UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

DOUGLAS ALVES DA SILVA

SISTEMA WEB *DELIVERY* PARA PADARIAS

Alegre-ES

2020

SISTEMA WEB *DELIVERY* PARA PADARIAS

Trabalho de Conclusão de Curso requisito parcial do Bacharelado em Sistemas de Informação da Universidade Federal do Espírito Santo.

Professor Orientador: Giuliano Prado de Morais Giglio

DOUGLAS ALVES DA SILVA

SISTEMA WEB *DELIVERY* PARA PADARIAS

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 09 de dezembro de 2020

BANCA EXAMINADORA

_____

Giuliano Prado de Morais Giglio - Universidade Federal do Espírito Santo - UFES

Professor orientador

_____

Antonio Almeida de Barros Junior - Universidade Federal do Espírito Santo - UFES

Examinador

_____

Marcelo Otone Aguiar - Universidade Federal do Espírito Santo - UFES

Examinador

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter sido fiel para terminar a boa obra que começou em minha vida.

Gratidão pelos meus pais, pela confiança no meu progresso que serviram de alicerce. Este presente trabalho é a prova de que os esforços deles pela minha educação não foram em vão e valeram a pena.

Agradeço ao meu orientador Giuliano Prado pela paciência e dedicação durante o projeto que serviram como pilares de sustentação para a conclusão deste trabalho.

Honro o fechamento deste ciclo dedicando aos meus colegas de curso David, Jessé, Francisco, Luiz e Giseli que sempre me apoiaram durante essa trajetória.

Douglas Alves da Silva

Resumo

Os sistemas de informação não são somente importantes para as empresas de grande porte ou industriais, mas para pequenas e médias empresas. Neste caso, os softwares estão relacionados com a sua sobrevivência no mercado globalizado onde estão inseridas atualmente, contribuindo para sobreviver à competição mercadológica e no amadurecimento estrutural das empresas, na definição de estratégias para um crescimento sólido em si.

Tendo em vista esse cenário, informatizar um sistema de entregas de produtos de consumo por meio de um sistema web é visto como uma oportunidade para empresas do ramo da panificação, partindo do pressuposto em que nesse ramo há pouca informatização empregada. Desta forma, o projeto detalha a construção de um sistema para este fim capaz de se adequar à todas as panificadoras, principalmente aonde o regime remoto de vendas é maior, tal como visto no presente trabalho, quando da implantação em meio a pandemia do Covid-19.

Partindo de um cliente real para o qual foi possível estabelecer o sistema, este projeto foi planejado desde o levantamento de requisitos e estudo do contexto das panificadoras em geral e suas necessidades de informatização de processos inerentes, passando pelo planejamento completo, implementação e validação do produto final, colhendo inclusive resultados do uso do sistema pelo cliente. Sendo assim, o trabalho conseguiu atender aos seus objetivos, sendo bem aceito por seus usuários finais e com resultados gerenciais positivos para a empresa.

Palavras-chave: padaria, delivery, sistema, web, entrega

Abstract

Information systems are not only important for large or industrial companies, but for small and medium-sized companies. In this case, the software is related to its survival in the globalized market where it is currently inserted, contributing to survive the market competition and in the structural maturation of the companies, in the definition of strategies for solid growth in itself.

In view of this scenario, computerizing a consumer product delivery system through a web system is seen as an opportunity for companies in the bakery industry, based on the assumption that there is little computerization employed in this industry. In this way, the project details the construction of a system for this purpose capable of adapting to all bakeries, especially where the remote sales regime is greater, as seen in the present work, when implanted in the middle of the Covid-19 pandemic.

Starting from a real customer for whom it was possible to establish the system, this project was planned from the survey of requirements and study of the context of the bakeries in general and their needs for computerization of inherent processes, through the complete planning, implementation and product validation end, including the results of the use of the system by the customer. Thus, the work managed to meet its objectives, being well accepted by its end users and with positive management results for the company.

Keyword: bakery, delivery, system, web, delivery

LISTA DE FIGURAS

Figura 1.: Fluxo do processo Scrum. 25
Figura 2.: Diagrama das classes de modelo. 26
Figura 3.: Interação das classes de modelo, controladoras e DAO. 27
Figura 4.: Relação entre a Estimativa de Prazo e de Esforço. 37
Figura 5.: Protótipo da Tela da página inicial. 39
Figura 6.: Protótipo da Tela de escolher pedido. 40
Figura 7.: Protótipo da Tela de login do usuário. 40
Figura 8.: Protótipo da Tela de Gerar Relatório de pedidos. 41
Figura 9.: Model View Controller. 42
Figura 10.: Modelo entidade relacionamento. 47
Figura 11.: Tela de Login. 53
Figura 12.: Tela da listagem das categorias. 54
Figura 13.: Tela de produtos de uma categoria selecionada. 54
Figura 14.: Tela da tabela do pedido. 55
Figura 15.: Tela de detalhes de um pedido feito. 56
Figura 16.: Tela da listagem de clientes cadastrados no sistema. 56
Figura 17.: Tela de cadastro de usuário sob a ótica de um dispositivo *mobile*. 57
Figura 18.: Resumo de todos os testes. 60
Figura 19.: Complexidade ciclomática. 61

LISTA DE QUADROS

Quadro 1.: Siglas de nomenclaturas 47
Quadro 2.: Modelo Físico de Dados: Categoria. 48
Quadro 3.: Modelo Físico de Dados: Cliente. 48
Quadro 4.: Modelo Físico de Dados: Cupom. 48
Quadro 5.: Modelo Físico de Dados: LocalEntrega. 49
Quadro 6.: Modelo Físico de Dados: Item. 49
Quadro 7.: Modelo Físico de Dados: Pedido. 50
Quadro 8.: Modelo Físico de Dados: Produto. 50
Quadro 9.: Modelo Físico de Dados: Pagamento. 51

LISTA DE TABELAS

Tabela 1.: Requisitos funcionais. 16
Tabela 2.: Descrição dos Atores. 21
Tabela 3.: Entradas Externas. 29
Tabela 4.: Saídas Externas. 29
Tabela 5.: Consultas Externas. 30
Tabela 6.: Classificação Entradas Externas. 30
Tabela 7.: Classificação Saídas Externas. 31
Tabela 8.: Classificação Consultas Externas. 31
Tabela 9.: Classificação ALI's. 31
Tabela 10.: Tabelas de Pesos FPA. 32
Tabela 11.: Nível de influência. 33
Tabela 12.: Pesos de Influência. 35
Tabela 13.: Fator de Ajuste. 35
Tabela 14.: Pontos Ajustados. 35
Tabela 15.: Esforço de Trabalho. 36
Tabela 16.: Tipo de Sistema. 37
Tabela 17.: Tempo de Desenvolvimento. 38
Tabela 18.: Alocação dos Módulos aos Elementos Arquiteturais. 53
Tabela 19.: Descrição dos erros encontrados. 61

SUMÁRIO

LISTA DE QUADROS 7

1. CONTEXTUALIZAÇÃO 11

1.1. Introdução 11

1.2. Motivação 12

1.3. Objetivo geral 12

1.4. Objetivos específicos 13

1.5. Levantamento preliminar de requisitos 13

1.6. Requisitos Funcionais 15

1.7. Requisitos não-funcionais 16

2. ESPECIFICAÇÃO DE REQUISITOS 17

2.1. Requisitos do Cliente 18

2.1.1. Descrição geral da Aplicação 18

2.2. Requisitos do Software 18

2.2.1. Fronteiras do Sistema 18

2.2.2. Itens de Software 19

2.2.3. Escopo da aplicação 19

2.2.4. Lista de Requisitos Não Funcionais 20

2.2.5. Requisitos Funcionais 20

2.2.5.1. Lista de Requisitos Funcionais 20

2.2.5.2. Descrição dos Atores 21

2.2.5.3. Modelo de Casos de Uso 21

2.2.5.4. Especificação de Caso de Uso 22

3. PROCESSO DE DESENVOLVIMENTO 24

3.1. Metodologia SCRUM 24

3.2. Metodologia de Desenvolvimento 25

3.3. Diagrama de classe 26

3.4. Estimativas (Tamanho, esforço e prazo) 27

3.4.1. Identificação das funcionalidades do sistema 28

3.4.2. Entrada Externa ou Input (I) 28

3.4.3. Saída Externa ou Output (O) 29

3.4.4. Consulta Externa (C) 29

3.4.5. Classificação conforme nível de complexidade 30

3.4.6. Calculando Pontos Brutos 31

3.4.7. Calculando Fator de Ajuste e Pontos Ajustados 32

3.4.8. Fator de Ajuste 35
3.4.9. Pontos Ajustados 35
3.4.10. Estimativa de Esforço de Trabalho 36
3.4.11. Estimativa de prazo em meses 36
3.5. Plano de Testes 38
4. PROTOTIPAGEM 38
5. MODELAGEM DO PROJETO 41
5.1. Arquitetura de Software 41
5.1.1. MVC 42
Figura 9.: Model View Controller. 42
5.2.1 Plataforma Php 43
5.2.2 Autenticação e usuários 43
5.2.3 Acesso à dados 44
5.2.4 JavaScript 44
5.2.4 CSS 45
5.3 Modelagem Física do Banco de Dados 47
5 PROCESSO DE TESTES 51
6 IMPLEMENTAÇÃO E TESTES 51
7.1 Funcionalidades Implementadas 52
7.2. Alocação dos fragmentos de código aos elementos arquiteturais 52
6.3 Implementação das Interfaces 53
6.4 Testes 57
6.4.1 Teste de usabilidade 58
6.4.2 Testes Funcionais 60
6.4.3 Resumo dos Testes 62
8 Considerações finais 62
REFERÊNCIAS BIBLIOGRÁFICAS 66
APÊNDICE 1 – CASOS DE USO 69

1. CONTEXTUALIZAÇÃO

1.1. Introdução

Os sistemas informatizados tem ganhado popularidade exponencial nos últimos anos, e segundo a Associação Brasileira de Empresas de Softwares (ABES), o Brasil ocupa a 9ª posição no ranking de serviços de Tecnologia da Informação do mundo e os investimentos na área tem crescido cada vez mais. Segundo projeções feitas em 2017, os investimentos mundiais em TI somariam US$ 3,8 trilhões em 2019, alta de 3,2% em relação a 2018 segundo o Gartner (2018) e com isso é visto a necessidade de tê-los, seja na vida pessoal ou no ramo empresarial.

De uma maneira geral, sistemas informatizados em padarias não é muito utilizado e embora se tenham muitos softwares no mercado do tipo, não está ao alcance de panificadoras pequenas que visam atender a população local, visto que se tratando de receita e de clientela os números são baixos, e de acordo com dados do SEBRAE (2013), 52% das padarias estão enquadradas como Microempresa, e os outros 48% são empresas de pequeno porte, e até culturalmente pequenas empresas de qualquer ramo não pensam em sistemas para seu negócio enquanto estão enquadradas como comércio de pequeno porte. O panorama da pequena empresa no Brasil, sob o aspecto de tecnologias de informação, nunca foi dos melhores. Uma pesquisa da Microsoft, citada por Silva (1997), indica que 58% das pequenas empresas na época acreditaram que a informatização e uso de sistemas seriam desnecessários no final da década de 90. E uma pesquisa especificamente sobre padarias e mais recente da SEBRAE (2017), mostra dados sobre a pretensão dos investimentos desse segmento, e apenas 8% das padarias responderam “introduzir inovações tecnológicas”.

Mas conforme os fatores cliente/receita vão aumentando, a visão de necessidade de algum sistema para o negócio, cresce, pois é nessa oportunidade que se vê uma maior organização e consequentemente maior retorno financeiro. O sistema web auxiliará a padarias no delivery de pedidos, partindo do princípio que será feito online por meio do website, possibilitando registros de pedidos, pedidos em tempo real e uma maior satisfação do cliente.

Segundo a cartilha do SEBRAE (2017), a “venda direta ou em domicílio” é uma tendência para próximos anos, ou seja, ocorrerá uma grande oportunidade de criar um diferencial e aumentar o faturamento neste segmento.

1.2. Motivação

Atualmente, em empresas de panificação os pedidos são realizados manualmente, sem uso de sistemas informatizados sujeitando assim a perda de dados por parte do administrador do processo. Estabelecendo um processo para os pedidos, será retirado muitas preocupações existentes relacionadas à histórico e pagamentos será reduzida. Para essas empresas, espera-se facilitar este processo com o uso da ferramenta web. A motivação inicial foram essas empresas, mas visto o foco em uma maior ampliação dos negócios, já que o *delivery* se torna o foco da expansão, assim como sua organização de tais dados.

Hoje em dia, as atividades de comércios do ramo de panificação são realizadas de forma manual e trabalhosa. A pessoa responsável por pedidos usa aplicativos terceiros que não são adequados para tal tarefa pois estes não possuem objetivos de atender a essa necessidade, o que possibilita uma grave perda de integridade dos dados. Portanto, a falta de controle de pedidos já citado, e a falta de sistemas específicos com baixo custo para panificadoras nos traz à tona de implantar tal sistema. Soma-se a isso o fato que, muito dos sistemas existentes e considerados “softwares de prateleiras” se preocupem com a parte gerencial e contábil da empresa, ou possuem requisitos em demasia para processos desejosos de serem mais simples no cotidiano das panificadoras.

1.3. Objetivo geral

O sistema tem como objetivo automatizar o controle de pedidos e entregas aos clientes de diversos produtos, tais como pães, lanches, bebidas e quaisquer outros que a padaria oferece, gerenciar os pedidos, proporcionar melhor atendimento aos seus clientes, já que não há necessidade de filas ou sequer sair do conforto do próprio lar, estabelecer acompanhamento das ações produzidas no sistema tanto pelo cliente, quanto pelo administrador e controle

de relatórios gerados, com o intuito de automatizar os processos referentes aos pedidos.

Com isso, é pretendido tornar o processo de *delivery* mais rápido, já que o processo informal possui amplos fatores humanos que induzem ao erro como a manuscrita errônea e desorganização de uma fila de pedidos.

1.4. Objetivos específicos

Apresentar um melhor atendimento à clientela vêm a partir do momento em que o mesmo se sente amigável e confortável com a interface com a qual ele está em acesso. Com isso, uma interface humano-computador que atenda essa forma de abordagem, consegue fidelizar um pouco mais a cada uso e quando o mesmo possui acesso à todas ações realizadas no sistema, o usuário se sente no controle de tudo o que ele já fez no sistema.

Oferecer o que o estabelecimento tem de melhor é essencial, já que o cliente terá a opção de ter os produtos em qualquer lugar que seja considerado derredor da padaria por meio do *delivery*. E se a concorrência da padaria sequer oferece serviço parecido, este diferencial se torna um grande fator a favor de uma fidelização.

Tem-se por proposição central comercial do sistema que este seja utilizado por qualquer padaria, a um custo baixo para a mesma, seja esta considerada de grande ou pequeno porte. Para isso, oferecer um *template* de adaptação fácil e com uma boa usabilidade se torna um dos principais requisitos não funcionais.

1.5. Levantamento preliminar de requisitos

Esta é uma fase na qual, as ideias e concepções serão, inicialmente, captadas, analisadas e avaliadas sob os diversos aspectos que a engenharia rege, e ocorre preliminar no processo de software. Ou seja, pretende-se ter uma ideia inicial geral dos requisitos de maior prioridade do cliente, em nível de abstração, sendo que sua otimização e captação de todos os requisitos serão feitos em levantamentos posteriores, gerando a especificação completa dos requisitos (PREESMAN, 2011). Esta especificação está detalhada em outra seção mais a frente neste presente documento, quando ocorrerá tal

refinamento dos requisitos e onde se terá o entendimento do projeto por completo.

Conforme proposto por Sommerville (2003), um requisito é tratado como funcional quando descreve um serviço ou função que o sistema deve realizar. Paralelamente pode haver requisitos não-funcionais, que são restrições impostas tanto ao sistema quanto ao seu desenvolvimento.

Após o primeiro contato com instituições do ramo do litoral sul fluminense do Brasil, foi definido que o sistema irá abranger toda a operação de *delivery*. E desse contato foram definidas as seguintes características que deverão estar presentes no sistema:

- **Cadastro de Clientes**

O cadastro de clientes e seus respectivos dados, devido à necessidade de controle sobre os pedidos e entregas.

- **Cadastro de Produtos**

O cadastro de produtos e seus respectivos atributos que terá que ser feito de forma diária pelo administrador.

- **Realizar pedidos**

Um usuário poderá fazer um pedido com mais de um item diferente.

- **Forma de pagamento**

O cliente poderá selecionar no próprio site a forma de pagamento conforme for sugerido pelo estabelecimento.

- **Administrador ter acesso de forma fácil às suas edições**

Diariamente haverá a necessidade, de forma a adequar a oferta fornecida pela empresa.

- **Registro das operações de vendas**

Todas as ações no sistema relacionados às vendas deverão ser registrados e mantidas pelo máximo de tempo possível.

Funcionalidades

O sistema terá as seguintes funcionalidades básicas:

- Escolha dos pedidos de forma detalhada; se for refeição, qual o tipo; quaisquer outros alimentos incluirão escolha dos respectivos recheios, sabores, etc.
- Recebimento dos pedidos pelo administrador em tempo real.
- Funções administrativas não precisam de cadastro para acesso.
- Armazenamento dos pedidos por cliente/empresa.
- Possibilidade de edição diária de itens como marmita, lanches, etc. e respectivos preços.
- Relatórios individuais por periodicidade de pedidos dos clientes.
- Relatório de clientes ordenados alfabeticamente.
- Saída de vendas em ordem recebida de forma ascendente ou descendente.

