

Controle de Pedidos por comandas para estabelecimentos à la carte

Maiara Goldacker¹,

Prof. Ricardo de la Rocha Ladeira (orientador)¹

¹Instituto Federal Catarinense - Campus Blumenau - Blumenau, SC - Brasil

maiaragoldacker@gmail.com, ricardo.ladeira@ifc.edu.br

1. Introdução

Diariamente, consumidores frequentam restaurantes à la carte, lanchonetes e hamburguerias que utilizam apenas papel e caneta para realizar controle de pedidos e pagamento. Muitas vezes uma comanda de papel é entregue ao cliente, e fica na mesa deste passando a responsabilidade do controle dos pedidos para o consumidor, podendo perder-se até na hora do pagamento.

É comum que estabelecimentos tenham a prática de cobrar um valor como taxa de perda dessas comandas, porém não é obrigação do cliente pagá-la, como está evidenciado nos artigos 39 e 51 do código de defesa do consumidor:

Inciso V do Artigo 39 da Lei nº 8.078 de 11 de Setembro de 1990

Art. 39. É vedado ao fornecedor de produtos ou serviços, dentre outras práticas abusivas: (Redação dada pela Lei nº 8.884, de 11.6.1994)

V - exigir do consumidor vantagem manifestamente excessiva;

Inciso IV do Artigo 51 da Lei nº 8.078 de 11 de Setembro de 1990

Art. 51. São nulas de pleno direito, entre outras, as cláusulas contratuais relativas ao fornecimento de produtos e serviços que:

IV - estabeleçam obrigações consideradas iníquas, abusivas, que coloquem o consumidor em desvantagem exagerada, ou sejam incompatíveis com a boa-fé ou a equidade;" (BRASIL, 1990).

Tal problema no controle de pedidos pode gerar uma perda substancial para os estabelecimentos, já que sem saber o que foi pedido, não há como saber o que deve ser cobrado, além de abrir uma margem de possibilidade de roubo por parte de algum funcionário mal intencionado.

Na rotina agitada de um estabelecimento, um pedido realizado por um cliente e anotado por um atendente pode não chegar até a cozinha, ou chegar de forma incorreta, causando transtornos para o consumidor não atendido, trazendo uma experiência negativa ao consumidor e afetando negativamente a imagem do estabelecimento.

O controle do atendimento das comandas, da realização do pedido, e da administração do consumo do cliente, além da quantificação do lucro sob o produto final torna-se difícil já que apenas o pagamento realizado pelos consumidores é controlado. Essa situação faz com que o estabelecimento não saiba o quanto de fato está lucrando, ou está desperdiçando. Pode acontecer por exemplo, de um lanche mais trabalhoso para ser produzido não seja tão consumido quanto imaginado, e o tempo utilizado para esta produção possa ser gasto para produtos mais vendidos.

Outra complicação encontrada pela falta no controle de pedidos, é a dificuldade de investir em promoções que possam atingir mais clientes, como por exemplo a criação de combos com lanches e bebidas mais pedidas, ou promoção de lanches que foram pouco vendidos, para que os produtos em estoque não se deterioreem pela falta de saída.

O ramo alimentício já entende a necessidade de automação em seus processos.

Para 2018, a tendência entre as empresas do setor é uma diversificação das soluções de automação, oferecendo mais ferramentas para que os empreendedores do ramo gastronômico invistam. “Acreditamos muito na automação de atendimento e pagamentos, essa é nossa grande aposta para 2018”, relata Hasson (DINO, 2017).

1.1. Tema/Problema

O controle e a organização nos pedidos de um estabelecimento são extremamente importantes para o bom atendimento e a satisfação dos clientes. Os pedidos precisam chegar na cozinha com suas particularidades, para que possam ser produzidos corretamente pelos cozinheiros, e para o estabelecimento não ter prejuízo os pedidos precisam ser anotados e cobrados perfeitamente.

Um estabelecimento que não possui um sistema de pedidos, e o realiza de forma manual, além de estar fadado a cometer erros, precisa de mais tempo na realização das funções, e por isso mais funcionários para atendimento. Além disso, até o pedido ser anotado e levado até a cozinha pode demorar e causar erros na ordem de produção dos produtos.

1.2. Objetivos Propostos/Solução dos Problemas

Esse trabalho possui como objetivo geral desenvolver um sistema desktop para gerenciamento de pedidos e pagamentos de um estabelecimento que trabalha com pedidos à la carte.

1.3. Escopo

O projeto possui como foco a entrega de um sistema desktop para controle de pedidos e pagamentos de um estabelecimento. O sistema deve oferecer módulos de usuários, produtos, pedidos, adicionais para produtos, pagamentos e comandas. Em todos os módulos é possível cadastrar, excluir, editar, consultar ou imprimir relatório de dados ativos. Os módulos de pedidos e pagamentos são os principais módulos do sistema, que são complementados com a função consulta de lucros, onde o usuário poderá realizar consulta de vendas e a consulta de pedidos, com a finalidade de prestar monitoramento da ordem e realização de pedidos para a cozinha.

O sistema não contemplará cadastros de clientes, integração com prefeitura para emissão de notas fiscais, ou integração com periféricos, como por exemplo, balanças. A solução proposta não prevê cópia segura de dados, tal funcionalidade deve ser configurada separadamente.

1.4. Viabilidade do Projeto

O requisito mínimo para uso do software será um computador com espaço para armazenamento do sistema: 3,56KB de armazenamento e 5,03MB em disco que é o

espaço ocupado com o bytecode já gerado, um monitor com uma resolução mínima de 1366 x 768, com sistema operacional Windows XP, 7, 8.1 ou 10. É importante também que o cliente possua Java8¹ instalado em sua máquina, para o correto funcionamento do software.

Não houve custo para implementação deste sistema, já que todas as ferramentas utilizadas são gratuitas, e toda a análise, desenvolvimento e testes foram realizados pela autora, sem necessidade de mais recursos.

1.5. Método de Trabalho

O desenvolvimento deste trabalho foi realizado utilizando o modelo de processo incremental. Este método de desenvolvimento permite que o cliente participe do processo de criação de cada função, aumentando as chances que ao final de cada ciclo incremental, o desenvolvimento chegue mais perto das expectativas iniciais do cliente.

No primeiro incremento de um produto que utiliza o modelo incremental temos apenas o essencial do produto, ou seja, os requisitos básicos que devem ser atendidos para o software entrar em operação. Portanto, no primeiro incremento muitos recursos complementares ainda não são entregues para os clientes. Após o término do primeiro incremento o cliente utiliza e avalia esse incremento fornecendo posteriormente um resultado ou feedback. Com base nesse resultado fornecido pelo cliente o próximo incremento é planejado considerando a modificação do primeiro incremento, caso seja necessário, de acordo com o feedback do cliente. Após a liberação de cada incremento é realizado esse mesmo processo até que o produto esteja completo (MEDEIROS, 2013).

Nos tópicos abaixo, estão listadas todas as ferramentas, padrões e tecnologias aplicadas no desenvolvimento deste sistema. Todas são gratuitas.

1.5.1 Padrão de Projeto MVC

Para o desenvolvimento do sistema proposto, foi utilizado o padrão de projeto MVC (*model-view-controller*), demonstrado graficamente pela Figura 1.

O padrão arquitetural Model-View-Controller (MVC) é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador, conforme a figura 1 que ilustra seu funcionamento. [...] Primeiramente o controlador (Controller) que interpreta as entradas do mouse ou do teclado enviado pelo usuário e mapeia essas ações do usuário em comandos que são enviados para o modelo (Model) e/ou para a janela de visualização (View) para efetuar a alteração apropriada. Por sua vez o modelo (Model) gerencia um ou mais elementos de dados, responde a perguntas sobre o seu estado e responde a instruções para mudar de estado. O modelo sabe o que o aplicativo quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema que está se tentando resolver. Por fim, a visão (View) gerencia a área retangular do display e é responsável por apresentar as informações para o usuário através de uma combinação de gráficos e textos (SOUZA, 2011).

¹ www.oracle.com/technetwork/pt/java/javase/downloads/jre8-downloads-2133155.html

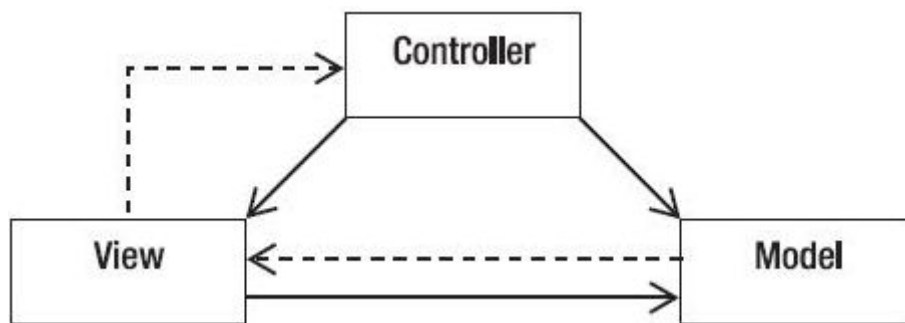


Figura 1. Interações utilizadas no projeto MVC. Elaborado por Marcio Ballem de Souza, 2013.

1.5.2 NetBeans IDE (*Integrated Development Environment*) 8.1

Para desenvolvimento do sistema e criação das interfaces, foi utilizado o ambiente de desenvolvimento integrado NetBeans IDE 8.1.

A plataforma NetBeans é uma estrutura genérica para aplicações Swing. Ela fornece facilidade ao desenvolvedor salvando estado, conectando ações a itens de menu, itens de barra de ferramentas e atalhos de teclado; gerenciamento de janelas e assim por diante. A plataforma NetBeans oferece todos esses recursos prontos para uso. Não é mais necessário codificar manualmente esses ou outros recursos básicos (NETBEANS, 2018).

1.5.3 MySQL WorkBench 6.3

Para criação do MER (Modelo entidade relacionamento), criação de tabelas, atributos e relacionamentos, além da realização de consultas no banco de dados foi utilizada a ferramenta MySQL WorkBench 6.3.

O MySQL Workbench é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs. O MySQL Workbench fornece modelagem de dados, desenvolvimento de SQL e ferramentas abrangentes de administração para configuração de servidores, administração de usuários, backup e muito mais. O MySQL Workbench está disponível no Windows, Linux e Mac OS X (MYSQL, 2018).

