



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO**

**FEIRA DO AGRICULTOR FAMILIAR: UMA INICIATIVA DIGITAL PARA
FOMENTAR O COMÉRCIO DA AGRICULTURA FAMILIAR NO PARÁ**

VINÍCIUS LOUZADA VALENTE

**TUCURUÍ - PA
2024**



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO**

VINÍCIUS LOUZADA VALENTE

**FEIRA DO AGRICULTOR FAMILIAR: UMA INICIATIVA DIGITAL PARA
FOMENTAR O COMÉRCIO DA AGRICULTURA FAMILIAR NO PARÁ**

Trabalho de Conclusão de Curso
apresentado para obtenção do grau de
Engenheiro de Computação da Faculdade de
Engenharia de Computação do Campus
Universitário de Tucuruí.

**TUCURUÍ - PA
2024**

**FEIRA DO AGRICULTOR FAMILIAR: UMA INICIATIVA DIGITAL PARA
FOMENTAR O COMÉRCIO DA AGRICULTURA FAMILIAR NO PARÁ**

Este trabalho foi julgado em 19/06/2024 para obtenção do grau de Engenheiro de Computação, aprovado em sua forma final pela banca examinadora que atribuiu o conceito EXCELENTE.

Profa. Dra. Viviane Almeida dos Santos
Orientadora

Prof. Dr. Carlos dos Santos Portela
Membro Externo da Banca Examinadora

Prof. Dr. Iago Lins de Medeiros
Membro Interno da Banca Examinadora

Prof. Dr. Daniel da Conceição Pinheiro
Diretor da Faculdade de Engenharia de Computação

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todas as pessoas que contribuíram para minha jornada acadêmica e para o sucesso deste trabalho:

Agradeço primeiramente a Deus, minha fonte constante de força e inspiração.

À minha família, pelo incentivo e carinho. Em especial, à minha mãe Livan, meu pai Leobaldo e meu irmão Emanuel, cujo apoio incondicional foi fundamental para que eu chegasse onde estou hoje.

À minha companheira Lorrane, por ser uma grande incentivadora em cada passo deste percurso.

Ao Udson, Andreia, Rui e Pinheiro, que desempenharam papéis essenciais no projeto da Feira da Agricultura Familiar.

À minha professora e orientadora, Dr^a. Viviane Almeida dos Santos, pelos valiosos ensinamentos e pela excelente orientação neste trabalho.

Aos demais professores do curso de Engenharia de Computação, por suas contribuições à minha formação.

Aos meus colegas da turma de Engenharia de Computação 2018, que compartilharam comigo momentos de aprendizado e crescimento. Um agradecimento especial aos meus amigos Brendo, Danilo e Rui, cuja amizade e apoio foram fundamentais nesta jornada.

E a todos que, de alguma forma, contribuíram para a minha formação. Muito obrigado!

RESUMO

A crescente relevância dos negócios digitais se tornou evidente, especialmente durante a pandemia de COVID-19, destacando-se a necessidade de soluções inovadoras para impulsionar os setores econômicos. Neste contexto, este trabalho se propõe a desenvolver, descrever o processo de desenvolvimento e avaliar o *e-commerce*: Feira do Agricultor Familiar. Esta solução visa ampliar as vendas dos agricultores familiares para regiões do Pará, onde foi identificada a carência de plataformas ativas. A solução adotada foi desenvolver uma aplicação que permite que os agricultores gerenciem seus próprios anúncios na plataforma, colocando-os como protagonistas e promovendo o empoderamento digital neste público que enfrenta desafios ao utilizar soluções digitais. O processo de desenvolvimento abrange as etapas de levantamento de requisitos, casos de uso, modelagem, implementação e refatoração, resultando em uma documentação detalhada. Além disso, foi aplicado um teste de usabilidade, que se baseia no System Usability Scale (SUS). Os resultados da avaliação foram positivos pelo fato de estarem acima da média, atingindo uma média de Índice de Satisfação do Cliente de 75. Esse resultado é crucial para orientar o crescimento e o desenvolvimento de novas funcionalidades na plataforma.

Palavras-chave: Comércio Eletrônico, Agricultura Familiar, Sistema Web, Engenharia de Software, Teste de Usabilidade.

ABSTRACT

The growing relevance of digital business has become evident, especially during the COVID-19 pandemic, highlighting the need for innovative solutions to boost economic sectors. In this context, this work claims to develop, describe the development process and evaluate the e-commerce: Family Farmer Fair. This solution aims to expand the sales of family farmers to regions of Pará where a lack of active platforms has been identified. The solution adopted was to develop an application that allows farmers to manage their own ads on the platform, placing them as protagonists and promoting digital empowerment in this public that faces challenges when using digital solutions. The development process covers the stages of system requirements gathering, use cases, modeling, implementation and refactoring, resulting in detailed documentation. In addition, a usability test was applied, based on the System Usability Scale (SUS). The results of the evaluation were positive in that they were above average, reaching an average Customer Satisfaction Index of 75. This result is crucial for guiding the growth and development of new features in the platform.

Keywords: E-commerce, Family Farming, Web System, Software Engineering, Usability Testing.

LISTA DE FIGURAS

Figura 1 - Faturamento do E-commerce de 2001 até 2020.....	15
Figura 2 - Comunicação entre cliente e servidor.....	21
Figura 3 - Funcionamento da arquitetura MVC.....	27
Figura 4 - Metodologia e linha do tempo seguida neste trabalho.....	42
Figura 5 - Abordagem Iterativa e Incremental adotada nesta fase do processo.....	47
Figura 6 - Cores Pantone utilizadas na identidade visual da Feira da Agricultura Familiar.....	55
Figura 7 - Simbologia de duas folhas com um coração que formam o logotipo da Feira da Agricultura Familiar.....	55
Figura 8 - Logotipo formato 1, formato circular.....	56
Figura 9 - Logotipo formato 2, formato horizontal.....	56
Figura 10 - Logotipo formato 3, formato de ícone.....	56
Figura 11 - Paleta de cores do e-commerce Feira da Agricultura Familiar com seus respectivos códigos hexadecimais.....	58
Figura 12 - Barra de menu do agricultor logado na plataforma.....	58
Figura 13 - Barra de menu do blog da plataforma.....	59
Figura 14 - Barra de menu da feira online da plataforma.....	59
Figura 15 - Arquivos da interface da plataforma.....	60
Figura 16 - Código-fonte do arquivo fotos.php.....	61
Figura 17 - Página inicial do blog com apresentação dos benefícios do projeto.....	62
Figura 18 - Página inicial do blog com detalhes do projeto.....	62
Figura 19 - Página inicial do blog com detalhes da localização da feira presencialmente.....	63
Figura 20 - Página inicial do blog com detalhes do rodapé.....	63
Figura 21 - Página de dicas do blog com lista de artigos.....	64
Figura 22 - Página de login.....	65
Figura 23 - Página de cadastro.....	65
Figura 24 - Página inicial da feira online.....	66
Figura 25 - Página de perfil de agricultor com seus produtos a venda.....	66
Figura 26 - Página de vendas do agricultor, pedidos em andamento e pedidos finalizados.....	67
Figura 27 - Página de anúncios que estão ativos pelo agricultor.....	67
Figura 28 - Página de criação de anúncios.....	68
Figura 29 - Página de perfil de usuário.....	68
Figura 30 - Página inicial de forma responsiva para dispositivos móveis.....	69
Figura 31 - Página da feira online e vendas do agricultor de forma responsiva para dispositivos móveis.....	70
Figura 32 - Página de anúncios do agricultor e criação de anúncios de forma responsiva para dispositivos móveis.....	71
Figura 33 - Diagrama da Modelagem Conceitual do sistema Feira da Agricultura	

Familiar.....	72
Figura 34 - Modelagem física do banco de dados do sistema Feira da Agricultura Familiar.....	74
Figura 35 - Trecho de código SQL responsável por gerar o banco de dados do sistema da agricultura familiar.....	75
Figura 36 - Server-Side Rendering na prática. Desde a requisição do cliente até a resposta preparada do servidor.....	76
Figura 37 - Estrutura básica de um monolítico.....	77
Figura 38 - Relação da estrutura de pastas com as camadas de uma aplicação monolítica.....	78
Figura 39 - Raiz do projeto. Estrutura de pastas da aplicação.....	78
Figura 40 - Detalhes de pastas e arquivos no pacote config.....	79
Figura 41 - Detalhes das pastas internas do pacote config.....	80
Figura 42 - Detalhes das pastas internas do pacote src.....	80
Figura 43 - Detalhes das pastas internas do pacote public.....	81
Figura 44 - Função password_hash sendo usada para criptografar a “senha123”....	82
Figura 45 - String de hash da senha “senha123”.....	83
Figura 46 - Função password_verify certificando a correspondência de uma string de senha com uma string de hash.....	83
Figura 47 - Cookie de identificador único (PHPSESSID). Este identificador é relacionado com uma sessão que está armazenada do lado do servidor web.....	85
Figura 48 - Trecho de código do arquivo ‘config-login.php’, responsável por autenticar o usuário.....	86
Figura 49 - Trecho de código do arquivo ‘verificaLoginVendedor.php’, responsável por verificar se o tipo de usuário é agricultor.....	87
Figura 50 - Processo de roteamento a partir de uma requisição para o recurso “/fotos” até a resposta pelo servidor.....	88
Figura 51 - Arquivo router.php responsável por mapear as rotas da aplicação.....	89
Figura 52 - Arquivo index.php atuando como Front Controller.....	89
Figura 53 - Pasta src com dados da biblioteca PHPMailer.....	90
Figura 54 - Arquivo BuscarProduto.php responsável por armazenar algumas consultas e persistência no banco de dados por meio da biblioteca MySQLi.....	90
Figura 55 - Logotipo do gerenciador de dependências do PHP — Composer.....	91
Figura 56 - Arquivo composer.json final do projeto, após refatoração.....	92
Figura 57 - Exemplo de instância do objeto PDO com as configurações do banco de dados.....	93
Figura 58 - Código do PaginaCriarAnunciosController.....	94
Figura 59 - O arquivo "rotas.php" contém as definições dos controladores e os níveis de acesso associados a cada recurso que o usuário deseja acessar.....	95
Figura 60 - Classe “AutorizacaoMiddleware” responsável por avaliar autorização do usuário para determinado recurso do sistema.....	96
Figura 61 - Processo de roteamento a partir de uma requisição para o recurso “/fotos” até a resposta pelo servidor.....	97

Figura 62 - Processo de roteamento atual da aplicação.....	98
Figura 63 - Definição do namespace da aplicação.....	100
Figura 64 - Exemplo de controller que implementa a interface RequestHandlerInterface.....	102
Figura 65 - Exemplo de Middleware que implementa a interface MiddlewareInterface.....	102
Figura 66 - Arquitetura tradicional em camadas.....	104
Figura 67 - Arquitetura monolítica em camadas.....	104
Figura 68 - Diagrama de Casos de Uso da Feira da Agricultura Familiar.....	105
Figura 69 - Diagrama de Classes de Modelo (Domínio).....	133
Figura 70 - Diagrama de Classes dos Controllers do pacote Admin.....	134
Figura 71 - Diagrama de Classes dos Controllers do pacote Blog.....	134
Figura 72 - Diagrama de Classes dos Controllers do pacote Cadastro.....	135
Figura 73 - Diagrama de Classes dos Controllers do pacote Carrinho.....	135
Figura 74 - Diagrama de Classes dos Controllers do pacote Feira.....	135
Figura 75 - Diagrama de Classes dos Controllers do pacote Frete.....	136
Figura 76 - Diagrama de Classes dos Controllers do pacote Login.....	136
Figura 77 - Diagrama de Classes dos Controllers do pacote Pagamento.....	136
Figura 78 - Diagrama de Classes dos Controllers do pacote Meu Perfil.....	137
Figura 79 - Diagrama de Classes dos Controllers do pacote Anuncio.....	137
Figura 80 - Diagrama de Classes dos Controllers do pacote Pedidos.....	137
Figura 81 - Diagrama de Classes dos Controllers do pacote Vendas.....	138
Figura 82 - Diagrama de Classes do pacote Repositories.....	138
Figura 83 - Diagrama de Objetos do Domínio da aplicação.....	139
Figura 84 - Diagrama de Sequência do Caso de Uso: Registrar Conta.....	140
Figura 85 - Diagrama de Sequência do Caso de Uso: Confirmar E-mail.....	141
Figura 86 - Diagrama de Sequência do Caso de Uso: Realizar Login.....	142
Figura 87 - Diagrama de Sequência do Caso de Uso: Pesquisar Produto.....	143
Figura 88 - Diagrama de Sequência do Caso de Uso: Realizar Pedido.....	144
Figura 89 - Diagrama de Sequência do Caso de Uso: Consultar Pedido.....	145
Figura 90 - Diagrama de Sequência do Caso de Uso: Cancelar Pedido.....	146
Figura 91 - Diagrama de Sequência do Caso de Uso: Editar Dados.....	147
Figura 92 - Diagrama de Sequência do Caso de Uso: Anunciar Produtos.....	148
Figura 93 - Diagrama de Sequência do Caso de Uso: Editar Anúncio.....	149
Figura 94 - Diagrama de Sequência do Caso de Uso: Excluir Anúncio.....	150
Figura 95 - Diagrama de Sequência do Caso de Uso: Adicionar Agricultor.....	151
Figura 96 - Diagrama de Sequência do Caso de Uso: Excluir Agricultor.....	152
Figura 97 - Diagrama de Sequência do Caso de Uso: Gerar Relatório.....	152
Figura 98 - Questionário do SUS preenchido pelo Potencial Cliente.....	159
Figura 99 - Questionário do SUS preenchido pelo Agricultor Familiar.....	160

LISTA DE TABELAS

Tabela 1 - Principais métodos HTTP.....	22
Tabela 2 - Lista de trabalhos incluídos.....	30
Tabela 3 - Lista de trabalhos excluídos.....	32
Tabela 4 - Distribuição macrorregional dos mercados digitais da agricultura familiar no território brasileiro.....	33
Tabela 5 - Plataformas pesquisadas no território brasileiro em 2021 para a Agricultura Familiar.....	34
Tabela 6 - Análise das plataformas digitais da agricultura familiar mencionadas por Gazolla e De Aquino (2021).....	36
Tabela 7 - Questionário SUS aplicado no teste de usabilidade.....	52
Tabela 8 - Especificação do caso de uso: Registrar conta.....	106
Tabela 9 - Especificação do caso de uso: Confirmar E-mail.....	109
Tabela 10 - Especificação do caso de uso: Realizar Login.....	111
Tabela 11 - Especificação do caso de uso: Pesquisar Produtos.....	113
Tabela 12 - Especificação do caso de uso: Realizar Pedidos.....	114
Tabela 13 - Especificação do caso de uso: Consultar Pedidos.....	119
Tabela 14 - Especificação do caso de uso: Cancelar Pedidos.....	121
Tabela 15 - Especificação do caso de uso: Editar dados.....	122
Tabela 16 - Especificação do caso de uso: Anunciar Produtos.....	125
Tabela 17 - Especificação do caso de uso: Editar Anúncios.....	126
Tabela 18 - Especificação do caso de uso: Excluir Anúncios.....	127
Tabela 19 - Especificação do caso de uso: Adicionar Agricultor.....	128
Tabela 20 - Especificação do caso de uso: Excluir Agricultor.....	130
Tabela 21 - Especificação do caso de uso: Gerar Relatório.....	131
Tabela 22 - Tarefas a serem realizadas pelo entrevistado.....	154

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 CONTEXTO.....	15
1.2 PROBLEMA.....	16
1.3 OBJETIVOS.....	17
1.4 ESTRUTURA DO TRABALHO.....	18
2 REFERENCIAL TEÓRICO.....	20
2.1 E-COMMERCE.....	20
2.2 HTTP: COMO FUNCIONA A WEB.....	21
2.2.1 HTTP: Métodos.....	22
2.2.2 Mensagem de requisição HTTP.....	22
2.2.3 Mensagem de resposta HTTP.....	23
2.3 ENGENHARIA DE SOFTWARE.....	24
2.3.1 Processo de software.....	24
2.3.2 Modelo de Software.....	25
2.4 ARQUITETURA DE SOFTWARE.....	26
2.4.1 Arquitetura MVC (Model View Controller).....	26
2.5 BANCO DE DADOS.....	27
2.5.1 MySQL.....	28
3 REVISÃO DA LITERATURA.....	29
3.1 PALAVRAS-CHAVE.....	29
3.2 STRINGS DE BUSCA.....	29
3.2.1 Pesquisas realizadas no Periódicos Capes.....	29
3.2.2 Pesquisas realizadas no Google Scholar.....	30
3.3 LISTA DE TRABALHOS ENCONTRADOS.....	30
3.3.1 Lista de trabalhos incluídos.....	30
3.3.2 Lista de trabalhos excluídos.....	31
3.4 ANÁLISE DOS TRABALHOS INCLUÍDOS.....	33
3.5 LACUNAS EXISTENTES.....	40
4 METODOLOGIA.....	42
4.1 PROJETO DE EXTENSÃO E PESQUISA - DEMANDA.....	43
4.1.1 Requisitos do sistema.....	43
4.2 ESTUDOS.....	45
4.3 DESENVOLVIMENTO.....	46
4.4 ESTUDO SOBRE BOAS PRÁTICAS.....	49
4.5 REFATORAÇÃO.....	49
4.6 DOCUMENTAÇÃO.....	50
4.7 TESTES DE USABILIDADE.....	51
4.8 AVALIAÇÃO DOS RESULTADOS.....	52

5 DESENVOLVIMENTO	54
5.1 IDENTIDADE VISUAL	54
5.2 FRONT-END	57
5.2.1 Desenvolvimento do blog	61
5.2.2 Desenvolvimento da feira online	64
5.2.3 Desenvolvimento responsivo	68
5.3 BANCO DE DADOS	71
5.3.1 Modelagem Conceitual	71
5.3.2 Modelagem Lógica e Modelagem Física	73
5.4 BACK-END	75
5.4.1 Armazenamento de senha	82
5.4.2 Autenticação	83
5.4.3 Autorização	84
5.4.4 Roteamento	87
5.4.5 Dependências	89
5.4.6 Persistência	90
5.5 REFATORAÇÃO	91
5.5.1 Gerenciador de dependências	91
5.5.2 Biblioteca de conexão e persistência	92
5.5.3 Separar as responsabilidades do arquivo de visualização	93
5.5.4 Adicionar um Middleware para efetuar o serviço de autorização	94
5.5.5 Atualização do Front Controller	96
5.5.6 Padronizar a aplicação conforme as PSRs	98
5.5.6.1 PSR-12	99
5.5.6.2 PSR-4	99
5.5.6.3 PSR-7	100
5.5.6.4 PSR-17	101
5.5.6.5 PSR-15	101
5.5.6.6 PSR-11	103
5.5.7 Segregar toda a aplicação em camadas	103
5.6 DOCUMENTAÇÃO	104
5.6.1 Diagramas de Casos de Uso	105
5.6.2 Especificação dos casos de uso	106
5.6.2.1 Registrar Conta	106
5.6.2.2 Confirmar E-mail	109
5.6.2.3 Realizar Login	111
5.6.2.4 Pesquisar Produtos	113
5.6.2.5 Realizar Pedidos	114
5.6.2.6 Consultar Pedidos	119
5.6.2.7 Cancelar Pedidos	121

5.6.2.8	Editar Dados.....	122
5.6.2.9	Anunciar Produtos.....	125
5.6.2.10	Editar Anúncios.....	126
5.6.2.11	Excluir Anúncios.....	127
5.6.2.12	Adicionar Agricultor.....	128
5.6.2.13	Excluir Agricultor.....	130
5.6.2.14	Gerar Relatório.....	131
5.6.3	Diagrama de Classes.....	133
5.6.3.1	Diagrama de Classes de Modelo.....	133
5.6.3.2	Diagrama de Classes de Controllers do pacote Admin.....	134
5.6.3.3	Diagrama de Classes de Controllers do pacote Blog.....	134
5.6.3.4	Diagrama de Classes de Controllers do pacote Cadastro.....	135
5.6.3.5	Diagrama de Classes de Controllers do pacote Carrinho.....	135
5.6.3.6	Diagrama de Classes de Controllers do pacote Feira.....	135
5.6.3.7	Diagrama de Classes de Controllers do pacote Frete.....	136
5.6.3.8	Diagrama de Classes de Controllers do pacote Login.....	136
5.6.3.9	Diagrama de Classes de Controllers do pacote Pagamento.....	136
5.6.3.10	Diagrama de Classes de Controllers do pacote Meu Perfil.....	137
5.6.3.11	Diagrama de Classes de Controllers do pacote Anuncio.....	137
5.6.3.12	Diagrama de Classes de Controllers do pacote Pedidos.....	137
5.6.3.13	Diagrama de Classes de Controllers do pacote Vendas.....	138
5.6.3.14	Diagrama de Classes do pacote Repositories.....	138
5.6.4	Diagrama de Objetos.....	138
5.6.4.1	Diagrama de Objetos de Modelo.....	139
5.6.5	Diagrama de Sequência.....	139
5.6.5.1	Registrar Conta.....	140
5.6.5.2	Confirmar E-mail.....	141
5.6.5.3	Realizar Login.....	142
5.6.5.4	Pesquisar Produto.....	143
5.6.5.5	Realizar Pedido.....	144
5.6.5.6	Consultar Pedido.....	145
5.6.5.7	Cancelar Pedido.....	146
5.6.5.8	Editar Dados.....	147
5.6.5.9	Anunciar Produtos.....	148
5.6.5.10	Editar Anúncio.....	149
5.6.5.11	Excluir Anúncio.....	150
5.6.5.12	Adicionar Agricultor.....	151
5.6.5.13	Excluir Agricultor.....	152
5.6.5.14	Gerar Relatório.....	152
5.7	VERSÃO FINAL DA APLICAÇÃO.....	153

6 AVALIAÇÃO DA PLATAFORMA.....	154
6.1 TESTE DE USABILIDADE.....	154
6.2 RESULTADOS DO TESTE DE USABILIDADE.....	156
6.2.1 Potencial Cliente.....	157
6.2.2 Agricultor Familiar de Breu Branco.....	157
6.3 AVALIAÇÃO DA PLATAFORMA.....	158
6.3.1 Potencial Cliente.....	159
6.3.2 Agricultor Familiar de Breu Branco.....	160
6.3.3 Considerações do teste de usabilidade.....	160
7 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS.....	161
7.1 CONCLUSÃO.....	161
7.2 SUGESTÕES PARA TRABALHOS FUTUROS.....	161
7.2.1 Para o desenvolvimento.....	162
7.2.2 Para a Comunidade Acadêmica.....	162
8 REFERÊNCIAS BIBLIOGRÁFICAS.....	164

1 INTRODUÇÃO

Os constantes avanços tecnológicos têm influenciado a maneira como a sociedade vive, tendo como exemplo a internet, que transformou a forma de comunicação, por meio de imagens, vídeos e áudios, diferentemente das gerações passadas que utilizavam códigos morse e o telégrafo como principal meio de comunicação (SUBTIL, 2014). Dado este contexto, torna-se crucial ressaltar a evolução do comércio, desde os seus primórdios até os dias atuais.

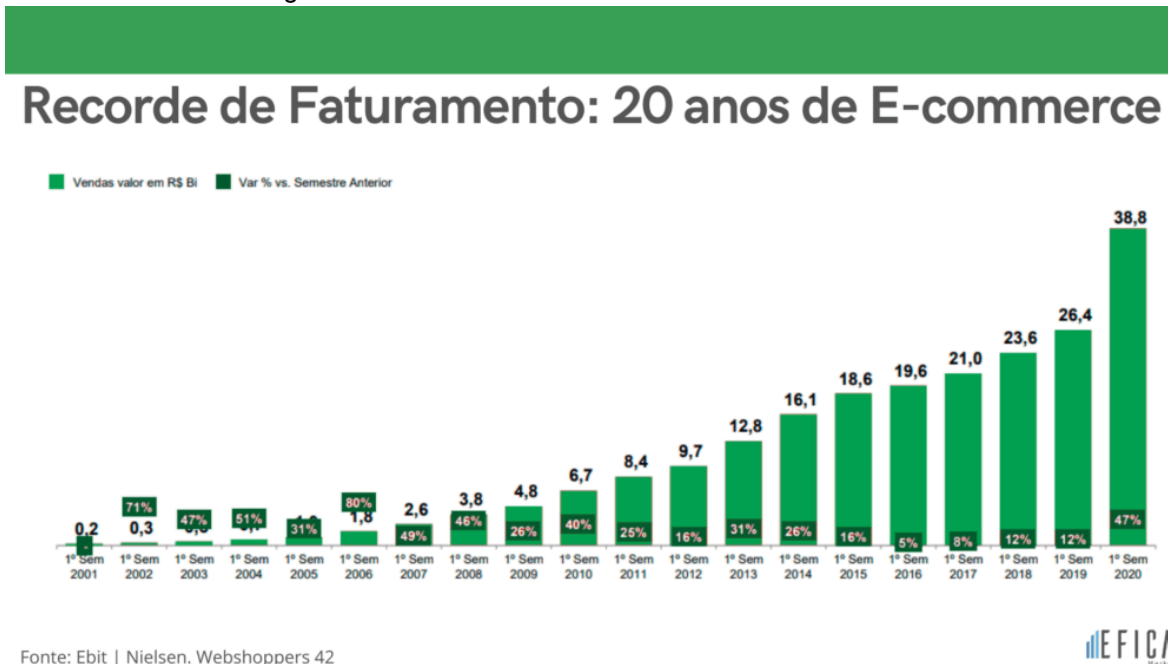
A forma primitiva do comércio ocorria por meio do escambo, uma simples troca de mercadorias sem a medição de valor ou o uso de dinheiro, por exemplo: trocava-se lã por couro, milho por batata e açúcar por tabaco. No entanto, o objeto de troca muitas vezes era perecível ou de difícil transporte, resultando em oscilações de valores e na impossibilidade de acumular riquezas. Para solucionar esse problema, surgiram metais preciosos que passaram a ser utilizados como meio de troca. Em seguida, as moedas foram gradualmente introduzidas, proporcionando uma forma mais eficiente e duradoura de facilitar as transações comerciais (DOS SANTOS, 2014).

É evidente que com o passar dos anos, o comércio passou por uma notável evolução para suprir as necessidades de cada geração. O surgimento da internet em 1960 exerceu um papel transformador nesse processo, pois abriu o caminho para a integração da tecnologia em diversas vertentes, como o comércio. Em 1979, Michael Aldrich utilizou uma televisão modificada conectada a uma linha telefônica e um computador para realizar transações em tempo real. Essa iniciativa marcou o surgimento do comércio eletrônico, popularmente conhecido como *e-commerce* (DE MENDONÇA, 2016).

O *e-commerce* promove a compra e venda de produtos por meio da internet, o que traz grandes benefícios tanto para o vendedor quanto para o consumidor. O consumidor consegue comprar sem sair de casa, e o vendedor consegue vender muito mais por alcançar uma grande quantidade de pessoas.

No Brasil, o *e-commerce* teve início na década de 1990, com a chegada da internet comercial. De acordo com levantamento da Ebit | Nielsen Webshoppers 42 (2020), os números de vendas através dessa modalidade apenas cresceram. Tendo um destaque para o crescimento de 47% de 2019 para 2020.

Figura 1 - Faturamento do E-commerce de 2001 até 2020.



Fonte: Ebit | Nielsen, Webshoppers. Adaptada por Eficaz Marketing (2020)¹

1.1 CONTEXTO

O aumento expressivo de 47% no faturamento do comércio eletrônico de 2019 para 2020, em comparação com anos anteriores, pode ser atribuído, em grande parte, ao surgimento da pandemia do COVID-19, uma doença altamente transmissível. Em razão disso, cientistas e autoridades do mundo recomendaram o isolamento e distanciamento social como medidas para conter a propagação do vírus, logo, as ruas, escolas, lojas ficaram vazias, promovendo uma grande crise global (DA SILVA, 2021).

Em função do isolamento social, vale uma especial atenção aos agricultores familiares, que costumavam vender seus alimentos em feiras livres e enfrentaram dificuldades para trabalhar, resultando no acúmulo ou deterioração dos produtos agropecuários. Isso impactou diretamente nas receitas dessas famílias, uma vez que a venda de sua produção constitui a principal fonte de renda (BREITENBACH, 2021).

¹ Disponível em:

<https://eficazmarketing.com/blog/aproveite-o-crescimento-do-e-commerce-no-brasil-para-vender-mais/>
. Acesso em: 30 mar. 2023

Essa pandemia acelerou de forma discrepante da vinda do “futuro”, as atividades do cotidiano passaram ser de forma remota: trabalho, estudos e compra de produtos. Ou seja, a internet funcionou como uma grande aliada frente a esta problemática (DA SILVA, 2021).

Conseqüentemente, o *e-commerce* deixou de ser uma tendência para se tornar um hábito, uma vez que as empresas investiram intensamente neste mercado com o intuito de fazer com que seus produtos chegassem ao consumidor final (DE MELO CRUZ, 2021). À vista disso, o faturamento do *e-commerce* brasileiro nos 5 primeiros meses de 2022 atingiu um aumento de 785% em comparação com o mesmo período em 2019, segundo uma pesquisa realizada pela SmartHint². Além disso, segundo dados da ABComm³, em 2019 eram abertas 10 mil lojas online por mês, após o anúncio do isolamento social, este número cresceu em 400%, atingindo a média de 50 mil lojas por mês.

1.2 PROBLEMA

O novo hábito de mercado, o *e-commerce*, trouxe desafios de maneira geral, dentre esses desafios, destacam-se o crescimento exponencial da demanda, gerando sobrecarga nos sistemas logísticos; a necessidade de adaptação ágil às tecnologias para garantir a presença efetiva de produtos online; e a mudança nas preferências dos consumidores em relação aos produtos.

Para os agricultores familiares, a urgência de suspender as feiras presenciais representou um desafio considerável, uma vez que muitos deles não conseguiram se ajustar ao novo panorama do mercado (RODRIGUES, 2022). Isto é preocupante, pois conforme indica o Censo Agropecuário de 2017⁴, a agricultura familiar possui uma importante participação na produção dos alimentos que vão para a mesa dos brasileiros.

² Disponível em: <https://www.smarthint.co/ecommerce-no-brasil/>. Acesso em: abril. 2024.

³ Disponível em: <https://abcomm.org/noticias/faturamento-do-e-commerce-cresce-568-neste-ano-e-chega-a-r-4192-bilhoes/>. Acesso em: abril. 2024.

⁴ Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/25789-censo-agro-2017-populacao-ocupada-nos-estabelecimentos-agropecuarios-cai>. Acesso em: abril. 2024

A agricultura familiar corresponde por 48% do valor da produção de café e banana; 80% do valor de produção da mandioca; 69% do abacaxi e 42% da produção do feijão, conforme aponta o Censo Agropecuário de 2017.

O isolamento social e o escasso comércio eletrônico para agricultura familiar, gerou impactos econômicos negativos, pois os produtos agropecuários não eram comercializados, situação esta que não gerava renda para estas famílias (SCHNEIDER, 2020).

As dificuldades pela difusão das novas tecnologias pelos produtores rurais são dadas pelos motivos descritos a seguir. Primeiro, são encontradas dificuldades em adotar novas tecnologias pelos produtores rurais, por exemplo: sistema de automação agrícola, sensoriamento remoto, irrigação inteligente, maquinário avançado, o uso de computadores, e aplicativos para smartphones. Um dos motivos para esta dificuldade é o atraso tecnológico e analfabetismo (BUAINAIN, 2021).

Segundo, no Brasil, sabe-se pouco a respeito da visão e conhecimentos do mercado digital na agricultura familiar e suas dinâmicas de venda, pois há poucos estudos sobre essa temática (GAZOLLA; DE AQUINO, 2021).

Para combater de frente essa problemática, se faz necessário a criação de aplicações específicas para atender este público, ou seja um *e-commerce* que promova e divulgue os produtos desses agricultores.

1.3 OBJETIVOS

O presente trabalho possui como objetivo geral o desenvolvimento de uma Aplicação Web que divulgue e comercialize os produtos da Agricultura Familiar, em especial para as cidades de Cametá, Tucuruí, Breu Branco e Belém que pertencem ao estado do Pará, de forma que os agricultores possam impulsionar suas vendas e estejam inseridos cada vez mais no ambiente tecnológico.

Os objetivos específicos do trabalho são:

- Realizar a coleta das regras de negócio, análise de requisitos e projeto do software, de acordo com as necessidades dos sujeitos envolvidos na pesquisa;
- Desenvolver um Sistema Web responsivo que comercialize os produtos da Agricultura Familiar de Cametá, Tucuruí, Breu Branco e

Belém, através das seguintes tecnologias: HTML, CSS, Javascript (Front-end), PHP (Back-end) e MySQL (Banco de Dados);

- Elaborar uma documentação abrangente para o software desenvolvido, incluindo: Casos de Uso, Especificação de Casos de Uso, Diagrama de Classes, Diagrama de Objetos e Diagrama de Sequência.
- Avaliar a aplicação por meio de um Teste de Usabilidade.

1.4 ESTRUTURA DO TRABALHO

O Capítulo 1 faz uma introdução sobre o tema ressaltando a importância da criação de um e-commerce para a Agricultura Familiar, contextualizando com os impactos gerados pela COVID-19 e as dificuldades deste público em se difundir a essas novas tecnologias, além disso apresenta os objetivos deste trabalho.

No Capítulo 2, é apresentado o referencial teórico, onde conceitos fundamentais são definidos, incluindo e-commerce, protocolo HTTP, engenharia de software, arquitetura de software e banco de dados.

O Capítulo 3 trata da revisão da literatura, descrevendo em detalhes as estratégias de busca em bases de dados acadêmicas, bem como os critérios de inclusão e exclusão dos trabalhos. O objetivo desta revisão é identificar estudos que possam contribuir com este trabalho e detectar possíveis lacunas no cenário atual, a fim de propor formas de solucioná-las.

No Capítulo 4, é descrita a metodologia usada para construir este trabalho, desde a demanda, estudos, desenvolvimento, refatoração, documentação, testes de usabilidade e o método de avaliação utilizado nos testes.

O Capítulo 5 apresenta o desenvolvimento do trabalho, dividido em seções como identidade visual, front-end, banco de dados e back-end. Cada seção detalha os processos de desenvolvimento, incluindo a modelagem, implementação e refatoração de diferentes aspectos do sistema.

No Capítulo 6, são discutidos os tópicos relacionados à avaliação da plataforma desenvolvida, com foco em testes de usabilidade e resultados obtidos. As percepções de diferentes grupos de usuários, como potenciais clientes e agricultores familiares, são analisadas para avaliar a eficácia da plataforma.

O Capítulo 7 consiste na discussão dos resultados obtidos, onde são apresentadas as conclusões do estudo, a importância das melhorias para os usuários e possíveis direções para futuras pesquisas e desenvolvimentos na área.

Por fim, o capítulo 8 lista as referências bibliográficas utilizadas ao longo do trabalho, fornecendo uma base para consulta e aprofundamento nos temas abordados.

2 REFERENCIAL TEÓRICO

Esta seção possui o objetivo de proporcionar um melhor entendimento dos conceitos que foram utilizados para desenvolver o projeto. Dentre eles estão: *e-commerce*, HTTP, engenharia de software, arquitetura de software e banco de dados.

2.1 E-COMMERCE

O *e-commerce* ou comércio eletrônico pode ser definido como uma rede virtual onde pessoas realizam transações de pedidos de produtos e serviços, para isso, é necessário ter um aparelho eletrônico com acesso à internet que a pessoa consumidora consegue efetuar a compra ou a venda de algum produto com o melhor preço e forma de pagamento. (MULLER, 2013)

No Brasil, o *e-commerce* se iniciou por volta de 1990, e de primeiro momento, o propósito era realizar transações comerciais, neste início, essas transações ainda possuíam um número pequeno, entretanto, com o advento da tecnologia e do mercado mundial, este conceito foi se firmando ainda mais no cotidiano das pessoas como uma maneira ampla para comercializar serviços e produtos. Para as empresas, é mais uma maneira de satisfazer e deixar seus clientes mais à vontade para a procura e escolha de seus produtos (MELO, 2021).

No comércio eletrônico existem alguns tipos de negócios, e cada um deles atende a uma necessidade. Segundo Potter e Turban (2005), os mais comuns tipos de *e-commerce* são:

- B2B - *BUSINESS-TO-BUSINESS*: é a negociação eletrônica entre empresas.
- B2C - *BUSINESS-TO-CONSUMERS*: é a negociação eletrônica entre empresas e consumidores.
- C2B - *CONSUMERS-TO-BUSINESS*: é a negociação eletrônica entre consumidores e empresas.
- C2C - *CONSUMERS-TO-CONSUMER*: é a negociação eletrônica entre consumidores.

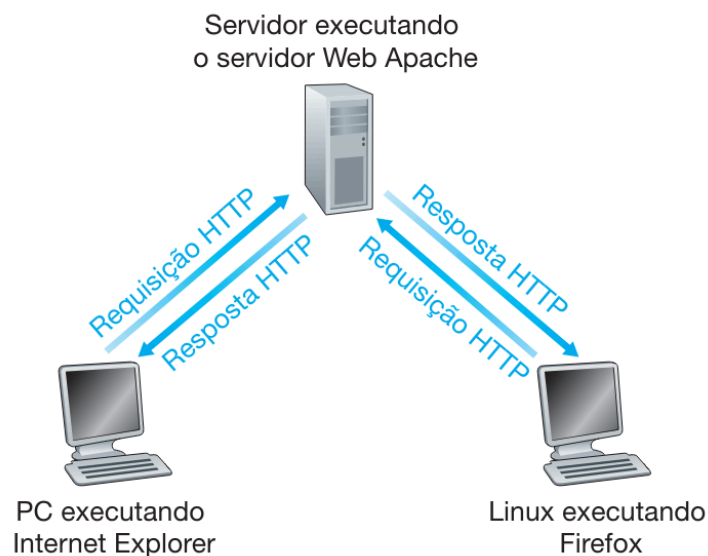
2.2 HTTP: COMO FUNCIONA A WEB

A base para o funcionamento da web é dada pelo protocolo HTTP (*Hypertext Transfer Protocol*), traduzindo para o português - Protocolo de transferência de Hipertexto. Segundo Kurose e Ross (2013), o protocolo HTTP é dividido em duas partes:

- 1º - Programa cliente: É relacionado ao usuário final quando acessa alguma página ou recurso da web.
- 2º - Programa servidor: É relacionado a uma aplicação que espera passivamente de forma assíncrona por uma conexão, com o propósito de responder a esta conexão.

O programa cliente e o programa servidor conversam por meio de mensagens HTTP, onde o cliente envia uma requisição e o servidor devolve uma resposta. Conforme mostra a Figura 2:

Figura 2 - Comunicação entre cliente e servidor.



Fonte: Kurose; Ross, 2013

Ou seja, quando um computador acessa uma URL, como: <https://www.portal.ufpa.br/>, o navegador envia ao servidor mensagens de requisição HTTP, o servidor recebe as requisições e responde com mensagens de respostas HTTP que contém o conteúdo desta aplicação web, que neste caso é o Portal da

Universidade Federal do Pará. De acordo com Tanenbaum (2003), URL é um ponteiro que aponta para um determinado local, neste caso é para uma página web.

As páginas web são constituídas por documentos ou arquivos de vários formatos, essas páginas são construídas com HTML (*HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto) é a linguagem base para o desenvolvimento de páginas web.

2.2.1 HTTP: Métodos

O protocolo HTTP possui uma grande quantidade de métodos que são utilizados na comunicação entre cliente e servidor. Os principais métodos HTTP estão listados na Tabela 1:

Tabela 1 - Principais métodos HTTP

GET	Método responsável por solicitar algum recurso ao servidor
POST	Método responsável por enviar dados ou formulário HTML ao servidor
DELETE	Método responsável por informar por meio da URL o objeto a ser deletado.
PUT/PATCH	Método responsável por atualizar algum dado do servidor

Fonte: Tabela elaborada pelo autor.