1.6. Requisitos Funcionais

Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários (PAULA FILHO, 2000). De acordo com SOMMERVILLE (2018), os requisitos funcionais de usuários definem recursos específicos que devem ser fornecidos pelo sistema. A tabela 1 apresenta os requisitos funcionais, identificados no formato de requisitos conforme Gilleane Guedes (GUEDES, 2009).

Identificador	Requisito	Descrição
RF01	Manter Cadastro de Clientes	Manter dados do cliente (nome, senha, contato e cpf), atribuir um ID único e permitir a manipulação desses dados. Será possível o cadastro, edição e listagem dos mesmos em um relatório total.
RF02	Edição de dados do usuário	Usuário poderá editar os seus dados de senha e contato bem como ter a possibilidade de adicionar informações de endereço de entrega no ato do pedido como nome da rua, número, complemento e bairro.
RF03	Fazer pedidos	O usuário poderá fazer pedidos por meio de um carrinho de compras disposto no próprio sistema.
RF04	Emitir relatório de pedidos feitos pelo próprio cliente	Cliente poderá ter acesso ao histórico de seus pedidos e seus respectivos valores por meio de um relatório contendo todos os pedidos.
RF05	Login do cliente/empresa	Permitir a autenticação de usuário por meio de login através de um nome para login e senha de acesso.
RF06	Emitir Relatório de vendas ao administrador.	Geração de relatório referente às vendas, seja por filtro de dias ou por cliente.
RF07	Atualizar pedido	Pedidos feitos deverão estar visíveis ao administrador e ao cliente bem como o seu respectivo status.
RF08	Manter produtos	O administrador poderá manter (editar, remover, adicionar e atualizar) dados dos produtos (nome, descrição, preço, status e sua respectiva categoria).
RF09	Acompanhar pedido	Tanto o administrador dos pedidos, quanto o cliente, terão acesso ao status do pedido, como “pedido feito” e “entregue”
RF10	Cadastro de cupom	O administrador poderá adicionar ou excluir cupom contendo nome e porcentagem de desconto, bem como visualizar os cadastrados do mesmo
RF11	Cadastro de categoria	O administrador poderá adicionar ou remover uma categoria na qual não há produto cadastrado na mesma, bem como visualizar os cadastrados do mesmo

Tabela 1.: Requisitos funcionais.

1.7. Requisitos não-funcionais

Requisitos não funcionais são restrições sobre serviços ou funções oferecidas pelo sistema. Dentre elas destacam-se restrições de tempo, sobre o processo de desenvolvimento e de padrões. A descrição das restrições complementa a definição de requisitos (PAULA FILHO, 2000).

Os requisitos não funcionais às vezes podem dizer algo a respeito do sistema como um todo. Isso significa que na maioria das vezes eles são mais importantes que os requisitos funcionais individuais. Se uma falha em cumprir um requisito funcional pode comprometer parte do sistema, uma falha em cumprir um requisito não funcional pode tornar todo o sistema inútil (SOMMERVILLE, 2008).

RNF01: O sistema deverá ser web;

RNF02: Fora do horário de delivery, o usuário não terá acesso às funções que permitem realizar pedido;

RNF03: O sistema deverá ser adaptado ao *mobile*;

2. ESPECIFICAÇÃO DE REQUISITOS

Segundo Pressman, o levantamento de requisitos combina elementos de resolução de problemas elaboração, negociação e especificação. (PRESSMAN, 2011).

A especificação é transfazer tudo o que se sabe no levantamento dos requisitos em algo que será entendido por quem tiver acesso ao sistema ou ao usuário. Com isso, vem a importância de especificar em detalhes como as funções do sistema irão ser executadas e como se portarão perante situações obedecendo os requisitos funcionais e não-funcionais.

A engenharia de requisitos, de acordo com Sommerville (2007), tem como objetivo definir o que o sistema deve fazer, quais as necessidades reais e identificar quais restrições existem para que o software seja desenvolvido. É nesse processo da engenharia de software que ocorre a comunicação entre o cliente e o analista da equipe de desenvolvimento. Quando essa comunicação não é bem-sucedida, o restante do projeto pode ficar comprometido, causando impacto no custo e prazo.

Pressman (2006) afirma que os requisitos devem ser examinados para que sejam encontradas inconsistências, ambiguidades e omissões. Para tal

atividade, Sommerville (2007) propõe algumas técnicas de validação, tais como revisões de requisitos, prototipação e geração de casos de teste

2.1. Requisitos do Cliente

Serão especificados dois tipos de requisitos, os requisitos do cliente que deverão levar em conta uma descrição mais abstrata que será direcionada ao usuário final do sistema, descrever funções e restrições do sistema de forma abstrata do ponto de vista do cliente, e a especificação do sistema que é a descrição mais detalhada do sistema que deve ser completa e consistente.

2.1.1. Descrição geral da Aplicação

O sistema web para padarias será desenvolvido para atender empresas do ramo da panificação que desejam inserir o *delivery* em sua carta de serviços.

Trata-se de um sistema genérico em que as regras de negócio precisam ser compatíveis com os requisitos do sistema.

2.2. Requisitos do Software

Os requisitos de software têm um papel central no desenvolvimento, sendo considerado um fator determinante para o sucesso ou fracasso de um projeto de software. O processo de levantar, analisar, documentar, gerenciar e controlar a qualidade dos requisitos é chamado de Engenharia de Requisitos (FALBO, 2007).

2.2.1. Fronteiras do Sistema

São descritas como fronteiras do sistema tudo que não faz parte do escopo:

- O backup (o sistema não fará backup).
- O sistema não fará qualquer controle financeiro que vá além dos relatórios.

2.2.2. Itens de Software

O sistema web possuirá as seguintes funcionalidades que delimitaram as fronteiras do sistema:

- **Incluir**: Cliente, Categoria, Cupom, Endereço de entrega, Pagamento, Pedido, Produto.
- **Exclusão**: Categoria, Cupom, Produto.
- **Alteração**: Cliente, Endereço de entrega, Produto,
- **Buscas:** Cliente, Categoria, Cupom, Endereço de entrega, Pagamento, Pedido, Produto.

2.2.3. Escopo da aplicação

O sistema web tem como intuito permitir o cadastro de dados necessários tais como, dados do cliente, dados do produto e dados de promoções, e dar suporte com a geração de relatórios para manutenção do sistema.

O sistema contará com os seguintes relatórios:

- Relatório de Pedidos realizados;
- Relatório de Clientes;
- Relatório de Pedidos em aberto;
- Relatório de Pedidos finalizados;
- Relatório de Cupons de promoção;
- Relatório de Categorias e seus devidos produtos;

O controle do sistema será através de um único nível de acesso, o nível administrador, que terá acesso à todas as funcionalidades do sistema. Cada usuário terá um perfil com um login e senha únicos, podendo para cada um o direito de alterar a senha caso precise, e o administrador terá o controle geral sobre atividades dos usuários do sistema.

2.2.4. Lista de Requisitos Não Funcionais

Requisitos não funcionais são restrições sobre serviços ou funções oferecidas pelo sistema. Dentre elas destacam-se restrições de tempo, sobre o processo de desenvolvimento e de padrões. A descrição das restrições complementa a definição de requisitos (PAULA FILHO, 2000).

RNF01: O sistema deverá ser web;

RNF02: Fora do horário de delivery, o usuário não terá acesso aos campos do site que são relacionados aos pedidos;

RNF03: O sistema deverá ser adaptado ao *mobile*;

2.2.5. Requisitos Funcionais

Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários (PAULA FILHO, 2000).

2.2.5.1. Lista de Requisitos Funcionais

Requisitos de Inclusão:

RF01: Manter Cliente;
RF03: Fazer Pedido;
RF08: Manter Produto;
RF10: Cadastro Cupom;
RF11: Cadastro Categoria;

Requisitos de Alteração:

RF01: Manter Cliente;
RF02: Edição de dados do usuário;
RF08: Manter Produto;

Requisitos de Exclusão:

RF08: Manter Produto;
RF10: Cadastro Cupom;
RF11: Cadastro Categoria;

Requisitos de Consulta:

RF01: Manter Cliente;
RF04: Emitir relatório de Pedido;
RF06: Emitir relatório de venda;
RF08: Manter Produto;
RF10: Cadastro Cupom;
RF11: Cadastro Categoria;

2.2.5.2. Descrição dos Atores

Os atores do sistema serão baseados em um perfil de acesso: Administrador.

Descrição dos atores	
Ator	**Descrição**
Administrador	O ator administrator (gestor) possuirá acesso a todas as funcionalidades do sistema.
Usuário	O usuário terá acesso as funcionalidades do sistema com exceção das do administrador.

Tabela 2.: Descrição dos Atores.

2.2.5.3. Modelo de Casos de Uso

Modelos de casos de uso (*use cases*) são uma forma de estruturar essa visão. Como o próprio nome sugere, um caso de uso é uma maneira de usar o sistema. Usuários interagem com o sistema, interagindo com seus casos de uso. Tomados em conjunto, os casos de uso de um sistema definem a sua funcionalidade. Como se pode perceber, casos de uso têm uma estreita

relação com requisitos funcionais e podem ser derivados diretamente deles (FALBO, 2007).

2.2.5.4. Especificação de Caso de Uso

CASO DE USO: FAZER PEDIDO

SUMÁRIO: Esse caso de uso serve para as seguintes entidades: pedido e item. O caso de uso inicia quando o usuário deseja fazer uma compra.

ATORES: Usuário

PRÉ-CONDIÇÕES: Produto(s) cadastrado(s) no sistema assim como sua respectiva categoria;

CURSO NORMAL:

1. O usuário solicita tela de pedido;
2. O sistema mostras todas categorias existentes para escolha;
3. O usuário seleciona a categoria;
4. O sistema mostra todos os produtos existentes de acordo com a escolha;
5. O usuário escolhe o produto;
6. O sistema exibe o carrinho de compras
7. O usuário insere a quantidade do produto;
8. O usuário solicita atualização do produto;
9. O sistema guarda a preferência e exibe o carrinho atualizado;
10. O usuário solicita finalização da compra;
11. Sistema salva o carrinho e exibe o checkout;
12. Usuário seleciona o endereço para entrega;

13. O sistema guarda a preferência e exibe checkout bem como o método de pagamento;
14. Usuário seleciona o método de pagamento;
15. Sistema salva todos os dados, exibe mensagem de compra finalizada;

CURSO ALTERNATIVO:

Entre os passos 1 e 3, o usuário abandona a operação; caso de uso termina.

10.1. O usuário deseja adicionar mais produtos ao carrinho; Caso de uso retorna ao passo 2;

14.1. O usuário escolhe a forma de pagamento em dinheiro; Caso de uso retorna ao passo 15

14.2. O usuário escolhe pagamento via cartão;

14.2.1 Sistema salva a preferência e exibe na tela o formulário para preenchimento dos dados do cartão.

14.2.2 Usuário preenche os campos: Número do cartão, data de validade e número de segurança e salva as alterações; Caso de uso retorna ao passo 15.

CURSOS DE EXCEÇÃO:

8.1. O usuário não solicitou atualização do item; Sistema exibe mensagem de alerta; Caso de uso retorna ao mesmo passo

11.1. O usuário não está logado ao sistema; Sistema exibe tela de login; Caso de uso retorna ao passo 11;

PÓS-CONDIÇÕES: Cadastro devidamente gravado no sistema; Sistema exibe página inicial.

REQUISITOS DE INTERFACE: Ao ser solicitado a mudança na quantidade de um produto no carrinho, o sistema exibe mensagem de alerta informando que o usuário deve solicitar a atualização do produto. As categorias e produtos devem ser mostrados em formato de cartões. Tabela de checkout deve contar cores alternadas por linha (escuro/claro).

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá que o usuário adicione apenas um produto por vez.

Vimos o fluxo principal do sistema que ocorre quando o usuário deseja realizar um pedido. Todo o restante dos casos de uso estão descritos no apêndice 1 deste documento.

3. PROCESSO DE DESENVOLVIMENTO

3.1. Metodologia SCRUM

O modelo do processo de desenvolvimento a ser adotado nesse projeto, será o modelo Ágil Scrum (SCRUM, 2017). Os métodos ágeis diferem largamente no que diz respeito a forma de serem gerenciados. Alguns métodos são suplementados com guias para direcionar o gerenciamento do projeto, mas nem todos são aplicáveis (ABRAHAMSSON, 2003).

Os princípios do modelo Scrum são consistentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades estruturais: requisitos, análise, projeto, evolução e entrega. Em cada atividade metodológica, ocorrem tarefas a realizar dentro de um padrão de processo (discutido no parágrafo a seguir) chamado Sprint. O trabalho realizado dentro de um Sprint (o número de sprints necessários para cada atividade metodológica varia dependendo do tamanho e complexidade do produto) é adaptado ao problema em questão e definido, e muitas das vezes modificado em tempo real, pela equipe Scrum. O fluxo do processo Scrum é ilustrado na Figura 1 (PRESSMAN, 2011).

O Scrum enfatiza o uso de um conjunto de padrões de processos de software que provaram ser eficazes para projetos com prazos de entrega apertados, requisitos mutáveis e críticos de negócio. Cada um desses padrões de processos define um conjunto de ações de desenvolvimento:

Registro Pendente de Trabalhos (*Backlog*) – uma lista com prioridades dos requisitos ou funcionalidades do projeto que fornecem valor comercial ao cliente. Os itens podem ser adicionados a esse registro em qualquer momento (é assim que as alterações são introduzidas). O gerente de produto avalia o registro e atualiza as prioridades conforme requisitado.

Urgências ou (corridas de curta distância) ***Sprints*** – consistem de unidades de trabalho solicitadas para atingir um requisito estabelecido no registro de trabalho (*backlog*) e que precisa ser ajustado dentro de um prazo já fechado, tipicamente 30 dias (PRESSMAN, 2011).

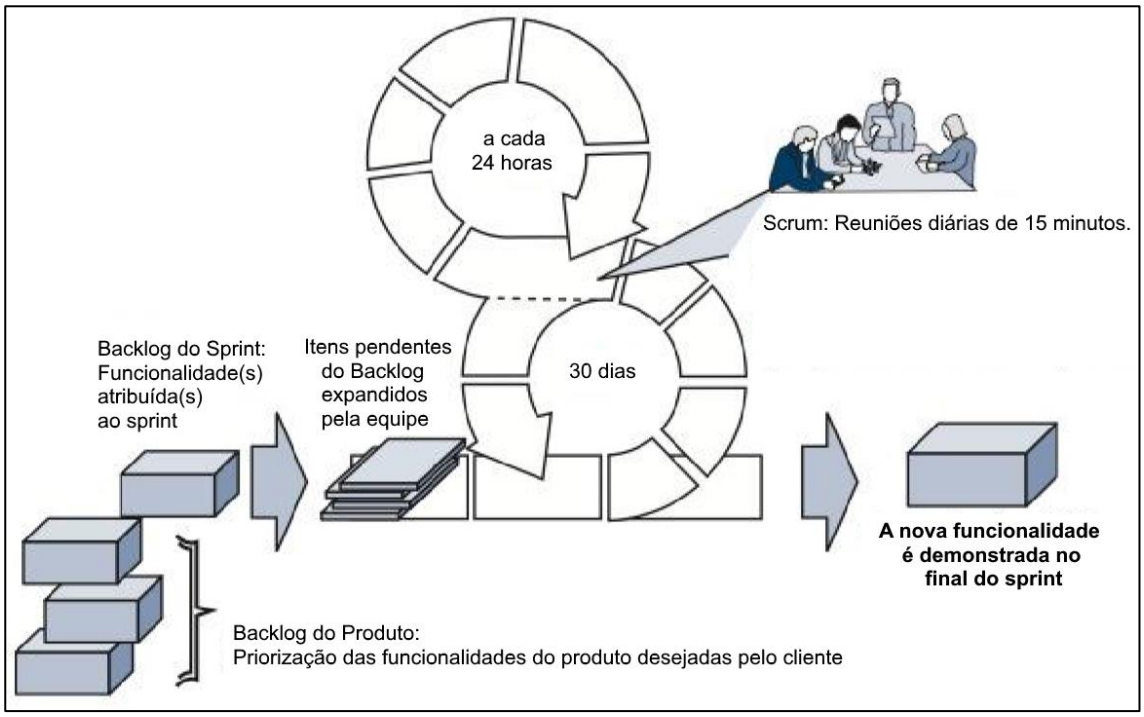


Figura 1.: Fluxo do processo Scrum.

FONTE: PRESSMAN (2011).

Um líder de equipe, chamado Scrum Master, conduz a reunião e avalia as respostas de cada integrante. As reuniões ajudam a equipe a revelar problemas potenciais o mais cedo possível. Promovendo uma estrutura de equipe auto organizada (PRESSMAN, 2011).

3.2. Metodologia de Desenvolvimento

Para este projeto foi escolhida o paradigma de programação orientada a objetos, já que a mesma é a o que mais temos conhecimento de aplicação.

Segundo Oglio (2007) a orientação a objetos é um paradigma que representa toda uma filosofia para construção de sistemas. Como alternativa de se construir um sistema formado por um conjunto de procedimentos e

variáveis, nem sempre agrupados de acordo com o contexto, como acontece nas linguagens estruturadas, na orientação a objetos utilizamos uma ótica mais próxima do mundo real, com objetos do dia-a-dia que conhecemos e possuímos melhor compreensão.

A programação orientada a objetos fornece uma visão do mundo de forma computacional, implementando um conjunto de classes para definir os objetos que compõem o sistema, cada classe é composta por atributos e métodos, ou seja, são as características e comportamentos que o objeto possui.

3.3. Diagrama de classe

De acordo com Guedes (2014), o Diagrama de Classes tem como foco principal permitir a visualização das classes que compõe o sistema e seus respectivos atributos e métodos, bem como a associação entre elas. Na figura 2 é apresentado o diagrama de classes de modelo do sistema proposto. É importante ressaltar que para cada atributo representado existem métodos construtores e métodos getter e setter.