1.5.4 Hibernate 4.3 Final

O Hibernate é um framework à parte utilizado no Java para mapeamento de objeto relacional, persistência e acesso aos dados. Ele facilita a ligação entre aplicação e o banco de dados, e foi utilizado neste trabalho.

O Hibernate foi desenvolvido por uma equipe de programadores Java liderada por Gavin King e teve sua primeira versão divulgada em 2004. Segundo King, um dos objetivos ao criar o projeto era resolver seus problemas referentes à persistência causados pelo EJB 2.0. [...]. O Hibernate hoje é o framework Java para mapeamento objeto relacional mais conhecido no mercado. Sua principal função é abstrair o mapeamento, economizando esforço e preocupações concernentes a tal tarefa[...] é importante para simplificar o desenvolvimento e automatizar tarefas repetitivas de acesso a dados (IZALMO, 2009).

1.5.5 Hibernate Validator 4.3.1

O Hibernate Validator é uma API do Hibernate que se integra ao Hibernate possibilitando que o desenvolvedor realize consistências, diretamente pelas classes mapeadas do banco de dados. A Figura 2 é um exemplo do funcionamento do Hibernate *Validator* implementado neste projeto.

O Hibernate *Validator* já possui um conjunto de *annotations* para validações mais comuns, embora a API possibilite que o programador faça customizações (SOUZA, 2010).

```
@Size(max = 150)
@Column(name = "NMUSUARIO")
@NotNull(message = "Campo Nome do usuário é obrigatório")
private String nmusuario;
@Basic(optional = false)
@NotNull(message = "Campo usuário é obrigatório")
@Size(min = 1, max = 150)
@Column(name = "DSUSUARIO")
private String dsusuario;
@Basic(optional = false)
@NotNull(message = "Campo Senha é obrigatório")
@Size(min = 3, max = 500,message= "O campo senha deve conter no mínimo, 3 caracteres e no máximo, 500")
@Column(name = "DSSENHA")
private String dssenha;
```

Figura 2. Exemplo de utilização do Hibernate Validator em classe Hibernate mapeada pelo banco de dados. Elaborado pela autora.

1.5.6 IReport 5.6.0

O iReport é uma ferramenta que auxilia na criação de relatórios PDF (*Portable Document Format*), HTML (*HyperText Markup Language*), ODT (*Open Document*) entre outras extensões. Ele possui uma interface gráfica intuitiva, que possibilita organização de componentes gráfico por drag and drop.

O design do relatório do IReport (localização dos campos a serem preenchidos e seus respectivos nomes, para futuro mapeamento) é definido em um arquivo XML, que obedece a estrutura declarada no arquivo `jasperreports.dtd`.

Usando o XML, o designer pode definir textos estáticos, imagens, linhas e formas geométricas, como retângulos e elipses, e suas localizações dentro do relatório.

Além disso, pode-se ainda definir os campos que serão preenchidos dinamicamente a partir de uma base de dados.

O arquivo XML é compilado gerando um arquivo `.jasper`, onde as expressões Java existentes dentro do XML serão verificadas em tempo de compilação (JACQUES, 2009).

1.5.7 GitHub

O GitHub foi utilizado neste trabalho para armazenamento do código fonte, auxiliando no acompanhamento de alterações no código.

O Controle de Versão permite que você registre as mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo de forma que se possa recuperar versões específicas. Um Sistema de Controle de Versão (VCS) também permite que você reverta arquivos para um estado anterior, compare mudanças feitas no decorrer do tempo, veja quem foi o último a modificar algo

que pode estar causando problemas, ou quem introduziu um bug e quando, além de muitas outras alternativas (SILVEIRA, 2016).

2. Trabalhos Correlatos Existentes

Existem algumas aplicações com soluções similares ao software proposto neste trabalho disponível no mercado. Esta seção disserta sobre duas soluções: Linx degust e Consumer, comparando-as com o software desenvolvido. A Linx Degust foi escolhida por ser um *software* desenvolvido por uma empresa da região: Linx, de Blumenau, Santa Catarina. A Consumer foi escolhida para comparação por sua facilidade de contratação de pacotes: diretamente pelo site, sem necessariamente entrar em contato com a empresa que o desenvolveu.

2.1. Linx Degust

A Linx Degust é um sistema web de gestão especialista em fast-food, lanchonetes, bares e restaurantes.

Este software possui um grande número de funcionalidades, desde integração com periféricos até módulos de gestão (LINX DEGUST, 2019).

Abaixo estão as principais características encontradas no Software Linx Degust:

- Integração com balança;
- Emissão de notas fiscais;
- Controle de desperdício;
- Terminal de autoatendimento;
- Controle de produção de pratos;
- Acesso ao sistema pelo aplicativo *mobile*;

Cada módulo da solução oferecida pela Linx Degust deve ser adquirida pelo cliente separadamente gerando grande custo para o estabelecimento que deseja adquiri-lo. Além disso não foi encontrada nesta solução controle de acesso e permissão por usuário, podendo gerar transtornos com vazamento de informações privilegiadas.

Não foi encontrado valores para a contratação do software, sendo este fornecido apenas para pessoas que possuem CNPJ (Cadastro Nacional da Pessoa Jurídica), ao entrar em contato diretamente com a empresa fornecedora.

2.2. Consumer

O programa Consumer prevê uma gestão descomplicada para restaurantes, lanchonetes e pizzarias (CONSUMER, 2019).

O Consumer é encontrado na web com dois tipos de contratação: desktop e web. A solução web prevê também soluções mobile, e todas as soluções são pagas após um pequeno teste de uso de 30 dias. A solução consumer que mais se encaixa com os

requisitos levantados neste trabalho é a consumer desktop. As principais funcionalidades da consumer desktop são:

- Gestão de relatórios
- Controle de produção
- Fluxo de pedidos.
- Monitoramento de preparo

As principais soluções previstas no consumer são oferecidas pelo sistema desenvolvido neste trabalho, porém gratuitamente. O controle de acesso é limitado nessa versão, podendo haver acesso ao sistema apenas por um computador.

2.3. Discussão

A Tabela 1 representa graficamente a comparação entre os trabalhos correlatos apresentados, e o software desenvolvido neste trabalho.

Tabela 1. Tabela comparativa entre concorrente Linx Degust, Consumer Desktop e o software elaborado neste projeto: Gestão de restaurante. Elaborado pela autora.

Características	Consumer Desktop	Linx Degust	Software apresentado: Gestão de Restaurante
Valor para contratação	Pago	Pago	Gratuito
Controle de acesso por permissão de usuário	Não existe, apenas há possibilidade de acesso em uma máquina.	Não especificado	Controle de permissão de acesso por função dividido em acesso total, parcial ou mínimo
Contratação de funções	possibilidade de contratação de funções em pacote, quanto mais limitação de uso, mais barato.	Funções são contratadas separadamente, e a cada função adicionada o valor do software contratado aumenta.	Todas as funções para controle e gerenciamento disponibilizadas gratuitamente.
Código Aberto	Não	Não	Sim

3. Requisitos

Nesta seção estão os requisitos funcionais e não-funcionais do sistema.

3.1. Requisitos Funcionais

RF001 - O sistema deve permitir gerenciar informações de produtos.

RF002 - O sistema deve permitir gerenciar informações de comandas.

RF003 - O sistema deve permitir gerenciar informações dos usuários.

RF004 - O sistema deve permitir gerenciar informações de pedidos.

RF005 - O sistema deve permitir gerenciar informações de pagamento.

RF006 - O sistema deve permitir gerenciar adicionais de produtos.

RF007 - O sistema deve permitir acesso apenas aos usuários autenticados e ativos.

RF008 - O sistema deve permitir que apenas usuários com permissão total possam ter acesso a todas as funções.

RF009 - O sistema deve permitir que apenas usuários com permissão parcial e total possam gerenciar produtos, adicionais e comandas.

RF010 - O sistema deve permitir que todos os usuários ativos tenham permissão de realizar pedidos, pagamentos e consultar pedidos não atendidos.

RF011 - O sistema deve permitir que mais de uma comanda possa ser paga no mesmo pagamento.

RF012 - O sistema deve permitir que os produtos selecionados na comanda possam ser removidos/alterados na hora do pagamento.

RF013 - O sistema deve permitir que o pedido de um produto possa ter vários adicionais.

RF014 - O sistema deve permitir gerenciar informações de adicionais.

RF015 - Deve ser possível consultar informações de pagamento por datas específicas.

RF016 - Deve ser possível consultar informações de pagamento e pedidos por usuário.

RF017 - Deve haver emissão de relatório PDF referente a todas as consultas disponíveis no sistema.

RF018 - O sistema deve possuir monitoramento de pedidos para produção da cozinha.

3.2. Requisitos Não-Funcionais

RNF001 - A linguagem utilizada deve ser Java.

RNF002 - O banco de dados utilizado deverá ser MySQL.

RNF003 - O tempo de resposta não deverá ser maior que 10 (dez) segundos.

4. Diagramas UML

4.1. Caso de uso

Nesta seção está o diagrama de caso de uso construído para esta solução de sistema. Ele está graficamente representado na Figura 3.

