa inclusão ou exclusão é realizada pelo caixa ou pelo garçom.

Partindo dos atores e suas permissões de acesso ao sistema, as principais funcionalidades do sistema são:

- Cadastro de estabelecimentos: ao acessar o sistema, a primeira tela a ser apresentada é a de controle. Nessa tela estará destacado se os estabelecimentos estão abertos ou fechados, quantas mesas do salão estão ocupadas, o número total de comandas abertas e fechadas e quanto dinheiro está em percurso pelo caixa. Menus levarão para o cadastro, a edição e a exclusão de estabelecimentos.

- Cadastro de funcionários: cadastro para quem usará o sistema. Nesse cadastro destaca-se o campo salário, que influenciará nos relatórios mensais do financeiro.

- Controle de estoque: será realizado a partir de cadastro de ingredientes que compõem os produtos. Muitos ingredientes utilizados em receitas, como porções de carne, por exemplo, são de controle de estoque difícil. Em geral, eles são comprados por quilo ou unidade e vendidos fracionados. O fracionamento nem sempre pode ser realizado de maneira que as partes componham o todo. É difícil estimar que 'x' filés compõem um quilo de *mignon*. Então, o gerente fará o controle das entradas no estoque diariamente, mensalmente ou como definido pelo restaurante. E a saída será realizada pelo cozinheiro-chefe sempre que necessário. Ainda assim, uma verificação de estoque periódica é recomendada para que os devidos ajustes sejam realizados no sistema.

- Controle de clientes: cadastro simples de clientes. Auxiliará no momento de gerar notas fiscais nominais e acompanhar a conta (débitos e pagamento) dos mensalistas.

- Movimento de caixa: uma tela que apresenta dados atuais do salão. Salão é o ambiente que possui as mesas cadastradas no sistema. Cada cliente que chegar ao local terá sua própria comanda, sendo assim, no momento que uma venda é realizada e registrada na respectiva comanda ela poderá ser associada a uma mesa.

Quando um cliente deixar o estabelecimento, o caixa fará o fechamento da comanda. Será necessário registrar se o pagamento foi realizado em dinheiro, crédito, débito ou se será registrado, indicando que o cliente ficará com valor pendente.

Haverá a opção de entrada e saída de dinheiro do caixa sem que seja por venda, mas será necessário justificar por meio de observação, o motivo desse crédito ou débito. Uma entrada pode ocorrer quando o cliente quebrar um copo, por exemplo, e fará a respectiva reposição do valor. Uma saída pode referir-se a um desconto oferecido a um cliente ou decisão gerencial de não cobrança do valor devido.

- Vendas do salão: tela com os produtos listados e enumerados para os garçons registrarem saídas de produtos. As saídas poderão ser relacionadas às comandas, ao balcão ou às justificativas por meio de observação (exemplo: produto vencido, ajuste de estoque).

Além disso, o sistema contará com *hardwares* externos que auxiliarão em seu funcionamento. Tais como:

- Balança eletrônica: iniciará a venda de produto por peso, sempre relacionada a uma comanda;

- Impressora vinculada à balança eletrônica: entregará a comanda ao cliente em casos específicos, que ocorrem quando o pedido do cliente inicia com a pesagem do prato na balança.

- Impressora de caixa: impressão de relatórios, recibos e outros documentos vinculados ao sistema.

- Impressora da copa: para impressão dos pedidos de bebida. A venda de bebida está vinculada a uma comanda ou a uma mesa e é tratada como comanda.

- Leitor de código de barras: buscará o número da comanda impressa.

4.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a listagem dos Requisitos Funcionais (RF) definidos para o sistema.

Identificação	Nome	Descrição
RF01	Manter estabelecimentos	O Administrador fará o cadastro, a exclusão e a edição dos estabelecimentos.
RF02	Manter funcionários	O Administrador fará o cadastro, a exclusão e a edição dos funcionários de qualquer estabelecimento. O Gerente fará essas operações apenas no estabelecimento que ele está cadastrado.
RF03	Manter produtos	Gerente fará o cadastrado, a exclusão e a edição de produtos vendidos no estabelecimento.
RF04	Manter ingredientes	Gerente fará o cadastro, a edição e a exclusão dos ingredientes usados no estabelecimento.
RF05	Registrar entrada estoque	Gerente registrará entrada de produtos no estoque, lançando o valor gasto na compra do produto.
RF06	Manter clientes	O Gerente e o Caixa farão o controle de cadastro, edição e exclusão de clientes.
RF07	Registrar pagamento salário	Gerente registrará pagamento de salários pré-agendados pelo sistema.
RF08	Manter contas monetárias	Gerente fará cadastro, exclusão e edição de contas para gerenciamento monetário. Aqui ficarão associadas as entradas e as saídas de valores.
RF09	Registrar entrada/saída de valor	Gerente e Caixa farão registro de entradas e saídas de caixa atípicos, sem haver comanda associada.
RF10	Abrir caixa	Caixa iniciará período de entrada/saída de dinheiro que serão registradas como vendas.
RF11	Iniciar venda	Caixa e Garçom iniciarão uma venda associando um produto a um número de mesa ou id de venda. O id de venda será entregue ao cliente como uma comanda. A balança eletrônica também iniciará venda associando-a um id de venda.
RF12	Associar produto a id de venda	Caixa e Garçom farão associação de produtos presentes em estoque a um id de vendas ou mesa com venda aberta.
RF13	Remover produtos a id de venda	Caixa e Garçom removerão produtos de id de vendas.
RF14	Fechar venda	Caixa fará fechamento de venda.
RF15	Emitir NF-e	Caixa fará emissão de NF-e baseada na venda.
RF16	Fechar caixa	Caixa fará fechamento de período de entrada/saída de dinheiro previamente aberto.
RF17	Compor produtos	Cozinheiro fará composição e respectiva inclusão de produtos baseando-se em ingredientes.
RF18	Manter cardápios	Cozinheiro fará a inclusão, a edição e a exclusão de cardápios que são compostos por produtos.
RF19	Registrar saída de ingredientes	Cozinheiro fará o registro de saída de ingredientes do estoque.
RF20	Emitir relatórios administrativos	Administrador e Gerente farão emissão de relatórios administrativos.
RF21	Emitir relatórios monetários	Administrador e Gerente farão emissão de relatórios referentes às entradas/saídas de dinheiro.
RF22	Emitir relatórios de vendas	Administrador, Gerente e Caixa emitirão relatórios referentes às vendas por estabelecimento.

Quadro 2 – Requisitos funcionais

A listagem do Quadro 3 apresenta os Requisitos Não-Funcionais (RNF) identificados para o sistema.

Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema será realizado por meio de <i>login</i> e senha.
RNF02	Sistema responsivo	Permite acesso por computadores e dispositivos distintos em termos de tamanho de tela.
RNF03	Atribuição de permissões	O sistema contará com diversos perfis de acesso.
RNF04	Cadastro de funcionários	O administrador poderá registrar funcionários em qualquer estabelecimento cadastrado. Já o gerente poderá cadastrar funcionários somente em seu estabelecimento de origem.

Quadro 3 – Requisitos não funcionais

O diagrama de casos de uso apresentado na Figura 1 contém as funcionalidades essenciais do sistema realizadas pelos seus atores que são: administrador, gerente, balança, caixa, garçom, cozinheiro e cliente.

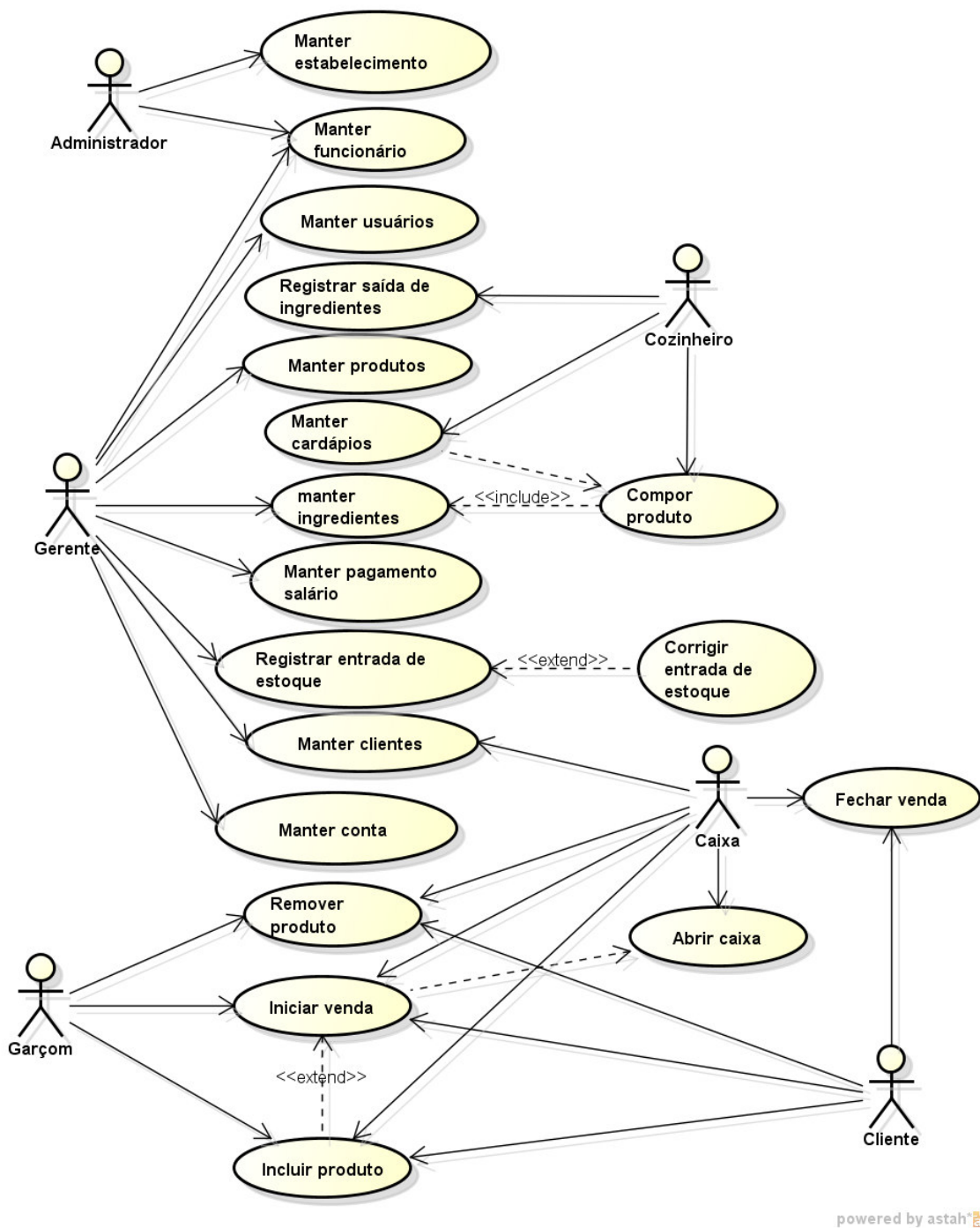


Figura 1 – Diagrama de casos de uso

O administrador é responsável pelos cadastros de usuários do sistema e de estabelecimentos. O garçom é responsável por iniciar vendas e lançar produtos nas comandas. O cozinheiro é responsável por registrar as saídas de estoque para cozinha e compor cardápios e produtos. O caixa é responsável por finalizar a venda

das comandas. A balança pode iniciar a venda, quando o cliente inicia o procedimento pela pesagem do prato, mas nesse caso o ator é o cliente porque é ele quem faz a ação de colocar o prato sobre a balança.