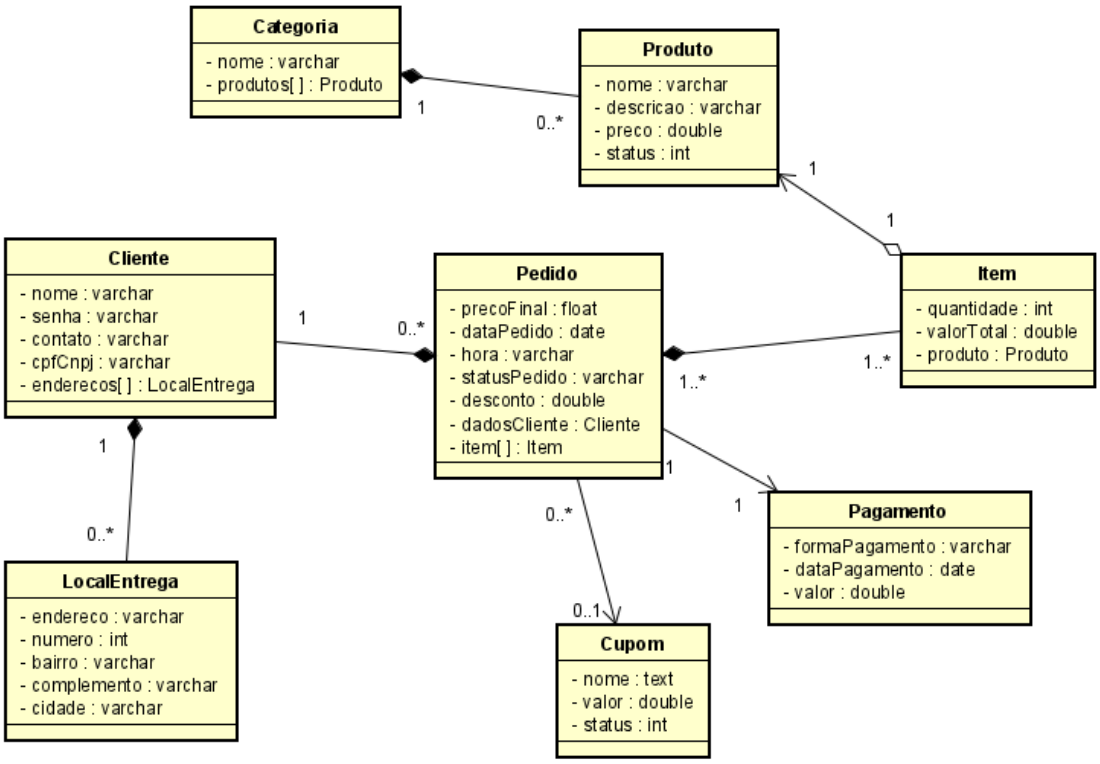


Figura 2.: Diagrama das classes de modelo.

A arquitetura MVC é dividida em três componentes e conforme afirmado por Zucher (2020), a camada Model possui a responsabilidade de gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e as regras de negócio estabelecidas. A camada Controller é responsável por intermediar as requisições enviadas pela View com as respostas fornecidas pela camada Model. Já a View é responsável por apresentar as informações de forma visual ao usuário.

A figura 3 ilustra a interação entre as camadas do sistema. Para tanto, foi utilizado o exemplo de Produto. Para cada entidade do sistema, contém o comportamento semelhante no que se diz respeito à interação.

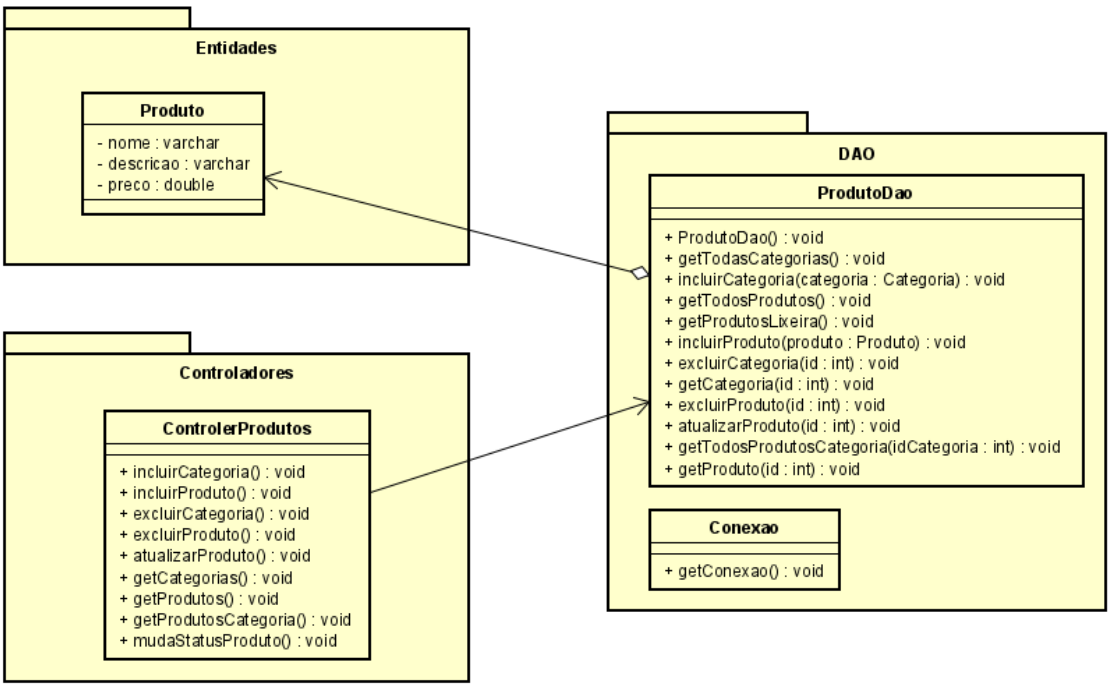


Figura 3.: Interação das classes de modelo, controladoras e DAO.

3.4. Estimativas (Tamanho, esforço e prazo)

Existem técnicas úteis para estimar tempo e esforço. As métricas de projeto e processo podem proporcionar perspectivas históricas e valiosas informações para determinar estimativas quantitativas (PRESSMAN, 2011).

Foram feitos ajustes iniciais desde a primeira estimativa até a entrega e a seção seguinte apresenta a última estimativa, sendo que no final da seção, será apresentada uma tabela da estimativa inicial e a final e os ajustes realizados.

3.4.1. Identificação das funcionalidades do sistema

A métrica para estimativa escolhida como base para as estimativas deste projeto é Análise de pontos de função – FPA ou (*Function Point Analysis)* (IFPUG, 2018), com base no levantamento preliminar será usado para mensurar o sistema, visto que como se trata de um levantamento preliminar pode-se ter necessidade de alteração nos cálculos.

A FPA é atualmente um instrumento utilizado por profissionais da área de sistemas e em empresas de todos os portes e segmentos da economia brasileira. Inicialmente o foco principal de sua aplicação era em estimativas. Atualmente ainda continua sendo uma ferramenta importante em estimativas, mas também tem sido aplicada cada vez com mais sucesso como unidade de medição de contratos de desenvolvimento de software e como ferramenta na gerência de projetos (VAZQUEZ, 2013).

3.4.2. Entrada Externa ou Input (I)

Este processo estabelece as entradas responsáveis por processar dados ou informações de controle vindos de fora da fronteira da aplicação. Uma das principais intenções de uma entrada externa é manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do software (PRESSMAN, 2011).

Entradas Externas
Incluir Cliente
Alterar Cliente
Incluir Categoria
Excluir Categoria
Incluir Cupom
Excluir Cupom
Incluir Endereço
Alterar Endereço

Excluir Endereço
Incluir Pagamento
Incluir Pedido
Incluir Produto
Alterar Produto
Excluir Produto

Tabela 3.: Entradas Externas.

3.4.3. Saída Externa ou Output (O)

Principal objetivo de uma saída externa é apresentar dados ao usuário por meio de lógica de processamento que não seja apenas recuperação de dados. A lógica de processamento deve obrigatoriamente conter ao menos uma fórmula matemática ou cálculo, ou criar dados derivados. Pode também manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do software. Ou seja, uma saída externa tem a função de gerar dados ou informações de controle que saem pela fronteira da aplicação (PRESSMAN, 2011).

Saídas Externas
Total de Pedidos do dia
Total de Clientes
Total de Produtos

Tabela 4.: Saídas Externas.

3.4.4. Consulta Externa (C)

A consulta externa é um processo elementar que há o envio de dados ou informações de controle para fora da fronteira da aplicação. Sua principal intenção é apresentar informação ao usuário por meio de uma simples recuperação de dados de arquivos lógicos internos e/ou arquivo lógico externo. A lógica de processamento não deve conter fórmula matemática ou cálculo, criar dados derivados, manter um ou mais ALI e/ou alterar o comportamento do software. Pode-se citar como um exemplo as telas de ajuda (PRESSMAN e MAXIM, 2011).

Consultas Externas
Buscar Cliente
Buscar Categoria
Buscar Cupom
Buscar Pagamento
Buscar Pedido
Buscar Produto
Buscar Categoria

Tabela 5.: Consultas Externas.

3.4.5. Classificação conforme nível de complexidade

A seguir nas Tabelas 6, 7, 8 e 9 serão apresentadas as Entradas Externas são transações com o objetivo de atualizar arquivos lógicos internos ou modificar o comportamento do sistema. Consultas Externas são transações que representam simples recuperação de dados de arquivos lógicos internos e/ou arquivos de interface externa e Saídas Externas são transações com o objetivo de apresentação de informação, porém envolvendo lógica de processamento adicional a uma consulta externa.

Entradas Externas	Tipo de Dados	Grau
Incluir Cliente	5	Simples
Alterar Cliente	5	Simples
Incluir Categoria	2	Simples
Excluir Categoria	1	Simples
Incluir Cupom	4	Simples
Excluir Cupom	1	Simples
Incluir Endereço	7	Simples
Alterar Endereço	7	Simples
Excluir Endereço	2	Simples
Incluir Pagamento	5	Médio
Incluir Pedido	8	Complexo
Incluir Produto	6	Simples
Alterar Produto	6	Simples
Excluir Produto	2	Simples

Tabela 6.: Classificação Entradas Externas.

Saídas Externas	Tipo de Dados	Grau
Total de Pedidos do dia	8	Médio
Total de Clientes	8	Simples
Total de Produtos	8	Simples

Tabela 7.: Classificação Saídas Externas.

Consultas Externas	Tipo de Dados	Grau
Buscar Cliente	5	Simples
Buscar Categoria	2	Simples
Buscar Cupom	4	Simples
Buscar Pagamento	5	Simples
Buscar Pedido	8	Complexo
Buscar Produto	6	Simples

Tabela 8.: Classificação Consultas Externas.

Arquivos Lógicos Internos	Tipo de Registro	Grau
Cadastrar Cliente	1	Simples
Cadastrar Categoria	1	Simples
Cadastrar Cupom	1	Simples
Cadastrar Endereço	1	Simples
Cadastrar Produto	1	Simples

Tabela 9.: Classificação ALI's.

3.4.6. Calculando Pontos Brutos

Como pode ser verificado na Tabela 10, de acordo com o número de ocorrências, sua complexidade e peso, obtivemos os resultados em cada função, e assim, ao somarmos todos os resultados, temos o total de pontos brutos de 124.

Função	Nº de Ocorrências	Complexidade	Peso	Resultado
Entradas Externas	12	Simples	x 3	36
	1	Médio	x 4	4
	1	Complexa	x 6	6
			Total	**46**
Saídas Externas	2	Simples	x 4	8
	1	Médio	x 5	5

	0	Complexa	x 7	0
			Total	**13**
Arquivos Lógicos Internos	7	Simples	x 7	49
	0	Médio	x 10	0
	0	Complexa	x 15	0
			Total	**49**
Arquivos de Interface Externa	0	Simples	x 5	0
	0	Médio	x 7	0
	0	Complexa	x 10	0
			Total	**0**
Consultas Externas	5	Simples	x 3	15
	0	Médio	x 4	0
	1	Complexa	x 6	1
			Total	**16**
		Total de Pontos Brutos		**124**

Tabela 10.: Tabelas de Pesos FPA.

No FPA, cada nível de complexidade tem um peso. Devemos então multiplicar o número de ocorrências de uma função pelo peso correspondente ao nível de complexidade (IFPUG, 2018).

3.4.7. Calculando Fator de Ajuste e Pontos Ajustados

Segundo o International Function Point Users Group (IFPUG) existem 14 características que podem influenciar no projeto e que devem ser levadas em conta para fins de cálculos e estimativas do tamanho do software.

Sendo assim, as características são: 1) Comunicação de dados, 2) Funções distribuídas, 3) Desempenho, 4) Carga de configuração, 5) Volume de transações, 6) Entrada de dados online, 7) Eficiência do usuário final, 8) Atualização online, 9) Processamento complexo, 10) Reutilização, 11) Facilidade de implantação, 12) Facilidade operacional, 13) Múltiplos locais e 14) Facilidade de mudança; onde são atribuídas para cada característica o peso de 0 (zero) a 5 (cinco), de acordo com o nível de influência da mesma no projeto conforme a tabela abaixo.

Peso	Nível de Influência (NI)
0	Nenhuma influência
1	Influência mínima
2	Influência moderada

3	Influência média
4	Influência significativa
5	Grande influência

Tabela 11.: Nível de influência.

1) Comunicação de dados:

A característica Comunicação de dados foi atribuída o peso 4, pois é mais que um *front-end,* mas suporta apenas um tipo de protocolo de comunicação.

2) Funções distribuídas:

As Funções distribuídas receberam peso 1, pois a aplicação prepara dados para o usuário final utilizar em outro componente do sistema, tal como planilhas eletrônicas ou banco de dados.

3) Performance:

A Performance foi atribuída peso 0, pois nenhum requerimento especial de performance foi solicitado pelo usuário.

4) Configuração do equipamento:

A Configuração do Equipamento recebeu peso 1, pois existem restrições operacionais leves. Não é necessário esforço especial para resolver as restrições.

5) Volume de transações:

O Volume de transações foi atribuído peso 1, pois existem restrições operacionais, mas nenhum esforço extra é necessário para suportá-las

6) Entrada de dados:

A Entrada de dados online recebeu peso 0, pois todas as transações são processadas em modo batch.

7) Interfaces com usuário:

A Interface com usuário foi atribuída peso 5, pois o sistema possui menus, utilização de mouse, *scrolling* vertical, *drop down list*, janelas pop-up, menos número de telas para executar funções.

8) Atualização Online:

A Atualização online recebeu peso 0, pois não há nenhuma atualização on-line.

9) Processamento Complexo:

O Processamento complexo foi atribuído peso 1, pois tem processamento matemático extensivo.

10) Reutilização:

A Reutilização recebeu peso 1, pois o código reutilizável é usado somente na aplicação.

11) Facilidade de implantação:

A Facilidade de implantação foi atribuída peso 1, pois nenhuma consideração especial foi estabelecida pelo usuário, mas procedimentos especiais são necessários na implantação, neste caso disponibilidade de hospedagem na web e configuração de banco de dados.

12) Facilidade operacional:

A Facilidade Operacional foi atribuída peso 1, pois possui o processo de backup e recuperação com a intervenção do operador.

13) Múltiplos Locais:

Os Múltiplos locais receberam peso 2, pois a necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi desenhada para operar apenas sobre o mesmo ambiente de software e hardware.

14) Facilidade De Mudança:

A Facilidade de mudanças recebeu peso 1, pois estão disponíveis facilidades como consultas e relatórios flexíveis para atender necessidades simples.

A Tabela 12 apresenta um resumo geral contendo todas as características já mencionadas anteriormente e seus respectivos níveis de influência.

	Característica Geral dos Sistemas	**Nível de Influência**
1	Comunicação de dados	4
2	Funções distribuídas	1
3	Performance	0
4	Configuração do equipamento	1
5	Volume de transações	1
6	Entrada de dados	0
7	Eficiência do usuário final	5

8	Atualização on-line	0
9	Processamento complexo	1
10	Reutilização	1
11	Facilidade de implantação	1
12	Facilidade de operação	1
13	Múltiplos locais	2
14	Facilidade de mudanças	1
	Total de Influência	**19**

Tabela 12.: Pesos de Influência.

3.4.8. Fator de Ajuste

A equação do fator de ajuste é calculada pela soma de todas as influências (definidas anteriormente), com base nessas influências os pontos brutos são ajustados 35% para mais ou 35% para menos, o resultado máximo que se pode alcançar no cálculo é 1,35, ou seja, quando todas as 14 características são de vital importância para o sistema, ou o mínimo de 0,65.

Fator de Ajuste	
Fórmula	(Total de Influência * 0,01) + 0,65
Desenvolvimento	(19 * 0,01) + 0,65
Resultado	**0,84**

Tabela 13.: Fator de Ajuste.

3.4.9. Pontos Ajustados

A equação de cálculo dos pontos ajustados é definida pelo total de pontos brutos vezes o fator de ajuste, logo:

Pontos Ajustados	
Fórmula	Total de Pontos Brutos * Fator de Ajuste
Desenvolvimento	124 * 0,82
Resultado	**101,68**

Tabela 14.: Pontos Ajustados.