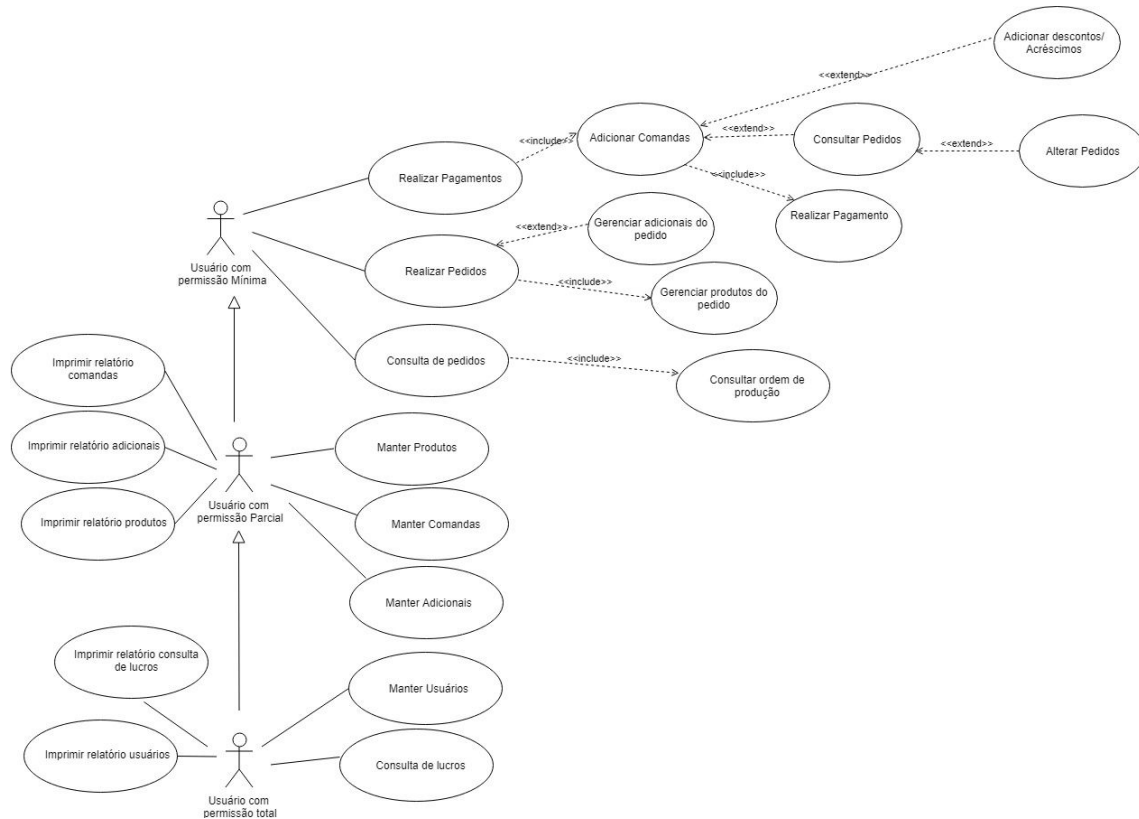


Figura 3. Diagrama de caso de uso. Fonte: Elaborado pela autora.

Nome: Manter Comandas.

Objetivos: Permitir que usuários com permissão total ou parcial possam consultar, adicionar, alterar ou inativar registros de Comandas.

Requisitos: RF002.

Pré-Condições: estar autenticado no sistema e ter permissão total ou parcial.

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Comandas.
3. O usuário preenche os campos de detalhe para cadastrar uma nova comanda, ou seleciona uma comanda no grid, altera seus campos e clica no botão alterar, ou seleciona uma comanda no grid, e clica em excluir para inativá-la. O usuário pode optar também, por imprimir um relatório PDF das comandas ativas do sistema, utilizando o botão imprimir.
4. O sistema atualiza o objeto conforme a ação do usuário.
5. O sistema atualiza a tabela de armazenamento de dados referente a comanda.

Fluxo de Exceção:

1. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.
2. Caso ocorra erros na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Manter Produtos.

Objetivos: Permitir que usuários com permissão total ou parcial possam consultar, adicionar, alterar ou inativar registros de Produtos.

Requisitos: RF001.

Pré-Condições: estar autenticado no sistema e ter permissão total ou parcial.

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Produtos.
3. O usuário preenche os campos de detalhe para cadastrar um novo produto, ou seleciona um produto no grid, altera seus campos e clica no botão alterar, ou seleciona um produto no grid, e clica em excluir para inativá-lo. O usuário pode optar também, por imprimir um relatório PDF dos produtos ativos do sistema, utilizando o botão imprimir.
4. O sistema atualiza o objeto conforme a ação do usuário.
5. O sistema atualiza a tabela de armazenamento de dados referente o produto.

Fluxo de Exceção:

1. Se o usuário não informar campos obrigatório no passo 3, o sistema emitirá uma mensagem de aviso informativa e não salvará os registros adicionados/modificados.
2. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.
3. Caso ocorra erros na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Manter Adicionais.

Objetivos: Permitir que usuários com permissão total ou parcial possam consultar, adicionar, alterar ou inativar registros de adicionais.

Requisitos: RF013.

Pré-Condições: estar autenticado no sistema e ter permissão total ou parcial.

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Adicionais.

3. O usuário preenche os campos de detalhe para cadastrar um novo adicional, ou seleciona um produto no grid, altera seus campos e clica no botão alterar, ou seleciona um adicional no grid, e clica em excluir para inativá-lo. O usuário pode optar também, por imprimir um relatório PDF dos adicionais ativos do sistema, utilizando o botão imprimir.
4. O sistema atualiza o objeto conforme a ação do usuário.
5. O sistema atualiza a tabela de armazenamento de dados referente ao adicional.

Fluxo de Exceção:

4. Se o usuário não informar campos obrigatório no passo 3, o sistema emitirá uma mensagem de aviso informativa e não salvará os registros adicionados/modificados.
5. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.
6. Caso ocorra erros na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Manter Usuários.

Objetivos: Permitir que usuários com permissão total possam consultar, adicionar, alterar ou inativar registros de Usuários.

Requisitos: RF003.

Pré-Condições: estar autenticado no sistema e ter permissão total.

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Usuários.
3. O usuário preenche os campos de detalhe para cadastrar um novo usuário, ou seleciona um usuário no grid, altera seus campos e clica no botão alterar, ou seleciona um usuário no grid, e clica em excluir para inativá-lo. O usuário pode optar também, por imprimir um relatório PDF dos usuários ativos do sistema, utilizando o botão imprimir.
4. O sistema atualiza o objeto conforme a ação do usuário.
5. O sistema atualiza a tabela de armazenamento de dados referente ao usuário.

Fluxo de Exceção:

1. Se o usuário não informar campos obrigatório no passo 3, o sistema emitirá uma mensagem de aviso informativa e não salvará os registros adicionados/modificados.
2. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.

3. Caso ocorra erros na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Realizar Pedido.

Objetivos: Permitir que usuários com permissão mínima, parcial ou total possam escolher uma comanda e adicionar ou remover vários produtos a ela e incluir adicionais aos produtos escolhidos, caracterizando estas ações em um pedido.

Requisitos: RF012.

Pré-Condições: estar autenticado no sistema e ter permissão mínima, parcial ou total

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Pedidos.
3. O usuário seleciona uma comanda, e tem a opção de adicionar ou remover produtos a ela, assim como informações complementares do pedido.
4. Após o produto ser selecionado, é possível incluir adicionais ao produto específico, clicando no ícone “+”.
5. Ao final do pedido, é necessário clicar no botão salvar para efetivar o pedido realizado.
6. O sistema atualiza a tabela de armazenamento de dados referente ao Pedido.

Fluxo de Exceção:

1. Se o usuário não informar uma comanda para alteração, ou não clicar em salvar após adicionar produtos ao pedido, o sistema emitirá uma mensagem de aviso informativa.
2. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.
3. Caso ocorra erros na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Realizar Pagamento.

Objetivos: Permitir que usuários possam escolher uma ou mais comandas e adicionar ou remover e consultar os produtos e adicionais da comanda selecionada. Permitir que o usuário tenha uma prévia do valor do total do pagamento, e possa adicionar um desconto e acréscimo se necessário, e em seguida realizar o pagamento das comandas, liberando-as para uso de outros clientes.

Requisitos: RF010, RF011.

Pré-Condições: estar autenticado no sistema e ter permissão administrativa mínima, primária ou secundária.

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Pagamentos.
3. O usuário clica no ícone referente à(s) comanda(s) que deseja adicionar, ou digita o número da mesma para realização do pagamento.
4. Ao inserir cada comanda, os produtos e adicionais consumidos ficarão visíveis em um grid possibilitando a conferência do mesmo.
5. Caso houver necessidade de alteração nos produtos listados, é possível clicar na linha do grid que representa o produto em questão, e clicar no botão alterar pedido para realizar alteração de produtos e adicionais.
6. O usuário poderá aplicar um desconto ou um acréscimo no pagamento, assim como conferir o valor total das comandas adicionadas.
7. Ao clicar no botão Realizar Pedido, o sistema atualiza o objeto conforme a ação do usuário.
8. O sistema atualiza a tabela de armazenamento de dados referente ao pagamento.

Fluxo de Exceção:

1. Se o usuário não informar nenhuma comanda, o sistema emitirá uma mensagem de aviso informativa.
2. Caso ocorra erro na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.
3. Caso ocorra erro na atualização da tabela no banco de dados, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Consulta de Lucros.

Objetivos: Permitir que usuários com permissão total possam visualizar pedidos, produtos e adicionais de pagamentos já realizados de forma clara, podendo realizar pesquisas específicas com utilização de filtros de data, por usuário que realizou o pagamento, e até pelo número de identificação de pagamento específico

Requisitos: RF014, RF015, RF016.

Pré-Condições: estar autenticado no sistema e ter permissão total

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Consulta de Lucros.
3. O usuário poderá consultar pagamentos, tendo informações de quantos e quais produtos e adicionais foram consumidos em cada pagamento. O usuário pode optar também, por imprimir um relatório PDF das vendas filtradas em tela, utilizando o botão imprimir.
4. O usuário poderá informar filtros específicos de data, usuário ou número de identificação de pagamento para realizar consultas específicas, ou usar

o botão limpar para voltar a listagem principal de pagamentos, e limpar os filtros selecionados.

Fluxo de Exceção:

1. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.

Nome: Consulta de Pedidos.

Objetivos: Permitir que usuários com permissão mínima, parcial ou total possam visualizar pedidos de produtos e adicionais por comanda de forma clara, na sequência em que os pedidos foram realizados, para que a produção da cozinha seja mais dinâmica. Após a produção de todos os produtos da comanda, é possível clicar no botão Atendido, para que a lista de produção seja atualizada.

Requisitos: RF017.

Pré-Condições: estar autenticado no sistema e ter permissão mínima, parcial ou total

Fluxo Normal:

1. O usuário conecta no sistema.
2. O usuário clica no ícone de Consulta de Pedidos.
3. O usuário poderá ver uma lista das 6 próximas comandas que deverão ser produzidas, e a quanto tempo foi realizado seu pedido.
4. Ao final da produção de cada comanda é necessário clicar no botão “Atendido”, para que a comanda atendida suma da lista, e a tela seja atualizada.
5. O usuário poderá clicar no botão Atualizar, para atualizar a lista de pedidos, caso julgar necessário, porém esta ação é executada a cada 2 minutos pelo sistema, para não haver necessidade de atualização manual.

Fluxo de Exceção:

1. Caso ocorra erros na atualização do objeto, o sistema emitirá uma mensagem informando o erro gerado.

4.2. Diagrama de atividades

Nesta sessão estão evidenciados os diagramas de atividades correspondentes ao sistema desenvolvido neste trabalho.