2.2.2 Mensagem de requisição HTTP

Cada método listado na Tabela 1 é responsável por tratar a conversa entre cliente e servidor de uma forma, por exemplo, o método GET é uma solicitação ao servidor de envio de uma página web:

GET /somedir/page.html HTTP/1.1

Host: www.someschool.edu

Connection: close

User-agent: Mozilla/5.0

Accept-language: fr

Acima é listado um exemplo de requisição ao servidor, onde esta requisição está solicitando o recurso: /somedir/page.html através do método GET, além disso, é possível notar que este recurso se encontra no hospedeiro: www.someschool.edu. O campo *Connection: close*, o navegador está comunicando ao servidor que feche a conexão após o envio do recurso requisitado, o campo: *User-agent: Mozilla/5.0* informa qual o navegador que está realizando esta requisição, o campo: *Accept-language*: mostra qual o idioma preferível que o usuário espera receber o conteúdo, neste caso é em uma versão em francês. (TANEMBAUM, 2003)

2.2.3 Mensagem de resposta HTTP

Abaixo será apresentada uma mensagem de resposta HTTP, que pode-se considerar uma resposta ao exemplo de mensagem de requisição que foi discutido na Seção 2.2.2

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(dados dados dados dados ....)
```

A resposta HTTP é composta por: linha de estado, linhas de cabeçalho e o corpo da mensagem que é a parte principal da mensagem.

A linha de estado informa: a versão do protocolo — HTTP/1.1, código de estado — 200 e a mensagem do estado — OK. Neste exemplo, tudo ocorreu bem, e o servidor está enviando ao cliente o recurso solicitado.

As linhas de cabeçalho informam: *Connection: close* — diz que o servidor irá fechar a conexão após enviar a resposta, *Date* — indica a data e hora em que a resposta HTTP foi enviada pelo servidor, *Server* — indica qual o tipo do servidor, *Last-modified* — indica a data e a hora em que o objeto sofreu a última modificação,

Content-Length — indica o número de bytes do objeto que será enviado, *Content-Type* — indica o formato do corpo da mensagem. (TANEMBAUM, 2003)

2.3 ENGENHARIA DE SOFTWARE

A engenharia de software pode ser definida por todos os métodos, procedimentos e ferramentas necessárias para desenvolver um software de qualidade de acordo com as necessidades do cliente (NOGUEIRA, 2008).

Os métodos são um conjunto de metas que constroem um objetivo (*software*), como o planejamento e estimativa de entrega do projeto, análise de requisitos do software, desenvolvimento, entrevista com usuários, testes automatizados.

Por outro lado, as ferramentas oferecem apoio aos métodos. Por exemplo: PHPStorm⁵ e VsCode⁶ oferecem suporte como IDE (Integrated Development Environment) no desenvolvimento de aplicações.

As técnicas são as sequências em que os métodos serão aplicados juntamente com o uso das ferramentas.

Para Sommerville (2011), a sociedade cada vez mais está dependente de sistemas de software, por isso devemos ser capazes de desenvolver sistemas robustos e duradouros rapidamente, para isso, a seguir é apresentada a ideia de como funciona um processo de construção de software.

2.3.1 Processo de software

O processo de software é um conjunto coerente de atividades a serem realizadas para a produção de um software. De acordo com Sommerville (2011), há diversos processos que levam ao desenvolvimento de um software, no entanto, todos devem incluir 4 atividades fundamentais para a engenharia de software:

1. **Especificação de software:** Atividade responsável por definir e restringir os requisitos da solução.
2. **Projeto de implementação de software:** Atividade responsável por desenvolver uma solução que cumpra com os requisitos especificados.

⁵ Disponível em: <https://www.jetbrains.com/pt-br/phpstorm/>. Acesso em: abril. 2024

⁶ Disponível em: <https://code.visualstudio.com/>. Acesso em: abril. 2024

3. **Validação de software:** Atividade responsável por testar e garantir que a solução não tenha erros e supra a necessidade do cliente.
4. **Evolução do software:** Atividade responsável por analisar se é factível implementar possíveis mudanças futuras na solução desenvolvida.

Sommerville (2011) cita que cada uma dessas atividades fundamentais são complexas e possuem uma gama de sub-atividades para a sua realização, por exemplo: validação de requisitos, projeto de arquitetura, testes unitários, testes de integração e etc.

2.3.2 Modelo de Software

Um modelo de software pode ser entendido como uma representação abstrata de como vai ocorrer o processo de construção de um software, pois cada modelo representa um processo a partir de uma perspectiva particular, oferecendo uma estrutura que orienta as atividades de desenvolvimento (SOMMERVILLE, 2011).

Alguns modelos de software são:

- **Modelo em Cascata:** Este modelo é caracterizado por ser linear e sequencial. Na primeira vez que uma fase é complementada, o desenvolvimento segue para a próxima fase e não há retorno (SOMMERVILLE, 2011).
- **Desenvolvimento Iterativo Incremental:** Essa estratégia integra as etapas de especificação, desenvolvimento e validação. O sistema é elaborado por meio de uma progressão incremental, onde cada versão sucessiva expande as funcionalidades presentes na anterior (SOMMERVILLE, 2011).
- **Desenvolvimento Iterativo Espiral:** Uma abordagem dinâmica e adaptativa que integra ciclos de desenvolvimento sucessivos, representados como uma espiral, onde cada volta é uma iteração que aborda várias fases do processo de desenvolvimento de software, como análise de requisitos, design, implementação e teste (SOMMERVILLE, 2011).

2.4 ARQUITETURA DE SOFTWARE

Santana (2011) descreve que para melhorar a qualidade de um software é necessário adotar padrões de desenvolvimento, dado que as aplicações necessitam de manutenções com foco na correção de erros, adição e remoção de funcionalidades. A partir destes padrões, moldar, descrever subsistemas e componentes: a isto é dado o nome de Arquitetura de Software.

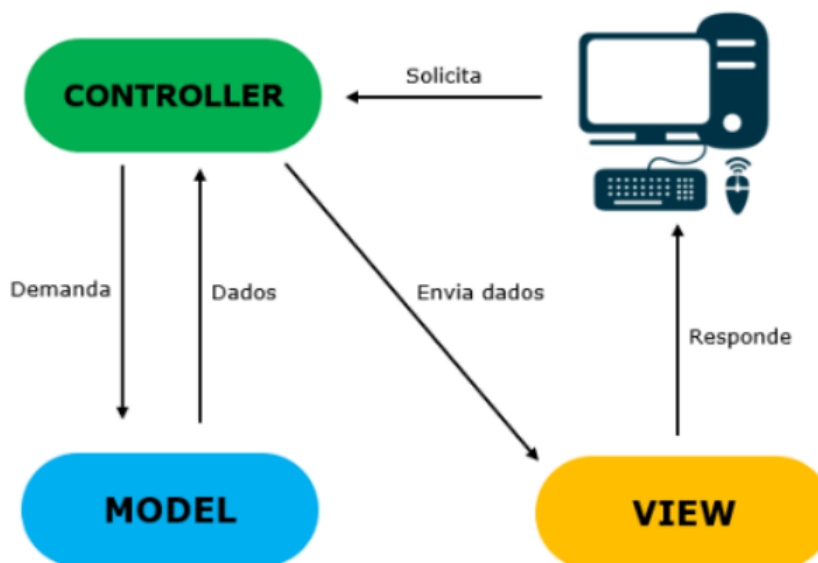
2.4.1 Arquitetura MVC (Model View Controller)

A arquitetura de software MVC (Model-View-Controller) divide a aplicação em três componentes principais:

- **Model (Modelo):** Componente responsável por representar uma entidade e gerenciar a forma como os dados se comportam por meio de métodos e regras de negócios estabelecidas.
- **Controller (Controlador):** Componente responsável por intermediar a requisição enviada pelo componente *View* com as respostas fornecidas pelo componente *Model*. Em outras palavras, o *controller* permite a comunicação entre um setor e outro da aplicação.
- **View (Visão):** Componente responsável por apresentar as informações de forma visual e agradável ao usuário da aplicação, ou seja, utilizando recursos como: cores, botões, imagens e etc.

Na Figura 3, é mostrado o funcionamento da arquitetura MVC. Nesse contexto, quando ocorre uma solicitação de dados por parte do cliente, essa requisição é inicialmente direcionada ao controlador (Controller). O controlador, por sua vez, encaminha essa demanda ao modelo (Model), o qual tem a responsabilidade de recuperar os dados solicitados. Após a obtenção dos dados, o modelo retorna ao controlador. Em seguida, o controlador encaminha os dados à camada de visualização (View), que então os apresenta ao usuário por meio de uma interface gráfica.

Figura 3 - Funcionamento da arquitetura MVC



Fonte: Oliveira, 2019

A arquitetura MVC auxilia a equipe de desenvolvedores no manuseio do código de uma aplicação, principalmente em sistemas extensos e complexos, devido às divisões de cada tipo de lógica estar bem delegada e separada em componentes de manipulações de entrada e saída de dados, acesso ao banco de dados, construção da visualização para o usuário e etc (SANTOS, 2010).

Além do mais, esta arquitetura é demasiadamente usada em soluções web, muitos *frameworks* como: Laravel⁷, Symfony⁸, Spring Boot⁹, Django¹⁰ e dentre outros implementam essa arquitetura devido a facilidade de uso e padronização da comunidade.

2.5 BANCO DE DADOS

Em um projeto de software é comum se deparar com situações em que seja necessário armazenar informações, com a finalidade de que sejam recuperadas posteriormente, por isso, surge a necessidade do uso de um banco de dados. Segundo Rob e Coronel (2011), banco de dados (ou base de dados) é uma estrutura computacional compartilhada e integrada que armazena um conjunto de dados de

⁷ Disponível em: <https://laravel.com/>. Acesso em: abril. 2024.

⁸ Disponível em: <https://symfony.com/>. Acesso em: abril. 2024.

⁹ Disponível em: <https://spring.io/>. Acesso em: abril. 2024.

¹⁰ Disponível em: <https://www.djangoproject.com/>. Acesso em: abril. 2024.

usuário final, metadados e configurações da aplicação. Para isso, é necessário utilizar softwares como: MySQL¹¹, Oracle¹² e PostgreSQL¹³, pois eles gerenciam a estrutura do banco de dados e controlam o acesso aos dados armazenados. Esses programas são chamados de SGBD (Sistema de Gerenciamento de Banco de Dados).

2.5.1 MySQL

O MySQL é um dos sistemas de gerenciamento de banco de dados mais populares e amplamente utilizados em projetos de software em todo o mundo. Onde a estrutura fundamental deste tipo de banco é o registro de informações em linhas de tabelas, e categorizado o tipo de informação por colunas ou atributos. Ademais, o modelo relacional pode ser caracterizado como uma coleção de relações, em outras palavras, os registros das tabelas podem ter relação com outras tabelas. (ELMASRI, 2011)

Algumas das vantagens do uso do MySQL são: (a) Portabilidade, pois é possível utilizar em conjunto com diversas linguagens de programação; (b) Armazenamento, pelo fato de disponibilizar vários tipos de tabelas para armazenamento; (c) Confiabilidade, pois sua arquitetura robusta e recursos avançados de replicação e recuperação de falhas (MILANI, 2007).

O MySQL utiliza a linguagem SQL (*Structured Query Language*) que é altamente veloz para fazer buscas em tabelas de armazenamento. Por exemplo, dado uma tabela de “MEUS_USUARIOS”, é possível buscar todos os registros usando o seguinte SQL: “SELECT * FROM MEUS_USUARIOS”. O resultado obtido será uma lista com linhas e colunas com todos os registros da tabela.

¹¹ Disponível em: <https://www.mysql.com/>. Acesso em: abril. 2024.

¹² Disponível em: <https://www.oracle.com/br/>. Acesso em: abril. 2024.

¹³ Disponível em: <https://www.postgresql.org/>. Acesso em: abril. 2024.

3 REVISÃO DA LITERATURA

Conforme foi exposto, o objetivo do trabalho é criar um *e-commerce* para os agricultores familiares das regiões de Tucuruí, Belém, Cametá e Breu Branco. Diante disso, a busca por trabalhos relacionados foi conduzida com o propósito de encontrar: (a) estudos que abordassem o processo de construção de um *e-commerce* para o público em questão, (b) mapeamento de *e-commerces* para agricultura familiar (c) e se existe alguma plataforma ativa na região do Pará.

Portanto, essas buscas foram direcionadas para as bases de dados do Periódicos Capes e do *Google Scholar*, adotando uma abordagem de revisão exploratória e qualitativa da literatura.

3.1 PALAVRAS-CHAVE

Para a pesquisa foram selecionadas as seguintes palavras-chave:

- Agricultura Familiar / *Family Farming*
- Sistema Web / *Web System*
- Plataforma Digital / *Digital Platform*
- Comércio Eletrônico / *E-commerce*
- Pará
- Marketplace

3.2 STRINGS DE BUSCA

Em seguida, foram combinadas as palavras-chave mencionadas para formar as seguintes *strings* de buscas:

3.2.1 Pesquisas realizadas no Periódicos Capes

- “Family Farming” AND “E-commerce”
- “Digital Platform” AND “Family Farming”
- “Web System” AND “Family Farming”
- “Agricultura Familiar” AND “E-commerce”
- “Agricultura Familiar” AND “E-commerce” AND “Pará”

- “Marketplace” AND “Family Farming”

3.2.2 Pesquisas realizadas no Google Scholar

- “Agricultura Familiar” AND “Comércio Eletrônico”
- “Plataforma Digital” AND “Agricultura Familiar”
- “Agricultura Familiar” AND “Comércio Eletrônico” AND “Pará”
- “Sistema Web” AND “Agricultura Familiar”
- “Sistema Web” AND “Agricultura Familiar” AND “Pará”
- “Marketplace” AND “Agricultura Familiar”

Foram estabelecidos critérios para a seleção dos materiais utilizados neste trabalho, através da definição de parâmetros de inclusão e exclusão, os quais serão descritos a seguir:

3.3 LISTA DE TRABALHOS ENCONTRADOS

3.3.1 Lista de trabalhos incluídos

A Tabela 2 mostra a lista de trabalhos incluídos nesta revisão. Os critérios de inclusão foram: (a) artigos e trabalhos publicados em periódicos confiáveis, (b) estudos relacionados a plataformas digitais para o público da agricultura familiar e (c) trabalhos que abordem a realidade dos meios digitais no contexto do estado do Pará.

Tabela 2 - Lista de trabalhos incluídos

Nº	Título	Autores	Veículo
1	Reinvenção dos mercados da agricultura familiar no Brasil: a novidade dos sites e plataformas digitais de comercialização em tempos de Covid-19	Marcio Gazolla; Joacir Rufino de Aquino	Estudos Sociedade e Agricultura, v. 29, n. 2, 2021.

2	Feira na palma da mão: uma plataforma digital para a venda direta dos produtos da agricultura familiar.	Luiz Carlos Leonardi Bricalli; Fabiane de Souza Correia; Joelma de Carvalho Barbosa; Nathália Zouain Messina	ETHNOSCIENTIA – ano 07, número 04 - 2022 - [ISSN: 2448-1998]
3	Análise e Desenvolvimento de um aplicativo e-commerce para a comercialização de produtos oriundos da agricultura familiar.	Guilhermy William Macedo de Andrade	Instituto Federal Goiano
4	Mercados alimentares digitais: inclusão produtiva, cooperativismo e políticas públicas	Paulo Niederle; Sergio Schneider; Abel Cassol	Editora Universidade Federal do Rio Grande do Sul
5	"Marketplace": uma nova esfera pública sobre os canais de mercados	Diego da Silva Smith; Adebaro Alves dos Reis; Maria José de Souza Barbosa; Miranilde Oliveira Neves	Cuadernos de Educación y Desarrollo. Vol. 15 No. 6 (2023)

Fonte: Tabela elaborada pelo autor.

3.3.2 Lista de trabalhos excluídos

A Tabela 3 mostra a lista de trabalhos excluídos desta revisão, os critérios de exclusão foi qualquer pesquisa relacionada a *e-commerces* que não estivessem voltadas diretamente para agricultores.

Tabela 3 - Lista de trabalhos excluídos

Nº	Título	Autores	Veículo
1	Extração de conhecimento no sistema de comércio eletrônico da livraria EMBRAPA	Natália Santos Fois	Universidade Federal do Rio de Janeiro
2	Tecnologia da informação e desenvolvimento local: O comércio eletrônico na promoção do comércio justo	Luiz Cláudio Santos, Karina Rocha	Universidade Federal de Goiás, Brasil
3	Tecnologias que auxiliam a produção sustentável de alimentos	Darlene Sausen; Lucas Palhares Marques; Maele Cavalcante de Melo; Gabriela de Araújo Costa; Letícia Bezerra Azevedo	Revista Eletrônica Competências Digitais para Agricultura Familiar
4	Registration and Georeferencing of the Family Farming Production Chain in Itapúa Department, Paraguay	Hector T. Roos; Fabio S. Pires; Enio Giotto; Bruna D. Pimenta; Enrique O. H. Villalba	Journal of Agricultural Science [1916-9752] Roos, Hector Ano:2019 v.:11 n.:5 p.:161
5	Sistema Fuzzy para Controle da Produtividade Leiteira no Âmbito da Agricultura Familiar do Sudoeste Goiano	Isaque Elcio Souza; Tatiane Melo Lima; Paulo Afonso Parreira Júnior; Paulo Filho Hellmeister	Departamento de Informática e Matemática Aplicada - DIMAp - UFRN

Fonte: Tabela elaborada pelo autor.

3.4 ANÁLISE DOS TRABALHOS INCLUÍDOS

No trabalho de Gazolla e De Aquino (2021), são analisados os processos de reinvenção dos mercados da agricultura familiar brasileira durante a pandemia da Covid-19. Os autores fizeram um levantamento das plataformas digitais que eram destinadas exclusivamente à venda de produtos dos agricultores familiares.

Em princípio, os autores chegaram no número de 50 sites e plataformas de vendas, contudo, foram descartadas 12 plataformas, os motivos da exclusão foram: (a) o comércio de produtos não alimentares, o qual foge do objetivo proposto pelos autores na pesquisa, (b) plataformas que possuíam públicos mistos, (c) outros sites que não apresentavam informações claras sobre os objetivos dos mesmos, e (d) em muitas plataformas remetiam apenas a número de WhatsApp.

Em consequência da exclusão, resultou-se em 38 sites e plataformas digitais para o prosseguimento da pesquisa. A Tabela 4 ilustra os resultados obtidos com sua respectiva localização no território nacional.

Tabela 4 - Distribuição macrorregional dos mercados digitais da agricultura familiar no território brasileiro.

Regiões	Número	Percentual %
Nordeste	14	36,85
Sul	12	31,57
Centro-Oeste	7	18,42
Sudeste	5	13,16
Norte	0	0,00
Total	38	100,00

Fonte: Gazolla; De Aquino, 2021.

Segundo a Tabela 4, a distribuição de plataformas digitais por região, apresenta a região Nordeste e Sul que detêm as maiores porcentagens de números de sites levantados, 36,85% e 31,57% respectivamente. Seguidos das regiões Centro-Oeste com 18,42% e Sudeste com 13,6%. Por fim, a região Norte com um percentual de 0,00%. O percentual referente a região Norte, é influenciado pelos motivos de: (a) pouco acesso à internet, (b) distância dos principais mercados

consumidores de alimentos e (c) dificuldades logísticas. Esses fatores dificultam a comercialização por meios digitais, portanto, não significa que na região Norte não exista mercado digital dos agricultores familiares, pois há outros meios digitais não investigados pela pesquisa, tais como os aplicativos WhatsApp, Instagram e Facebook que podem estar sendo utilizados (GAZOLLA; DE AQUINO, 2021).

As 38 plataformas digitais, citadas na Tabela 4 como total, foram descritas pelo autor na Tabela 5:

Tabela 5 - Plataformas pesquisadas no território brasileiro em 2021 para a Agricultura Familiar

Nº	Nome	Endereço eletrônico	Município e UF
1	Agro Orgânica	www.agroorganica.com.br	Brasília/DF
2	Alimento de Origem	vitrine.alimentodeorigem.com.br/cep	Frederico Westphalen/R S
3	Amazônia Hub	amazoniahub.com	São Paulo/SP
4	Aprofam	aprofam.vendizap.com	Mossoró/RN
5	Bahia Cacau	bahiacacau.com.br	Ibicaraí/BA
6	Balaio Orgânico	www.balaioorganico.com.br/produtos	Curitiba/PR
7	Biobá	bioba.com.br/#/loja	Brasília/DF
8	Carona Agro	kyte.site/carona-agro	Chapecó/SC
9	Cecafes	loja.cecafes.net.br/	Natal/RN
10	Central do Cerrado	www.centraldocerrado.org.br/loja	Brasília/DF
11	Cesta Saudável	fb.cestasaudavel.com.br	Francisco Beltrão/PR
12	Cirandas	cirandas.net/frutodaterra	Recife/PE
13	Comadre Fulozinha	www.comadrefulozinha.com.br	Recife/PE
14	Compre do Produtor	guarapuava.compredoprodutor.com.br	Guarapuava/PR
15	Coopafs	sarandi.feiradopequenoprodutor.com.br/	Sarandi/RS
16	CoopHub	centraldacaatinga.com.br	Salvador/BA

17	Coeres da Terra	cooperativacoeresdaterra.com.br	Cascavel/PR
18	Direto da Roça	diretodarocamt.com.br	Tangará da Serra/MT
19	Empório da AF	balcao.online/coophub	Juazeiro/BA
20	Enafes	kyte.site/enaf--escritorio-de-negocios-da-af	Cuiabá/MS
21	Feira em Casa	feiraemcasa.caaf.agr.br	Caxias do Sul/RS
22	Feira da AF Delivery	joorafaelasantos2.wixsite.com/feiradaagricultura	Janaúba/MG
23	FrigBahia	frigbahia.com.br	Pintadas/BA
24	Girassol	coopgirassol.com.br/loja	Porto Alegre/RS
25	Gravetero	www.coopercuc.com.br	Uauá/Curaçá/Canudos/BA
26	Junta Local	juntalocal.com	Rio de Janeiro/RJ
27	Manuçã	manuca.semagro.ms.gov.br	Dourados/MS
28	Monte Sabores	balcao.online/montesabores	Monte Santo/BA
29	Natucoa	www.natucoa.com.br	Ilhéus/BA
30	Naturingá	naturinga.commercesuite.com.br	Maringá/PR
31	Porta Porta Orgânicos	www.portaaportaorganicos.com.br	Recife/PE e Região
32	Sítio a Boa Terra	www.aboaterra.com.br	Itobi/SP
33	Sítio Recanto Nativo	www.sitiorecantonativo.com.br	Campo Magro/PR
34	Orgânicos da Fátima	organicosdafatima.com.br	Rio de Janeiro/RJ
35	Orgânicos da Zabelê	www.organicosrn.eco.br	São Raimundo Nonato/PI
36	Verduras Luzzi	www.verdurasluzzi.com.br	Chapecó/SC
37	Vitrine da AF	sistemas.agricultura.gov.br/vitrine/que-m-somos	Brasília/DF

38	Vitrine Virtual	www.vitrinevirtualrn.com	Mossoró/RN
----	-----------------	--------------------------	------------

Fonte: Gazolla; De Aquino, 2021.

Com base nos dados apresentados na Tabela 5, o presente trabalho buscou realizar investigações nas plataformas mencionadas, caracterizando, assim, uma abordagem de fonte primária de informação. Portanto, foram acessadas todas as 38 plataformas com os seguintes objetivos: (a) analisar suas funcionalidades principais, (b) verificar a disponibilidade atual para acesso e (c) examinar a presença de produtos da agricultura familiar. Os resultados dessas investigações estão detalhados na Tabela 6.

Tabela 6 - Análise das plataformas digitais da agricultura familiar mencionadas por Gazolla e De Aquino (2021)

Nº	Nome	Está disponível?	Características e funcionalidades
1	Agro Orgânica	Não	Sem características e funcionalidades, pois está indisponível.
2	Alimento de Origem	Sim	Comercialização de produtos de agroindústrias e agricultura familiar; Plano gratuito e pago; Informações de origem dos produtos, local de produção, quantidades e os períodos do ano disponíveis para comercialização; Disponível para os estados: RS, PR, SC, MG, SP, BA, RN, RJ, PE e CE; Página para os vendedores com seus produtos disponíveis; Interface pouco intuitiva.
3	Amazônia Hub	Sim	Comercialização de produtos derivados de matéria prima da amazônia; não há perfil de agricultores com seus produtos; página inicial com produtos disponíveis da plataforma; Conteúdo para blog; Programa de afiliados.
4	Aprofam	Não	Sem características e funcionalidades, pois está indisponível.
5	Bahia Cacau	Sim	Marca que comercializa produtos derivados da produção da agricultura familiar; destinado a vender chocolates; não há perfil de vendedores na plataforma; página inicial com produtos disponíveis da plataforma.
6	Balaio Orgânico	Não	Sem características e funcionalidades, pois está indisponível.

7	Biobá	Sim	Comercializa produtos provenientes da agricultura familiar; não há perfil de agricultores com seus produtos; página inicial com produtos disponíveis da plataforma; Ambiente de venda B2B, ou seja, intermediária entre produtores e outras entidades comerciais.
8	Carona.Agro	Sim	Comercializa produtos da agricultura familiar; não há páginas de perfil de agricultor; página inicial com produtos disponíveis da plataforma; Categoria de produtos bem definidos na plataforma; Alguns produtos não tem foto.
9	Cecafes	Não	Sem características e funcionalidades, pois está indisponível.
10	Central do Cerrado	Sim	Comercializa produtos da agricultura familiar; não há páginas de perfil de agricultor; página inicial com produtos disponíveis da plataforma; Localização para compra presencial.
11	Cesta Saudável	Sim	A Plataforma está online, mas não há produtos para serem comercializados; não há perfil de agricultores.
12	Cirandas	Sim	Comercialização de produtos da agricultura familiar, não há perfil de agricultores; página inicial com produtos disponíveis da plataforma; os anúncios não possuem foto; há bastante informação na plataforma, é difícil distinguir o propósito.
13	Comadre Fulozinha	Sim	Comercialização de produtos da agricultura, mas não é informado se é da agricultura familiar; existe página de agricultores, mas está com dados genéricos; página de agricultores não existe anúncio de produtos; página inicial com produtos disponíveis da plataforma; categoria de produtos bem definida.
14	Compre do Produtor	Sim	Comercialização de produtos da agricultura familiar; não há perfil de agricultores; página inicial com produtos disponíveis na plataforma; plataforma bem organizada com informações de data de entrega, como funciona os pedidos e produtos com fotos;
15	Coopafs	Não	Sem características e funcionalidades, pois está indisponível.
16	CoopHub	Sim	Comercialização de produtos da agricultura familiar; seção destinada a notícias e blog; não há perfil de agricultor; página de produtos destinada a exibir os produtos disponíveis na plataforma.

17	Coors da Terra	Sim	Comercialização de produtos da agricultura familiar; não há perfil de agricultor; página de produtos destinada a exibir os produtos disponíveis na plataforma; página de receitas e blog.
18	Direto da Roça	Não	Sem características e funcionalidades, pois está indisponível.
19	Empório da AF	Não	Sem características e funcionalidades, pois está indisponível.
20	Enafes	Sim	Comercializa produtos da agricultura familiar; não há páginas de perfil de agricultor; página inicial com produtos disponíveis da plataforma; Categoria de produtos bem definidas na plataforma; Alguns produtos não tem foto- Design pode melhorar.
21	Feira em Casa	Não	Sem características e funcionalidades, pois está indisponível.
22	Feira da AF Delivery	Não	Sem características e funcionalidades, pois está indisponível.
23	FrigBahia	Sim	Divulga produtos da agricultura familiar; não comercializa produtos na plataforma; compartilha localização para realizar a compra presencialmente; página de blog.
24	Girassol	Sim	Plataforma está online, mas não há produtos para serem comercializados.
25	Gravetero	Sim	Divulga produtos da agricultura familiar; não comercializa produtos na plataforma; compartilha localização para realizar a compra presencialmente; página de blog.
26	Junta Local	Sim	Divulga produtos da agricultura familiar; comercializa assinatura de cesta de produtos da agricultura familiar, onde é possível escolher cestas já prontas ou montar uma cesta personalizada; não há página de agricultores.
27	Manucã	Sim	Comercializa produtos da agricultura familiar; não há perfil de agricultores; atualmente não possui anúncios ativos.
28	Monte Sabores	Não	Sem características e funcionalidades, pois está indisponível.
29	Natucoa	Sim	Comercializa produtos provenientes da agricultura familiar; não há página de agricultores; página inicial

			com produtos disponíveis para compra.
30	Naturingá	Não	Sem características e funcionalidades, pois está indisponível.
31	Portal Porta Orgânicos	Não	Comercializa produtos da agricultura familiar; não há perfil de agricultores; página de blog; categorias bem definidas; página inicial com produtos disponíveis para compra; programa de fidelização.
32	Sítio a Boa Terra	Sim	Comercializa produtos da agricultura familiar; não há perfil de agricultores; página de blog; comercializa assinatura de cesta de produtos da agricultura familiar, onde é possível personalizar a cesta de acordo com a assinatura escolhida.
33	Sítio Recanto Nativo	Sim	Comercializa produtos da agricultura familiar; não há perfil de agricultores; lista de depoimentos; página inicial com produtos disponíveis para compra.
34	Orgânicos da Fátima	Sim	Comercializa produtos da agricultura familiar em forma de assinatura de cestas, onde o usuário pode montar sua cesta de acordo com o plano escolhido; não há página do agricultor.
35	OrgânicosZa belê	Não	Sem características e funcionalidades, pois está indisponível.
36	Verduras Luzzi	Sim	Comercializa produtos da agricultura familiar; pelo fato de ser uma empresa familiar, não há perfil do agricultor; página inicial com produtos disponíveis para compra.
37	Vitrine da AF	Sim	Divulga produtos da agricultura familiar; não comercializa produtos na plataforma; página de blog.
38	Vitrine Virtual	Não	Sem características e funcionalidades, pois está indisponível.

Fonte: Tabela elaborada pelo autor

Analisando os dados da Tabela 6, observa-se que das 38 plataformas mencionadas por Gazolla e De Aquino (2021), 24 delas estão operacionais, o que representa aproximadamente 63,2% do total. Enquanto isso, 14 plataformas foram desativadas, correspondendo a 36,8%.

No trabalho de Andrade (2019), foi descrita a concepção e implementação de um aplicativo móvel destinado a oferecer uma plataforma inovadora para apoiar a comercialização dos produtos da agricultura familiar. O principal objetivo do

aplicativo é simplificar o processo de venda de produtos, agilizar a realização de pedidos e introduzir a praticidade do comércio eletrônico na comercialização dos itens provenientes da agricultura familiar.

No trabalho de Nierdele, Schneider e Cassol (2021), foi destacado o processo de digitalização nos meios de produção, como a adesão de agricultores familiares à plataformas digitais. Onde essas plataformas, mostravam o catálogo de produtos disponíveis, e que permitia a comercialização dos mesmos.

No trabalho de Da Silva Smith (2023), foi realizada uma revisão da literatura dos *marketplaces* com um destaque para o público da agricultura familiar, onde afirma que os mercados digitais para esse meio precisam ter estratégias a serem mobilizadas. O autor ainda exalta que a existência dessa modalidade no Estado do Pará é desconhecida, muito pelo motivo de não haver debate sobre a temática na região, o que reflete na problemática de acesso aos benefícios dados pelo meio digital.

Quanto ao trabalho de Bricalli (2022), foi desenvolvido um aplicativo móvel disponível para dispositivos *Android* e *IOS*, denominado de “Feira Palma na Mão”, para a comercialização de produtos da agricultura familiar, com foco no estado do Espírito Santo. O APP foi desenvolvido por uma empresa contratada, enquanto os pesquisadores realizaram testes de funcionalidade e capacitação aos agricultores para o uso da plataforma.

Quando um pedido é realizado pelo aplicativo “Feira Palma na Mão”, o consumidor é encaminhado para o *WhatsApp* do produtor, onde é possível combinar detalhes de entrega ou retirada, bem como verificar a quantidade solicitada. Embora o APP ainda não tenha sido lançado, acredita-se que será bem recebido pela comunidade devido às suas funcionalidades, como uma interface intuitiva, um painel detalhando os pedidos, perfis dos produtores e uma seção mostrando os produtos disponíveis.

3.5 LACUNAS EXISTENTES

A revisão da literatura realizada neste trabalho revela lacunas significativas no contexto do comércio eletrônico voltado para a agricultura familiar, especialmente na região Norte. Embora existam *e-commerces* voltados para esse segmento em outras

regiões, a escassez de plataformas no Norte do país cria uma oportunidade para explorar o desenvolvimento de um modelo específico adaptado para a região.

No trabalho de Andrade (2019) foi destacada uma lacuna no uso de aplicativos de mensagens instantâneas, como o WhatsApp, pelos agricultores familiares em Goiás. A dificuldade em centralizar informações de vendas, como endereço de entrega, produtos e valores, evidenciou a necessidade de uma plataforma mais específica para o setor agrícola. Este problema, compartilhado pelos agricultores paraenses, sinaliza a importância de uma abordagem mais adaptada às necessidades específicas da agricultura familiar.

Ao analisar as plataformas estudadas por Gazolla e De Aquino (2021), nota-se uma tendência em que muitas funcionam como intermediárias entre os agricultores e os consumidores, assumindo o gerenciamento dos anúncios dos produtos em vez de permitir que os agricultores tenham controle sobre seus próprios perfis na plataforma. Essa dinâmica sugere uma possível limitação na autonomia e no empoderamento dos agricultores familiares no ambiente digital.

Outro ponto relevante a considerar é que a aplicação “Feira Palma na Mão” de Bricalli (2022) foi desenvolvida exclusivamente para o uso de dispositivos móveis. Uma abordagem complementar a ser explorada é o desenvolvimento de uma aplicação *web* responsiva, acessível tanto por meio de computadores quanto de dispositivos móveis por meio de navegadores de internet, ampliando assim o alcance e a acessibilidade da plataforma.

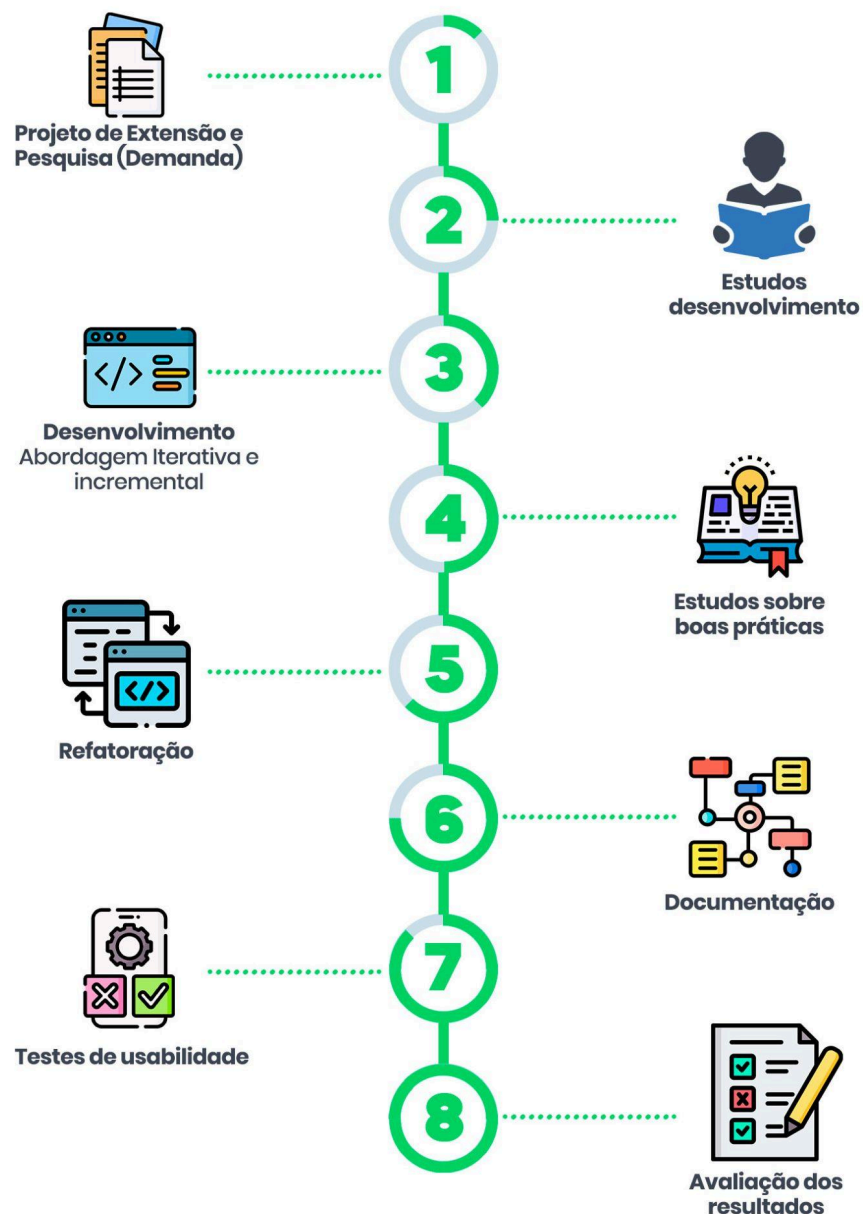
Nierdele, Schneider e Cassol (2021) traz uma preocupação relevante sobre a possível desigualdade gerada pela digitalização no campo, onde dificuldades no manejo de dispositivos digitais e o acesso limitado à internet podem afetar alguns agricultores. Logo ele destaca que esse problema poderia ser adequadamente solucionado se as cooperativas e associações de agricultores fossem apoiados por políticas públicas.

Com base nessas análises, fica evidente a necessidade de mais projetos que visam corroborar com a temática de inserção dos agricultores familiares no comércio eletrônico, principalmente na região Norte brasileira. Esses projetos podem contribuir significativamente para a redução das disparidades regionais e para o fortalecimento da economia local.

4 METODOLOGIA

Este capítulo se caracteriza por descrever detalhes sobre todo o processo envolvido no desenvolvimento deste software, desde a demanda até a avaliação dos resultados. A Figura 4 apresenta de forma simplificada a metodologia adotada neste trabalho, com ênfase no progresso da construção da plataforma destinada à Feira da Agricultura Familiar dos municípios envolvidos.

Figura 4 - Metodologia e linha do tempo seguida neste trabalho



Fonte: Imagem elaborada pelo autor.

Como ilustra a Figura 4, o projeto foi desenvolvido em 8 etapas: Projeto de Extensão e Pesquisa (Demanda); Estudos (Desenvolvimento); Desenvolvimento; Estudos (Boas práticas); Refatoração; Documentação; Testes de usabilidade e Avaliação dos resultados. Tais etapas podem ser descritas da seguinte forma:

4.1 PROJETO DE EXTENSÃO E PESQUISA - DEMANDA

Em meados de 2020, na UFPA Campus Tucuruí, foi iniciada uma oportunidade de participação em um Projeto de Extensão e Pesquisa voltado para o desenvolvimento de uma plataforma digital. Este projeto visava fornecer suporte aos agricultores familiares das regiões de Tucuruí, Breu Branco, Belém e Cametá nas vendas de seus produtos.

Nessa fase inicial do projeto foram seguidas algumas boas práticas da metodologia ágil SCRUM, como reuniões para definir requisitos do sistema. Nessas sessões, participaram três profissionais-chave, sendo eles: um *Product Owner*, responsável por definir as prioridades e metas do produto; um *Scrum Master*, encarregado de facilitar o processo e remover obstáculos; e um Desenvolvedor, para implementar as funcionalidades necessárias.

A seguir, apresenta-se uma descrição detalhada desses requisitos.

4.1.1 Requisitos do sistema

Nesta fase inicial do projeto, a equipe de desenvolvimento era composta apenas pelo autor deste trabalho. O autor é graduando da faculdade de Engenharia de Computação, e não havia cursado a disciplina de Engenharia de Software naquele momento, logo ele não tinha nenhum conhecimento sobre os processos que envolviam a Engenharia de Software. Além disso, os responsáveis pela demanda eram profissionais de outras áreas, fora da computação. Em consequência disso, os requisitos do sistema foram descritos em um documento no *Google Docs*.

O documento de requisitos preliminar continha as seguintes informações:

- **Informações pessoais:** Dados dos agricultores e clientes, como nome completo, telefone, bairro, rua, número, cep, email e senha.

- **Tipos de usuários:**
 - **Agricultores:** aqueles que podem comprar e vender produtos pela plataforma;
 - **Clientes:** aqueles que somente realizam compra de produtos;
 - **Administrador:** Em cada região haverá um administrador responsável por administrar todos os pedidos de uma cidade e baixá-los em PDF. Além disso, adicionar, remover cargos de agricultor. O administrador não pode realizar nenhum pedido.
- **Produto:** O produto principal do e-commerce são os alimentos dos agricultores. Estes alimentos devem conter tais informações: nome do produto, foto do produto, preço do produto, descrição contendo a unidade de medida.
 - Os produtos já estão cadastrados no site. Cada agricultor deve selecionar e adicionar os produtos do seu interesse em seu perfil;
 - Produtos novos são cadastrados na plataforma somente através do suporte.
- **Compra:** Um cliente pode realizar compra de diversos produtos de vários agricultores em um único pedido. Dessa forma, cada informação deve ser repassada para o respectivo vendedor.
 - O cliente só poderá comprar produtos de anúncios realizados em sua cidade. Com exceção de Tucuruí e Breu Branco, que podem realizar compras intermunicipais, entretanto, será permitido ter anúncios de uma cidade por vez.
- **Pagamento:** O pagamento será realizado para o entregador, após o cliente receber o produto.
 - A plataforma não deve ter sistema de pagamento incluso. O pagamento deve ser feito de forma presencial.
 - As formas de pagamento são: Pix, Débito, Crédito e Dinheiro.
- **Entrega:** Cada cidade terá um dia específico para realizar todas as entregas das vendas realizadas na última semana.
- **Taxa de entrega:** A compra é realizada em forma de delivery, onde cada cidade possui seu valor/taxa de entrega.

- Belém: R\$ 5,00
- Cametá: R\$ 5,00
- Breu-Branco: R\$ 5,00
- Tucuruí: R\$ 5,00
- Tucuruí/Breu-Branco: R\$ 10,00

Com base nas informações preliminares dos requisitos, foi necessário avançar para a próxima etapa, denominada Estudos, em que foi necessário realizar diversas buscas sobre o que era necessário para se criar um site estilo *e-commerce*. Em seguida, foi entendido que um site é composto, de maneira geral, por: *Front-end*, *Back-end* e Banco de Dados. Até então, a familiaridade com essas áreas era limitada.

4.2 ESTUDOS

No momento após a definição dos requisitos do sistema, ainda não tinha sido possível cursar nenhuma disciplina que abordasse os temas de desenvolvimento de sites. Contudo, conhecimentos de lógica de programação e programação orientada a objetos guiaram esta trajetória. Diante dessa lacuna, foram buscados materiais *online*, como apostilas, vídeos no *Youtube*¹⁴, e investimento em cursos.

Destaca-se especialmente o investimento na plataforma Alura¹⁵, uma renomada plataforma de cursos online voltados para tecnologia. Através dela, foi possível estudar e aplicar os conhecimentos adquiridos.

De maneira geral, os estudos se concentraram na plataforma da Alura, onde foi possível iniciar com os estudos de *Front-end*, posteriormente ao de *Back-end*, utilizando a linguagem PHP¹⁶, e, por último, foi explorado os fundamentos de um banco de dados relacional — MySQL¹⁷.

No *Front-end*, os tópicos estudados incluíram a estrutura de uma página web por meio de tags HTML, a estilização da página com CSS, bem como o estudo do recurso *Flexbox* para o posicionamento de elementos na página, incluindo a

¹⁴ Disponível em: <https://www.youtube.com/>. Acesso em: jan. 2024.

¹⁵ Disponível em: <https://www.alura.com.br/>. Acesso em: jan. 2024.

¹⁶ Disponível em: <https://www.php.net/>. Acesso em: jan. 2024.

¹⁷ Disponível em: <https://www.mysql.com/>. Acesso em: jan. 2024.

responsividade da interface. Além disso, o estudo de Javascript foi acrescentado para auxiliar nas ações na página web.

No *Back-end*, direcionou-se os estudos para temas como o protocolo HTTP, que é essencial para entender a comunicação entre cliente e servidor na web; arrays, para permitir a manipulação e armazenamento de múltiplos dados durante a execução do programa; programação orientada a objetos, para auxiliar na construção de um software na representação do mundo real em objetos; e o uso do PHP na web, abrangendo desde a manipulação de requisições e respostas até a interação com o banco dados para fornecer dados dinâmicos aos clientes.

Quanto ao Banco de Dados, foram explorados conceitos como: Modelagem Conceitual, para representar de forma abstrata as regras de negócio do *e-commerce*; Modelagem Entidade Relacionamento, para moldar e definir como os dados do sistema estão relacionados, usando entidades para representar objetos do mundo real; e SQL, como linguagem para escrever consultas e operações de persistências de dados no banco de dados.

É importante dizer que este processo de estudo, sempre foi guiado pelo objetivo central: o desenvolvimento da Plataforma da Feira da Agricultura Familiar. Com esse embasamento, pode-se agora descrever a próxima etapa, voltada ao desenvolvimento efetivo do projeto.

4.3 DESENVOLVIMENTO

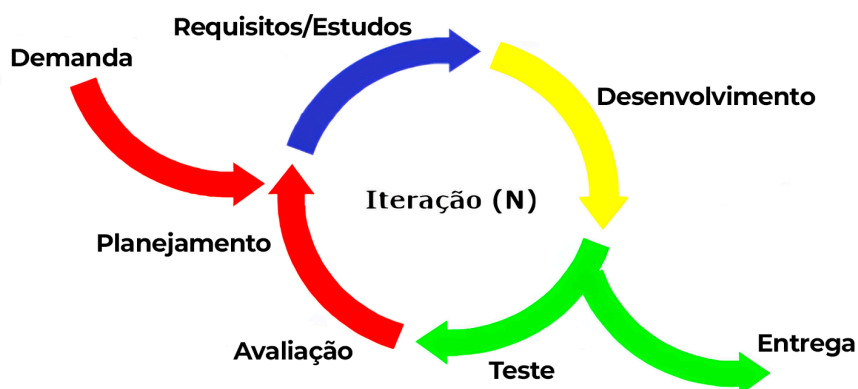
O desenvolvimento da aplicação foi planejado a partir das demandas vindas do projeto de extensão da UFPA. Para atender a essas demandas, procedeu-se a análises e estudos a fim de desenvolver as funcionalidades solicitadas. Em seguida, a avaliação das entregas ocorria por meio de reuniões síncronas com os responsáveis pelo projeto, e com base nas análises dessas interações, ajustes e refinamentos no planejamento eram realizados.

Portanto, pode-se caracterizar que o processo de desenvolvimento deste software adotou uma abordagem iterativa e incremental para atender de forma eficiente as demandas do projeto. Conforme enfatizado por Sommerville (2011), este modelo destaca-se por ser uma estratégia eficaz em projetos que demandam a gestão de mudanças e incertezas, promovendo ciclos de desenvolvimento que

incorporam feedback contínuo e entregas mais alinhadas com as necessidades do cliente.

A Figura 5 oferece uma visão gráfica do processo de desenvolvimento de software anteriormente descrito, estabelecendo uma conexão direta com a abordagem iterativa e Incremental da Engenharia de Software:

Figura 5 - Abordagem Iterativa e Incremental adotada nesta fase do processo.



Fonte: Imagem elaborada pelo autor.

As fases do modelo iterativo e incremental adotado no desenvolvimento, conforme ilustrado na Figura 5, podem ser descritas da seguinte maneira:

Demanda: Cada ciclo desse processo iterativo e incremental tem início com uma demanda específica, a qual foi subdividida em pequenas *features*¹⁸ a partir da demanda inicial registrada no *Google Docs*, conforme discutido na Seção 4.1.1. Essas *features* foram categorizadas em: *front-end do blog*; *front-end da feira online*; *front-end* e *back-end* do painel do agricultor; *front-end* e *back-end* das telas de compras; *front-end* e *back-end* do usuário administrador da região; e desenvolvimento responsivo.

Planejamento: Na etapa de planejamento, foi projetado o escopo de estudo com base na demanda de *feature* atual. Em seguida, foi estabelecido um prazo inicial de dois meses para a conclusão de cada ciclo após o aceite da avaliação. No caso de não cumprimento desse prazo, foi prevista uma extensão de mais um mês para realizar ajustes necessários.

Requisitos/Estudos: Com base nas informações de demanda e na definição do escopo do estudo, realizou-se uma pesquisa primária do tipo qualitativa, no qual

¹⁸ Funcionalidades ou características específicas de um software.

foram realizados diálogos com agricultores de Breu Branco e Tucuruí, além de interações com representantes de Cametá e Belém. O objetivo dessas conversas era compreender o ecossistema da feira da agricultura familiar, desde o processo de plantação até os desafios enfrentados na comercialização dos produtos. Mais adiante, buscou-se capacitação técnica para atender à demanda atual, por meio de pesquisas secundárias, ou seja, o consumo de recursos disponíveis no *YouTube*, cursos na plataforma Alura, revisão de artigos e trabalhos acadêmicos.

Desenvolvimento: Na fase de desenvolvimento, a ênfase foi direcionada para a construção da aplicação, pois o objetivo era atender aos requisitos da funcionalidade em desenvolvimento do ciclo atual. Neste processo, adotou-se a metodologia conhecida como *Pair Programming*¹⁹, em parceria com outro estudante da graduação de Engenharia de Computação, que juntou-se à equipe com o projeto em andamento. O propósito desse método consiste na execução colaborativa da programação, sujeita à revisão conjunta, com o intuito de minimizar as possibilidades de ocorrência de falhas.

Entrega: A finalização do desenvolvimento de uma *feature*, caracteriza-se em uma entrega. Para armazenar os resultados da aplicação, preservar o histórico de mudanças e promover a facilidade de contribuição entre os participantes do time de desenvolvimento, foi utilizado o Github. Essa plataforma é reconhecida como uma hospedagem de código-fonte e um ambiente colaborativo para o desenvolvimento de software. Deste modo, o código-fonte da aplicação encontra-se mantido em um repositório privado, acessível apenas a pessoas autorizadas para contribuir e visualizar o conteúdo. Por meio dessa metodologia foi possível alcançar a primeira versão da aplicação.

Testes: Durante a fase de teste, foram conduzidas ações manuais pela equipe para verificar se os resultados correspondiam às expectativas. Nesse estágio, a decisão foi tomada de não implementar testes automatizados, em virtude das limitações técnicas vigentes e dos prazos de entrega curtos.

Avaliação: Na fase de avaliação, era realizada uma reunião chamada de *Sprint Review*, destinada à análise do que havia sido projetado durante o ciclo atual. Esta reunião era conduzida de forma remota no *Google Meet*²⁰, e contava com a

¹⁹ É uma prática de programação em que dois desenvolvedores trabalham juntos em um único computador, colaborando na criação de código.

²⁰ Disponível em: <https://meet.google.com/>. Acesso em: mar. 2024

participação do: *Product Owner*, *Scrum Master* e a equipe de desenvolvimento. A avaliação das funcionalidades foi feita de forma manual, por meio do compartilhamento de tela. Caso todas as verificações estivessem corretas, um novo ciclo era iniciado; do contrário, a equipe retornava ao estágio de planejamento, com um prazo de um mês para conclusão das correções necessárias.

4.4 ESTUDO SOBRE BOAS PRÁTICAS

Após o desenvolvimento da primeira versão do software, a jornada de aprendizado prosseguiu em cursos da Alura, videoaulas no YouTube, leituras em blogs e artigos. Os estudos focaram em temas específicos, como boas práticas de programação, arquitetura de software, camadas de uma aplicação web, gerenciamento de dependências, persistência de dados, entre outros. Em 2023, foi possível cursar a disciplina de Engenharia de Software, que agregou conhecimentos para a área de análise, projeto e implementação do site.

Cada um desses temas foi descoberto e explorado inicialmente por motivos profissionais e acadêmicos, resultando em uma abordagem de consumo de conteúdos não ordenada. Em consequência disso, à medida que cada tema era consumido, era criado um pensamento crítico sobre o que havia sido desenvolvido até então, ou seja, a primeira versão da feira da agricultura familiar.

Por consequência, este período foi marcado pela identificação de diversos pontos de melhoria na aplicação, visando boas práticas de programação, e uma boa interoperabilidade e manutenção do código. Os principais pontos identificados foram descritos na seção a seguir.

4.5 REFATORAÇÃO

Após os estudos sobre boas práticas de programação, foram identificados os seguintes pontos de melhoria:

1. Adicionar um gerenciador de dependências na aplicação;
2. Alterar a biblioteca de conexão e persistência ao banco de dados de MySQLi para PDO (*PHP Data Objects*);
3. Separar as responsabilidades do arquivo de visualização;

4. Adicionar um Middleware para efetuar o serviço de autorização;
5. Atualização do *Front-Controller*
6. Padronizar a aplicação conforme as PSRs (*PHP Standards Recommendations*);
7. Segregar toda a aplicação em camadas;

Esses pontos foram implementados seguindo a sequência de 1 a 7. Para gerenciar o processo de refatoração e garantir um histórico claro das mudanças, foi utilizado o Git²¹ e GitHub²² como sistema de controle de versão e repositório de código-fonte. Cada ponto de melhoria foi tratado como uma tarefa separada, permitindo uma abordagem iterativa e controlada no processo de refatoração.

Um detalhe importante a mencionar, é que o processo de refatoração pode ocasionar em quebras no sistema, por isso, após a conclusão de cada tarefa de refatoração, foram conduzidos testes manuais para certificar que o sistema continuasse a desempenhar suas funcionalidades, conforme a versão original. Neste período não tinha sido estudado testes automatizados, por isso os testes manuais foram usados como recurso primário. Cada um dos pontos citados foram implementados e descritos na Seção 4.3.

4.6 DOCUMENTAÇÃO

A documentação da aplicação é um recurso indispensável para o entendimento de uma aplicação, desde os requisitos estabelecidos até o sequenciamento de funções que são desencadeadas para realizar uma operação do sistema. Além disso, serve como recurso visual e gráfico para novos desenvolvedores da aplicação, pois auxiliam na compreensão de cada funcionalidade.

Por esses motivos, foram projetados os seguintes recursos da documentação de software:

1. Diagrama de Casos de Uso
2. Especificação de Casos de Uso

²¹ Disponível em: <https://git-scm.com/> . Acesso em: mar. 2024

²² Disponível em: <https://github.com/> . Acesso em: mar. 2024

3. Diagramas de Sequência
4. Diagrama de Classes
5. Diagrama de Modelagem Conceitual
6. Diagrama de Modelagem Física

Para construir o diagrama de casos de uso, diagramas de sequência e diagrama de classes foi utilizado o software *Lucidchart*²³. A modelagem conceitual do banco de dados foi realizada utilizando o software *BrModelo*²⁴. Quanto à modelagem física do banco de dados, foi utilizado um recurso do MySQL Workbench²⁵ denominado "MySQL Workbench Modeling". Já os requisitos funcionais, não funcionais e as especificações de casos de uso foram documentados no *Google Docs*.

Cada um desses recursos da documentação, são descritos e ilustrados ao longo da seção de "Desenvolvimento".

4.7 TESTES DE USABILIDADE

Os testes de usabilidade são essenciais para avaliar a interface do ponto de vista de quem irá utilizar a plataforma, para que seja possível coletar opiniões da experiência do uso de uma ferramenta.

Para testar a usabilidade da plataforma, foi adotada uma metodologia composta por cinco etapas, conforme descrito por Pressman (2016). Primeiramente, definiu-se as categorias de teste e seus respectivos objetivos. Em seguida, foram criados cenários que possibilitasse uma avaliação abrangente da usabilidade de acordo com os casos de uso da aplicação. Depois foram selecionados candidatos para participar do teste. Durante o teste, o avaliador deve orientar os participantes e coletar dados relevantes sobre a experiência do usuário. Por fim, deve-se utilizar métricas para avaliar a usabilidade do produto.

Os testes de usabilidades escolhidos foram: (a) moderado, pois o avaliador (autor deste trabalho) guiou o teste orientando o entrevistado sobre os objetivos que ele deveria fazer. Ao mesmo tempo realizando perguntas com o propósito de ter

²³ Disponível em: <https://www.lucidchart.com/pages/> . Acesso em: mar. 2024

²⁴ Disponível em: <http://www.sis4.com/brModelo/> . Acesso em: mar. 2024

²⁵ Disponível em: <https://www.mysql.com/products/workbench/> . Acesso em: mar. 2024

uma visão sobre a experiência do usuário. (b) Presencial, pois o avaliador se dirigiu à casa de cada um dos participantes. (c) Avaliativo, pois ao final do teste o avaliador métricas para medir a satisfação do usuário com o produto.

Os cenários criados foram baseados nos casos de uso mais comuns do *e-commerce*: (a) acesso à plataforma para analisar o conteúdo e cadastrar-se, (b) realizar a compra de produtos e (c) criar anúncios para vender produtos.

Sobre a seleção dos participantes para realizar os testes, foram escolhidas 2 pessoas que representam o público alvo do produto: (a) cliente e (b) agricultor. Mais especificamente um morador de Tucuruí e um Agricultor Familiar da cidade de Breu Branco - PA.

Durante a execução do teste, o avaliador observou as ações dos usuários e fez perguntas direcionadas para entender suas percepções, tais como a adequação do tamanho da fonte, a visibilidade dos botões e a presença de dúvidas. Todas as informações foram registradas em uma folha de papel A4. Por fim, foi aplicado um formulário para medir a satisfação do cliente com a plataforma, este assunto será discutido na seção a seguir.

4.8 AVALIAÇÃO DOS RESULTADOS

Para complementar a avaliação de usabilidade, foi aplicado o *System Usability Scale* (SUS), conforme proposto por Brooke (1996). Este método oferece uma métrica objetiva para avaliar a usabilidade percebida pelos usuários, permitindo a quantificação dos resultados obtidos durante os testes por meio de um questionário composto por 10 itens com 5 opções de resposta, seguindo a escala Likert que varia de 1 (Discordo totalmente) a 5 (Concordo totalmente).

A Tabela 7 apresenta o questionário SUS, inicialmente concebido em inglês, porém, para viabilizar a realização do teste, foi traduzido para o idioma português.

Tabela 7 - Questionário SUS aplicado no teste de usabilidade

Questionário SUS	Discordo Totalmente		Concordo totalmente		
	1	2	3	4	5
Eu gostaria de usar este sistema com frequência.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Eu achei o sistema desnecessariamente complexo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema fácil de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar este sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei as várias funções do sistema bem integradas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema muito inconsistente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei que a maioria das pessoas aprenderia a usar este sistema rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema muito complicado.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu me senti confiante usando o sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu precisei aprender muitas coisas antes de começar a usar o sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: Tabela adaptada pelo autor.

O cálculo do resultado do SUS é derivado da contribuição individual de cada item. Onde, aos itens ímpares é aplicada uma subtração de 1 à resposta do usuário, enquanto nos itens pares, é aplicada uma subtração de 5 na resposta do usuário. Após obter a pontuação de cada item, estes são somados e o resultado é multiplicado por 2,5, conforme sugerido por Broke (1996). Deste modo, o resultado final representa um índice de satisfação do usuário, variando de 0 a 100.

5 DESENVOLVIMENTO

Essa seção é destinada a mostrar detalhes do desenvolvimento do software, desde a escolha da identidade visual até o produto final.

5.1 IDENTIDADE VISUAL

A primeira etapa desse desenvolvimento foi criar a identidade visual, cujo é primordial para o sucesso em qualquer negócio. Segundo Strunck (2001), a identidade visual é o conjunto de elementos gráficos que vão enfatizar a personalidade visual de um nome, ideia ou serviço, ou seja, ela precisa transmitir uma boa comunicação entre o produto e o usuário final.

Atualmente, se uma empresa ou negócio não transmite uma boa imagem, não causa boa impressão à primeira vista, isso pode refletir em sua receita (STRUNCK, 2001).

A cor é um outro fator preponderante, seguindo esta perspectiva, pois transmite sentimentos, causa desejos e realiza associações. De acordo com Teixeira, Silva e Bona (2012), experimentos científicos comprovam que a sensibilidade humana tem uma alteração quando exposta às mais variadas gamas de cores.

Sabendo que este projeto se trata de um *e-commerce* destinado a Agricultura Familiar, foi pensado nos atributos de marca com o propósito de transmitir os seguintes sentimentos nesta identidade visual: agricultura, plantação, família, felicidade, modernidade e amor.

Para alcançar esses objetivos, optou-se pelas cores verde e amarelo. O verde, sendo a quintessência da natureza, comunica um estilo de vida voltado para a consciência ambiental e um profundo amor pela natureza. Por sua vez, o amarelo simboliza a felicidade e faz alusão ao brilho do sol (Heller, 2022).

Para garantir a consistência e fidelidade das cores em diferentes tipos de materiais, sejam eles de mídia física ou digital, a escolha baseou-se no padrão Pantone²⁶. Este padrão, reconhecido internacionalmente, oferece uma linguagem

²⁶ Disponível em: <https://www.pantone.com/>. Acesso em: mar. 2024

universal de cores, garantindo uma representação uniforme em diversos contextos. A Figura 6 ilustra as cores selecionadas conforme esse padrão.

Figura 6 - Cores Pantone utilizadas na identidade visual da Feira da Agricultura Familiar



Fonte: Pantone

Posteriormente à definição das cores, foi pensado em um elemento muito importante em uma identidade visual — logotipo, que funciona como uma assinatura, no uniforme, papelaria e embalagens, tudo isso para que exista uma associação com o negócio. (VÁSQUEZ, 2007)

O logotipo deste projeto foi formado pela composição simbólica de duas folhas completadas por um coração posicionado acima. As duas folhas representam a base deste negócio, fazendo referência à agricultura. A duplicidade destas folhas denota a colaboração e sinergia que é essencial para o sucesso. O coração situado no topo de cor amarela, faz alusão ao sol e à fotossíntese, cujo é a fonte de energia do ecossistema agrícola. O resultado visual dessa simbologia é apresentado na Figura 7.

Figura 7 - Simbologia de duas folhas com um coração que formam o logotipo da Feira da Agricultura Familiar



Fonte: Imagem elaborada pelo autor.

Para a criação deste logotipo, foi utilizado o *Adobe Illustrator*²⁷. As Figuras 8 a 10 mostram algumas variações desse logotipo:

Figura 8 - Logotipo formato 1, formato circular.



Fonte: Imagem elaborada pelo autor.

Figura 9 - Logotipo formato 2, formato horizontal.



Fonte: Imagem elaborada pelo autor.

Figura 10 - Logotipo formato 3, formato de ícone.



Fonte: Imagem elaborada pelo autor.

A etapa subsequente à criação da identidade visual é a elaboração das interfaces, um tópico que será abordado na sequência.

²⁷ Disponível em: <https://www.adobe.com/br/products/illustrator.html>. Acesso em: mar. 2024

5.2 FRONT-END

A partir da identidade visual criada, partiu-se para a criação das telas da aplicação, comumente conhecido como o *Front-end*.

O desenvolvimento do Front-end foi dividido em três fases:

1. **Desenvolvimento do blog**
2. **Desenvolvimento da feira online**
3. **Desenvolvimento responsivo**

Em nenhuma dessas fases foi criado algum tipo de protótipo para o projeto, seja ele de baixo, médio ou alta fidelidade. O método usado foi o de desenvolvimento sem prototipagem, que implica na implementação direta das interfaces de usuário, sem uma representação de algum protótipo visual preliminar.

No desenvolvimento das telas, foram utilizadas referências de interfaces de projetos em geral encontrados através de buscas no *Google Imagens*²⁸ e *Behance*²⁹. A partir disso, foi usada a linguagem HTML para construir a estrutura da página, em seguida o CSS, para estilizar a página, e por fim o Javascript para incrementar interatividade e dinamismo nas páginas.

O desenvolvimento de cada tela foi planejada para atender ao formato *desktop*, adotando a resolução de 1366 pixels x 768 pixels. No contexto do formato *mobile*, as interfaces foram projetadas considerando a resolução de 360 pixels x 640 pixels. Essa decisão foi baseada em uma pesquisa conduzida por Avellare (2020), indicando que tais resoluções são as mais frequentemente utilizadas pelos usuários.

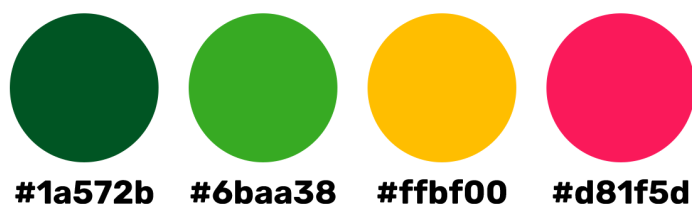
No entanto, embora as interfaces tenham sido desenvolvidas para essas resoluções específicas, sua adaptabilidade não se restringe a esses parâmetros. O projeto é flexível a diferentes tipos de resoluções, mas é crucial reconhecer que o uso de resoluções diversas, podem impactar a apresentação visual da interface.

No que diz respeito às cores adotadas na interface da plataforma, foram realizados ajustes com o intuito de realçar e proporcionar um maior contraste às informações apresentadas. A Figura 11 apresenta a paleta de cores escolhida, acompanhada dos respectivos códigos hexadecimais de cada tonalidade.

²⁸ Disponível em: <https://images.google.com/>. Acesso em abril. 2024

²⁹ Disponível em: <https://www.behance.net/>. Acesso em abril. 2024

Figura 11 - Paleta de cores do e-commerce Feira da Agricultura Familiar com seus respectivos códigos hexadecimais



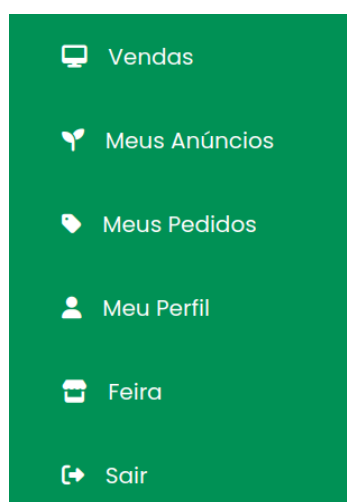
Fonte: Imagem elaborada pelo autor.

A tonalidade mais escura de verde (#1a572b) foi utilizada em grande parte da plataforma, principalmente nas barras de menus. O verde mais claro (#6baa38) foi selecionado para realçar e diferenciar determinadas informações, como mensagens de sucesso. A cor amarela (#ffbf00) foi aplicada para destacar informações importantes e servir como suporte para mensagens de alerta. Por fim, o vermelho (#d81f5d) foi usado como auxílio para mensagens de erro.

A seleção das cores foi realizada por meio de ajustes manuais, buscando alcançar tonalidades visualmente agradáveis. Para auxiliar nesse processo, recorreu-se à plataforma *Adobe Color*³⁰.

Durante a navegação pela plataforma é possível notar alguns ícones presentes, para auxiliar na descrição de algumas informações. Conforme mostram as Figuras 12 a 14:

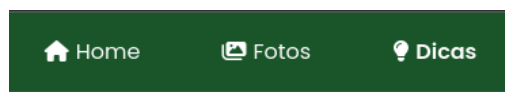
Figura 12 - Barra de menu do agricultor logado na plataforma



Fonte: Imagem elaborada pelo autor.

³⁰ Disponível em: <https://color.adobe.com/create/color-wheel>. Acesso em: mar. 2024

Figura 13 - Barra de menu do blog da plataforma



Fonte: Imagem elaborada pelo autor.

Figura 14 - Barra de menu da feira online da plataforma



Fonte: Imagem elaborada pelo autor.

Os ícones mostrados acima são obtidos por meio da CDN (Content Delivery Network) da plataforma *Font Awesome*³¹, que constitui um conjunto de ferramentas de fontes e ícones fundamentados em CSS.

As imagens utilizadas na plataforma, incluindo banners e fotos de produtos, foram obtidas por meio das plataformas *Pexels*³², *Unsplash*³³ e *Pixabay*³⁴. Todas essas plataformas oferecem um acervo de imagens de forma gratuita e livre de direitos autorais.

Quanto à organização dos arquivos de interface do projeto, cada tela possui um arquivo ao qual está descrita sua estrutura HTML, estilização e regras de negócios. Conforme mostra a Figura 15:

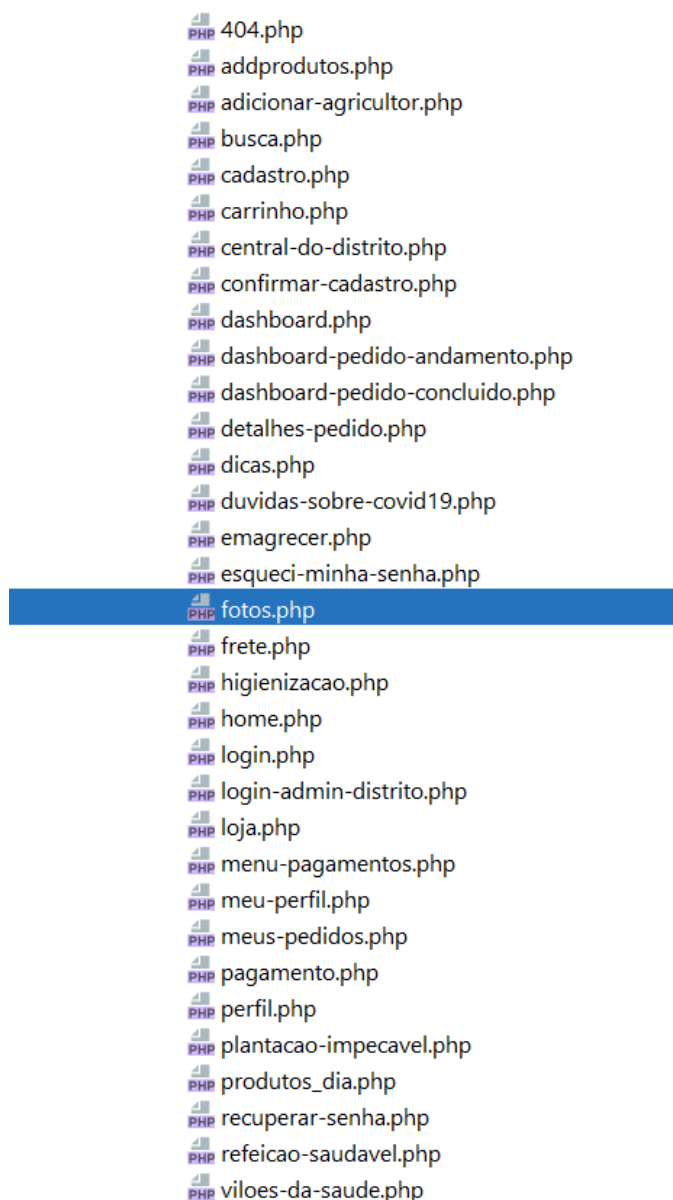
³¹ Disponível em: <https://fontawesome.com/>. Acesso em: mar. 2024

³² Disponível em: <https://www.pexels.com/pt-br/>. Acesso em: mar. 2024

³³ Disponível em: <https://unsplash.com>. Acesso em: mar. 2024

³⁴ Disponível em: <https://pixabay.com/pt/>. Acesso em: mar. 2024

Figura 15 - Arquivos da interface da plataforma



Fonte: Imagem elaborada pelo autor.

Vale ressaltar que, por mais que a extensão dos arquivos seja .php, quando este arquivo é lido pelo navegador, o navegador identifica que foi retornado um texto, então ele reconhece a estrutura daquele texto, e retorna a página estruturada e estilizada.

Na Figura 16, é apresentada a estrutura do arquivo fotos.php, uma página da plataforma destinada a exibir fotografias do projeto Feira da Agricultura Familiar.

Figura 16 - Código-fonte do arquivo fotos.php

