



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS DE CASTANHAL
FACULDADE DE COMPUTAÇÃO

UMA PROPOSTA DE FERRAMENTA DIGITAL PARA FACILITAR A
COMERCIALIZAÇÃO DA PRODUÇÃO RURAL NO MUNICÍPIO DE ACARÁ - PA

Discente: Isaque M. da Silva

UFPA / CCAST / FACOMP

Polo de Mãe do Rio

Mãe do Rio – Pará – Brasil

2023

ISAQUE MENEZES DA SILVA

**UMA PROPOSTA DE FERRAMENTA DIGITAL PARA FACILITAR A
COMERCIALIZAÇÃO DA PRODUÇÃO RURAL NO MUNICÍPIO DE ACARÁ - PA**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Computação como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação pela UFPA - Universidade Federal do Pará.

Orientador: Prof. Dr. Clenilson Rodrigues da Silveira

UFPA / CCAST / FACOMP

Polo de Mãe do Rio

Mãe do Rio – Pará – Brasil

2023

SUMÁRIO

1. INTRODUÇÃO	9
2. TEMA PROPOSTO: CONCEITOS	12
2.1 TEMA PROPOSTO	12
2.2 APLICAÇÕES EM TRÊS CAMADAS	13
2.3 ARQUITETURA DE MODELO MVC	14
2.3.1 As 3 Camadas MVC	15
2.4 <i>WEB SERVICES</i>	17
2.4.1 O protocolo SOAP (<i>Simple Object Access Protocol</i>)	17
2.4.2 O protocolo REST (<i>Representational State Transfer</i>)	17
3. METODOLOGIA DA CONSTRUÇÃO	22
3.1 TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO	22
3.1.1 Ambiente de Desenvolvimento Integrado	22
3.1.2 IDE <i>Visual Studio Code®</i>	22
3.1.3 <i>Git</i> - Ferramenta de Versionamento de Código	23
3.1.4 Linguagens de marcação, de <i>script</i> e de programação	24
3.1.5 SGBD (Sistemas Gerenciadores de Banco de Dados) <i>MySQL</i>	26
4. ESPECIFICAÇÃO DO SISTEMA	28
4.1 DIAGRAMA DE CASOS DE USO	28
4.2 Engenharia de Requisitos	34
4.2.1 Lista de Requisitos Não-Funcionais.	34
4.2.2 Lista de Requisitos Funcionais.	35
5. ESTRUTURA DO SISTEMA PROPOSTO	36
5.1 REALIZAR <i>LOGIN</i>	38
5.2 CADASTRAR ANÚNCIO	40
5.3 ATUALIZAR ANÚNCIO	45
5.4 REMOVER ANÚNCIO	49
5.5 EXIBIR ANÚNCIO	52
6. IMPLANTAÇÃO E INFRAESTRUTURA	54
6.1 <i>Gitlab®</i>	54
6.2 Hospedagem em Nuvem	54
6.3 Gerenciamento dos Dados do Usuário	56
7. CONSIDERAÇÕES FINAIS	58

7.1 TRABALHOS FUTUROS	58
7.1.1 Segurança da Informação	59
8. REFERÊNCIAS	61

AGRADECIMENTOS

Gostaria de agradecer a todos os professores dessa instituição de ensino que em muito contribuíram para a realização deste trabalho. Professores que com seus ensinamentos tornaram a minha formação acadêmica possível. Em especial aos professores, Tássio Costa de Carvalho, Thiago Antônio Sidônio Coqueiro, Igor Ruiz Gomes e Yomara Pinheiro Pires. Agradeço ao meu orientador, Clenilson Rodrigues da Silveira que me guiou pelo caminho deste trabalho de Conclusão de Curso, sem o qual nada disso seria possível.

RESUMO

Este trabalho é uma proposta de *software* que visa conectar vendedores e compradores em regiões rurais por meio de uma plataforma amigável. A plataforma, permitirá que os vendedores listem seus produtos e serviços, definam preços e se comuniquem com potenciais compradores. Os compradores poderão pesquisar produtos e serviços, consultar e comprar itens diretamente do vendedor. A plataforma é voltada para áreas rurais do município de Acará, Estado do Pará, onde os métodos tradicionais de compra e venda de produtos agrícolas, e serviços podem ser limitados. Este projeto pretende melhorar as oportunidades econômicas disponíveis para essas comunidades. Este trabalho propõe que, para o desenvolvimento do sistema sejam utilizadas tecnologias como *Web Services*, PHP para o *back-end* e HTML, CSS para o *front-end* e padrão de projetos para nortear a construção da aplicação.

Palavras-chave: Tecnologia PHP. *Web Services*. Padrões de Projetos. Plataforma de compra e venda de produtos agrícolas.

ABSTRACT

This work is a software proposal that aims to connect sellers and buyers in rural areas through a friendly platform. It will allow sellers to list their products and services, set prices and communicate with potential buyers. Buyers will be able to search for products and services, browse and purchase items directly from the seller. The platform is aimed at rural areas of the municipality of Acará, State of Pará, where traditional methods of buying and selling agricultural products and services can be limited. This project aims to improve the economic opportunities available to these communities. This work proposes that, for the development of the system, technologies such as Web Services, PHP for the back-end and HTML, CSS for the front-end and pattern of projects to guide the construction of the application are used.

Keywords: PHP Technology. Web Services. Design Patterns. Platform for buying and selling agricultural products.

LISTA DE FIGURAS

Figura 1 - APLICAÇÃO EM 3 CAMADAS.....	14
Figura 2 - ARQUITETURA MVC.....	16
Figura 3 - CÓDIGO DO SISTEMA PROPOSTO	19
Figura 4 - <i>JSON OBJECT</i>	20
Figura 5 - <i>JSON ARRAY</i>	20
Figura 6 - <i>JSON VALUE</i>	21
Figura 7 - SISTEMA DE BANCO DE DADOS	26
Figura 8 - DIAGRAMA DE CASO DE USO REALIZAR LOGIN.....	29
Figura 9 - DIAGRAMA DE CASO DE USO CADASTRAR PRODUTO	30
Figura 10 - DIAGRAMA DE CASO DE USO ATUALIZAR PRODUTO	31
Figura 11 – DIAGRAMA DE CASO DE USO DELETAR PRODUTO	32
Figura 12 - DIAGRAMA DE CASO DE USO EXIBIR PRODUTO	33
Figura 13 – ESTRUTURA DE DIRETÓRIOS.....	37
Figura 14 - TELA REALIZAR LOGIN.....	38
Figura 15 - TELA DE CONSISTÊNCIA DA AUTENTICAÇÃO.....	39
Figura 16 - DIAGRAMA DE ESTADOS REALIZAR LOGIN.....	40
Figura 17 - MENU DE CONTEXTO.....	41
Figura 18 - TELA CADASTRAR NOVO ANÚNCIO.....	43
Figura 19 - DIAGRAMA DE ESTADOS CADASTRAR ANÚNCIO.....	44
Figura 20 - MENU ATUALIZAR ANÚNCIO.....	45
Figura 21 - TELA ATUALIZAR ANÚNCIO.....	46
Figura 22 - MENSAGEM ANÚNCIO ATUALIZADO.....	47
Figura 23 - DIAGRAMA DE ESTADOS EDITAR ANÚNCIO.....	48
Figura 24 - MENU REMOVER ANÚNCIO.....	49
Figura 25 - DIÁLOGO REMOVER ANÚNCIO.....	50
Figura 26 - DIAGRAMA DE ESTADOS REMOVER ANÚNCIO.....	51

Figura 27 - TELA EXIBIÇÃO ANÚNCIO CADASTRADO.....	52
Figura 28 - TELA EXIBIÇÃO ANÚNCIOS PUBLICADOS.....	53

LISTA DE SIGLAS

AJAX – *Asynchronous Javascript And XML*

API – *Application Programming Interface*

SBC – *Sociedade Brasileira de Computação*

CSS – *Cascading Style Sheets*

CID – *Confidencialidade, Integridade, Disponibilidade*

DOM – *Document Object Model*

HTML – *Hypertext Markup Language*

HTTP – *Hypertext Transfer Protocol*

IDE – *Integrated Development Environment*

IP – *Internet Protocol Address*

JSP – *Java Server Pages*

JSON – *JavaScript Object Notation*

MVC – *Model View Controller*

SGBD – *Sistemas Gerenciadores de Banco de Dados*

SQL – *Structured Query Language*

SOAP – *Simple Object Access Protocol*

OMG – *Object Management Group*

REST – *Representational State Transfer*

PHP – *PHP:Hypertext Preprocessor*

P2P – *Peer-to-Peer*

PA – *Pará*

TCP – *Transmission Control Protocol*

XHTML – *Extensible Hypertext Markup Language*

XML – *Extensible Markup Language*

WSDL – *Web Services Definition Language*

UML – *Unified Modeling Language*

1. INTRODUÇÃO

O município de Acará é parte integrante da Mesorregião do Nordeste Paraense e da Microrregião de Tomé-Açu. Sua área total territorial é de 4.344 km², distante da capital Belém, cerca de 100 quilômetros. Atualmente, o município de Acará tem uma população estimada em 55 mil habitantes. Tanto na cultura quanto na economia acaraense, há o estabelecimento de uma significativa e próspera colônia de imigrantes japoneses. Sendo que a maioria da população, cerca de 80%, reside na zona rural e são agricultores familiares que produzem, além de mandioca, o açaí, a pimenta-do-reino, o cacau, dentre outras culturas, mas, dentre todas essas, a de pimenta-do-reino é predominante. (ACARÁ. SOBRE O MUNICÍPIO, 2023).

E tendo alguns produtos com bastante produção diária, como farinha, pupunha, e pimenta, o município é um local com bastante potencial de crescimento e expansão comercial. Nesse ponto precisam recorrer a ferramentas tecnológicas, para alavancar e facilitar a negociação entre as partes envolvidas para assim alcançar o esse crescimento mais exponencial, proporcionando maiores oportunidades econômicas para compradores e vendedores.

O modelo de negócio vigente no município de Acará é a produção de produtos agrícolas, tais como farinha-d'água, pupunha, açaí, pimenta-do-reino, dentre outros. Essa comercialização é feita sem nenhum uso de ferramenta ou auxílio tecnológico como internet ou smartphone.

O uso de ferramentas tecnológicas como estratégia de valorização e de facilitação na negociação de produtos, bens, serviços e outros, é substancialmente importante para qualquer área do negócio ou determinado setor da economia. Essa aplicação tecnológica facilita a conexão entre as partes envolvidas, eliminando assim possíveis intermediários, e conseqüentemente, aumentando a margem de lucratividade sobre o produto ou serviço.

O município de Acará, está se tornando um importante nicho para serviços com base tecnológica. Como exemplo destes serviços estão as empresas prestadoras de internet, tanto de fibra óptica quanto de banda larga que, devido ao saturamento e

demanda baixa nas regiões de grandes centros urbanos, como na capital, Belém-PA, estão expandindo seus produtos e serviços para regiões do interior do Estado.

Com essa expansão da oferta de serviços de internet para os municípios com baixa densidade populacional, como Acará, uma ferramenta que utilize esses serviços será mais propícia para ser implementada e obter seu êxito.

O desenvolvimento de uma ferramenta online tal como plataforma web pode melhorar a gestão e a eficácia dos negócios entre vendedores e compradores da região de Acará.

Isto é possível com tecnologia de baixo custo, já que não será cobrado nenhum valor dos vendedores (produtor rural) e nem dos compradores; e de fácil acesso por um *link* na *internet*.

Desta forma, o objetivo geral deste trabalho foi o desenvolvimento de uma ferramenta tecnológica, denominada Conexão Acará, que possa ser uma opção viável para melhorar as oportunidades econômicas para compradores e vendedores em regiões rurais, fornecendo uma plataforma para eles se conectarem e realizarem transações.

Para que se possa alcançar uma aplicação com um bom funcionamento e com os principais requisitos aceitos pelos usuários. Destaca-se como objetivos específicos:

- Fornecer uma plataforma para os vendedores das regiões rurais listarem produtos e serviços, definindo preços e se comunicando com potenciais compradores.
- Permitir que compradores de regiões diferentes, pesquisem produtos e serviços, façam consultas e comprem itens diretamente do vendedor, eliminando os atravessadores ou qualquer outro intermediário que possa onerar seus produtos finais.
- Preencher a lacuna entre as comunidades rurais e o mercado mais amplo, oferecendo maiores oportunidades econômicas para compradores e vendedores.
- Tornar o *software* facilmente acessível às comunidades rurais, projetando-o para ser compatível com dispositivos móveis.

- Fornecer um software que respeite as informações dos usuários, obedecendo os três pilares da segurança da informação: confidencialidade, integridade, disponibilidade (CID).

A importância do projeto e análise de algoritmos eficientes, e a necessidade de desenvolver algoritmos que sejam capazes de lidar com problemas computacionais de forma otimizada, levando em consideração critérios como tempo de execução, consumo de recursos e escalabilidade, é um requisito fundamental em qualquer projeto de *software*. (ZIVIANI, 2007).

2. TEMA PROPOSTO: CONCEITOS

Neste tópico, será descrito a base teórica das ferramentas tecnológicas, e caso de uso do sistema web desenvolvido, chamado de Conexão Acará. Também será feita uma análise do problema, e como o desenvolvimento do sistema tecnológico Conexão Acará se propõe a resolvê-los.

2.1 TEMA PROPOSTO

Este trabalho foi proposto para ser uma opção viável diante da necessidade de integração dos produtores rurais à utilização de tecnologia moderna e de fácil acesso para os compradores interessados em seus produtos, a fim de garantir maior visibilidade, atratividade e competitividade sobre seus produtos agrícolas, expandir e comercializá-los para além das fronteiras territoriais do município de Acará e alcançar um mercado consumidor mais abrangente. Tudo isso de forma rápida, segura, e que esses dados possam persistir em um banco de dados em servidores na nuvem, para evitar perdas.

Por se tratar de uma região territorialmente grande, a internet na maioria das casas dos produtores chega de forma via rádio ou banda larga. Neste caso, optou-se por utilizar tecnologia de construção de sistema tido como “puro”, ou seja, sem uso de *framework*. Um caso de exemplo foi a utilização do PHP, sem *framework* utilizado no “*backend*” da aplicação, isso gera um ganho de desempenho e conseqüentemente de velocidade na comunicação com outros serviços como *Web Services*, o que aumenta as chances de sucesso com conexões com a base de dados remota.

Na parte de “*frontend*” do sistema Conexão Acará, utilizou-se apenas HTML versão 5 e CSS versão 3 para formatação e estilos das páginas, o que deixa o carregamento das mesmas mais rápido para o usuário final. E há “responsividade” no sistema, ou seja, a se adaptar ao tamanho da tela do usuário devido ao uso de CSS3.

A aplicação segue a arquitetura de modelo MVC (Modelo, Visão, Controle). A arquitetura de um sistema tem diversos elementos como: elementos utilitários, de interação, elementos que fazem parte do domínio do problema, elementos de conexão, de persistência, etc. Dessa forma, na arquitetura sempre definimos os seus elementos que serão utilizados no *software* e como eles se conectam.