3.4.10. Estimativa de Esforço de Trabalho

Para estimar o esforço de trabalho é utiliza a expressão que calcula o tempo de trabalho em horas é dado pela expressão abaixo:

```
Esforço Trabalho = Pontos Ajustados * índice
produtividade
```

Para esse projeto foi escolhido usar como base no cálculo de esforço por hora, linguagem orientada a objeto que tem um esforço de 6,5 a 12 horas para cada ponto de função, julgando um conhecimento mediano sobre linguagens orientadas a objeto será usado 9 horas por ponto de função, logo:

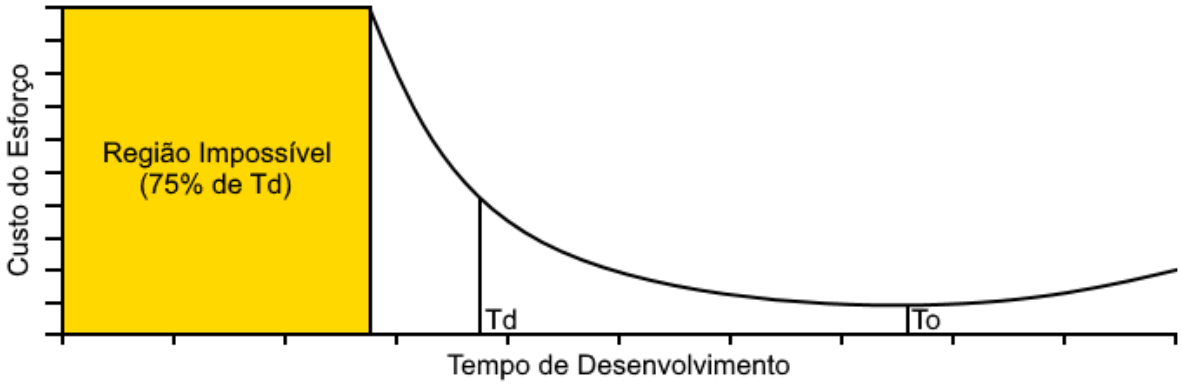
Esforço de Trabalho	
Fórmula	Pontos Ajustados * Índice Produtividade
Desenvolvimento	102 * 9 horas
Resultado	**918 horas**

Tabela 15.: Esforço de Trabalho.

3.4.11. Estimativa de prazo em meses

O monitoramento e controle do prazo é realizado para garantir que o projeto seja finalizado dentro do prazo planejado, será realizado comparando se as tarefas que estão sendo realizados estão de acordo com o cronograma.

As revisões de prazo serão realizadas ao final de cada marco ocorrendo algum problema, será ajustado o cronograma, com isso garantindo um prazo coerente durante todo o projeto. Ocorrendo uma extensão muito alta na data de entrega do projeto, se fará necessário fazer hora extra para cumprimento do prazo de entrega.



Observações:

1. Td é o tempo ótimo de desenvolvimento.
2. To é o tempo que acarreta o menor custo.
3. To = 2 Td.
4. É impossível terminar em menos que 0,75 * Td.

Figura 4.: Relação entre a Estimativa de Prazo e de Esforço.

FONTE: JONES (2007).

Tipo de sistema	Expoente t
Sistema comum - *Mainframe* (desenvolvimento de sistema alto grau de reuso ou manutenção evolutiva)	0,32 a 0,33
Sistema comum - Web ou cliente servidor	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis, considerar sistema comum	0,36
Sistema cliente/servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,37
Sistemas Gerenciais complexos com muitas integrações, *DataWarehousing*, Geoprocessamento, *Workflow*	0,39
Software Básico, *Frameworks*, Sistemas Comerciais	0,40

Tabela 16.: Tipo de Sistema.

Conforme analisado, esse sistema foi classificado como um Sistema Orientado a Objeto usando como expoente de tempo de desenvolvimento 0,36.

A fórmula ficou da seguinte forma:

$$\text{Tempo de Desenvolvimento} = \text{Pontos Ajustados}^{\text{Expoente t}}$$

Logo, os valores finais são apresentados na Tabela 17.

Tempo de Desenvolvimento	
Fórmula	Pontos Ajustados $^{\text{Expoente t}}$
Desenvolvimento	$102^{0,36}$
Resultado	**5,28 meses**

Tabela 17.: Tempo de Desenvolvimento.

3.5. Plano de Testes

Esse projeto adorará o teste de software Caixa Preta. Este teste consiste em verificar a saída de dados mediante diversas entradas, onde são comparadas a saída dos dados com uma informação “ideal”. Esse tipo de teste pode ser feito em todas as partes do desenvolvimento.

Também será utilizado os testes de Usabilidade, o qual o usuário pode interagir efetivamente com a aplicação (PRESSMAN, 2011), Unitário que procura aferir a correção do código em sua menor fração (DEVMEDIA, 2018). O objetivo é verificar a facilidade que o software ou site possui de ser claramente compreendido e manipulado pelo usuário.

4. PROTOTIPAGEM

Frequentemente, o cliente define uma série de objetivos gerais para o software, mas não identifica, detalhadamente, os requisitos para funções e recursos. Em outros casos, o desenvolvedor encontra-se inseguro quanto à eficiência de um algoritmo, quanto à adaptabilidade de um sistema operacional ou quanto à forma em que deva ocorrer a interação homem/máquina. Em situações como essas, e em muitas outras, o paradigma de prototipação pode ser a melhor escolha de abordagem (PRESSMAN, 2011).

A partir de cada requisito fornecido pelo cliente, foram elaboradas as visões da interface. Com os modelos de protótipo, ganha-se significativamente auxílio para validar os requisitos, como também, ter feedback da aceitação por parte do cliente. Foi de extrema importância o uso dos protótipos para outros processos durante o desenvolvimento, como por exemplo, a fase de implementação. Logo, assim que a fase de validação dos requisitos foi concluída, o protótipo não foi descartado, uma vez que foi muito útil para todo o

restante do trabalho. As Figuras 5,6,7 e 8 ilustram os protótipos propostos para o sistema.

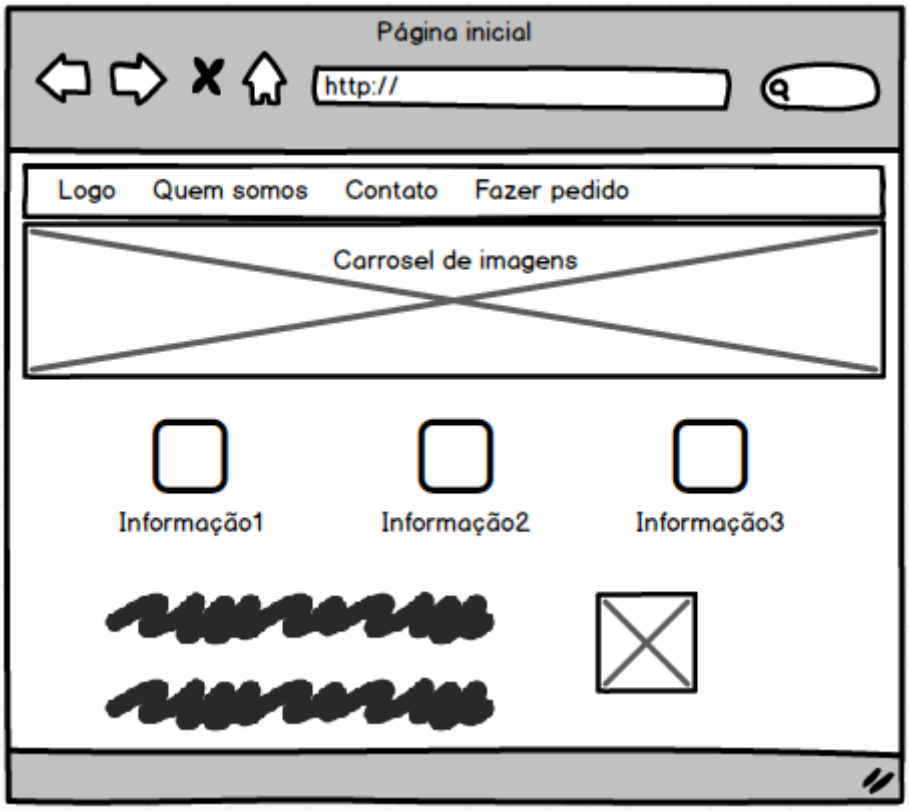


Figura 5.: Protótipo da Tela da página inicial.

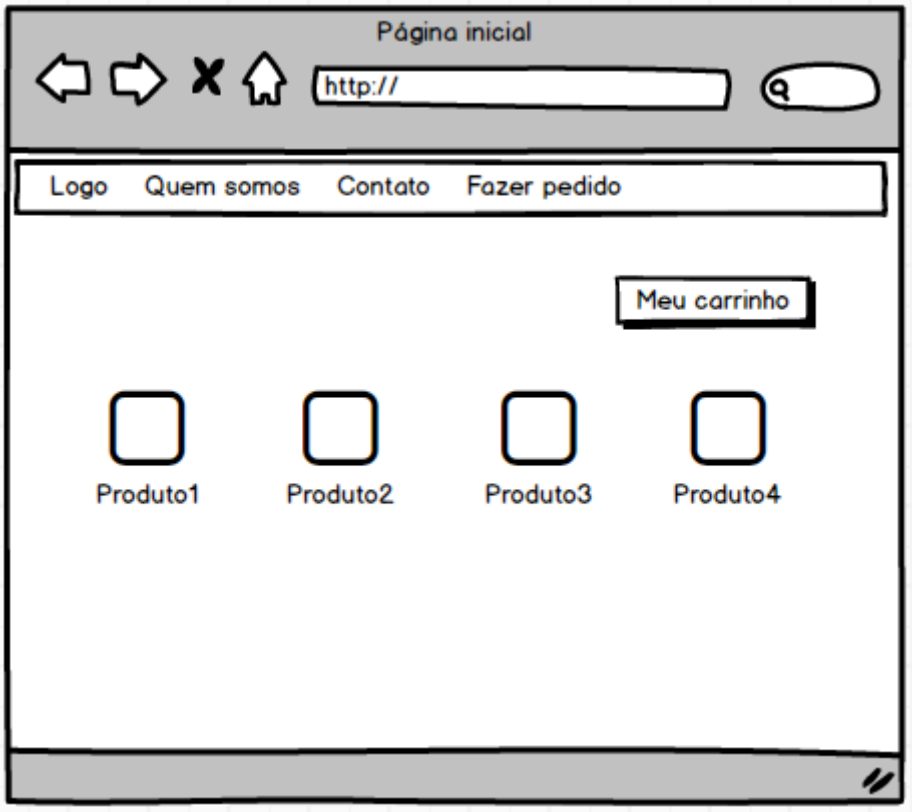


Figura 6.: Protótipo da Tela de escolher pedido.

Figura 7.: Protótipo da Tela de login do usuário.

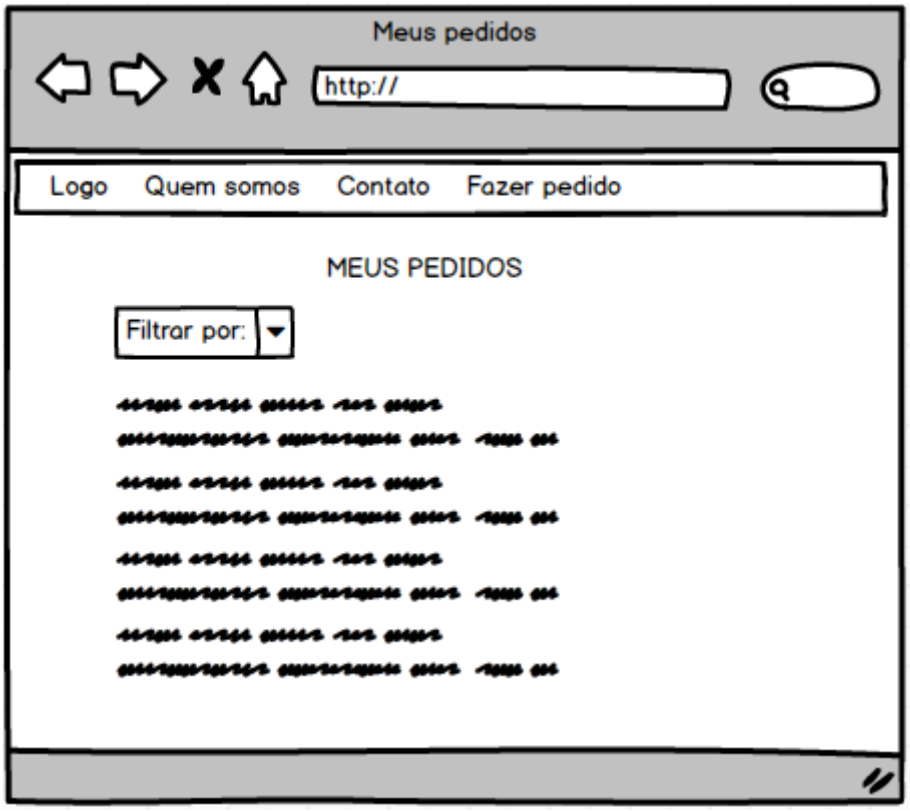


Figura 8.: Protótipo da Tela de Gerar Relatório de pedidos.

5. MODELAGEM DO PROJETO

O Modelo de Projeto tem como objetivo explicitar como o software deve ser feito, diferente do Modelo de Requisitos cujo objetivo é representar “o que” o software faz.

5.1. Arquitetura de Software

O projeto arquitetural do software é toda estrutura de dados e os componentes de programas necessários para concluir um sistema computacional, levando em consideração o tipo de arquitetura, a estrutura e as propriedades dos componentes que constituem o sistema (PRESSMAN, 2011).

Para o desenvolvimento desse software o padrão utilizado foi o MVC (*Model-View-Controller*).

5.1.1. MVC

Segundo Silveira (2012) houve uma grande evolução no desenvolvimento *Web* com o surgimento do padrão arquitetural *Model View Controller*, o MVC, pois era diferente dos outros modelos que não focavam a facilitação com designers, boas práticas de OO e encapsulamento, por exemplo.

Segundo Molski (2017) esse padrão separa a aplicação em três camadas: *model, view* e *controller*, como o próprio nome sugere. Uma das vantagens da utilização do padrão MVC além da organização dos componentes dentro do projeto é a facilidade de manutenção da aplicação, uma vez que a troca de algum componente pode ser resolvida sem maiores problemas, sem a necessidade de descartar todo o restante do código/projeto.

A Figura 9 ilustra o esquema em que a aplicação fica estruturada.

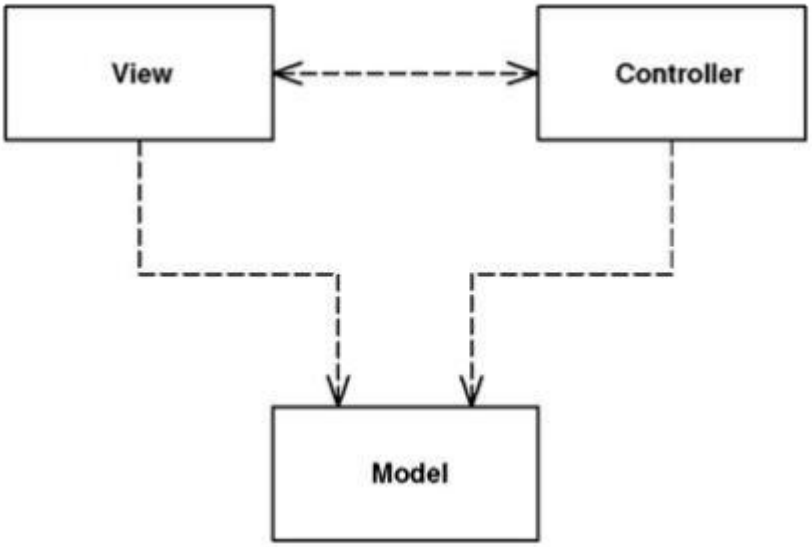


Figura 9.: Model View Controller.

FONTE: FOWLER (2002).

A camada *Model* é reponsável por representar as informações do domínio da aplicação, como o próprio nome sugere é um objeto contendo todos os dados e comportamentos que os mesmos tem na interface.

A camada *View* representa a exibição do modelo na interface, ou seja, apenas exibe as informações.

A camada *Controller* é reponsável por parte relacionada à alteração das informações, isto é, leva a entrada do usuário, manipula o modelo e faz com que a *view* atualize os dados para o usuário.

5.2.1 Plataforma Php

Para Vasquez (2020), o Php é uma linguagem de programação para construir sites dinamicos, extensões de integração de aplicações e agilizar o desenvolvimento de um sistema. Essa linguagem é mundialmente conhecida e uma das mais utilizadas pela facilidade em aprendê-la, manuseá-la, além de ser compatível com quase todos os sistemas operacionais que existem.

Para o desenvolvimento desse software foi utilizado essa linguagem juntamente com o *framework* denominado *Bootstrap* que contém diversos tipos de templates baseados em HTML e CSS para várias funções e componentes, ou seja, uma auxiliadora no desenvolvimento front-end. Para Souza(2019), Bootstrap é um framework front-end usado para desenvolver aplicações web e sites *mobile-first*, ou seja, com o layout adaptado à tela do dispositivo utilizado pelo usuário, utilizado em aplicações front-end, ou seja, na camada de interface com o usuário para o desenvolvimento de aplicações adaptáveis à tela de qualquer dispositivo. Por isso, ele conta com diversos recursos para configurar os estilos dos elementos da página de forma simples e eficiente, além de facilitar a construção de páginas que sejam, ao mesmo tempo, adaptadas para web e para dispositivos móveis. No sistema, o Bootstrap foi amplamente utilizado, incluindo botões de navegação, layout, carrosel de imagem, apresentação de imagens e etc. E uma de suas principais aplicações no sistema foi a geração das páginas responsivas. Isso permitiu que a interface do usuário fosse otimizada para qualquer tamanho de tela, desde dispositivos móveis de quaisquer sistemas operacionais até televisores. Fora o grande acervo de material que a própria comunidade distribui gratuitamente.