```

<link rel="short icon" type = "image/x-icon" href = "public/assets/img/logo.png">
</head>

<body>
  <header class = "cabecalhoPrincipal">
    <div class = "container container-menu">
      <a href = "index" class = "logoLink"></a>
    <div class="menu-section">
      <div class="menu-toggle">
        <div class="one"></div>
        <div class="two"></div>
        <div class="three"></div>
      </div>
    </div>
    <nav>
      <ul class = "lista_barra">
        <li class = "lista_Principal"><a class = "link_lista" href="home">Home</a></li>
        <li class = "lista_Principal"><a class = "link_lista" href="fotos">Fotos</a></li>
        <li class = "lista_Principal"><a class = "link_lista" href="dicas">Dicas</a></li>
        <li class = "lista_Principal">
          <a class = "link_lista" href = "loja">
            <Button class = "botao_Feira" type = "button">Feira Virtual</Button>
          </a>
        </li>
      </ul>
    </nav>
  </div>
</header>

<main class = "conteudoPrincipal">

```

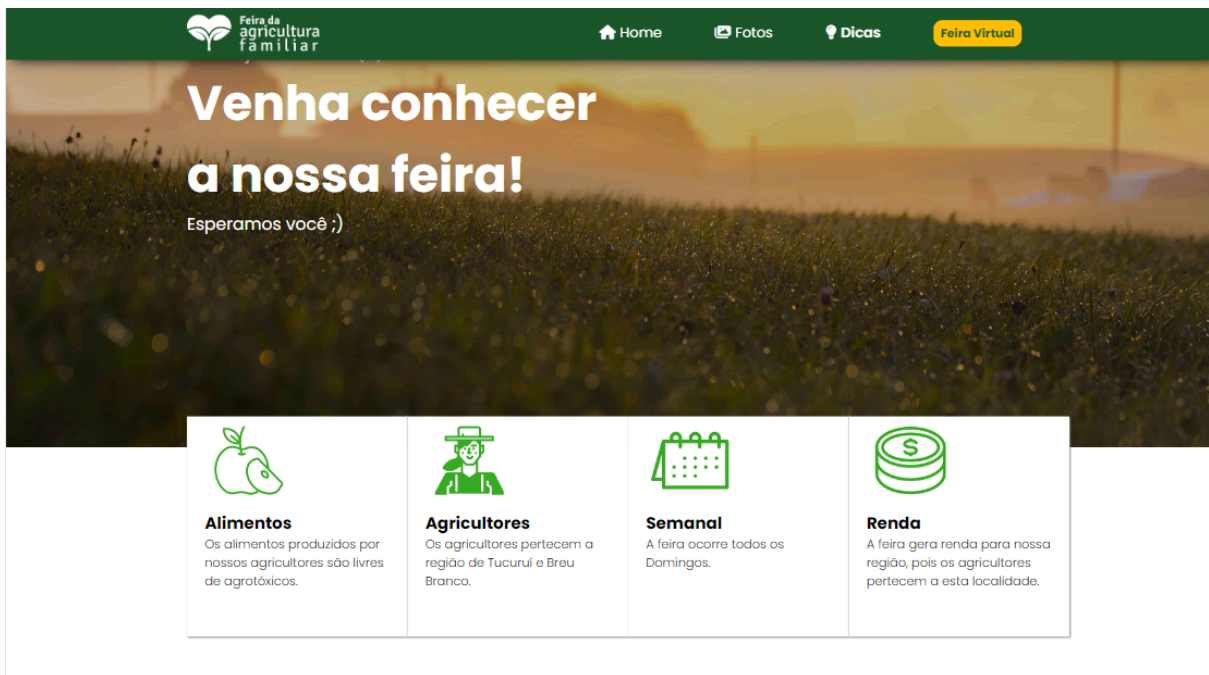
Fonte: Imagem elaborada pelo autor.

5.2.1 Desenvolvimento do blog

O blog da plataforma é o espaço onde existem páginas estáticas, cujo objetivo é apresentar e divulgar o projeto da feira da agricultura através de fotos, artigos informativos, informações e localização das feiras.

As Figuras 17 a 21 mostram o resultado do blog desenvolvido:

Figura 17 - Página inicial do blog com apresentação dos benefícios do projeto



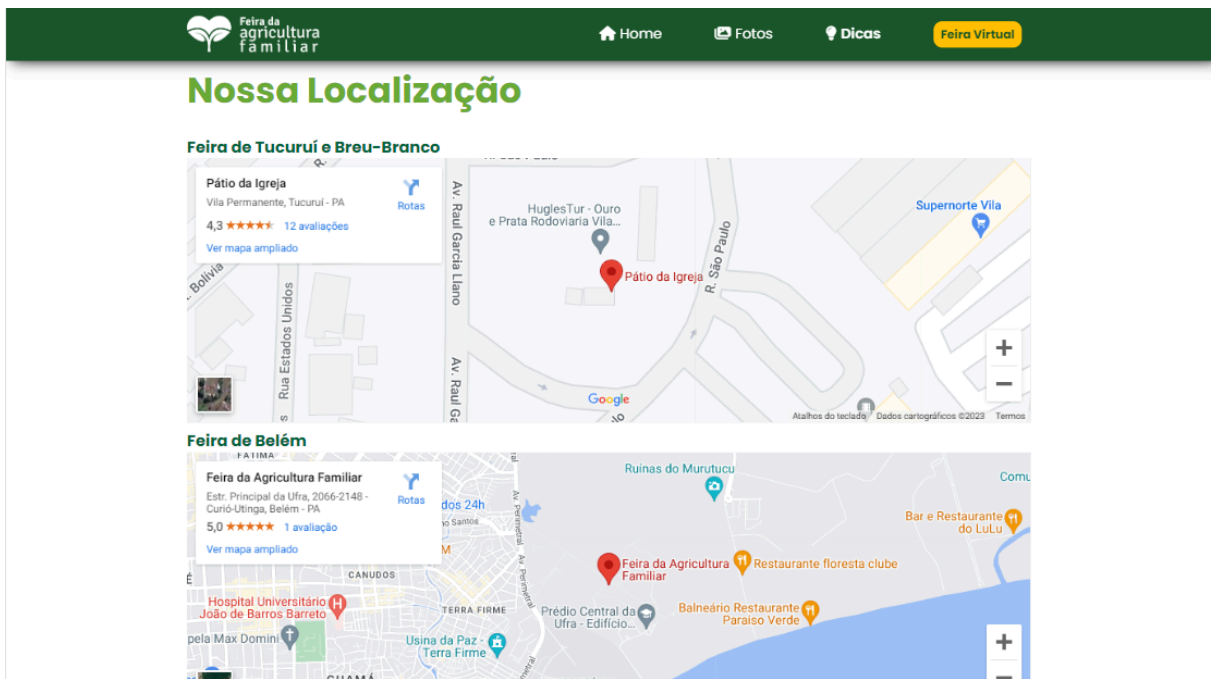
Fonte: Imagem elaborada pelo autor.

Figura 18 - Página inicial do blog com detalhes do projeto



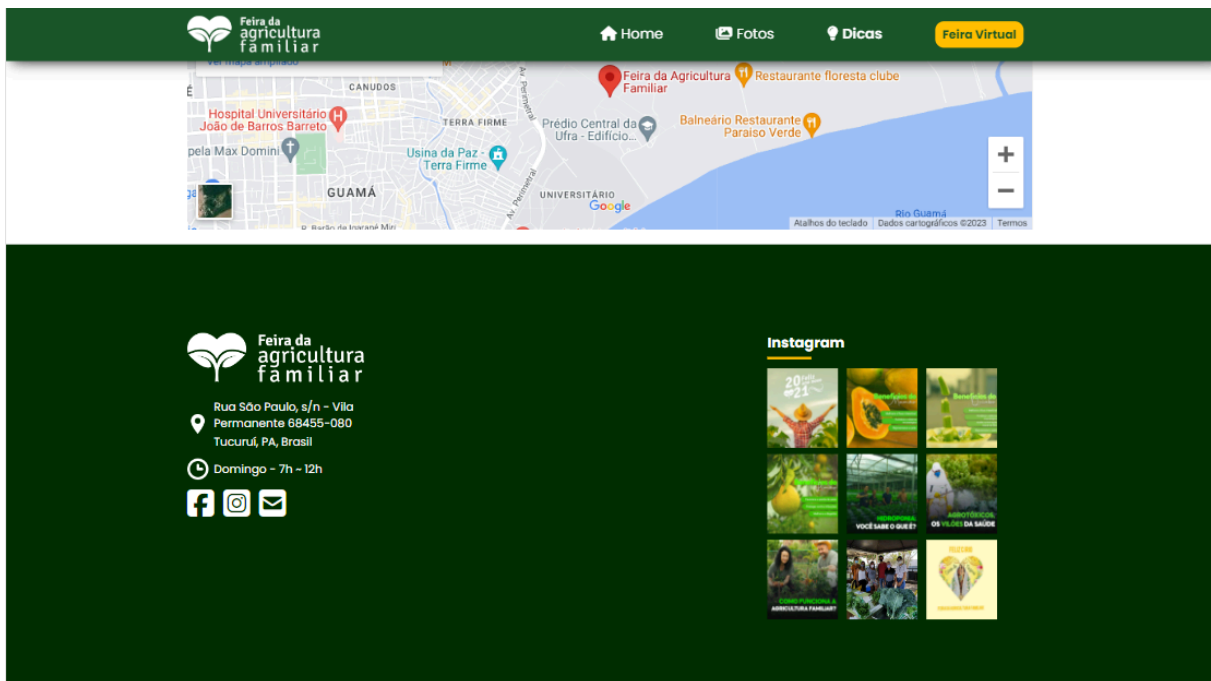
Fonte: Imagem elaborada pelo autor.

Figura 19 - Página inicial do blog com detalhes da localização da feira presencialmente



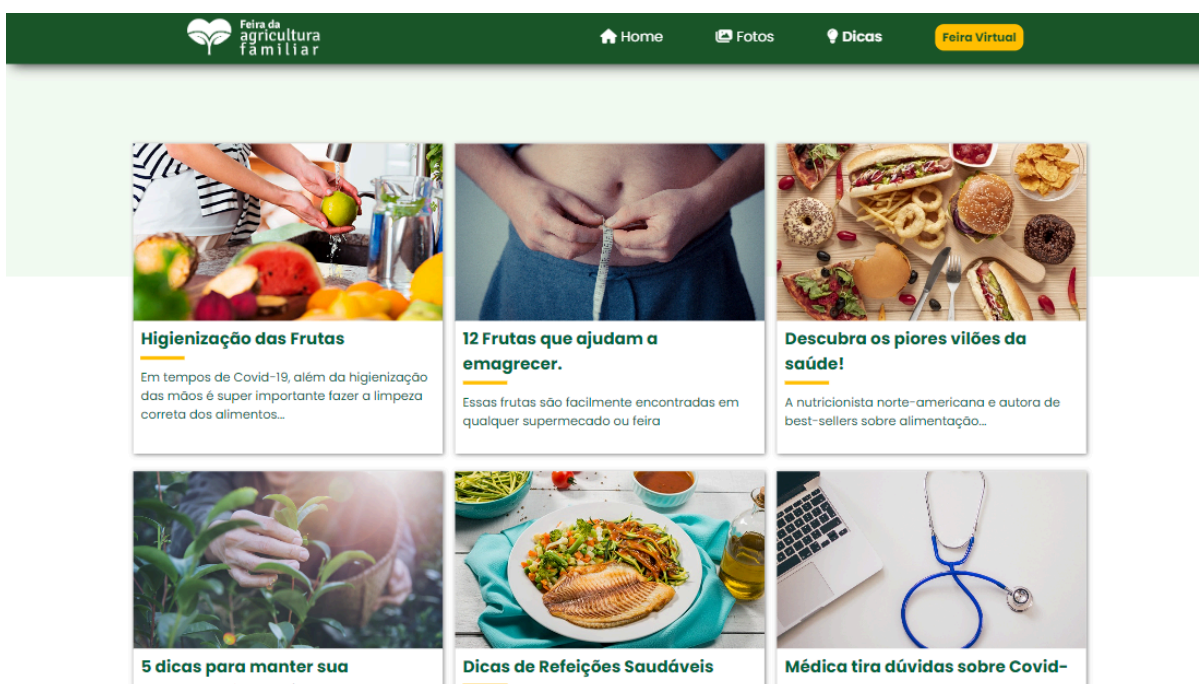
Fonte: Imagem elaborada pelo autor.

Figura 20 - Página inicial do blog com detalhes do rodapé



Fonte: Imagem elaborada pelo autor.

Figura 21 - Página de dicas do blog com lista de artigos



Fonte: Imagem elaborada pelo autor.

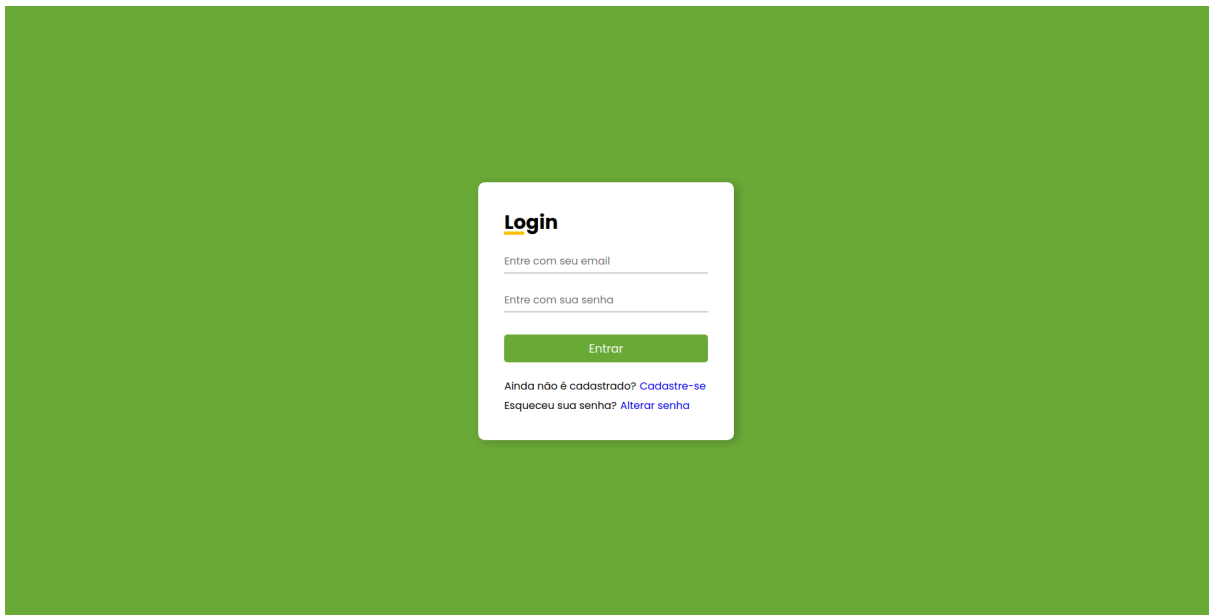
5.2.2 Desenvolvimento da feira online

A feira online da plataforma é o espaço destinado à interação entre agricultores que desejam vender seus produtos e clientes que desejam comprá-los. Como resultado, haverá uma demanda significativa por telas específicas, tais como: login, cadastro, carrinho de compras, informações sobre frete e o perfil do usuário, entre outras funcionalidades correlatas.

É importante destacar que esta etapa do desenvolvimento foi realizada em paralelo com a construção do back-end, visando uma abordagem integrada. No âmbito da Engenharia de Software, tal estratégia alinha-se com princípios fundamentais de desenvolvimento ágil, permitindo uma implementação eficiente e incremental.

As Figuras 22 a 29 mostram o resultado da feira online desenvolvida:

Figura 22 - Página de login

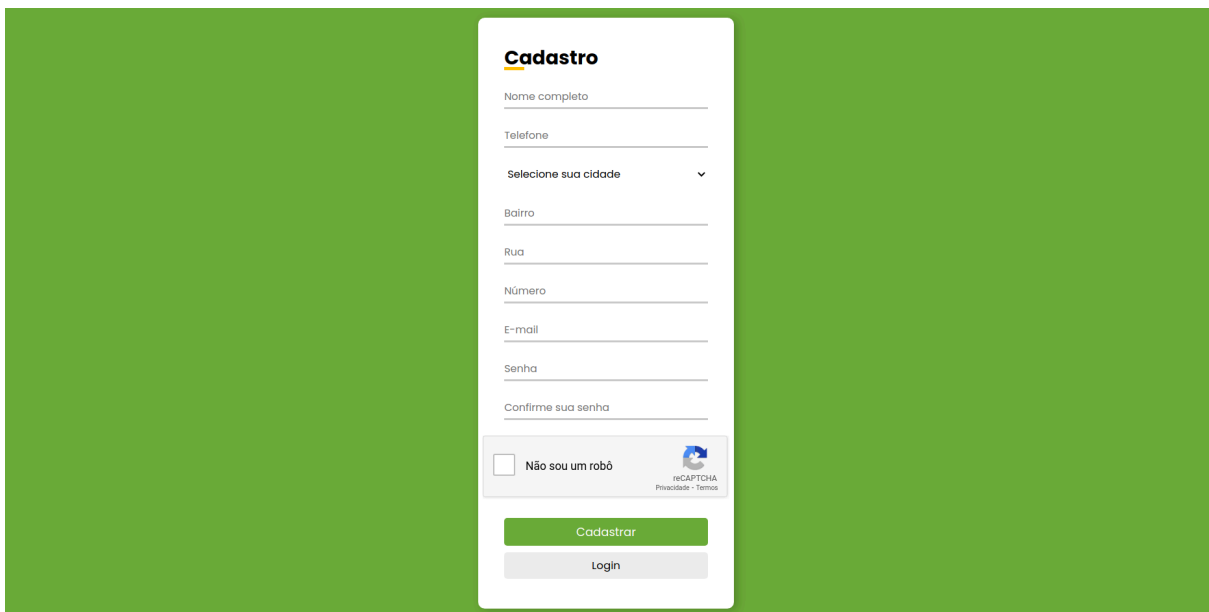


The image shows a login form centered on a solid green background. The form is white with a thin border and contains the following elements:

- Header:** The word "Login" in bold black text, with a small yellow underline under the "i".
- Input Fields:** Two text input fields. The first is labeled "Entre com seu email" and the second is labeled "Entre com sua senha".
- Submit Button:** A green button with the text "Entrar" in white.
- Links:** Below the button, there are two links: "Ainda não é cadastrado? [Cadastre-se](#)" and "Esqueceu sua senha? [Alterar senha](#)".

Fonte: Imagem elaborada pelo autor.

Figura 23 - Página de cadastro

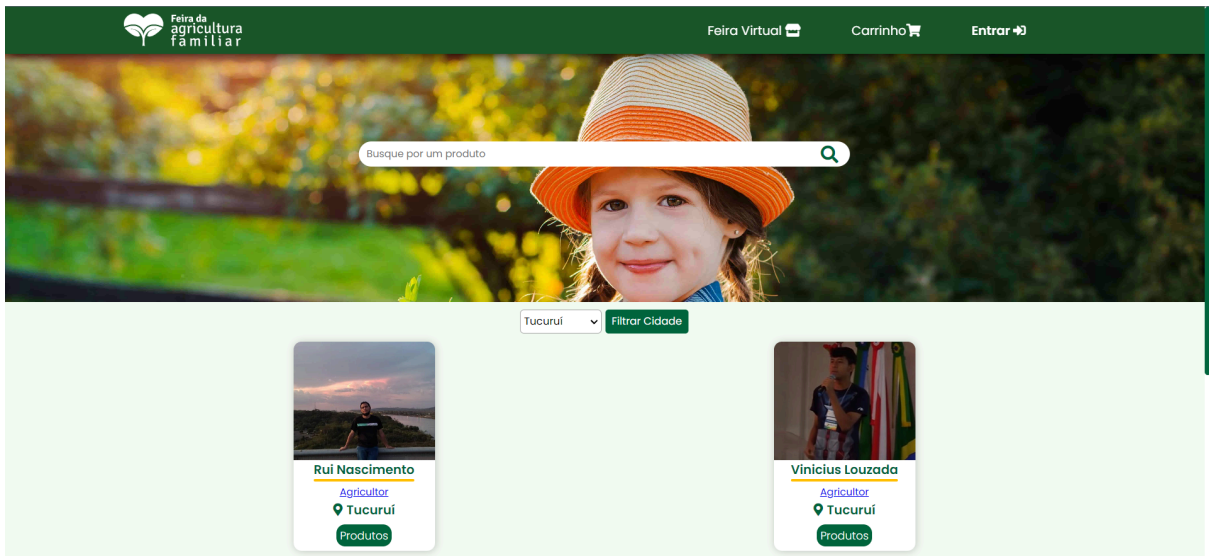


The image shows a registration form centered on a solid green background. The form is white with a thin border and contains the following elements:

- Header:** The word "Cadastro" in bold black text, with a small yellow underline under the "a".
- Input Fields:** A series of text input fields for "Nome completo", "Telefone", "Rua", "Número", "E-mail", "Senha", and "Confirme sua senha".
- Dropdown:** A dropdown menu labeled "Selecione sua cidade" with a downward arrow.
- Form Elements:** A "Bairro" input field and a "Rua" input field.
- Security:** A checkbox labeled "Não sou um robô" next to a reCAPTCHA logo and the text "reCAPTCHA Privacidade - Termos".
- Buttons:** A green "Cadastrar" button and a grey "Login" button.

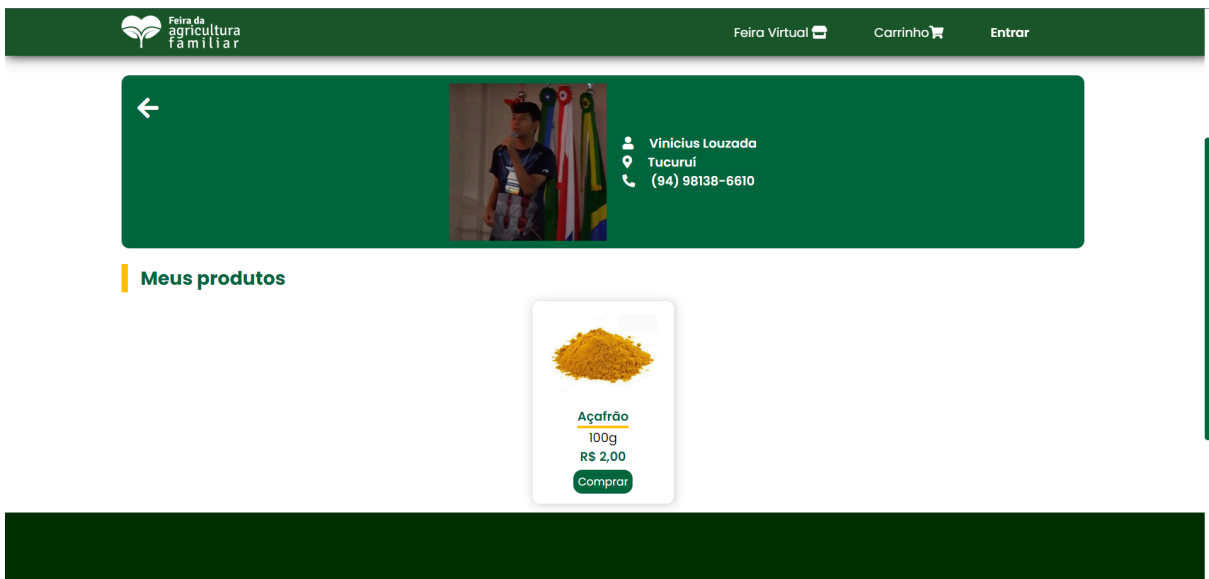
Fonte: Imagem elaborada pelo autor.

Figura 24 - Página inicial da feira online



Fonte: Imagem elaborada pelo autor.

Figura 25 - Página de perfil de agricultor com seus produtos a venda



Fonte: Imagem elaborada pelo autor.

Figura 26 - Página de vendas do agricultor, pedidos em andamento e pedidos finalizados.

The screenshot displays the farmer's sales management interface. On the left is a green sidebar with the user's profile (Vinicius Louzada Valente) and navigation options: Vendas, Meus Anúncios, Meus Pedidos, Meu Perfil, Feira, and Sair. The main content area is divided into two sections:

Pedidos em andamento

Nº Pedido	Data e Hora	Cliente	Descrição do Pedido	Situação	Finalizar Pedido
#8	28/11/2023 08:39:00	Emanuel	Ver Detalhes	Em andamento	Concluir Pedido

Below the table is a **Ver tudo** button.

Pedidos finalizados

Nº Pedido	Data e Hora	Cliente	Descrição do Pedido	Situação
-----------	-------------	---------	---------------------	----------

Below this table is also a **Ver tudo** button.

Fonte: Imagem elaborada pelo autor.

Figura 27 - Página de anúncios que estão ativos pelo agricultor

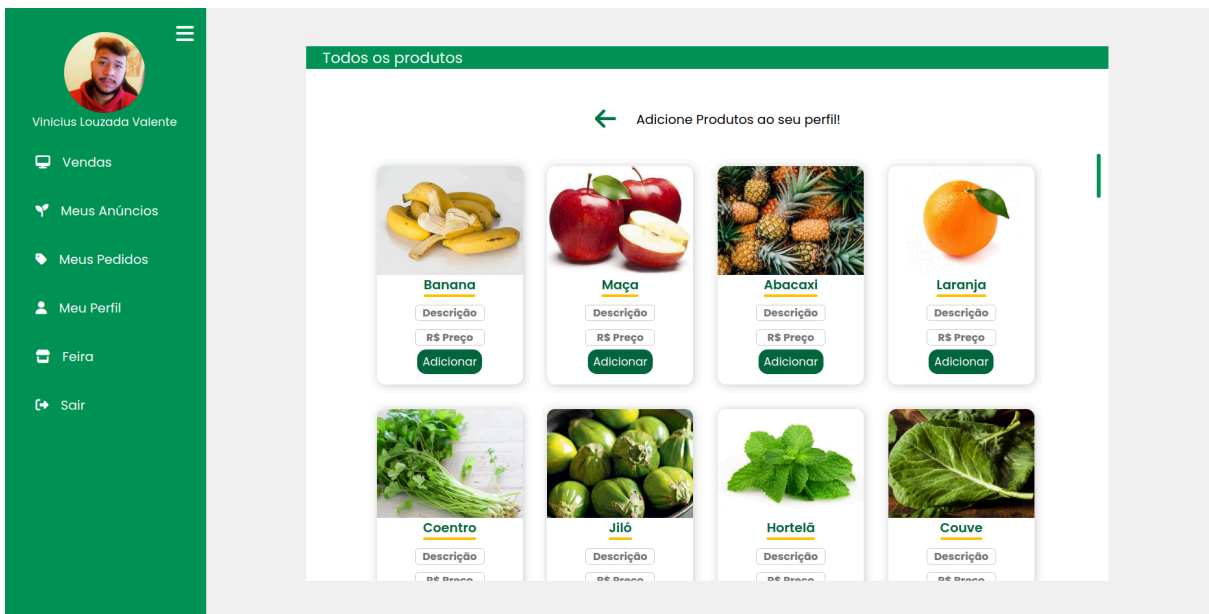
The screenshot shows the farmer's active advertisements page. The sidebar is identical to the previous page. The main content area is titled **Seus produtos** and includes the text "Atualize os produtos que estão disponíveis!" and an **Adicionar mais produtos** button.

Two product cards are displayed:

- Banana**: 1 Dúzia, R\$ 8.60. Buttons: **Editar** (green), **Excluir** (red).
- Laranja**: 5 Unidades, R\$ 4.00. Buttons: **Editar** (green), **Excluir** (red).

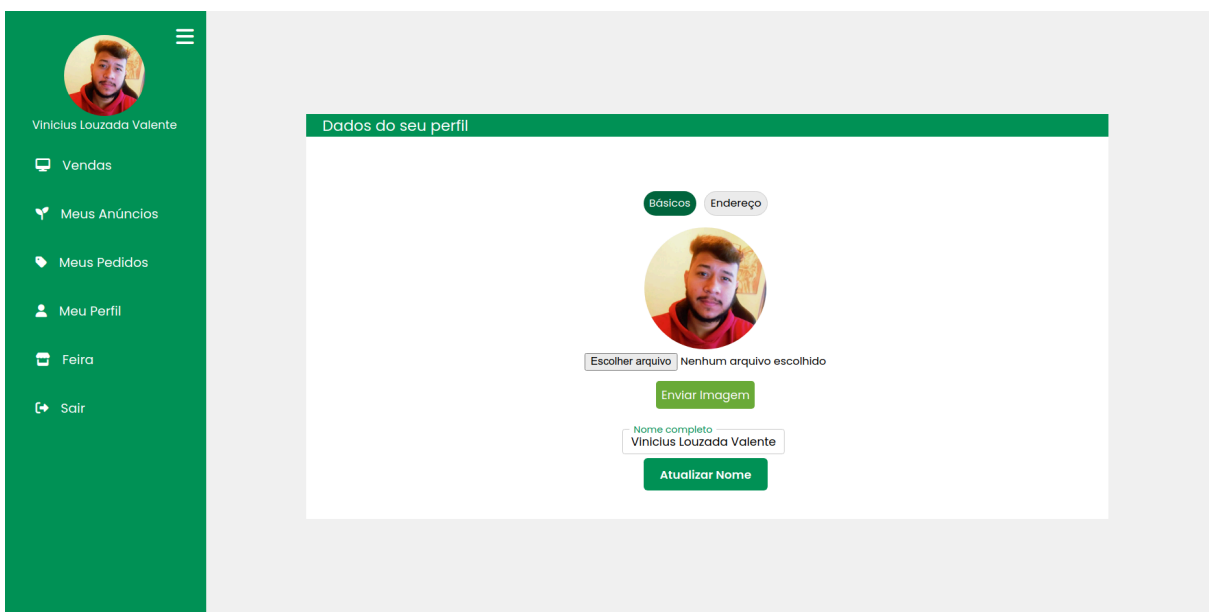
Fonte: Imagem elaborada pelo autor.

Figura 28 - Página de criação de anúncios



Fonte: Imagem elaborada pelo autor.

Figura 29 - Página de perfil de usuário



Fonte: Imagem elaborada pelo autor.

5.2.3 Desenvolvimento responsivo

A diversidade de dispositivos utilizados pelos usuários, como *smartphones*, *tablets* e computadores, gera uma preocupação a mais no projeto, onde a interface desenvolvida deve se adaptar a cada um desses dispositivos para proporcionar uma experiência fluida e otimizada em todas as telas.

Este período do desenvolvimento foi tratado por último, levando em consideração que não foi criado protótipo e que seria mais vantajoso se todas as interfaces para *desktops* estivessem desenvolvidas para usá-las como base neste desenvolvimento de adaptação.

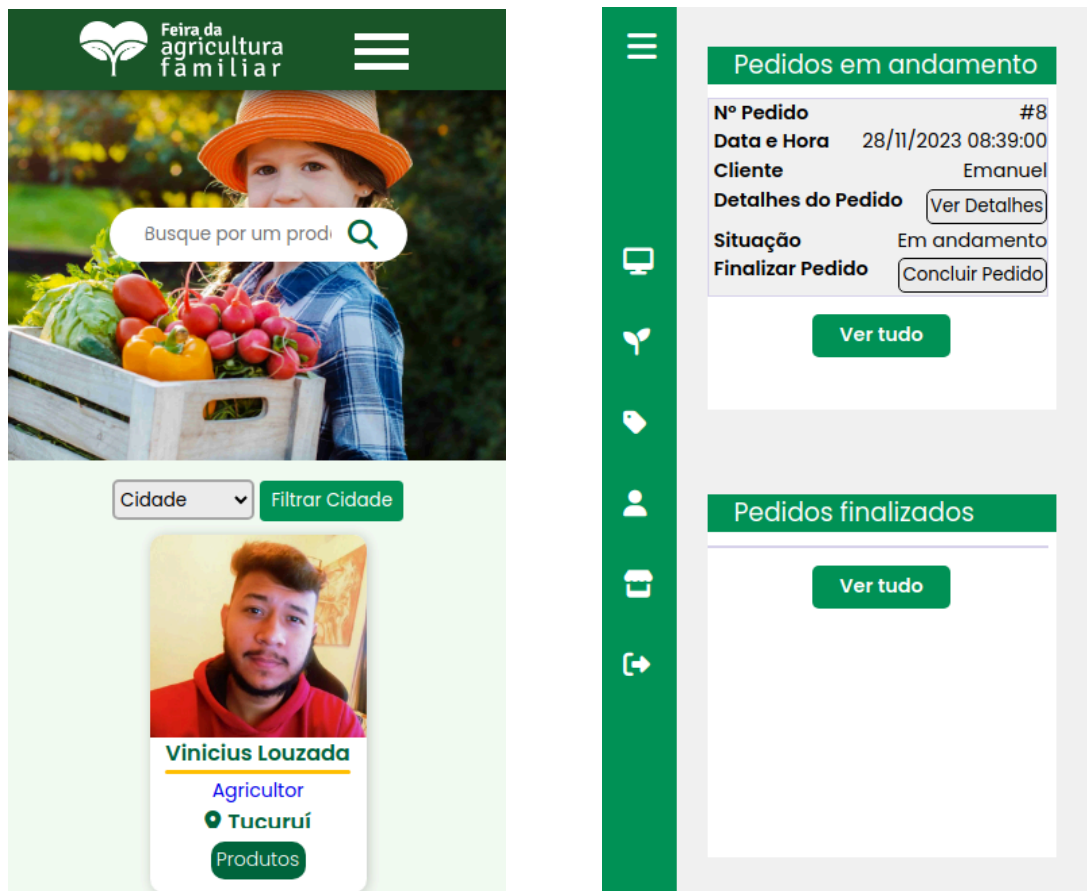
As Figuras 30 a 32 mostram o resultado do desenvolvimento responsivo:

Figura 30 - Página inicial de forma responsiva para dispositivos móveis



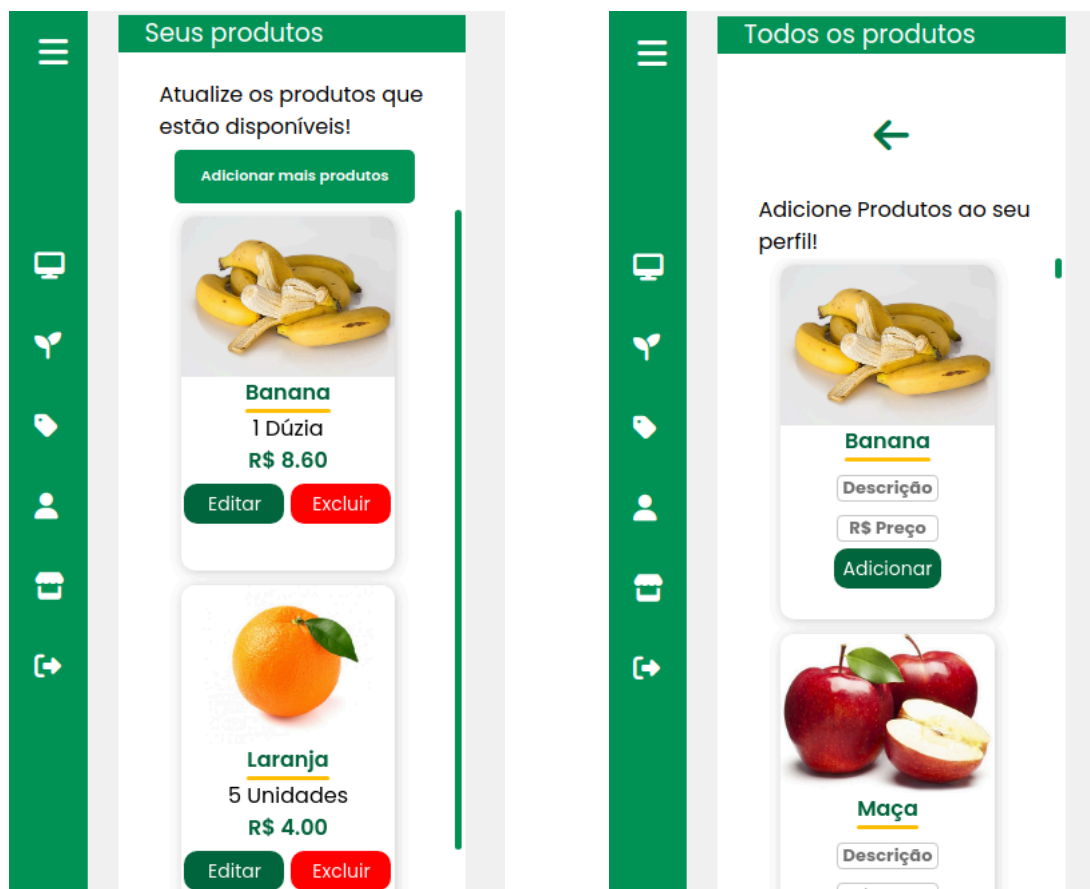
Fonte: Imagem elaborada pelo autor.

Figura 31 - Página da feira online e vendas do agricultor de forma responsiva para dispositivos móveis



Fonte: Imagem elaborada pelo autor.

Figura 32 - Página de anúncios do agricultor e criação de anúncios de forma responsiva para dispositivos móveis



Fonte: Imagem elaborada pelo autor.

5.3 BANCO DE DADOS

Para o armazenamento dos dados do sistema da Feira de Agricultores Familiares, foi escolhido o uso de um banco de dados relacional, neste caso o MySQL.

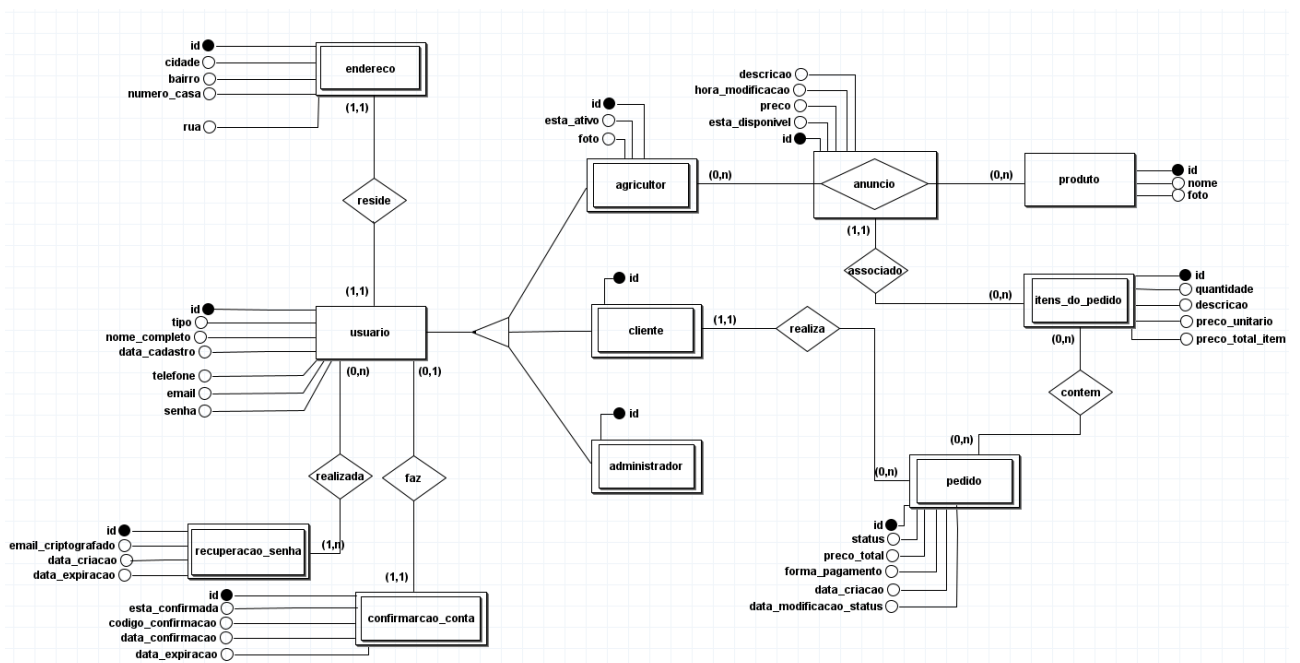
5.3.1 Modelagem Conceitual

Uma fase fundamental no processo de desenvolvimento de software está na comunicação eficaz. Manter uma ótima interação entre o cliente e a equipe de desenvolvimento apresenta-se como um desafio, uma vez que a equipe detém conhecimentos técnicos, e o cliente, em sua maioria, está mais preocupado com a entrega final do produto. Em virtude disso, a modelagem conceitual exerce um papel importante ao facilitar o entendimento entre essas duas vertentes.

A modelagem conceitual facilita a comunicação, tendo em vista que o diagrama impõe uma linguagem que é diretamente alinhada com o contexto do negócio. Fowler (1997) destaca que ao desenvolver um sistema com um propósito específico, é crucial que o diagrama conceitual atenda às necessidades do sistema, incluindo os requisitos estabelecidos pelo cliente, conforme discutido na Seção 4.1.

Para o sistema da feira da agricultura familiar, a problemática central que deve ser solucionada é: agricultores precisam estar relacionados com produtos, e usuários com pedidos de compras. Diante desse desafio, foi utilizado o programa *BRModelo*³⁵ para desenvolver o diagrama mostrado na Figura 33:

Figura 33 - Diagrama da Modelagem Conceitual do sistema Feira da Agricultura Familiar



Fonte: Imagem elaborada pelo autor.

De acordo com a Figura 33, pode-se afirmar que:

- Foi adotado o conceito de generalização e especialização para hierarquia de usuários. Onde a entidade genérica é “usuario”, e as entidades especializadas são: “agricultor”, “cliente” e “administrador”. Essa escolha de modelo foi feita para segregar melhor os diferentes tipos de usuário, sem recorrer a uma única tabela de usuários com várias colunas vazias, dependendo do tipo de usuário.

³⁵ Disponível em: <https://www.brmodeloweb.com/lang/pt-br/index.html>. Acesso em abril. 2024

- Para as entidades de “agricultor” e “anuncio” optou-se por adotar a técnica de exclusão lógica, onde foi criada uma coluna na tabela para indicar se o registro está ativo ou não. Esse recurso foi adotado para preservar os dados no banco, uma vez que, ao deletar o registro, ele não é removido definitivamente, mas sim desativado. Isso significa que este dado pode ser recuperado posteriormente.
- Existem dois tipos de classificações de entidades: Entidade Forte, que existe independentemente de outra entidade, e Entidade Fraca, que depende da existência de outra entidade. Portanto, as Entidades Fortes representadas com um retângulo incluem "usuario" e "produto", enquanto as Entidades Fracas representadas por 2 retângulos um sobre o outro, são: "endereco", "recuperacao_senha", "confirmacao_conta", "agricultor", "cliente", "administrador", "anuncio", "itens_de_pedido" e "pedido".

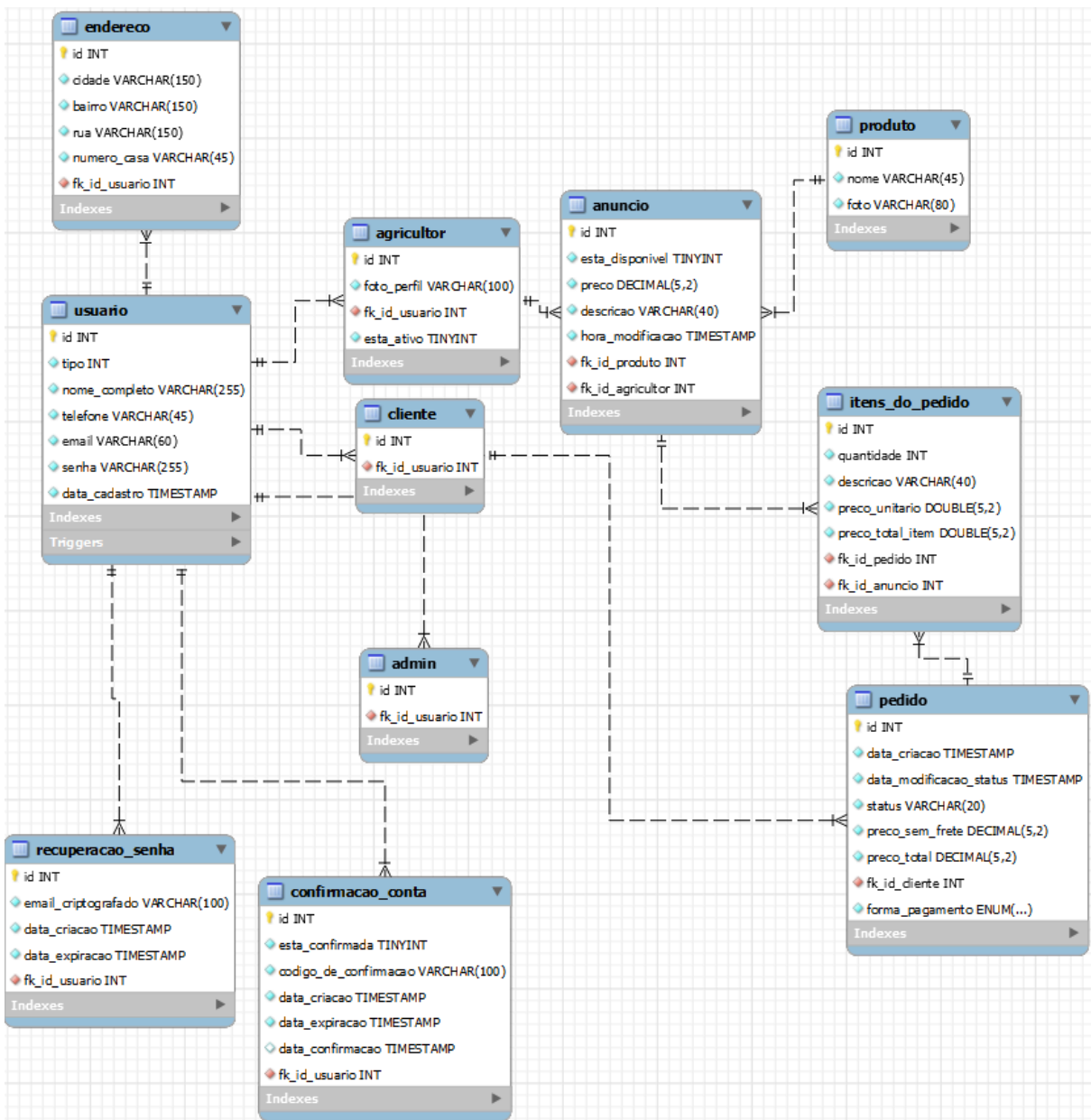
5.3.2 Modelagem Lógica e Modelagem Física

O propósito da modelagem lógica é converter as entidades e atributos apresentados na modelagem conceitual em tabelas e colunas, respectivamente, proporcionando uma representação abstrata e estruturada dos dados. Por outro lado, a modelagem física se baseia na modelagem lógica e acrescenta detalhes sobre componentes de estrutura física do banco de dados, incluindo tipos de valores, restrições e etc, ou seja, a modelagem física se concentra nos detalhes de implementação e particularidades do banco de dados escolhido (Heuser, 2009).

No decorrer do processo de modelagem deste projeto, o diagrama da modelagem conceitual, apresentado na Figura 33, serviu como base para o desenvolvimento da modelagem física. Para essa etapa, recorreu-se ao software MySQL Workbench, cujo propósito é simplificar a criação de modelos visuais de dados e a administração de bancos de dados MySQL.

A Figura 34 exibe o diagrama da Modelagem Física, elaborado por meio da ferramenta EER Diagram do MySQL Workbench:

Figura 34 - Modelagem física do banco de dados do sistema Feira da Agricultura Familiar



Fonte: Imagem elaborada pelo autor.

Ao observar a Figura 34, é possível analisar os detalhes específicos mencionados acerca do relacionamento entre tabelas, a definição de chaves primárias e estrangeiras, bem como a especificação de tipos e restrições das colunas.

É importante destacar que, considerando a estreita relação entre modelagem lógica e física, foi adotada uma abordagem mais direta. A opção por omitir a etapa da modelagem lógica foi influenciada pela capacidade do software MySQL

Workbench de integrar aspectos tanto lógicos quanto físicos de maneira contínua no processo.

Uma outra vantagem de utilizar o MySQL Workbench para desempenhar o papel de modelagem é a facilidade de converter o diagrama para códigos SQL, que são responsáveis por gerar o banco de dados. A Figura 35 mostra um trecho de código SQL para gerar o banco do projeto.

Figura 35 - Trecho de código SQL responsável por gerar o banco de dados do sistema da agricultura familiar

```
-----  
-- Schema bd_agricultura  
-----  
CREATE SCHEMA IF NOT EXISTS `bd_agricultura` DEFAULT CHARACTER SET utf8 ;  
USE `bd_agricultura` ;  
  
-----  
-- Table `bd_agricultura`.`produto`  
-----  
CREATE TABLE IF NOT EXISTS `bd_agricultura`.`produto` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `foto` VARCHAR(80) NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB  
  
-----  
-- Table `bd_agricultura`.`usuario`  
-----  
CREATE TABLE IF NOT EXISTS `bd_agricultura`.`usuario` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `tipo` INT NOT NULL,  
  `nome_completo` VARCHAR(255) NOT NULL,  
  `telefone` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(60) NOT NULL,  
  `senha` VARCHAR(255) NOT NULL,  
  `data_cadastro` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB
```

Fonte: Imagem elaborada pelo autor.

5.4 BACK-END

A partir de uma parcela do *front-end* desenvolvida e modelagem do banco de dados feita, o desenvolvimento do *back-end* surgiu com mais ênfase neste momento.

Para o *back-end* desta aplicação, a linguagem escolhida foi o PHP, que é utilizada por aproximadamente 78% de todos os sites da internet. Além disso, comparada a outras linguagens de programação o PHP é fácil de aprender e possui um conjunto de recursos disponíveis para iniciantes (HOSTINGER, 2023).

Na implementação da aplicação, o PHP foi designado para desempenhar papéis tanto no *back-end* quanto na manipulação do HTML (*front-end*), adotando assim uma abordagem de PHP *Server-Side Rendering*. Este termo refere-se ao

processo pelo qual as páginas web são geradas no lado do servidor dinamicamente utilizando PHP antes de serem enviadas para o navegador do cliente. Conforme mostra a Figura 36:

Figura 36 - *Server-Side Rendering* na prática. Desde a requisição do cliente até a resposta preparada do servidor.

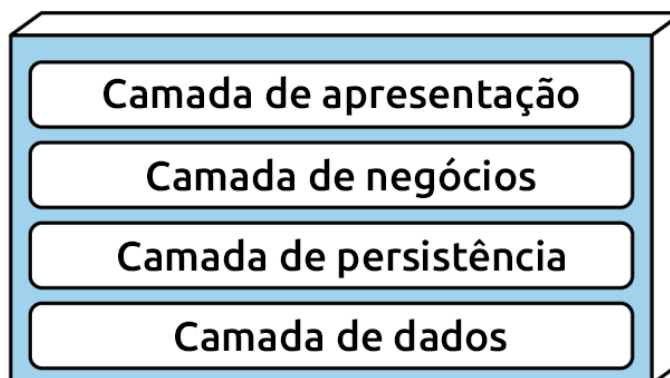


Fonte: Imagem elaborada pelo autor.

Em outras palavras, pode-se afirmar que o PHP foi usado como uma ferramenta *full-stack*, atuando tanto no front-end quanto no back-end.

Se tratando de padrões arquiteturais, este modelo de desenvolvimento se alinha com o modelo de arquitetura monolítica, a qual refere-se a um software criado com uma única base de código, ou seja, seus componentes trabalham compartilhando o mesmo espaço de memória e recursos formando uma unidade de código coesa. Diferentemente da abordagem de microsserviços, onde o projeto é um conjunto de processos separados e independentes, se comunicando por meio de algum protocolo (MENDES, 2021). A Figura 37 apresenta uma estrutura básica de uma aplicação monolítica.

Figura 37 - Estrutura básica de um monolítico



Fonte: Richards; Ford, 2020, tradução pelo autor.

Camada de apresentação: destinada a apresentar o conteúdo ao usuário final através de interfaces gráficas;

Camada de negócios: destinada a programar as regras de negócios da aplicação;

Camada de persistência: destinada para estabelecer comunicação com o banco de dados;

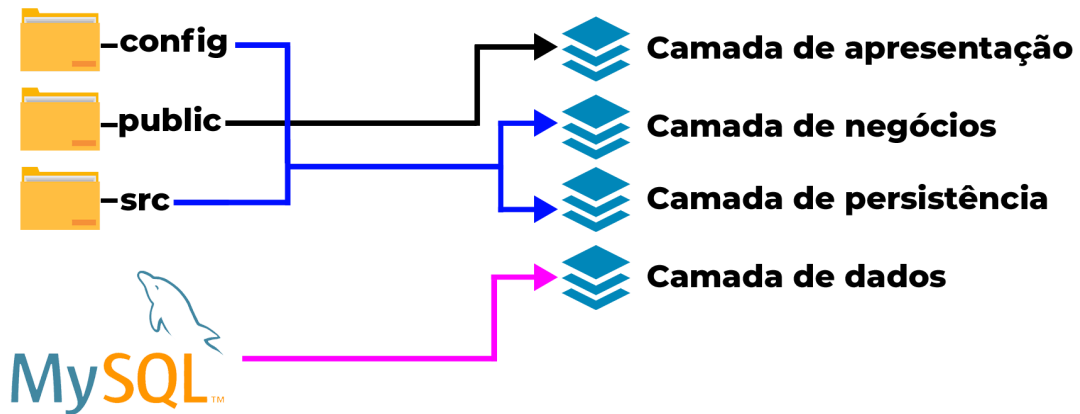
Camada de dados: destinada aos armazenamentos dos dados da aplicação.

Embora este projeto se alinhe com a arquitetura monolítica, no desenvolvimento das primeiras versões da aplicação, os conceitos relacionados a esta arquitetura não estavam devidamente claros, uma vez que a implementação foi realizada em paralelo aos estudos, caracterizando uma dinâmica de aprendizado durante o processo.

É possível considerar que tal abordagem se alinha com um dos princípios destacados no Manifesto Ágil, que afirma: “Software em funcionamento é a principal medida de progresso” (Manifesto Ágil, 2001). Desse modo, nas primeiras versões o foco estava no funcionamento do software, sendo que as boas práticas não foram consideradas como prioridade durante este estágio inicial. Entretanto, isso gerou o que é chamado de dívida de técnica, a qual será discutida na Seção de 5.5.

A relação entre a arquitetura monolítica mostrado na Figura 37 e a estrutura de pastas do projeto, é mostrada na Figura 38:

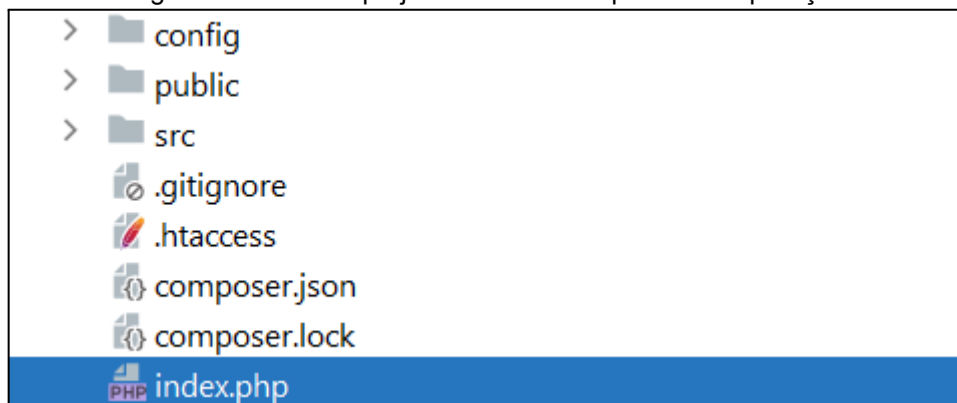
Figura 38 - Relação da estrutura de pastas com as camadas de uma aplicação monolítica.



Fonte: Imagem elaborada pelo autor.

A seguir será discutido o que há em cada uma dessas pastas, antes disso a Figura 39 mostra a raiz do projeto.

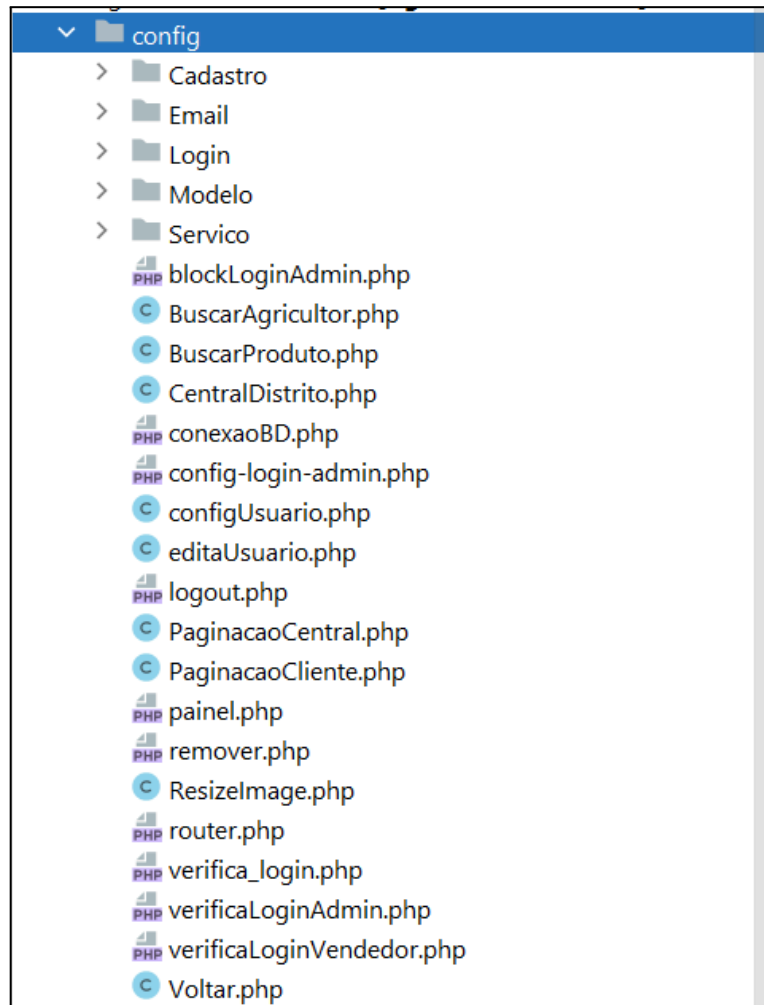
Figura 39 - Raiz do projeto. Estrutura de pastas da aplicação



Fonte: Imagem elaborada pelo autor.

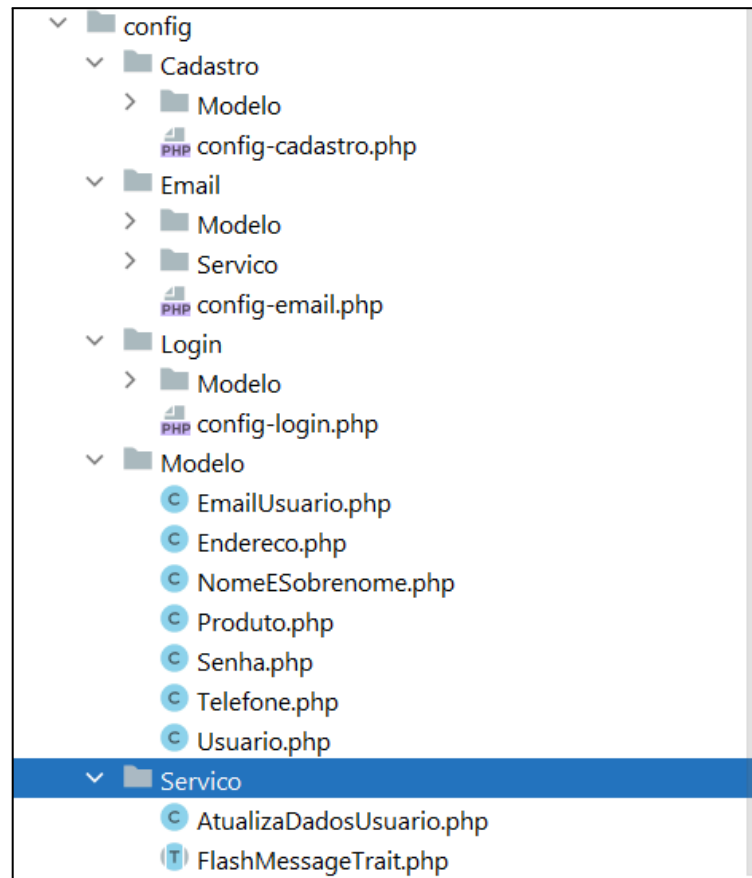
A pasta “**config**” é responsável por armazenar toda a regra de negócio da aplicação, juntamente com a busca e persistência dos dados. A Figura 40 mostra mais detalhes:

Figura 40 - Detalhes de pastas e arquivos no pacote config.



Fonte: Imagem elaborada pelo autor.

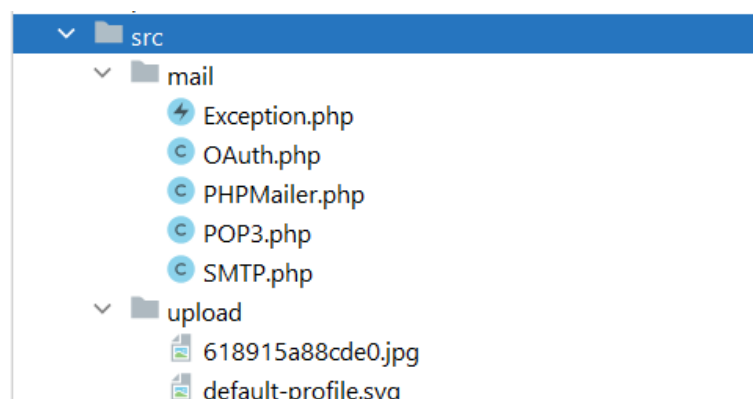
Figura 41 - Detalhes das pastas internas do pacote config



Fonte: Imagem elaborada pelo autor.

Além da pasta “config”, a pasta "src" também é responsável pelas regras de negócios. Ela armazena o código da biblioteca *PHPMailer*, responsável pelo o envio de e-mails, e a pasta "upload", onde são armazenadas as imagens enviadas de perfil dos agricultores. Conforme mostra a Figura 42:

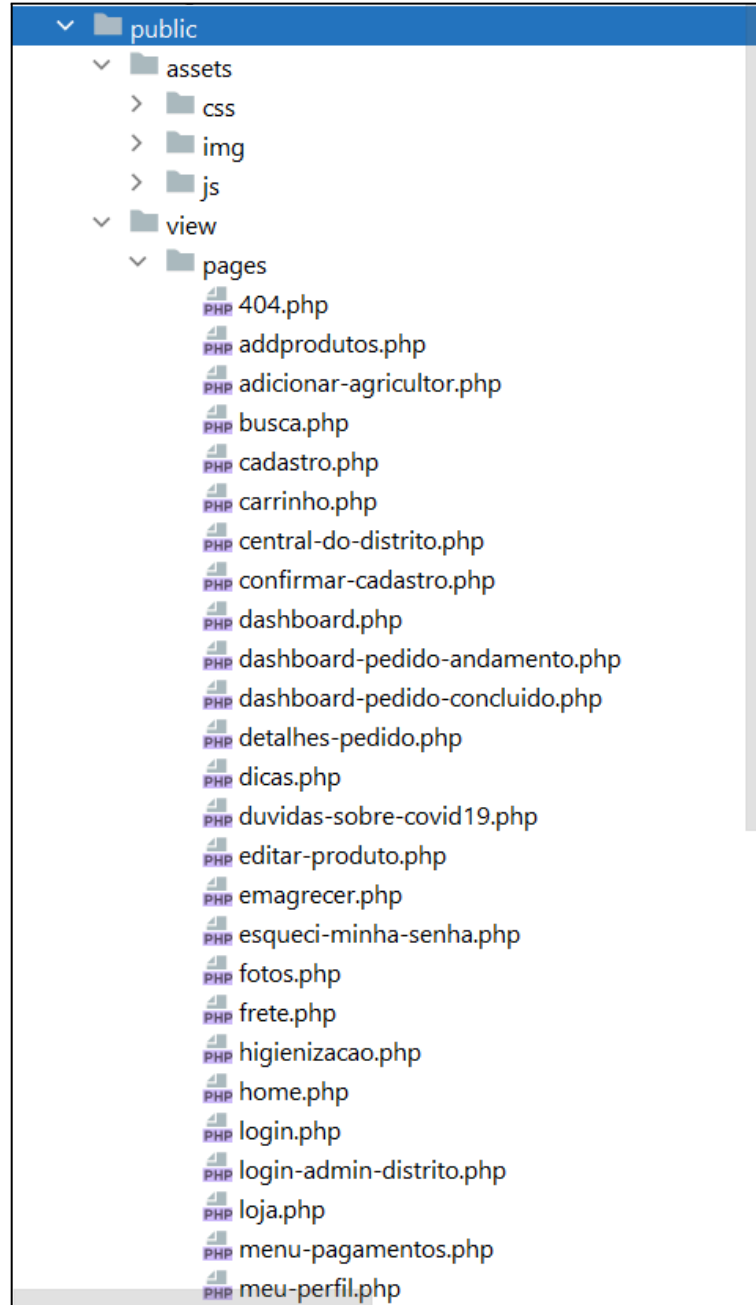
Figura 42 - Detalhes das pastas internas do pacote src



Fonte: Imagem elaborada pelo autor.

A pasta “public” é responsável por guardar os arquivos relacionados a visualização da aplicação, como é mostrada na Figura 43:

Figura 43 - Detalhes das pastas internas do pacote public



Fonte: Imagem elaborada pelo autor.

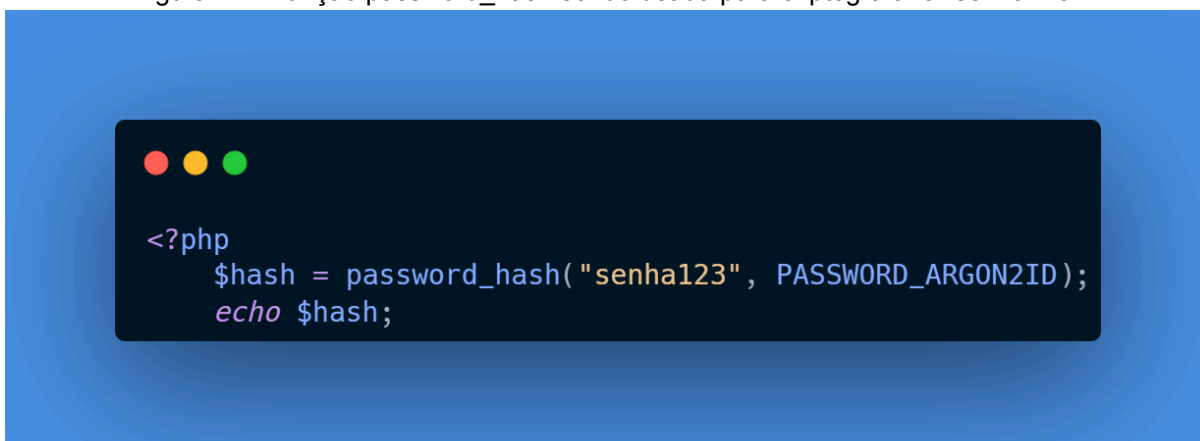
Cada arquivo corresponde a uma interface e, uma vez que se trata de um código em PHP, foram incorporados trechos de código HTML, juntamente com código PHP contendo as regras de negócio associadas.

5.4.1 Armazenamento de senha

A senha de um usuário é um dado confidencial que se não for armazenado de maneira correta pode acarretar em roubo de conta. Uma das soluções para amenizar este tipo de problema, é não salvar a senha no banco de dados como texto puro, em vez disso, é recomendável utilizar um método de criptografia para ofuscar essa informação.

O PHP fornece uma função nativa chamada de **password_hash**, que é comumente utilizada para gerar *hashes* seguros de senhas. Como parâmetros, essa função espera uma *string* que é a senha pura, em seguida, espera um algoritmo de hash (PHP, 2023a), conforme mostra a Figura 44:

Figura 44 - Função password_hash sendo usada para criptografar a "senha123"

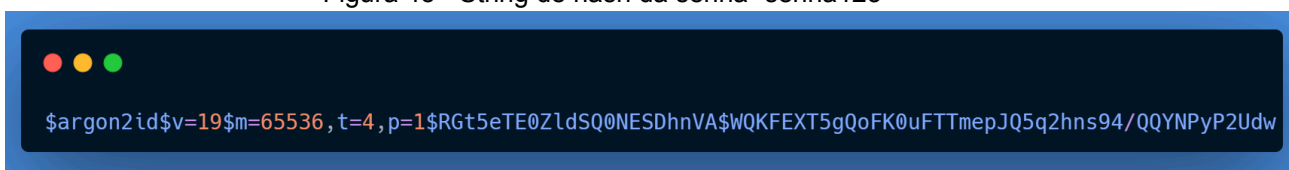
A screenshot of a terminal window with a blue background. The terminal has a dark gray window with three colored window control buttons (red, yellow, green) in the top left corner. The code displayed is PHP code for hashing the password "senha123" using the PASSWORD_ARGON2ID algorithm. The code is:

```
<?php
$hash = password_hash("senha123", PASSWORD_ARGON2ID);
echo $hash;
```

Fonte: Imagem elaborada pelo autor.

Conforme ilustrado na Figura 44, o algoritmo criptográfico de hash escolhido é o “*Argon2id*”, desenvolvido como uma variação do Argon2, vencedor da competição *Password Hashing Competition* (PHC) em 2015. O Argon2id foi concebido para resistir a ataques de força bruta e de dicionário (Sierti, 2023). Para uma mesma senha, como "senha123", cada execução do código da Figura 44 resultará em um hash diferente, ampliando a segurança da senha. A Figura 45 demonstra o resultado obtido.

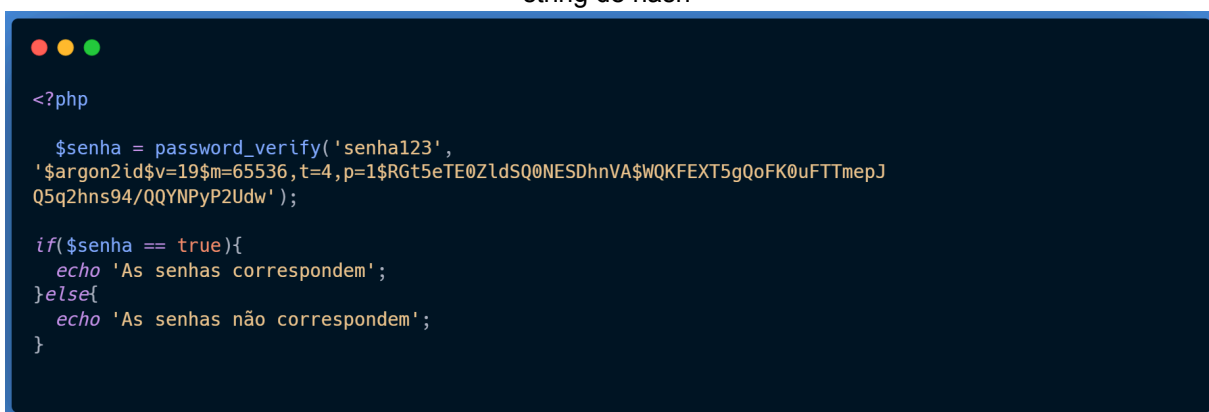
Figura 45 - String de hash da senha “senha123”



Fonte: Imagem elaborada pelo autor.

Para realizar o caminho inverso — descriptografar, o PHP possui outra função nativa intitulada de **password_verify**, responsável por verificar se uma senha coincide com uma string de hash. Como parâmetros, ela espera receber uma string representando a senha, seguida por uma string de hash (PHP, 2023b). Se a senha coincidir a string de hash, a função retornará *true*, caso contrário, *false*. Como mostra a Figura 46:

Figura 46 - Função password_verify certificando a correspondência de uma string de senha com uma string de hash



Fonte: Imagem elaborada pelo autor.

As funções **password_hash** e **password_verify** são utilizadas no procedimento de cadastro e autenticação de usuários na plataforma.

5.4.2 Autenticação

A autenticação representa uma etapa crucial no desenvolvimento de software, certificando a identidade declarada pelo usuário por meio de suas credenciais — e-mail e senha. Quando essas credenciais coincidem com as registradas no banco de dados, o acesso é liberado; caso contrário, o acesso é negado (FREECODECAMP, 2023).

O método de autenticação adotado na aplicação é o de **autenticação de fator único**, demandando apenas um único elemento para autenticação, neste caso,

a senha (FREECODECAMP, 2023). A escolha desse método de autenticação segue a premissa de tornar a aplicação mais prática e menos burocrática para os usuários, especialmente considerando que o público-alvo são agricultores, muitos dos quais enfrentam desafios ao adotar novas tecnologias devido ao atraso tecnológico e analfabetismo (BUAINAIN, 2021). No entanto, é importante observar que essa abordagem, embora simplificada, resulta em uma redução de segurança, tornando a aplicação mais vulnerável a ataques de *phishing*³⁶ e *keyloggers*³⁷.

5.4.3 Autorização

Após confirmar quem o usuário diz ser, por meio do processo de autenticação, torna-se necessário verificar suas permissões no sistema, procedimento realizado por meio da autorização.

Sabendo que cada requisição HTTP é independente, é fundamental transportar informações que identifiquem se o usuário está autenticado ou se possui permissão para acessar determinada rota. Para atender a essa necessidade, foram utilizados mecanismos como sessões e *cookies*.

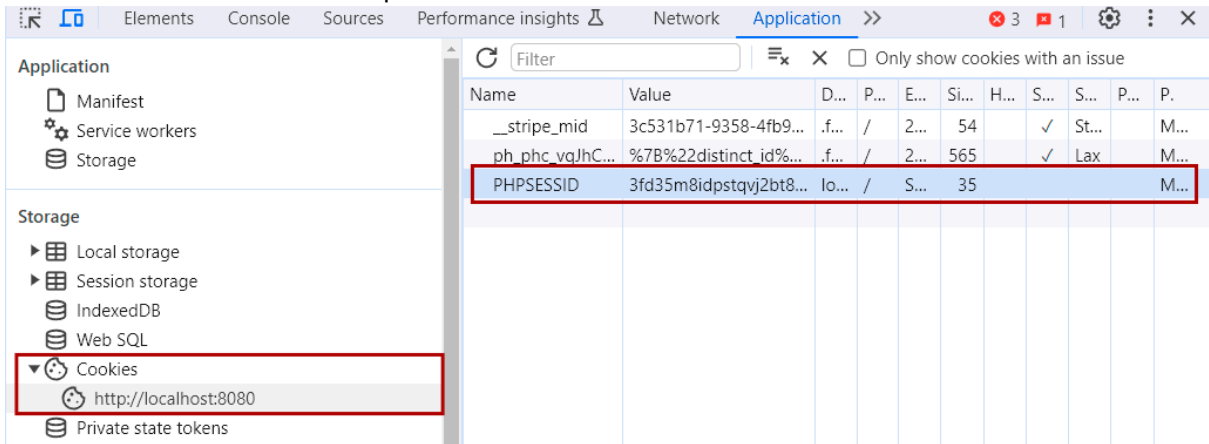
Cookies é um recurso criado em 1994 que permite o salvamento de estado entre cliente e servidor. Trata-se de uma string de texto armazenada no navegador do cliente, a qual é enviada junto com as requisições ao servidor por meio dos cabeçalhos HTTP quando o cliente acessa um determinado site. Estes cookies servem para personalizar a experiência do usuário, armazenando dados como nomes, carrinhos de compras, entre outros (CAHN, 2016).

Diferentemente dos cookies, as sessões são dados armazenados no lado do servidor web, normalmente dados sensíveis de usuário que deseja-se transportar entre uma requisição e outra. Sabendo disso, as sessões utilizam cookies para relacionar um identificador único chamado PHPSESSID com os dados contidos no servidor (GANESHICMC, 2023). A Figura 47 mostra o valor do PHPSESSID como cookie no navegador.

³⁶ Phishing é uma tentativa fraudulenta de obter dados sensíveis disfarçando-se como entidade confiável.

³⁷ Keyloggers são programas que registram as teclas digitadas para roubar informações.

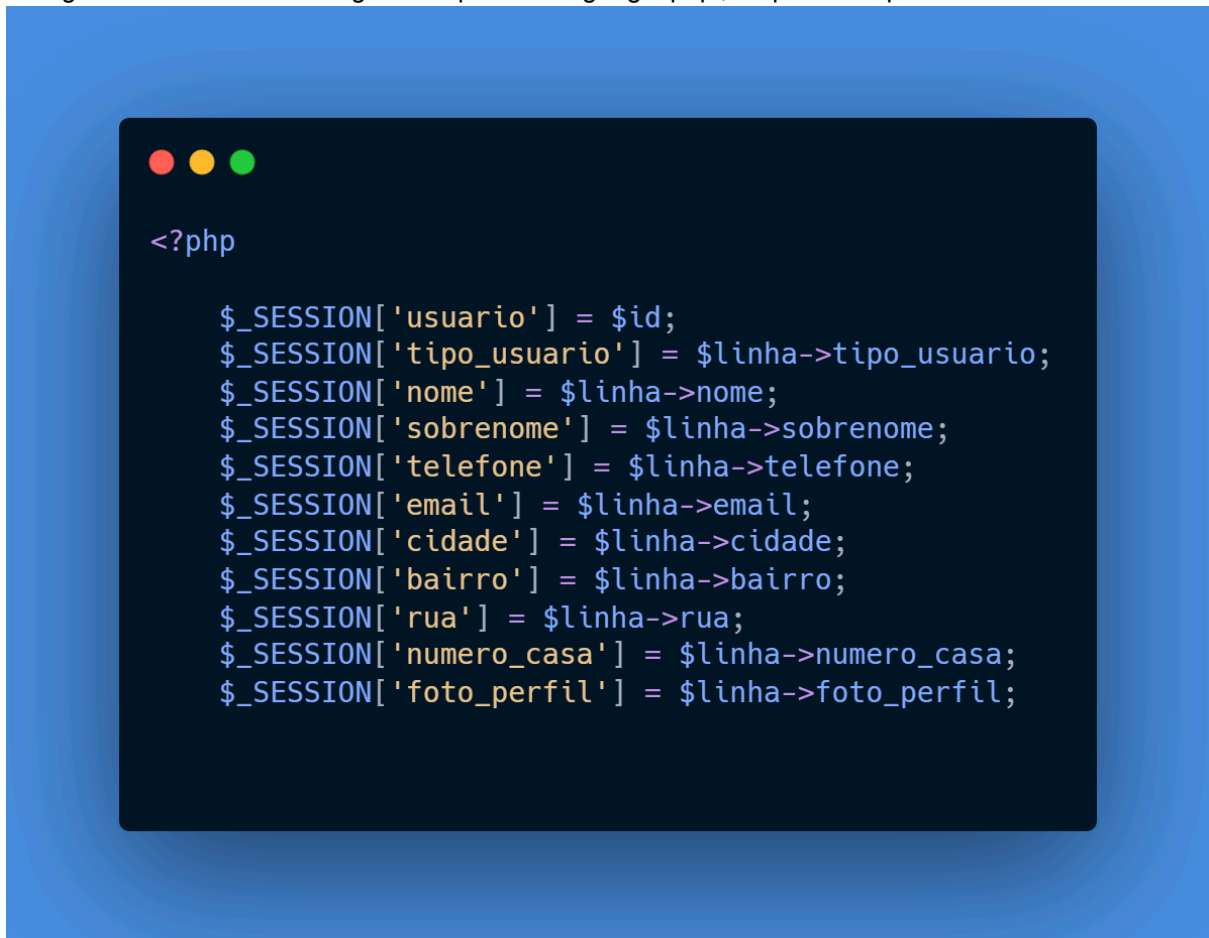
Figura 47 - Cookie de identificador único (PHPSESSID). Este identificador é relacionado com uma sessão que está armazenada do lado do servidor web.



Fonte: Imagem elaborada pelo autor.

Para iniciar uma sessão em PHP, utiliza-se a função **session_start**, a partir desse ponto, torna-se possível utilizar a variável superglobal **\$_SESSION** para armazenar e acessar dados de qualquer escopo do projeto. Diante destas condições, alguns dados do usuário autenticado foram atribuídos à sessão, como ilustrado na Figura 48:

Figura 48 - Trecho de código do arquivo 'config-login.php', responsável por autenticar o usuário.



Fonte: Imagem elaborada pelo autor.

Determinadas rotas da aplicação são restritas a usuários autorizados, mais especificamente àqueles que possuem a autorização de agricultores. A representação gráfica na imagem acima exemplifica o array `$_SESSION['tipo_usuario']`, no qual os valores podem ser: 1 para Cliente e 2 para Agricultor.

Considerando essa informação, foi desenvolvido um arquivo responsável por efetuar a verificação se um usuário é do tipo agricultor, conforme exemplificado na Figura 49:

Figura 49 - Trecho de código do arquivo 'verificaLoginVendedor.php', responsável por verificar se o tipo de usuário é agricultor.



```
<?php

if($_SESSION['tipo_usuario'] !== 1){
    header('Location: loja');
    exit();
}
```

Fonte: Imagem elaborada pelo autor.

Dessa forma, caso uma rota específica (arquivo de interface) esteja restrita apenas a agricultores, utiliza-se o recurso *require* do PHP para incluir esse arquivo antes da execução das regras de negócio.

Conseqüentemente, esta abordagem de validação de tipo de usuário é aplicada de maneira semelhante para as demais rotas durante o desenvolvimento.

5.4.4 Roteamento

Em sistemas online, o processo de navegação entre diferentes páginas e funcionalidades requer uma estrutura programada de forma robusta para organizar o fluxo da aplicação, proporcionando uma experiência agradável ao usuário. Diante disso, o conceito de roteamento surge como uma solução centralizada.

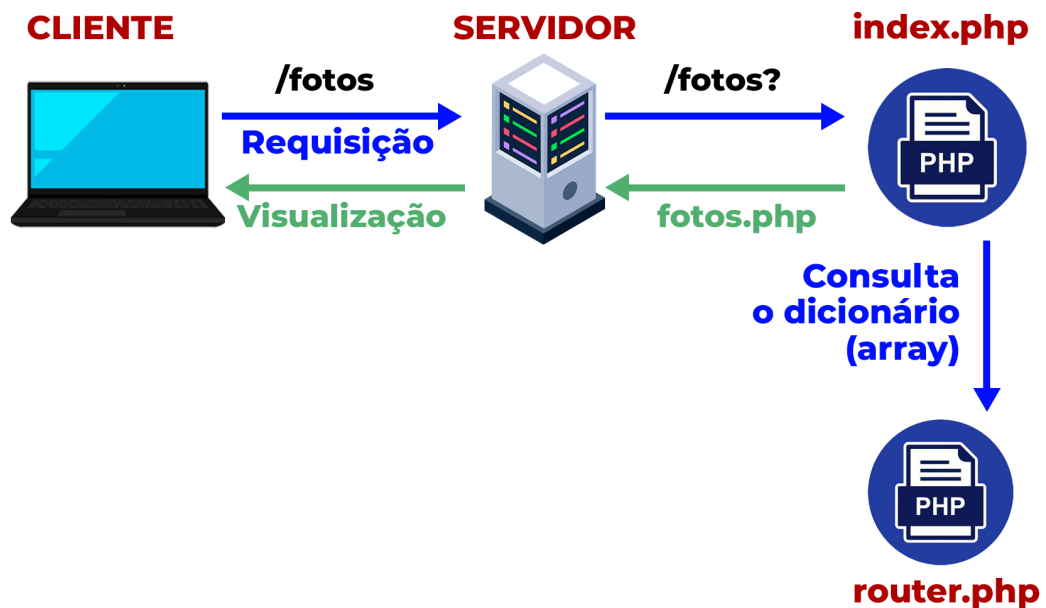
Na aplicação da agricultura familiar, optou-se por não utilizar nenhum *framework* para desempenhar o papel de roteamento. Em virtude disso, o roteamento foi implementado inspirado no padrão *Front Controller*, a qual em um servidor web, o arquivo `index.php` atua como o ponto de entrada único da aplicação, assumindo a responsabilidade pelo encaminhamento das requisições aos controladores correspondentes (SINGH, 2020).

O padrão *Front Controller* é usado comumente em aplicações que adotam a arquitetura MVC — frameworks robustos do mercado, como *Laravel*, *Symfony*, *CakePHP* e entre outros. As vantagens oferecidas por esse padrão são diversas: centralização do controle da aplicação, manutenção sintetizada, roteamento flexível, implementação de middlewares, camada de visualização simplificada e

padronização de requisições e respostas (APPMASTER.IO, 2023). Estes ganhos significativos serão explorados e discutidos na seção dedicada à Refatoração.

A Figura 50 ilustra um exemplo de como ocorre o processo de roteamento da aplicação, desde a requisição do cliente até a resposta do servidor.

Figura 50 - Processo de roteamento a partir de uma requisição para o recurso “/fotos” até a resposta pelo servidor.



Fonte: Imagem elaborada pelo autor.

Considerando que o servidor está configurado para redirecionar todas as requisições para o arquivo index.php, a situação ilustrada na Figura 50 ocorre da seguinte forma: quando o cliente acessa a página de fotos da aplicação, a requisição é recebida pelo servidor. Este, por sua vez, invoca o arquivo index.php, que realiza uma verificação para determinar se o recurso desejado pelo cliente (/fotos) está mapeado na aplicação. Essa verificação é consultada por meio de um array associativo do arquivo router.php, onde a chave representa o recurso (/fotos) e o valor indica o diretório do arquivo responsável por executar a lógica de negócios e retornar à visualização ao cliente.

A Figura 51 ilustra uma porção do array associativo presente no arquivo router.php, enquanto a Figura 52 representa o trecho de código do arquivo index.php.

Figura 51 - Arquivo router.php responsável por mapear as rotas da aplicação.

```

<?php

define('PASTA_PUBLICA', 'public/view/pages');
define('PASTA_CONFIG', 'config/');
define('PASTA_CONFIG_CADASTRO', 'config/Cadastro');
define('PASTA_CONFIG_LOGIN', 'config/Login');

$routeur = [
    'login' => PASTA_PUBLICA,
    'logout' => PASTA_CONFIG,
    'carrinho' => PASTA_PUBLICA,
    'loja' => PASTA_PUBLICA,
    'config-login' => PASTA_CONFIG_LOGIN,
    'config-cadastro' => PASTA_CONFIG_CADASTRO,
    'meu-perfil' => PASTA_PUBLICA,
    'perfil' => PASTA_PUBLICA
];

return $routeur;

```

Fonte: Imagem elaborada pelo autor

Figura 52 - Arquivo index.php atuando como Front Controller

```

<?php

$routeur = require __DIR__ . "/config/router.php";
$address = $_SERVER['PATH_INFO'];

$url = explode('/', $address);

if (empty($url[1])) {
    require 'public/view/pages/home.php';
    exit();
}

if (!array_key_exists($url[1], $routeur)) {
    header('Location: 404');
    exit();
}

require $routeur[$url[1]] . '/' . $url[1] . '.php';

```

Fonte: Imagem elaborada pelo autor.

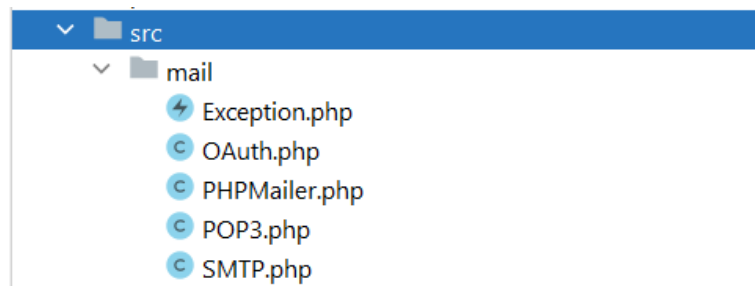
5.4.5 Dependências

Durante o processo de desenvolvimento, não havia familiaridade com o uso de ferramentas para gerenciamento de dependências do projeto. Assim, a

abordagem adotada foi o download direto do código-fonte da biblioteca utilizada no projeto. A biblioteca externa empregada no projeto foi o *PHPMailer*, desempenhando a função de facilitar o envio de e-mails no processo de confirmação de cadastro da aplicação.

Os arquivos da biblioteca foram armazenados na pasta src. Como mostrado na Figura 53:

Figura 53 - Pasta src com dados da biblioteca PHPMailer



Fonte: Imagem elaborada pelo autor.

5.4.6 Persistência

Para realizar a comunicação com o Banco de Dados, foi utilizado uma biblioteca nativa do PHP chamada de MySQLi, a qual proporciona suporte para o banco de dados MySQL. A Figura 54 exemplifica a utilização desta biblioteca:

Figura 54 - Arquivo BuscarProduto.php responsável por armazenar algumas consultas e persistência no banco de dados por meio da biblioteca MySQLi.

```
<?php
class BuscarAgricultor{
    private $agricultor;

    public function __construct(mysqli $agricultor){
        $this->agricultor = $agricultor;
    }

    public function exibirAgricultor(): array{
        $resultado = $this->agricultor->query('SELECT nome, sobrenome,cidade,id_usuario, foto_perfil
        FROM usuario
        WHERE tipo_usuario = 1
        ORDER BY cidade ASC, nome');
        $todos_agricultores = $resultado->fetch_all(MYSQLI_ASSOC);

        return $todos_agricultores;
    }
}
```

Fonte: Imagem elaborada pelo autor.

5.5 REFATORAÇÃO

Essa seção é destinada a detalhar o processo de refatoração do software, desde a análise das melhorias necessárias até a implementação do produto final.

5.5.1 Gerenciador de dependências

A ausência de uma ferramenta para gerenciar as dependências do projeto na primeira versão do software derivou-se da limitação de conhecimento na época, agravada pela pressão temporal para atender as demandas. No entanto, durante o processo de refatoração, implementou-se o *Composer*, principal gerenciador de dependências do PHP.

Figura 55 - Logotipo do gerenciador de dependências do PHP — Composer

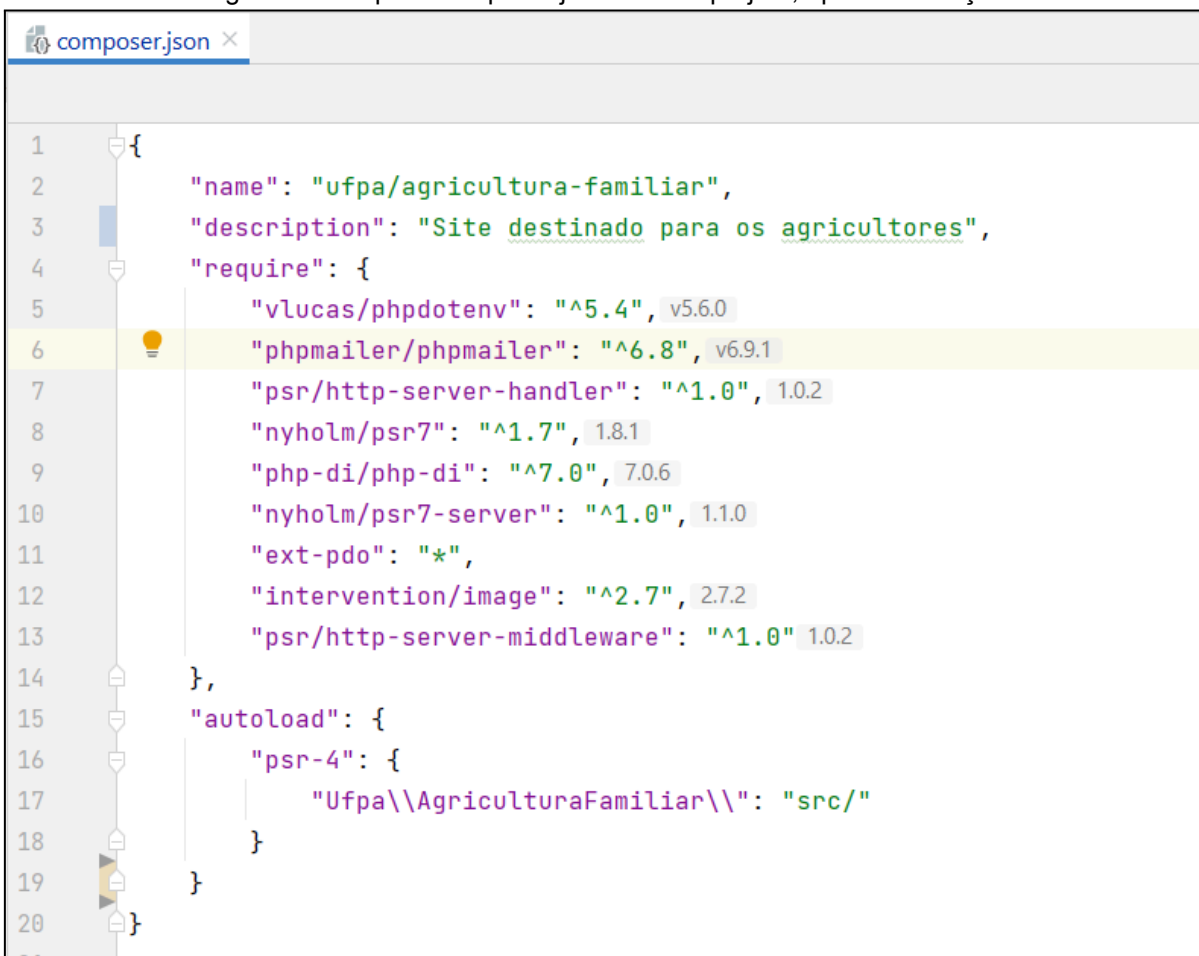


Fonte: getcomposer.org

O Composer simplifica o gerenciamento de dependências no PHP por meio do arquivo `composer.json`. Esse arquivo lista e registra as bibliotecas necessárias para o projeto, juntamente com suas versões compatíveis. Por padrão, essas dependências podem ser obtidas através do packagist.org e são armazenadas em uma pasta intitulada de *vendor*.

A Figura 56 mostra a estrutura do arquivo `composer.json` usado no projeto.

Figura 56 - Arquivo composer.json final do projeto, após refatoração



```
1 {
2     "name": "ufpa/agricultura-familiar",
3     "description": "Site destinado para os agricultores",
4     "require": {
5         "vlucas/phpdotenv": "^5.4", v5.6.0
6         "phpmailer/phpmailer": "^6.8", v6.9.1
7         "psr/http-server-handler": "^1.0", 1.0.2
8         "nyholm/psr7": "^1.7", 1.8.1
9         "php-di/php-di": "^7.0", 7.0.6
10        "nyholm/psr7-server": "^1.0", 1.1.0
11        "ext-pdo": "*",
12        "intervention/image": "^2.7", 2.7.2
13        "psr/http-server-middleware": "^1.0" 1.0.2
14    },
15    "autoload": {
16        "psr-4": {
17            "Ufpa\\AgriculturaFamiliar\\": "src/"
18        }
19    }
20 }
```

Fonte: Imagem elaborada pelo autor.

A Figura 56 mostra diversas bibliotecas integradas ao projeto final, na primeira versão a única biblioteca usada no projeto era a *PHPMailer*. Contudo, ao longo desta seção, serão citadas outras bibliotecas que desempenharam um papel significativo durante o desenvolvimento.

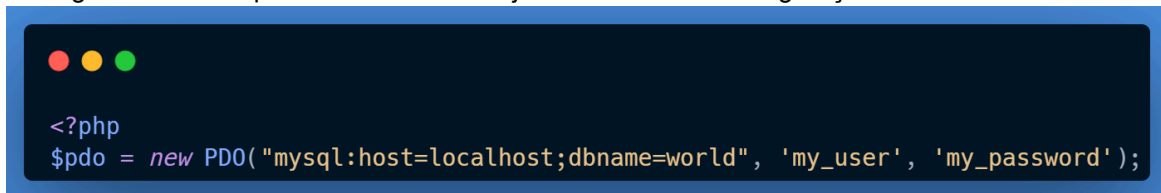
5.5.2 Biblioteca de conexão e persistência

Na primeira versão, optou-se por empregar a biblioteca nativa do PHP — MySQLi — para estabelecer a conexão com o banco de dados MySQL. No entanto, essa escolha apresenta desvantagens em termos de portabilidade do código, visto que o MySQLi é específico para o MySQL. Isso implica que, caso a aplicação precise ser migrada para um banco diferente no futuro, modificações substanciais no código poderão ser necessárias.

Diante dessa problemática, tomou-se conhecimento de uma alternativa viável para conexão e persistência de dados — o PDO, que também é uma biblioteca nativa do PHP. O PDO proporciona uma camada de abstração de banco de dados mais abrangente, permitindo uma transição mais suave entre diferentes sistemas de gerenciamento de banco de dados.

Na Figura 57, é possível observar que o construtor da classe PDO aceita alguns parâmetros, sendo que o primeiro parâmetro espera o nome do driver de banco de dados. No caso da imagem em questão, foi utilizado o driver "mysql", entretanto, é relevante notar que esta biblioteca oferece suporte a diversos outros drivers, tais como Oracle, *PostgreSQL*, *Firebird*, *SQL Server*, entre outros.

Figura 57 - Exemplo de instância do objeto PDO com as configurações do banco de dados

A screenshot of a terminal window with a dark background and a blue border. The terminal shows PHP code for creating a PDO instance. The code is:

```
<?php
$pdo = new PDO("mysql:host=localhost;dbname=world", 'my_user', 'my_password');
```

Fonte: Imagem elaborada pelo autor.

Pode-se concluir que a alteração da biblioteca foi motivada com o objetivo de promover a manutenibilidade e a flexibilidade do código a longo prazo.

5.5.3 Separar as responsabilidades do arquivo de visualização

Na versão inicial da aplicação, cada rota possuía um arquivo designado para lidar com sua respectiva requisição. Por exemplo, a rota "/loja" era associada ao arquivo "loja.php", o qual não apenas executava a lógica de negócios, mas também continha os códigos HTML responsáveis pela apresentação visual. Contudo, essa abordagem resultava em confusões e na mistura de responsabilidades no código.

A solução para essa questão é a criação de um *controller* para cada requisição. O *controller* será encarregado de executar a lógica de negócios e, em seguida, retornar um arquivo de visualização.

Na Figura 58, é possível visualizar o código da classe "PaginaCriarAnuncioController". Este *controller* é encarregado de executar a regra de negócios, que, neste caso, consiste em invocar a classe "ProdutoRepository"

para recuperar todos os produtos disponíveis e retornar esses produtos para serem exibidos ao agricultor no arquivo "criar-anuncio".

Figura 58 - Código do PaginaCriarAnunciosController

```
<?php

namespace Ufpa\AgriculturaFamiliar\Controller\MeuPerfil\Anuncio;

use Nyholm\Psr7\Response;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Ufpa\AgriculturaFamiliar\Helper\RenderizadorDeHtmlTrait;
use Ufpa\AgriculturaFamiliar\Repository\ProdutoRepository;

class PaginaCriarAnunciosController implements RequestHandlerInterface
{
    use RenderizadorDeHtmlTrait;

    public function __construct(private ProdutoRepository $produtoRepository)
    {
    }

    public function handle(ServerRequestInterface $request): ResponseInterface
    {
        $produtos = $this->produtoRepository->buscarTodos();

        return new Response(
            "200",
            body: $this->renderizarHtml("criar-anuncio", ["produtos" => $produtos])
        );
    }
}
```

Fonte: Imagem elaborada pelo autor.

A introdução de classes do tipo controller aprimora a conformidade da aplicação da Feira da Agricultura Familiar com a arquitetura MVC. Agora, as responsabilidades estão mais claramente segmentadas entre os arquivos de Model, View e Controller.

5.5.4 Adicionar um Middleware para efetuar o serviço de autorização


O Middleware é um componente que desempenha o papel de uma camada intermediária entre o sistema operacional e as aplicações. Em termos simples, o middleware proporciona a capacidade de incorporar lógica adicional tanto antes quanto depois de uma solicitação HTTP, por exemplo: autorização, registro de logs e manipulação de erros. Dessa maneira, o uso de middleware não apenas aprimora

a modularidade do sistema, mas também contribui para a robustez e extensibilidade da aplicação.

Nesta aplicação, o recurso de Middleware foi usado para desempenhar as funções de Autorização, verificando se o usuário que tenta acessar determinado recurso possui as permissões adequadas.

O nível de autorização de cada recurso foi estabelecido no dicionário de rotas, em conjunto com seus respectivos controladores. A Figura 59 ilustra como o arquivo "rotas.php" foi definido.

Figura 59 - O arquivo "rotas.php" contém as definições dos controladores e os níveis de acesso associados a cada recurso que o usuário deseja acessar.

A screenshot of a code editor with a dark background and blue border. The code is in PHP and defines an array of routes. Each route is an associative array with 'controller' and 'nivel_acesso' keys. The routes are: 1. GET /dicas: controller is PaginaDicasController::class, nivel_acesso is NivelDeAcesso::Todos. 2. GET /meu-perfil: controller is PaginaMeuPerfilController::class, nivel_acesso is NivelDeAcesso::Logado. 3. POST /alterar-foto: controller is AlterarFotoPerfilController::class, nivel_acesso is NivelDeAcesso::Agricultor. 4. GET /central-do-distrito: controller is CentralDoDistritoController::class, nivel_acesso is NivelDeAcesso::Admin.

```
<?php
$rotas = [
    'GET|/dicas' => [
        'controller' => PaginaDicasController::class,
        'nivel_acesso' => NivelDeAcesso::Todos
    ],
    'GET|/meu-perfil' => [
        'controller' => PaginaMeuPerfilController::class,
        'nivel_acesso' => NivelDeAcesso::Logado
    ],
    'POST|/alterar-foto' => [
        'controller' => AlterarFotoPerfilController::class,
        'nivel_acesso' => NivelDeAcesso::Agricultor
    ],
    'GET|/central-do-distrito' => [
        'controller' => CentralDoDistritoController::class,
        'nivel_acesso' => NivelDeAcesso::Admin
    ]
];
```

Fonte: Imagem elaborada pelo autor.

Com as informações do array associativo mostrado na Figura 59, foi criada a classe "AutorizacaoMiddleware" para verificar a autorização do usuário sobre cada recurso. Caso o usuário não tenha acesso, ele será redirecionado para a página inicial da plataforma. Este comportamento é mostrado na Figura 60:

Figura 60 - Classe "AutorizacaoMiddleware" responsável por avaliar autorização do usuário para determinado recurso do sistema.