A balança lê o valor do peso que define uma nova venda, em seguida a impressora conectada à balança imprime o número que identifica a venda, o respectivo valor, data e hora. Essa impressão é utilizada para identificar a venda do cliente. Posteriormente, os garçons podem lançar mais produtos por esse número impresso a partir da balança que identifica o cliente. E por fim o caixa fecha a venda também através desse identificador.

A seguir são expandidos os casos de uso apresentados na Figura 2, representando as funcionalidades essenciais do sistema.

No Quadro 4 está a descrição da operação cadastrar (incluir) do caso de uso manter estabelecimento.

<p>Caso de uso: Manter estabelecimento: operação cadastrar.</p> <p>Descrição: Quando necessário, o administrador cadastrará um novo estabelecimento.</p> <p>Atores: Administrador.</p> <p>Pré-condição: Estar conectado ao sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Administrador acessa a tela para cadastrar estabelecimento. 2. O sistema apresenta formulário de cadastro. 3. Administrador preenche e envia formulário. 4. Sistema apresenta dados cadastrados e pede confirmação. 5. Administrador confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Dados do novo estabelecimento inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 3: Administrador informa campos inválidos ou não informa campos obrigatórios.	4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou de preenchimento obrigatório não informados. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.

Quadro 4 – Caso de uso manter estabelecimento: operação cadastrar

No Quadro 5 está a descrição da operação remover do caso de uso manter estabelecimento.

<p>Caso de uso: Caso de uso manter estabelecimento: operação remover</p> <p>Descrição: Quando necessário, o administrador removerá um estabelecimento.</p> <p>Atores: Administrador.</p> <p>Pré-condição: Estar conectado ao sistema. Possuir estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Administrador acessa a tela de listagem de estabelecimentos e seleciona o estabelecimento que deseja excluir. 2. Administrador pressiona o botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros relacionados ao referido estabelecimento. 4. Administrador confirma exclusão. 5. Sistema faz exclusão dos dados no banco. 6. Sistema informa que os dados foram excluídos. <p>Pós-Condição: Dados estabelecimento removidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Administrador não confirma exclusão.	4.1 Sistema retorna a tela de listagem de estabelecimentos, sem realizar a operação de exclusão.

Quadro 5 – Caso de uso manter estabelecimento: operação remover

No Quadro 6 está a descrição da operação editar do caso de uso manter estabelecimento.

<p>Caso de uso: Caso de uso manter estabelecimento: operação editar.</p> <p>Descrição: Quando necessário, o administrador editará um estabelecimento.</p> <p>Atores: Administrador.</p> <p>Pré-condição: Estar conectado ao sistema. Possuir estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Administrador acessa a tela de listagem de estabelecimentos e seleciona o estabelecimento que pretende editar dados de cadastro. 2. Administrador pressiona o botão editar. 3. Sistema apresenta tela com campos de cadastro preenchidos com os dados do estabelecimento atual . 4. Administrador edita os campos desejados e envia o formulário para que seja persistido no banco de dados. 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição: Dados estabelecimento editados são persistidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Administrador informa campos incorretamente ou não informa campos de preenchimento obrigatório.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 6 – Caso de uso manter estabelecimento: operação editar

No Quadro 7 está a descrição da operação incluir (cadastrar) do caso de uso manter funcionário.

<p>Caso de uso: Caso de uso manter funcionário: operação incluir.</p> <p>Descrição: Quando necessário, o administrador ou o gerente cadastrará um novo funcionário. Atentando que o administrador pode registrar funcionários em qualquer estabelecimento cadastrado. Já o gerente poderá cadastrar funcionários somente em seu estabelecimento de origem.</p> <p>Atores: Administrador ou gerente.</p> <p>Pré-condição: Estar conectado ao sistema e possuir um estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para cadastrar funcionário. 2. O sistema apresenta formulário de cadastro. 3. Ator preenche e envia formulário. 4. Sistema apresenta dados cadastrados e pede confirmação. 5. O ator confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Dados do novo funcionário inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Ator informa campos inválidos ou não informa dados para campos obrigatórios.	4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou campos obrigatórios não preenchidos. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.

Quadro 7 – Caso de uso manter funcionário: operação incluir

No Quadro 8 está a descrição da operação remover do caso de uso remover funcionário.

<p>Caso de uso: Caso de uso manter funcionário: operação remover.</p> <p>Descrição: Quando necessário o administrador ou gerente removerá um funcionário.</p> <p>Atores: Administrador ou gerente.</p> <p>Pré-condição: Estar conectado ao sistema e possuir funcionário cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de listagem de funcionários e seleciona o funcionário que deseja excluir. 2. Ator pressiona no botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros ligados aquele funcionário. 4. Ator confirma exclusão. 5. Sistema faz exclusão dos dados no banco. 6. Sistema informa que os dados foram excluídos.

Pós-Condição: Dados do funcionário removidos no banco de dados.	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Administrador não confirma exclusão.	4.1 Sistema retorna a tela de listagem de funcionários.

Quadro 8 – Caso de uso manter funcionário: operação remover

No Quadro 9 está a descrição da operação editar do caso de uso manter funcionário.

<p>Caso de uso: Caso de uso manter funcionário: operação editar funcionário.</p> <p>Descrição: Quando necessário, o administrador ou o gerente editará dados cadastrais de um funcionário.</p> <p>Atores: Administrador ou gerente.</p> <p>Pré-condição: Estar conectado ao sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de listagem de funcionários e seleciona um registro. 2. Ator pressiona no botão editar. 3. Sistema apresenta tela com campos de cadastro preenchidos com os dados do funcionário atual. 4. Ator edita os campos desejados e envia o formulário. 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição: Dados do funcionário editados no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Ator informa campos incorretamente ou não informa campos de preenchimento obrigatório.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente u que há campos obrigatórios sem preenchimento.

Quadro 9 – Caso de uso manter funcionário: operação editar

No Quadro 10 está a descrição da operação incluir do caso de uso manter produto.

<p>Caso de uso: Manter produto: operação incluir</p> <p>Descrição: Quando necessário, o gerente cadastrará um novo produto.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema e possuir um estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O gerente acessa a tela para cadastrar produto. 2. O sistema apresenta formulário de cadastro. 	
---	--

<p>3. O gerente preenche e envia o formulário. 4. Sistema apresenta os dados cadastrados e pede confirmação. 5. O gerente confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados.</p> <p>Pós-Condição: Dados do novo produto inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
<p>Linha 4: Gerente informa campos inválidos ou não informa campos de preenchimento obrigatório.</p>	<p>4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou campos obrigatórios não preenchidos. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.</p>

Quadro 10 – Caso de uso manter produto: operação incluir

No Quadro 11 está a descrição da operação remover do caso de uso manter produto.

<p>Caso de uso: Manter produto: operação remover.</p> <p>Descrição: Quando necessário, o gerente removerá um produto.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema e possuir produto cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de produtos e seleciona o registro a ser removido. 2. Gerente pressiona no botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros ligados aquele produto. 4. Gerente confirma exclusão. 5. Sistema faz exclusão dos dados no banco. 6. Sistema informa que os dados foram excluídos. <p>Pós-Condição: Dados produto removidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
<p>Linha 4: Gerente não confirma exclusão.</p>	<p>4.1 Sistema retorna a tela de listagem de produtos.</p>

Quadro 11 – Caso de uso manter produto: operação remover

No Quadro 12 está a descrição da operação de edição do caso de uso manter produto.

<p>Caso de uso: Manter produto: operação editar.</p> <p>Descrição: Quando necessário, o gerente editará um produto.</p> <p>Atores: Gerente.</p>	
---	--

<p>Pré-condição: Estar conectado ao sistema e possuir produto cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de produtos e escolhe o registro a ser editado. 2. Gerente pressiona no botão editar. 3. Sistema mostra tela com campos de cadastro preenchidos com os dados do produto atual. 4. Gerente edita os campos desejados e envia o formulário. 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição: Dados produto editados no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Gerente informa campos incorretamente o não informa dados de campos obrigatórios.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou há campos obrigatórios não preenchidos.

Quadro 12 – Caso de uso manter produto: operação editar

No Quadro 13 está a descrição da operação incluir do caso de uso manter ingrediente.

<p>Caso de uso: Manter ingrediente: operação incluir.</p> <p>Descrição: Quando necessário, o gerente cadastrará um novo ingrediente.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema e possuir um estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela para cadastrar ingrediente. 2. O sistema apresenta formulário de cadastro. 3. Gerente preenche e envia o formulário. 4. Sistema apresenta dados cadastrados e pede confirmação. 5. O gerente confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Dados do novo ingrediente inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente informa campos inválidos ou não informa campos de preenchimento obrigatório.	4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou campos obrigatórios não preenchidos. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.

Quadro 13 – Caso de uso manter ingrediente: operação incluir

No Quadro 14 está a descrição da operação de exclusão do caso de uso manter ingrediente.