No entanto, alguns padrões arquiteturais já foram pensados para resolver problemas corriqueiros. Por isso, é sempre interessante entender as características básicas de cada um dos estilos e escolher ou combinar aqueles que atendem melhor às necessidades de um projeto específico.

Isso tudo deve ser feito após uma análise do sistema a ser desenvolvido. Entre as arquiteturas existentes temos: cliente-servidor, *Peer-to-Peer* (P2P), dados compartilhados, máquina virtual, camadas, MVC e muitos outros.

2.2 APLICAÇÕES EM TRÊS CAMADAS

Com o advento da internet houve um movimento para separar a lógica de negócio da interface com o usuário. A ideia é que os usuários da *WEB* possam acessar as mesmas aplicações sem ter que instalar estas aplicações em suas máquinas locais ou qualquer outro componente. Como a lógica do aplicativo, inicialmente contida no cliente “cheio”, não reside mais na máquina do usuário, este tipo de cliente passou a ser chamado de “cliente pobre” ou “magro”.

O principal benefício da arquitetura de três camadas é que, pelo fato de cada camada executar sua própria infraestrutura, cada camada pode ser desenvolvida simultaneamente por uma equipe de desenvolvimento separada e pode ser atualizada ou ajustada conforme a necessidade sem impactar as outras camadas. (IBM. TOPICS, 2023).

Outros benefícios:

- Desenvolvimento mais rápido: Uma vez que cada camada pode ser desenvolvida simultaneamente por diferentes equipes, uma organização pode trazer o aplicativo para o mercado mais rapidamente e os programadores podem utilizar as mais recentes e melhores linguagens e ferramentas para cada camada.
- Escalabilidade melhorada: Qualquer camada pode ser ajustada em escala independentemente das outras, conforme necessário.
- Confiabilidade melhorada: Uma indisponibilidade em uma camada é menos propensa a impactar a disponibilidade ou o desempenho das outras camadas.

- Maior segurança: Uma vez que a camada de apresentação e a camada de dados não podem se comunicar diretamente, uma camada do aplicativo bem projetada pode ter a função de uma espécie de *firewall* interno, impedindo injeções SQL e outros exploradores de vulnerabilidade mal-intencionados.

A Figura 1, seguinte, exemplifica o conceito de uma arquitetura em 3 camadas.

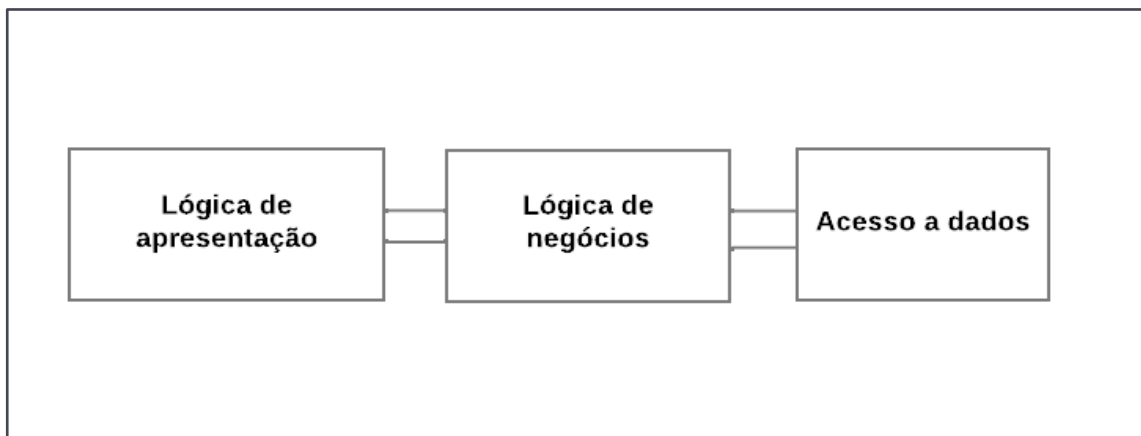


Figura 1 - APLICAÇÃO EM 3 CAMADAS

Fonte: Elaboração do autor

2.3 ARQUITETURA DE MODELO MVC

A arquitetura MVC (Modelo, Visão, Controle) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. (MACORATTI, 2005).

Nessa arquitetura, o modelo representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

Dentre as características desta arquitetura podemos destacar:

O isolamento das camadas proporciona flexibilidade e oportunidades de reuso das classes. A arquitetura MVC pode ser utilizada em vários tipos de projetos como, por exemplo, *desktop*, *web* e *mobile*.

2.3.1 As 3 Camadas MVC

A seguir serão descritos cada uma das 3 camadas (*M-Model*, *V- Views*, *C-Controller*).

A Camada de Modelo é a camada central. Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam por meio das funções, métodos, lógica e regras de negócios estabelecidas.

- Modela os dados e o comportamento por trás do processo de negócios
- Preocupa-se apenas com o armazenamento, manipulação e geração de dados
- É um encapsulamento de dados e de comportamento independente da apresentação.

A Camada de Apresentação - Não está preocupada em como a informação foi obtida ou onde ela foi obtida, apenas exibe as informações repassadas pela camada de Controle.

- Inclui os elementos de exibição no cliente: HTML (*Hypertext Markup Language*), XML (*Extensible Markup Language*), ASP (*Active Server Pages*), JS (*Javascript*).
- É a camada de interface com o usuário.
- É usada para receber a entrada de dados e apresentar o resultado

A Camada de Controle. Determina o fluxo da apresentação servindo como uma camada intermediária entre a camada de apresentação (Visualização) e a lógica (Modelo).

- Controla e mapeia as ações

A Figura 2 demonstra uma estrutura MVC aplicada conjuntamente com o uso da tecnologia PHP e *Web Services*.

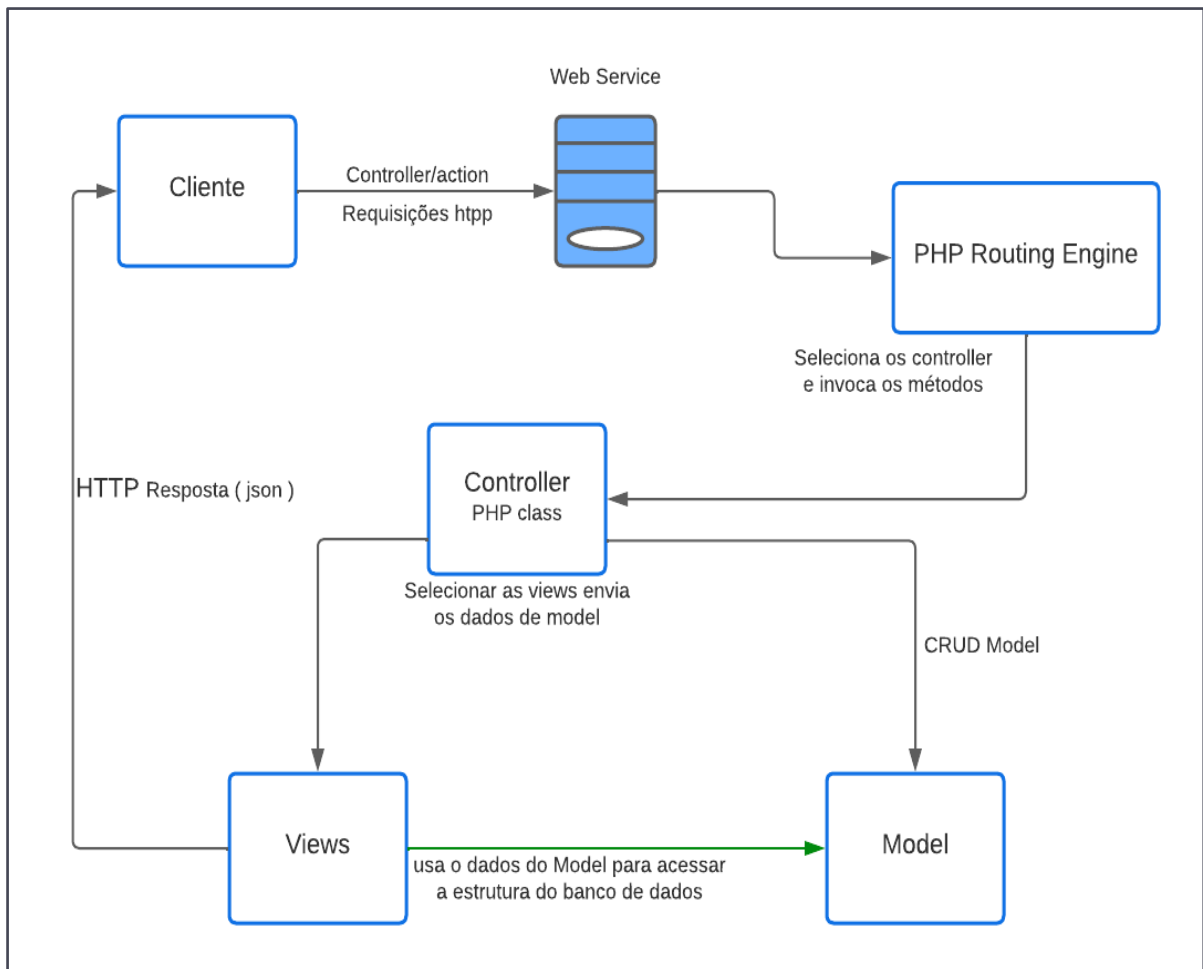


Figura 2 ARQUITETURA MVC

Fonte: Elaboração do autor

No PHP, o modelo pode ser implementado usando classes e objetos que interagem com um banco de dados para recuperar e manipular dados.

A visualização, que é a parte exposta aos usuários e clientes, pode ser criada usando HTML, CSS (*Cascading Style Sheets*), *JavaScript* e pode ser renderizada por PHP, usando um mecanismo de modelo como “*Twig*” ou “*Smarty*” ou mesmo aplicar “*tag*” PHP para alcançar uma integração com a tecnologia de “*front-end*”, na qual é o HTML.

O controlador pode ser implementado usando classes e métodos PHP que manipulam a entrada do usuário e direcionam o fluxo do aplicativo conforme a resposta obtida do modelo.

2.4 WEB SERVICES

Um *Web Services* é um termo genérico para uma função de *software* máquina a máquina interoperável hospedada em um determinado local com um endereço de rede na internet. (REVERBEL, 2006).

Web Services possui uma interface, que oculta os detalhes de implementação para poder ser usada independentemente da plataforma de *hardware* ou *software* em que está implementado e independentemente da linguagem de programação em que está escrita. Essa independência incentiva os aplicativos baseados em *Web Services* a terem suas implementações de tecnologia cruzada, orientadas a componentes e fracamente acopladas.

É necessária uma linguagem intermediária que garanta a comunicação entre a linguagem do *Web Services* e o sistema que faz o pedido ao *Web Services*. Para tal, existem protocolos de comunicação como o SOAP (*Simple Object Access Protocol*) e o REST.

2.4.1 O protocolo SOAP (*Simple Object Access Protocol*)

O protocolo SOAP utiliza XML para enviar mensagens e, geralmente, serve-se do protocolo HTTP para transportar os dados. Associado ao protocolo SOAP está o documento WSDL (*Web Services Definition Language*) que descreve a localização do *Web Services* e as operações que dispõe. Além disso, fornece a informação necessária para que a comunicação entre sistemas seja possível.

2.4.2 O protocolo REST (*Representational State Transfer*)

Os conceitos do REST foram submetidos à tese de doutorado de Roy Fielding nos anos 2000, onde o princípio fundamental é usar o protocolo HTTP para comunicação de dados.

REST é um estilo de arquitetura, não um protocolo e, portanto, anunciá-lo como uma ligação de protocolo é um comportamento absurdamente ignorante para um grupo de empresas de tecnologia (FIELDING, 2008). Ele surgiu recentemente visando simplificar o acesso aos *Web Services*. Este baseia-se no protocolo HTTP e permite utilizar vários formatos para representação de dados, como JSON (*JavaScript Object Notation*) (um dos mais utilizados), XML, RSS, entre outros.

Um dos princípios do REST é a chamada “comunicação sem estado”, que permite a transferência independente de dados, por API (*Application Programming Interface*) ou navegadores, principalmente pelo protocolo HTTP.

O HTTP é o caminho mais conhecido nas transferências de dados. Quanto às operações, ele permite:

Criar dados no servidor (*POST*);

Atualizar (*PUT*);

Pesquisar (*GET*);

Executar;

Remover (*DELETE*).

Assim, o uso do REST, nos proporciona estabelecer a comunicação entre aplicações. Ao usar o navegador em um dispositivo qualquer com acesso à internet, será estabelecido uma conexão TCP/IP com o servidor de destino e uma requisição GET HTTP, que será enviada com o endereço buscado. O Cliente receberá uma resposta via hipertexto que pode ser HTML, ou outro formato como JSON. (EQUIPE TOTVS. 2020).

Os *Web Services* que adotam REST são mais leves. Outro diferencial é a flexibilidade, sendo possível escolher o formato que melhor se encaixa para as mensagens do sistema. Os mais utilizados, além do texto puro, são JSON e XML.

2.4.2.1 JSON (*JavaScript Object Notation*)

A figura 3 representa um objeto json utilizado dentro do código-fonte do sistema Conexão Acará.

```
        $arrayResponse=[
            "retorno"=>"success",
            "page"=>"myaccount",
            "tentativas"=>$this->tentativas
        ];
    }
    return json_encode($arrayResponse);
}
```

Figura 3 CÓDIGO DO SISTEMA PROPOSTO

Fonte: Elaboração do autor

JSON como uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação *JavaScript*. Ele é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, PHP, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados. (STANDARD ECMA-262, 2022).

JSON está constituído em duas estruturas:

Uma coleção de pares, nome/valor. Em várias linguagens, isto é caracterizado como um object, record, struct, dicionário, hash table, keyed list, ou arrays associativas.

Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma array, vetor, lista ou sequência.

Estas são estruturas de dados universais. Virtualmente todas as linguagens de programação modernas as suportam, de uma forma ou de outra. É aceitável que um

formato de troca de dados que seja independente de linguagem de programação se baseie nestas estruturas.

Um objeto é um conjunto desordenado de pares, nome/valor. Um objeto começa com chave de abertura ({) e termina com chave de fechamento (}). Cada nome é seguido por (:) dois pontos e os pares nome/valor são seguidos por vírgula. A Figura 4 explica em detalhes um objeto JSON.

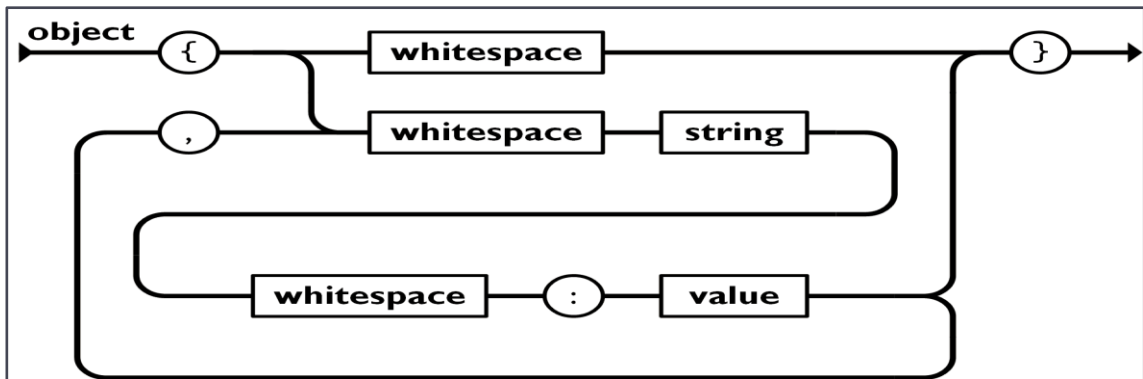


Figura 4 JSON OBJECT

Fonte: JSON (*JavaScript Object Notation*)

Um *array* é uma coleção de valores ordenados. Um *array* começa com colchete de abertura ([) e termina com colchete de fechamento (]). Os valores são separados por vírgula. Notamos isso na Figura 5.