```

<?php

namespace Ufpa\AgriculturaFamiliar\Middlewre;

use GuzzleHttp\Psr7\Response;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\MiddlewareInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Ufpa\AgriculturaFamiliar\Enums\NivelDeAcesso\NivelDeAcesso;
use Ufpa\AgriculturaFamiliar\Enums\TipoDeUsuario\TipoDeUsuario;

class AutorizacaoMiddleware implements MiddlewareInterface
{
    public function process(ServerRequestInterface $request, RequestHandlerInterface $handler):
    ResponseInterface
    {
        $nivelDeAcessoDaRota = $request->getAttribute("nivel_acesso");
        $tipoUsuario = $_SESSION['UsuarioAutenticado']->tipo ?? TipoDeUsuario::Visitante;

        switch ($nivelDeAcessoDaRota){
            case NivelDeAcesso::Todos:
                break;
            case NivelDeAcesso::Logado:
                if (!isset($_SESSION['UsuarioAutenticado'])){
                    return new Response(403, ['Location' => '/']);
                }
                break;
            case NivelDeAcesso::Agricultor:
                if ($tipoUsuario != TipoDeUsuario::Agricultor->value){
                    return new Response(403, ['Location' => '/']);
                }
                break;
            case NivelDeAcesso::Admin:
                if ($tipoUsuario != TipoDeUsuario::Admin->value){
                    return new Response(403, ['Location' => '/']);
                }
                break;
        }

        return $handler->handle($request);
    }
}

```

Fonte: Imagem elaborada pelo autor.

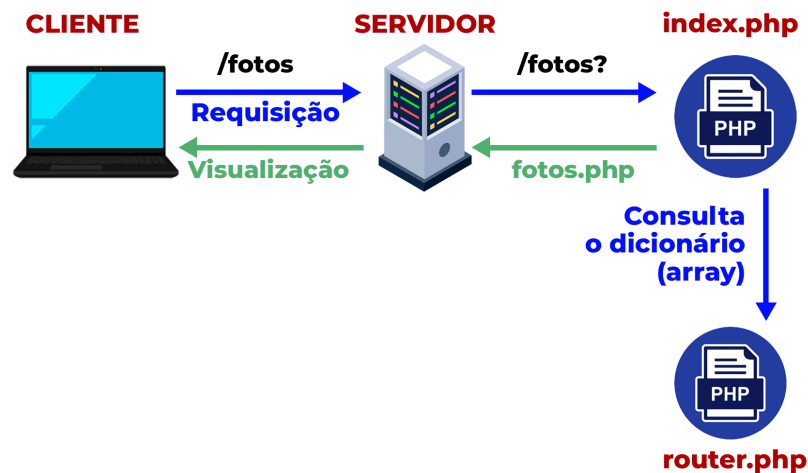
Portanto, se o usuário possuir acesso a um determinado recurso, este Middleware executará o controlador responsável por essa rota e, em seguida, retornará a resposta correspondente.

5.5.5 Atualização do *Front Controller*

Ao realizar o processo de refatoração do código, várias alterações foram introduzidas, afetando principalmente o arquivo "index.php", que desempenha a

função de um Front Controller. Conforme ilustrado na Figura 61, quando uma requisição alcançava a aplicação, o arquivo "index.php" consultava o seu dicionário para identificar a pasta que continha o arquivo correspondente ao recurso, em seguida, executava a regra de negócio e devolvia a visualização ao usuário.

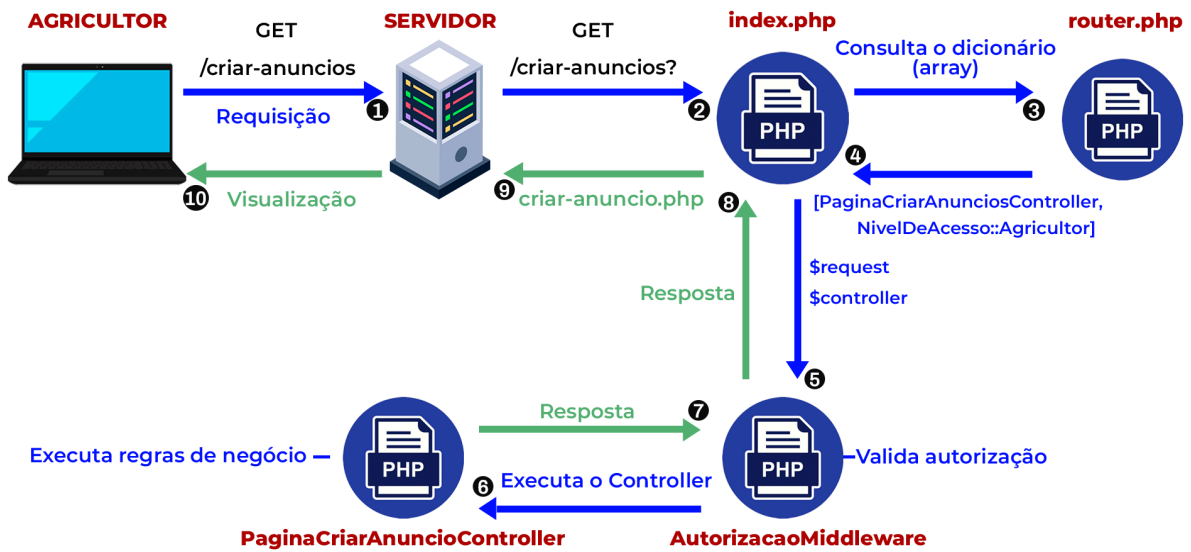
Figura 61 - Processo de roteamento a partir de uma requisição para o recurso "/fotos" até a resposta pelo servidor.



Fonte: Imagem elaborada pelo autor.

A introdução de *middleware* e *controllers* na aplicação invalidou o comportamento mostrado na Figura 61. Por este motivo, foi necessário realizar uma refatoração no fluxo, adotando o padrão atual, conforme apresentado na Figura 62.

Figura 62 - Processo de roteamento atual da aplicação



Fonte: Imagem elaborada pelo autor.

Conforme ilustrado na Figura 62, ao acessar a página "criar-anuncios", a requisição é encaminhada para o servidor. No servidor, essa solicitação é processada pelo arquivo `index.php`, que inicia a aplicação e realiza uma consulta ao dicionário de rotas para obter o *controller* e o nível de acesso associados a essa rota. Neste caso, os valores correspondentes são "PaginaCriarAnunciosController" e "NívelDeAcesso::Agricultor". Com base nessas informações, o arquivo `index.php` chama o "AutorizacaoMiddleware" responsável por validar a autorização. Em seguida, o *middleware* invoca o controller apropriado para lidar com a requisição, e, por fim, o controller retorna a resposta ao middleware, que a repassa ao `index.php` para ser enviada de volta ao cliente.

O fluxo atual, apresentado na Figura 62, demonstra uma robustez significativamente maior em comparação ao anterior (Figura 61). Isso se deve à centralização, segregação de responsabilidades e padronização das informações, garantindo a manutenibilidade do sistema, uma vez que cada componente desempenha sua responsabilidade específica.

5.5.6 Padronizar a aplicação conforme as PSRs

Durante o desenvolvimento, um grande desafio foi encontrar algum padrão de código para seguir, pois é notável que a uniformidade e padronização de código favorecem a colaboração entre desenvolvedores, melhora a legibilidade e promove

uma estrutura mais consistente da aplicação, além de contribuir para a interoperabilidade entre bibliotecas e *frameworks*.

Essa problemática não foi única a este projeto, sendo observada por diversos times de desenvolvimento de software. Diante dessa necessidade, em 2009, foi fundado o *PHP Framework Interop Group* (PHP-FIG), um grupo seletivo de profissionais motivados a buscar melhores práticas de programação e estilos que pudessem definir recomendações seguidas por qualquer desenvolvedor. Assim, surgiram as *PHP Standards Recommendations* (PSRs).

Ao tomar conhecimento das PSRs — www.php-fig.org/psr, foi realizado um processo de estudos sobre essas recomendações, com o objetivo de acrescentá-las na aplicação, os padrões adotados foram:

5.5.6.1 PSR-12

A PSR-12, conhecida como *Extended Coding Style* (Estilo de Codificação Estendido, em tradução livre para o português), foi o primeiro padrão a ser adotado. Este padrão estabelece diretrizes sobre como deve ser o estilo do código fonte. Por exemplo, sugere que nomes de classes devem seguir o padrão *PascalCase*; a tag "`?>`" deve ser omitida de todos os arquivos que contêm apenas código PHP; as visibilidades de uma propriedade, métodos e funções devem ser sempre declaradas; e a abertura de chaves de métodos e funções deve ocorrer na linha seguinte da declaração, entre outros critérios.

5.5.6.2 PSR-4

A PSR-4, denominada *Autoloader* (Carregamento Automático, em tradução livre para o português), foi implementada com o propósito de facilitar o carregamento automático das classes da aplicação por meio do Composer. No "`composer.json`", é especificada a PSR-4 juntamente com o *namespace* da aplicação, o qual é direcionado para a pasta "`src`". A partir dessa configuração, são definidos os *sub-namespaces*, como mostra na Figura 63:

Figura 63 - Definição do *namespace* da aplicação

```

1  {
2      "name": "ufpa/agricultura-familiar",
3      "description": "Site destinado para os agricultores",
4      "require": {
5          "vlucas/phpdotenv": "^5.4", v5.6.0
6          "phpmailer/phpmailer": "^6.8", v6.9.1
7          "psr/http-server-handler": "^1.0", 1.0.2
8          "nyholm/psr7": "^1.7", 1.8.1
9          "php-di/php-di": "^7.0", 7.0.6
10         "nyholm/psr7-server": "^1.0", 1.1.0
11         "ext-pdo": "*",
12         "intervention/image": "^2.7", 2.7.2
13         "psr/http-server-middleware": "^1.0" 1.0.2
14     },
15     "autoload": {
16         "psr-4": {
17             "Ufpa\\AgriculturaFamiliar\\": "src/"
18         }
19     }
20 }
21

```

Fonte: Imagem elaborada pelo autor.

A PSR-4 especifica que o *namespace* deve seguir a seguinte estrutura: “\<NamespaceName>(\<SubNamespaceNames>)*\<ClassName>”, ou seja, para a localizada em “src/Controller/Login/RealizaLoginController.php”, o namespace correspondente é: “Ufpa\AgriculturaFamiliar\Controller>Login”.

5.5.6.3 PSR-7

Posteriormente aderiu-se a PSR-7, conhecida como *HTTP messages interface* (Interfaces de Mensagens HTTP, em tradução livre para o português), que visa aprimorar a representação de mensagens HTTP, abrangendo tanto requisições quanto respostas. Esse padrão é utilizado pela maioria dos *frameworks* e contribui para assegurar a consistência e clareza dos dados HTTP em diversas partes da aplicação. Além disso, a PSR-7 facilita a criação e utilização de *middlewares*.

Para implementar a PSR-7, foi utilizado o componente "psr/http-message" disponibilizado pelo próprio grupo PHP-FIG, que contém interfaces para serem usadas na implementação. Na aplicação, essas interfaces foram utilizadas para

definir o tipo de requisição e o tipo de resposta de um método. Em seguida, obteve-se a biblioteca "nyholm/psr7", que realiza a implementação da PSR-7, permitindo o acesso aos objetos de requisição e resposta de maneira eficaz.

5.5.6.4 PSR-17

Para contribuir com a padronização na representação do protocolo HTTP, adotou-se a PSR-17, também conhecida como *HTTP Factories* (Fábricas HTTP, em tradução livre para o português). Esta norma estabelece uma abordagem orientada a objetos para a criação de requisições a partir dos dados provenientes do navegador. A integração foi facilitada pela utilização da biblioteca "nyholm/psr7-server". Dessa forma, foi possível criar objetos do tipo requisição que podem ser passados como parâmetros para cada um dos controllers.

A vantagem do uso da PSR-17 é que não foi necessário utilizar as superglobais: `$_POST`, `$_GET`, `$_FILES` e entre outras, pois ela empacota todas essas informações em um objeto do tipo *ServerRequestInterface*.

5.5.6.5 PSR-15

Em seguida, adotou-se a PSR-15 conhecida como *HTTP Server Request Handlers* (Manipuladores de solicitação de servidor HTTP, em tradução livre para o português). Essa norma define interfaces para serem utilizadas nos *controllers* e *middleware*, visando garantir uma padronização essencial para esses componentes da aplicação. Para obter tais interfaces, empregamos o pacote "psr/http-server-handler" que disponibilizá-los.

Ao usar o pacote da PSR-15, todos os *controllers* devem implementar a interface "*RequestHandlerInterface*", estabelecendo um contrato que requer a implementação do método denominado "*handle*", responsável por lidar com a requisição. O mesmo princípio se aplica aos *middlewares*, os quais devem implementar a "*MiddlewareInterface*" que contém o método "*process*".

Figura 64 - Exemplo de controller que implementa a interface *RequestHandlerInterface*

```

<?php

namespace Ufpa\AgriculturaFamiliar\Controller\MeuPerfil;

use Nyholm\Psr7\Response;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Ufpa\AgriculturaFamiliar\Helper\RenderizadorDeHtmlTrait;

class PaginaMeuPerfilController implements RequestHandlerInterface
{
    use RenderizadorDeHtmlTrait;

    public function handle(ServerRequestInterface $request): ResponseInterface
    {
        return new Response(
            "200",
            body: $this->renderizarHtml("meu-perfil")
        );
    }
}

```

Fonte: Imagem elaborada pelo autor.

Figura 65 - Exemplo de Middleware que implementa a interface *MiddlewareInterface*

```

<?php

namespace Ufpa\AgriculturaFamiliar\Middleware;

use GuzzleHttp\Psr7\Response;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;
use Psr\Http\Server\MiddlewareInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Ufpa\AgriculturaFamiliar\Enums\NivelDeAcesso\NivelDeAcesso;
use Ufpa\AgriculturaFamiliar\Enums\TipoDeUsuario\TipoDeUsuario;

class AutorizacaoMiddleware implements MiddlewareInterface
{
    public function process(ServerRequestInterface $request, RequestHandlerInterface $handler):
    ResponseInterface
    {
        $nivelDeAcessoDaRota = $request->getAttribute("nivel_acesso");
        $tipoUsuario = $_SESSION['UsuarioAutenticado']->tipo ?? TipoDeUsuario::Visitante;

        switch ($nivelDeAcessoDaRota){
            case NivelDeAcesso::Todos:
                break;
            case NivelDeAcesso::Logado:
                if (!isset($_SESSION['UsuarioAutenticado'])){
                    return new Response(403, ['Location' => '/']);
                }
                break;
            case NivelDeAcesso::Agricultor:
                if ($tipoUsuario != TipoDeUsuario::Agricultor->value){
                    return new Response(403, ['Location' => '/']);
                }
                break;
            case NivelDeAcesso::Admin:
                if ($tipoUsuario != TipoDeUsuario::Admin->value){
                    return new Response(403, ['Location' => '/']);
                }
                break;
        }

