
Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

GABRIEL MARTINS DE OLIVEIRA
HUGO TAVARES RODRIGUES

**DESENVOLVIMENTO DO PROTÓTIPO FUNCIONAL DE UMA
APLICAÇÃO WEB DE CONTROLE DE ESTOQUE DOMÉSTICO NA
ÁREA DE ALIMENTOS**

Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

GABRIEL MARTINS DE OLIVEIRA
HUGO TAVARES RODRIGUES

**DESENVOLVIMENTO DO PROTÓTIPO FUNCIONAL DE UMA
APLICAÇÃO WEB DE CONTROLE DE ESTOQUE *DOMÉSTICO* NA
ÁREA DE ALIMENTOS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Antônio Alfredo Lacerda
Área de concentração: Análise e Desenvolvimento de Sistemas

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

O47d OLIVEIRA, Gabriel Martins de

Desenvolvimento do protótipo funcional de uma aplicação Web de controle de estoque doméstico na área de alimentos. / Gabriel Martins de Oliveira, Hugo Tavares Rodrigues. – Americana, 2021.
33f.

Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientadores: Prof. Antonio Alfredo Lacerda

1 Desenvolvimento de software 2. Dispositivos móveis – aplicativo 3. Domótica
I. RODRIGUES, Hugo Tavares II. LACERDA, Antonio Alfredo III. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.3.05

GABRIEL MARTINS DE OLIVEIRA

HUGO TAVARES RODRIGUES

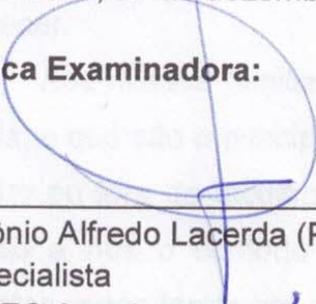
**DESENVOLVIMENTO DO PROTÓTIPO FUNCIONAL DE UMA
APLICAÇÃO WEB DE CONTROLE DE ESTOQUE DOMÉSTICO NA
ÁREA DE ALIMENTOS**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

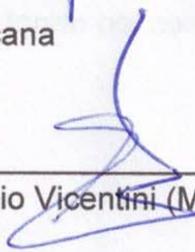
Área de concentração: Programação.

Americana, 07 de dezembro de 2021.

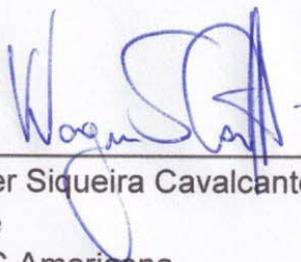
Banca Examinadora:



Antônio Alfredo Lacerda (Presidente)
Especialista
FATEC Americana



Eduardo Antonio Vicentini (Membro)
Mestre
FATEC Americana



Wagner Siqueira Cavalcante (Membro)
Mestre
FATEC Americana

DEDICATÓRIA

“A força não provém da capacidade física. Provém de uma vontade indomável.”

Mahatma Gandhi

RESUMO

Este Projeto trata do desenvolvimento do Protótipo funcional de uma aplicação WEB destinada ao controle de estoque de alimentos no ambiente doméstico. O levantamento dos requisitos foi realizado a partir da própria observação do ambiente doméstico dos autores deste documento, além de entrevistas com aqueles responsáveis por esta área em suas respectivas residências e/ou ambientes de trabalho caso aplicável. O Protótipo tem o objetivo de apresentar um sistema que poderá servir como registro móvel do estoque doméstico, como ferramenta rápida de pesquisa pelas quantidades necessárias para fins pré-definidos pelo usuário, e como uma lista de compras determinada pelo usuário, que também pode ser automatizada.

Palavras Chave: Estoque; Lista; Receitas; Cadastro

ABSTRACT

This Project is about the development of a WEB application functional prototype destined for food storage control in the domestic environment. The requirements were raised from the observation of the domestic environment of the authors of this subject, as well as interviews with those responsible for this field in their respective residences and/or work environments when applicable. The objective of the prototype is to present a system that will be able to serve as a mobile domestic storage register, as well as a quick search tool for the needed quantities for purposes pre-determined by the user, and a shop list that can be created by the user, or automatically generated.