Caso de uso:

<p>Manter ingrediente: operação excluir.</p> <p>Descrição: Quando necessário o gerente removerá um ingrediente cadastrado.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema e haver ingrediente cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de ingredientes e seleciona o ingrediente que deseja excluir. 2. Gerente pressiona no botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros ligados aquele ingrediente. 4. Gerente confirma exclusão. 5. Sistema faz exclusão dos dados no banco e informa que dados foram excluídos. <p>Pós-Condição: Dados ingrediente removidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente não confirma exclusão.	4.1 Sistema retorna para a tela de listagem de ingredientes.

Quadro 14 – Caso de uso manter ingrediente: operação excluir

No Quadro 15 está a descrição da operação editar do caso de uso manter ingrediente.

<p>Caso de uso: Manter ingrediente: operação editar.</p> <p>Descrição: Quando necessário, gerente editará um ingrediente cadastrado.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema e haver ingredientes cadastrados.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de ingredientes e seleciona o registro a ser editado. 2. Gerente pressiona no botão editar. 3. Sistema apresenta tela com campos de cadastro preenchidos com os dados do ingrediente atual. 4. Gerente edita os campos desejados e envia o formulário. 5. Sistema registra alterações no banco de dados e informa da edição. <p>Pós-Condição: Dados ingrediente editados no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Gerente informa campos incorretamente ou não informa dados para campos de preenchimento obrigatório,	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou há campos obrigatórios não preenchidos.

Quadro 15 – Caso de uso manter ingrediente: operação editar

O caso de uso registrar entrada de estoque é apresentado no Quadro 16.

<p>Caso de uso: Registrar entrada de estoque.</p>
--

<p>Descrição: Quando produtos ou ingredientes forem entregues no estabelecimento, o gerente fará o registro da entrada deles no sistema.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Possuir ingredientes ou produtos para serem cadastrados no sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela para registrar entrada de estoque. 2. O sistema apresenta formulário de entrada. 3. Gerente inclui os itens e suas quantidades e envia o formulário. 4. Sistema apresenta dados inseridos e solicita conferência. 5. O gerente confirma a entrada. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Quantidade de produtos ou ingredientes em estoque atualizada no banco de dados</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente percebe que informou quantidades ou itens de forma equivocada.	4.1 Ator clica em corrigir. 4.2 Retorna para o formulário de entrada em estado de edição – passo 3.

Quadro 16 – Caso de uso registrar entrada de estoque

O caso de uso corrigir entrada de estoque é apresentado no Quadro 17.

<p>Caso de uso: Corrigir entrada de estoque.</p> <p>Descrição: Quando o gerente percebe que errou uma entrada de estoque ele pode corrigir. A diferença entre corrigir e editar é que as correções ficarão registradas no banco.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Possuir entrada de estoque registrada e com necessidade de correção.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O gerente acessa a tela para corrigir entrada de estoque. 2. O sistema apresenta formulário de correção. 3. O gerente ajusta os itens e/ou quantidades e envia o formulário. 4. Sistema apresenta dados inseridos e solicita conferência. 5. O gerente confirma a entrada. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Quantidade dos produtos em estoque atualizada no banco de dados. Dados da correção incluídos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente percebe que informou quantidades ou itens de forma equivocada.	4.1 Ator clica em corrigir. 4.2 Retorna para o formulário de entrada em estado de edição – passo 3.

Quadro 17 – Caso de uso corrigir entrada de estoque

No Quadro 18 está a descrição da operação incluir do caso de uso manter cliente.

Caso de uso:

<p>Manter cliente: operação incluir.</p> <p>Descrição: Quando necessário, o gerente ou caixa cadastrará um novo cliente.</p> <p>Atores: Gerente ou caixa.</p> <p>Pré-condição: Estar conectado ao sistema e possuir um estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para cadastrar cliente. 2. O sistema apresenta formulário de cadastro. 3. Ator preenche e envia formulário. 4. Sistema apresenta dados cadastrados e pede confirmação. 5. O ator confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Dados do novo cliente inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Ator informa campos inválidos ou não preenche campos obrigatórios.	4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou dados obrigatórios não preenchidos. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.

Quadro 18 – Caso de uso manter cliente: operação incluir

No Quadro 19 está a descrição da operação excluir do caso de uso manter cliente.

<p>Caso de uso: Caso de uso manter cliente: operação excluir.</p> <p>Descrição: Quando necessário, o gerente removerá um cliente.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema. Possuir cliente cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de clientes. 2. Gerente aperta no botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros ligados aquele ingrediente. 4. Gerente confirma exclusão. 5. Sistema faz exclusão dos dados no banco. 6. Sistema informa que os dados foram excluídos. <p>Pós-Condição: Dados cliente removidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente não confirma exclusão.	4.1 Sistema retorna a tela de listagem de clientes.

Quadro 19 – Caso de uso manter cliente: operação excluir

No Quadro 20 está a descrição da operação editar do caso de uso manter cliente.

<p>Caso de uso: Manter cliente: operação editar</p> <p>Descrição: Quando necessário, o gerente editará um cliente.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de clientes e seleciona o registro a ser editado. 2. Gerente pressiona no botão editar. 3. Sistema mostra tela com campos de cadastro preenchidos com os dados do cliente atual . 4. Gerente edita os campos desejados e envia o formulário. 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição: Dados cliente editados no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Ator informa campos incorretamente ou não preenche campos obrigatórios.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios faltando.

Quadro 20 – Caso de uso manter cliente: operação editar

No Quadro 21 está a descrição da operação registrar pagamento de salário do caso de uso manter pagamento de salário.

<p>Caso de uso: Manter pagamento de salário: operação registrar pagamento de salário</p> <p>Descrição: O gerente confirma o pagamento dos salários previamente cadastrados.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Funcionários cadastrados</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de funcionários. 2. Gerente pressiona no botão pagar salários. 3. Sistema mostra tela tabela informando os funcionários e os salários. O valor a ser pago a cada salário é informado em um <i>input</i> editável. 4. Gerente edita os campos desejados e envia o formulário. 5. Sistema apresenta os dados informados e pede confirmação 6. Gerente pressiona no botão confirmar; 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição: Valores adicionados ao banco de dados.</p>	
--	--

Nome do fluxo alternativo (extensão)	Descrição
Linha 6. Gerente identifica dados incorretos e decide não confirmar.	6.1 Sistema retorna ao formulário com os campos como previamente preenchidos.

Quadro 21 – Caso de uso manter pagamento de salário: operação registrar pagamento de salário

A operação corrigir registro de pagamento de salário do caso de uso manter pagamento de salário é apresentado no Quadro 22.

<p>Caso de uso: Manter pagamento de salário: operação corrigir pagamento de salário.</p> <p>Descrição: Gerente corrige pagamento de salário</p> <p>Atores: Gerente.</p> <p>Pré-condição: Possuir pagamento de salário registrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de pagamentos. 2. Gerente pressiona botão corrigir 3. O sistema apresenta formulário de correção. 4. Gerente corrige os valores desejados. 5. Sistema apresenta dados inseridos e pede atenciosamente uma conferência. 6. O gerente confirma a entrada. 7. Sistema informa que os dados foram incluídos no banco de dados. <p>Pós-Condição: Valores atualizadas no banco de dados. Dados da correção incluídos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6: Ator percebe que informou valores de forma equivocada.	6.1 Ator clica em corrigir. 6.2 retorna para o formulário de entrada em estado de edição – passo 3.

Quadro 22 – Caso de uso manter pagamento de salário: operação corrigir registro de pagamento de salário

No Quadro 23 está a descrição da operação cadastrar do caso de uso manter conta monetária.

<p>Caso de uso: Manter conta monetária: operação incluir.</p> <p>Descrição: Cadastro de contas para controle financeiro.</p> <p>Atores: Gerente ou caixa.</p> <p>Pré-condição: Estar conectado ao sistema e possuir um estabelecimento cadastrado.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de contas monetárias. 2. O sistema apresenta formulário de cadastro. 3. Ator preenche e envia formulário. 	
--	--

<p>4. Sistema apresenta dados cadastrados e pede confirmação. 5. O ator confirma o cadastro. 6. Sistema informa que os dados foram incluídos no banco de dados.</p> <p>Pós-Condição: Dados do nova conta inseridos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Ator informa campos inválidos ou não inclui dados em campos de preenchimento obrigatório.	4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. É emitida mensagem informando o usuário. 4.2 Retorna para o formulário de cadastro em estado de edição – passo 3.

Quadro 23 – Caso de uso manter conta: operação incluir

No Quadro 24 está a descrição da operação excluir do caso de uso manter conta.

<p>Caso de uso: Manter conta: operação excluir</p> <p>Descrição: Quando necessário, o gerente removerá uma conta.</p> <p>Atores: Gerente.</p> <p>Pré-condição: Estar conectado ao sistema. Possuir conta cadastrada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de contas e seleciona a conta que deseja excluir. 2. Gerente pressiona no botão excluir. 3. Sistema retorna aviso que serão excluídos todos os registros vinculados àquela conta. 4. Gerente confirma exclusão. 5. Sistema faz exclusão dos dados no banco. 6. Sistema informa que os dados foram excluídos. <p>Pós-Condição: Dados cliente removidos no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4: Gerente não confirma exclusão	4.1 Sistema retorna a tela de listagem de clientes
Linha 3: Conta possui valores monetários vinculados.	3.1 Sistema apresenta opção para registro de movimentação do valor na conta. 3.2 Ator seleciona opção desejada

Quadro 24 – Caso de uso manter conta monetária: operação excluir

No Quadro 25 está a descrição da operação editar do caso de uso conta monetária.

<p>Caso de uso: Manter conta: operação editar.</p> <p>Descrição: Quando necessário, o gerente editará uma conta.</p> <p>Atores: Gerente.</p> <p>Pré-condição:</p>	
--	--

<p>Estar conectado ao sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Gerente acessa a tela de listagem de contas. 2. Gerente aperta no botão editar. 3. Sistema mostra tela com campos de cadastro preenchidos com os dados da conta atual . 4. Gerente edita os campos desejados e envia o formulário. 5. Sistema registra alterações no banco de dados. 6. Sistema informa que os dados foram editados. <p>Pós-Condição:</p> <p>Dados conta editados no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Gerente informa campos incorretamente. Ou não informa obrigatórios	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente

Quadro 25 – Caso de uso manter conta: operação editar

No Quadro 26 está a descrição da operação registrar entrada/saída do caso de uso manter conta monetária.