O diagrama de atividades é um diagrama UML utilizado para modelar o aspecto comportamental de processos.[...] Neste diagrama, uma atividade é modelada como uma sequência estruturada de ações, controladas potencialmente por nós de decisão e sincronismo (GUDWIN, 2018).

A Figura 4 representa o diagrama de atividade da opção de menu Produtos. Nessa função é possível adicionar, editar, inativar, consultar ou imprimir relatório de

produtos ativos do sistema. Esse diagrama de atividades representa a atividades de todas as funções de gerenciamento de dados: produtos, comandas, adicionais e usuários.

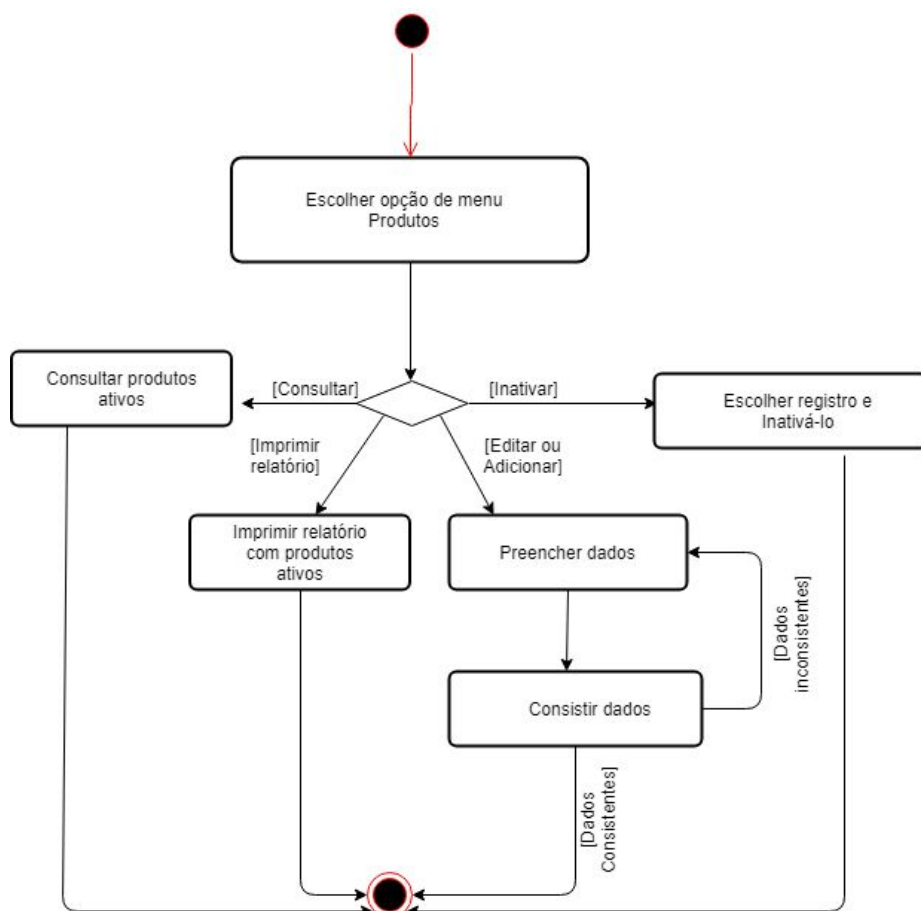


Figura 4. Diagrama de atividade correspondente à opção de menu produtos. Fonte: Elaborado pela autora.

As Figuras 5 e 6 representam os diagramas de atividades das duas principais telas do sistema: realizar pedidos e realizar pagamentos. Na função realizar pedidos é necessário informar uma comanda, escolher produtos e adicionar à comanda, e se houver necessidade, incluir adicionais a esses produtos. Na função pagamento, é possível escolher uma ou mais comandas para realização de pagamentos, conferir produtos e adicionais das comandas, se necessário, realizar alterações nos pedidos e realizar o pagamento.

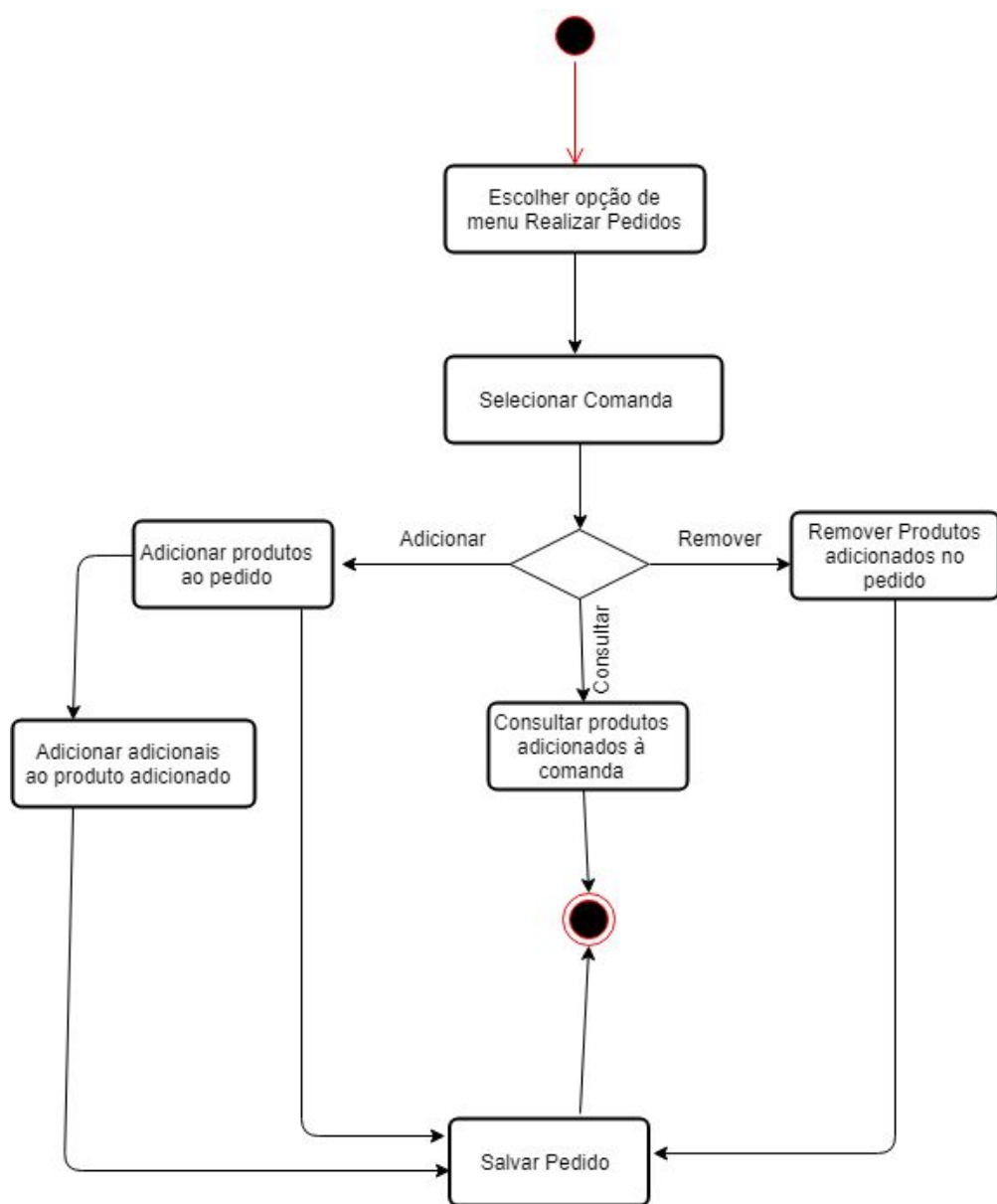


Figura 5. Diagrama de atividade correspondente à opção de menu realizar pedido.
Fonte: Elaborado pela autora.

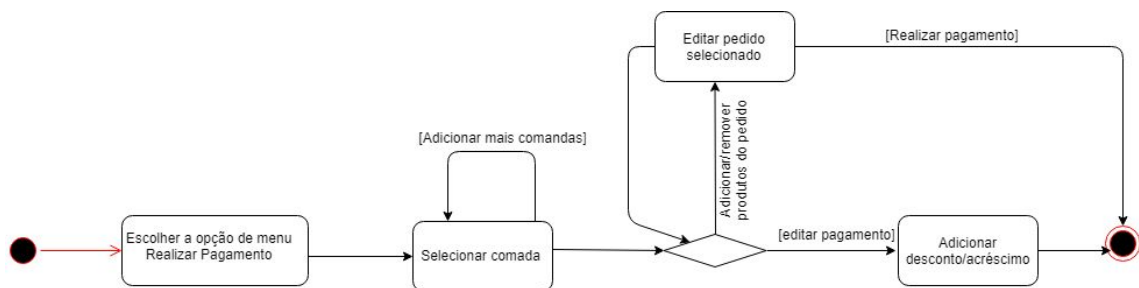


Figura 6. Diagrama de atividade correspondente à opção de menu realizar pagamento.
Fonte: Elaborado pela autora.

4.3. Diagrama de classes

O diagrama de classes representa a estrutura de classes do sistema utilizada para manipulação dos objetos. O diagrama de classes desse projeto está na Figura 7.

O Diagramas de classes estão entre os tipos mais úteis de diagramas UML pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos (LUCIDCHART, 2019).