Keywords: Storage; List; Recipes; Register

LISTA DE FIGURAS

Figura 1 - Diagrama de Caso de Uso.....	18
Figura 2 - Autenticação	19
Figura 3 - Consulta Estoque.....	20
Figura 4 - Abastecer e Consumir o Estoque.....	20
Figura 5 - Adicionar Lista.....	21
Figura 6 - Gerar Lista	22
Figura 7 - Adicionar Receita	24
Figura 8 - Editar Receita.....	24
Figura 9 - Deletar Receita	25
Figura 10 - Usar Receita	26
Figura 11 - Visualizar Receita / Consultar Estoque	26
Figura 12 - Reabastecer Estoque.....	27
Figura 13 - Diagrama de Classes.....	28
Figura 14 - Estrutura do banco de dados	29
Figura 15 - Ferramenta ArgoUML em execução	34
Figura 16 - Exemplo de linguagem em CSS	35
Figura 17 - Exemplo de programação em HTML	36
Figura 18 - Exemplo de JavaScript que exibe um pop-up escrito “ReLista, 2021”	37
Figura 19 - Exemplo de seleção de registros em SQL	38
Figura 20 - Imagem da ferramenta Notepad++	39
Figura 21 - Imagem da ferramenta Paint.NET.....	40
Figura 22 - Exemplo de PHP que puxa a data do dia do servidor.....	41
Figura 23 - Ferramenta phpMyAdmin em execução	42
Figura 24 - Imagem do XAMPP em execução	43
Figura 25 - Tela de Estoque	45
Figura 26 - Código PHP de adicionar item do estoque	46
Figura 27 - Função de adicionar no banco de dados	46
Figura 28 - Tela da Lista de Compras	47
Figura 29 - Código HTML de adicionar item da lista de compras.....	48
Figura 30 - Código JS de adicionar uma linha na tabela.....	48
Figura 31 - Código PHP para adicionar uma lista de compras.....	49
Figura 32 - Tela de Login do Protótipo	49
Figura 33 - Tela de Cadastro de Conta do Protótipo.....	50
Figura 34 - Tela de criação de Receitas.....	51
Figura 35 - Tela da Lista de Receitas.....	51
Figura 36 - Tela Sobre	52

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais do Projeto	16
Tabela 2 - Requisitos não funcionais do Projeto	17
Tabela 3 - Tabela de dados do <i>login</i>	30
Tabela 4 - Tabela de dados do estoque	30
Tabela 5 - Tabela de dados da lista	31
Tabela 6 - Tabela de dados dos itens da lista	31
Tabela 7 - Tabela de dados da receita	31
Tabela 8 - Tabela de dados do passo da receita	32
Tabela 9 - Tabela de dados do ingrediente da receita	32
Tabela 10 - Comparativo de funcionalidades de outros aplicativos.....	44

LISTA DE ABREVIATURAS E SIGLAS

ASP.NET – *Active Server Page Network*

CERN – *European Organization for Nuclear Research*

CGI – *Common Gateway Interface*

CSS – *Cascading Style Sheets*

HTML – *HyperText Markup Language*

PHP – *Personal Home Page*

SGBD – *Sistema Gerenciador de Banco de Dados*

SQL – *Structured Query Language*

WWW – *World Wide Web*

XAMPP – *Cross-Platform, Apache HTTP Server, MySQL, PHP and Perl*

XML – *Extensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	13
2	ENGENHARIA DE SOFTWARE	14
2.1	LEVANTAMENTO DE REQUISITOS	16
2.1.1	REQUISITOS FUNCIONAIS	16
2.1.2	REQUISITOS NÃO FUNCIONAIS	16
2.2	DIAGRAMA DE CASO DE USO	17
2.3	DIAGRAMA DE ATIVIDADE	19
2.4	DIAGRAMA DE CLASSES	28
2.5	BANCO DE DADOS	29
2.6	DICIONÁRIO DE DADOS	30
3	TECNOLOGIAS A SEREM EMPREGADAS	34
3.1	<i>ArgoUML</i>	34
3.2	CSS	35
3.3	HTML	36
3.4	JavaScript	36
3.5	MySQL	37
3.6	<i>Notepad++</i>	38
3.7	<i>Paint.NET</i>	39
3.8	PHP	40
3.9	<i>PHPMyAdmin</i>	41
3.10	XAMPP	42
4	PROJETO	44
4.1	PROBLEMÁTICA	44
4.2	PROTÓTIPO	44
4.3	ARQUITETURA E TECNOLOGIA	45
4.4	TELAS	49
5	CONSIDERAÇÕES FINAIS	53
	REFERÊNCIAS	54