<p>Caso de uso:</p> <p>Manter conta: operação registrar entrada/saída</p> <p>Descrição:</p> <p>Quando é realizada alguma transação ou registro de pagamento que não está incluso nas funções do sistema, o gerente ou o caixa poderão acrescentá-la a seu movimento identificando com observações.</p> <p>Atores:</p> <p>Gerente ou caixa.</p> <p>Pré-condição:</p> <p>Possuir conta cadastrada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de movimento de contas. 2. Ator pressiona no botão adicionar registro. 3. Sistema mostra tela com formulário. 4. Ator edita os campos desejados e envia o formulário. 5. Sistema acrescenta valores no banco de dados. 6. Sistema informa que os dados foram salvos. <p>Pós-Condição:</p> <p>Valores adicionados no banco de dados</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Ator informa campos incorretamente ou não inclui dados de campos obrigatórios.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 26 – Caso de uso manter conta: operação registrar entrada/saída de caixa

No Quadro 27 está a descrição da operação alterar entrada/saída do caso de uso conta monetária.

<p>Caso de uso:</p> <p>Manter conta: operação corrigir entrada/saída em conta monetária.</p>

<p>Descrição: Caso tenha registrado uma entrada/saída de caixa de forma incorreta, o gerente ou caixa poderão corrigi-la.</p> <p>Atores: Gerente ou caixa.</p> <p>Pré-condição: Ter algum lançamento de entrada/saída de conta.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de movimento de contas. 2. Ator escolhe o registro qual deseja editar. 3. Sistema mostra tela com formulário preenchido. 4. Ator edita os campos desejados e envia o formulário. 5. Sistema acrescenta valores no banco de dados. 6. Sistema informa que os dados foram salvos. <p>Pós-Condição: Valores adicionados no banco de dados. Correção adicionada no banco.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 4. Ator informa campos incorretamente ou não preenche campos obrigatórios.	4.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios não preenchidos.

Quadro 27 – Caso de uso manter conta: operação corrigir entrada/saída em conta monetária

No Quadro 28 está a descrição do caso de uso abrir caixa.

<p>Caso de uso: Abrir caixa</p> <p>Descrição: Ao iniciar um período de vendas o funcionário caixa inicia uma sessão que é chamada de caixa de movimento por período.</p> <p>Atores: Caixa.</p> <p>Pré-condição: Ter uma conta monetária registrada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de movimento de contas. 2. Ator clica em abrir caixa. 3. Sistema mostra tela com formulário. 4. Ator edita os campos desejados e envia o formulário. 5. Sistema abre sessão para armazenar vendas e cancelamentos nesse período de caixa. <p>Pós-Condição: Sessão para registro de vendas/cancelamentos aberta</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 3. Ator informa campos incorretamente ou não insere dados em campos de preenchimento obrigatório.	3.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que campos obrigatórios estão sem preenchimento.

Quadro 28 – Caso de uso abertura de caixa

No Quadro 29 está a descrição do caso de uso iniciar venda.

<p>Caso de uso: Iniciar venda</p> <p>Descrição: Se inicia uma sessão para armazenar no banco um registro de saída de produtos e entrada de dinheiro.</p> <p>Atores: Caixa ou garçom, balança, cliente.</p> <p>Pré-condição: Caixa estar aberto.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cliente pede ao garçom ou ao caixa um produto ou realiza a pesagem de um prato na balança. 2. Garçom ou caixa entra na tela de vendas e clica em nova venda. 3. Sistema apresenta tela com o id da nova venda e a lista de produtos. 4. Garçom ou caixa informa o produto e a quantidade. 5. Garçom ou caixa pressiona confirmar. 6. Sistema inicia id de venda. 7. Sistema retorna a tela de vendas. <p>Pós-Condição: Sessão com id de venda iniciada.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 1. Cliente adere produto por quilo, e o coloca prato na balança,	1.1 Balança informa valor da pesagem ao sistema. 1.2 Sistema inicia id de venda com produto vinculado à balança. 1.3 Sistema envia à impressora arquivo com id de venda para ser impresso. 1.4 Impressora entrega ao cliente papel com seu id de venda.

Quadro 29 – Caso de uso iniciar venda

No Quadro 30 está a descrição do caso de uso incluir produto a venda.

<p>Caso de uso: Incluir produto a venda</p> <p>Descrição: Cliente pede ao garçom ou caixa para incluir produto em venda aberta.</p> <p>Atores: Caixa ou garçom, cliente.</p> <p>Pré-condição: Ter uma venda iniciada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cliente pede ao garçom ou caixa para incluir produto em venda aberta 2. Garçom ou caixa acessa tela de vendas. 3. Garçom ou caixa clica em adicionar produto. 4. Sistema apresenta formulário com lista de produtos e id de vendas abertos. 5. Garçom ou caixa preenche o formulário e confirma 6. Sistema registra produto à venda 7. Sistema retorna a tela de vendas <p>Pós-Condição: Acrescenta produto relacionado a venda aberta</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 5 Ator informa campos	5.1 Sistema retorna ao formulário informando

incorretamente ou não informa campos de preenchimento obrigatório.	que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.
--	---

Quadro 30 – Caso de uso incluir produto a venda

No Quadro 31 está a descrição do caso de uso remover produto de venda.

<p>Caso de uso: Remover produto de venda.</p> <p>Descrição: Cliente pede ao garçom ou caixa para cancelar um produto</p> <p>Atores: Caixa ou garçom, cliente.</p> <p>Pré-condição: Ter uma venda iniciada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cliente pede ao garçom ou caixa para cancelar um produto que foi incluído em uma venda. 2. Garçom ou caixa acessa tela de vendas. 3. Garçom ou caixa clica na venda de id fornecido pelo cliente. 4. Garçom ou caixa clica em remover item. 5. Sistema apresenta formulário com lista de produtos já relacionados a venda informada. 6. Garçom ou caixa preenche o formulário e confirma 7. Sistema remove produto da venda 8. Sistema retorna a tela de vendas <p>Pós-Condção: Acrescenta produto relacionado a venda aberta</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6 Ator informa campos incorretamente ou não preenche dados de campos obrigatórios.	6.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.
Linha 6 Ator informa que produto deverá ser reinserido no estoque.	6.1 Sistema atualiza quantidade do produto no estoque.

Quadro 31 – Caso de uso remover produto de venda

No Quadro 32 está a descrição do caso de uso fechar venda.

<p>Caso de uso: Fechar venda.</p> <p>Descrição: Caixa fecha sessão aberta para venda.</p> <p>Atores: Caixa, cliente.</p> <p>Pré-condição: Ter uma venda iniciada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cliente se direciona ao caixa para fechar venda. 2. Caixa acessa tela de vendas. 3. Caixa encontra venda com id fornecido pelo cliente e clica em fechar. 4. Sistema lista produtos comprados e valor total. 	
--	--

<p>5. Cliente paga valor total. 6. Caixa confirma fechamento. 7. Sistema atualiza quantidade de produtos em estoque. 8. Sistema registra transição monetária. 9. Sistema encerra sessão de venda.</p> <p>Pós-Condição: Venda aberta finalizada.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6 Ator informa campos incorretamente ou não inclui dados para campos de preenchimento obrigatório.	6.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 32 – Caso de uso fechar venda

No Quadro 33 está a descrição do caso de uso manter cardápio.

<p>Caso de uso: Manter cardápio.</p> <p>Descrição: Cozinheiro compõe cardápio tendo como base produtos cadastrados. O cardápio define o menu do restaurante.</p> <p>Atores: Cozinheiro.</p> <p>Pré-condição: Não há.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cozinheiro abre o formulário para composição de cardápio e escolhe os produtos, previamente cadastrados, que farão parte do cardápio. 2. Cozinheiro envia dados. 3. Sistema inclui cardápio no banco de dados. <p>Pós-Condição: Cardápio cadastrado no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6 Cozinheiro informa campos incorretamente ou não inclui dados para campos de preenchimento obrigatório.	6.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 33 – Caso de uso manter cardápio

No Quadro 34 está a descrição do caso de uso compor produto.

<p>Caso de uso: Compor produto.</p> <p>Descrição: Cozinheiro compõe produtos a partir de ingredientes. Os produtos são utilizados na composição de cardápios. Os produtos podem ser definidos genericamente como os pratos, as sobremesas e outros ofertados no cardápio do restaurante.</p> <p>Atores: Cozinheiro.</p> <p>Pré-condição: Não há.</p>	
--	--

<p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cozinheiro abre o formulário para composição de produtos. 2. Cozinheiro envia dados. 3. Sistema inclui produto no banco de dados. <p>Pós-Condição:</p> <p>Cardápio cadastrado no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6 Cozinheiro informa campos incorretamente ou não inclui dados para campos de preenchimento obrigatório.	6.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 34 – Caso de uso compor produto

No Quadro 35 está a descrição do caso de uso registrar saída de ingredientes.

<p>Caso de uso:</p> <p>Registrar saída de ingredientes.</p> <p>Descrição:</p> <p>Cozinheiro registra a saída de ingredientes do estoque como forma de atualização. Esses ingredientes são utilizados na composição dos produtos e na elaboração dos alimentos.</p> <p>Atores:</p> <p>Cozinheiro.</p> <p>Pré-condição:</p> <p>Necessidade de atualizar estoque de ingredientes.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Cozinheiro acessa formulário para registrar saída de ingredientes do estoque. 2. Cozinheiro escolhe o produto e informa a quantidade a ser debitada do estoque, em seguida envia o formulário. 3. Sistema atualiza dados na quantidade dos respectivos produtos, debitando os valores informados. <p>Pós-Condição:</p> <p>Quantidade em estoque dos produtos atualizada.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 6 Cozinheiro informa campos incorretamente ou não inclui dados para campos de preenchimento obrigatório.	6.1 Sistema retorna ao formulário informando que dados foram preenchidos incorretamente ou que há campos obrigatórios sem preenchimento.

Quadro 35 – Caso de uso registrar saída de ingredientes

Na Figura 2 está o diagrama de classes de análise do sistema.