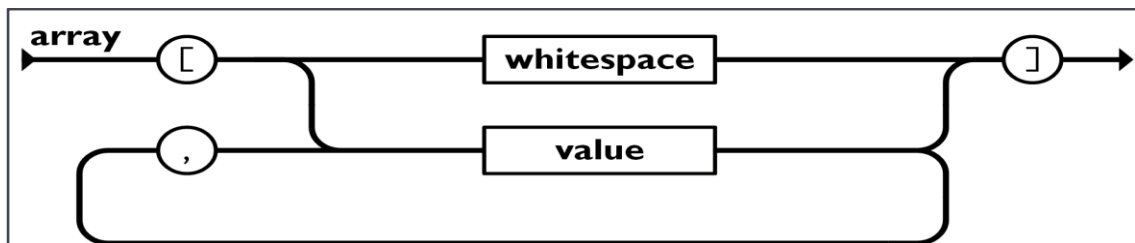


Figura 5 JSON ARRAY

Fonte: JSON (*JavaScript Object Notation*)

Um valor (*value*, na imagem acima) pode ser uma cadeia de caracteres (*string*), ou um número, ou “*true*” ou “*false*”, ou “*null*”, ou um objeto ou uma *array*. Estas estruturas podem estar aninhadas como podemos notar na Figura 6.

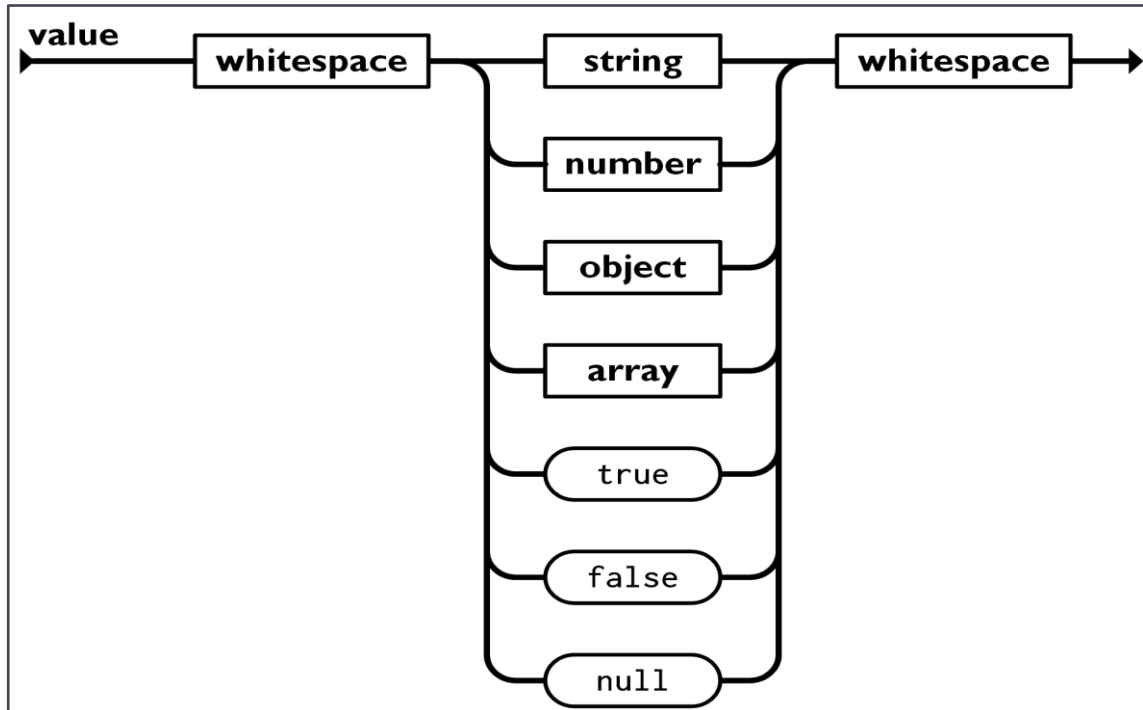


Figura 6 JSON VALUE

Fonte: JSON (*JavaScript Object Notation*)

Nesse caso, JSON (*JavaScript Object Notation*) é um modelo para armazenamento e transmissão de informações no formato texto que facilita o acesso aos *Web Services*. Apesar de muito simples, tem sido bastante utilizado por aplicações *Web* devido a sua capacidade de estruturar informações de uma forma bem mais resumida ao invés do modelo XML. Isto o torna mais rápido no “*parsing*” (parseamento entre as camadas *Web Services*) dessas informações.

Como foi dito, ele é acessível à maioria das linguagens de programação, e PHP é uma dessas linguagens.

Na aplicação específica do sistema Conexão Acará ele atua como um *Web Services RESTful* usado para recuperar e atualizar dados armazenados em um servidor.

Por exemplo, a aplicação pode exibir uma lista de produtos recuperados do banco de dados para poder ser usado em outro serviço da web fora da plataforma. O *Web Services* retornaria os dados no formato JSON, a aplicação analisa se os dados estão corretos e entrega para o usuário final. Um uso bastante similar é aplicativo de comércios eletrônicos e cotação de moedas.

3. METODOLOGIA DA CONSTRUÇÃO

Neste tópico será abordado o processo de desenvolvimento do sistema, bem como as tecnologias utilizadas, linguagem de código (programação), casos de uso, especificações técnicas e conjunto de métodos e procedimentos que compõem todo o sistema da plataforma Conexão Acará.

3.1 TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO

3.1.1 Ambiente de Desenvolvimento Integrado

Temos uma grande variedade de IDE`s (*Integrated Development Environments*), ou Ambiente de Desenvolvimento Integrado, trata-se de um aplicativo de software que fornece um conjunto abrangente de ferramentas para o desenvolvimento de software. Essas ferramentas podem incluir um editor de código, um compilador ou interpretador, um depurador e outros recursos, como integração de controle de versão, gerenciamento de projeto e navegação de código.

O Ambiente de Desenvolvimento Integrado também pode melhorar muito a eficiência e a produtividade de um desenvolvedor de *software*, fornecendo um local centralizado para todos os desenvolvedores da aplicação, nele os programadores podem obter ferramentas e recursos necessários para o desenvolvimento. Além disso, geralmente oferecem recursos como conclusão de código, verificação de erros e depuração, o que pode ajudar a reduzir o tempo e o esforço necessários para identificar e corrigir erros no código.

No geral, IDE 's podem auxiliar muito no processo de desenvolvimento de software e ajudar a garantir que o produto final seja de alta qualidade.

3.1.2 IDE Visual Studio Code®

Neste projeto de software, tema deste trabalho, foi usado a *IDE Visual Studio Code®*.

Ela foi lançada em 2015 pela Microsoft Corporation® como um editor de código destinado ao desenvolvimento de aplicações web chamado de *Visual Studio Code®*, ou simplesmente *VSCode®*. (WIKIPEDIA, VISUAL_STUDIO_CODE 2023).

Anunciada durante o *build*, evento voltado a desenvolvedores que ocorre nos Estados Unidos anualmente, trata-se de uma ferramenta leve e multiplataforma que está disponível tanto para *Windows*®, quanto para *Mac OS*® e Linux e atende a uma gama enorme de projetos, não apenas *ASP.NET*, como também *Node.js*. (*MICROSOFT, BUILD 2015 NEWS*).

Adicionalmente, o editor possui suporte à sintaxe de diversas linguagens como *python*, *ruby*, *c++* e *PHP*.

Foi decidido pelo seu emprego devido à sua natureza de ferramenta de código aberto (onde o código-fonte é acessível a todos) e às amplas vantagens que oferece na identificação de erros em códigos *PHP*, nas sugestões de correções, na depuração e na otimização da legibilidade e desempenho do *software*. Além disso, apresenta facilidade de integração com outros programas necessários para o desenvolvimento, como o *Git*, *GitHub*® ou *GitLab*®.

3.1.3 *Git* - Ferramenta de Versionamento de Código

O objeto de estudo deste trabalho usou uma ferramenta tecnológica chamada *Git*. Os próximos parágrafos descrevem a ferramenta e sua importância em um projeto de construção de *software*.

O *Git* é um sistema de controle de versão distribuído gratuito e de código aberto. Projetado para lidar com uma grande variedade de *software*, desde projetos pequenos ou extremamente grandes, tudo isso com velocidade e eficiência, ele permite aos desenvolvedores rastrear as alterações feitas no código-fonte de um projeto de *software* de maneira fácil e organizada.

Dentre suas inúmeras vantagens, podemos citar: a colaboração e coordenação entre os membros da equipe, capacidade de reverter facilmente para versões anteriores do código, se necessário. Ajuda a manter um histórico das alterações feitas no código, facilitando a identificação e correção de bugs.

Além disso, o *Git* permite que os desenvolvedores ramifiquem a base de código, onde podem desenvolver novos recursos ou corrigir bugs sem afetar a ramificação de desenvolvimento principal. (*GIT, 2023*).

Por todas essas vantagens citadas acima, o *Git* é uma ferramenta indispensável na construção de qualquer software.

3.1.4 Linguagens de marcação, de script e de programação

Utilizam-se neste projeto as linguagens mais indicadas para a construção de aplicativos *web* dinâmicos, tais como: PHP, HTML, CSS e *Javascript*, cuja a descrição segue nos próximos parágrafos.

PHP (originalmente significava *Personal Home Page*, porém de acordo com a convenção para atribuição de nomes recursiva passou a significar PHP *Hypertext Pre-processor*) é uma linguagem de script *open-source*, de uso geral, muito utilizada e especialmente para o desenvolvimento de aplicações *web* embutida dentro do HTML. Verifica-se que as características da linguagem tendem a cair em 4 (quatro) categorias chave: praticidade, poder, possibilidade e preço (é disponibilizado gratuitamente), (GILMORE, 2020).

O PHP também possui uma variedade de funções e bibliotecas integradas que podem ser usadas para executar tarefas comuns, como conectar-se a um banco de dados, enviar e-mail e trabalhar com arquivos.

Além disso, existem muitas bibliotecas e estruturas de terceiros disponíveis, como *Laravel*, *Symfony*, *CodeIgniter* e *Adianti*, que podem ser usadas para acelerar o desenvolvimento e melhorar a estrutura do código, tornando-o mais robusto e completo. (DALL’OGLIO, 2009).

Depois que os scripts PHP são criados, eles podem ser executados no servidor da *web* e a saída pode ser exibida em um navegador da *web*.

Devido a estas várias características vantajosas da tecnologia PHP, resolveu-se implantá-la neste projeto de software na parte, principalmente, de “*back-end*”. Atuando nas camadas de Controles, Modelos e também de Visualização.

Na Camada de Controle, ele atua como funções de validação de dados fornecidos pelo usuário nos campos de formulários, tais como login, cadastro, registros de produtos, exclusão de produtos, e exclusão de dados cadastrados no banco de dados referentes ao próprio usuário.

Na Camada de Modelo, ele atua com requisições ao banco de dados, fazendo chamadas e emitindo estas chamadas ao controlador como sugere o padrão MVC já descrito anteriormente.

E por último, trata das entregas das informações e dados do usuário na página de Exibição ou de Visão, nesta fase o PHP atua conjuntamente com a tecnologia “*front-end*”, HTML e CSS, sendo, portanto, um intermediador entre o mundo externo e os dados gerados e guardado no banco de dados da aplicação desenvolvida.

Ainda sobre a etapa de Visualização, é nela que os dados são inseridos pelo usuário para poderem ser tratados e gravados no banco de dados.

A tecnologia HTML é um acrônimo para *HyperText Markup Language*, cuja tradução para o português é Linguagem para Marcação de Hipertexto, (SILVA, 2014).

No contexto deste trabalho, hipertexto pode ser entendido como todo o conteúdo inserido em um documento para a web e tem como principal característica a possibilidade de interligação a outros documentos lá existentes.

A linguagem *Javascript* nasceu da necessidade de prover interatividade às páginas eletrônicas, uma vez que a HTML não dispunha de elementos capazes de disponibilizar tal funcionalidade. Uma parceria da *Netscape* com a *Sun Microsystems* possibilitou seu lançamento em 1995.

Javascript é uma linguagem desenvolvida para rodar no lado cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. (SILVA, 2010). Esta possibilidade é fornecida pelo interpretador *javascript* hospedado nesse navegador.

Cascading Style Sheets (CSS), cuja tradução é Folhas de Estilo em Cascata, também são definidas e padronizadas pelo W3C. Assim, cabe às CSS todas as funções de apresentação de um documento de forma agradável ao usuário final na web.

A importância do uso da tecnologia CSS na construção de um software é que ela permite a separação do conteúdo da forma como ele é apresentado, o que facilita a manutenção e atualização do design do software. Diante disso, o uso de CSS

permite a reutilização de estilos em várias páginas, o que economiza tempo e esforço na criação de *layouts*.

Ele também permite que o desenvolvedor crie *design* responsivo, permitindo que o *layout* se adapte a diferentes tamanhos de tela, tornando a aplicação acessível em diferentes dispositivos.

Essa tecnologia é usada no desenvolvimento da plataforma Conexão Acará, principalmente para alcançar um dos objetivos específicos que é tornar o *software* facilmente acessível às comunidades rurais, projetando-o para ser compatível com dispositivos móveis. Já que a grande maioria das pessoas utiliza esses dispositivos como única forma de acesso à *internet*.

3.1.5 SGBD (Sistemas Gerenciadores de Banco de Dados) *MySQL*

A figura 7 representa um esquema de Sistema de Banco de Dados, onde podemos perceber as etapas da informação.