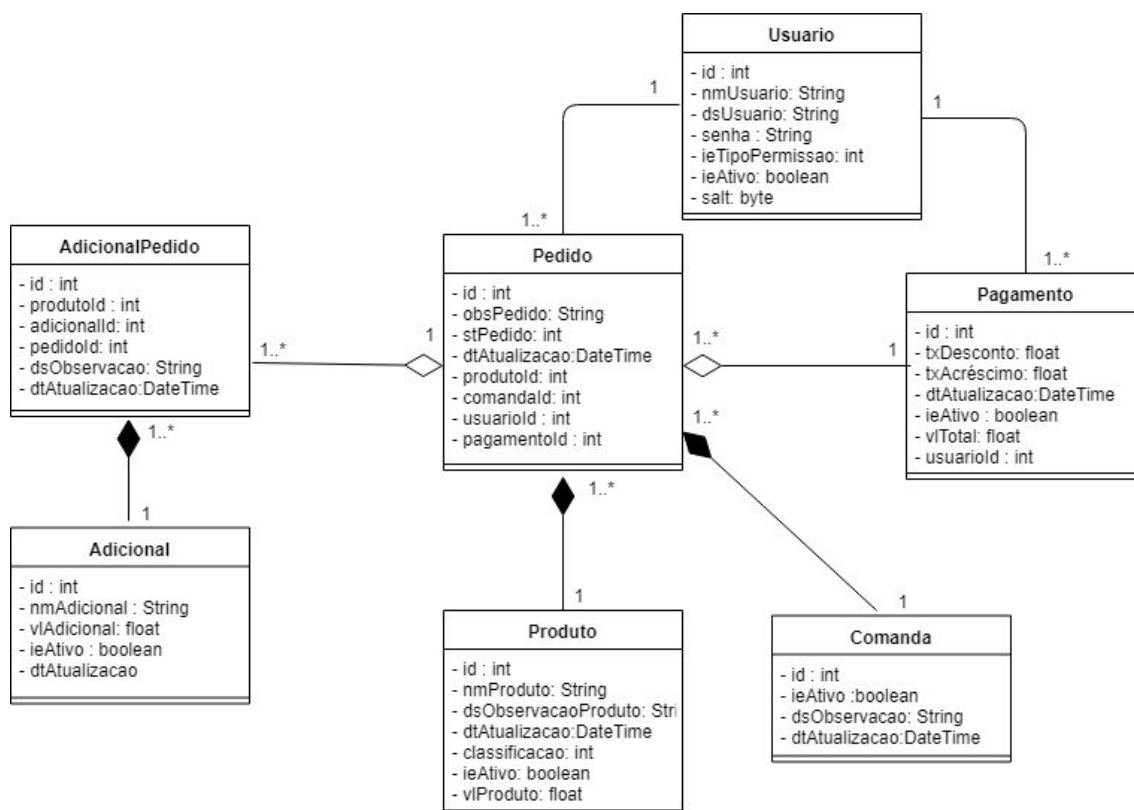


Figura 7. Diagrama de classes. Fonte: Elaborado pela autora.

5. Modelagem de Dados

As tabelas, atributos e entidades utilizadas para armazenamento de dados foram projetadas e criadas manualmente pela autora. Na Figura 8 está a representação do Modelo de entidades relacionais (MER).

- **PRODUTO**: Tabela responsável pelo armazenamento de dados dos produtos cadastrados pelos usuários. Os campos desta tabela são:

- ID: Identificador interno do produto;
 - NMPRODUTO: Responsável por armazenar o nome do produto;
 - DSOBSERVACAOPRODUTO: Armazena uma pequena observação de no máximo 255 caracteres do produto: pode ser informações de quais ingredientes compõem o produto;
 - CLASSIFICACAO: Define qual é a classificação do produto, como por exemplo, se o mesmo é uma sobremesa, bebida, prato quente e etc;
 - IEATIVO: Campo que define se o produto está ativo no sistema;
 - VLPRODUTO: guarda a informação de quanto (em R\$) esse produto está custando no momento;
- COMANDA: Tabela responsável por armazenar informações das comandas entregue aos clientes quando estes entrarem no estabelecimento. Os campos desta tabela são:
 - ID: identificador da comanda;
 - IEATIVO: Campo que define se a comanda está ativa no sistema;
 - DSOBSERVACAO: Pequena observação de no máximo 250 caracteres que pode ser adicionada na comanda;
- USUARIO: tabela responsável pelo controle de acesso dos usuários no sistema. Seus campos são:
 - ID: Identificador do usuário;
 - NMUSUARIO: Nome do usuário;
 - DSUSUARIO: Login do usuário;
 - DSENHA: senha do usuário criptografada com utilização de salt;
 - IEATIVO: campo que define se o usuário está ativo no sistema;
 - IETIPOPERMISSAO: Campo que armazena o tipo de permissão do usuário: total, parcial ou mínima;
 - SALT: campo necessário para criptografia de senhas;
- PAGAMENTO: Tabela responsável por guardar informações do pagamento dos pedidos realizados. Seus campos são:
 - ID: Identificador do Pagamento;
 - VLTOTAL: Valor total pago pelo cliente, sem desconto e acréscimo;
 - TXDESNCONTO: Responsável por armazenar a porcentagem de desconto dada em cima do valor final do pagamento;
 - TXACRESCIMO: Responsável por armazenar a porcentagem de acréscimo dado em cima do valor final do pagamento;
 - IEATIVO: Campo que define se o pagamento está ativo no sistema;
 - USUARIO_ID: Identificador do usuário responsável por realizar o pagamento do cliente;
- ADICIONAL: tabela responsável pelo armazenamento de dados de adicionais.

Seus campos são:

- ID: Identificador do adicional;
 - NMADICIONAL: Nome do adicional;
 - VLADICIONAL: Valor do adicional (em R\$) que será somado ao valor do produto;
 - IEATIVO: define se o adicional está ou não ativo no sistema;
-
- ADICIONAL_PEDIDO: tabela de ligação entre o pedido, o produto e os adicionais escolhidos para aquele produto e pedido.
 - ID: Identificador do produto_pedido, de uso interno;
 - PRODUTO_ID: Identificador do produto;
 - ADICIONAL_ID: Identificador do adicional;
 - PEDIDO_ID: Identificador do pedido que será acrescentado o adicional;
 - DSOBSERVACAO: Pequena observação de no máximo 400 caracteres que pode ser adicionada ao acrescentar um adicional em um produto de um pedido;
-
- PEDIDO: tabela responsável por guardar informações de pedidos de clientes, utilizando a comanda do cliente e o(s) produto(s) pedido(s). Seus campos são:
 - ID : Identificador do pedido;
 - OBSPEDIDO: Pequena observação de no máximo 200 caracteres que pode ser utilizada para informar particularidades do pedido;
 - DTATUALIZACAOPEDIDO: Campo responsável por armazenar data e hora que o pedido foi realizado. Com essa informação, o pedido entrará em uma fila de pedidos realizados para ser atendido na cozinha;
 - STPEDIDO: Status que o pedido se encontra no momento: se já foi realizado, se foi atendido, ou se já foi pago;
 - PRODUTO_ID: Identificador do produto pedido;
 - COMANDA_ID: Identificador da comanda utilizada pelo cliente;
 - USUARIO_ID: Identificador do usuário responsável por realizar o pedido do cliente;
 - PAGAMENTO_ID: Identificador do pagamento realizado pelo cliente;

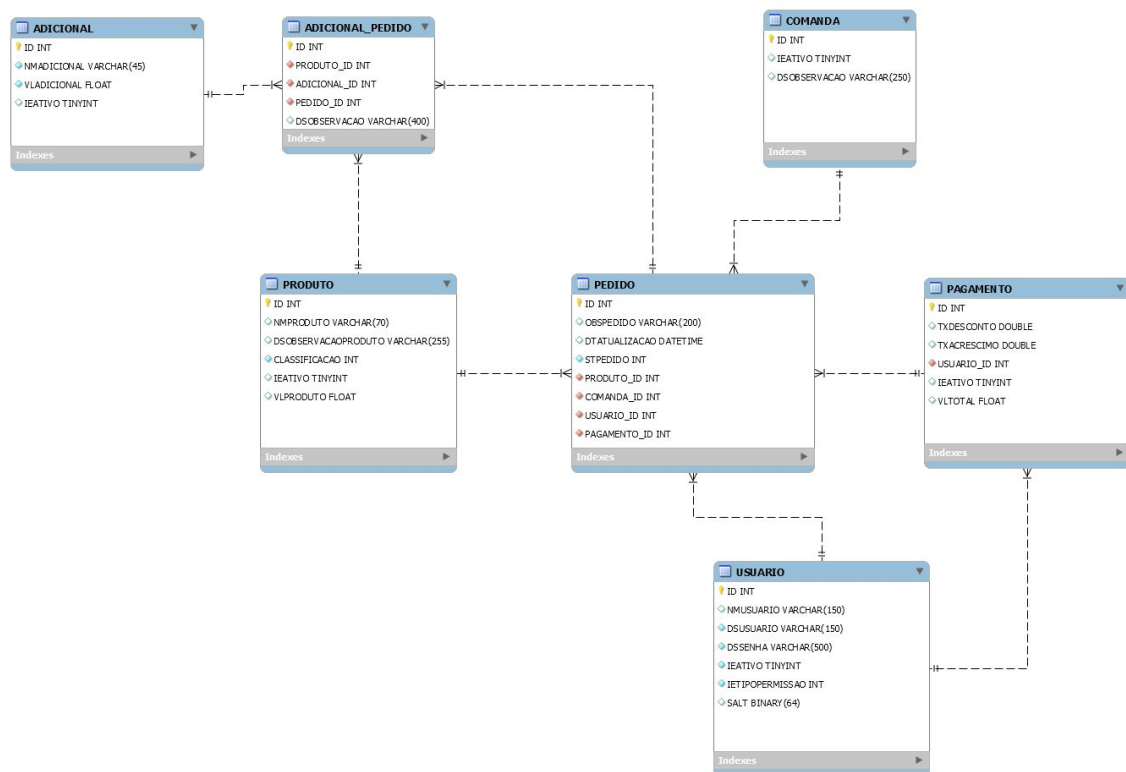


Figura 8. MER. Fonte: Elaborado pela autora.

6. Projeto da Interface

Uma interface pode ser definida como a parte visual do sistema responsável pela interação dos usuários com o software para a entrada de dados e saída de informação. A interface desse sistema foi pensada para ser de fácil acesso e de entendimento do usuário, sem gerar ambiguidade, com cores contrastantes em suas fontes facilitando o uso por várias horas. Todas as telas são padronizadas e limpas, e podem ser identificadas pelo seu ícone quando as mesmas forem maximizadas ou minimizadas.

A Figura 9 representa a primeira tela do sistema, onde o usuário deverá informar Login e Senha para acesso ao software. Para o primeiro acesso, o usuário deverá informar o usuário ADMIN e senha ADMIN, e logo em seguida abrirá a tela de Gerenciamento de usuários para cadastrar um novo usuário e senha.