5.2.2 Autenticação e usuários

O sistema possui dois perfis os quais são autenticados de maneiras diferentes. Para acesso geral e principal do propósito do sistema, há o perfil de usuário, o qual tem acesso à todas as áreas do site exceto a área

administrativa. Sua autenticação se dá por meio de um único cadastro de dados pessoais e posteriormente por login de usuário e senha, também únicos no sistema. Para finalização de um pedido, é necessário a autenticação do usuário.

O outro perfil ao qual o sistema possui, é o perfil administrativo. Sua autenticação se dá por meio de um login padrão contendo um usuário e senha do sistema o qual é definido pelo desenvolvedor e somente esses dados de login dão acesso à essa área. O perfil de administrador possui acesso somente à página inicial e à página administrativa o qual possui todas as áreas de cadastro de produtos, visualização de relatórios, pedidos feitos e pedidos em aberto.

5.2.3 Acesso à dados

Para armazenamento, leitura, atualização de dados e exclusão dos mesmos foi usado o SQL, uma linguagem de banco de dados que possui a função de gerenciar tais dados.

O banco de dados relacional utilizado foi o MySql que modela tudo o que se refere à cliente-servidor. Segundo Longen (2019), o MySql tem a função de criar o banco de dados para que se manipule os dados e faça o devido armazenamento a partir da definição das tabelas. Com isso, a partir de comandos SQL a aplicação do servidor responde com a informação solicitada pelo comando.

A utilização portanto no sistema é na leitura, edição, gravação e exclusão de dados como produtos e suas respectivas informações, pedidos e suas respectivas informações.

5.2.4 JavaScript

O Javascript é uma linguagem de programação de comportamento que permite a criação de conteúdos dinâmicos, animações para deixar seu o site mais interativo e com maior aproveitamento de elementos da interface com usuário seja qual for o perfil (LONGEN, 2011).

Sua utilização no sistema se deve ao fato da linguagem dar uma resposta mais imediata por não precisar fazer uma verificação no servidor de banco de dados, o que demandaria um maior tempo para processar a informação. No sistema há o uso de tal linguagem na verificação de campos de formulários para ver por exemplo se senhas coincidem ou se não há caracteres proibidos nos campo e o bloqueio das mesmas.

Abaixo há uma função JavaScript na qual é utilizado no sistema ao realizar um cadastro de usuário no sistema. Há dois campos no formulário que o usuário deve digitar a senha no primeiro campo e repeti-la no segundo campo. Ao enviar a submissão do formulário, primeiramente é feito a validação das senhas, se elas estão iguais nos dois campos. A função capta o que foi digitado nos dois campos, se for igual, segue a compilação, se forem diferentes, é exibido na tela uma mensagem de erro.

```
function validacao() {


        senha = document.getElementById("pSenha").value;
        senha2 = document.getElementById("pSenha2").value;

        if (senha != senha2) {
          span = document.getElementById("SenhaIncorreta");
          // alert("Confirmação de Senha Incorreta");
          span.innerHTML = "Senhas não conferem!"
          document.form1.pSenha.focus();
          return false;
        }
      }
```

5.2.4 CSS

Segundo Gonçalves (2019), CSS é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

No sistema, o Css foi utilizado para estilizar diversas páginas. O código abaixo mostra uma sintaxe a ser aplicada no botão de submissão do formulário de login. Cada comando estiliza um elemento do botão, ou seja, é composto por vários comandos.

```
.ghost-round {

  cursor: pointer;
  background: none;
  border: 1px solid rgba(255, 255, 255, 0.65);
  border-radius: 25px;
  color: rgba(255, 255, 255, 0.65);
  -webkit-align-self: flex-end;
  -ms-flex-item-align: end;
  align-self: flex-end;
  font-size: 19px;
  font-size: 1.2rem;
  font-family: roboto;
  font-weight: 300;
  line-height: 2.5em;
  margin-top: auto;
  margin-bottom: 25px;
  -webkit-transition: all .2s ease;
  transition: all .2s ease;

}
```

O código abaixo, ilustra um exemplo da sintaxe com o elemento *hover*, porém quando há um *hover* no botão de submissão, ou seja, quando o usuário passa o mouse pelo botão. É reaproveitado toda a sintaxe anterior com a adição e/ou substituição de comandos.

```
.ghost-round:hover {
  background: rgba(255, 255, 255, 0.15);
  color: #fff;
  -webkit-transition: all .2s ease;
  transition: all .2s ease;
}
```

5.3 Modelagem Física do Banco de Dados

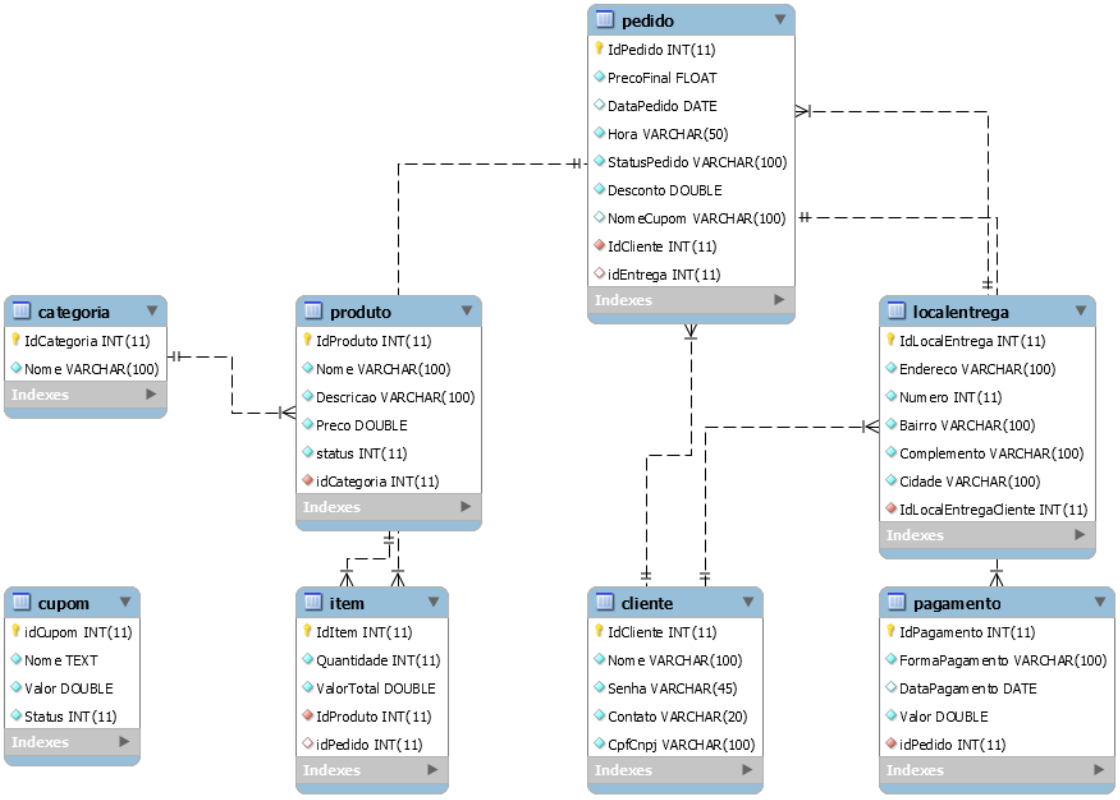


Figura 10.: Modelo entidade relacionamento.

Nos itens a seguir são detalhadas as tabelas do banco, exibindo cada atributo com seu tipo, tamanho, restrição e chave de acesso.

Quadro 1.: Siglas de nomenclaturas

Símbolo	Restrição	Descrição
pk	Chave primária (`primary key`)	Campo(s) que identifica, de forma única, cada registro da tabela.
fk	Chave estrangeira (`foreign key`)	Campo que estabelece o relacionamento entre tabelas. Este campo é originalmente chave primária na tabela de origem.
nn	Preenchimento obrigatório (`not null`)	Propriedade do campo, que define a obrigatoriedade do preenchimento. Significa que o campo não aceita nulo.

FONTE: PUGA et al (2013)

Quadro 2.: Modelo Físico de Dados: Categoria.

Categoria				
Descrição	Entidade responsável por armazenar os dados das categorias de produtos.			
Nome da tabela	categoria			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
idCategoria	INTEGER	5	nn, pk	Identificador da tabela categoria.
nome	VARCHAR	100	nn	Nome da categoria.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
idCategoria	pkCategoria		Chave primária.	

Quadro 3.: Modelo Físico de Dados: Cliente.

Cliente				
Descrição	Entidade responsável por armazenar os dados dos clientes.			
Nome da tabela	cliente			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
idCliente	INTEGER	5	nn, pk	Identificador da tabela cliente.
nome	VARCHAR	100	nn	Nome do cliente.
senha	VARCHAR	45	nn	Senha de acesso do cliente.
contato	VARCHAR	20	nn	Telefone de contado do cliente.
cpfCnpj	VARCHAR	20	nn	Cpf do cliente ou Cnpj da empresa.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
idCliente	pkCliente		Chave primária.	

Quadro 4.: Modelo Físico de Dados: Cupom.

Cupom				
Descrição	Entidade responsável por armazenar os cupons cadastrados no sistema.			
Nome da tabela	Cupom			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdCupom	int	5	nn, pk	Identificador da tabela cupom
Nome	text	256	nn	Nome do cupom.
Valor	double	5	nn	Valor em porcentagem do cupom.

Status	int	11	nn	Status do cupom.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
IdCupom	pkCupom		Chave primária.	

Quadro 5.: Modelo Físico de Dados: LocalEntrega.

LocalEntrega				
Descrição	Entidade responsável por armazenar os endereços cadastrados dos clientes.			
Nome da tabela	LocalEntrega			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdLocalEntrega	int	5	nn, pk	Identificador da tabela localEntrega
Endereco	varchar	100	nn	Nome do endereço.
Numero	int	11	nn	Número do endereço
Bairro	varchar	100	nn	Bairro da entrega.
Complemento	varchar	100	nn	Informação adicional para entrega.
Cidade	varchar	100	nn	Nome da cidade
IdLocalEntregaCliente	int	5	Nn, fk	Identificador da tabela Cliente.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
IdLocalEntrega	pkCliente		Chave primária.	
IdLocalEntregaCliente	fkLocalEntregaCliente		Chave estrangeira para a tabela Cliente.	

Quadro 6.: Modelo Físico de Dados: Item.

Item				
Descrição	Entidade responsável por armazenar os itens de um pedido feito pelo cliente.			
Nome da tabela	Item			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdItem	int	5	nn	Identificador da tabela item.
Quantidade	int	11	nn	Quantidade de itens.
ValorTotal	double	11	nn	Total do valor dos itens.
IdProduto	int	11	nn, fk	Identificador da tabela Produto.
IdPedido	int	11	nn, fk	Identificador da tabela Pedido.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
IdItem	pkItem		Chave primária.	
IdProduto	fkProduto		Chave estrangeira para a tabela	

		Produto.
IdPedido	fkPedido	Chave estrangeira para a tabela Pedido.

Quadro 7.: Modelo Físico de Dados: Pedido.

Pedido				
Descrição	Entidade responsável por armazenar os pedidos registrados no sistema.			
Nome da tabela	Pedido			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdPedido	int	-	nn, pk	Identificador da tabela pedido.
PrecoFinal	float	-	nn	Preço final da compra.
DataPedido	date	-	-	Data do pedido.
Hora	varchar	50	nn	Hora do pedido.
StatusPedido	varchar	100	nn	Status referente ao pedido.
Desconto	double	-	nn	Desconto aplicado no valor total da compra.
IdCliente	int	11	nn, fk	Identificador da tabela Cliente.
IdEntrega	int	11	nn, fk	Identificador da tabela localEntrega.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
IdPedido	pkPedido		Chave primária.	
IdCliente	fkCliente		Chave estrangeira para a tabela Cliente.	
IdEntrega	fkLocalEntrega		Chave estrangeira para a tabela LocalEntrega.	

Quadro 8.: Modelo Físico de Dados: Produto.

Produto				
Descrição	Entidade responsável por armazenar os produtos no sistema.			
Nome da tabela	Produto.			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdProduto	int	-	nn, pk	Identificador da tabela produto.
Nome	varchar	100	nn	Nome do produto.
Descricao	varchar	100	-	Descrição do produto.
Preco	double	-	nn	Preço do produto.
Status	int	11	nn	Status referente ao produto se ele está ativo ou inativo.
IdCategoria	int	11	nn	Identificador da tabela Categoria.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	

IdProduto	pkProduto	Chave primária.
IdCategoria	fkCategoria	Chave estrangeira para a tabela Categoria.

Quadro 9.: Modelo Físico de Dados: Pagamento.

Pagamento				
Descrição	Entidade responsável por armazenar os dados de pagamentos de pedidos.			
Nome da tabela	Pagamento.			
Tempo de retenção	Permanente			
Rotina de limpeza	Não se aplica			
ATRIBUTOS				
Atributo	**Tipo de dado**	**Tamanho**	**Restrição**	**Descrição**
IdPagamento	int	-	nn, pk	Identificador da tabela pagamento.
FormaPagamento	varchar	100	nn	Meio de pagamento do pedido
DataPagamento	date	-	-	Data realizada do pagamento.
Valor	double		nn	Valor referente ao pagamento.
IdPedido	int	11	nn	Identificador da tabela Pedido.
CONSTRAINTS				
Atributo(s)	**Nome Restrição**		**Observações**	
IdPagamento	pkPagamento		Chave primária.	
IdPedido	fkCategoria		Chave estrangeira para a tabela Pedido.	

5 PROCESSO DE TESTES

Segundo o Plano de Testes delineado na seção 3.5, o processo de teste que será realizado em todo o software é o Teste Caixa Preta ou Funcional que segundo Pressman (2011) consistem em validar os requisitos funcionais do software. Além deste, será realizado também o teste de Caixa branca na qual se possui acesso ao código fonte a ponto de conhecer a estrutura interna do projeto.

Dessa forma, os testes unitários foram aplicados logo após cada implementação de cada funcionalidade à que ela se propõe, além disso foram testados os feedbacks dados na interface do software.

6 IMPLEMENTAÇÃO E TESTES

Neste capítulo serão apresentados os detalhes de implementação e testes do software Sistema web delivery para padarias. A implementação das funcionalidades se deu através da análise da especificação de cada caso de

uso, seguindo assim as prioridades funcionais para a escrita do código conforme a documentação.

7.1 Funcionalidades Implementadas

Como citado anteriormente, o software foi desenvolvido para atender às necessidades funcionais segundo a especificação dos casos de uso.

7.2. Alocação dos fragmentos de código aos elementos arquiteturais

A Tabela 18 ilustra uma descrição sucinta da alocação dos módulos aos elementos da arquitetura apresentada no capítulo 5 deste documento.

Caso de Uso	Elemento Arquitetural	Módulo
Model	Cadastro	classes/cadastro.inc
	Cupom	classes/cupom.inc
	Endereço	classes/endereco.inc
	Item	classes/item.inc
	Pagamento	classes/pagamento.inc
	Pedido	classes/pedido.php
	Produto	classes/produto.inc
Control	Cadastro	controlers/controlerCadastro.php
	Carrinho	controlers/controlerCarrinho.php
	Cupom	controlers/controlerCupom.php
	Endereço	controlers/controlerEndereco.php
	Pagamento	controlers/controlerPagamento.php
	Pedidos feitos	controlers/controlerPedidosFeitos.php
	Produto	controlers/controlerProduto.php
View	Cliente	/cadastro.php
		/formClienteAtualizar.php
		/verComprasCliente.php
		/clientesCadastrados.php
	Categoria	/cadastroProduto.php
	Cupom	/cadastrarCupons.php
		/verCupons.php
	Endereco	/formClienteEndereco.php
		/formClienteEnderecoCarrinho.php
		/formEnderecoAtualizar.php
	Pagamento	/pagarNaEntrega.php
	Pedido	/escolherPedido.php
		/fazerPedido.php

		/finalizarCompra.php
		/meusPedidosCliente.php
		/ultimosPedidos.php
		/pedidosEntreguesPaginacao.php
	Produto	/cadastroProduto.php
		/formProdutoAtualizar.php
		/lixeiraProdutos.php

Tabela 18.: Alocação dos Módulos aos Elementos Arquiteturais.

6.3 Implementação das Interfaces

Inicialmente temos um modo de cabeçalho, em que não há usuário logado, conforme a figura 11, tem-se recursos Css para efeito visuais e animações ao clique no campo. O sistema ao receber o clique no botão “Login”, faz a verificação se os campos informados estão cadastrados no banco de dados. É usado tanto para acesso administrativo quanto para acesso geral.

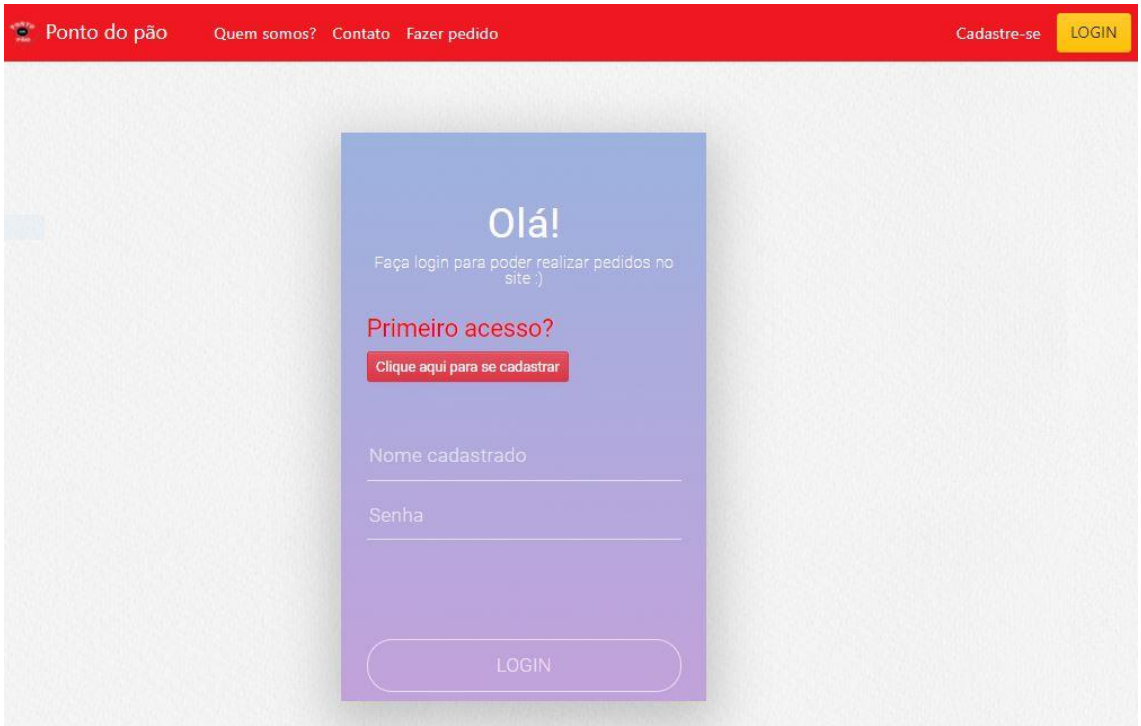


Figura 11.: Tela de Login.