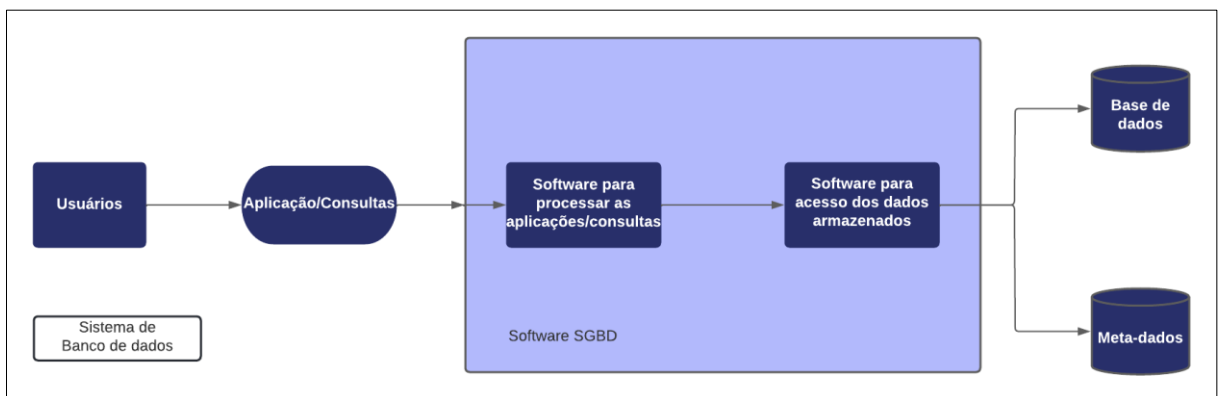


Figura 7 SISTEMA DE BANCO DE DADOS

Fonte: Elaboração do autor

O Sistema de Gerenciamento de Banco de Dados (SGBD) *MySQL* “é um servidor de banco de dados relacional que se aproxima cada vez mais das facilidades encontradas em produtos proprietários competitivos a cada versão”. (GILMORE, 2020).

O SGBDR adotado no sistema deste objeto de trabalho, foi o *MySQL*. Ele é um popular sistema de gerenciamento de banco de dados de código aberto amplamente usado para gerenciar e organizar dados em aplicativos da *Web* e móveis.

É conhecido por sua confiabilidade, robustez e facilidade de uso. O SGBDR usa SQL (*Structured Query Language*) para gerenciar e consultar dados e pode ser usado em combinação com linguagens de programação como PHP, *python* e *java* para criar aplicativos da *Web* dinâmicos. Também é uma escolha popular para aplicativos *web* de armazenamento de dados.

A popularidade é atribuída à sua gratuidade para aplicações não comerciais e às seguintes características por ele apresentadas: flexibilidade, capacidade, recursos de SQL a nível de empresa, indexação e busca “*full-text*”, “*query caching*”, replicação e segurança. (FERRARI, 2007).

No sistema proposto o *MySQL* atua em conjunto com a linguagem *PHP*, armazenando os dados inseridos pelo usuário respeitando a segurança da informação é baseada em três diferentes atributos que constituem os seus pilares: confidencialidade, disponibilidade, integridade.

O conceito de confidencialidade se refere à proteção de informações que não devem ser acessadas por indivíduos não autorizados.

Isso significa dizer que, literalmente, determinadas informações são confidenciais e só dispõem de seu acesso aqueles que possuem autorização para tal.

Integridade está relacionado à plenitude do armazenamento dos dados, isto é, da mesma forma que as informações são fornecidas, elas devem ser armazenadas, sem qualquer alteração em seu conteúdo.

Disponibilidade diz respeito ao acesso dos dados sempre que este for necessário, isto é, significa, literalmente, a garantia da disponibilidade das informações.

Esse princípio está diretamente relacionado à eficácia do sistema e do funcionamento do mesmo para que, conseqüentemente, a informação possa ser acessada quando for necessário.

4. ESPECIFICAÇÃO DO SISTEMA

4.1 DIAGRAMA DE CASOS DE USO

Um diagrama de caso de uso é um tipo de diagrama UML (*Unified Modeling Language*) usado para representar as interações entre atores e as ações que eles executam em um sistema. Um exemplo é o caso de uso de login, exibido na Figura 6, especificando as interações e ações relacionadas ao processo de login no sistema.

A UML é a sucessora da linguagem de modelagem encontrada nos métodos *Booch*, *Jacobson* e outros como Modelos de Entidades e Relacionamentos. Durante 1996 a *Rational Software Corporation* contou com a participação de outras Instituições parceiras como: *Hewlett-Packard*, *IBM*, *Microsoft*, *Oracle*, *Unisys*, *Platinum Technology*, etc. Essa colaboração contribuiu para a produção da versão 1.0, uma versão bem definida, expressiva, poderosa e aplicável, a qual foi submetida a *OMG (Object Management Group)* para adoção.

Em setembro de 1997 liberaram a versão 1.1 da UML e em dezembro a mesma foi aprovada como padrão pela *OMG*.

Diagrama de caso de uso normalmente inclui atores como usuário, administrador do sistema e serviço de autenticação, bem como casos de uso como "Inserir credenciais de *login*", "Verificar credenciais" e "Conceder acesso". O diagrama também mostraria os relacionamentos entre os atores e os casos de uso, como o usuário iniciando o caso de uso "Inserir credenciais de *login*" e o administrador do sistema gerenciando o caso de uso "Verificar credenciais".

O sistema web *Conexão Acará* foi concebido para implementar os casos de uso, ditos como primários, estes são listados a seguir:

- **Efetuar *login***: necessário para poder entrar no sistema. Trata da entrada e captura dos dados do usuário com seu e-mail e senha para realizar a requisição ao *Web Services*. Este vai analisar os dados enviados e validá-los, permitindo ou não a entrada no sistema.

- **Cadastrar produto:** serve uma *interface* para o usuário efetuar o cadastramento de produtos, com seus respectivos preços e quantidade disponíveis.
- **Atualizar produto:** disponibilizar a lista de produtos cadastrados para o usuário fazer a seleção e em seguida efetuar as devidas modificações.
- **Remover produto:** remove um produto selecionado pelo usuário a partir da lista de exibição de produtos.
- **Exibir produtos:** a partir da consulta ao banco de dados, realiza uma busca nos produtos cadastrados e mostra ao usuário do sistema e também aos usuários fora dele.

A seguir temos a Figura 8 do diagrama de caso de uso, onde podemos ver o usuário com a efetuação do *login*.

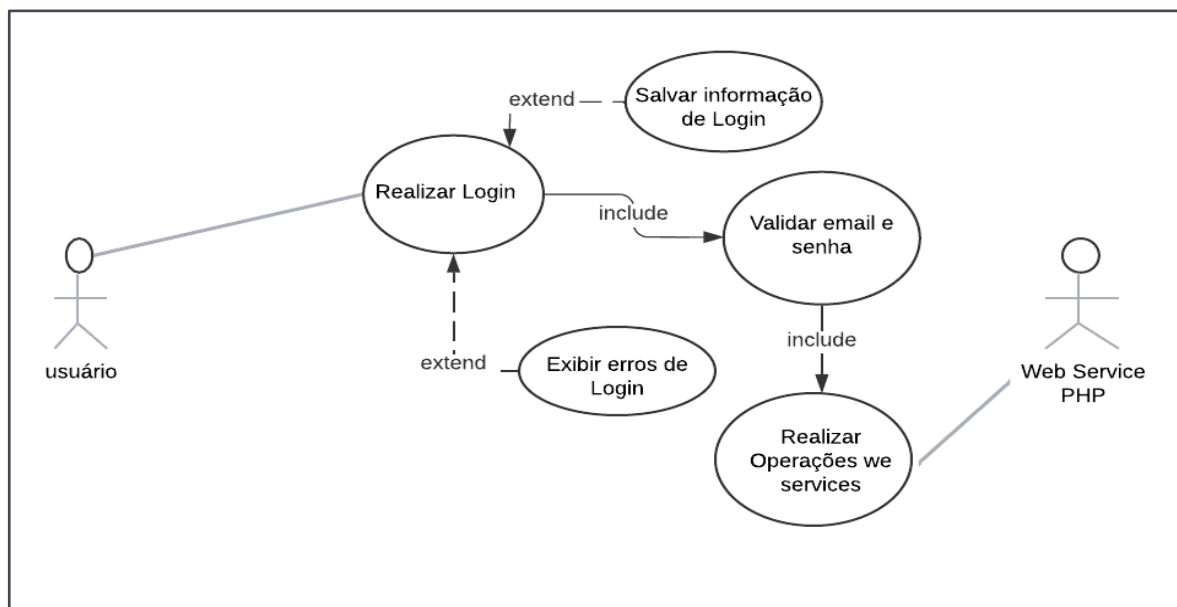


Figura 8 - DIAGRAMA DE CASO DE USO REALIZAR *LOGIN*

Fonte: Elaboração do autor

SISTEMA:

- Nome do Caso de Uso: realizar *Login*
- Ator Principal: usuário
- Ator Auxiliar: *Web Services* PHP.

- Breve Descrição: autentica o usuário no sistema dando a ele acesso às informações de sua conta, bem como às funcionalidades de cadastrar, alterar e deletar produto, bem como visualizar seus produtos cadastrados, etc.
- Pré-condições: o usuário tenha feito o registro do e-mail e senha pela plataforma e tenha conexão com a internet.
- Pós-condições: o usuário é logado no sistema e tem acesso à sua conta.

A seguir temos a Figura 9 do diagrama de caso de uso, onde podemos ver o usuário com cadastro do produto.

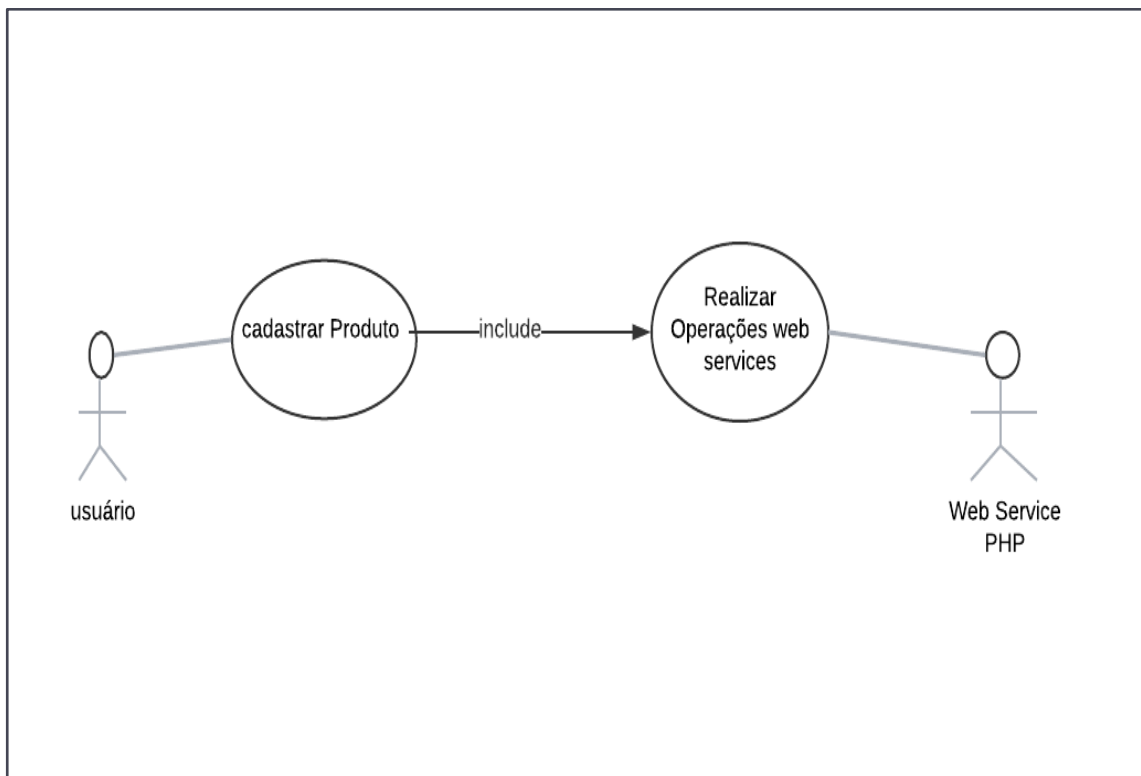


Figura 9 - DIAGRAMA DE CASO DE USO CADASTRAR PRODUTO

Fonte: Elaboração do autor

SISTEMA:

- Nome do Caso de Uso: cadastrar produto.
- Ator Principal: usuário.
- Ator Auxiliar: *Web Services* PHP.

- Breve Descrição: permite ao usuário cadastrar um novo produto.
- Pré-condições: o usuário deve estar conectado à *internet*. O usuário deve estar logado no sistema.
- Pós-condições: o usuário consegue realizar o cadastro. Cadastro realizado!

A seguir temos a Figura 10 do diagrama de caso de uso, onde podemos ver o usuário com a atualização do produto.

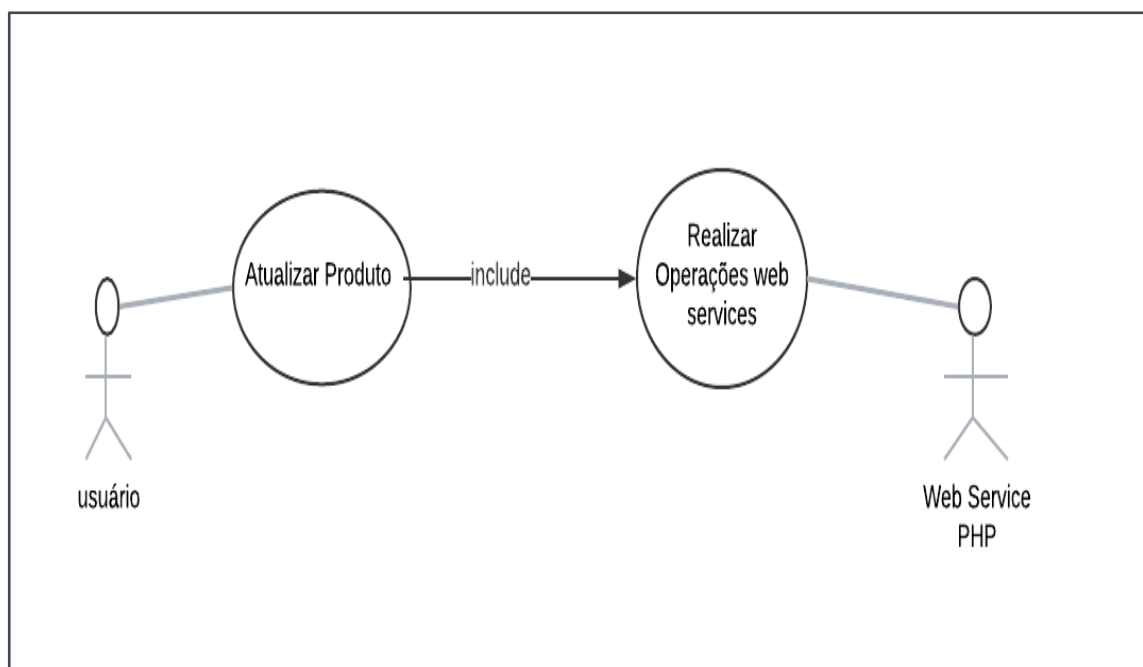


Figura 10 - DIAGRAMA DE CASO DE USO ATUALIZAR PRODUTO

Fonte: Elaboração do autor

SISTEMA:

- Nome do Caso de Uso: atualizar produto
- Ator Principal: usuário
- Ator Auxiliar: *Web Services* PHP.
- Breve Descrição: permite ao usuário atualizar um produto previamente cadastrado.
- Pré-condições: o usuário deve estar conectado à internet. O usuário deve estar logado no sistema.

- Pós-condições: o usuário consegue atualizar o produto. Produto alterado com sucesso!

A seguir temos a Figura 11 do diagrama de caso de uso, onde podemos ver o usuário com a exclusão do produto.