        return $handler->handle($request);
    }
}

```

Fonte: Imagem elaborada pelo autor.

Vale destacar que pelo fato de seguir a PSR-15, torna-se possível criar uma fila de *middlewares* para encadear mais processos antes ou depois da requisição. No entanto, no projeto atual, foi implementado apenas um *middleware*, cuja funcionalidade é validar a autorização do usuário.

5.5.6.6 PSR-11

A padronização dos controllers, que requer que cada um receba no construtor um objeto de requisição do tipo "ServerRequestInterface", gera uma problemática. Na linguagem PHP, na versão atual 8.2, não é possível utilizar mais de um construtor por classe. Além disso, cada controller pode necessitar de outras dependências específicas, como alguma classe de serviço, além da classe de requisição. Para resolver essa questão, foi implementada a PSR-11, conhecida como Container interface (Interface de Contêiner, em tradução livre para o português).

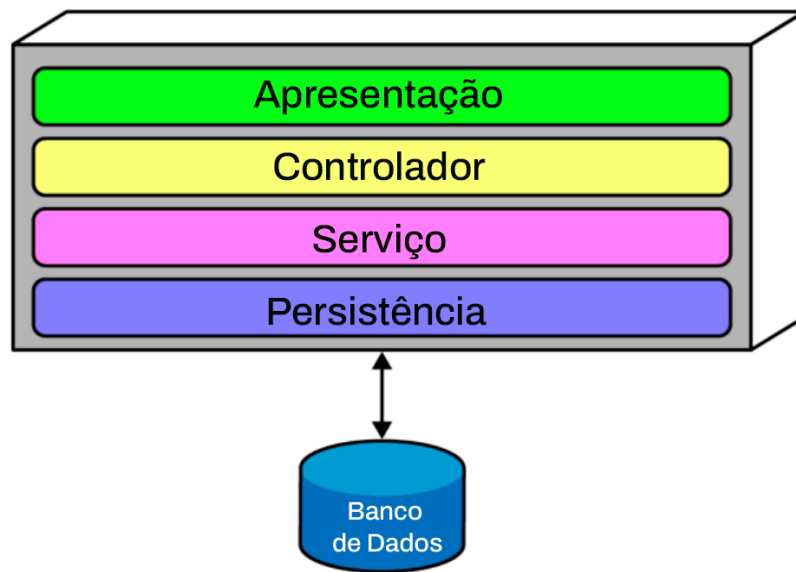
A PSR-11 estabelece uma interface para a implementação de um container de injeção de dependência, que é uma ferramenta de gerenciamento facilitadora na resolução e provisão de dependências em um sistema. Em termos simples, esse container recebe uma classe, verifica suas dependências, cria automaticamente essas dependências e, por fim, instancia a classe, retornando o respectivo objeto criado.

Para usufruir dessa facilidade, optou-se por usar um componente que já utiliza esta PSR-11, como o: "php-di/php-di".

5.5.7 Segregar toda a aplicação em camadas

Na primeira versão da aplicação, a criação de funcionalidades era confusa devido à presença de vários arquivos soltos, muitos dos quais tinham mais de uma funcionalidade. Diante disso, decidiu-se fortalecer a arquitetura monolítica, adotando uma abordagem em camadas mais robusta, conforme ilustrado na Figura 66:

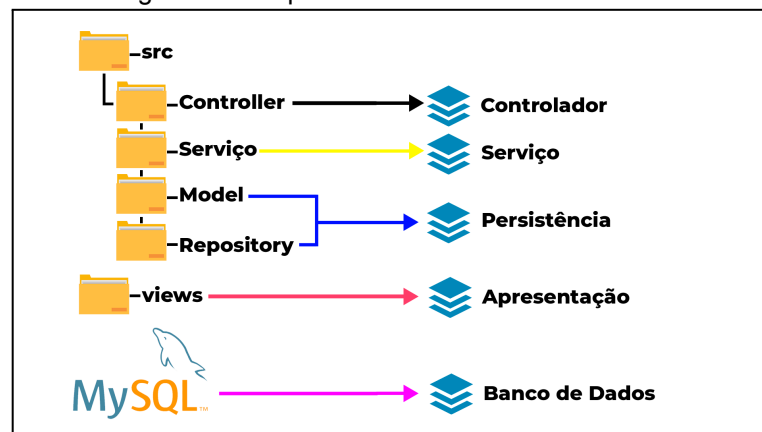
Figura 66 - Arquitetura tradicional em camadas



Fonte: Richards; Ford, 2020, tradução pelo autor.

A relação entre a arquitetura mostrada na Figura 66 e a estrutura de pastas do projeto, é mostrada na Figura 67:

Figura 67 - Arquitetura monolítica em camadas



Fonte: Imagem elaborada pelo autor.

É importante destacar que na pasta "src" não se limita apenas às camadas apresentadas na Figura 67. Existem outras pastas, como "DTO" e "Enums", que não se relacionam diretamente com nenhuma camada mencionada acima, sendo classificadas como auxiliares.

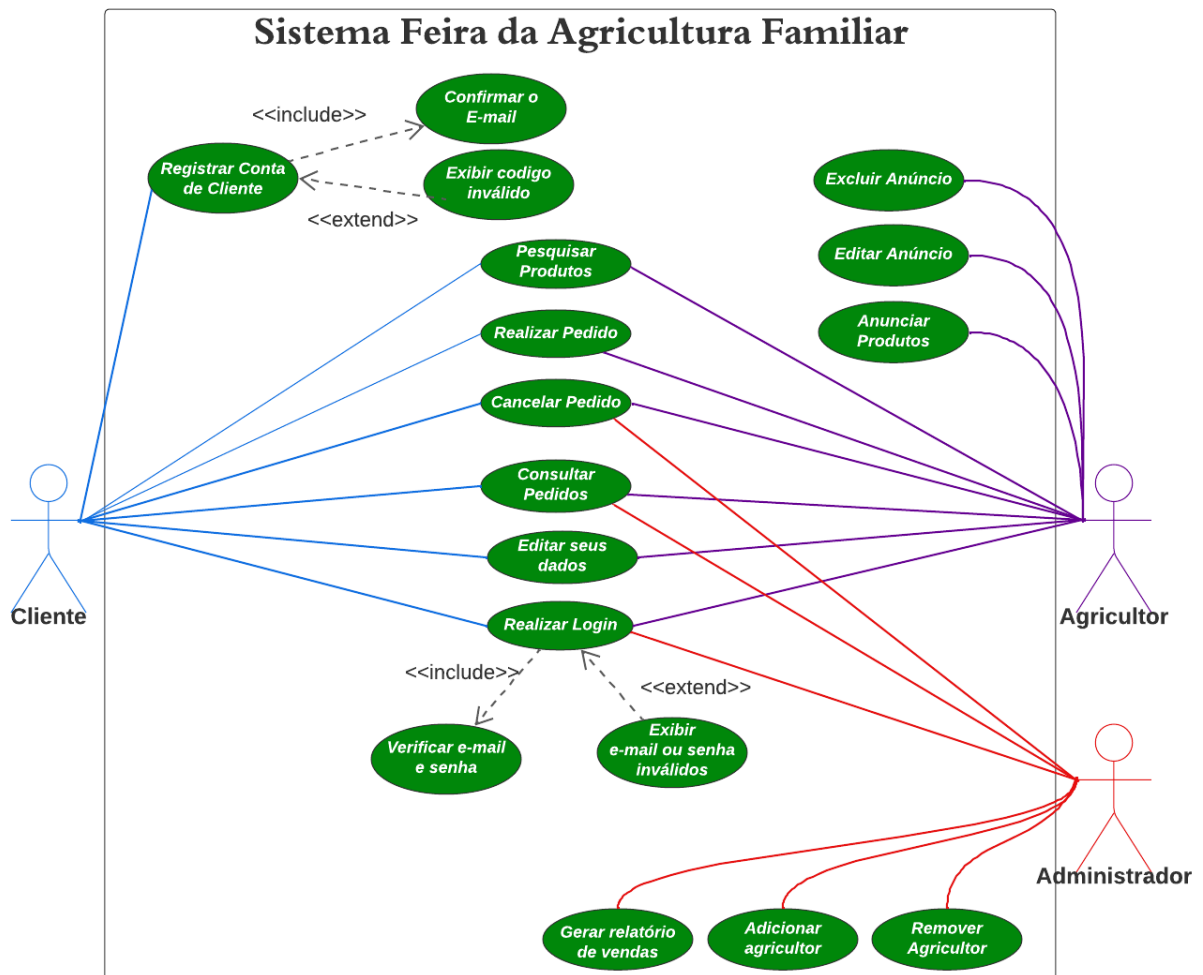
5.6 DOCUMENTAÇÃO

Essa seção destina-se a apresentar a documentação do software, abrangendo desde a definição dos casos de uso até os diagramas de sequência.

5.6.1 Diagramas de Casos de Uso

O diagrama de caso de uso possui o objetivo de descrever os principais comportamentos dos autores no sistema de maneira geral. Pois é uma linguagem simplificada, de modo que possibilita a qualquer pessoa ter uma compreensão do comportamento do mesmo (COCKBURN, 2005). A Figura 68 mostra o diagrama de casos de uso do sistema da Feira do Agricultor Familiar:

Figura 68 - Diagrama de Casos de Uso da Feira da Agricultura Familiar



Fonte: Imagem elaborada pelo autor.

5.6.2 Especificação dos casos de uso

O documento de Especificação de Casos de Uso é um registro formal que descreve um passo-a-passo para cada funcionalidade do software. Este guia contém o fluxo da interação entre usuário e software, desde o fluxo a sucesso do caso de uso, até como o sistema se comporta em situações alternativas (COCKBURN, 2005).

A seguir será apresentado o guia detalhado para cada caso de uso da aplicação.

5.6.2.1 Registrar Conta

Tabela 8 - Especificação do caso de uso: Registrar conta

Registrar Conta	
Finalidade/Objetivo:	Permite que o usuário forneça seus dados para que seja possível ter uma conta na plataforma.
Atores:	Cliente, Agricultores
Pré-Condições:	
Evento Inicial:	O sistema apresenta a tela de cadastro
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra tela de cadastro de clientes	
	2. O cliente informa seu nome completo
	3. O cliente informa seu telefone
	4. O cliente informa sua cidade (Cidade possíveis: Belém, Breu Branco, Cametá e Tucuruí)
	5. O cliente informa seu Endereço completo
	6. O cliente informa seu e-mail
	7. O cliente informa sua senha (E4, E5)

	8. O cliente seleciona o botão “Cadastrar”
8. O sistema registra as informações fornecidas pelo cliente	
9. O sistema envia um e-mail para a confirmação do cliente	
10. É incluído o caso de uso “ Confirmar Email ”	
11. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 2,3,4,5,6,7 do Fluxo Principal, caso algum desses dados não seja preenchido, o sistema mostra na tela que todos os dados são obrigatórios e devem ser preenchidos	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E2)	
Ações do Sistema	Ações do Ator
1. No passo 4 do Fluxo Principal, caso a cidade preenchida não seja permitida, o sistema mostra na tela que a cidade informada no formulário de	

cadastro pelo cliente não é válida	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E3)	
Ações do Sistema	Ações do Ator
1 . No passo 6 do Fluxo Principal, caso o e-mail informado seja inválido ou já esteja cadastrado, o sistema mostra na tela que o e-mail informado no formulário de cadastro pelo cliente não é válido ou já está cadastrado.	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E4)	
Ações do Sistema	Ações do Ator
1 . No passo 7 do Fluxo Principal, caso a senha informada tenha menos que 6 caracteres, o sistema mostra na tela que a senha informada no formulário de cadastro pelo cliente precisa ter no mínimo 6 caracteres	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E5)	

Ações do Sistema	Ações do Ator
1 . No passo 7 do Fluxo Principal, caso os campos de “senha” e “confirmação de senha” não coincidirem, o sistema mostra na tela que a senha informada no formulário de cadastro pelo cliente não coincide	
2. O sistema retorna ao passo 1º do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.2 Confirmar E-mail

Tabela 9 - Especificação do caso de uso: Confirmar E-mail

Confirmar E-mail	
Finalidade/Objetivo:	Verifica se o e-mail informado pelo usuário no cadastro é um e-mail válido e ativo
Atores:	Cliente, Agricultores
Pré-Condições:	Preencher o formulário da tela de cadastro
Evento Inicial:	Após preencher o formulário de cadastro, o sistema apresenta a tela que o e-mail deve ser confirmado.
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra tela com aviso que o cliente precisa confirmar seu e-mail	
	2. O cliente acessa sua caixa de entrada do e-mail e busca pelo e-mail “Confirmação de E-mail” do remetente : “contato.agriculturafamiliar@gmail.com”

	3. O cliente clica no link de confirmação
4. O sistema mostra tela com aviso de que o e-mail foi confirmado com sucesso	
5. O caso de uso é encerrado	
Fluxo Alternativo (A1)	
Ações do Sistema	Ações do Ator
1 . No passo 1 do Fluxo principal o sistema mostra na tela um aviso perguntando se o e-mail chegou, caso contrário, o usuário deve clicar no botão “Reenviar e-mail”	
	2. O usuário clica no botão “Reenviar e-mail”
	3. O cliente acessa sua caixa de entrada do e-mail e busca pelo e-mail “Confirmação de E-mail” do remetente : “contato.agriculturafamiliar@gmail.com”
	4. O cliente clica no link de confirmação
5. O sistema mostra tela com aviso de que o e-mail foi confirmado com sucesso	
6. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1 . No passo 4 do Fluxo Principal, caso o código de confirmação não	

seja válido, o sistema mostra na tela que o código de confirmação é inválido, ou seja, já foi confirmado ou não existe.	
2. O sistema mostra um botão para o usuário voltar para a página de login	
	3. O usuário clica no botão para voltar para a página de login
4. O usuário é redirecionado para a página de login	
5. O caso de uso é encerrado	

Fonte: Tabela elaborada pelo autor.

5.6.2.3 Realizar Login

Tabela 10 - Especificação do caso de uso: Realizar Login

Realizar Login	
Finalidade/Objetivo:	Verifica se o e-mail e senha informados pelo usuário correspondem à uma conta válida no sistema
Atores:	Cliente, Agricultores e Administradores
Pré-Condições:	Ter uma conta cadastrada e confirmada
Evento Inicial:	O sistema apresenta a tela de login
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de login	
	2. O usuário informa seu e-mail
	3. O usuário informa sua senha

	4. O usuário clica no botão de login
5. O sistema valida os dados	
6. O sistema redireciona o usuário para a página de login informando que o login foi efetuado com sucesso	
7. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 5 do Fluxo Principal, caso a conta ainda não esteja confirmada, é incluído o caso de Uso “Confirmar-Email”	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E2)	
Ações do Sistema	Ações do Ator
1. No passo 5 do Fluxo Principal, retorna uma mensagem informando que o e-mail ou senha é inválido.	
2. O sistema retorna ao passo 1º do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.4 Pesquisar Produtos

Tabela 11 - Especificação do caso de uso: Pesquisar Produtos

Pesquisar Produtos	
Finalidade/Objetivo:	Permite que o usuário forneça pesquisa por determinados produtos
Atores:	Cliente, Agricultores e Administradores
Pré-Condições:	
Evento Inicial:	O sistema apresenta a tela da feira, onde possui a barra de pesquisa.
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela da feira, onde possui a barra de busca.	
	2. O usuário informa o produto que está buscando
	3. Após informar o nome o usuário clica no botão da lupa para realizar a busca.
4. O sistema recebe os dados de busca e processa os dados de busca	
5. O sistema mostra a tela de resultados da busca.	
6. O caso de uso de é encerrado.	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 4 do Fluxo Principal, o sistema retorna uma mensagem informando que não	

foi possível realizar a busca, pois a string de busca é vazia ou não existe	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E2)	
Ações do Sistema	Ações do Ator
1. No passo 5 do Fluxo Principal, o sistema retorna uma mensagem informando que: "Não foi encontrado nenhum agricultor vendendo este produto"	
2. O sistema retorna ao passo 1º do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.5 Realizar Pedidos

Tabela 12 - Especificação do caso de uso: Realizar Pedidos

Realizar Pedidos	
Finalidade/Objetivo:	Permite que os usuários façam compras na plataforma
Atores:	Clientes e Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor ou Cliente cadastrada na plataforma.
Evento Inicial:	O sistema apresenta a tela da feira, onde possui o perfil de cada agricultor
Fluxo Principal	

Ações do Sistema	Ações do Ator
1. O sistema mostra a tela da feira, onde estão listados todos os agricultores.	
	2. O usuário escolhe um agricultor
	3. Após escolher o agricultor, ele clica no botão "Produtos"
4. O sistema irá mostrar o perfil daquele agricultor, mostrando detalhes de nome completo, endereço e telefone. Além disso, é mostrado todos os anúncios de produtos do agricultor	
	5. O usuário escolhe algum produto
	6. Após escolher algum produto, ele clica no botão de "Comprar".
7. O sistema mostra a tela do carrinho de compras, com o pedido adicionado e com informações de: "Valor por unidade", "Subtotal", "Quantidade", e os botões de: "Adicionar", "Remover" e "Continuar comprando"	
	8. Após analisar todas as informações, o usuário clica no botão de "Prosseguir"
9. O sistema mostra a tela com informações do	

frete. Obs. Frete padrão de R\$ 5,00 reais.	
	10. Após analisar todas as informações, o usuário clica no botão de “Prosseguir”
11. O sistema mostra na tela informações de formas de pagamento: “Cartão de Débito, Cartão de Crédito, Pix ou Dinheiro”	
	12. Após escolher alguma das formas de pagamento, o usuário clica no botão de “Finalizar Pedido”
13. O sistema mostra a tela da feira com uma mensagem: “Pedido realizado com sucesso. ”	
14. O caso de uso é encerrado.	
Fluxo Alternativo (A1)	
Ações do Sistema	Ações do Ator
1 . No passo 7 do Fluxo principal o sistema mostra a tela do carrinho de compras, com o pedido adicionado e com informações de: “Valor por unidade”, “Subtotal”, “Quantidade”, e os botões de: “Adicionar”, “Remover” e “Continuar comprando”	

	2. O usuário clica no botão de “Continuar comprando”
3. O sistema retorna ao ao 1º Passo do fluxo principal	
Fluxo Alternativo (A2)	
Ações do Sistema	Ações do Ator
1 . No passo 7 do Fluxo principal o sistema mostra a tela do carrinho de compras, com o pedido adicionado e com informações de: “Valor por unidade”, “Subtotal”, “Quantidade”, e os botões de: “Adicionar”, “Remover” e “Continuar comprando”	
	2. O usuário clica no botão de “Adicionar”
3. O sistema mostra na tela o acréscimo de unidade e valor daquele produto	
4. O sistema retorna ao passo 7 do fluxo principal.	
Fluxo Alternativo (A3)	
Ações do Sistema	Ações do Ator
1 . No passo 7 do Fluxo principal o sistema mostra a tela do carrinho de compras, com o	

pedido adicionado e com informações de: “Valor por unidade”, “Subtotal”, “Quantidade”, e os botões de: “Adicionar”, “Remover” e “Continuar comprando”	
	2. O usuário clica no botão de “Remover”
3. O sistema mostra na a remoção daquele produto	
4. O sistema retorna ao passo 7 do fluxo principal.	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 3 e 6 do Fluxo Principal, o sistema retorna uma mensagem informando: “Não encontrado ou inválido” caso algum identificador (Agricultor ou Produto) informado seja desconhecido	
2. O sistema retorna ao passo 1º do fluxo principal	
Fluxo Exceção (E2)	
Ações do Sistema	Ações do Ator
1. No passo 11 do	

Fluxo Principal, o sistema retorna uma mensagem informando: “Método de pagamento inválido” método de pagamento seja informado de forma inválida	
2. O sistema retorna ao passo 11 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.6 Consultar Pedidos

Tabela 13 - Especificação do caso de uso: Consultar Pedidos

Consultar Pedidos	
Finalidade/Objetivo:	Permite que os usuários consultem seus pedidos feitos na plataforma
Atores:	Clientes e Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor ou Cliente cadastrada na plataforma.
Evento Inicial:	O sistema apresenta a tela de “Meus Pedidos”, onde há informações de cada pedido
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema apresenta a tela de “Meus Pedidos”, onde há informações de: Número de pedido, data do pedido, Vendedor e situação de cada pedido. Além disso, cada pedido há um botão de “Ver detalhes”	
	2. O usuário verifica os pedidos e escolhe um para ver detalhes.

	3. O usuário clica no botão “ver detalhes”
4. O sistema apresenta a tela de “Detalhes do Pedidos”, tela responsável por mostrar detalhes da compra: produtos, total, quantidade e etc. Além disso, nesta tela há botões de “Cancelar pedido” e “Baixar PDF”	
5. O caso de uso é encerrado.	
Fluxo Alternativo (A1)	
Ações do Sistema	Ações do Ator
1 . No passo 4 do fluxo principal há dois botões “Cancelar Pedido” e “Baixar PDF”	
	2. O usuário clica no botão de “Cancelar Pedido”
3. É incluído o caso de uso “Cancelar pedidos”	
Fluxo Alternativo (A2)	
Ações do Sistema	Ações do Ator
1 . No passo 4 do fluxo principal há dois botões “Cancelar Pedido” e “Baixar PDF”	
	2. O usuário clica no botão de “Baixar PDF”
3. O sistema abre uma janela solicitando	

que o usuário informe onde ele gostaria de salvar este PDF	
	4. O usuário informa o local e clica no botão "OK"
5. O PDF é baixado e o sistema retorna ao passo 4 do fluxo principal.	

Fonte: Tabela elaborada pelo autor.

5.6.2.7 Cancelar Pedidos

Tabela 14 - Especificação do caso de uso: Cancelar Pedidos

Cancelar Pedidos	
Finalidade/Objetivo:	Permite que os usuários cancelem seus pedidos feitos na plataforma
Atores:	Clientes e Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor ou Cliente cadastrada na plataforma e ter feito algum pedido.
Evento Inicial:	O sistema apresenta a tela "Detalhes do pedido", onde possui o botão de cancelar pedido
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema apresenta a tela "Detalhes do pedido", onde possui o botão de cancelar pedido	
	2. O usuário verificar qual pedido ele quer cancelar e clica no botão de "Cancelar pedido"
	3. O usuário clica no botão "ver detalhes"
4. O sistema apresenta a tela de	

“Meus pedidos” informando que o pedido foi cancelado.	
5. O caso de uso é encerrado.	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 2 do Fluxo Principal, o sistema retorna uma mensagem informando: “pedido inválido” caso o usuário esteja tentando cancelar um pedido que não é seu ou inexistente.	
2. O sistema retorna ao passo 1 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.8 Editar Dados

Tabela 15 - Especificação do caso de uso: Editar dados

Editar Dados	
Finalidade/Objetivo:	Permite que os usuários editem seus dados na plataforma
Atores:	Clientes e Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor ou Cliente cadastrada na plataforma e ter feito algum pedido.
Evento Inicial:	O sistema apresenta a tela “Meu perfil”, com os dados de Nome completo, Telefone, Endereço ou Foto de perfil
Fluxo Principal	
Ações do Sistema	Ações do Ator

1. O sistema apresenta a tela “Meu perfil”, com os dados de Nome completo, Endereço ou Foto de perfil	
	2. O usuário deve inserir os dados que ele quer atualizar nos campos de: “Nome completo”, “Telefone” e “Rua”, “Cidade”, “Bairro” e “Número”
	3. Após preencher o usuário deve clicar no botão de “Atualizar”
4. O sistema apresenta a tela de “Meu perfil” informando que os dados foram atualizados com sucesso	
5. O caso de uso é encerrado.	

Fluxo Alternativo (A1)

Ações do Sistema	Ações do Ator
1. No passo 1 do fluxo principal o sistema mostrará os dados para serem atualizados, porém, se o usuário for do Tipo Agricultor, aparecerá um campo extra, campo de alterar foto.	
	2. O usuário clica no botão de “Selecionar foto”.
3. O sistema mostra uma janela para selecionar a foto	
	4. Após selecionar a foto, o usuário clica no

	botão “Enviar foto”;
4. O sistema mostra uma mensagem informando que a foto foi enviada com sucesso.	
5. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 2 do Fluxo Principal, o sistema retorna uma mensagem informando: “dados inválidos ou vazios” caso o usuário esteja tentando atualizar para um valor inválido ou vazio.	
2. O sistema retorna ao passo 1 do fluxo principal	
Fluxo Exceção (E2)	
Ações do Sistema	Ações do Ator
1. No passo 4 do Fluxo Alternativo (A1), o sistema retorna uma mensagem informando: “Arquivo enviado inválido” caso o usuário esteja tentando enviar algum arquivo que não seja uma imagem.	