Na figura 12 temos a listagem de todas as categorias cadastradas no sistema com a condição de terem produtos cadastrados na mesma. Há o uso

do framework Bootstrap para os chamados cards. Um card é um container de conteúdo flexível e extensível. Todos eles sempre responsivos.

Figura 12.: Tela da listagem das categorias.

Na figura 13 há a listagem de todos os produtos cadastrados no sistema da categoria selecionada pelo usuário. Há o uso do CSS para a formatação dos cards na tela. No cabeçalho, mostra-se o modo quando há usuário logado no sistema, com a liberação de botões e personalização de acordo com o nome do usuário.

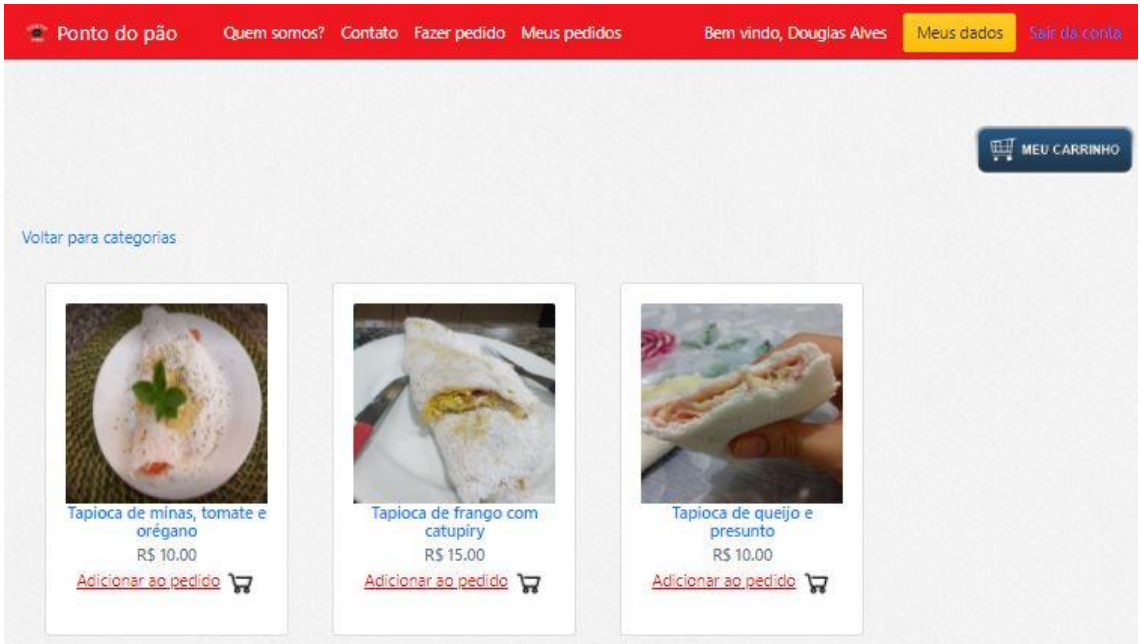


Figura 13.: Tela de produtos de uma categoria selecionada.

Na figura 14 tem-se a listagem dos produtos que o usuário vai escolhendo para o carrinho. Há o uso do comando “table-responsive-sm table-striped” que é um recurso Bootstrap para criação de tabelas responsivas com tamanho definido no comando e a listragem de cores em cada linha.

Figura 14.: Tela da tabela do pedido.

Na seguinte interface ilustrada na Figura 15, tem-se os detalhes de um pedido que foi feito com recursos Bootstrap para a geração das tabelas. Nota-se no cabeçalho o modo administrativo, aonde todas opções do menu são de uso restrito com acesso somente do administrador.

Figura 15.: Tela de detalhes de um pedido feito.

Na Figura 16 tem-se a listagem de todos usuários cadastrados no sistema e uma filtragem de visualização para o mesmo

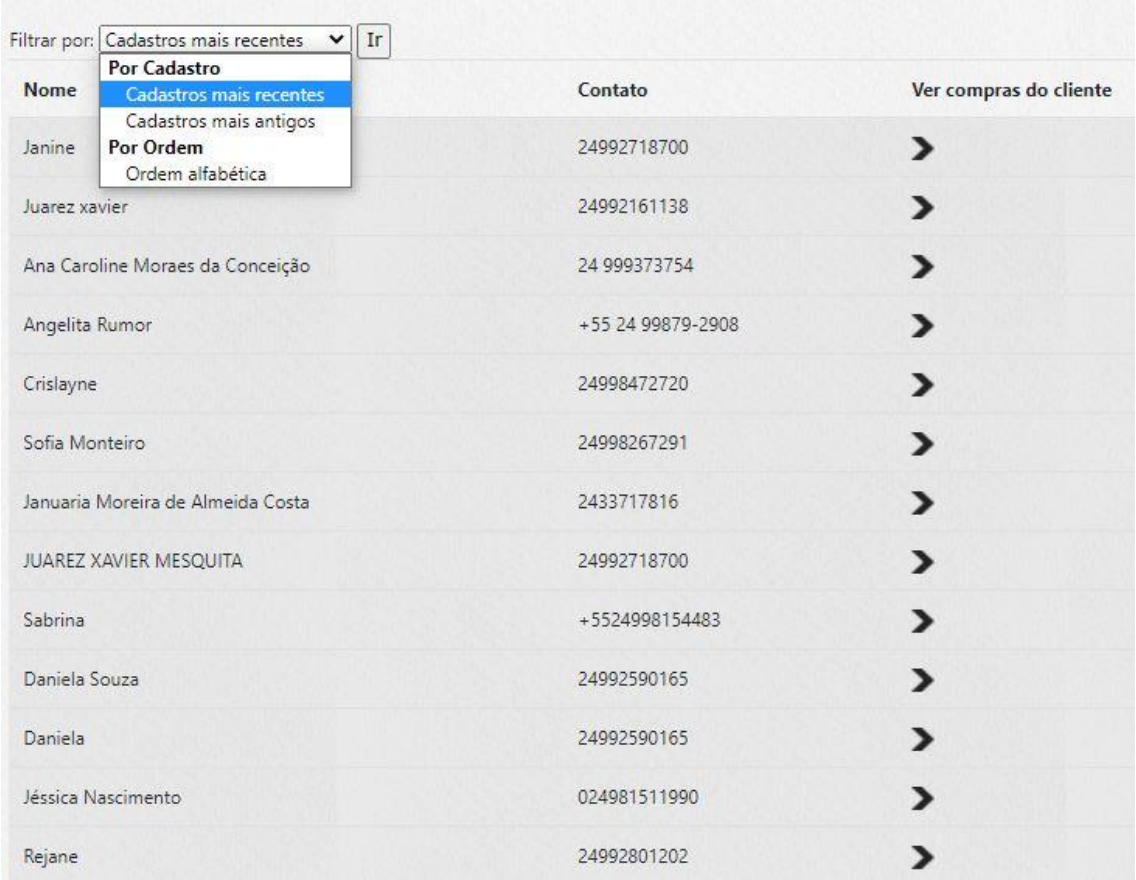


Figura 16.: Tela da listagem de clientes cadastrados no sistema.

.

Na representação da Figura 17, tem-se o formulário de cadastro de usuário visto na forma responsiva, ou seja, a tela se adapta a qualquer resolução de tela. Isso é possível pelo recurso de tabela que o Bootstrap proporciona.

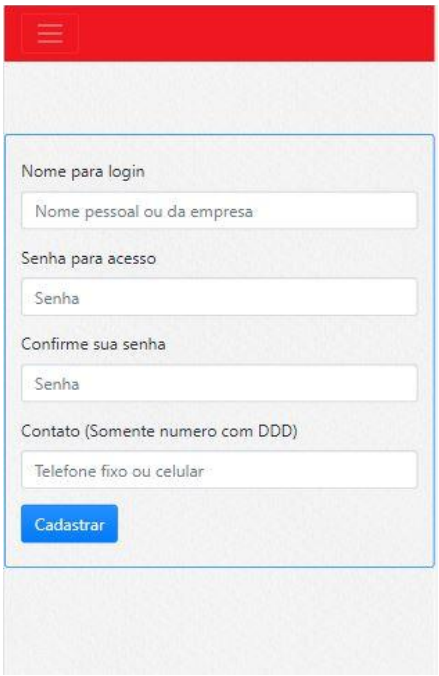


Figura 17.: Tela de cadastro de usuário sob a ótica de um dispositivo *mobile*.

6.4 Testes

Foram aplicados dois tipos de testes: usabilidade, e além disso, para as principais funcionalidades foram executados testes extras com a utilização da ferramenta de teste automatizado usando PHP 7.3.3 com Xdebug 2.9.8 e PHPUnit 9.4.2.

O PhpUnit é um framework de teste de unidade para o Php utilizado para escrever e rodar testes automatizados. Com ele, podemos verificar se o projeto se comporta como o esperado e a partir de vários fragmentos de código executável podemos testar se as unidades do projeto estão corretas, dando uma maior confiança e diminuindo custos de reparo e às vezes até um reparo tardio do projeto.

O XDebug é uma extensão Php que fornece recursos de depuração, incluindo recursos visuais como relatórios e gráficos detalhados sobre a bateria de testes realizados. Um importante dado que ele proporciona é o da complexidade ciclomática, que segundo McCabe (1976), é uma medida de software para medir a complexidade de um programa (classe, método, rotina etc). Pode-se dizer que ela indica a dificuldade de se construir testes de unidade em um determinado código uma vez que ela mede a quantidade de caminhos linearmente independentes neste código. De acordo com Pressman (2006), caminho Linearmente Independente é qualquer caminho do programa que introduz pelo menos um novo conjunto de comandos de processamento ou uma nova condição. Quando definido em termos de grafo de fluxo, um caminho independente deve incluir pelo menos uma aresta que não tenha sido atravessada antes de o caminho ser definido.

A seção 7.4.1 apresenta um resumo dos resultados dos testes de usabilidade realizados, as datas de realização e o status aprovado para os testes que se comportaram adequadamente (com sucesso) e reprovado para os que precisariam de maior refinamento.

6.4.1 Teste de usabilidade

O sistema foi projetado tendo como ideia base uma futura aplicação em uma padaria do mundo real, e pensando nisso, os testes mais importantes foram realizados com os usuários finais reais dessa ideia de aplicação do sistema. Têm-se como usuários finais, clientes do sistema, pessoas que utilizam computador ou celular para realização de pedidos via *delivery.*

O primeiro teste de usabilidade do sistema aconteceu em novembro de 2019. Para esse primeiro teste foram convidadas três pessoas das quais já possuíam familiaridade com tecnologia e eram estudantes de sistemas de informação.

Para a realização desse teste foi solicitado o fluxo principal do sistema, o qual é a realização de um pedido pelo sistema e solicitado também a gravação das telas enquanto os usuários acessavam pela primeira vez.

Com esse primeiro teste pôde-se concluir que os usuários não encontravam com rapidez a função de realizar pedido, a disposição dos botões não estava amigável e houve discordância quanto ao redirecionamento de páginas. Todos os tópicos citados foram posteriormente reparados e aprovados por quem participou do teste.

No segundo teste de usabilidade do sistema foram convidadas três pessoas das quais possuíam pouca familiaridade com tecnologia e eram parte do público alvo final.

Para a realização desse teste foi solicitado o fluxo principal do sistema, o qual é a realização de um pedido pelo sistema, porém nesse fluxo se passa pelo cadastro no sistema de usuário e endereço e solicitado também a gravação das telas enquanto os usuários acessavam.

Com este segundo teste pôde-se notar que os usuários não concordaram com a disposição dos elementos das páginas quando o mesmo era visto sob a ótica *mobile*. Com isso, concluiu-se que os usuários finais, possuem a tendência de acessar o sistema via dispositivo móvel, o que acarretou numa geral reparação de quase todas as páginas do sistema para que se adequasse à um conforto visual na experiência no site. Foram posteriormente reparados e aprovados por quem participou do teste.

No terceiro teste de usabilidade foram convidadas cinco pessoas das quais estavam munidas de dispositivos móveis para acesso ao sistema.

Para a realização desse teste foi solicitado o fluxo principal do sistema, o qual é a realização de um pedido pelo sistema, e após isso o uso livre do sistema.

Neste terceiro teste que ocorreu de maneira presencial e com feedbacks instantâneos, foi unânime entre os participantes a opinião de que os botões do sistema não estavam uniformes (seguindo um padrão, modelo e cor) nas páginas do sistema. Por conseguinte, foi reparado e implementado os próprios botões modelos do *Bootstrap*.

No quarto teste de usabilidade foram convidadas oito pessoas considerados usuários finais, sendo três deles os responsáveis pela aplicação do sistema em sua padaria.

Para a realização desse teste, também foi solicitado o fluxo principal do sistema. Os feedbacks colhidos foram totalmente positivos, visto que todos os usuários do teste conseguiram realizar todas as operações com sucesso.

6.4.2 Testes Funcionais

A Figura 54 mostra o resumo de todos os testes realizados na ferramenta. Foram executados um total de 11 arquivos de testes contendo 43 testes no total, todos envolvendo as classes correspondentes à Model e teve uma duração de 00.057segundos.

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total		97.26%	71 / 73		96.00%	48 / 50		75.00%	6 / 8
cadastro.php		100.00%	9 / 9		100.00%	5 / 5		100.00%	1 / 1
cupom.php		100.00%	7 / 7		100.00%	4 / 4		100.00%	1 / 1
endereco.php		75.00%	3 / 4		75.00%	3 / 4		0.00%	0 / 1
item.php		100.00%	13 / 13		100.00%	9 / 9		100.00%	1 / 1
pagamento.php		100.00%	9 / 9		100.00%	5 / 5		100.00%	1 / 1
pedido.php		83.33%	5 / 6		83.33%	5 / 6		0.00%	0 / 1
produto.php		100.00%	25 / 25		100.00%	17 / 17		100.00%	2 / 2

Legend

Low: 0% to 50% **Medium**: 50% to 90% **High**: 90% to 100%

Generated by php-code-coverage 9.2.3 using PHP 7.3.3 with Xdebug 2.9.8 and PHPUnit 9.4.2 at Thu Nov 5 13:21:30 CET 2020.

Figura 18.: Resumo de todos os testes.

Após o término da execução dos casos de testes, os dois erros encontrados e apresentados no quadro 28 foram corrigidos.

Arquivo	Erro	Ajuste
Cupom	Estava sendo aceito cupons com valores de zero por cento de desconto	Foi criado uma verificação ao submeter os dados do formulário de criação de cupom
Produto	Estava sendo aceito produtos com preço de valor zero	Foi criado uma verificação ao submeter os dados do formulário de criação de produto

Tabela 19.: Descrição dos erros encontrados.

Segundo TEDESCO (2017) Complexidade ciclomática é uma métrica do campo da engenharia de software, quee serve para mensurar a complexidade de um determinado módulo (uma classe, um método, uma função e etc), a partir da contagem do número de caminhos independentes que ele pode executar até o seu fim. Um caminho independente é aquele que apresenta pelo menos uma nova condição (possibilidade de desvio de fluxo) ou um novo conjunto de comandos a serem executados.

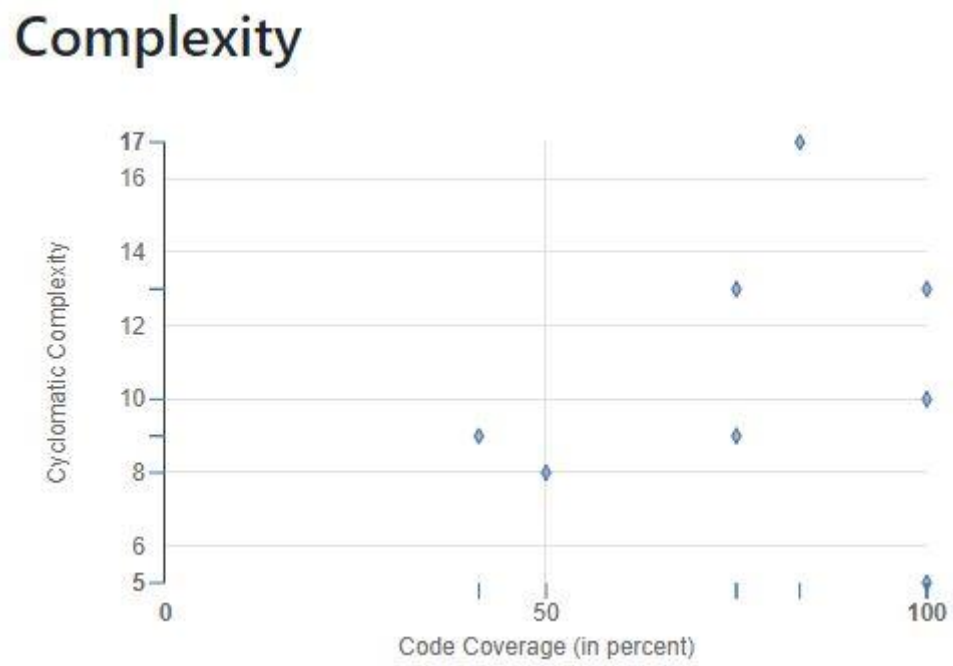


Figura 19.: Complexidade ciclomática.