1 INTRODUÇÃO

Este Projeto tem como objetivo o desenvolvimento do Protótipo de uma aplicação de gestão de estoque que oferece uma ampla visão do seu estoque, permitindo que o usuário anexe receitas, e gerando automaticamente uma lista de compras daquilo que o próprio usuário determina como necessário, reduzindo assim o desperdício.

Não é incomum acontecer de um cliente comprar um item no mercado por achar que o mesmo está em falta em sua residência, porém, ao chegar em casa, percebe que já tinha tal item. Situações como essa podem se repetir múltiplas vezes em uma única visita ao mercado, e não só gera desperdício, mas caso o item comprado seja de curta validade, pode acarretar o despejo e desperdício dele.

Claro que parte das escolhas do indivíduo são feitas a partir das necessidades e pela comparação de preços, mas muitas delas também são feitas inconscientemente, ou seja, os estímulos despertam o interesse por determinado produto, mas também pode haver uma questão de erro humano envolvida, e portanto, seria útil uma ferramenta que ajude a mitigar a ocorrência de tais erros humanos.

O Projeto descrito neste documento busca ajudar as famílias e até mesmo empresas a ter o consumo consciente e ter acesso móvel e rápido ao estoque virtual, facilitando assim o dia a dia, e otimizando a ida ao supermercado.

O Projeto foi estruturado em 5 Capítulos, sendo que o Capítulo 1, Conceitua a Introdução do Projeto, o Capítulo 2, Engenharia de Software, o Capítulo 3, Tecnologia a ser Empregada, Ferramentas e Linguagens de Programação, o Capítulo 4, Objeto de Estudo – Projeto, Capítulo 5, Considerações Finais.

Atualmente algumas pessoas não têm o controle do que de fato tem em sua residência, e não são raras as ocasiões em que ou são adquiridos itens em excesso, ou não são comprados em quantidade suficiente, gerando desperdício não só de dinheiro, mas também de alimentos perecíveis.

A pandemia do novo coronavírus resultou em uma corrida às compras, com pessoas estocando alimentos tendo como base, não a necessidade imediata, mas a pré-formulação de cenários futuros baseados no medo e na incerteza, e muitas vezes sendo compras desnecessárias, onde diariamente quantidades absurdas de comida foram parar

no lixo por uma mistura de erro humano e ansiedade. Como dito anteriormente, embora as questões emocionais sejam algo delicado, e que um simples sistema não possa tratar, é possível que grande parte do desperdício que foi gerado neste tumulto pudesse ter sido evitado, caso as pessoas pudessem ter acesso de forma mais simples ao que elas realmente têm em suas casas, inicialmente trazendo a informação, mas também, de maneira indireta, um pouco de tranquilidade, já que a informação ajuda a afastar a incerteza.

Com o avanço da tecnologia, não se encontram aplicativos que tem o controle da quantidade de estoque residencial, e a ideia da existência de tal sistema não é difundida no ambiente popular, desincentivando assim o surgimento de tal tecnologia. Com a criação desse Protótipo, a existência de um sistema capaz de realizar esta tarefa pode ser algo mais concreto e fácil de ser realizado.

Hoje em dia algumas pessoas possuem o costume de gerar lista de compras quando está para preparar uma receita, e quando está realizando as compras, podem esquecer de comprar um produto, ou acaba comprando outro no impulso. Com isso, surgiu a ideia do desenvolvimento de um sistema que contenha o controle de estoque, e, baseado nele, será gerada a lista de compras e as consultas de receitas que foram criadas dentro do sistema

O objetivo principal é desenvolver o Protótipo de uma aplicação web, apresentando os planos para o sistema na parte do desenvolvimento, facilitando assim a vida dos consumidores nas compras e, assim, tendo um consumo consciente.

Diante das ideias apontadas no *brainstorm*, o objeto é desenvolver o Protótipo de um software na web para que os usuários domésticos tenham o controle de seu estoque de suas residências.