2. O sistema retorna ao passo 1 do fluxo alternativo (a1)	

Fonte: Tabela elaborada pelo autor.

5.6.2.9 Anunciar Produtos

Tabela 16 - Especificação do caso de uso: Anunciar Produtos

Anunciar Produtos	
Finalidade/Objetivo:	Permite que os agricultores anuncie seus produtos que estão à venda
Atores:	Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor na plataforma
Evento Inicial:	O sistema apresenta a tela de produtos que o agricultor pode vender
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de produtos, onde estão listados todos os produtos que o agricultor pode vender. Cada um desses produtos é necessário informar: preço e descrição. Ex. R\$ 5,00, Meia dúzia	
	2. O agricultor escolhe o produto
	3. O agricultor define o preço do produto
	4. O agricultor define a descrição do produto
	5. O agricultor aperta no botão de adicionar
6. O sistema mostra a tela com todos os anúncios ativos do agricultor	
7. O caso de uso é encerrado	

Fonte: Tabela elaborada pelo autor.

5.6.2.10 Editar Anúncios

Tabela 17 - Especificação do caso de uso: Editar Anúncios

Editar Anúncios	
Finalidade/Objetivo:	Permite que os agricultores editem os seus anúncios.
Atores:	Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor na plataforma
Evento Inicial:	O sistema apresenta a tela de anúncios ativos do agricultor
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de “Meus anúncios” que diz respeito aos anúncios ativos do agricultor. Para cada anúncio há o botão de “Editar anúncio”	
	2. O agricultor escolhe o anúncio que deseja editar
	3. O agricultor clica no botão de “Editar anúncio”
4. O sistema redireciona para a página “Editar anúncio”	
	5. O agricultor deve definir o preço do produto e a descrição do mesmo
	6. Após definir o agricultor clica no botão de “Atualizar”
7. O sistema redireciona o usuário para a página “Meus anúncios” com uma mensagem informando que o anúncio foi editado com	

sucesso.	
8. O caso de uso é encerrado.	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 6 do Fluxo Principal, se o usuário adicionar nos campos de Preço e descrição valores inválidos, o sistema retorna uma mensagem informando que os valores estão inválidos.	
2. O sistema retorna ao passo 1 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.11 Excluir Anúncios

Tabela 18 - Especificação do caso de uso: Excluir Anúncios

Excluir Anúncios	
Finalidade/Objetivo:	Permite que os agricultores excluam seus anúncios de produtos
Atores:	Agricultores
Pré-Condições:	O usuário precisa ter uma conta do tipo Agricultor na plataforma e ter algum anúncio disponível
Evento Inicial:	O sistema apresenta a tela de anúncios ativos do agricultor
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de “Meus anúncios” que diz respeito aos anúncios ativos do	

agricultor. Para cada anúncio há o botão de “Excluir anúncio”	
	2. O agricultor escolhe o anúncio que deseja excluir
	3. O agricultor clica no botão de “Excluir anúncio”
6. O sistema exclui o anúncio e informa uma mensagem dizendo que o anúncio foi excluído com sucesso.	
7. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 2 do Fluxo Principal, o sistema retorna uma mensagem informando: “anúncio inválido” caso o usuário esteja tentando cancelar um pedido que não é seu ou inexistente.	
2. O sistema retorna ao passo 1 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.12 Adicionar Agricultor

Tabela 19 - Especificação do caso de uso: Adicionar Agricultor

Adicionar Agricultor	
Finalidade/Objetivo:	Permite que os administradores tornem usuários em agricultor
Atores:	Administrador

Pré-Condições:	O usuário precisa ter uma conta do tipo Administrador e precisa ter o e-mail do cliente alvo.
Evento Inicial:	O sistema apresenta a tela de administração
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de “Administração”, onde há um barra de pesquisa para buscar agricultor através do e-mail	
	2. O administrador informa o e-mail do cliente que deseja ser agricultor
	3. O administrador clica no botão de busca.
4. O sistema retorna o usuário a qual pertence aquele e-mail informado	
	5. O administrador clica no botão “Tornar agricultor” ao lado do nome listado
6. O sistema retorna uma mensagem informando que o usuário foi habilitado para ser um agricultor na plataforma	
7. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 4, caso o sistema não encontre nenhum usuário com e-mail informado, será retornada uma mensagem informando que não foi	

encontrado nenhum usuário vinculado com tal e-mail.	
2. O sistema retorna ao passo 1 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.13 Excluir Agricultor

Tabela 20 - Especificação do caso de uso: Excluir Agricultor

Excluir Agricultor	
Finalidade/Objetivo:	Permite que os administradores retirem os privilégios de agricultor de usuários
Atores:	Administrador
Pré-Condições:	O usuário precisa ter uma conta do tipo Administrador e precisa ter o e-mail do cliente alvo.
Evento Inicial:	O sistema apresenta a tela de administração
Fluxo Principal	
Ações do Sistema	Ações do Ator
1. O sistema mostra a tela de "Administração", onde há um barra de pesquisa para buscar agricultor através do e-mail	
	2. O administrador informa o e-mail do cliente que deseja ser agricultor
	3. O administrador clica no botão de busca.
4. O sistema retorna o usuário a qual pertence aquele e-mail informado	
	5. O administrador clica no botão "Remover agricultor" ao lado do nome listado

6. O sistema retorna uma mensagem informando que o agricultor foi removido e agora terá apenas privilégios de clientes	
7. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 4, caso o sistema não encontre nenhum usuário com e-mail informado, será retornada uma mensagem informando que não foi encontrado nenhum usuário vinculado com tal e-mail.	
2. O sistema retorna ao passo 1 do fluxo principal	

Fonte: Tabela elaborada pelo autor.

5.6.2.14 Gerar Relatório

Tabela 21 - Especificação do caso de uso: Gerar Relatório

Gerar Relatório	
Finalidade/Objetivo:	Permite que os administradores gerem relatórios com base nas vendas da sua cidade
Atores:	Administrador
Pré-Condições:	O usuário precisa ter uma conta do tipo Administrador
Evento Inicial:	O sistema apresenta a tela de Acompanhamento
Fluxo Principal	

Ações do Sistema	Ações do Ator
1. O sistema apresenta a tela de Acompanhamento, onde é possível analisar os pedidos realizados. Além disso, é mostrado um botão de gerar relatório.	
	2. O administrador informa qual o período que ele deseja ter o relatório. Ex. últimos 7 dias, último mês e etc
	3. Após isto, ele clica no botão gerar relatório.
4. O sistema retorna uma janela para o usuário indicar onde será baixado o arquivo	
	5. O usuário informa o local e aperta no botão de Salvar
6. O sistema retorna uma mensagem que o relatório foi realizado com sucesso	
7. O caso de uso é encerrado	
Fluxo Exceção (E1)	
Ações do Sistema	Ações do Ator
1. No passo 6, caso haja algum problema com o relatório será informado para tentar gerar o relatório mais tarde pois algum erro inesperado aconteceu.	
2. O sistema retorna ao passo 1 do fluxo principal	

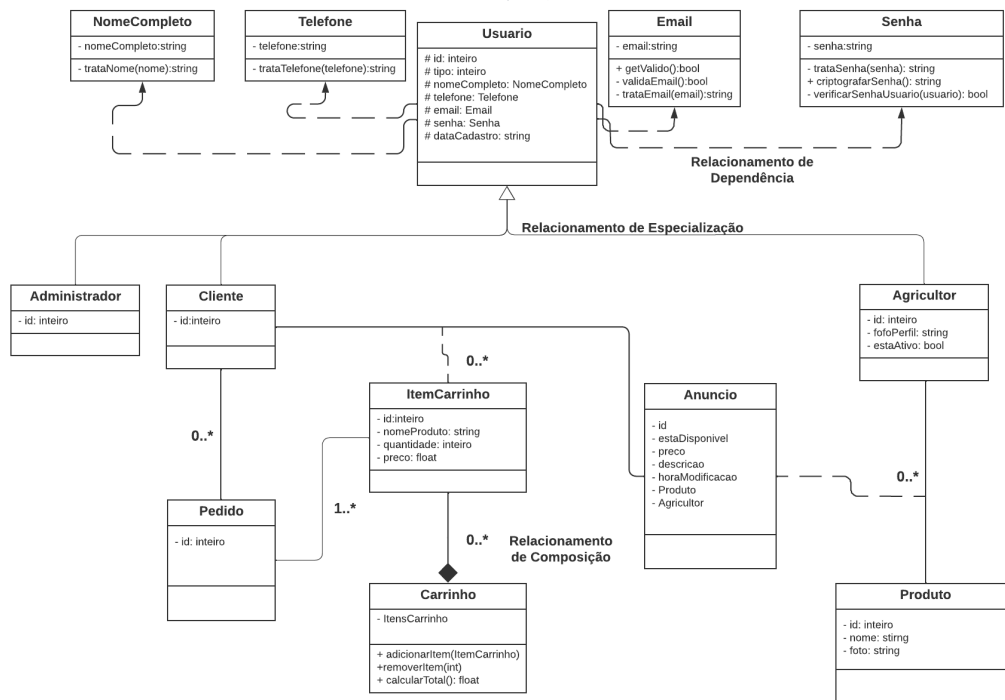
Fonte: Tabela elaborada pelo autor.

5.6.3 Diagrama de Classes

Os diagramas de classe são utilizados para representar um sistema orientado a objetos, por meio de representações UML. Esses diagramas auxiliam o time de desenvolvimento a entender a arquitetura do sistema, identificar suas principais classes, atributos e métodos, e como elas se relacionam umas com as outras, facilitando assim o processo de design e implementação do software (SOMMERVILLE, 2011). A seguir é apresentada a modelagem de todas as classes da aplicação.

5.6.3.1 Diagrama de Classes de Modelo

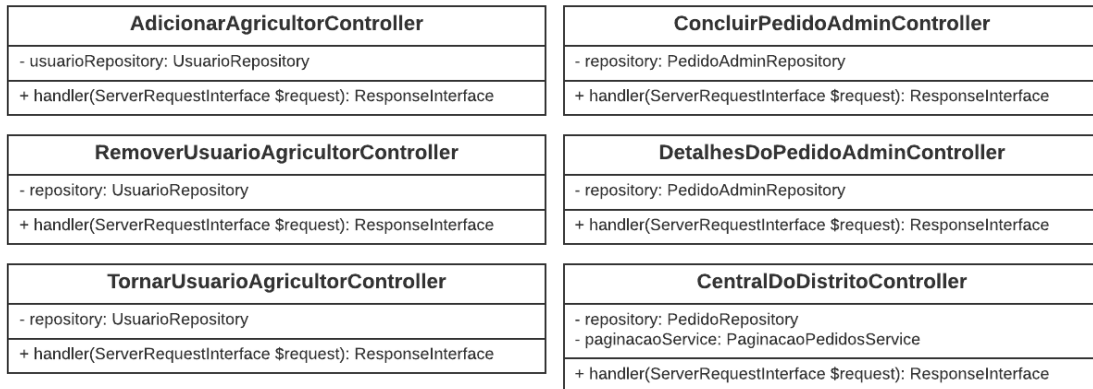
Figura 69 - Diagrama de Classes de Modelo (Domínio)



Fonte: Imagem elaborada pelo autor.

5.6.3.2 Diagrama de Classes de Controllers do pacote Admin

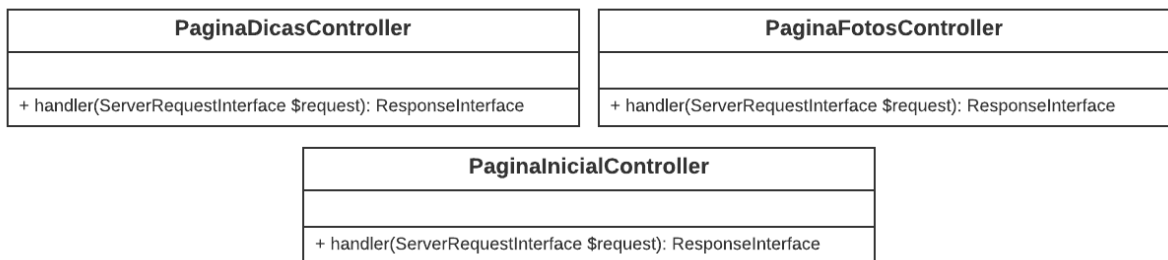
Figura 70 - Diagrama de Classes dos Controllers do pacote Admin



Fonte: Imagem elaborada pelo autor.

5.6.3.3 Diagrama de Classes de Controllers do pacote Blog

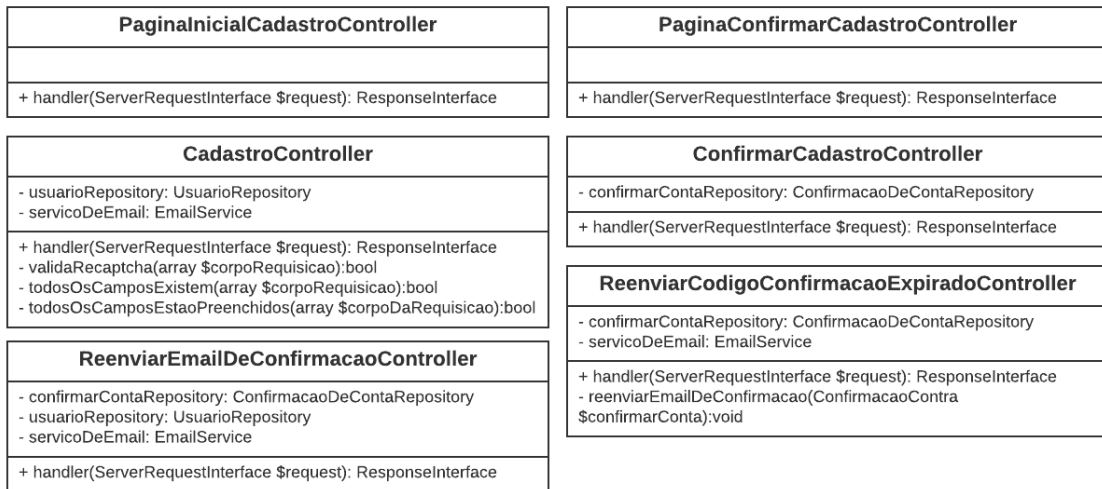
Figura 71 - Diagrama de Classes dos Controllers do pacote Blog



Fonte: Imagem elaborada pelo autor.

5.6.3.4 Diagrama de Classes de Controllers do pacote Cadastro

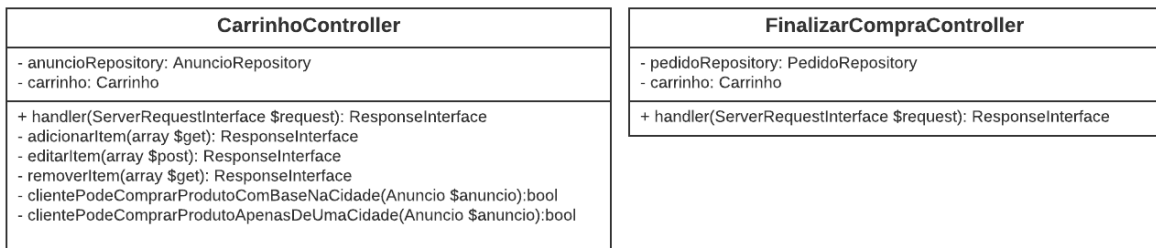
Figura 72 - Diagrama de Classes dos Controllers do pacote Cadastro



Fonte: Imagem elaborada pelo autor.

5.6.3.5 Diagrama de Classes de Controllers do pacote Carrinho

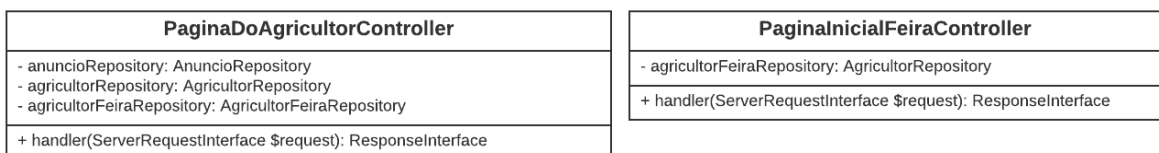
Figura 73 - Diagrama de Classes dos Controllers do pacote Carrinho



Fonte: Imagem elaborada pelo autor.

5.6.3.6 Diagrama de Classes de Controllers do pacote Feira

Figura 74 - Diagrama de Classes dos Controllers do pacote Feira



Fonte: Imagem elaborada pelo autor.

5.6.3.7 Diagrama de Classes de Controllers do pacote Frete

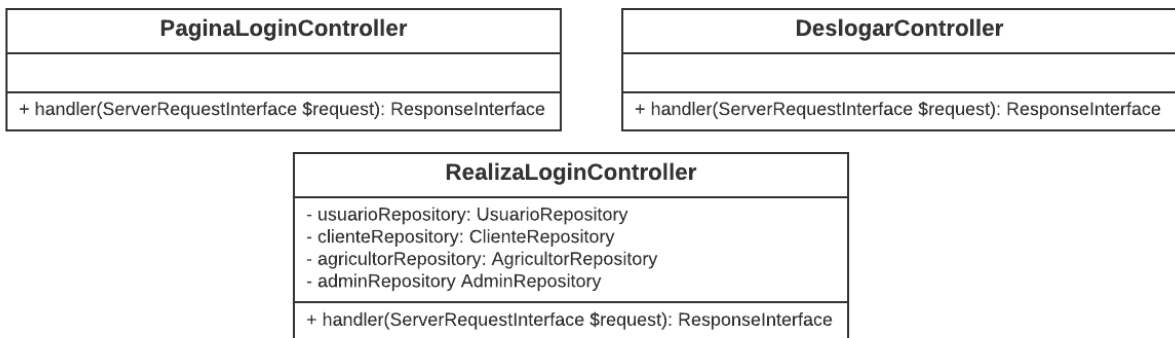
Figura 75 - Diagrama de Classes dos Controllers do pacote Frete



Fonte: Imagem elaborada pelo autor.

5.6.3.8 Diagrama de Classes de Controllers do pacote Login

Figura 76 - Diagrama de Classes dos Controllers do pacote Login



Fonte: Imagem elaborada pelo autor.

5.6.3.9 Diagrama de Classes de Controllers do pacote Pagamento

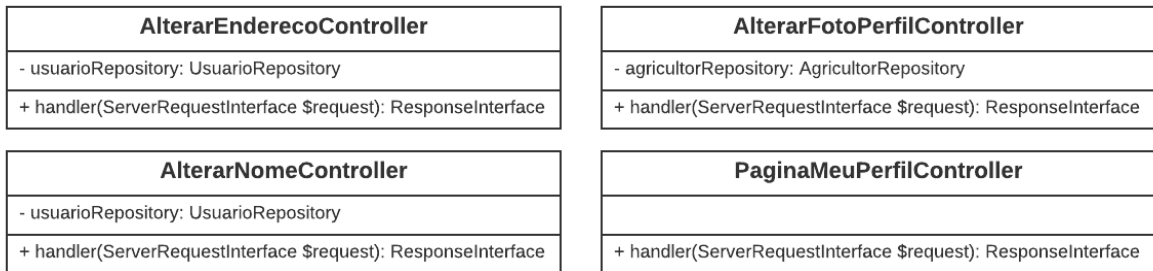
Figura 77 - Diagrama de Classes dos Controllers do pacote Pagamento



Fonte: Imagem elaborada pelo autor.

5.6.3.10 Diagrama de Classes de Controllers do pacote Meu Perfil

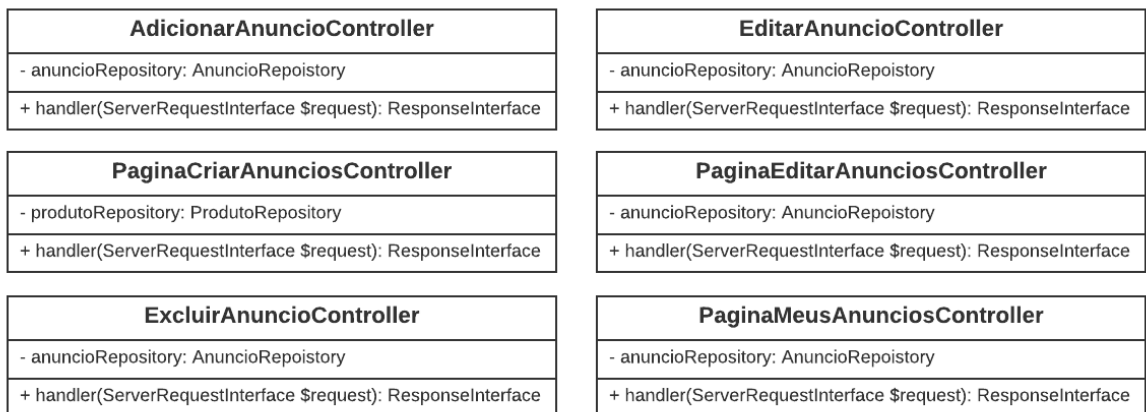
Figura 78 - Diagrama de Classes dos Controllers do pacote Meu Perfil



Fonte: Imagem elaborada pelo autor.

5.6.3.11 Diagrama de Classes de Controllers do pacote Anuncio

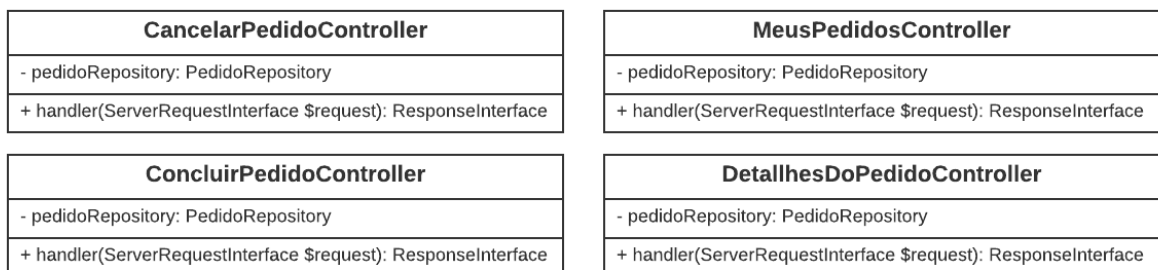
Figura 79 - Diagrama de Classes dos Controllers do pacote Anuncio



Fonte: Imagem elaborada pelo autor.

5.6.3.12 Diagrama de Classes de Controllers do pacote Pedidos

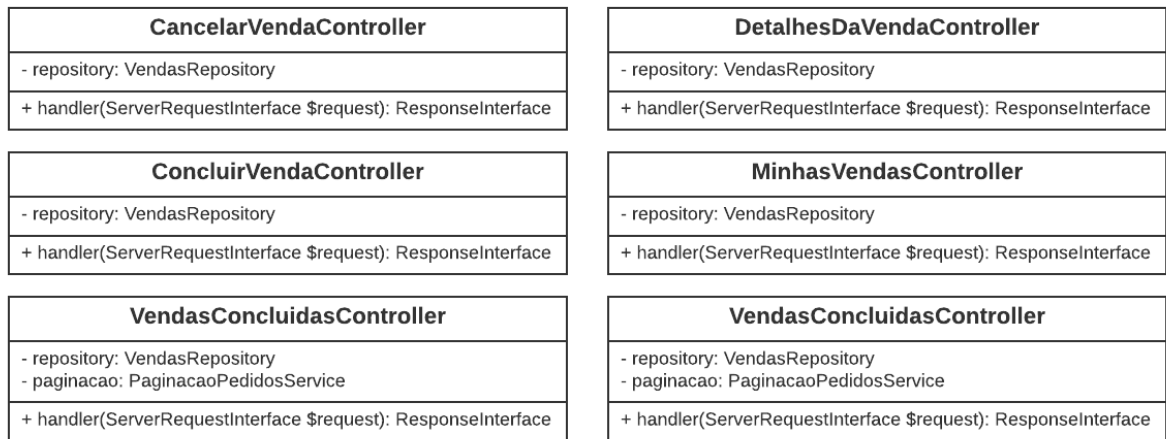
Figura 80 - Diagrama de Classes dos Controllers do pacote Pedidos



Fonte: Imagem elaborada pelo autor.

5.6.3.13 Diagrama de Classes de Controllers do pacote Vendas

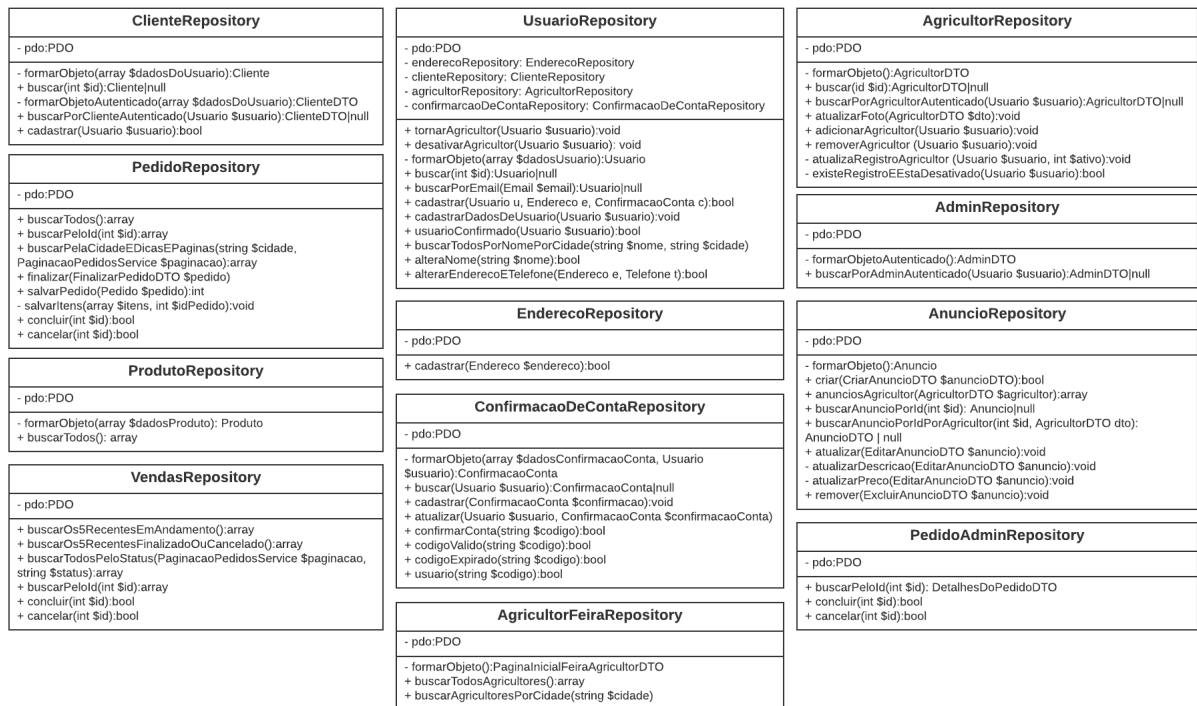
Figura 81 - Diagrama de Classes dos Controllers do pacote Vendas



Fonte: Imagem elaborada pelo autor.

5.6.3.14 Diagrama de Classes do pacote Repositories

Figura 82 - Diagrama de Classes do pacote Repositories



Fonte: Imagem elaborada pelo autor.

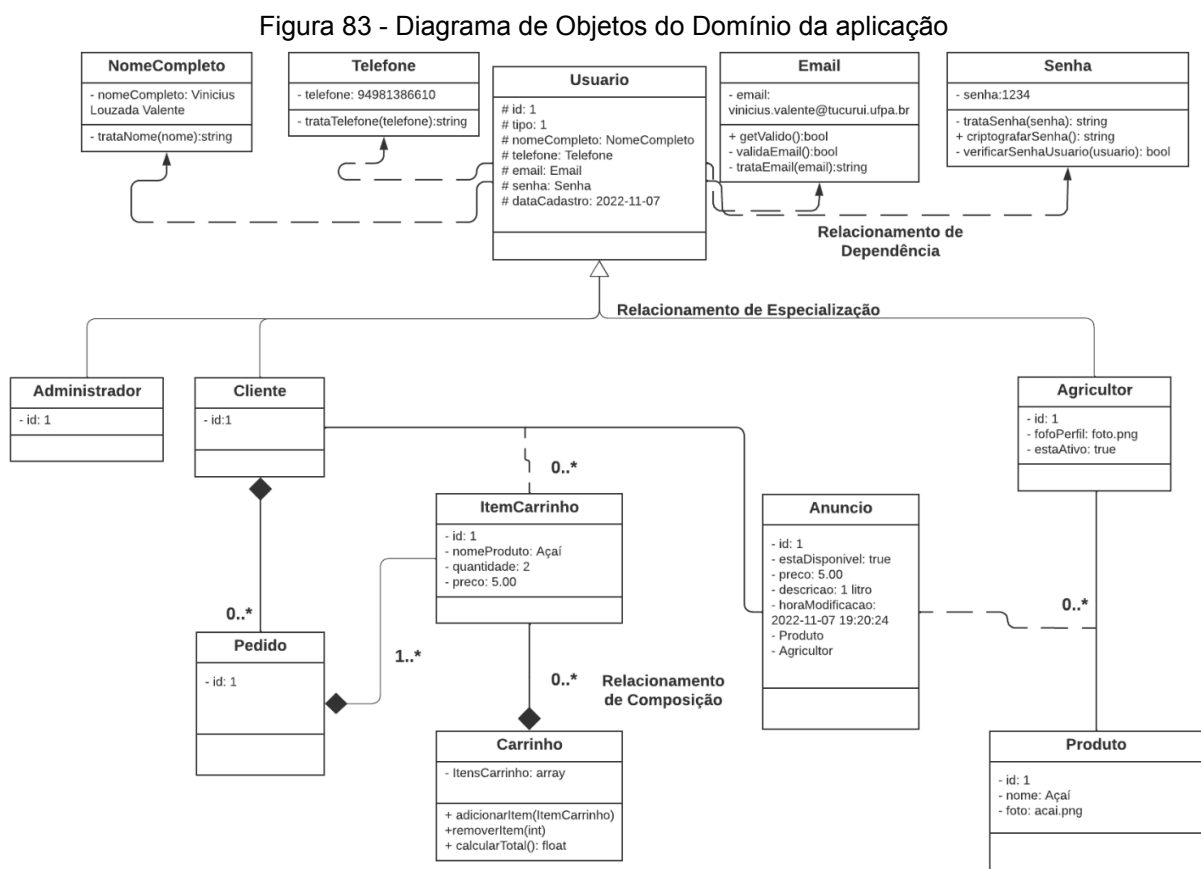
5.6.4 Diagrama de Objetos

O diagrama de objetos é uma representação visual das instâncias de classes em um determinado momento durante a execução de um sistema orientado a

objetos. Esses diagramas são úteis para entender como as classes interagem em tempo de execução, mostrando quais objetos existem, seus estados e como se comunicam uns com os outros para realizar determinadas tarefas (SOMMERVILLE, 2011).

A seguir é exemplificado as instâncias da classe de Modelo em tempo de execução do sistema.

5.6.4.1 Diagrama de Objetos de Modelo



Fonte: Imagem elaborada pelo autor.

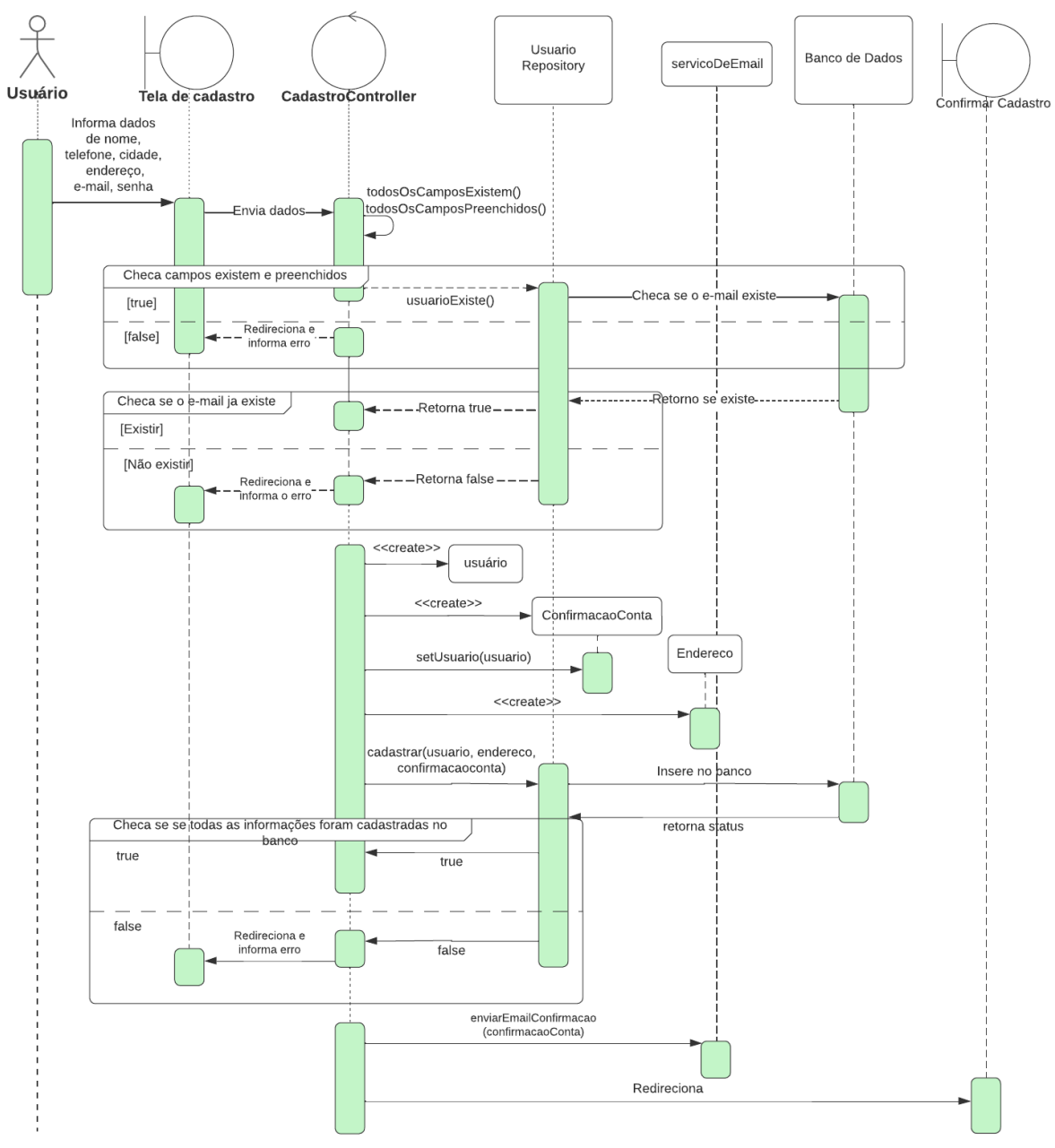
5.6.5 Diagrama de Sequência

Os diagramas de sequência, são usados para modelar e especificar o fluxo de interação entre os componentes do sistema (classes, objetos, banco de dados e serviços) e os atores. Este tipo de documento possui uma riqueza de detalhes, pois expõe cada detalhe do funcionamento do caso de uso em questão (SOMMERVILLE, 2011).

A seguir será apresentado os diagramas de seqüência para cada caso de uso da aplicação.

5.6.5.1 Registrar Conta

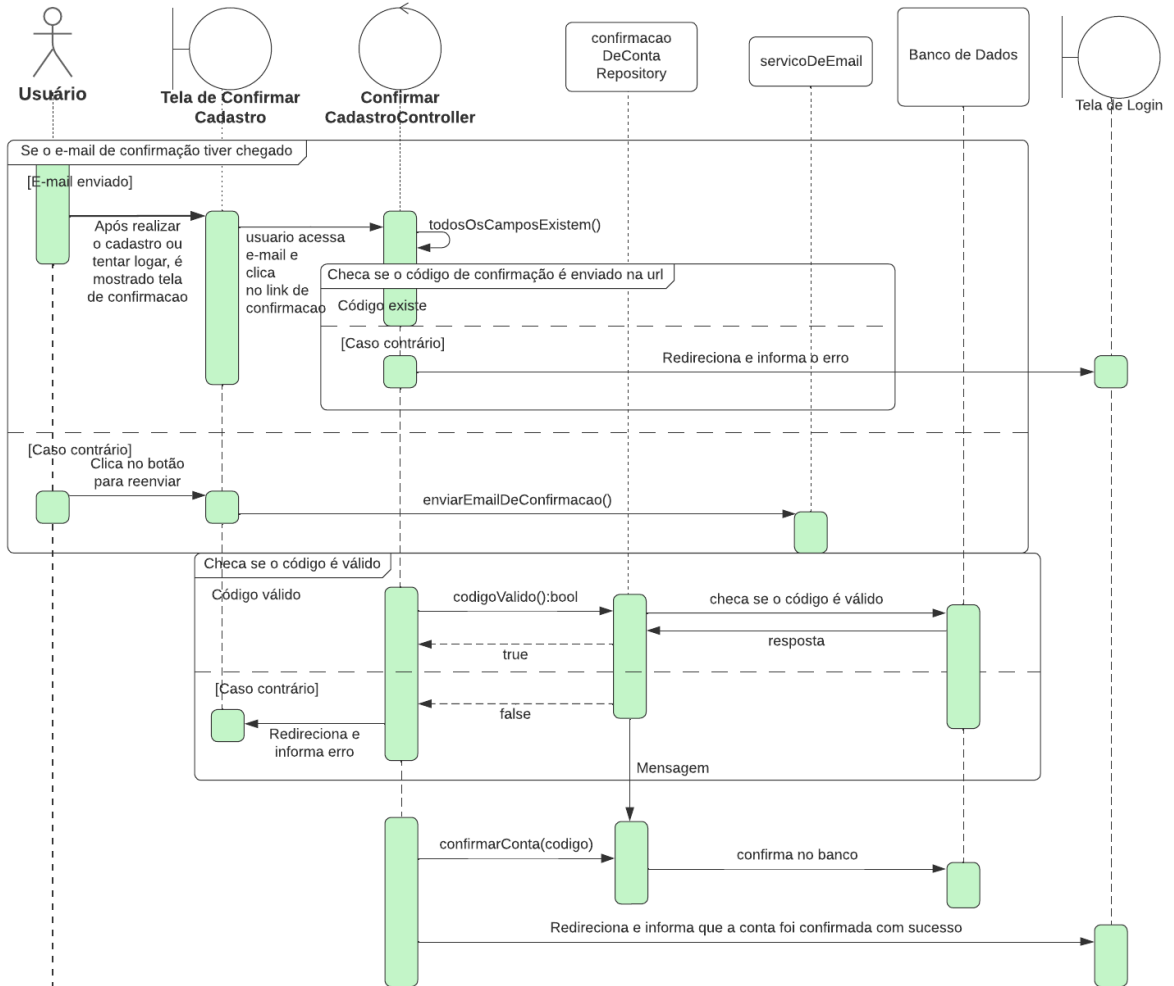
Figura 84 - Diagrama de Sequência do Caso de Uso: Registrar Conta



Fonte: Imagem elaborada pelo autor

5.6.5.2 Confirmar E-mail

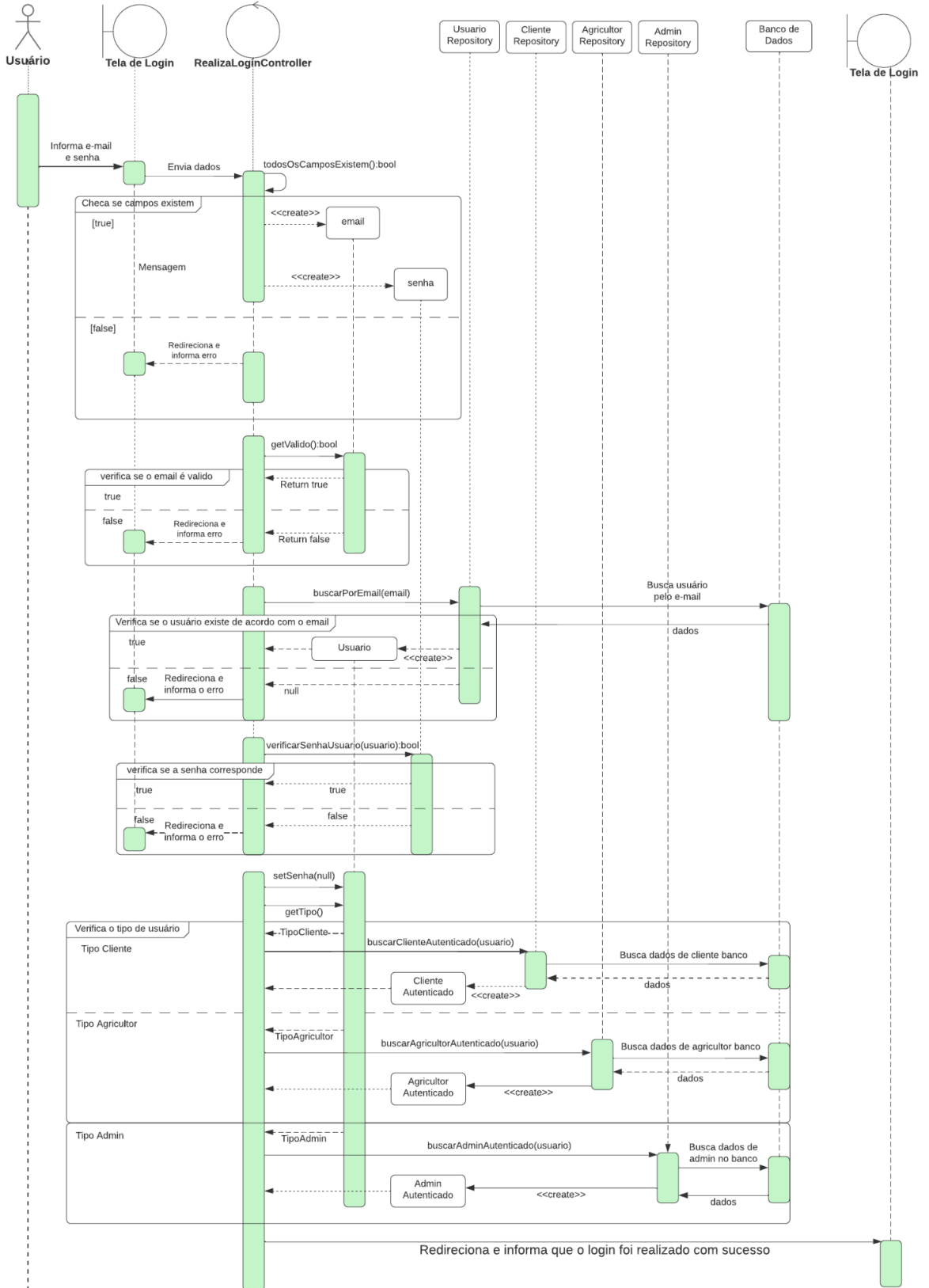
Figura 85 - Diagrama de Sequência do Caso de Uso: Confirmar E-mail



Fonte: Imagem elaborada pelo autor.

5.6.5.3 Realizar Login

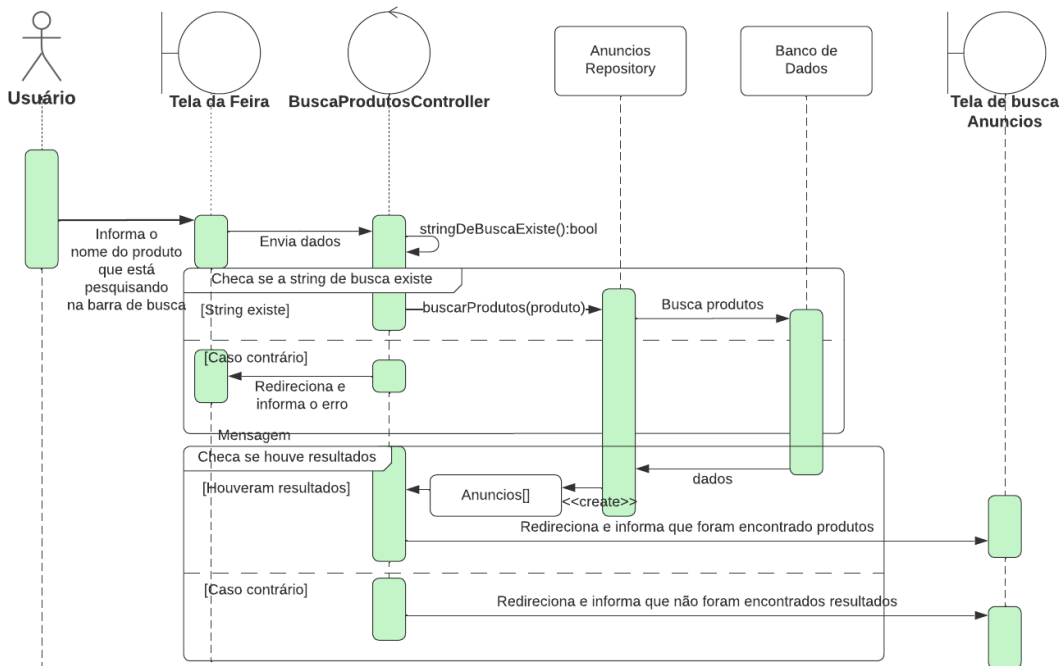
Figura 86 - Diagrama de Sequência do Caso de Uso: Realizar Login



Fonte: Imagem elaborada pelo autor.

5.6.5.4 Pesquisar Produto

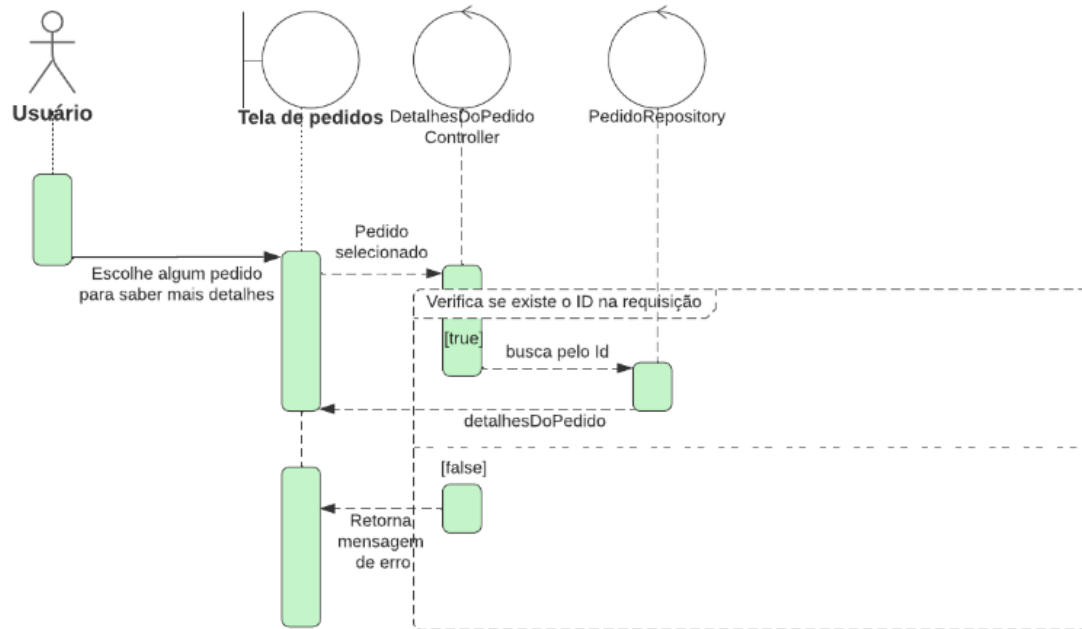
Figura 87 - Diagrama de Sequência do Caso de Uso: Pesquisar Produto



Fonte: Imagem elaborada pelo autor.

5.6.5.6 Consultar Pedido

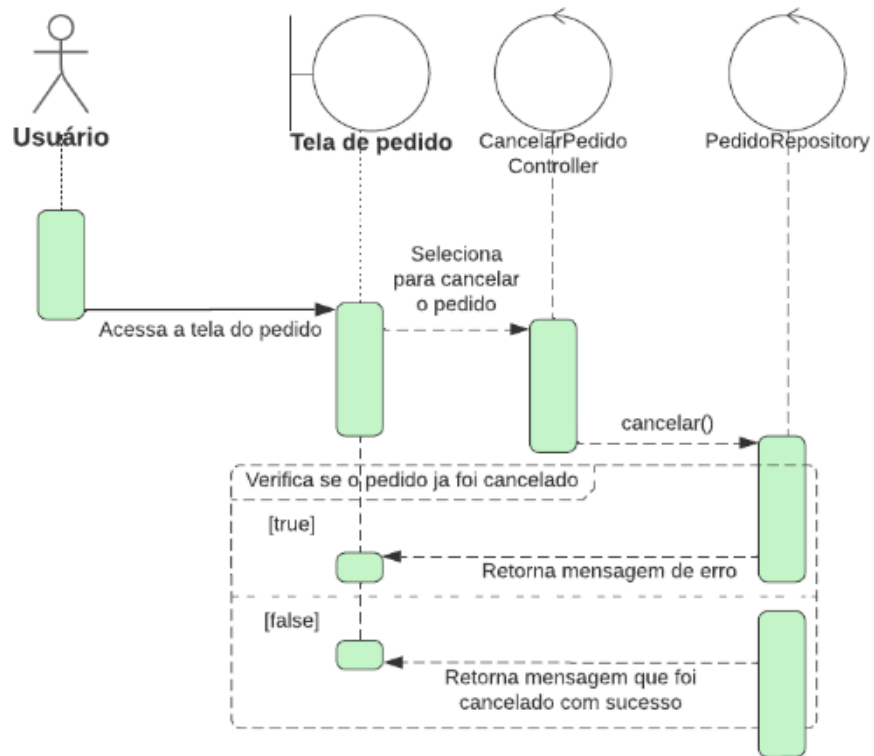
Figura 89 - Diagrama de Sequência do Caso de Uso: Consultar Pedido



Fonte: Imagem elaborada pelo autor.

5.6.5.7 Cancelar Pedido

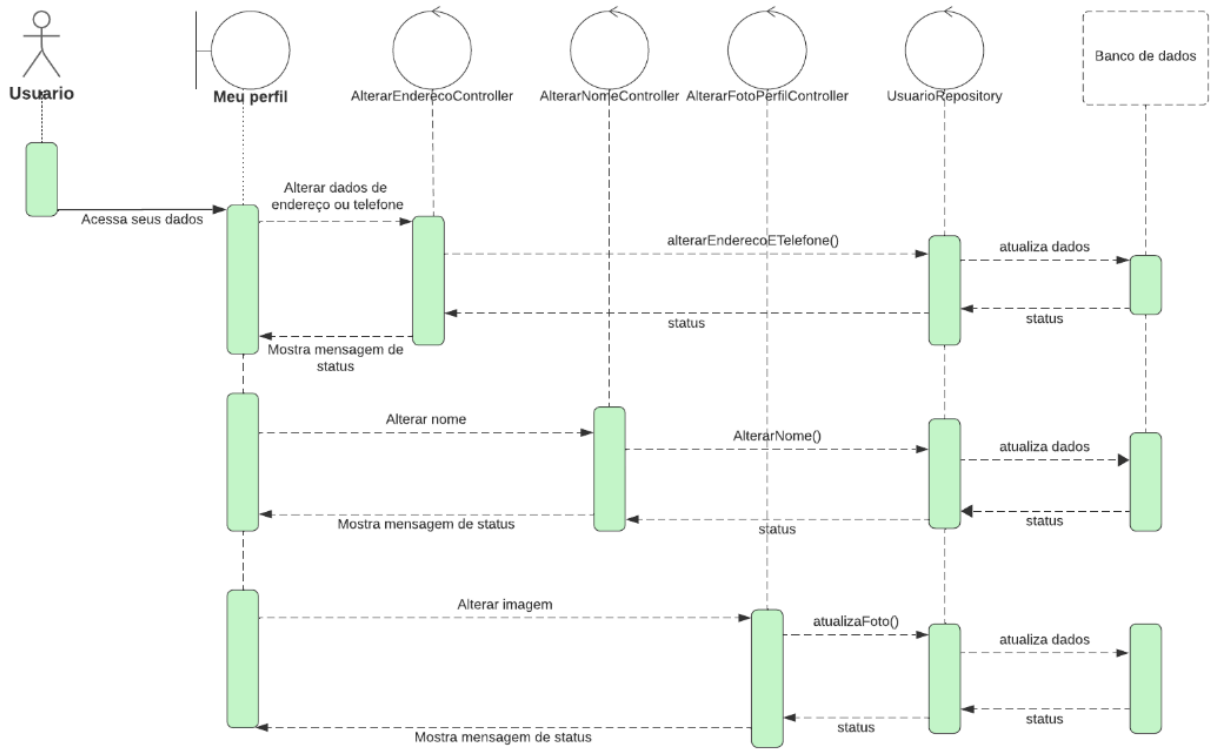
Figura 90 - Diagrama de Sequência do Caso de Uso: Cancelar Pedido



Fonte: Imagem elaborada pelo autor.

5.6.5.8 Editar Dados

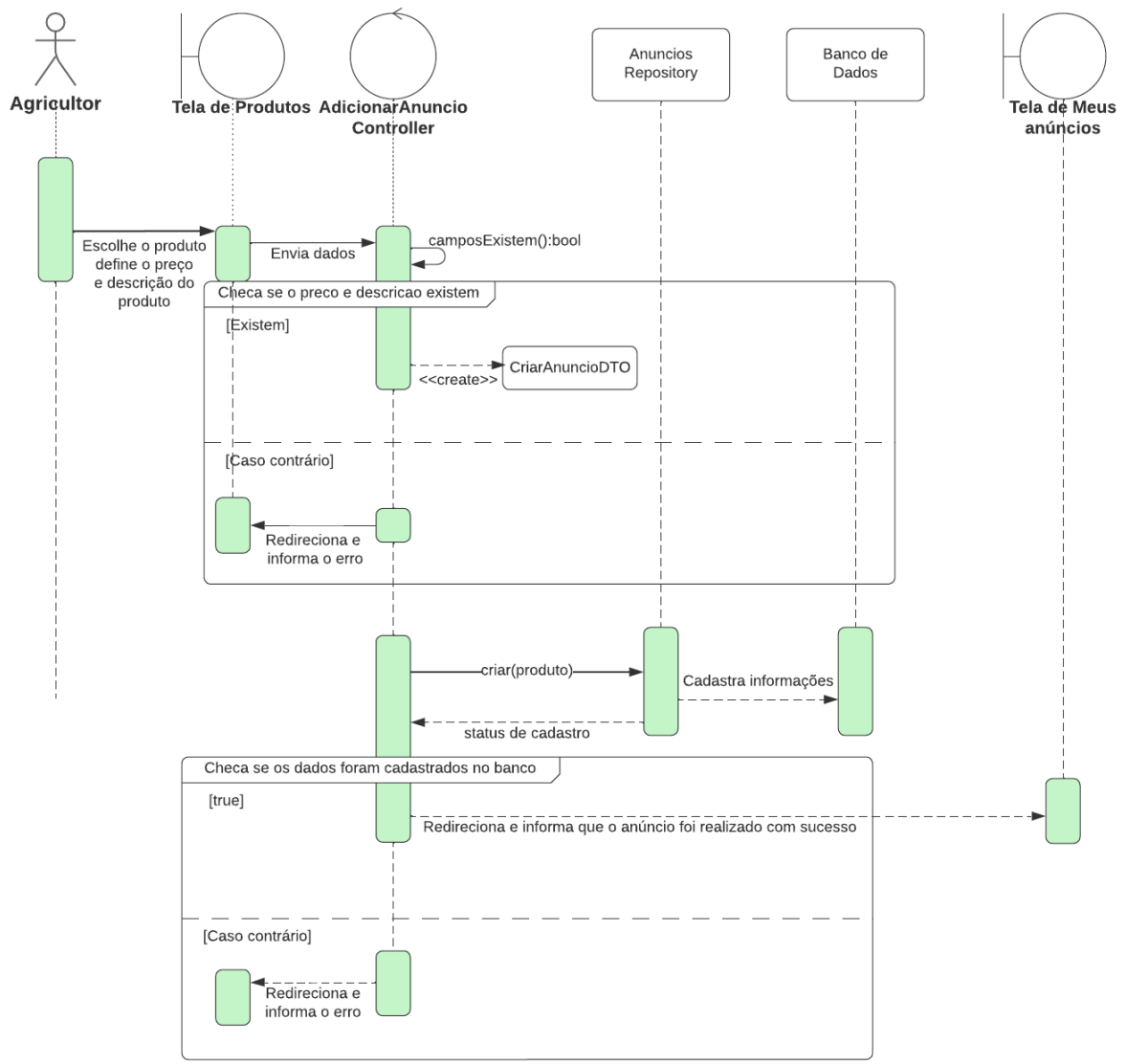
Figura 91 - Diagrama de Sequência do Caso de Uso: Editar Dados



Fonte: Imagem elaborada pelo autor.

5.6.5.9 Anunciar Produtos

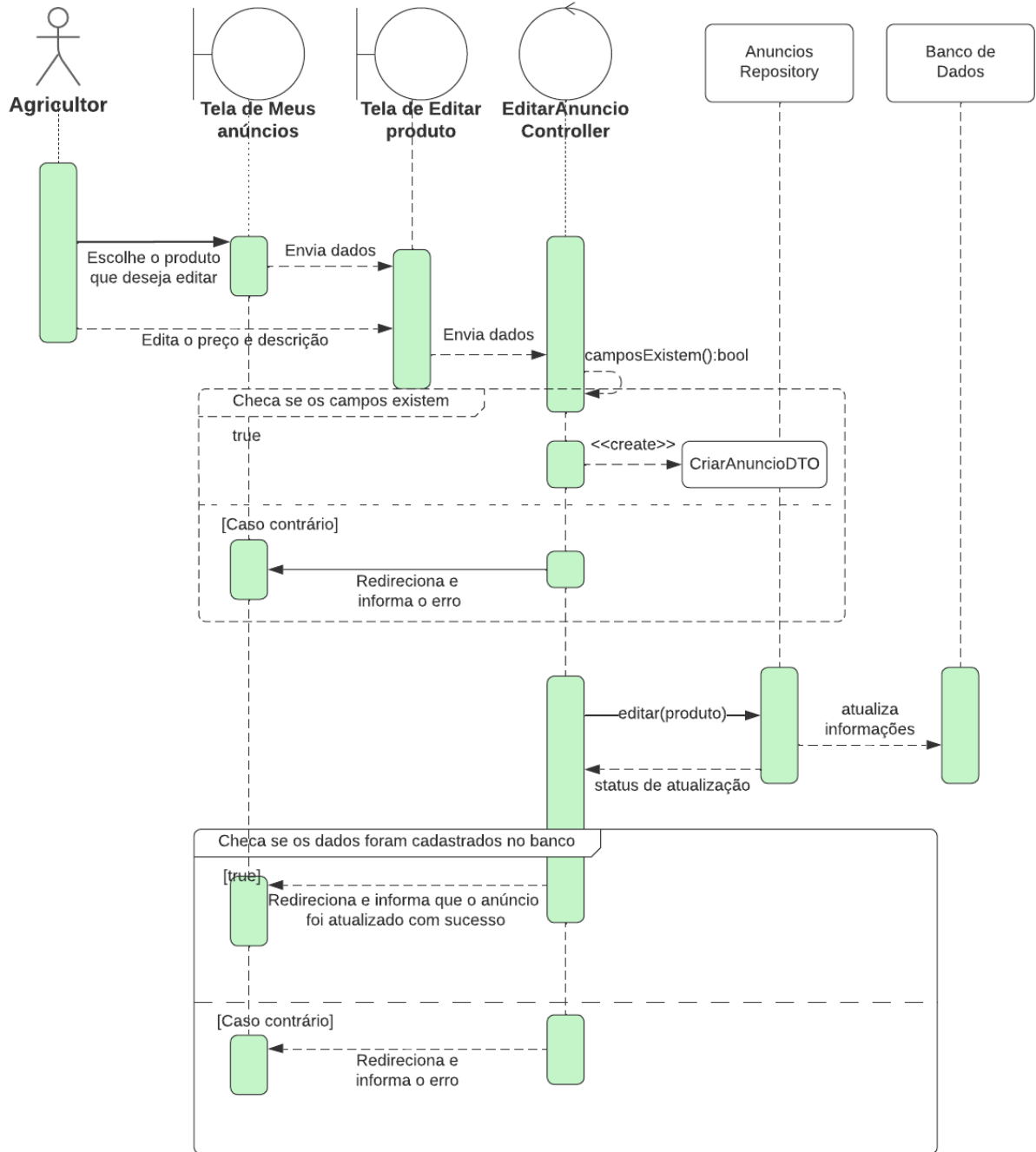
Figura 92 - Diagrama de Sequência do Caso de Uso: Anunciar Produtos



Fonte: Imagem elaborada pelo autor.

5.6.5.10 Editar Anúncio

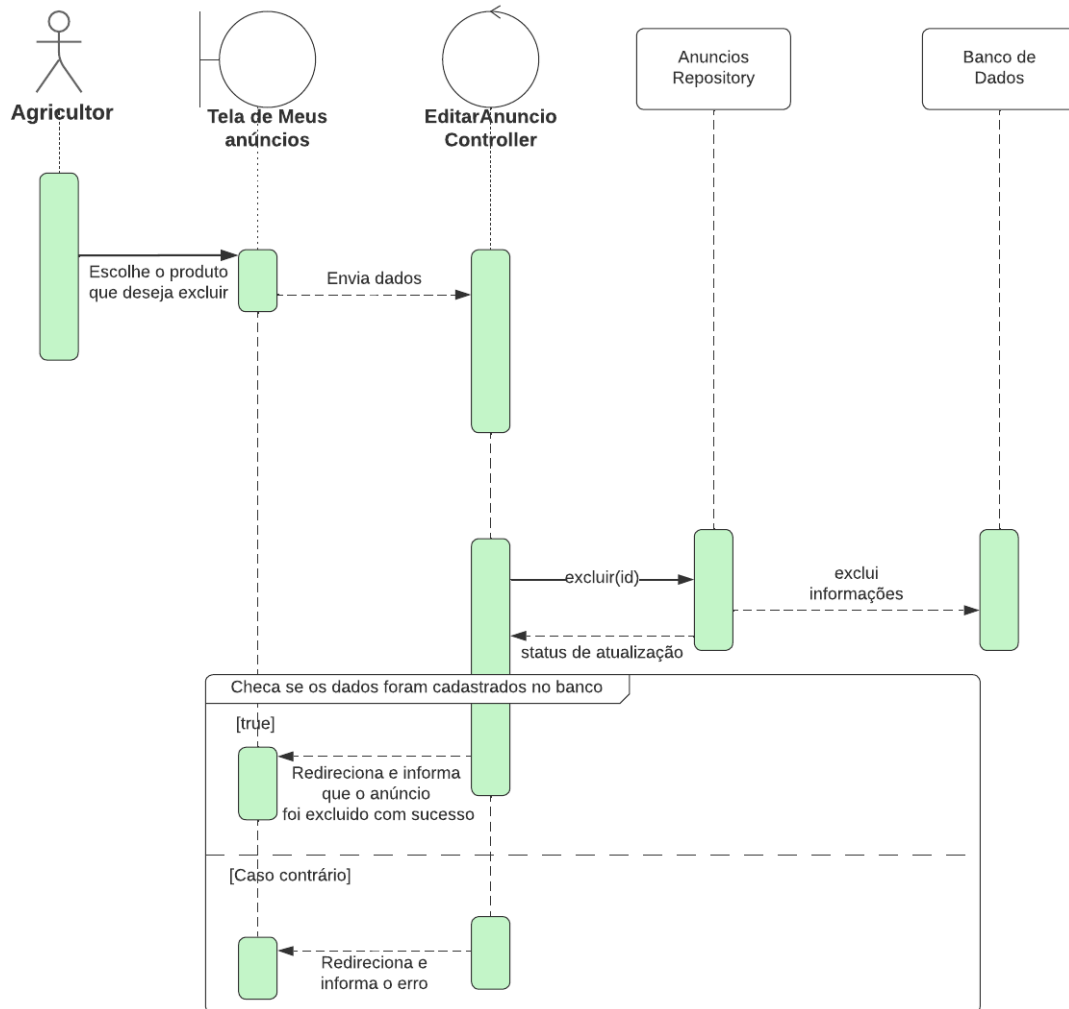
Figura 93 - Diagrama de Sequência do Caso de Uso: Editar Anúncio



Fonte: Imagem elaborada pelo autor.

5.6.5.11 Excluir Anúncio

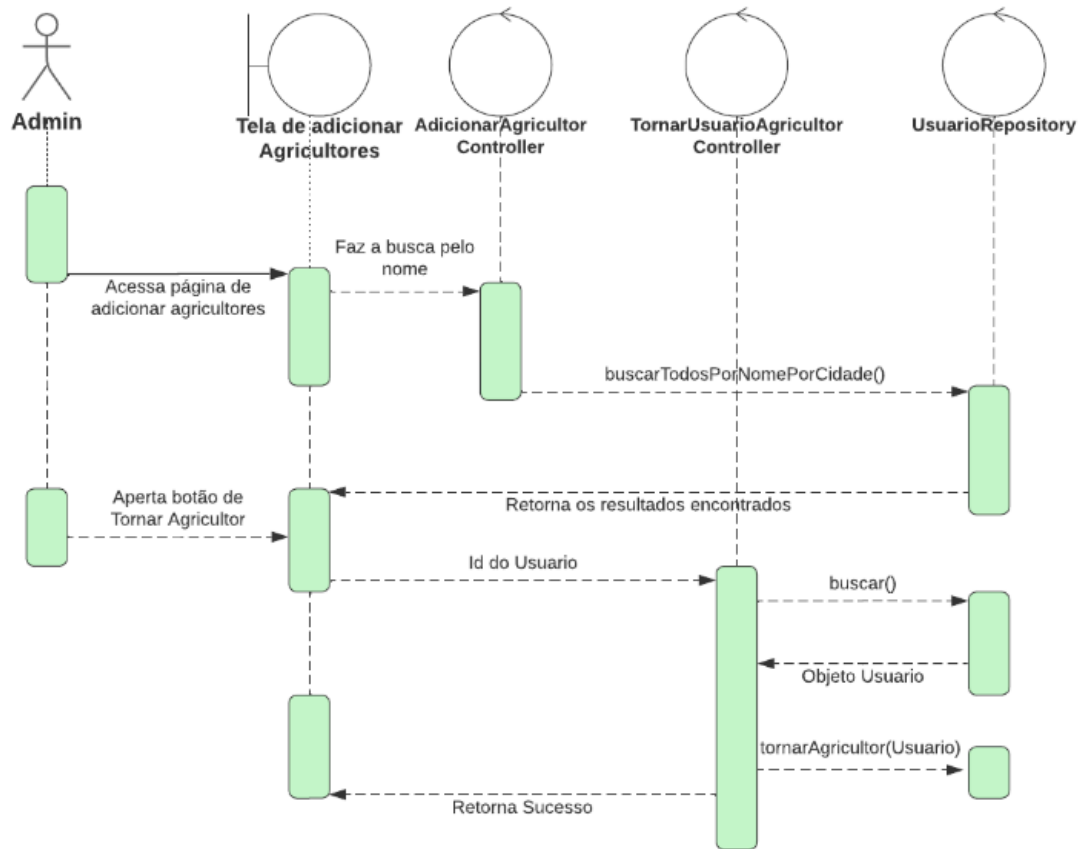
Figura 94 - Diagrama de Sequência do Caso de Uso: Excluir Anúncio



Fonte: Imagem elaborada pelo autor.

5.6.5.12 Adicionar Agricultor

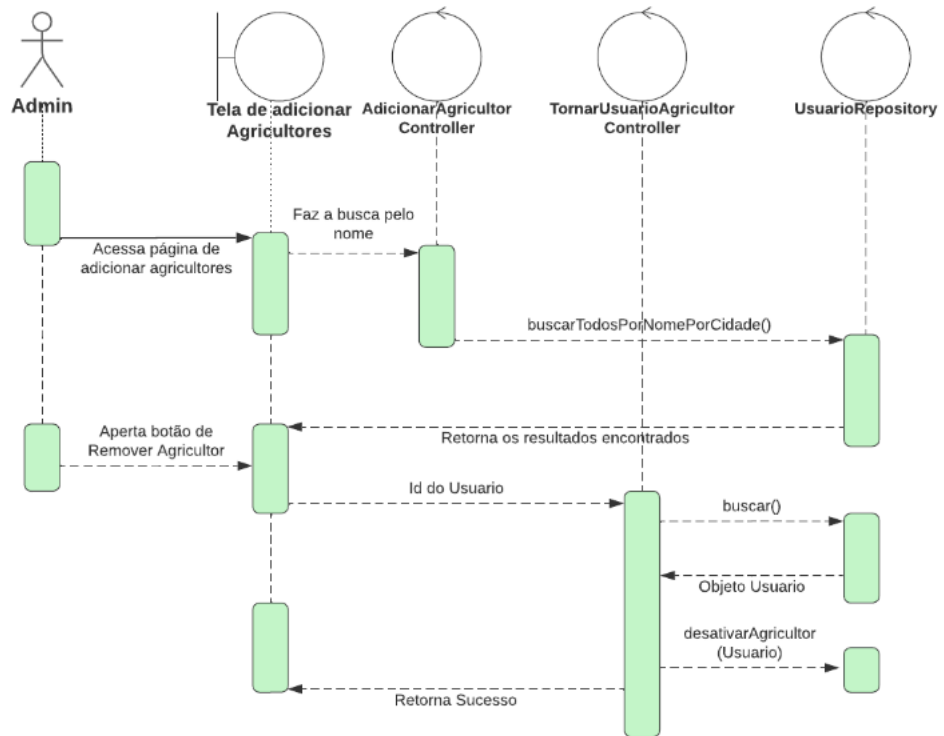
Figura 95 - Diagrama de Sequência do Caso de Uso: Adicionar Agricultor



Fonte: Imagem elaborada pelo autor.

5.6.5.13 Excluir Agricultor

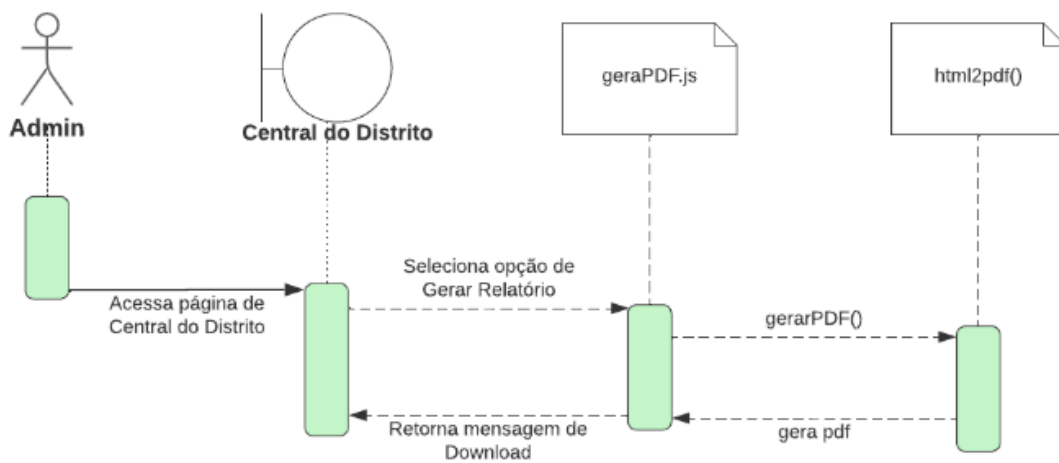
Figura 96 - Diagrama de Sequência do Caso de Uso: Excluir Agricultor



Fonte: Imagem elaborada pelo autor.

5.6.5.14 Gerar Relatório

Figura 97 - Diagrama de Sequência do Caso de Uso: Gerar Relatório



Fonte: Imagem elaborada pelo autor.

5.7 VERSÃO FINAL DA APLICAÇÃO

Essa seção visa disponibilizar uma forma do leitor acessar o produto final do software desenvolvido. A plataforma pode ser acessada por meio do seguinte link³⁸.

Explore cada uma das funcionalidades disponíveis para obter uma experiência de como o software atende às necessidades do projeto. Para facilitar a navegação, seguem abaixo as credenciais de acesso para diferentes perfis de usuário:

- Usuário com permissão de Cliente:
 - E-mail: teste-cliente@hotmail.com
 - Senha: 123456
- Usuário com permissão de Agricultor:
 - E-mail: teste-agricultor@hotmail.com
 - Senha: ufpatcc2024
- Usuário com permissão de Administrador de Tucuruí:
 - E-mail: teste-admin@hotmail.com
 - Senha: feira@tucuruíADMIN

Além disso, o código-fonte da aplicação está armazenado no seguinte repositório³⁹.

³⁸ Acesse a plataforma aqui: <https://feiradoagricultorfamiliar.com.br/> . Acesso em: maio. 2024.

³⁹ Acesse o código-fonte aqui: <https://github.com/vinelouzada/agricultura-familiar>. Acesso em: maio. 2024.

6 AVALIAÇÃO DA PLATAFORMA

Neste capítulo, é apresentado o processo e métodos de avaliação utilizados no sistema da Feira da Agricultura Familiar pelos usuários, com o objetivo subsequente de discutir os resultados obtidos.

6.1 TESTE DE USABILIDADE

Para alcançar uma compreensão sobre a usabilidade e a eficácia da plataforma, foram conduzidos testes de usabilidade de forma presencial, onde o avaliador (autor deste trabalho) se dirigiu até a casa dos entrevistados.

Foram selecionados dois usuários representativos para participar dos testes: a) um potencial cliente e b) um agricultor familiar de Breu Branco. A escolha dessas duas pessoas foi baseada na ideia de simular a perspectiva tanto do consumidor final quanto do fornecedor.

O teste de usabilidade foi conduzido seguindo diretrizes estabelecidas por Nielsen (1994), que ressalta a importância de observar diretamente os usuários em interação com o sistema para identificar possíveis obstáculos e áreas de melhoria.

Para facilitar a condução dos testes, foram estabelecidos cenários comuns da plataforma que os participantes deveriam seguir. Durante a execução dos testes, o avaliador forneceu instruções verbais aos participantes sobre as ações a serem realizadas. A Tabela 22 detalha os cenários:

Tabela 22 - Tarefas a serem realizadas pelo entrevistado

Cenário	Ações
1. Cadastro da plataforma	Explore a plataforma, veja cada uma das informações presente nela;
	Realize seu cadastro;
	Confirme o seu cadastro no e-mail;
	Realize o login na plataforma;

2. Realização de Pedido	Analise os agricultores que estão disponíveis e explore seus anúncios;
	Realize a compra de um determinado produto.
	Na página de carrinho, analise as opções disponíveis e aumente a quantidade requerida daquele produto.
	Escolha um outro produto para adicionar a este carrinho, e prossiga para concluir o pedido.
	Na página de Frete analise as informações;
	Na página de Formas de pagamento selecione seu método de pagamento e finalize o pedido;
	Clique na aba de “Meu perfil” e depois em “Meus pedidos” para verificar seus pedidos realizados na plataforma.
	Clique para ver detalhes de um pedido;
3. Administração de anúncios do agricultor	Observação: O avaliador concederá privilégios de agricultor e o usuário deverá logar novamente na plataforma para as modificações entrarem em vigor.
	Abra o menu “Menu perfil” e explore cada uma das opções, pois aparecerá opções novas.
	Crie um anúncio de algum produto;

	Edite esse anúncio. Exemplo: Altere o preço ou a descrição;
	Exclua esse anúncio.
	Crie novamente um outro anúncio.
	Observação: O inspetor que está aplicando este teste vai realizar a compra de um produto para prosseguir com os próximos passos.
	Verifique se já conseguiu vender algo no painel “Meus Anúncios”
	Clique para ver detalhes da sua venda
	Baixe o PDF
	Conclua o Pedido

Fonte: Tabela elaborada pelo autor.

6.2 RESULTADOS DO TESTE DE USABILIDADE

Para os testes de usabilidade, considerando que a plataforma é web e responsiva, optou-se por realizar o teste com o potencial cliente, utilizando um computador, enquanto o agricultor realizou o teste por meio de um celular. Essa escolha permitiu obter uma visão abrangente da plataforma em diferentes dispositivos, para entender a experiência do usuário em ambas as interfaces.

Os cenários realizados pelo cliente incluíram os números 1 e 2, enquanto o agricultor executou os cenários 1, 2 e 3, como evidenciado na Tabela 22. Essa escolha se baseou nos perfis distintos que cada um representa dentro da plataforma.

6.2.1 Potencial Cliente

Durante o teste de usabilidade aplicado ao usuário representando o potencial cliente, foram observadas algumas questões relevantes sobre a experiência do usuário.