6.4.3 Resumo dos Testes

Os resultados obtidos foram satisfatórios e todos aprovados. A opinião de usuários no teste de usabilidade, mudou drasticamente o projeto, visto que o mesmo não estava preparado para uso em dispositivos móveis. Foi verificado o anseio por essa responsividade já que todos os feedbacks apontavam para uso em celulares e não em computadores.

8 Considerações finais

O presente trabalho reúne a modelagem e documentação realizada para o posterior desenvolvimento do sistema *delivery* para padarias. Sendo um sistema genérico (ou seja, aplicado à maioria das panificadoras), teve como iniciativa de desenvolvimento abranger estabelecimentos comerciais que tenham o mesmo ramo, o da panificação. Tendo como objetivo principal de sistematizar e implementar um processo de serviço de entrega de produtos que a própria padaria vende a fim de obter um ganho financeiro para a empresa e maior apreço por parte de seus clientes.

A ideia do presente trabalho partiu da experiência pessoal e familiar com o ramo da panificação, pelo aluno autor. E durante todo o tempo, deu-se a percepção do que poderia ser feito, o que facilitou o levantamento dos requisitos. O sistema inicialmente possuía o objetivo de automatizar o controle de pedidos e os gerenciar, abandonando o processo informal do mesmo. Todos objetivos iniciais foram atendidos na implantação do sistema e uso por parte dos usuários.

Em abril de 2020 o sistema foi implantado em uma padaria na cidade de Paraty-RJ, logo no início dos decretos municipais advindos da pandemia do Covid-19. Com todas as restrições impostas, viu-se a alternativa proposta neste trabalho e por consequência foi elaborado um plano de marketing e divulgação do site que entraria no ar. Logo na primeira semana, o sistema se comportou como esperado e atendeu perfeitamente seus usuários em todos seus formulários e realização de pedidos, ou seja, cumpriu seu objetivo em todo o tempo em que esteve no ar. Esteve em operação por quatro semanas, e foi posto por esse tempo de operação para contornar os prejuízos causados

pelos decretos em razão da pandemia do Covid-19. O sistema teve uma aceitação bastante satisfatória baseado na opinião pessoal dos usuários do sistema e entrevistas informais com os mesmos, não havendo sugestões do mesmo, somente na parte operacional que ficou a encargo da própria padaria a qual o administrava.

Um dos usuários assíduos do sistema foi o Victor de Souza Guerra Vieira que afirma que “em época de pandemia, ter um aplicativo fácil de usar e focado em fazer o simples e necessário foi de extrema importância para nós, moradores de Paraty. A destreza e compreensibilidade na execução dos comandos foi uma sacada incrível e que simplesmente qualquer pessoa, em qualquer nível "tecnológico" pode utilizar. Sem falar que quem é cliente como eu há 3 anos da Padaria, gostou mais ainda por ter a prontidão de ter todos os produtos fornecidos na "palma das mãos". Indico o app e Super indico os produtos!”. Outro usuário foi a Djhennyfer da Cruz que diz que “foi de ótima qualidade e facilidade o aplicativo , com a pandemia e o decreto proibindo o fluxo nas ruas o aplicativo da padaria ajudou muito facilitando o consumo. Fácil acesso e agilidade no atendimento”. Já a cliente e usuária do sistema Sabrina Lessa afirma que possui uma forte resistência à novas tecnologias mas conseguiu obter a facilidade que o sistema proporcionou.

Foi incrível a funcionalidade do sistema, de fácil compreensão e acesso, e também de resposta rápida.

Foram realizados um total de 24 pedidos e 16 cadastros de clientes durante o período em operação. Havia maior uso do sistema quando haviam promoções via cupom, a qual eram cadastrados pelo próprio administrador. Houve uma visibilidade maior do estabelecimento aos moradores locais, já que muitos não conheciam a padaria ou só tinham “ouvido falar”, e com o site puderam conhecer um pouco mais do estabelecimento. Mesmo fora de operação, esses mesmos clientes começaram a frequentar a padaria presencialmente.

A importância desse trabalho está na visão possibilitada ao aluno da prática dos conteúdos aprendidos durante o curso, assim como, foram abertas novas pesquisas em novas fontes de conhecimento para desenvolver o projeto,

desde a fase inicial até a validação do sistema. Tais pesquisas possibilitaram ao aluno autor um aprofundamento na área de desenvolvimento web, e a partir destas, abriu-se um “leque” de opções derivativas já que os recursos utilizados são os mesmos e o aprendizado de algo amplia a novos conhecimentos e novas técnicas, na formação profissional do aluno.

Durante todo o desenvolvimento, podem-se destacar as fases de testes e cálculo do esforço de trabalho, onde foram identificadas as maiores dificuldades em realiza-los. Em relação aos testes, a dificuldade maior foi encontrar uma ferramenta para realizar os procedimentos assim como as mudanças no projeto a partir dos feedbacks na mesma fase. Para o contorno de tais problemas, foram necessários maior tempo de pesquisa na busca por tais conhecimentos, já que por exemplo ferramentas de teste existem de formas variadas. E na busca por tais ferramentas, algumas foram testadas para serem averiguadas aquelas que se encaixariam mais no propósito dos testes do sistema e que retornaria resultados relevantes (e até visuais, no ponto de vista de apresentação) como os do gráfico na seção 7.4.2.

A relevância desse sistema está na possível inovação no tocante ao controle de pedidos e entregas de panificadoras, ainda escasso em sua maioria, bem como no aumento da eficiência destas, pois panificadoras que se enquadrem nas características e regras de negócio atendidas pelo sistema, aumentará a eficiência de suas atividades, culminando em um novo setor que virá a ser um potencial ganho financeiro e captação de clientes.

Para melhoria do presente sistema, pode-se haver o uso de novos módulos e frameworks tais como pagamento online pelo próprio site, geolocalização da entrega, chat direto com o estabelecimento por parte do usuário e descrição de experiências pessoais com a entrega. O próximo passo é a criação de um aplicativo *web-view* que faz com que o aplicativo do estabelecimento fique disponível para download na loja de aplicativos do sistema operacional do celular e ao instalar, consequentemente permanece no celular do usuário. Outro ponto a ser adicionado, é a inclusão de notificação de que um pedido novo foi incluído. E a partir desse projeto, pode haver uma

adaptação para outros ramos de estabelecimentos comerciais, tais como lanchonetes, mercados ou até mesmo bares.

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAHAMSSON, Pekka et al. **New directions on agile methods: a comparative analysis.** 25th International Conference On Software Engineering, 2003. Proceedings., [s.l.], p.244-254, 28 maio 2003. IEEE. http://dx.doi.org/10.1109/icse.2003.1201204.

DEVMEDIA. **O que são testes unitários?** 2018. Disponível em: https://www.devmedia.com.br/e-ai-como-voce-testa-seus-codigos/39478. Acesso em: 05 novembro 2020.

FALBO, Ricardo et al. **Um Processo de Engenharia de Requisitos Baseado em Reutilização e Padrões de Análise.** VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'07), Lima, Peru, 2007.

GUEDES, Gilleane T. A. **UML 2 – Uma abordagem prática.** Ed Novatec, 2009.

GONÇALVES, Ariane. **O que é Css? Guia básico para iniciantes**, 2019. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/. Acesso em 16 novembro de 2020

IFPUG. **About Function Point Analysis.** Disponível em: <http://www.ifpug.org/about-function-point-analysis/>. Acesso em: 05 novembro 2020.

LONGEN, Andrei. **O que é MySql? Guia para iniciantes,** 2019. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-mysql/. Acesso em: 10 novembro 2020

MCCABE, Thomas. "A Complexity Measure", IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, dezembro 1976, disponível em http://www.literateprogramming.com/mccabe.pdf.

MOLSKI, Fernando Ricardo. **Customização de software com ênfase na arquitetura MVC e Struts.** DevMedia, s.d. Disponível em: Acesso em: 10 set. 2017.

OGLIO, Pablo D. **PHP Programando com Orientação a Objetos.** São Paulo: Novatec, 2007.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: fundamentos, métodos e padrões.** São Paulo: LTC Editora, 2000.

PHPUNIT. The Php testing framework. Página inicial. Disponível em: https://phpunit.de/. Acesso em: 17 novembro 2020.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: Uma Abordagem Profissional.** 7. ed. Porto Alegre: AMGH, 2011.

PRESSMAN, Roger. **Engenharia de Software**. 6a edição. São Paulo. McGraw-Hill, 2006.

PUGA, Sandra et al. **Banco de Dados: Implementação em SQL, PL/SQL e Oracle 11g**. 1. ed. São Paulo: Pearson Education do Brasil, 2013.

RAMOS, Allan. **O que é MVC?** 2015. https://tableless.com.br/mvc-afinal-e-o-que/. Acesso em 02 de dezembro de 2019.

RODRIGUES, Márcio. Economia Estadão, **Pesquisa indica que 56% das pessoas fazem pedidos semanais via delivey**, 2016. Disponível em http://economia.estadao.com.br/noticias/suas-contas,pesquisa-indica-que-56-das-pessoas-fazem-pedidos-semanais-via-delivery,10000060142. Acesso em 11 de novembro de 2019.

SANTOS, Luan. **Padaria: Vantagens de ter um sistema para padaria**, 2018. Disponível em https://sischef.com/vantagens-de-ter-um-sistema-para-padaria/. Acesso em 19 de novembro de 2019.

SCRUM. **Scrum.** 2017. Disponível em: https://www.scrum.org/. Acesso em: 26 novembro 2019.

SEBRAE. **Estudo de mercado – Indústria: Panificação,** 2017. Disponível em:https://www.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/BA/Anexos/Ind%c3%bastria%20da%20panifica%c3%a7%c3%a3o.pdf. Acesso em: 12 novembro 2019.

SEBRAE. **Indústria da panificação**, 2018. Disponível em https://m.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/BA/Anexos/Ind%C3%BAstria%20da%20panifica%C3%A7%C3%A3o.pdf. Acesso em 12 de novembro de 2019.

SILVA, V. L. **O grande salto das pequenas. Pequenas empresas, grandes negócios**, São Paulo: Editora Globo, v.IX, n.104, p.86-88, 1997.

SOMMERVILLE, I. **Engenharia de Software.** 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p. Tradução Ivan Bosnic e Kalinga G. de O. Gonçalves.

SOUZA, Ivan. **Bootstrap: Saiba neste guia para iniciantes o que é, por que e como usá-lo,** 2019. Disponível em: https://rockcontent.com/br/blog/bootstrap/. Acesso em: 17 novembro 2020.

TEDESCO, Kennedy. **Complexidade ciclomática, análise estática e fatoração**, 2017. Disponível em: https://www.treinaweb.com.br/blog/complexidade-ciclomatica-analise-estatica-erefatoracao/#:~:text=Complexidade%20ciclom%C3%A1tica%20%C3%A9%20uma%20m%C3%A9trica,executar%20at%C3%A9%20o%20seu%20fim. Acesso em: 16 dezembro 2020.

VAZQUEZ, Carlos Eduardo et al. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software.** 13. ed. Brasil: Érica, 2013.

XDEBUG. XDebug and profiler tool for php. Página inicial. Disponível em: https://xdebug.org/. Acesso em: 17 novembro 2020.

ZUCHER, Vitor. **O que é padrão MVC?**, 2020. Disponível em https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc. Acesso em 18 de dezembro de 2019.

APÊNDICE 1 – CASOS DE USO

Cada caso de uso descrito abaixo está identificado pela sigla UC (*use case*) juntamente com o seu respectivo número e nome.

UC01 - CADASTRAR USUÁRIO

SUMÁRIO: Esse caso de uso serve para a entidade de nome cliente.

ATORES: Usuário.

PRÉ-CONDIÇÕES: Dados do usuário disponíveis.

CURSO NORMAL:

1. O caso de uso inicia quando o cliente necessita fazer o seu cadastro;
2. O usuário seleciona a opção de criar cadastro;
3. Sistema busca e exibe o formulário de cadastro;
4. O usuário preenche os campos: Nome para login, senha, confirmação de senha, contato e cpf/cnpj;
5. O sistema realiza as seguintes validações: Verifica se já não existe o mesmo nome para login cadastrado no sistema, verifica se os dois campos de senhas coincidem e verifica se todos os campos foram preenchidos;
6. O sistema armazena os dados do cadastro;
7. O sistema exibe mensagem de cadastro com sucesso, caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 2 e 4, o usuário abandona a operação; caso de uso termina.

CURSOS DE EXCEÇÃO:

5.1. O Usuário não preencheu os dados obrigatórios. O sistema exibe uma mensagem informando o erro, retornando ao passo 3, solicitando o preenchimento dos campos obrigatórios.

5.1.2 Caso o nome de usuário já esteja cadastrado, o sistema não grava os dados e retorna ao passo 3, informando a necessidade da mudança no nome.

5.1.3 Os dois campos de senha não coincidem. O sistema não grava os dados e retorna ao passo 3, informando a necessidade de dos dois campos estarem idênticos.

PÓS-CONDIÇÕES: Cadastro devidamente gravado no sistema e sistema exibe a página inicial.

REQUISITOS DE INTERFACE: Os campos devem conter identificação sobre para qual campo está ativo no momento. Todos elementos (campos, descrição, link e botão) devem estar centralizados.

REQUISITOS NÃO-FUNCIONAIS: O sistema permitirá que o usuário realize apenas um cadastro por vez.

REGRAS DE NEGÓCIOS: Os dados referentes ao CNPJ/CPF deverão ser controlados de forma que não haja duplicidade de cadastro, ou seja, já que os números são únicos e não repetem.

UC02 - CADASTRAR CATEGORIA DE PRODUTO.

SUMÁRIO: Esse caso de uso serve para a entidade categoria. O caso de uso inicia quando o administrador precisa cadastrar o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado e logado no sistema; Dados de categoria disponíveis.

CURSO NORMAL:

1. O caso de uso inicia quando o administrador necessita realizar o cadastro de categoria;
2. O administrador seleciona a opção de cadastro de produtos
3. O administrador insere no campo categoria o nome a qual deseja adicionar;
4. O sistema valida o campo categoria;
5. O sistema armazena os dados do cadastro;

6. O sistema atualiza a lista de categorias, caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 2 e 3, o administrador abandona a operação; caso de uso termina.

CURSOS DE EXCEÇÃO:

4.1. Caso o nome inserido seja menor que três caracteres, o sistema exibe uma mensagem de erro. Retornando ao passo 3, solicitando o preenchimento correto do campo categoria

PÓS-CONDIÇÕES: Cadastro de categoria devidamente gravado no sistema. Sistema exibe a lista de categorias e produtos atualizados.

REQUISITOS DE INTERFACE: O campo deve conter identificação sobre para qual campo está ativo no momento.

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá que o administrador realize apenas um cadastro por vez.

UC03 - CADASTRAR CUPOM

SUMÁRIO: Esse caso de uso serve para a entidade cupom. O caso de uso inicia quando o administrador precisa cadastrar o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Dados de cupom disponíveis.

CURSO NORMAL:

1. O caso de uso inicia quando o administrador necessita realizar o cadastro de cupom;
2. O administrador solicita buscar cupons cadastrados;
3. O sistema busca e exibe os cupons ativos e desativados;
4. O administrador solicita cadastrar cupom;

5. O administrador preenche os seguintes campos para cadastro: Nome e valor;
6. O sistema valida os campos de nome e valor;
7. O sistema armazena os dados do cupom;
8. O sistema exibe tela de todos os cupons cadastrados, caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 2 e 5, o administrador abandona a operação; caso de uso termina.

CURSOS DE EXCEÇÃO:

5.1. Administrador digita qualquer caractere que não seja número no campo de valor, o sistema não permite sua inserção. Mantendo o caso de uso no mesmo passo.

PÓS-CONDIÇÕES: Cadastro devidamente gravado no sistema. Sistema exibe tela de cupons.

REQUISITOS DE INTERFACE: O campo deve conter identificação sobre para qual campo está ativo no momento.

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá que o administrador realize apenas um cadastro por vez.

UC04 - CADASTRAR ENDEREÇO

SUMÁRIO: Esse caso de uso serve para a entidade localEntrega. O caso de uso inicia quando o usuário precisa cadastrar o mesmo no sistema.

ATORES: Usuário.

PRÉ-CONDIÇÕES: Usuário cadastrado e autenticado no sistema; Dados de endereço disponíveis.

CURSO NORMAL:

1. O caso de uso inicia quando o usuário necessita realizar o cadastro de endereço;
2. O usuário solicita a seção de dados do usuário;
3. O sistema busca e exibe todos os dados de usuários e endereços cadastrados
4. O usuário solicita a adição de endereço;
5. Sistema exibe o formulário;
6. O usuário preenche os campos: Endereço, Bairro, Cidade, Número e Complemento;
7. O sistema valida os campos obrigatórios;
8. O sistema salva o endereço;
9. O sistema exibe todos endereços cadastrados; caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 2 e 5, o usuário abandona a operação; caso de uso termina;

CURSOS DE EXCEÇÃO:

6.1. Administrador digita qualquer caractere que não seja número no campo número, o sistema não permite sua inserção. Mantendo o caso de uso no mesmo passo.