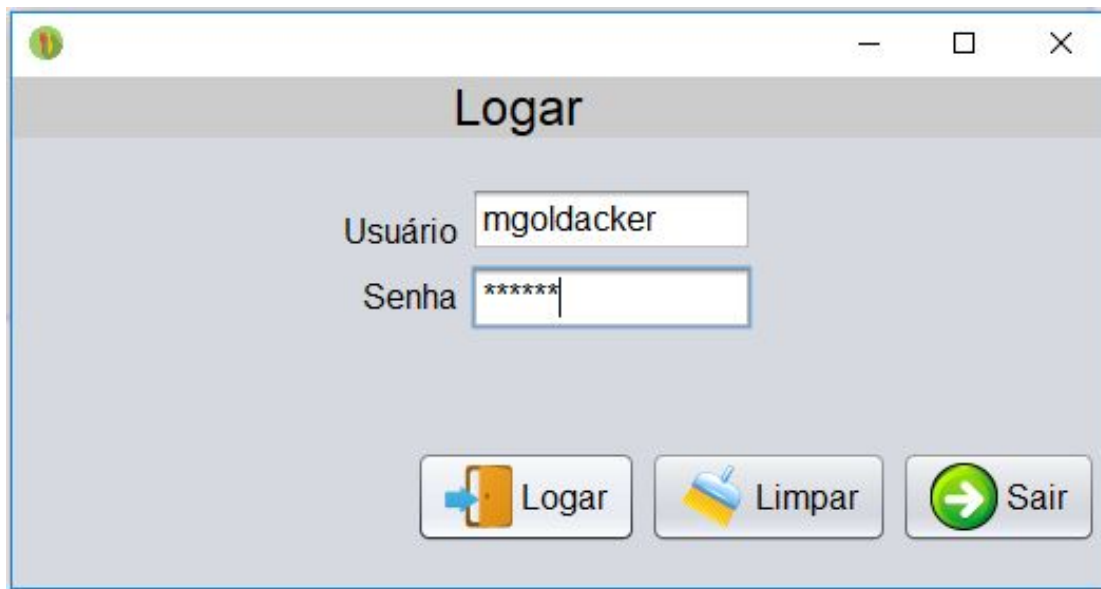


Figura 9. Projeto de interface para login no sistema. Fonte: Elaborado pela autora.

A Figura 10 ilustra a tela principal do sistema, o menu mostra os ícones de todas as telas que o usuário logado possui permissão de acessar. O usuário logado nesse exemplo possui permissão total.

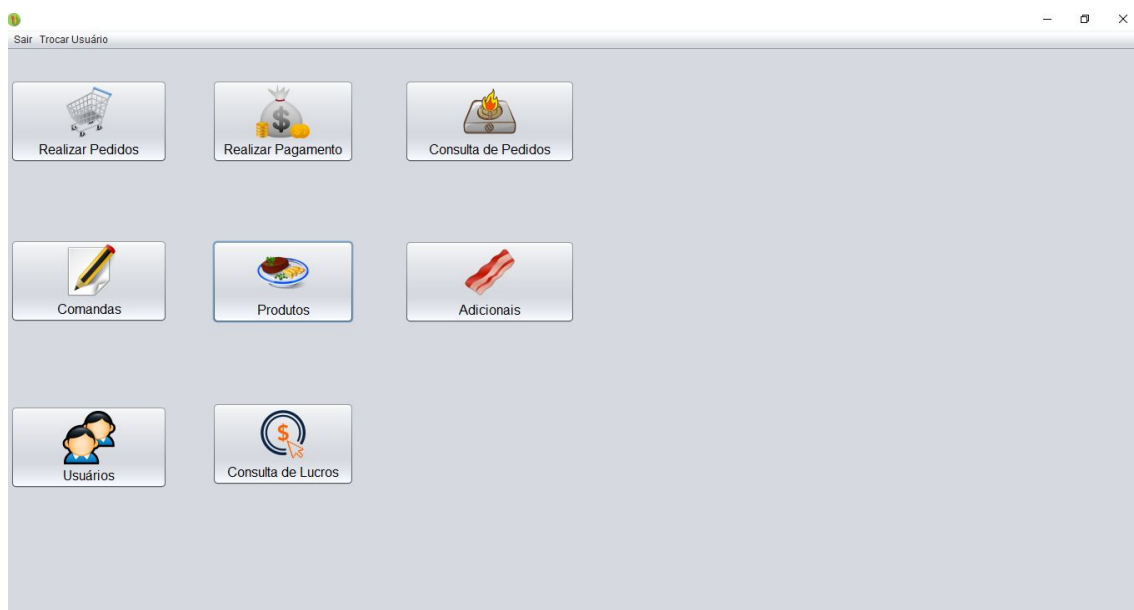


Figura 10. Projeto de interface do menu principal. Fonte: Elaborado pela autora.

A Figura 11 representa a função gerenciamento de usuários. Nesta função é possível consultar, cadastrar, editar, inativar ou realizar impressão de relatórios PDF de usuários ativos no sistema. É nesta função que é definido o nível de permissão do usuário cadastrado/alterado. De acordo com o nível da permissão concedida (total, parcial ou mínima), o usuário poderá ou não ter acesso à determinadas funções.

Nome	Usuário	Tipo de permissão
Bruno Horn	bhorn	Mínima
Maiara Goldacker	mgoldacker	Total
Larissa Nunes	LNunes	Parcial
Carlos Theis	ctheis	Mínima

Nome:

Usuário: Senha:

Tipo de permissão:

Figura 11. Projeto de interface do gerenciamento de usuários. Fonte: Elaborado pela autora.

A Figura 12 representa a função de gerenciamento de comandas. Nesta função é possível consultar, cadastrar, editar, inativar ou realizar impressão de relatórios PDF de comandas ativas no sistema.

Código	Observação
1	
2	
3	
4	
5	
6	
7	
8	
9	

Comanda:

Observação:

Figura 12. Projeto de interface do gerenciamento de comandas. Fonte: Elaborado pela autora.

A Figura 13 representa a função de gerenciamento de produtos. Nesta função é possível consultar, cadastrar, editar, inativar ou imprimir relatórios PDF contendo informações de produtos ativos no sistema. Nesta função será possível definir quais produtos serão vendidos no estabelecimento, qual será a composição deles, e adicionar um valor para cada produto.

Produto	Valor	Observação	Classificação
Costela Burger	R\$ 30,00	Acompanha Cheddar	Hamburguer
Costela Bacon e Cheddar	R\$ 32,00		Hamburguer
Costela Onion	R\$ 28,00	Acompanha Onion Rings	Hamburguer
Coca-cola 600ml	R\$ 6,00		Bebida
Guaraná 600ml	R\$ 5,00		Bebida
Suco de Laranja	R\$ 7,50		Bebida
Trufa de chocolate	R\$ 3,00		Sobremesa
Trufa de café	R\$ 2,50		Sobremesa
Mousse de maracujá	R\$ 10,00		Sobremesa

Nome: Mousse de maracujá Valor: 10,00

Classificação: Sobremesa

Observação:

Imprimir Salvar Limpar Alterar Excluir Sair

Figura 13. Projeto de interface do gerenciamento de produtos. Fonte: Elaborado pela autora.

A Figura 14 representa a função de gerenciamento de adicionais. Nesta função é possível consultar, cadastrar, editar, inativar ou realizar impressão de relatórios PDF com dados de adicionais ativos no sistema. Nesta função será possível definir quais adicionais poderão ser incluídos em um produto e qual será o valor acrescentado ao adicioná-lo.

Nome	Valor do adicional
Bacon	R\$ 2,00
Cebola caramelizada	R\$ 2,50

Nome do adicional: Cebola caramelizada Valor: 2,50

Imprimir Salvar Limpar Alterar Excluir Sair

Figura 14. Projeto de interface do gerenciamento de adicionais. Fonte: Elaborado pela autora.

A Figura 15 representa a função Realizar Pedidos. Nesta função é possível selecionar uma comanda já cadastrada e disponível para uso no sistema e adicionar ou remover produtos a ela. Além disso é possível adicionar uma observação sobre o produto escolhido, como por exemplo, especificar que o mesmo será sem salada. Após escolher o produto, é possível adicionar vários adicionais no botão de '+'. A figura 16 mostra o processo de seleção dos adicionais. Após adicioná-los, é possível conferir a descrição dos adicionais selecionados na própria tela de realizar pedidos, ao clicar no produto correspondente ao mesmo.

Após os produtos e adicionais serem selecionados, o usuário deve clicar no botão de salvar, para que o pedido realizado entre na fila de espera da cozinha para ser produzido. Caso o usuário não clique em salvar, ao tentar fechar a tela, selecionar outra comanda, ou limpar a tela, o sistema emitirá uma mensagem de aviso informando que o pedido ainda não foi salvo.

Realizar Pedido

Selecionar Comanda

Comanda 2

Selecionar Produtos

Classificação: Sem filtro

A	B
Costela Burger	R\$ 30,00
Costela Bacon e Cheddar	R\$ 32,00
Costela Onion	R\$ 28,00
Coca-cola 600ml	R\$ 6,00
Guaraná 600ml	R\$ 5,00
Suco de Laranja	R\$ 7,50
Trufa de chocolate	R\$ 3,00
Trufa de café	R\$ 2,50
Mousse de maracujá	R\$ 10,00

Observação

Produtos pedidos

Produto	Valor	Observação
Costela Burger	R\$ 30,00	sem cebola
Trufa de café	R\$ 2,50	
Costela Burger	R\$ 30,00	

Adicionais selecionados

Bacon

Salvar Limpar Sair

Figura 15. Projeto de interface para realizar pedido. Fonte: Elaborado pela autora.



Figura 16. Projeto de interface realizar pedido ao incluir adicionais a um produto selecionado. Fonte: Elaborado pela autora.

As Figuras 17 e 18 representam a função Realizar Pagamento. Nesta função é possível selecionar uma ou mais comandas. Ao ser adicionadas os produtos que foram pedidos nas comandas adicionadas serão listados para conferência. Nesta função, ao clicar em um determinado produto do grid de produtos selecionados, é possível abrir a tela de Realizar Pedido (conforme Figura 18), e alterar os pedidos da comanda: esta funcionalidade servirá caso um produto não tenha sido adicionado, ou adicionado a mais, e o usuário poderá alterá-lo, removê-lo da comanda.

O usuário poderá também adicionar um desconto ou um acréscimo ao valor total, e conferir os adicionais de cada produto na descrição dos adicionais, ou clicando no botão '+’.