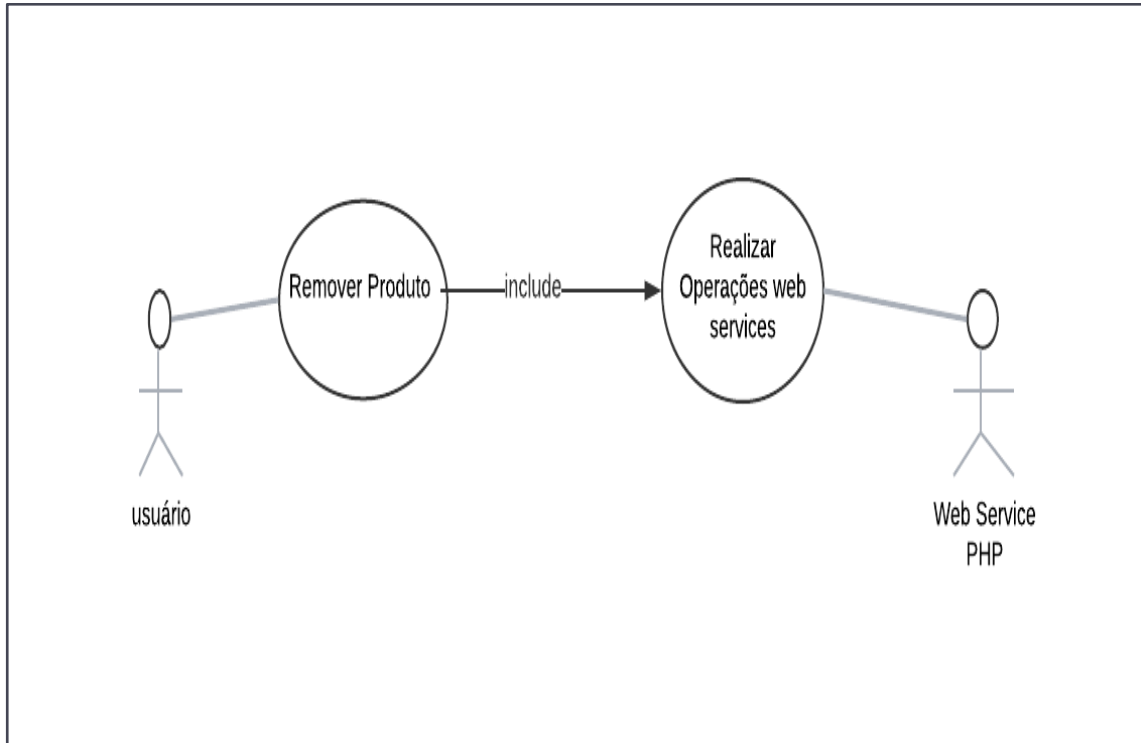


Figura 11 - DIAGRAMA DE CASO DE USO DELETAR PRODUTO

Fonte: Elaboração do autor

SISTEMA:

- Nome do Caso de Uso: deletar um produto.
- Ator Principal: usuário.
- Ator Auxiliar: *Web Services PHP*.
- Breve Descrição: permite ao usuário deletar um produto previamente cadastrado.
- Pré-condições: o usuário deve estar conectado à internet. O usuário deve estar logado no sistema.
- Pós-condições: o usuário consegue deletar o produto. Produto deletado com sucesso!

A seguir temos a Figura 12 do diagrama de caso de uso, onde podemos ver o usuário com a exibição do produto.

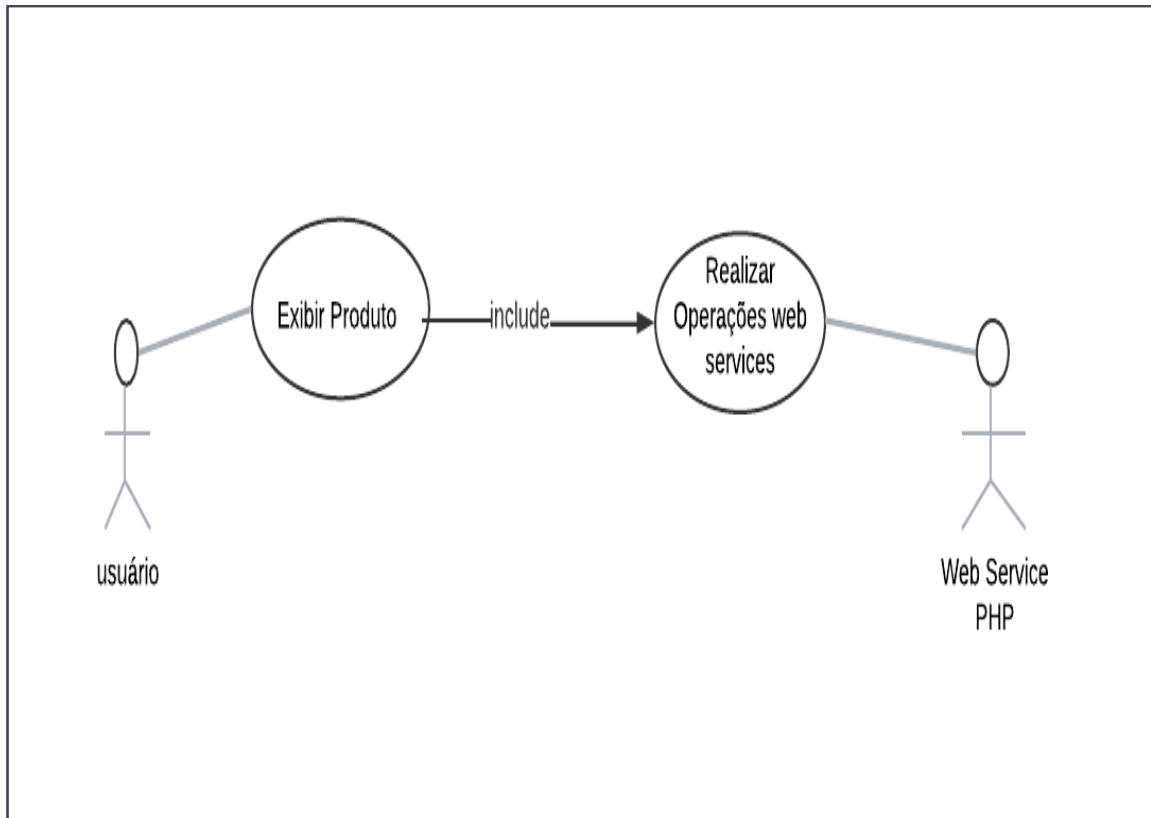


Figura 12 - DIAGRAMA DE CASO DE USO EXIBIR PRODUTO

Fonte: Elaboração do autor

SISTEMA:

- Nome do Caso de Uso: exibir um Produto.
- Ator Principal: usuário.
- Ator Auxiliar: *Web Services PHP*.
- Breve Descrição: permite ao usuário ver um produto previamente cadastrado.
- Pré-condições: o usuário deve estar conectado à *internet*. O usuário deve estar logado no sistema.
- Pós-condições: o usuário consegue ver o (s) produto (s). Produto exibido com sucesso!

4.2 Engenharia de Requisitos

A Engenharia de Requisitos é extremamente importante para garantir de os processos e suas definições juntamente com a documentação sejam devidamente cumpridos. Além de ajudar os profissionais envolvidos na construção a seguirem a caminho correto para alcançar o produto desejado.

Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações, (SOMMERVILLE, 2011).

Os requisitos de software são frequentemente classificados como requisitos funcionais e requisitos não funcionais:

4.2.1 Lista de Requisitos Não-Funcionais.

Os requisitos não funcionais estão relacionados ao que o sistema tem de oferecer em termos de qualidades, usabilidades, etc... A seguir serão listados os requisitos não-funcionais do sistema em questão:

- Para cadastrar um produto, é exigido que os usuários façam login com suas credenciais (senha e e-mail), assim a plataforma pode verificar a identidade do usuário antes de conceder acesso a este e a outros recursos específicos;
- O *software* Conexão Acará permiti que as senhas sejam criptografadas ao inserir no banco de dados. Isso reduz o risco de acessos não autorizados, protege informações sensíveis e evita atividades maliciosas;
- O *software* Conexão Acará segue os Requisitos de Legais - LGPD — Lei Geral de Proteção de Dados. Isso significa que não coletamos dados sensíveis do usuário, tais como identidade de gênero ou religião.
- São requisitos Não-Funcionais do *software* Conexão Acará: Alta-disponibilidade, interoperabilidade, escalabilidade, desempenho, usabilidade, facilidade e compatibilidade.

4.2.2 Lista de Requisitos Funcionais.

Os requisitos funcionais da Engenharia de Requisitos nada mais são do que as funcionalidades de um determinado *software*, ou seja, o que ele faz. A seguir serão listados a funcionalidades do sistema Conexão Acará:

- O *software* Conexão Acará permiti que o cadastro de clientes e/ou produtores seja efetuado;
- O *software* Conexão Acará permiti que os produtores possam cadastrar seus produtos agrícolas para vendas;
- O *software* Conexão Acará permiti que o usuário possa filtrar produtos rurais por categoria (farinha, semente, pimentas e frutos).

5. ESTRUTURA DO SISTEMA PROPOSTO

Trataremos nesta seção da utilização dos casos de uso, de como o sistema se comporta e demonstra sua função de acordo com os objetivos propostos.

Utilizar a *POO* para criar estruturas modulares e expansíveis em *PHP*, permitindo a construção de sistemas mais robustos e escaláveis. Isso enfatiza a importância de projetar classes bem estruturadas e seguir os princípios de design orientado a objetos, como o princípio da responsabilidade única (*SRP*) e o princípio aberto/fechado (*OCP*), para facilitar a manutenção e a evolução do código ao longo do tempo. (DALL'OGGIO, 2009).

Este projeto utilizou-se de base teórica para compor e estruturar todas os diretórios e arquivos necessários para construção da aplicação

A Figura 13 apresenta a estrutura do sistema desenvolvido com a intenção de mostrar a disposição das classes dos controles, modelos e a disposição das telas conforme o padrão *MVC* sugere.

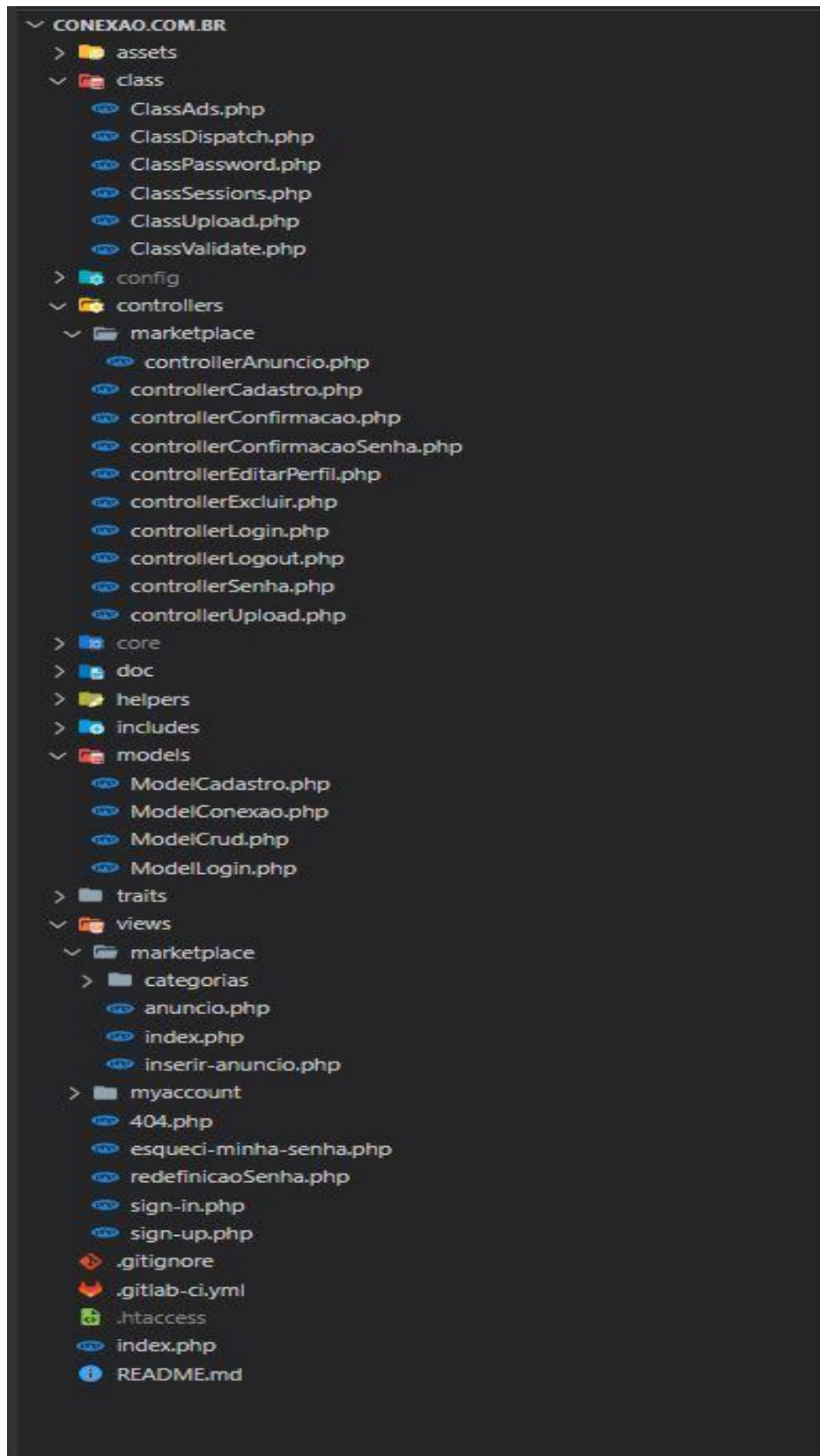


Figura 13 - ESTRUTURA DE DIRETÓRIOS

Fonte: Elaboração do autor

5.1 REALIZAR LOGIN

As funcionalidades de cadastrar, atualizar e deletar produtos são internas do sistema, o que significa que o usuário que deseja fazê-la tem que estar devidamente registrado e ter feito a autenticação nele.

Para a autenticação é necessário informar o e-mail e a sua senha, previamente cadastrados como mostrados na Figura 14.

A imagem mostra a interface de login de um sistema. No topo, o título "Entrar" está centralizado e sublinhado com uma linha azul. Abaixo dele, há dois campos de entrada de texto amarelos: o primeiro contém o e-mail "isaque-menezes@hotmail.com" e o segundo contém dez pontos pretos para mascarar a senha. Abaixo dos campos, há um botão de login azul arredondado com o texto "Entrar" em branco. Na base da tela, há dois links em texto vermelho: "esqueci minha senha" à esquerda e "Cadastrar-se" à direita.

Figura 14 - TELA REALIZAR LOGIN

Fonte: Elaboração do autor

Todo processo de autenticação e demais casos, são tratados e higienizados (processo de verificação de dados informados no campo específico solicitado) pelo sistema. Na Figura 15 a seguir, o usuário digita de forma errada um dos campos solicitados no *login*.