Inicialmente, o participante informou que não havia utilizado um computador nos últimos sete anos. Essa informação é crucial para compreender seu nível de familiaridade e conforto com interfaces digitais, o que pode influenciar sua interação com a plataforma de *e-commerce*. Mesmo com essa pessoa sentindo alguns desconfortos ao usar o computador, a ajuda do avaliador foi primordial para dar início ao teste da plataforma.

Durante a navegação pela plataforma, o participante expressou dúvidas em relação à comunicação pós-compra. Ele perguntou se receberia notificações no celular, seja por mensagem no WhatsApp ou por e-mail, após realizar um pedido. Essa preocupação sugere uma expectativa por comunicação proativa por parte da plataforma, indicando uma oportunidade de melhoria na comunicação pós-venda.

Além disso, o participante sugeriu a inclusão do contato do vendedor nos detalhes do pedido. Essa sugestão ressalta a importância da transparência e da facilidade de comunicação entre comprador e vendedor. Ainda que a informação de telefone esteja na página do vendedor, é importante incluí-la nos detalhes do pedido.

Apesar das dúvidas e sugestões apresentadas, o participante demonstrou uma recepção positiva à aplicação. Ele destacou pontos positivos da experiência e ofereceu feedbacks construtivos, contribuindo para a identificação de áreas de melhoria e aprimoramento da plataforma.

6.2.2 Agricultor Familiar de Breu Branco

O teste foi realizado na fazenda do agricultor, o que exigiu que o autor do trabalho se deslocasse até o local para simular o cotidiano do agricultor com o uso da plataforma. Portanto, durante a aplicação do teste, foi possível observar questões relevantes sobre a experiência do usuário.

O agricultor utilizou seu próprio aparelho celular para acessar a plataforma e, inicialmente, elogiou sua estética, incluindo cores, tipografia e imagens. Ele demonstrou atenção aos detalhes da plataforma, identificando informações

desatualizadas, como o dia da semana em que a feira de Breu Branco ocorria e sua localização. Essa defasagem de informação ocorreu pelo longo período de desenvolvimento da plataforma, ou seja, alguns detalhes foram mudando com o passar do tempo.

Além disso, no decorrer do uso na plataforma, o agricultor foi deixando muito claro que esta ferramenta é muito importante para todos os agricultores da região, porém citou que os agricultores enfrentam muitos problemas com o uso de novas tecnologias. Ou seja, comprova o que já foi dito por (BUAINAIN, 2021).

Ao concluir sua experiência com a plataforma, o agricultor expressou grande satisfação e afirmou que era exatamente o que eles precisavam. Ele demonstrou interesse em começar a utilizá-la imediatamente para adquirir experiência.

6.3 AVALIAÇÃO DA PLATAFORMA

Ao concluir a interação dos usuários com a plataforma, eles preencheram o formulário do SUS, o qual foi impresso e posteriormente marcado à caneta. Os resultados são apresentados nas Figuras 98 e 99:

6.3.1 Potencial Cliente

Figura 98 - Questionário do SUS preenchido pelo Potencial Cliente

Questionário SUS	Discordo Totalmente			Concordo totalmente	
	1	2	3	4	5
Eu gostaria de usar este sistema com frequência.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema desnecessariamente complexo.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema fácil de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu achei que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar este sistema.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei as várias funções do sistema bem integradas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema muito inconsistente.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei que a maioria das pessoas aprenderia a usar este sistema rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema muito complicado.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu me senti confiante usando o sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu precisei aprender muitas coisas antes de começar a usar o sistema.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: Imagem elaborada pelo autor.

Com base nos dados preenchidos acima, podemos realizar o cálculo do índice de satisfação do usuário:

$$\text{Perguntas Ímpares} = (4 - 1) + (5 - 1) + (4 - 1) + (4 - 1) + (5 - 1) = 17$$

$$\text{Perguntas Pares} = |2 - 5| + |2 - 5| + |1 - 5| + |1 - 5| + |2 - 5| = 17$$

$$\text{Índice de Satisfação} = (17 + 17) * 2.5 = 85$$

6.3.2 Agricultor Familiar de Breu Branco

Figura 99 - Questionário do SUS preenchido pelo Agricultor Familiar

Questionário SUS	Discordo Totalmente		Concordo totalmente		
	1	2	3	4	5
Eu gostaria de usar este sistema com frequência.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu achei o sistema desnecessariamente complexo.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema fácil de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eu achei que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar este sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu achei as várias funções do sistema bem integradas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu achei o sistema muito inconsistente.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu achei que a maioria das pessoas aprenderia a usar este sistema rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eu achei o sistema muito complicado.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu me senti confiante usando o sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Eu precisei aprender muitas coisas antes de começar a usar o sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Fonte: Imagem elaborada pelo autor.

$$\text{Perguntas Ímpares} = (5 - 1) + (4 - 1) + (5 - 1) + (4 - 1) + (5 - 1) = 18$$

$$\text{Perguntas Pares} = |3 - 5| + |5 - 5| + |2 - 5| + |2 - 5| + |5 - 5| = 8$$

$$\text{Índice de Satisfação} = (18 + 8) * 2.5 = 65$$

6.3.3 Considerações do teste de usabilidade

Com base nas avaliações realizadas utilizando o formulário do SUS, foi possível observar que a plataforma de e-commerce voltada para a agricultura familiar alcançou uma média de Índice de Satisfação do Cliente de 75. Esse resultado, superior à média padrão de 68 (SAURO, 2012), reflete um desempenho positivo e estimulante para os próximos passos do crescimento da plataforma.

7 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Este capítulo apresenta as conclusões derivadas deste estudo e oferece sugestões para direcionar futuras pesquisas que possam enriquecer não apenas este projeto, mas também outros trabalhos correlatos.

7.1 CONCLUSÃO

O presente trabalho teve o objetivo o desenvolvimento de uma Aplicação Web responsiva que divulgasse a iniciativa e os objetivos da Feira da Agricultura Familiar, além disso, a comercialização dos seus produtos na plataforma, pois é uma forma de impulsionar as vendas e estarem cada vez mais em contato com o ambiente tecnológico.

A aplicação proporcionou uma riqueza de conhecimento para todos os envolvidos, além de disponibilizar uma extensa documentação explicativa. Esta documentação inclui diagramas de casos de uso, especificações de casos de uso, diagramas de sequência e muito mais, o que servirá como um recurso valioso para os futuros envolvidos no projeto.

Além disso, a avaliação de usabilidade do sistema apontou uma média de Índice de Satisfação do Cliente de 75, o que é considerável um bom nível, o que nos pressupõe um certo interesse dos usuários no uso da plataforma em um futuro bem próximo.

Conseqüentemente, conclui-se que o sistema desenvolvido atendeu às expectativas propostas e tem o potencial de gerar melhorias significativas nas receitas das famílias produtoras rurais.

7.2 SUGESTÕES PARA TRABALHOS FUTUROS

A seguir são apresentadas algumas sugestões de trabalhos que podem ser realizados no âmbito de desenvolvimento e de comunidade acadêmica.

7.2.1 Para o desenvolvimento

Para um software, sempre existirá espaços para melhorias, por isso, a aplicação “Feira do Agricultor Familiar” pode aperfeiçoar nos seguintes pontos:

1. **Inserção de Testes Automatizados:** É essencial implementar testes automatizados para aumentar a eficiência na detecção de falhas ao adicionar ou remover funcionalidades na aplicação. Atualmente, a aplicação carece de testes automatizados, incluindo testes de unidade, integração ou e2e (ponta a ponta).
2. **Melhoria na Responsividade:** A responsividade da aplicação pode ser aprimorada para garantir que os elementos da tela se adaptem melhor a diferentes dispositivos, tornando-os visíveis e intuitivos para o usuário. Embora a aplicação seja atualmente responsiva, algumas telas podem ser melhoradas nesse aspecto.
3. **Refatoração do Código:** É recomendável implementar mais classes do tipo "Service" para lidar com as regras de negócio, enquanto as classes do tipo "Controller" devem se concentrar apenas em receber requisições e fornecer respostas. Atualmente, as classes do tipo "Controller" estão assumindo a responsabilidade pela execução de todas as regras de negócio.
4. **Inclusão de Funcionalidades:** Com base nas sugestões dos participantes do teste de usabilidade, é importante adicionar as seguintes funcionalidades à aplicação:
 - (a) Implementar o envio de mensagens por e-mail e WhatsApp para o vendedor e cliente quando um pedido for realizado, fornecendo detalhes da compra.
 - (b) Incluir o contato telefônico do vendedor nos detalhes do pedido para facilitar a comunicação entre o vendedor e o cliente.

7.2.2 Para a Comunidade Acadêmica

Como iniciativa para a comunidade acadêmica sugere-se a aplicação de mais testes de usabilidade com o objetivo de coletar informações para futuros trabalhos e implementações nesta aplicação *web*.

Além disso, sugere-se a oferta de cursos de informática básica direcionados ao público da agricultura familiar, juntamente com programas de alfabetização, como meio de fomentar o interesse e aumentar a adesão à plataforma. A análise dos dados obtidos desses testes pode permitir a identificação de padrões entre os agricultores que participaram ou não dos cursos, facilitando a compreensão das dificuldades enfrentadas por cada grupo de usuários.

8 REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, Guilhermy William Macedo et al. **ANÁLISE E DESENVOLVIMENTO DE UM APLICATIVO E-COMMERCE PARA COMERCIALIZAÇÃO DE PRODUTOS ORIUNDOS DA AGRICULTURA FAMILIAR**. 2019.

APPMMASTER.IO. **Padrão do Controlador Frontal**. Disponível em: <https://appmaster.io/pt/glossary/padrao-do-controlador-frontal>. Acesso em: 21 de dezembro de 2023.

AVELLARE, João. Variações para cada usuário: **Tamanho e resolução da tela digital**. Disponível em: <https://www.avellareduarte.com.br/interface/variacoes-para-cada-usuario/tamanho-e-resolucao-da-tela-digital/>. Acesso em: 2020

BREITENBACH, Raquel. **Estratégias de enfrentamento dos efeitos da pandemia na agricultura familiar**. Desafio Online, v. 9, n. 1, 2021.

BRICALLI, Luiz Carlos Leonardi et al. **FEIRA NA PALMA DA MÃO: UMA PLATAFORMA DIGITAL PARA A VENDA DIRETA DOS PRODUTOS DA AGRICULTURA FAMILIAR**. Ethnoscintia-Brazilian Journal of Ethnobiology and Ethnoecology, v. 7, n. 4, p. 68-75, 2022.

BROOKE, John et al. **SUS-A quick and dirty usability scale**. Usability evaluation in industry, v. 189, n. 194, p. 4-7, 1996.

BUAINAIN, Antônio Márcio; CAVALCANTE, Pedro; CONSOLINE, Letícia. **Estado atual da agricultura digital no Brasil: Inclusão dos agricultores familiares e pequenos produtores rurais**. 2021.

CAHN, Aaron et al. **An empirical study of web cookies**. In: Proceedings of the 25th international conference on world wide web. 2016. p. 891-901.

Censo Agropecuário 2017. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/mda/agricultura-familiar-1#:~:text=O%20Censo%20Agropecu%C3%A1rio%20de%202017>. Acesso em: 31 mar. 2023.

COCKBURN, Alistair. **Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software**. Bookman Editora, 2005.

DA SILVA, Walyson Monteiro et al. Marketing digital, **E-commerce e pandemia: uma revisão bibliográfica sobre o panorama brasileiro**. Research, Society and Development, v. 10, n. 5, p. e45210515054-e45210515054, 2021.

DA SILVA SMITH, Diego et al. " Marketplace": **Uma nova esfera pública sobre os canais de mercados**. Cuadernos de Educación y Desarrollo, v. 15, n. 6, p. 5804-5828, 2023.

DE MELO CRUZ, Wander Luis. **Crescimento do e-commerce no Brasil: desenvolvimento, serviços logísticos e o impulso da pandemia de Covid-19.** *GeoTextos*, 2021.

DE MENDONÇA, Herbert Garcia. **E-commerce.** *Revista Inovação, Projetos e Tecnologias*, v. 4, n. 2, p. 240-251, 2016.

DOS SANTOS, Edson Luiz. **Do escambo à inclusão financeira: a evolução dos meios de pagamento.** Linotipo Digital, 2014.

Ebit | Nielsen Webshoppers 42. **WEBSHOPPERS 42ª EDIÇÃO** | 2020. Disponível em: <https://myagmexpress.com/downloads/Webshoppers_42.pdf>. Acesso em: 30 mar. 2023.

E-commerce no Brasil: principais dados do mercado em 2022. Disponível em: <<https://www.smarthint.co/ecommerce-no-brasil/>>. Acesso em: 30 mar. 2023.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados.** 6ª ed. São Paulo: Editora Pearson, 2011.

FOWLER, Martin. **Analysis Patterns, Reusable Object Models.** London: Addison Wesley, 1997.

FREECODECAMP. **Autenticação x Autorização: Qual é a Diferença?** Disponível em: <https://www.freecodecamp.org/portuguese/news/autenticacao-x-autorizacao-qual-e-a-diferenca/>. Acesso em: 11 de dezembro de 2023.

GANESHICMC. **Cookies e Sessões.** Disponível em: https://gitbook.ganeshicmc.com/web/semana-1/11_cookies_e_sessoes. Acesso em: 19 de dezembro de 2023.

GAZOLLA, Marcio; DE AQUINO, Joacir Rufino. **Reinvenção dos mercados da agricultura familiar no Brasil: a novidade dos sites e plataformas digitais de comercialização em tempos de Covid-19.** *Estudos Sociedade e Agricultura*, v. 29, n. 2, p. 427-460, 2021.

HELLER, Eva. **A psicologia das cores: como as cores afetam a emoção e a razão.** Editora Olhares, 2022.

HEUSER, Carlos Alberto. **Projeto de banco de dados:** Volume 4 da Série Livros didáticos informática UFRGS. Bookman Editora, 2009.

HOSTINGER. **Linguagens de Programação Mais Usadas.** Disponível em: <https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas>. Acesso em: 11 de dezembro de 2023.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores, uma abordagem top-down**. 2013.

Manifesto Ágil. **Princípios do Manifesto Ágil**. Disponível em: <https://agilemanifesto.org/iso/ptbr/principles.html>. Acesso em: 15 dez. 2023.

MELO, Max Yan Alves. **Empreendedorismo digital: evolução do e-commerce**. 2021.

MENDES, Iasmin Santos. **Arquitetura monolítica vs microsserviços: uma análise comparativa**. 2021.

MILANI, André. **MySQL-guia do programador**. Novatec Editora, 2007.

MULLER, Vilma Nilda. E-commerce: vendas pela internet. **Fundação Educacional do Município de Assis**, 2013.

NIEDERLE, Paulo André; SCHNEIDER, Sergio; CASSOL, Abel Perinazzo. **Mercados alimentares digitais: inclusão produtiva, cooperativismo e políticas públicas**. 2021.

NIELSEN, Jakob. **Usability engineering**. Morgan Kaufmann, 1994.

NOGUEIRA, Marcelo, Abe, Jair Minoru, **Normas e modelos de qualidade como política de produção de software no contexto brasileiro**. In: XV SIMPEP Simpósio Engenharia de Produção, 2008, Bauru. Anais. Bauru: UNESP, 2008.

OLIVEIRA, Maria et al. **Aplicação web para gerenciamento de resíduos sólidos recicláveis**. In: Anais do X Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais. SBC, 2019. p. 145-153.

PHP. **Função password_hash**. Disponível em: https://www.php.net/manual/pt_BR/function.password-hash.php. Acesso em: 11 de dezembro de 2023a.

PHP. **Função password_verify**. Disponível em: https://www.php.net/manual/pt_BR/function.password-verify.php. Acesso em: 11 de dezembro de 2023b.

PHP. **Introdução ao Controle de Sessão**. Disponível em: https://www.php.net/manual/pt_BR/intro.session.php. Acesso em: 18 de dezembro de 2023c.

POTTER, Richard. TURBAN, Efraim. RAINER, Kelly. **Administração de Tecnologia da Informação**. 3. Ed. – São Paulo: Campus, 2005.

PRESSMAN, Roger. **Engenharia de Software - Uma abordagem Profissional**. 8a. ed. New York: McGraw Hill, 2016.

RICHARDS, Mark; FORD, Neal. **Fundamentals of software architecture: an engineering approach**. O'Reilly Media, 2020.

ROB, Peter; CORONEL, Carlos. Sistemas de banco de dados. **Projeto, implementação e**, 2011.

RODRIGUES, Julia Letícia Alves. **Agricultura familiar e os impactos da covid-19: uma breve análise**. 2022

SANTANA, Vagner Figuerêdo de. **Padrões Arquiteturais de Sistemas**. 2011. Disponível em: <<https://pt.slideshare.net/santanavagner/padroes-arquiteturais-de-sistemas>>. Acesso em 11 abril. 2023

SANTOS, Isaias et al. **Possibilidades e limitações da arquitetura mvc (model – view – controller) com ferramenta ide (integrated development environment)**. 2010. 56f. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) - Universidade José do Rosário Vellano, Alfenas, Mg.

SAURO, Jeff. **Measuring Usability With The System Usability Scale (SUS)**. 2011. Disponível em: <<https://measuringu.com/sus/>>. Acesso em: 18 abril. 2024

SCHNEIDER, S. et al. **Os efeitos da pandemia da Covid-19 sobre o agronegócio e a alimentação**. Estudos Avançados, v. 34, p. 167-188, 2020

SIERTI. **Argon2id: A Segurança nas Senhas**. Disponível em: <https://sierti.com.br/blog/argon2id-seguranca-senhas>. Acesso em: 11 de dezembro de 2023.

SINGH, Siddharth et al. **MVC framework: a modern web application development approach and working**. International Research Journal of Engineering and Technology, v. 7, n. 01, p. 51-55, 2020.

SOMMERVILLE, Ian. Engenharia de software, 9a. **São Paulo, SP, Brasil**, p. 63, 2011.

STRUNCK, Gilberto. **Como Criar Identidades Visuais para Marcas de Sucesso**. Rio de Janeiro: Rio Books, 2001.

SUBTIL, Filipa Mónica de Brito Gonçalves. Tecnologia, economia e política: o telégrafo como antecessor da Internet. **Estudos em Comunicação**, p. 25-40, 2014.

TANEMBAUM, Andrew S. **Redes de Computadores**, 7ª Edição, Editora Campus, Rio de Janeiro – RJ, 2003

TEIXEIRA, Felipe Colvara; SILVA, Roberta D. de O.; BONA, Rafael José. **O processo de desenvolvimento de uma identidade visual**. In: Congresso brasileiro de ciências da comunicação da região sul. 2012.

VÁSQUEZ, Ruth Peralta. **Identidade de marca, gestão e comunicação.** *Organicom*, v. 4, n. 7, p. 198-211, 2007.