7. Usuário não preencheu todos os campos. Sistema informa o erro ao usuário; Caso de uso retorna ao passo 5.

PÓS-CONDIÇÕES: Endereço devidamente cadastrado. Sistema exibe os dados do usuário.

REQUISITOS DE INTERFACE: O campo deve conter identificação sobre para qual campo está ativo no momento.

REQUISITOS NÃO-FUNCIONAIS: o sistema não permitirá que o usuário cadastre vários endereços por vez.

UC05 - CADASTRO PRODUTO

SUMÁRIO: Esse caso de uso serve para a entidade produto. O caso de uso inicia quando o administrador precisa cadastrar o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador cadastrado e autenticado no sistema; Dados do produto disponível.

CURSO NORMAL:

1. O caso de uso inicia quando o administrador necessita realizar o cadastro de produto;
2. O administrador seleciona a opção de cadastro de produtos
3. O administrador insere os campos: Produto, categoria, descrição e Valor unitário;
4. O sistema valida o campo de preço;
5. O sistema armazena os dados do produto;
6. O sistema atualiza a lista de produtos, caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 1 e 2, o administrador abandona a operação; caso de uso termina.

CURSOS DE EXCEÇÃO:

4.1. O campo de preço não aceita inserção de valores reais negativos; Caso de uso retorna ao passo 3.

PÓS-CONDIÇÕES: Cadastro de produto devidamente gravado no sistema. Sistema exibe a tela de cadastro de produtos.

REQUISITOS DE INTERFACE: O campo deve conter identificação sobre para qual campo está ativo no momento.

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá que o administrador realize apenas um cadastro por vez.

UC06 - FAZER PEDIDO

SUMÁRIO: Esse caso de uso serve para as seguintes entidades: pedido e item. O caso de uso inicia quando o usuário deseja fazer uma compra.

ATORES: Usuário

PRÉ-CONDIÇÕES: Produto(s) cadastrado(s) no sistema assim como sua respectiva categoria;

CURSO NORMAL:

1. O usuário solicita tela de pedido;
2. O sistema mostras todas categorias existentes para escolha;
3. O usuário seleciona a categoria;
4. O sistema mostras todos os produtos existentes de acordo com a escolha;
5. O usuário escolhe o produto;
6. O sistema exibe o carrinho de compras
7. O usuário insere a quantidade do produto;
8. O usuário solicita atualização do produto;
9. O sistema guarda a preferência e exibe o carrinho atualizado;
10. O usuário solicita finalização da compra;
11. Sistema salva o carrinho e exibe o checkout;
12. Usuário seleciona o endereço para entrega;
13. O sistema guarda a preferência e exibe checkout bem como o método de pagamento;
14. Usuário seleciona o método de pagamento;
15. Sistema salva todos os dados, exibe mensagem de compra finalizada;

CURSO ALTERNATIVO:

Entre os passos 1 e 3, o usuário abandona a operação; caso de uso termina.

10.1. O usuário deseja adicionar mais produtos ao carrinho; Caso de uso retorna ao passo 2;

14.1. O usuário escolhe a forma de pagamento em dinheiro; Caso de uso retorna ao passo 15

14.2. O usuário escolhe pagamento via cartão;

14.2.1 Sistema salva a preferência e exibe na tela o formulário para preenchimento dos dados do cartão.

14.2.2 Usuário preenche os campos: Número do cartão, data de validade e número de segurança e salva as alterações; Caso de uso retorna ao passo 15.

CURSOS DE EXCEÇÃO:

8.1. O usuário não solicitou atualização do item; Sistema exibe mensagem de alerta; Caso de uso retorna ao mesmo passo

11.1. O usuário não está logado ao sistema; Sistema exibe tela de login; Caso de uso retorna ao passo 11;

PÓS-CONDIÇÕES: Cadastro devidamente gravado no sistema; Sistema exibe página inicial.

REQUISITOS DE INTERFACE: Ao ser solicitado a mudança na quantidade de um produto no carrinho, o sistema exibe mensagem de alerta informando que o usuário deve solicitar a atualização do produto. As categorias e produtos devem ser mostrados em formato de cartões. Tabela de checkout deve contar cores alternadas por linha (escuro/claro).

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá que o usuário adicione apenas um produto por vez.

UC07 - GERAR RELATÓRIO DE PEDIDOS REALIZADOS PELO USUÁRIO LOGADO

SUMÁRIO: Este caso de uso se inicia quando o usuário deseja gerar um relatório de pedidos feitos.

ATORES: Usuário.

PRÉ-CONDIÇÕES: Usuário cadastrado e autenticado no sistema; Pedidos previamente realizados.

CURSO NORMAL:

1. O usuário solicita tela de pedidos feitos;
2. O sistema busca o relatório ordenado pelos mais recentes;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Data, preço da compra e status; Caso de uso termina.

CURSO ALTERNATIVO:

3. Usuário seleciona um filtro (Mais recentes, mais antigos, últimos 7 dias, últimos 15 dias ou últimos 40 dias);

3.1. O sistema busca o relatório ordenado pelo filtro escolhido; Caso de uso vai para o passo 3.

CURSOS DE EXCEÇÃO:

2.1. Não há pedidos realizados. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório.

REQUISITOS DE INTERFACE: A tela deverá exibir a lista de pedidos feitos de acordo com o filtro de exibição com os campos: Data, preço da compra e status. Os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal.

REQUISITOS NÃO-FUNCIONAIS: Opções de filtro devem estar disponíveis no topo da página.

UC08 - GERAR RELATÓRIO DE CLIENTES CADASTRADOS

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja consultar os clientes cadastrados no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema e clientes cadastrados no sistema.

CURSO NORMAL:

1. O administrador solicita tela de clientes;
2. O sistema busca o relatório ordenado pelos cadastros mais recentes;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Nome, contato e cpf/cnpj; Caso de uso termina.

CURSO ALTERNATIVO:

3. Usuário seleciona um filtro (Mais recentes, mais antigos ou por ordem alfabética);

3.1 O sistema busca o relatório ordenado pelo filtro escolhido; Caso de uso vai para o passo 3.

CURSOS DE EXCEÇÃO:

2.1 Não há clientes cadastrados. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório.

REQUISITOS DE INTERFACE: a tela deverá exibir a lista de clientes cadastrados de acordo com o filtro de exibição com os campos: Nome, contato e cpf/cnpj. Os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal.

REQUISITOS NÃO-FUNCIONAIS:

N/A

UC09 - GERAR RELATÓRIO DE PEDIDOS EM ABERTO

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja consultar os pedidos em aberto no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Pedidos previamente cadastrados;

CURSO NORMAL:

1. O administrador solicita tela de pedidos em aberto;
2. O sistema busca o relatório ordenado pelos pedidos feitos mais antigos;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Data, cliente, preço da compra e status; Caso de uso termina.

CURSO ALTERNATIVO:

3.1. Administrador solicita mudança de status do pedido; Caso de uso retorna ao passo 2.

CURSOS DE EXCEÇÃO:

2.1. Não há pedidos em aberto. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório

REQUISITOS DE INTERFACE: a tela deverá exibir a lista de pedidos em aberto por ordem de chegada dos pedidos com os campos: Data, cliente, preço da compra e status. Os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal.

REQUISITOS NÃO-FUNCIONAIS:

UC10 - GERAR RELATÓRIO DE PEDIDOS FINALIZADOS

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja consultar os pedidos finalizados no sistema

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Pedidos previamente cadastrados.

CURSO NORMAL:

1. O administrador solicita tela de pedidos finalizados;
2. O sistema busca o relatório ordenado pelos pedidos finalizados mais recentes;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Data, cliente, preço da compra e status; Caso de uso termina.

CURSO ALTERNATIVO:

3.1. Geração do relatório no formato PDF com as informações Data, cliente, preço da compra e status;

CURSOS DE EXCEÇÃO:

2.1 Não há pedidos em aberto. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório.

REQUISITOS DE INTERFACE: A tela deverá exibir a lista de pedidos finalizados sempre por ordem descendente com os campos: Data, cliente, preço da compra e status. Os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal. A paginação deve estar descrita acima da tabela, por números contendo um link para tal acesso.

REQUISITOS NÃO-FUNCIONAIS: A página deve conter o formato de paginação com vinte e cinco itens por página.

UC11 - GERAR RELATÓRIO DE CUPONS ATIVOS

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja consultar os cupons ativos.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Cupons previamente cadastrados no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cupons;
2. O sistema busca o relatório ordenado por status;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Nome, valor e status; Caso de uso termina.

CURSO ALTERNATIVO:

3.1. Caso o usuário mude o status, caso de uso vai para o passo 3.

CURSOS DE EXCEÇÃO:

2.1. Não há cupons cadastrados. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório.

REQUISITOS DE INTERFACE: a tela deverá exibir a lista de cupons ativos e desativados com os campos: Nome, valor e status. Os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal.

REQUISITOS NÃO-FUNCIONAIS:

UC12 - GERAR RELATÓRIO DE CATEGORIAS E PRODUTOS

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja consultar todas as categorias e produtos cadastrados no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema. Categorias previamente cadastradas no sistema; Produtos previamente cadastrados no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cadastro de produtos;
2. O sistema busca os relatórios ordenados por ordem de inserção no sistema;

3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Categoria, nome, descrição e preço; Caso de uso termina.

CURSO ALTERNATIVO:

3.1. O administrador poderá imprimir o relatório a partir da geração de um PDF.

CURSOS DE EXCEÇÃO:

2.1. Não há categorias e/ou produtos cadastrados. Relatório não é emitido.

PÓS-CONDIÇÕES: Visualização do relatório.

REQUISITOS DE INTERFACE: a tela deverá exibir a lista de produtos e categorias com os campos: Categoria, nome, descrição e preço. Deverá ter duas tabelas distintas para categorias e produtos onde os dados da listagem devem estar na cor cinza escuro, contrapondo a cor da janela principal.

REQUISITOS NÃO-FUNCIONAIS:

UC13 - EXCLUIR CUPOM

SUMÁRIO: Esse caso de uso serve para a entidade cupom. O caso de uso inicia quando o administrador precisa excluir o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Cupom previamente cadastrado no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cupons;
2. O sistema busca o relatório ordenado por status;
3. O sistema gera o relatório e exibe na tela os resultados e exibe as informações: Nome, valor e status;
4. O administrador seleciona o cupom a ser excluído;
5. O sistema realiza a exclusão do cupom;

6. O sistema exibe mensagem de exclusão com sucesso; caso de uso termina.

CURSO ALTERNATIVO:

No passo 3, o administrador abandona a operação, a exclusão é cancelada; caso de uso termina.

CURSOS DE EXCEÇÃO:

4.1. O cupom selecionado está com o status ativado; Sistema informa que só pode excluir cupom desativado; Caso de uso retorna ao passo 3.

PÓS-CONDIÇÕES: Visualização do relatório de cupons cadastrados.

REQUISITOS DE INTERFACE: A função será ativada por meio de um ícone ao final de cada linha correspondente da tabela.

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá a exclusão de apenas um cupom por vez.

UC14 - EXCLUIR CATEGORIA

SUMÁRIO: Esse caso de uso serve para a entidade categoria. O caso de uso inicia quando o administrador precisa excluir o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Categoria previamente cadastrado no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cadastro de produtos;
2. O sistema busca o relatório de produtos cadastrados;
3. O sistema gera o relatório e exibe na tela os resultados;
4. O administrador seleciona a categoria a ser excluída;
5. O sistema verifica se não existe produto cadastrado na categoria selecionada.

6. O sistema realiza a exclusão da categoria;
7. O sistema exibe mensagem de exclusão com sucesso; caso de uso termina.

CURSO ALTERNATIVO:

No passo 3, o administrador abandona a operação, a exclusão é cancelada; caso de uso termina.

CURSOS DE EXCEÇÃO:

3.1. Se não houver resultados de busca; caso de uso termina;

5.1. Se houver produto cadastrado na categoria selecionada, sistema exibe mensagem de erro; Exclusão não é concluída. Caso de uso retorna ao passo 2.

PÓS-CONDIÇÕES: Visualização do relatório de produtos cadastrados

REQUISITOS DE INTERFACE: A exclusão será ativada a partir de um link descritivo ao final de cada linha da tabela

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá a exclusão de apenas uma categoria por vez.

UC15 - ARQUIVAR PRODUTO

SUMÁRIO: Esse caso de uso serve para a entidade produto. O caso de uso inicia quando o administrador precisa arquivar o mesmo no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Produto previamente cadastrado no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cadastro de produtos;
2. O sistema busca o relatório de produtos cadastrados;

3. O sistema gera o relatório e exibe na tela os resultados;
4. O administrador seleciona o produto a ser arquivado;
5. O sistema realiza o arquivamento do produto;
6. O sistema exibe mensagem de arquivamento com sucesso; caso de uso termina.

CURSO ALTERNATIVO:

No passo 3, o administrador abandona a operação, o arquivamento é cancelado; caso de uso termina.

CURSOS DE EXCEÇÃO:

3.1 Se não houver resultados de busca; caso de uso termina;

PÓS-CONDIÇÕES: Visualização do relatório de produtos cadastrados não-arquivados.

REQUISITOS DE INTERFACE: O arquivamento será ativada a partir de um link descritivo sendo o penúltimo elemento de cada linha da tabela

REQUISITOS NÃO-FUNCIONAIS: o sistema permitirá a exclusão de apenas um produto por vez.

UC16 - ALTERAR DADOS DO CLIENTE

SUMÁRIO: Esse caso de uso serve para a entidade cliente. O caso de uso inicia quando o usuário precisa editar o mesmo no sistema.

ATORES: Usuário.

PRÉ-CONDIÇÕES: Usuário cadastrado e autenticado no sistema.

CURSO NORMAL:

1. O usuário solicita a visualização dos dados pessoais
2. O sistema busca os dados;
3. O sistema exibe os dados pessoais e endereços cadastrados;
4. O usuário seleciona a opção de alteração dos dados pessoais;

5. O sistema exibe os dados pessoais;
6. Usuário edita os seguintes campos: Senha, contato e cpf/cnpj;
7. O sistema valida o campo de contato e salva as alterações;
8. O sistema exibe mensagem de com sucesso; caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 2 e 6, o administrador abandona a operação, a validação é cancelada; caso de uso termina.

CURSOS DE EXCEÇÃO:

7.1 Usuário preencheu o campo contato com uma letra correspondente de A-Z; Sistema informa o erro; Caso de uso retorna ao passo 5.

PÓS-CONDIÇÕES: Visualização da *homepage.*

REQUISITOS DE INTERFACE: Campo Nome completo não pode ser editado. Os campos do formulário devem ser mostrados preenchidos com o atual registro de cada campo.

REQUISITOS NÃO-FUNCIONAIS:

UC17 - ALTERAR ENDEREÇO DE ENTREGA

SUMÁRIO: Esse caso de uso serve para a entidade localEntrega. O caso de uso inicia quando o usuário precisa editar seus endereços de entrega no sistema.

ATORES: Usuário.

PRÉ-CONDIÇÕES: Usuário cadastrado e autenticado no sistema; Endereço(s) cadastrado(s) no sistema.

CURSO NORMAL:

1. O usuário solicita visualização dos dados pessoais
2. O sistema busca os dados;
3. O sistema exibe os dados pessoais e endereços cadastrados;

4. O usuário seleciona a alteração do endereço;
5. O sistema exibe os seguintes campos do formulário;
6. Usuário preenche os campos: Endereço, bairro, cidade, número e complemento;
7. Usuário solicita atualização do endereço selecionado;
8. O sistema exibe mensagem de sucesso; caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 3 e 6, o administrador abandona a operação, a validação é cancelada; caso de uso termina.

CURSOS DE EXCEÇÃO:

6.1 Usuário digita algo que não seja um numeral no campo número; Sistema não permite a inserção no campo; Caso de uso continua no mesmo passo.

PÓS-CONDIÇÕES: Visualização da *homepage.*

REQUISITOS DE INTERFACE: Campos do formulário deve conter: Endereço, bairro, cidade, número e complemento. Os campos do formulário devem ser mostrados preenchidos com o atual registro de cada campo.

REQUISITOS NÃO-FUNCIONAIS: O campo deve conter identificação sobre para qual campo está ativo no momento.

UC18 - ALTERAR PRODUTO

SUMÁRIO: Este caso de uso se inicia quando o administrador deseja alterar um produto cadastrado no sistema.

ATORES: Administrador.

PRÉ-CONDIÇÕES: Administrador autenticado no sistema; Produto(s) cadastrado(s) no sistema.

CURSO NORMAL:

1. O administrador solicita tela de cadastro de produtos;

2. O sistema busca a solicitação;
3. O sistema gera o relatório de categorias cadastradas e produtos cadastrados e exibe na tela os resultados;
4. Administrador seleciona o produto a ser editado
5. O sistema exibe os seguintes campos: Produto, categoria, descrição e valor unitário;
6. Usuário preenche os campos: Produto, categoria, descrição e valor unitário;
7. Usuário solicita atualização do produto;
8. O sistema exibe mensagem de com sucesso; caso de uso termina.

CURSO ALTERNATIVO:

Entre os passos 3 e 6, o administrador abandona a operação, a validação é cancelada; caso de uso termina.

CURSOS DE EXCEÇÃO:

6.1 Usuário digita letra no campo valor unitário; Sistema não permite a inserção no campo; Caso de uso continua no mesmo passo.

PÓS-CONDIÇÕES: Visualização da página inicial

REQUISITOS DE INTERFACE: a tela deverá exibir a lista de produtos e categorias. Os campos do formulário devem ser mostrados preenchidos com o atual registro de cada campo. Campo categoria deve estar no formato cascata.

REQUISITOS NÃO-FUNCIONAIS: O campo deve conter identificação sobre para qual campo está ativo no momento.