The image shows a login interface with the following elements:

- Header: "Entrar" with a blue underline.
- Error message: "Usuário ou Senha Inválidos!" in red text.
- Email input field: Contains "isaque-menezes@hotmail.com".
- Password input field: Contains 10 black dots.
- Login button: A blue rounded rectangle with the text "Entrar".
- Links: "esqueci minha senha" and "Cadastrar-se" in red text.

Figura 15 - TELA DE CONSISTÊNCIA DA AUTENTICAÇÃO

Fonte: Elaboração do autor

Por questão de segurança não é informado se ele errou o campo e-mail ou campo senha.

Em todo esse processo de validação e autenticação o *Web Services* está envolvido. Ele é a principal ferramenta que conecta vários outros serviços e faz a comunicação direta através de mensagem para o mundo externo.

Sendo que o login segue a seguinte estrutura de tarefas. O diagrama da Figura 16 mostra todo esse processo:

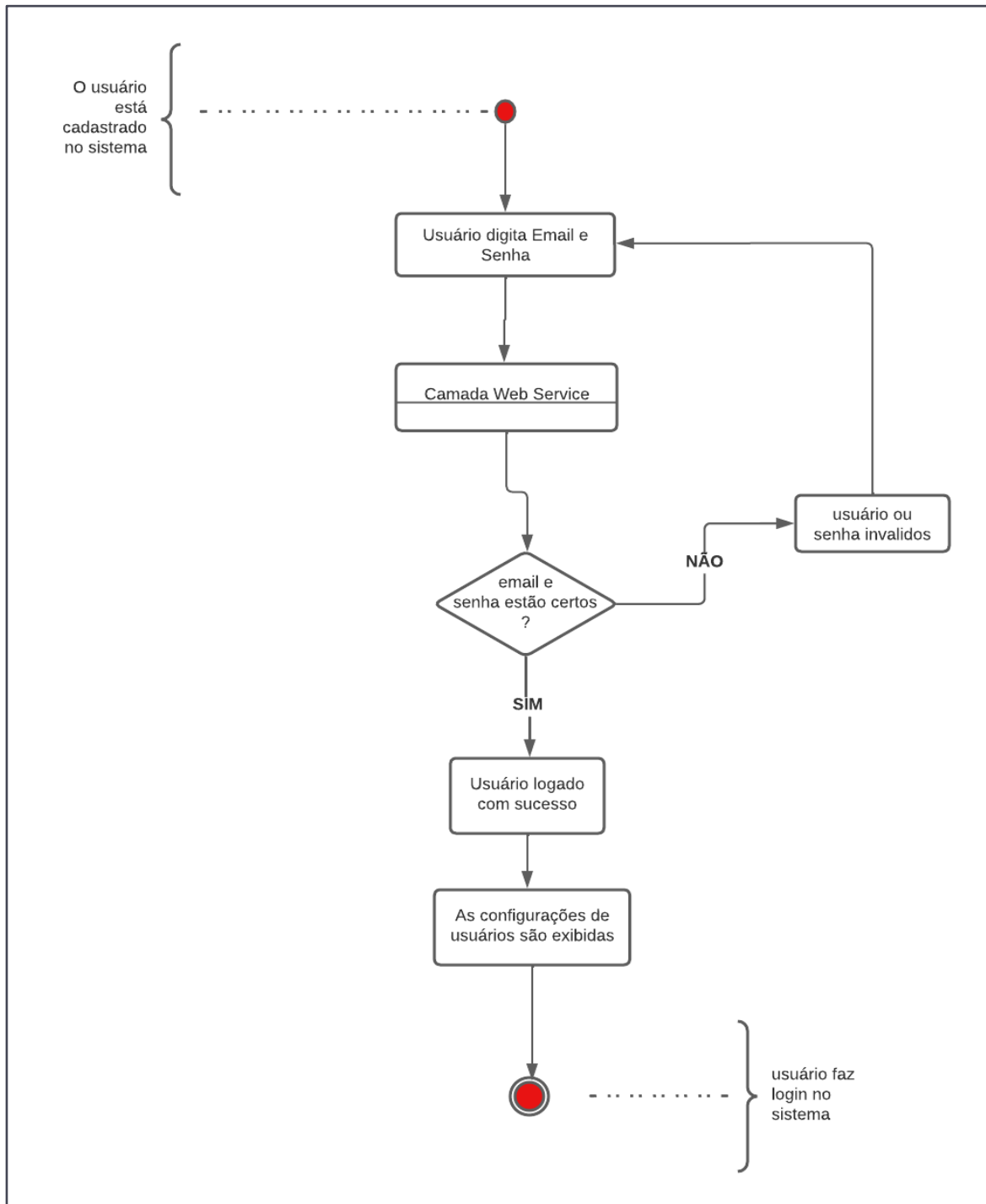


Figura 16 - DIAGRAMA DE ESTADOS REALIZAR *LOGIN*

Fonte: Elaboração do autor

5.2 CADASTRAR ANÚNCIO

Subsequente ao realizar o *login*, o usuário poderá ter acesso ao seu painel de usuário - denominado conta, com o menu de mesmo nome. Já dentro da sua conta, ele conta com o menu de contexto listado abaixo, onde na opção “Meus Anúncios”

tem outra opção de “Novo Anúncio”, sendo este a opção de cadastrar um novo produto.

Para efetuar o cadastro de um novo produto o sistema busca no banco de dados e lista os anúncios cadastrados, sendo que estes não podem ultrapassar o valor de 3 anúncios cadastrados pelo mesmo usuário.

A Figura 17 mostra o menu de contexto onde tem a opção de cadastrar um novo anúncio.



Figura 17 - MENU DE CONTEXTO

Fonte: Elaboração do autor

Ao acessar a opção de novo anúncio, é realizado uma requisição *Web Services* ao sistema para obter e redirecionar o usuário para a tela com sua interface, exibindo-a para que ele possa realizar a inserção dos dados necessários. Nela serão solicitadas as seguintes informações:

- Título:

Trata-se do título do anúncio (produto) que será exibido na plataforma para ser visualizado por visitantes da mesma

- Descrição:

O usuário descreve com mais detalhes o que está anunciando, e quais as características do seu produto

- Preço:

O valor em moeda corrente oficial do seu país do produto

- Categoria:

Opção com uma lista pré-disponível de categoria de produto, o usuário escolhe qual a melhor se enquadra no seu produto

- Arquivos de Mídias:

O usuário pode enviar fotos do seu produto nos formatos png, jpeg.

A Figura 18 seguinte exemplifica uma tela de requisição de dados para um cadastro de um novo anúncio.

Cadastrar Novo Anúncio

Título

Descrição

Preço

Categoria

Nenhum arquivo selecionado. Enviar arquivos

Figura 18 - TELA CADASTRAR NOVO ANÚNCIO

Fonte: Elaboração do autor

O processo de cadastrar produto tem seu diagrama de estado listado na Figura 19.

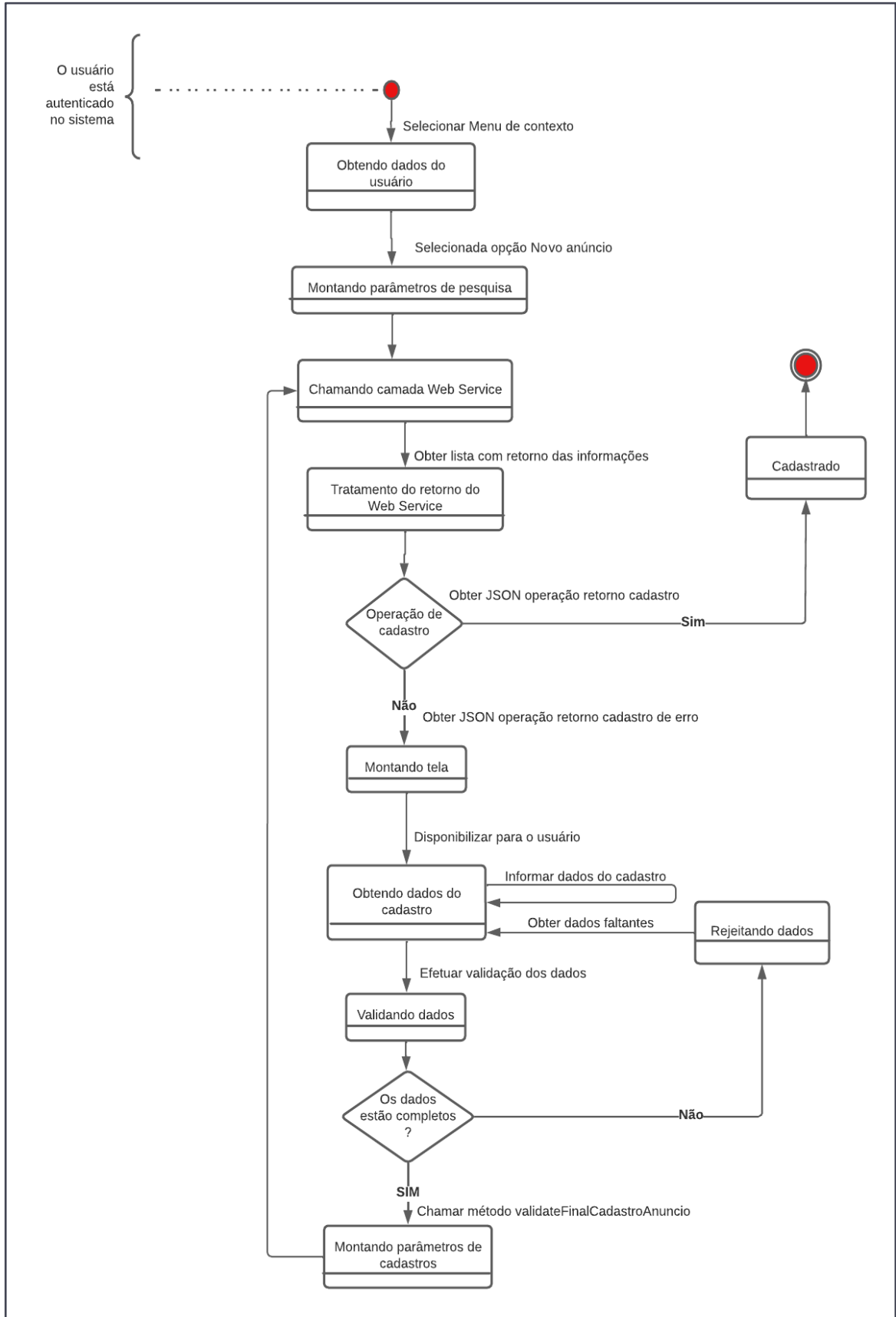


Figura 19 - DIAGRAMA DE ESTADOS CADASTRAR ANÚNCIO

Fonte: Elaboração do autor

5.3 ATUALIZAR ANÚNCIO

Tomando como base as informações disponíveis na página “meus-anuncios” o usuário tem a opção de fazer a atualização do anúncio de seus produtos.

Ao pressionar o botão cujo nome é “Editar”, o sistema o redireciona para a página “editar-anuncio” ao qual faz uma requisição ao banco de dados para exibir as informações cadastradas do anúncio em questão para o usuário realizar as devidas correções.

A Figura 20 mostra o caso de atualizar o anúncio.

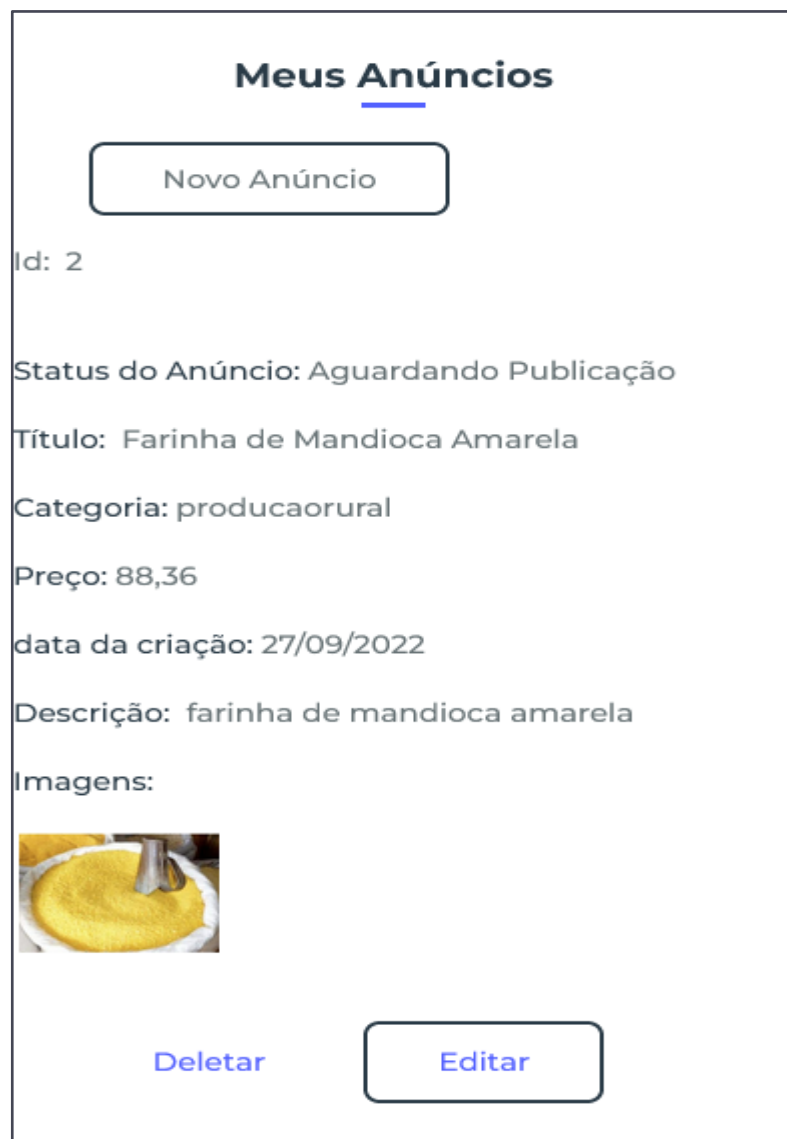


Figura 20 - MENU ATUALIZAR ANÚNCIO

Fonte: Elaboração do autor

Após a indicação de que o usuário deseja editar o anúncio selecionado, o sistema efetua a consulta dos detalhes desse anúncio utilizando o *Web Services*, informando qual o código do anúncio.

O *Web Services* realiza a consulta do anúncio específico e devolve as informações para que sejam apresentadas em tela a fim de que o usuário possa realizar a alteração dos dados.

A Figura 21 demonstra a tela utilizada para edição de um anúncio, bem como o momento em que o usuário é informado que a operação foi realizada com sucesso e a alteração já consta no sistema.

A imagem mostra a interface de usuário para a edição de um anúncio. No topo, há uma barra azul com o texto "ConexaoAcara" e um ícone de menu. O título centralizado é "Editar Anúncio".

Os campos de formulário são:

- Título:** Um campo de texto contendo "Farinha de Mandioca Amarela".
- Descrição:** Um campo de texto contendo "farinha de mandioca amarela".
- Preço:** Um campo de texto contendo "88,36".
- Categoria:** Um menu suspenso com o valor selecionado "producaorural".

Na base da tela, há um botão "Procurar..." e o texto "Nenhum arquivo selecionado. Enviar arquivos". Abaixo disso, um botão "Editar" está disponível.