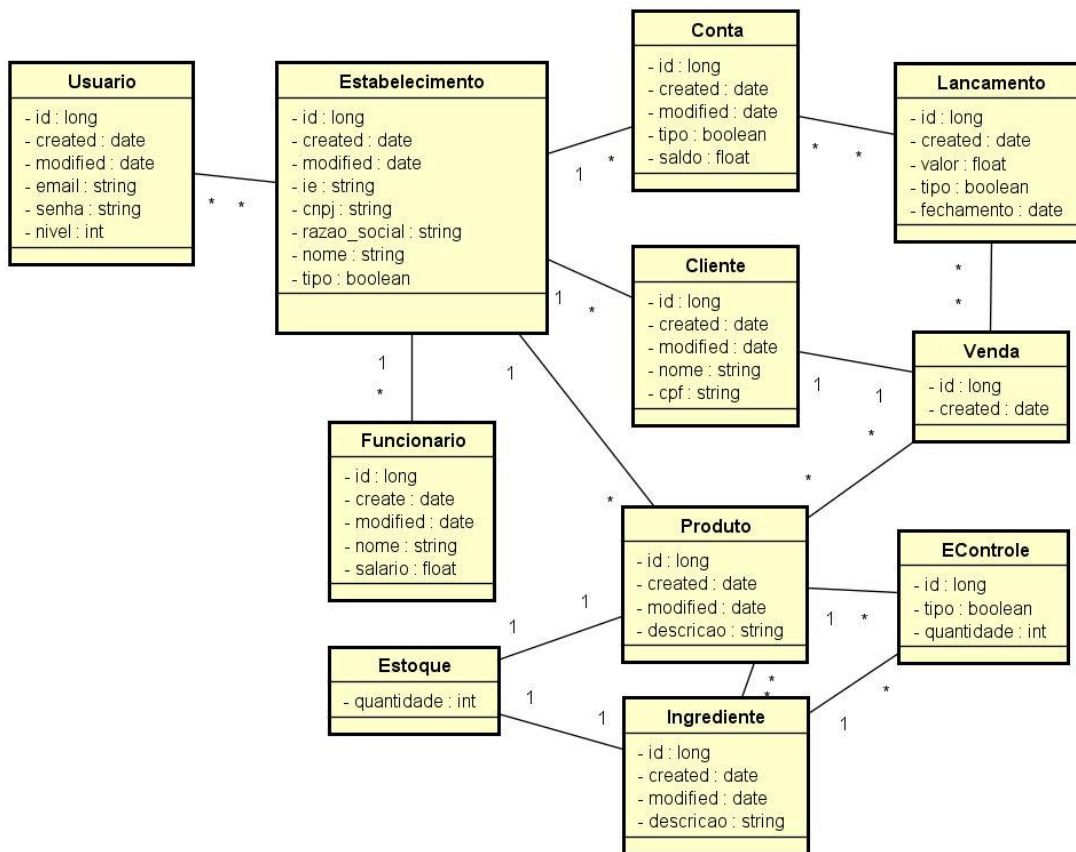


Figura 2 – Diagrama de classes de análise do sistema

As classes apresentadas no diagrama da Figura 2 estão documentadas a seguir. Para cada classe são criados dois arquivos de controle, um *controller* e um AJAX. Por exemplo: para a classe usuário são feitos os arquivos usuarioController.php e usuarioAjax.js. O primeiro faz as interações entre a classe e o banco, e o segundo faz as comunicações entre a apresentação e os serviços.

No Quadro 36 está a apresentação da classe Usuario.

Identificação	Usuario
Descrição	Todos os usuários que terão acesso ao sistema
Requisitos	RF02
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data das últimas modificações; email e senha: identificadores de acesso; nível: define as funcionalidades que terá acesso
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 36 – Descrição da classe Usuario

No Quadro 37 está a apresentação da classe Estabelecimento.

Identificação	Estabelecimento
Descrição	Os estabelecimentos cadastrados para determinados usuários
Requisitos	RF01
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data das últimas modificações; tipo: define as funções de controle; saldo: um campo para facilitar o find do saldo.
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 37 – Descrição da classe estabelecimento

No Quadro 38 está a apresentação da classe Conta.

Identificação	Conta
Descrição	As contas de controle monetário para cada estabelecimento
Requisitos	RF08
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data das últimas modificações; ie, cnpj, razão social e nome são apenas descrições do objeto; tipo: define as funcionalidades para aquele estabelecimento
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 38 – Descrição da classe conta

No Quadro 39 está a apresentação da classe Lançamento.

Identificação	Lancamento
Descrição	Toda rotação monetária que ocorre no sistema
Requisitos	RF09
Atributos	Id: Interações com o banco; created: data de inclusão; valor: o valor de entrada ou saída; tipo: definição de como a função tratara o lançamento; fechamento: o momento em que se fecha o lançamento, caso null o lançamento fica em aberto
Métodos	include; find[“campo”]by[“campo”]; findAndFinish;

Quadro 39 – Descrição da classe lançamento

No Quadro 40 está a apresentação da classe Cliente.

Identificação	Cliente
Descrição	Os clientes dos estabelecimentos cadastrados
Requisitos	RF06
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data das últimas modificações; nome e cpf são apenas descrições do objeto. Que nesse caso recebem unique;
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 40 – Descrição da classe cliente

No Quadro 41 está a apresentação da classe Venda.

Identificação	Venda
Descrição	As sessões de venda abertas. Armazenará os ids dos produtos vendidos e criará os lançamentos. Ela que dará início as funções de alterar estoque
Requisitos	RF11
Atributos	Id: Interações com o banco; created: data de inclusão;
Métodos	include; find[“campo”]by[“campo”]; .

Quadro 41 – Descrição da classe venda

No Quadro 42 está a apresentação da classe funcionário.

Identificação	Funcionário
Descrição	Os funcionários cadastrados ao sistema, que servirão para ajuda nas definições de gastos.
Requisitos	RF02
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data da última modificação; campos de descrição do objeto
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 42 – Descrição da classe funcionário

No Quadro 43 está a apresentação da classe produto.

Identificação	Produto
Descrição	Os Produtos serão todo objeto que será vendido no estabelecimento.
Requisitos	RF02
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data da última modificação; descrição: campo de descrição do objeto
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 43 – Descrição da classe produto

No Quadro 44 está a apresentação da classe ingrediente.

Identificação	Ingrediente
Descrição	Serão usados para compor produtos, porém não serão vendidos separadamente
Requisitos	RF04
Atributos	Id: Interações com o banco; created: data de inclusão; modified: data da última modificação; descrição: campo de descrição do objeto
Métodos	include; find[“campo”]by[“campo”]; merge.

Quadro 44 – Descrição da classe ingrediente

No Quadro 45 está a apresentação da classe econtrole.

Identificação	EControle
Descrição	Esta classe manterá o registro de entrada e saída de estoque. Dizendo quando foi feito, com qual quantidade.
Requisitos	RF14
Atributos	Id: Interações com o banco; created: data de inclusão; tipo: se é entrada ou

	saída de produto; quantidade: a quantidade do produto.
Métodos	include; find[“campo”]by[“campo”]; .

Quadro 45 – Descrição da classe econtrole

4.4 APRESENTAÇÃO DO SISTEMA

A seguir é apresentado o módulo de pagamento de mensalistas do sistema. A Figura 3 apresenta a tela de acesso (*login*) no sistema. Essa tela é composta apenas pelos campos *login* e senha.

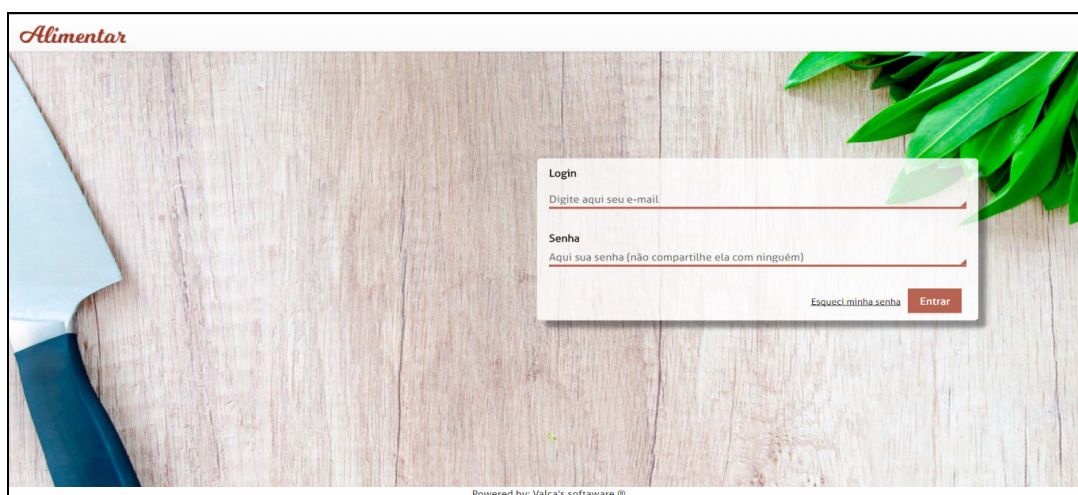


Figura 3 – Tela de acesso ao sistema

Ao acessar o sistema, o usuário gerente possuirá uma visão dos clientes cadastrados e seus saldos em aberto, como mostra a Figura 4. Esses dados são apresentados em uma tabela no centro da tela. Além disso, no menu superior destacam-se as principais funcionalidades referentes ao controle de mensalistas, que são os clientes que efetuam pagamento mensal das refeições que realizam. Por fim, acima da tabela há um botão para incluir um novo cliente.

Nome do cidadão	Telefone	Trabalha onde?	Nos deve..	
Aline Mjsezok	(46) 9999-9999	Consórcio Intermunicipal de Saude	R\$ 0.00	
Ana Paula Silstir	(46) 9999-9999	Restaurante Estação Gourmet	R\$ 0.00	
Barbara C França da Silva	(46) 9932-0501	Consórcio Intermunicipal de Saude	R\$ 15.93	
Bruno Behr Neto	(46) 9834-2107	Justiça do Trabalho	R\$ 106.66	
Bruno Ziem	(64) 9923-2295	Fórum da Comarca de Pato Branco	R\$ 46.56	
Cacilda Ap Santos	(46) 9121-5020	Consórcio Intermunicipal de Saude	R\$ 0.00	
Camila Faretti	(46) 9464-6464	Restaurante Estação Gourmet	R\$ 146.67	
Clímene Sutili	(46) 9133-5554	Consórcio Intermunicipal de Saude	R\$ 905.68	
Danieli Sloboda	(46) 8822-7073	Restaurante Estação Gourmet	R\$ 0.00	
Denise Valente	(46) 3225-5454	Restaurante Estação Gourmet	R\$ 132.85	
Dilson Vlasoft	(46) 9946-4494	Vlasoft Informatica Ltda - Me	R\$ 0.00	
Dirceu Consoli	(46) 3025-1357	Fórum da Comarca de Pato Branco	R\$ 668.36	

Figura 4 – Tela inicial do módulo mensalistas

O cadastro de um novo mensalista (clientes que utilizam o restaurante e fazem pagamento mensal) é realizado pelo botão novo que fica à direita acima da listagem de mensalistas cadastrados. A tela de cadastro de mensalistas, apresentada na Figura 5, possui verificação em todos os campos.