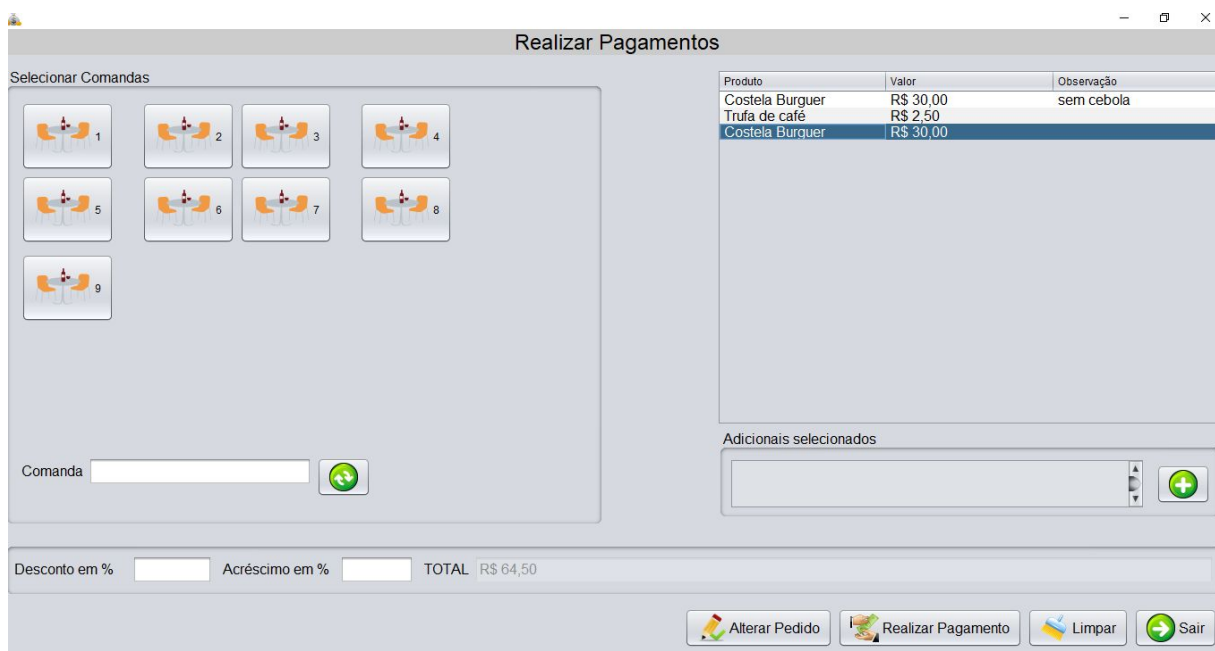


Figura 17. Projeto de interface para realizar pagamento. Fonte: Elaborado pela autora.

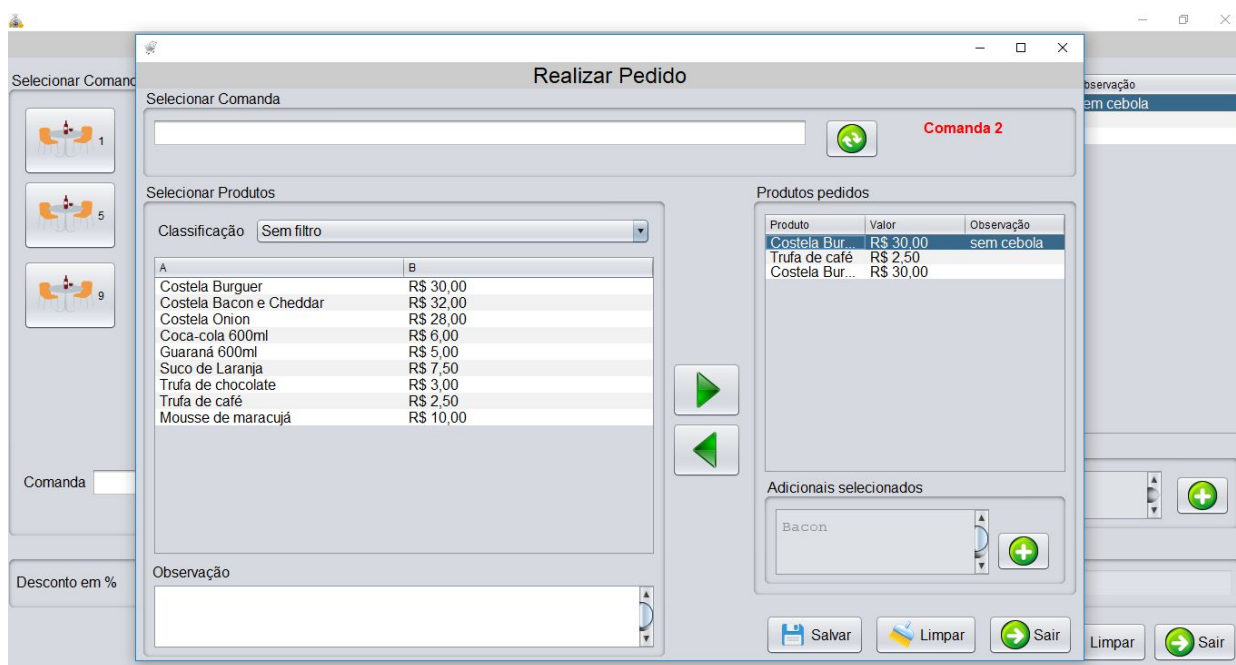


Figura 18. Projeto de interface para alteração de pedido ao realizar pagamento. Fonte: Elaborado pela autora.

A Figura 19 representa o projeto de interface para a consulta de pedidos realizados pelos usuários. Essa interface mostrará os pedidos que devem ser atendidos em sequência pela cozinha. Cada pedido listará os produtos e adicionais pedidos, e as observações adicionadas na realização do pedido. Essa tela será atualizada a cada 2 minutos, pois verificou-se que seria um bom tempo para a preparação de pratos, e como no momento não foi desenvolvido nenhum parâmetro para essa informação, foi decidido

deixar 2 minutos de forma fixa. Quando um pedido estiver pronto, o usuário deve clicar no botão “atendido” correspondente a comanda atendida, para que a lista se atualize com novos pedidos. Nenhuma tecla de atalho foi implementada para esta operação até o momento. Caso o pedido demore mais de uma hora para ser atendido, a informação do tempo que o pedido foi realizado, passará a ficar vermelha. Caso houver mais de 6 pedidos realizados, eles ficarão em lista de espera, e aparecerão nos painéis quando os primeiros 6 pedidos forem atendidos.

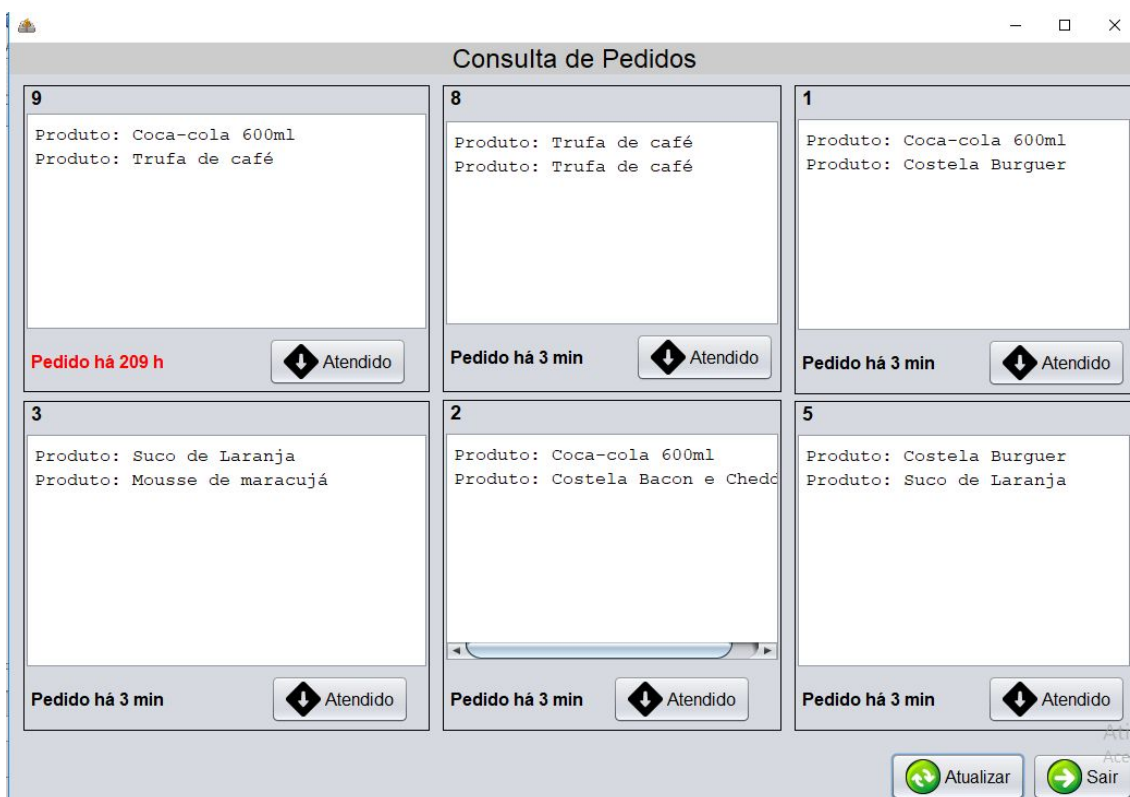



Figura 19. Projeto de interface de consulta de pedidos. Fonte: Elaborado pela autora.

A Figura 20 representa o projeto de interface da consulta de pagamentos. Usando esta interface, o usuário com permissão total poderá consultar informações de pagamentos, filtrando por pagamentos específicos, data ou por usuários que realizaram pagamentos. Ao clicar em um pagamento específico, o grid de produtos carregará os produtos que foram consumidos neste pagamento em específico, e ao clicar em um produto, se houver adicionais pedidos para o produto, carregará sua descrição em tela, ou o usuário poderá consultá-los no botão ‘+’. O usuário também saberá o lucro obtido por período/usuário/pagamento específico filtrado.

Consulta de Pagamento

Filtros

De: 01/01/2019 Até: 10/02/2019 Usuário Pagamento: Sem filtro ID:



Pagamentos

Pagamento	Acréscimo	Desconto	Valor Total Pago
1	0.0	0.0	R\$ 12,50
2	0.0	0.0	R\$ 67,00
3	0.0	0.0	R\$ 36,00

Produto	Valor	Observação
Trufa de café	R\$ 2,50	
Mousse de maracujá	R\$ 10,00	

Adicionais selecionados:

Total de vendas no período: R\$115,50






Figura 20. Projeto de interface de consulta de pagamentos. Fonte: Elaborado pela autora.

As Figuras 21, 22, 23, 24 e 25 mostram os relatórios desenvolvidos para a aplicação. Eles listam todos os atributos ativos mostrados na consulta de suas respectivas telas. O relatório de lucro (Figura 21), obedece os filtros adicionados na tela de consulta de lucros, pela qual pode ser emitido.