Figura 21 - TELA ATUALIZAR ANÚNCIO

Fonte: Elaboração do autor

Na Figura 22 abaixo o sistema informa ao usuário que a operação de edição foi realizada com sucesso!

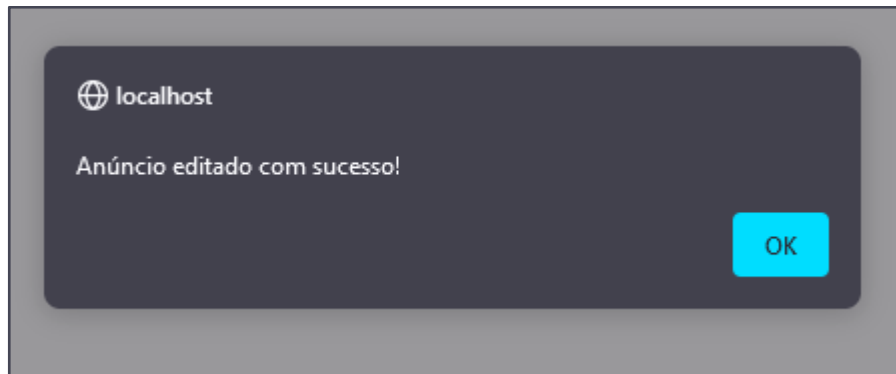


Figura 22 - MENSAGEM ANÚNCIO ATUALIZADO

Fonte: Elaboração do autor

A Figura 23 representa o diagrama de estados que demonstra como ocorre o processo de edição de um anúncio.

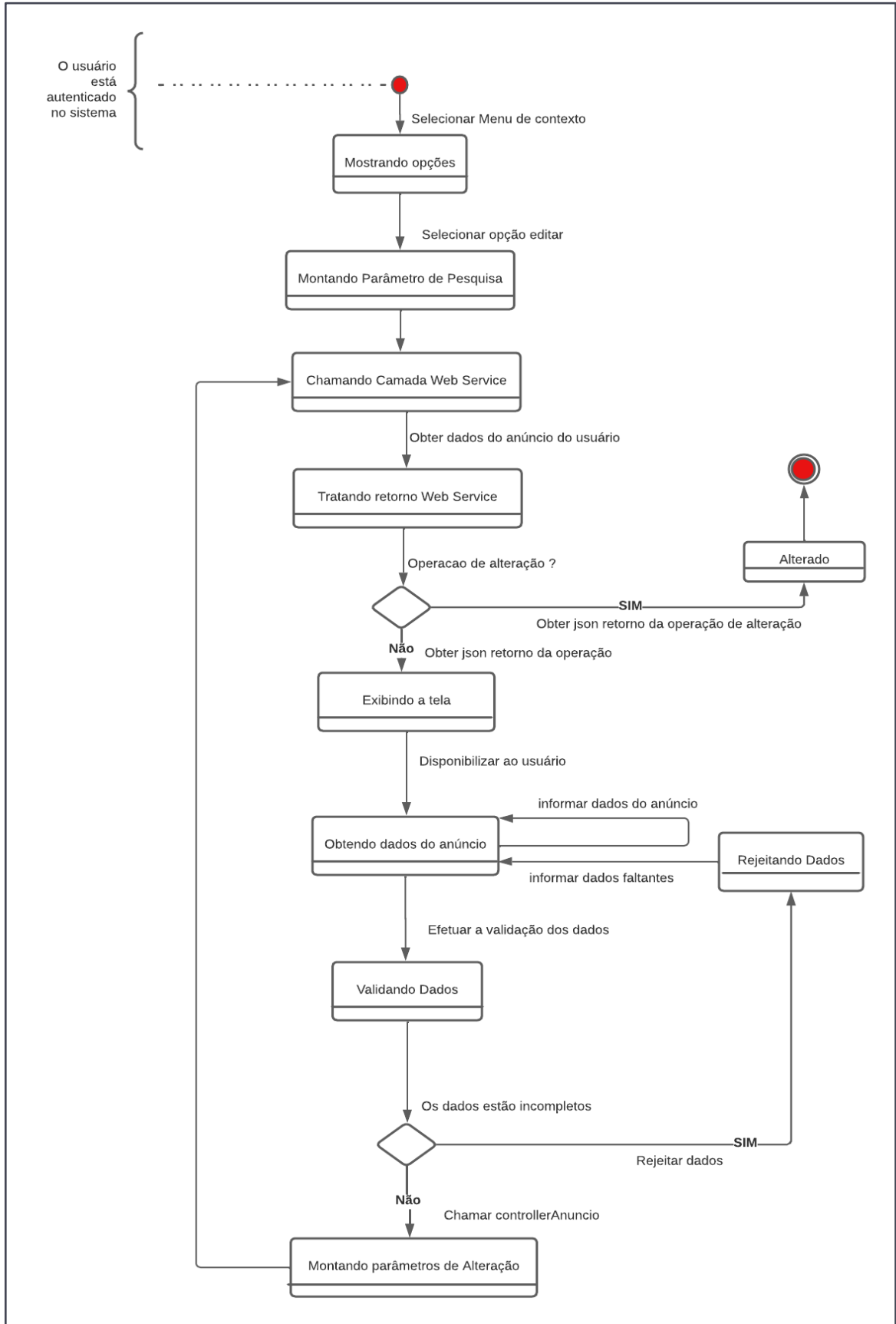


Figura 23 - DIAGRAMA DE ESTADOS EDITAR ANÚNCIO

Fonte: Elaboração do autor

5.4 REMOVER ANÚNCIO

As informações de anúncios coletados do banco de dados são listadas na conta do usuário, mais especificamente na página “meus-anuncios”. Em cada anúncio específico o usuário do sistema tem a opção de deletar o mesmo com o botão “Deletar”.

A Figura 24 abaixo exemplifica a disponibilidade desta função ao usuário.



Figura 24 - MENU REMOVER ANÚNCIO

Fonte: Elaboração do autor

Assim como a maioria das operações realizadas por um sistema computacional, uma operação de remoção requer que o usuário confirme a ação.

Ao pressionar o botão “Deletar” um *pop-up* (é uma janela que se abre no navegador do usuário para solicitar a ele alguma ação ou exibir alguma informação relevante) do *javascript* solicita a confirmação da ação como mostrado na Figura 25 abaixo.

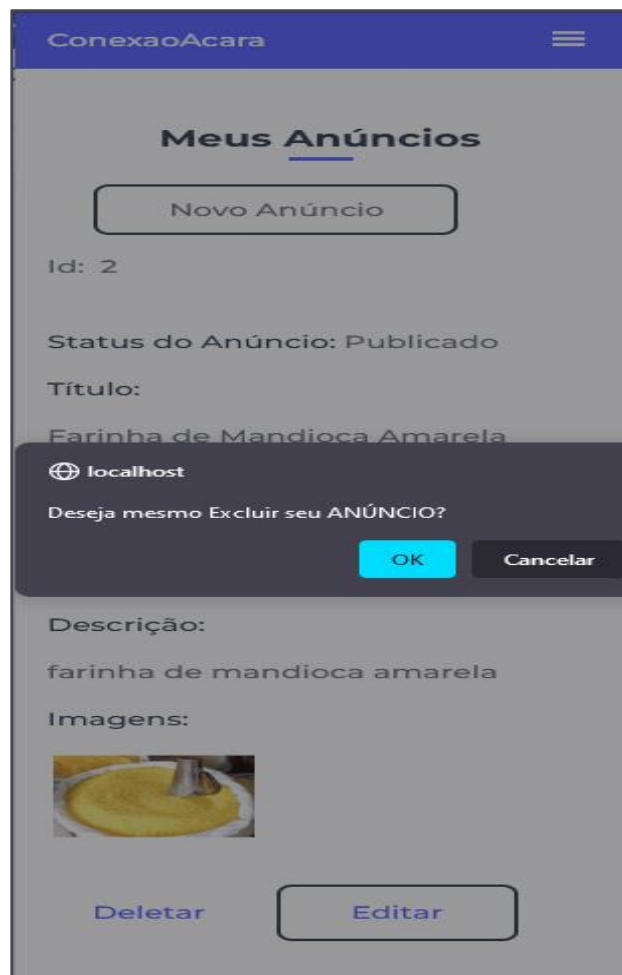


Figura 25 - DIÁLOGO REMOVER ANÚNCIO

Fonte: Elaboração do autor

Caso deseje fazer a remoção, deverá pressionar o botão “OK”; caso contrário deverá pressionar o botão “Cancelar”.

O diagrama de estados de remoção do anúncio é apresentado, logo na Figura 26 a seguir.

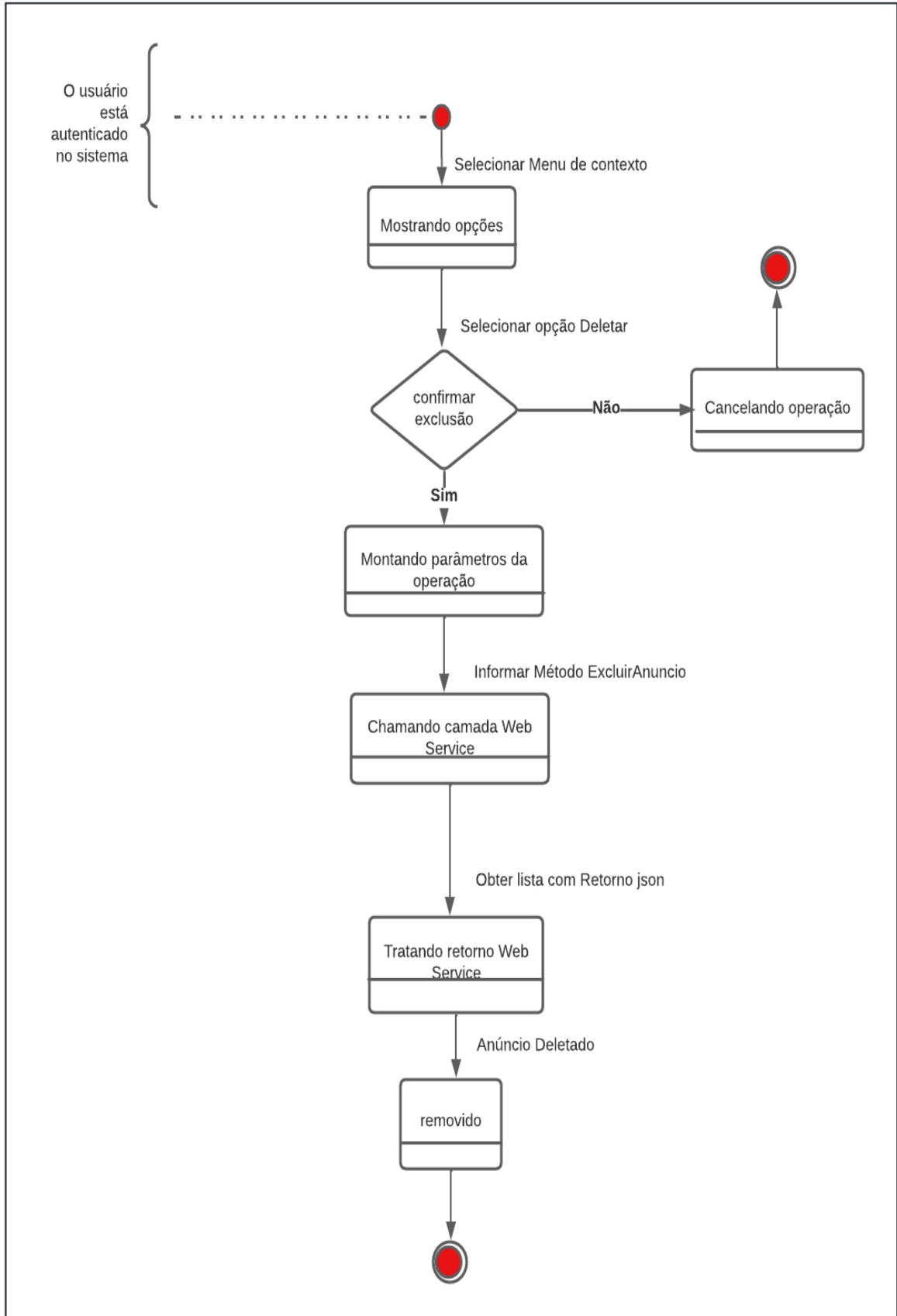


Figura 26 - DIAGRAMA DE ESTADOS REMOVE ANÚNCIO

Fonte: Elaboração do autor

5.5 EXIBIR ANÚNCIO

A forma que os anúncios são exibidos tem duas representações: a primeira é a exibição interna para o usuário logado no sistema. Esta apresenta as opções internas de anúncios tais como editar e deletar.

A Figura 27 representa a tela de exibição interna na página “meus-anuncios” do sistema.



Figura 27 - TELA EXIBIÇÃO ANÚNCIOS CADASTRADOS

Fonte: Elaboração do autor

A segunda forma de representação de exibição do anúncio é a geral, que é exposta na página principal do sistema para que outros usuários possam visualizar informações do produto cadastrado. A Figura 28 representa essa forma de exibição. Ela contém informações essenciais para que o usuário comprador possa tomar a decisão de compra ou não do produto oferecido, ou mesmo possa pesquisar mais produtos e detalhes do mesmo. Também é possível fazer uma pesquisa através de filtros.

Todos os anúncios



Farinha de Mandioca Amarela
Cidade:acara
Bairro:centro
R\$:88,36
Data:27-09-2022 14:51



Pupunha
Cidade:acara
Bairro:zero
R\$:56
Data:01-01-2022 17:28

Figura 28 - TELA EXIBIÇÃO ANÚNCIOS PUBLICADOS

Fonte: Elaboração do autor

6. IMPLANTAÇÃO E INFRAESTRUTURA

6.1 *Gitlab*®

O sistema proposto usa a ferramenta *Gitlab*® para o processo de “*DevOps*” e local onde estão o espelho do código-fonte original, que está hospedado na nuvem. A seguir descreve-se o que é, e como ela fornece um ambiente robusto para o processo de entrega de *software*.

O *Gitlab*® é uma plataforma de hospedagem de código-fontes de programas que fornece todas as ferramentas essenciais de “*DevOps*”. Da ideia à produção, o *Gitlab*® ajuda as equipes a melhorar o tempo de ciclo de semanas para minutos, reduzir os custos de desenvolvimento, acelerar o tempo de lançamento no mercado e fornecer aplicativos mais seguros e compatíveis. (*GITLAB. ABOUT, 2023*).

“*DevOps*” é uma abordagem de desenvolvimento de *software* que enfatiza a colaboração e a comunicação entre as equipes de desenvolvimento e operações para melhorar a velocidade e a qualidade da entrega de software. O objetivo é preencher a lacuna entre essas duas equipes, automatizando e simplificando os processos envolvidos no desenvolvimento, teste, implantação e manutenção de software.

No geral, o “*DevOps*” é uma abordagem importante para organizações que desejam melhorar a velocidade, a qualidade e a confiabilidade de seus processos de entrega de software, além de promover maior colaboração e comunicação em suas equipes. Essa ferramenta é, portanto, indispensável em qualquer projeto e implementação de sistemas de alta qualidade.