Cadastro de Mensalista

São apenas obrigatórios os campos com (*)

Nome:* Sobrenome:*

CPF: R.G.:

Trabalha em:*

Telefone Pessoal:* Pagamento: Mensal Quinzenal

Desconto: Dia do pagamento:

Figura 5 – Cadastro mensalistas

As *tags* em cor amarela (região destacada na Figura 5) que ficam na parte superior da tela são informativas e podem ser removidas. Elas foram criadas para informar que houve um erro no momento do cadastro, que a operação foi realizada adequadamente e para exibir *alerts*. A região destacada da Figura 6 apresenta uma *tag* em cor verde indicando que a operação solicitada foi realizada.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

Cadastro de Mensalista

Mensalista mais que salvo! x

Nome: João Sobrenome: Almeida

CPF: 200.000.000-00 R.G.: 00.000.000-00

Trabalha em: Viasoft Informatica Ltda - Me

Telefone Pessoal: (64) 9999-4949 Pagamento: Mensal Quinzenal

Desconto: 10 Dia do pagamento: (dosdois) 1 10

Cadastrar

Powered by: Valca's software ©

Figura 6 – Cadastro mensalista: detalhe de tag de mensagem

A verificação dos campos é exibida na parte lateral do formulário. Como apresentado na Figura 7.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

Cadastro de Mensalista

Por favor, um nome com mais de 1 letra :)

O sobrenome com mais de 1 letra também

Confere o CPF lá

Oi, esse RG confere?

O campo #diapagamento é obrigatório

Selecione um estabelecimento né

O campo #prcDesconto é obrigatório

Informe o telefone :D

Nome: a Sobrenome: a

CPF: 4 R.G.: 4

Trabalha em: Seleccione uma empresa...

Telefone Pessoal: (46) Pagamento: Mensal Quinzenal

Desconto: 0% Dia do pagamento: (dosdois) 00 00

Cadastrar

Powered by: Valca's software ©

Figura 7 – Cadastro mensalista: verificação de campos

Para lançar um valor para um cliente é apresentado um *auto-complete field*, o campo valor e a data. Nessa tela, além do desenvolvimento do *auto-complete*, foram implementados detalhes nos outros dois campos. No campo valor, a máscara foi personalizada para facilitar a entrada de valores em reais. E no campo data, além da máscara, a data atual é completada no momento que o campo recebe o foco. A tela para lançamentos de pagamento de mensalista é apresentada na Figura 8.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

Lançar Vias

Lembre-se de conferir a data!

Cliente: Bruno Behr Neto

Valor: R\$ 50,00 Data: 26/11/2017

Lançar

Figura 8 – Tela para lançamento de pagamento de mensalistas

Para registrar o pagamento, a tela de recebimentos também é composta por um *auto-complete field*. Ao encontrar o nome do cliente são mostrados dos dados base do cliente e são apresentadas algumas possibilidades de pagamento. Uma dessas possibilidades é pagar todos os débitos do cliente, quitar apenas o mês e a última opção, que exibe detalhadamente os débitos, é a possibilidade de pagar cada um dos débitos individualmente. Na caixa que exibe todos os débitos são marcados em vermelho os que ultrapassaram a data de vencimento. A Figura 9 mostra a tela de pagamento dos mensalistas.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

De quem iremos receber dinheiros:

Bruno Behr Neto

Total: R\$ 106,66 com 15% de desconto: **R\$ 90,66**
Pagar todas

Nome: Bruno Behr Neto
Telefone: (46) 9834-2107
Empresa: Justiça do Trabalho

Novembro

Total: R\$ 50,00 com 15% de desconto: **R\$ 42,50**
Pagar mês

Data	Valor
17/11/2017	R\$ 50,00
10/09/2017	R\$ 56,66

Selecionar vencidas! Pagar Selecionadas

Powered by: Valca's software®

Figura 9 – Tela de pagamento de débitos de mensalistas

Esse será o padrão de tela para todas as funcionalidades do sistema.

4.5 IMPLEMENTAÇÃO DO SISTEMA

Para iniciar a implementação foi definida uma estrutura de pastas para organizar o projeto. Essa estrutura é apresentada na Figura 10. Usando parte da estrutura *Model-View-Controller* (MVC) foram criadas as pastas *ajax* e *controller*. Nelas são criadas as rotas de conexão com o *backend* e os arquivos de função para o controle de dados respectivamente. Além disso, os *controllers* chamam os arquivos de conexão com banco na pasta *conn*.

Para o *front-end* há a pasta *style*, *img*, *pattern* e *access*. A *style* armazena os arquivos que dão estilo para o sistema e os arquivos de funções executadas na página em nível de usuário. Em conjunto com ela, a pasta *img* arquiva as imagens necessárias para qualquer ambiente.

Por fim, estão as pastas *pattern* e *access* que contêm os arquivos de visualização na *webview*. A pasta *pattern* contém os arquivos que são padrão, como *nav*, *footer* e *imports*. E a pasta *access* contém formulários, *dashboards* e listas.

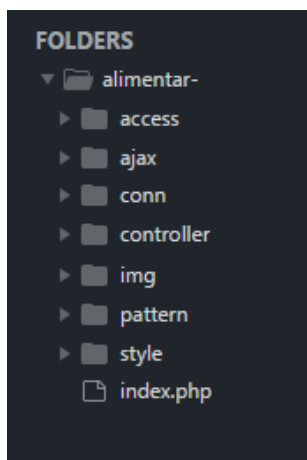


Figura 10 – Estrutura de pastas para organização do projeto

Iniciando pelos arquivos de conexão, foi utilizada a biblioteca *PDO* do *PHP*. Além de fazer a conexão de forma clara e simples, estrutura os comandos *Structured Query Language* (SQL) com variáveis auxiliando na inclusão por métodos. A Listagem 1 apresenta o arquivo de conexão.

```
$servidor = 'localhost';
$port = '3306';// Servidor MySQL
$usuario = '****';           // Usuário MySQL
$senha = '*****';         // Senha MySQL
```

```

$banco = 'alimentar';
$parametros = array(PDO::MYSQL_ATTR_INIT_COMM
$file = "mysql:host=".$servidor.";dbname=".$banco.";port=".$port
$pdo = new PDO($file, $usuario, $senha, $parametros);
if(!$pdo){
    echo ' Erro na conexão com o banco';
    die();
}AND => "SET NAMES UTF8", PDO::DATE_DEFAULT_TIMEZONE => "America/Sao_Paulo");

```

Listagem 1 – Arquivo de conexão

No código apresentado na Listagem 1 são inicializadas variáveis exigidas pelo *PHP Data Objects (PDO)* para a conexão com o banco. Nelas estão as propriedades da conexão com o MySQL, definição de estilo de codificação binária e fuso horário padrão.

Assim, é instanciada a variável *\$pdo* com um tipo de objeto *PDO*, passando na ordem: *String* de conexão com MySQL, usuário de acesso ao banco, senha para esse usuário e um *array* com parâmetros da conexão. Sempre que será necessário manipular dados no banco, são utilizados os atributos da classe por esse objeto instanciado.

Para exemplificar essa conexão em uso é apresentada uma função de busca de nível de usuário que está na Listagem 2.

```

function select_nivel_acesso_usuario(){
    $.ajax({
        method: 'POST',
        url: url + '/controller/usuario_controller.php',
        data: {
            acao: 'get_nivel',
        },
        async: false,
        cache: true,
        success: function(data){
            var result = JSON.parse(data);
            var ul_categoria_usuario = $('#ul_categoria_usuario');
            ul_categoria_usuario.empty();
            for (var i = 0; i < result.length ; i++) {
                var li = document.createElement('li');
                var a = document.createElement('a');
                a.innerHTML = result[i]['nivel'];
                a.href = '../access/' + result[i]['nivel'].toLowerCase() + '.php';
                li.append(a);
            }
        }
    });
}

```

```

        ul_categoria_usuario.append(li);
    } } }); }

```

Listagem 2 – Função para buscar as permissões de acesso de usuários

A função apresentada na Listagem 2 está na pasta *ajax*, no arquivo *usuario_ajax.js*. Aqui é usado a biblioteca JavaScript *jQuery* para enviar requisições *Hypertext Transfer Protocol* (HTTP) ao servidor utilizando a função *jQuery.ajax()*. Nessa função é explicitado que o método de envio será *post*. É passada a *Uniform Resource Locator* (URL) destino e um *JavaScript Object Notation* (JSON) de dados que leva como *array* de *strings* nominadas. Nesse JSON é indicado que a função será para apanhar o nível do usuário. Além disso, são declaradas algumas propriedades de funcionamento e, por fim, é alimentado um menu de característica que o usuário pode assumir em decorrência das suas permissões (nível) de acesso que virá como retorno no *success* como *data*.

Ao alcançar a *Uniform Resource Locator* (URL) destino é realizada a conexão com o banco de dados como apresentado na Listagem 3.

```

    $acao = $_POST['acao'];
    if($acao == 'get_nivel'){
        $select = $pdo->prepare('SELECT nau.usuario_id as usuario_id, na.descricao as nivel FROM
nível_acesso_usuario as nau, nível_acesso as na WHERE nau.usuario_id = :id AND nau.nível_acesso_id =
na.id');
        $select->execute(array (":id" => SESSION('ur000id')));
        while ($row = $select->fetch(PDO::FETCH_OBJ)) {
            $row2[] = $row;
        }
        echo json_encode($row2);
    }

```

Listagem 3 – Conexão com o banco de dados

O código apresentado na Listagem 3 corresponde às linhas 4 e 46 até 58 do arquivo *usuario_controler.php*. Nelas é recebida a *string* que foi passada como parâmetro pela ação pelo *Ajax* e é armazenada em uma variável *\$acao*.