REL01 - RELATÓRIO DE GESTÃO DE VENDAS

nº Pagamento	Desconto	Acréscimo	Tot. Venda Bruta	Tot. Venda Liquida
4	10.0	0.0	R\$ 45,50	R\$ 40,95
Produto			Valor do Produto	
Mousse de maracujá			R\$ 10,00	
Suco de Laranja			R\$ 7,50	
Costela Onion			R\$ 28,00	

nº Pagamento	Desconto	Acréscimo	Tot. Venda Bruta	Tot. Venda Liquida
5	0.0	0.0	R\$ 63,00	R\$ 63,00
Produto			Valor do Produto	
Costela Burguer			R\$ 30,00	
Trufa de café			R\$ 2,50	

Figura 21. Relatório de gestão de vendas. Fonte: Elaborado pela autora.



REL02 - RELATÓRIO DE USUÁRIOS ATIVOS

nº Usuário	Usuário	Nome	Tipo de permissão
------------	---------	------	-------------------

36	admin	mgolda	Total
----	-------	--------	-------

Figura 22. Relatório de usuários ativos. Fonte: Elaborado pela autora.



REL03 - RELATÓRIO DE PRODUTOS ATIVOS

nº Produto	Produto	Observação	Valor de venda
------------	---------	------------	----------------

9	Coca-cola 600ml		R\$ 6,00
---	-----------------	--	----------

7	Costela Bacon e Cheddar		R\$ 32,00
---	-------------------------	--	-----------

6	Costela Burguer	Acompanha Cheddar	R\$ 30,00
---	-----------------	-------------------	-----------

8	Costela Onion	Acompanha Onion Rings	R\$ 28,00
---	---------------	-----------------------	-----------

10	Guaraná 600ml		R\$ 5,00
----	---------------	--	----------


14	Mousse de maracujá		R\$ 10,00
----	--------------------	--	-----------

11	Suco de Laranja		R\$ 7,50
----	-----------------	--	----------

13	Trufa de café		R\$ 2,50
----	---------------	--	----------

12	Trufa de chocolate		R\$ 3,00
----	--------------------	--	----------

Figura 23. Relatório de produtos ativos. Fonte: Elaborado pela autora.



REL04 - RELATÓRIO DE ADICIONAIS ATIVOS

nº Adicional	Adicional	Valor de Venda
6	Bacon	R\$ 2,00
7	Cebola caramelizada	R\$ 2,50

Figura 24. Relatório de adicionais ativos. Fonte: Elaborado pela autora.



REL05 - RELATÓRIO DE COMANDAS ATIVOS

nº Comanda	Observação
1	
2	
3	
4	
5	
6	
7	
8	

Figura 25. Relatório de comandas ativas. Fonte: Elaborado pela autora.

7. Implementação

O código fonte do sistema desenvolvido pode ser visto em <<https://github.com/MaiaraGoldacker/Restaurante>>, todas as figuras e relatórios deste artigo podem ser encontrados em <<https://github.com/MaiaraGoldacker/Restaurante/tree/master/src/Imagens>>.

A primeira etapa da implementação desse sistema foi a criação do MER. Em seguida, com o banco de dados já criado, implementou-se as classes, mapeando o MER para o Java. Logo foram desenvolvidos protótipos de telas, e foram realizados testes

funcionais, para verificação e validação das classes.

Após os testes funcionais, foram criadas efetivamente as interfaces utilizadas no sistema, e em seguida foram ajustadas as regras da aplicação em fontes separados, para melhor organização do código, caracterizando o modelo MVC. As classes foram divididas por pacotes, agrupando-as por regras, comunicação com banco de dados (DAO), classes entre outros, conforme mostrado na Figura 26.

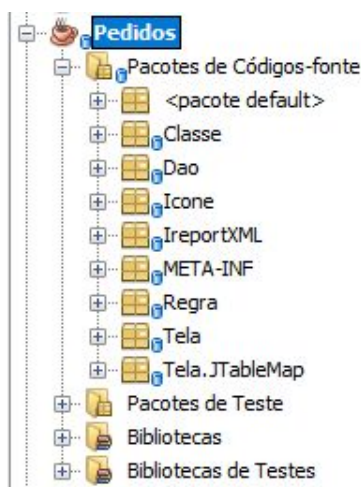


Figura 26. Estrutura de projeto do sistema no Netbeans. Fonte: Elaborado pela autora.

Os relatórios foram desenvolvidos por último, com a ferramenta iReport 5.6.0, que após serem desenhados, são guardados como arquivo .XML, e geram um arquivo .jasper que são utilizados na apresentação do PDF.

Houve preocupação com a criptografia de senhas, por ser um dado sensível, de uso pessoal. Foi utilizada uma solução com hash e salt, evitando que ataques de dicionário e de rainbow table possam prosperar. Essa solução transforma uma senha comum em caracteres randômicos, que não podem ser descriptografados.

Ao fim da implementação foi realizada a última etapa de testes manuais, que consistiu em:

- Testes funcionais: validação de métodos implementados. Verificados se cada método estava realizando o que deveria ser feito.
- Testes unitários: realização de testes isolados por função.

8. Resultados Obtidos

A solução desenvolvida neste trabalho, apesar de possuir menos funcionalidades em comparação aos trabalhos correlatos, apresenta vantagens como aplicação gratuita, código aberto permitindo que o software possa se desenvolver futuramente. Há também possibilidade de restrição ao acesso de função, interface e usabilidade simplificados.

Todos os requisitos levantados foram atingidos, desde o controle de acesso de usuários, a realização de pedidos, o monitoramento de pedidos para a cozinha, e o acompanhamento de lucros obtidos.

9. Considerações Finais e Trabalhos Futuros

O sistema cumpre o prometido para solução de software: oferece os módulos necessários para cadastros e realiza solução para pedidos e pagamentos, proporcionando maior controle na demanda, ajudando um possível cliente contratante a tomar decisões para o seu negócio.

Durante o desenvolvimento do trabalho, foram encontradas algumas dificuldades, como o levantamento de requisitos, já que não havia uma pessoa que trabalhava diretamente em um estabelecimento alimentício e que necessitava de uma solução de sistema para ser consultada, então os requisitos surgiram com base em pesquisas do autor. A realização de testes abrangentes foi uma dificuldade, já que a implementação e testes foram realizados pelo autor, tornando-os viciados, ou seja, é comum que o desenvolvedor acabe realizando apenas testes válidos, e esquecendo dos inválidos. Outra dificuldade foi o uso de soluções não vistas em curso, já que toda a solução não vista, necessitava de pesquisas para verificar como deveriam ser utilizadas, antes de sua implementação, como por exemplo, o uso do hash e salt para codificação de senhas, o uso git e IReport, entre outras soluções.

Como trabalhos correlatos para desenvolvimento futuro, propomos uma aplicação Java web renderizável, para que seja possível ser acessada em qualquer computador sem instalação, além de ser possível acessar em dispositivos móveis.

Também seria interessante listar a possibilidade de criação de um instalador, para que o cliente não precise cumprir pré-requisitos para utilização do sistema, como por exemplo, baixar manualmente o Java8, já que muitos clientes não possuem conhecimento sobre extensões e APIs, dificultando a contratação do software. Outro ponto interessante seria a melhora dos relatórios, colocando informações do usuário responsável pela emissão do relatório, e a data e hora que este foi emitido, além da possibilidade de escolher filtros na hora da emissão. Futuramente, caso o sistema seja pago, será muito importante a integração do sistema com periféricos, como TEF (Transferência Eletrônica de Fundos) e balanças.

10. Referências Bibliográficas

CONSUMER. **Programa para restaurantes, lanchonetes, pizzarias e similares.** Disponível em: <<http://www.programaconsumer.com.br/oprograma>> Acesso em 23 fev. 2019.

DINO. **Expansão da automação para restaurantes e aposta para 2018.** Disponível em: <<https://exame.abril.com.br/negocios/dino/expansao-da-automacao-para-restaurantes-e-aposta-para-2018-confira-tendencias/>> Acesso em 23 fev. 2019.

FERREIRA, Alexandre Aquiles. **Conheça a nova api de datas do java8.** Disponível em: <<http://blog.caelum.com.br/conheca-a-nova-api-de-datas-do-java-8/>> Acesso em 23 fev. 2019

LANHELLAS, Ronaldo. **JasperReport: Relatórios em java com IReport.** Disponível em: <<https://www.devmedia.com.br/jasperreport-relatorios-em-java-com-ireport/31075>> Acesso em 23 fev. 2019.

Lei nº 8.078 de 11 de Setembro de 1990. **Código de Proteção e defesa do consumidor.** Governo do estado de São Paulo. Maio de 2018.

LINUX. **Automação comercial com custos reduzidos para facilitar a gestão do seu negócio.** Disponível em:

<<http://www.linx.com.br/servicos-sofware>> Acesso em 04 dez. 2018.

LUCIDCHART. **O que é diagrama de classe uml.** Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml?a=0>> . Acesso em 23 fev. 2019.

MEDEIROS, Higor. **Introdução ao Padrão MVC.** Disponível em:

<<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>> Acesso em 03 dez. 2018.

MEDEIROS, Higor. **Introdução aos Processos de Software e o Modelo Incremental Evolucionário.** Disponível em:

<<https://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>>. Acesso em 04 mar. 2019.

MYSQL. MySql The world's most popular open source database. Disponível em:

<<https://www.mysql.com/products/workbench/>>. Acesso em 01 dez. 2018.

NETBEANS. Building a desktop application on top of the NetBeans Platform can save years of development time. Disponível em:

<<https://netbeans.org/features/platform/index.html>> Acesso em 01 dez. 2018.

SAUVÉ, Dr. Jacques Philippe. Document View. Disponível em:
<<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/documentviews/relatorios.htm>>. Acesso em 23 fev. 2019.

SILVA, Izalmo Primo da. Desenvolvimento com Hibernate. Disponível em:
<<https://www.devmedia.com.br/desenvolvendo-com-hibernate/14756>> Acesso em 06 fev. 2019.

SILVEIRA, Débora Priscila. O que é e como funciona o git e github. Disponível em :
<<https://www.oficinadanet.com.br/post/16111-o-que-e-e-como-funciona-o-git-e-github>>. Acesso em 23 fev. 2019.

SOUZA, Marcio Ballem de. Padrão MVC - Java Magazine. Disponível em:

<<https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>> Acesso em 02 dez. 2018.