6.2 Hospedagem em Nuvem

Uma estrutura de nuvem refere-se à arquitetura e design de um sistema que é hospedado e operado em uma plataforma de computação em nuvem ou o termo mais usado “*Cloud*”.

Para esse contexto de software foi utilizado os serviços da *Oracle*®, que fornece uma máquina dentro de certos limites de “*hardware*” de forma gratuita por tempo ilimitado.

As maiores vantagens da computação em nuvem são:

Mais agilidade nas mudanças tecnológicas: Se uma aplicação tem seu servidor instalado na nuvem, é mais rápido para realizar uma mudança, seja geográfica ou de atualização tecnológica dos servidores e com custos baixos.

Escalabilidade no uso de serviços: Uma das maiores vantagens ao se usar a nuvem é a escalabilidade. Graças a essa característica, “*upgrades*” e “*downgrades*” ocorrem de forma facilitada, permitindo a manutenção dos servidores sempre na sua capacidade ideal.

Disponibilidade em todos os momentos, especialmente nos críticos: A chance de indisponibilidade nos serviços de TI por falhas na infraestrutura é muito baixa ao usar soluções de “*cloud computing*”. São aplicadas diversas proteções e boas práticas para prevenir esse tipo de situação, como backups, estruturas elétricas e de rede redundantes e distribuição geográfica de servidores, que minimizam a possibilidade de interrupção do serviço.

Maior segurança das informações: com a “*cloud computing*”, a segurança é aumentada pelo uso de *firewalls*, grupos e zonas de segurança organizadas em camadas. A vigilância pode ser realizada por profissionais especializados que se encarregam de combater eventuais tentativas de ataques.

Gerenciamento do uso de recursos: o controle de informações sobre os serviços em nuvem é amplo. Diversos indicadores são disponibilizados, sendo possível mensurar dados em grande escala, ajudando nas tomadas de decisão da melhor forma para alocar ou reduzir certos recursos para determinada aplicação. Além disso, o gerenciamento de uso de recursos permite um maior controle dos gastos financeiros por parte da empresa.

Com todas essas possibilidades de agregar valor ao negócio, a computação em nuvem é uma opção de transformação que deve ser buscada por quem almeja se diferenciar e obter sucesso frente à concorrência no mercado e consequentemente entregar produtos robustos e completos.

6.3 Gerenciamento dos Dados do Usuário

Quando um usuário decide deletar sua conta da plataforma Conexão Acará, todos seus dados são deletados no mesmo instante.

Apagar todos os registros de dados de um usuário, incluindo o e-mail, é uma prática importante para garantir a segurança dos dados. Aqui estão algumas razões pelas quais isso é importante:

Minimizar o risco de vazamento de dados: Manter informações pessoais, como endereço de e-mail, mesmo após a exclusão da conta, pode representar um risco de vazamento de dados, caso ocorra uma violação de segurança ou um acesso não autorizado. Ao excluir completamente essas informações, se reduz a exposição e o risco associado ao armazenamento de dados pessoais.

Conformidade com regulamentações de proteção de dados: Em muitas regiões, existem leis e regulamentos específicos sobre privacidade e proteção de dados, como o Regulamento Geral de Proteção de Dados (GDPR) na União Europeia. Essas regulamentações exigem que as organizações excluam os dados pessoais dos usuários quando solicitado, a menos que exista uma base legal para mantê-los. Cumprir essas regulamentações é crucial para evitar penalidades legais e garantir a privacidade dos usuários. Isso é um requisito fundamental para garantir que qualquer empreendimento ou negócio que envolva dados do usuário obtenha sucesso.

Evitar uso indevido de informações: Mesmo que um usuário tenha optado por excluir sua conta, manter seu endereço de e-mail pode abrir caminho para o uso indevido das informações por parte de terceiros. Isso pode levar a “spam”, “phishing” ou outras formas de abuso. Excluir completamente o e-mail do usuário ajuda a evitar essas situações e protege sua privacidade.

Transparência e confiança: Ao permitir que os usuários tenham controle sobre seus dados e excluir completamente suas informações pessoais, demonstra-se transparência. Isso pode resultar em uma reputação positiva para sua aplicação web e incentivar outros usuários a confiar em sua abordagem de privacidade e segurança. Além de evitar possíveis entraves e penalidades legais.

Seguir as melhores práticas de segurança ao implementar qualquer funcionalidade relacionada ao gerenciamento de dados pessoais dos usuários, incluindo o cumprimento das regulamentações aplicáveis e a garantia da proteção adequada dos dados durante o processo de exclusão foi e será uma preocupação constante para o sistema tema deste trabalho.

7. CONSIDERAÇÕES FINAIS

Segundo o Código de Ética do Profissional de Informática da Sociedade Brasileira de Computação, entre os deveres de um profissional de informática, está o de contribuir para o bem-estar social, promovendo, sempre que possível, a inclusão de todos os setores da sociedade. (SBC. CÓDIGO DE ÉTICA DO PROFISSIONAL DE INFORMÁTICA,2019).

Portanto, o sistema construído neste trabalho foi desenvolvido para ser uma opção factível de inclusão dos produtores rurais da região de Acará/PA na economia digital. Para isso acontecer pesquisaram-se as práticas mais recomendadas.

Como um dos objetivos dos profissionais de tecnologia, que trabalham no desenvolvimento de sistemas, é o de elaborar soluções que proporcionem agilidade e segurança na disponibilização de informações, bem como auxiliar seus usuários nas tomadas de decisões. Acreditamos que o sistema proposto, Conexão Acará, está apto para solucionar os entraves mercadológicos, pois oferece uma solução mais eficiente e acessível para os problemas específicos que afetam essa comunidade.

Diante disso, o software em questão pode levar a uma melhoria contínua do produto, com base nas opiniões e *feedback* dos usuários. Isso leva à, resultados de soluções mais eficazes e customizadas para as necessidades da comunidade, criando oportunidades de negócios.

7.1 TRABALHOS FUTUROS

Como trabalhos futuros cabe expandir a aplicação para construção de um aplicativo móvel (App), feitos para dispositivos (*smartphone* ou *tablet*) com sistema operacional “*Android* ®”, isso permitirá que o usuário sincronize as informações remotas com um banco de dado local (no dispositivo), utilizando o *SQLite*, que é um gerenciador de banco de dados nativo em dispositivos “*Android* ®”, para torna ainda melhor a experiência do usuário.

Com a implantação de um aplicativo a necessidade de acessos frequentes ao *Web Services* diminui, uma vez que os registros estejam disponíveis em base de dados local, que é o próprio *smartphone* do usuário. Assim ele não tem a necessidade de estar conectado à internet o tempo todo para ter acesso às informações

cadastradas. Tornando-se o *software* mais útil no contexto em questão, que é de baixa disponibilidade de conexão com a *internet* ou mesmo com baixa velocidade.

O *software* também incluirá recursos como um *gateway* de pagamento seguro para que os compradores possam realizar transações diretamente com os vendedores.

Cabe aumentar as funcionalidades do sistema a fim de suprir outros desafios da região. Um exemplo disto seria um espaço onde as pessoas pudessem visualizar os eventos culturais do município, abrindo assim espaço para o crescimento, difusão e valorização da cultura local.

Pretende-se também, aplicar técnicas de análise de dados para solucionar possíveis entraves que possam surgir entre os vendedores e compradores, além de fidelizar os compradores com ponto de fidelidades.

Construir um blog para ajudar os usuários do sistema a dominarem a plataforma e também ajudar na direção de seus negócios e de suas finanças, com artigos periódicos sobre negociações, planejamento financeiro, e técnicas de manejo e plantio.

7.1.1 Segurança da Informação

A importância da segurança da informação nas redes. Principalmente a necessidade de proteger os dados e sistemas contra ameaças, como *hackers*, malware, ataques de negação de serviço (DoS) e espionagem. Conceitos de criptografia, autenticação, controle de acesso e políticas de segurança, e técnicas de detecção e prevenção de intrusões devem ser levados em consideração. A segurança deve ser uma preocupação central na concepção, implementação e manutenção de redes, enfatizando a importância de práticas adequadas de segurança para garantir a integridade, confidencialidade e disponibilidade dos dados transmitidos. (TANENBAUM, 2012).

Nesse sentido, o sistema proposto pretende continuar aprimorando medidas para assegurar o sigilo e segurança dos dados do usuário.

Cookies são pequenos arquivos de texto que são armazenados no navegador do usuário e são usados para diversas finalidades, como autenticação, gerenciamento de sessões, personalização de conteúdo e rastreamento de atividades.

Armazenar *cookies* dos usuários em uma aplicação *web* pode desempenhar um papel importante na segurança dos usuários.

A seguir serão listados alguns motivos pelos quais os *cookies* podem desempenhar um papel importante na segurança:

Autenticação: *Cookies* são frequentemente usados para autenticar usuários, armazenando informações de sessão ou *tokens* de acesso. Essa abordagem permite que os usuários acessem áreas restritas do aplicativo ou executem ações específicas, evitando a necessidade de inserir suas credenciais repetidamente. Isso pode melhorar a experiência do usuário, desde que a implementação seja segura.

Prevenção de ataques de CSRF: *Cross-Site Request Forgery* (CSRF) é um tipo de ataque em que um invasor explora a confiança entre um usuário e um aplicativo para realizar ações indesejadas em nome do usuário. O uso de *cookies* para armazenar *tokens CSRF* e vinculá-los a formulários ou solicitações pode ajudar a prevenir esse tipo de ataque.

Personalização de segurança: Ao armazenar preferências de segurança do usuário em um cookie, como configurações de privacidade ou níveis de segurança, é possível adaptar a experiência do usuário e garantir que suas preferências sejam respeitadas.

Gerenciamento de sessões: *Cookies* são frequentemente usados para rastrear e gerenciar sessões de usuários em aplicativos *web*. Ao fazer isso de maneira segura, é possível controlar o tempo de expiração das sessões e fornece uma camada adicional de proteção contra ataques de sessão.

É importante destacar que a segurança não depende apenas do uso ou não de *cookies*, mas de uma implementação adequada das medidas de segurança em todos os aspectos da aplicação.

8. REFERÊNCIAS

As Maiores Vantagens em Computação em Nuvem. Disponível em: <<https://blog.centralserve.com.br/as-maiores-vantagens-da-computacao-em-nuvem-para-empresas/>>. Acesso em 28 de janeiro de 2023.

Arquitetura de Três Camadas (*tiers*). Disponível em: <<https://www.ibm.com/br-pt/cloud/learn/three-tier-architecture>>. Acesso em 29 de dezembro de 2022.

DALL’OGLIO, P, **PHP Programando com Orientação a Objetos.** Segunda Edição, Novatec, Nova Edição 2009.

EQUIPE TOTVS, **Arquitetura REST.** Saiba o que é e seus diferenciais, 2020. Disponível em: <<https://www.totvs.com/blog/developers/rest>>. Acesso em: 11 de outubro de 2022.

ECMA-262, **ECMAScript® 2022 language specification.** Disponível em: <<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>>. Acesso em 20 de dezembro de 2022.

FERRARI, Fabrício Augusto. **Crie banco de dados em MySQL.** São Paulo, Digerati Books, 2007.

FIELDING, Roy Thomas et al. **Hypertext Transfer Protocol - HTTP/1.1. 1997. RFC 2068.** Disponível em: <<http://www.ietf.org/rfc/rfc2068.txt>>. Acesso em 24 de dezembro de 2023.

FIELDING, Roy Thomas et al. **No-rest-in-cmis.** 2008. Disponível em: <<https://roy.gbiv.com/untangled/2008/no-rest-in-cmis>>. Acesso em: 17 jan. 2023.

GIT. Disponível em: <<https://git-scm.com/>>. Acesso em 30 de dezembro de 2022.

GITLAB. ABOUT. Disponível em: <<https://about.gitlab.com/company/>>. Acesso em 28/01/2023.

IBM. **O que é um web services?** Disponível em: <<https://www.ibm.com/docs/en/cics-ts/5.2?topic=services-what-is-web-service>>. Acesso em: 30 de agosto de 2022.

JSON. Disponível em: <<https://json.org/json-pt.html>>. Acesso em 10 outubro de 2022.

Lei Geral de Proteção de Dados Pessoais. Disponível em: <<https://www.gov.br/esporte/pt-br/acao-a-informacao/lgpd>>. Acesso em: 17 de maio de 2023.

MICROSOFT. **BUILD 2015 NEWS.** Disponível em: <<https://devblogs.microsoft.com/visualstudio/build-2015-news-visual-studio-code-visual-studio-2015-rc-team-foundation-server-2015-rc-visual-studio-2013-update-5/>>. Acesso em 18 de fevereiro de 2022.

MACORATTI, José Carlos. **Entendendo o Padrão MVC - Model - View - Controller.** Disponível em: <https://macoratti.net/14/05/net_mvc.htm>. Acesso em 30 de dezembro de 2022.

MACORATTI, José Carlos. **Padrões de Projeto: O modelo MVC - Model View Controller.** Disponível em: <https://www.macoratti.net/vbn_mvc.htm>. Acesso em: 11 de outubro de 2022.

WIKIPEDIA. **Visual_Studio_Code.** Disponível em: <https://pt.wikipedia.org/wiki/Visual_Studio_Code>. Acesso em 09 de dezembro de 2022.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Código de Ética para o Profissional de Informática.** Disponível em: <<https://www.sbc.org.br/institucional-3/codigo-de-etica>>. Acesso em 18 de fevereiro de 2022.

Sobre o município de Acará. Disponível em: <<https://acara.pa.gov.br/o-municipio/sobre-o-municipio>>. Acesso em: 15 de outubro de 2022.

PA: Acará Fest destaca produção de mandioca do Estado. Disponível em: <<https://www.paginarural.com.br/noticia/168952/acara-fest-destaca-producao-de-mandioca-do-estado>>. Acesso em 10 de outubro de 2022.

Segurança da informação Disponível em: <<https://www.certifiquei.com.br/confidencialidade-integridade-disponibilidade>>. Acesso em 27 de janeiro de 2023.

W3C Brasil. Disponível em: <<https://www.w3c.br/>>. Acesso em: 17 de maio de 2023.

REVERBEL, Francisco. **O que são *web services***. USP, São Paulo, mai. 2006.
Disponível em: <[http://www.ime.usp.br/~reverbel/SOD-06/trabalhos/fachada-
ws/node2.html](http://www.ime.usp.br/~reverbel/SOD-06/trabalhos/fachada-
ws/node2.html)>. Acesso em: 17 de maio de 2022.

SILVA, Maurício Samy. **HTML5 - A Linguagem de Marcação que Revolucionou a Web** - 2ª Edição. São Paulo: Novatec, 2014.

SILVA, Maurício Samy. **Javascript. Guia do Programador: Guia Completo das Funcionalidades de Linguagem JavaScript**. 1. ed. São Paulo: Novatec, 2010.

TANENBAUM, A. S. **Redes de Computadores**. Sexta Edição, Bookman, 2021.

ZIVIANI, Nivio. **Projetos de Algoritmos: com implementação em *java* e *c++***, Thomson Learning, 2007.

Sommerville, Ian. **Engenharia de Software** / Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011. Título original: Software engineering. ISBN 978-85-7936-108-1 1.