Ao comparar a variável com a ação desejada e obter retorno verdadeiro é criada uma nova variável que será a de acesso ao banco. Nessa variável é armazenada uma instância do objeto *\$pdo* preparada para executar um *select* no banco. Nessa instrução é passado como parâmetro o *id* do usuário que está gravado na sessão. O retorno dessa chamada é armazenado em um *array* de *strings* que é

codificado para JSON. E é enviado junto a um código 200 para chamar o *success* do Ajax. Assim, é construído o menu em que o administrador pode fazer as funções de caixa ou gerente de estabelecimento e afins. Esse menu é apresentado na Figura 11.

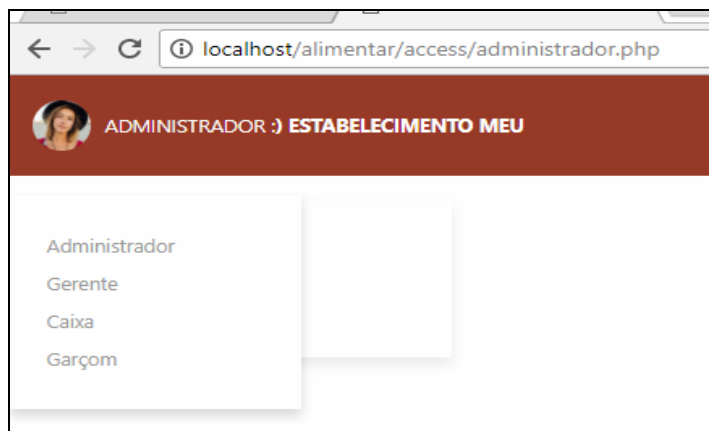


Figura 11 – Tela inicial de acesso ao sistema

Para exemplificar a criação do *front* é apresentada a função que exibe para o caixa ou para o gerente as vias em atraso de um cliente. Esse código é apresentado na Listagem 4.

```
function buscaVias(listitem, lista, divtotal, divapresentar, viasvazio){
    var id = $(listitem).attr('idcliente');
    var listaEl = $(lista);
    var total = 0.0;
    var valordouble = 0.0;
    var divtotalEl = $(divtotal);
    var viasvazioEl = $(viasvazio);
    var desconto = document.createElement('div');
    var totaldiv = document.createElement('div');

    listaEl.empty();
    $(divapresentar).hide();
    viasvazioEl.hide();
    $("#apresentar-vias").show();
    $.ajax({
        method: 'POST',
        url: url + "/controller/pendente_controller.php",
        data: {
            idcli : id,
            acao: 'busca'
        },
        async: false,
```

```

cache: false,
success: function (data) {
  if(data != 'Cliente não possui vias'){
    $(divapresentar).show();
    var vias = JSON.parse(data);

    for (var i = 0; i < vias.length; i++) {

      var diaAtual = new Date();
      var datavia = document.createElement('div');
      var valorvia = document.createElement('div');
      var chkbox = document.createElement('input');
      var divlinha = document.createElement('div');
      var length = document.createElement('div');

      if(diaAtual >= new Date(vias[i]['datavenc'])){
        divlinha.style = 'background-color: #ffb4b4; position: relative; padding: 4px';
      }else{
        divlinha.style = 'position: relative; padding: 4px';
      }
      length.name = 'v-selected';
      length.id = 'v-selected';
      divlinha.className = 'col-md-12';

      chkbox.name = 'v-selected' + i;
      chkbox.id = 'v-selected' + i;
      chkbox.type = 'checkbox';
      length.setAttribute('lengthvia', i);
      chkbox.value = vias[i]['id'];
      chkbox.style = 'position: absolute; padding: 4px; top: 3px; right: 35px; z-index: 99;';
      datavia.className = 'col-md-6';
      valorvia.className = 'col-md-6';
      valorvia.name = 'valorvia';
      valorvia.id = 'valorvia';
      datavia.innerHTML = tratardata(vias[i]['datavia']);
      valorvia.innerHTML = tratarvalor(vias[i]['valor']);
      valordouble = parseFloat(vias[i]['valor']);
      total = total + valordouble;
      total = parseFloat(total.toFixed(2));
      var totaldesconto = total - (total * vias[i]['prcdesc']/100);
      totaldesconto = parseFloat(totaldesconto.toFixed(2));
      desconto.innerHTML = tratarvalor(totaldesconto);
      desconto.style = 'text-align: center; margin-top: 7px; margin-bottom: 7px; font-weight: bold';
      totaldiv.innerHTML = 'Total: ' + tratarvalor(total) + ' com ' + vias[i]['prcdesc'] + '% de desconto: ';
      totaldiv.style = 'font-size: 10pt;';
    }
  }
}

```



```

        divtotalEl.empty();
        divtotalEl.append(totaldiv);
        divtotalEl.append(desconto);
        divlinha.append(chkbox);
        divlinha.append(datavia);
        divlinha.append(valorvia);
        listaEl.append(divlinha);
    }
} else {
    viasvazioEl.show();
}
listaEl.append(length);
}
});
}

```

Listagem 4 – Para seleção de vias de clientes em atraso

O código apresentado na Listagem 4 que é parte arquivo *pendentes_ajax.js*. A essência desse código é a mesma da função anterior, apresentada na Listagem 3. Porém, há a criação de outros elementos. É, bsicamente, realizada uma requisição HTTP à URL do *controller*, que retorna um JSON com todos os pagamentos pendentes de determinado usuário. Em seguida, usando funções do *JavaScript.document*, são criados os elementos necessários que são vinculados aos elementos já criados no HTML.

```

<div class='container'>
  <div class="row">
    <div class="col-md-2"></div>
    <div class="col-md-8">
      <label class="titulo">De quem iremos receber dinheiros:</label>
    </div>
  </div>
  <div class="col-md-2"></div>
</div>

<div class='row'>
  <div class="col-md-2"></div>
  <div class="col-md-8" style="position: relative">
    <input type="text" id="cliente" name="cliente"
      class="inputText"
      style="width: 100%;"
      onkeyup="completar('#cliente', '#listacli');"
      placeholder="Digite aqui o nome do cliente..."/>
  </div>
</div>

```

```

hidden="true"
name="listacli" id="listacli"
style=" background-color: #f5f5f5;
width: 90%;
position: absolute;
top:50px;
left: 25px;
z-index: 100;
">
</div>
</div>
<div class="col-md-2"></div>
</div>

<div class="row">
<div class="col-md-2"></div>
<div class="col-md-4" hidden='true'
id="vias-vazio" name='vias-vazio'>
<div class="row">
<div class="col-md-11" style=' margin-top: 20px;
margin-top: 20px;
padding: 10px;
background-color: #fff5d1;'>
Infelizmente, no momento, não temos dinheiro pra receber dessa pessoa :(
</div>
</div>
</div>
<div class="col-md-4" hidden='true'
id="vias-a-cobrar" name='vias-a-cobrar'>
<div class="row">
<div class="col-md-12" style='margin-top: 20px;'>
<div class='col-md-12' style=' background-color: #FDFFB1;
padding-top: 10px;
padding-bottom: 10px;'>
<div class="col-md-12" name='totalvias' id='totalvias'
style='margin-top: 8px;
font-size: 13pt;'></div>
<div class='col-md-12'>
<button class='btn btn-warning col-md-12' onclick="mostralabel('#confirm')"
name='pagar-todas' id='pagar-todas'>Pagar todas</button>
</div>
<div class='col-md-12' style='display: none'
name='confirm' id='confirm'>
<div class='col-md-12' style='text-align: center;
margin-top: 10px;

```

```

        margin-bottom: 10px;
        font-size: 11pt'>
        Realmente deseja pagar tudo?
    </div>
    <div class='col-md-6'>
        <button class='btn btn-warning col-md-12' onclick="hideLabel('#confirm')"
            name='pagar-todas' id='pagar-todas'>Era brincadeira</button>
    </div>
    <div class='col-md-6'>
        <button class='btn btn-warning col-md-12' onclick="pagartodas('#cliente')"
            name='pagar-todas' id='pagar-todas'>Sim desejo!</button>
    </div>
</div>

<div class='col-md-12' style=' background-color: #FDFFB1;
    padding-top: 10px;
    padding-bottom: 10px;
    margin-top: 10px;'>
    <div class="col-md-12">
        <select name="mes-soma" id="mes-soma"
            class="col-md-12"
            onchange="somarMes('#cliente', '#total-vias-mes', '#mes-soma')"
            style=" border: 0;
                background-color: #FDFFB1;
                text-align: center">
            <option value='01'>Janeiro</option>
            <option value='02'>Fevereiro</option>
            <option value='03'>Março</option>
            <option value='04'>Abril</option>
            <option value='05'>Maio</option>
            <option value='06'>Junho</option>
            <option value='07'>Julho</option>
            <option value='08'>Agosto</option>
            <option value='09'>Setembro</option>
            <option value='10'>Outubro</option>
            <option value='11'>Novembro</option>
            <option value='12'>Dezembro</option>
        </select>
    </div>
    <div class='col-md-12' style='margin-top: 8px;
        font-size: 13pt;
        text-align: center;
        margin-bottom: 10px'
        name='total-vias-mes' id='total-vias-mes'></div>

```

```

<div class='col-md-12'>
  <button class="btn btn-warning col-md-12"
    onclick="mostralabel('#confirm-2')"
    >Pagar mês</button>
</div>
<div class='col-md-12' style='display: none'
  name='confirm-2' id='confirm-2'>
  <div class='col-md-12' style='text-align: center;
    margin-top: 10px;
    margin-bottom: 10px;
    font-size: 11pt'>
    Quer pagar esse mês mesmo?
  </div>
  <div class='col-md-6'>
    <button class='btn btn-warning col-md-12' onclick="hideLabel('#confirm-2')"
      name='pagar-todas' id='pagar-todas'>Era brincadeira</button>
  </div>
  <div class='col-md-6'>
    <button class='btn btn-warning col-md-12' onclick="pagarmes('#cliente', '#mes-soma');
hideLabel('#confirm-2')"
      name='pagar-todas' id='pagar-todas'>Sim desejo!</button>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-4">
  <div class="col-md-12 container-blue"
    id='apresenta-cliente' name='apresenta-cliente'
    style='margin-top: 20px;
    display: none;'>
  </div>

  <div name='apresentar-vias' id='apresentar-vias'
    class='divescondeescroll col-md-12'
    style='display: none'>
    <div class='escondeescroll col-md-12' style=' margin-top: 20px;
      padding: 8px;
      overflow: auto;
      max-height: 210px;'>

      <div class='row'>
        <div class="col-md-6">Data</div>
        <div class="col-md-6">Valor</div>
      </div>
    </div>
  </div>

```

```

<div class='row' id='lista-vias' name='lista-vias'>
</div>
</div>
<button style='margin-top: 10px; float: left'
class="btn btn-warning"
onclick="selecionarvencidas(this, '#cliente')">
Selecionar vencidas</button>
<button style='margin-top: 10px; float: right'
class="btn btn-warning"
onclick="mostralabel('#confirm-3'); somarsel('#cliente', '#v-selected', '#confimr-pagar-sel')">
Pagar Seleccionadas</button>
</div>
<div class='col-md-12' style='display: none'
name='confirm-3' id='confirm-3'>
<div id="confimr-pagar-sel" name="confimr-pagar-sel"
class='col-md-12' style='text-align: center;
margin-top: 10px;
margin-bottom: 10px;
font-size: 11pt'>
</div>
<div class='col-md-6'>
<button class='btn btn-warning col-md-12' onclick="hideLabel('#confirm-3')"
name='pagar-todas' id='pagar-todas'>Era brincadeira</button>
</div>
<div class='col-md-6' style="margin-bottom: 50px">
<button class='btn btn-warning col-md-12' onclick="pagarseleccionadas('#v-selected');
hideLabel('#confirm-3')"
name='pagar-todas' id='pagar-todas'>Sim desejo!</button>
</div>
</div>
</div>

```

Listagem 5 – Definição de elementos de menu

Na codificação apresentada na Listagem 5 são trazidos vários contêineres previamente criados, porém ocultos. E conforme o resultado da requisição eles são exibidos e alimentados. Vale ressaltar que todas as classes presentes no documento vem do *framework Bootstrap*. Os resultados possíveis ao final dessa execução são, caso haja vendas pendentes ou não, como os apresentados na Figura 12.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

De quem iremos receber dinheiros:

Bruno Behr Neto

Infelizmente, no momento, não temos dinheiro pra receber dessa pessoa. :(

Nome:
Bruno Behr Neto
 Telefone:
 (46) 9834-2107
 Empresa:
 Justiça do Trabalho

Data Valor

Selecionar vencidas! Pagar Selecionadas

Figura 12 – Tela de recebimento de clientes: não há valores a receber

Na Figura 12 é apresentada a tela de um cliente do qual não há valores pendentes para serem recebidos. Na Figura 13 é apresentada a tela para recebimento de clientes, com valores pendentes (a receber). Nesse caso são, ainda, apresentadas as opções possíveis de pagamento.

Alimentar Anotar Vias Receber Dinheiros Cadastrar Sair

De quem iremos receber dinheiros:

Bruno Behr Neto

Total: R\$ 106,66 com 15% de desconto:
R\$ 90,66
 Pagar todas

Janeiro
 Total: R\$ 0,00 com 15% de desconto:
R\$ 0,00
 Pagar mês

Nome:
Bruno Behr Neto
 Telefone:
 (46) 9834-2107
 Empresa:
 Justiça do Trabalho

Data	Valor
17/11/2017	R\$ 50,00
10/09/2017	R\$ 56,66

Selecionar vencidas! Pagar Selecionadas

Figura 13 – Tela de recebimento de clientes: há valores a receber

5 CONSIDERAÇÕES FINAIS

Dos objetivos propostos alguns foram concluídos com sucesso e outros encontram-se em desenvolvimento. Os controles de mensalistas e de financeiro já possuem implementadas todas as funcionalidades definidas. Já o controle de salão e a implementação de relatórios ainda estão em desenvolvimento.

O desenvolvimento sofreu alguns atrasos, além de o sistema ser extenso, por vários fatores. O principal desses fatores é decorrente das mudanças realizadas na estrutura de desenvolvimento. Ao iniciar o projeto, o conhecimento da linguagem estava no processo inicial de aprendizagem. Com o passar do desenvolvimento, houve melhoria no entendimento do funcionamento das tecnologias sendo utilizadas, percebeu-se, então, que era necessário mudar várias técnicas de programação. Essas mudanças incluíam desde a estrutura do projeto, como as definições de pasta e organização de funções, até a estruturação de códigos e uso de simplificações.

Analisando esses aspectos, optou-se por reiniciar o projeto quando ele já estava em estágio avançado de desenvolvimento. Essa decisão veio, pois, o objetivo final é que o sistema funcione com fluidez, responsividade e atendendo os requisitos de criação HTML definidas para *websites* (como *tags* de marcação definidas pela Google e instruções da *w3schools*).

Embora tenha sido necessário reiniciar o projeto é possível afirmar que as ferramentas e as tecnologias escolhidas são adequadas para o desenvolvimento de sistemas *web* no escopo e contexto do que é o sistema objeto deste trabalho. A maneira de trabalhar com cada uma delas é simples e atinge o seu propósito.

Entre as perspectivas futuras está a de continuidade do desenvolvimento do sistema até que todos os objetivos e requisitos sejam implementados. O módulo de controle de recebimento de clientes mensalistas e os cadastros básicos já estão sendo utilizados em produção por um estabelecimento. O sistema será utilizado por outros estabelecimentos do mesmo grupo proprietário assim que concluído.

REFERÊNCIAS

AGUSTIN, José Luis Herrero. **Model-driven web applications**. In: Science and Information Conference, 2015, p. 954-964.

AMALFITANO, Domenico; FASOLINO, Anna Rita; TRAMONTANA, Porfírio. **Reverse engineering finite state machines from rich internet applications**. In: 15th Working Conference on Reverse Engineering (WCRE'08). 2008, p. 69–73.

BENJAMIN, Kamara; BOCHMANN, Gregor von; JOURDAN, Guy-Vincent; ONUT, Iosif-Viorel. **Some modeling challenges when testing rich Internet applications for security**. In: 3rd International Conference on Software Testing, Verification and Validation Workshop (ICST'10), 2010, p.403–409.

BERNARDI, Mario Luca; DI LUCCA, Giuseppe Antonio; DISTANTE, Damiano. **Model-driven fast prototyping of RIAs: from conceptual models to running applications**. In: 13th IEEE International Symposium on Web Systems Evolution (WSE 2011), 2011, p. 250-258.

BOZZON, Alessandro; COMAI, Sara; FRATERNALI, Pietro; CARUGHI, Giovanni Toffetti. **Conceptual modeling and code generation for rich internet applications**. In: ICWE, ACM Press, 2006, p. 353–360.

BRAMBILLA, Marco; PRECIADO, Juan Carlos; LINAJE, Marino; SANCHEZ-FIGUEROA, Fernando. **Business process-based conceptual design of rich internet applications**. In: 8th International Conference on Web Engineering (ICWE 08), IEEE CS Press, 2008, p. 155-161.

CASTELEYN, Sven; GARRIGÓS, Irene; MAZÓN, Jose-Norberto. **Ten years of rich internet applications: a systematic mapping study, and beyond**. ACM Transactions on the Web, v. 8, n. 3, art 18, 2014.

CARUGHI., Giovanni Toffetti. **Modeling data-intensive rich Internet applications with server push support**. In: 3rd International Workshop on Model-Driven Web Engineering (MDWE'07), 2007, p. 1-15.

FARRELL, Jason; NEZLEK, George S.; **Rich Internet Applications The Next Stage of Application Development**. Allendale: Editora Grand Valley State Univ., 2007.

FERNANDES, Nádia; COSTA, Daniel; NEVES, Sergio; DUARTE, Carlos; CARRIÇO, Luís. **Evaluating the accessibility of Rich Internet Applications**. In: International Cross-Disciplinary Conference on Web Accessibility (W4A '12), 2012, p. 1-4.

FRATERNALI, Piero; ROSSI, Gustavo; SANCHEZ-FIGUEROA, Fernando. Rich Internet Applications. **IEEE Internet Computing**, v. 9, n. 12, may/june 2010, p. 9-12.

JEREMY, Allaire. Macromedia flash MX - A next-generation rich client. **Macromedia White Paper**. March 2002. Disponível em: <<http://www.c2isoft.in/white-papers/richclient.pdf>>. Acesso em: 17 set. 2016.

LAWTON, George. New ways to build rich Internet applications. **Computer. Published by the IEEE Computer Society**, p. 10-12, 2008.

MELIÁ, Santiago; GÓMEZ, Jaime; PÉREZ, Sandy; DÍAZ, Oscar. Architectural and Technological Variability in Rich Internet Applications. **IEEE Internet Computing**, may/june 2010, 2010, p. 24-32.

MESBAH, Ali; BOZDAG, Engin; DEURSEN, Arie van. **Crawling Ajax by inferring user interface state changes**. In: 8th International Conference on Web Engineering (ICWE'08), 2008, p. 122–134.

PORTAL BRASIL. **Alimentação em bares e restaurantes cresce no Brasil**. 2014. Disponível em: <<http://www.brasil.gov.br/turismo/2014/06/alimentacao-em-bares-e-restaurantes-cresce-no-brasil>>. Acesso em: 07 set. 2016.

PRECIADO, Juan Carlos; LINAJE, Marino; COMAI, Sara; SANCHEZ-FIGUEROA, Fernando. **Designing rich internet applications with web engineering methodologies**. In: 9th IEEE Workshop Web Site Evolution (WSE 07), IEEE CS Press, 2007, p. 23–30.

ROUBI, Sarra; ERRAMDANI, Mohammed; MBARKI, Samir. **A Model driven approach to generate graphical user interfaces for Rich Internet Applications using interaction flow modeling language**. In: International Conference on Intelligent Systems Design and Applications (ISDA), 2015, p. 272-276

SCOTT, Kendall. **O processo unificado explicado**. Porto Alegre: Bookman, 2003.

WESCO. **10 apps para ajudar na gestão de bares e restaurantes**. Disponível em: <<http://wesco.com.br/10-apps-para-ajudar-na-gestao-de-bares-e-restaurantes/>>. Acesso em: 07 set. 2016.