O Protótipo visa ter inserção, modificação e remoção de itens da residência, criará lista de compras que será gerada automaticamente baseado no que foi configurado cada item do seu estoque, e por fim, podem ser adicionadas receitas que consultam a disponibilidade de cada ingrediente cadastrado no estoque.

2 ENGENHARIA DE SOFTWARE

Antes de colocar um Projeto em prática, deve-se antes estabelecer o escopo real do que se espera alcançar no processo, com objetivos realistas e uma organização clara, e é esta a função da engenharia de software, pois, como SOMMERVILLE mencionou:

“Os sistemas de software são abstratos e intangíveis. Eles não são restringidos pelas propriedades dos materiais, nem governados pelas leis da física ou pelos processos de manufatura [...] devido a essa falta de restrições físicas, os sistemas de software podem se tornar extremamente complexos de modo muito rápido, difíceis de entender e caros para aletrar.” (SOMMERVILLE, 2019, p.02)

A função da Engenharia de Software é determinar os objetivos do Projeto, e delinear o caminho que será tomado para alcançá-lo, e a Engenharia de Software dispõe de várias ferramentas que tornam isto possível. Dentre tais ferramentas estão os diagramas e o levantamento de requisitos.

O levantamento de requisitos é a parte mais básica do processo, pois é ele que determina O QUE deve ser feito pelo sistema, ou seja, qual o papel que ele deve cumprir, de várias maneiras diferentes, sendo divididos entre requisitos Funcionais e Não Funcionais.

Já os diagramas são usados para determinar COMO o sistema deve cumprir seus objetivos, e diagramas diferentes demonstram aspectos diferentes de um Projeto. Entre os diagramas que podem ser criados estão: Diagrama de Caso de Uso, de Classes, de Sequência, de Atividades, de Objetos, e de Componentes. Neste Protótipo foram modelados os diagramas de Caso de Uso, Diagrama de Classes, e Diagrama de Atividades.

O Projeto foi decidido conforme a realização do *brainstorm* nas dependências da FATEC Americana, e, com base na realização das telas, foram desenhados à mão os Protótipos de como seria a aparência do software, e considerando que se trata de um Projeto em web, chegou-se à conclusão de que as ferramentas mais adequadas seriam: Notepad++ para auxiliar na produção da linguagem HTML, CSS, PHP e *JavaScript*. O Banco de Dados que foi escolhido para o Projeto foi o *phpMyAdmin* com ajuda do software XAMPP. Baseado nas tarefas que seriam realizadas ao longo do

desenvolvimento, foi feito o levantamento de requisitos com o intuito de determinar quais seriam as principais funções realizadas pelo software.

2.1 LEVANTAMENTO DE REQUISITOS

A Engenharia de Requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.1.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). Como descreve Sommerville, os requisitos funcionais têm a função de tabelar as funcionalidades do sistema que será projetado. Os requisitos funcionais do Protótipo descrito neste documento se encontram na Tabela 1.

Tabela 1 – Requisitos funcionais do Projeto

Identificação	Requisitos Funcionais	Propriedade
RF1	(Des)mascarar a senha de <i>login</i>	Desejável
RF2	Avisar quando o estoque tiver baixo	Essencial
RF3	Adicionar na Lista de Compras o estoque a faltar	Importante
RF4	Consultar a disponibilidade dos ingredientes da receita	Essencial
RF5	Aparecer no estoque os itens favoritos	Desejável

Fonte: Elaborado pelo Autor

2.1.2 REQUISITOS NÃO FUNCIONAIS

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2019). Requisitos não

funcionais são aqueles que não envolvem funcionalidades diretas do sistema, mas sim itens “passivos” do sistema, tais como questões legais, estéticas, relativas à performance, e outros itens que não requerem interação para que o sistema realize sua função, tais como os descritos na Tabela 2.

Tabela 2 - Requisitos não funcionais do Projeto

Identificação	Requisitos Não Funcionais	Categoria	Propriedade
RNF1	Cores da tela clara	Padrões	Opcional
RNF2	Letras da fonte visível ao olho	Padrões	Importante
RNF3	Página de web responsivo	Padrões	Essencial

Fonte: Elaborado pelo Autor

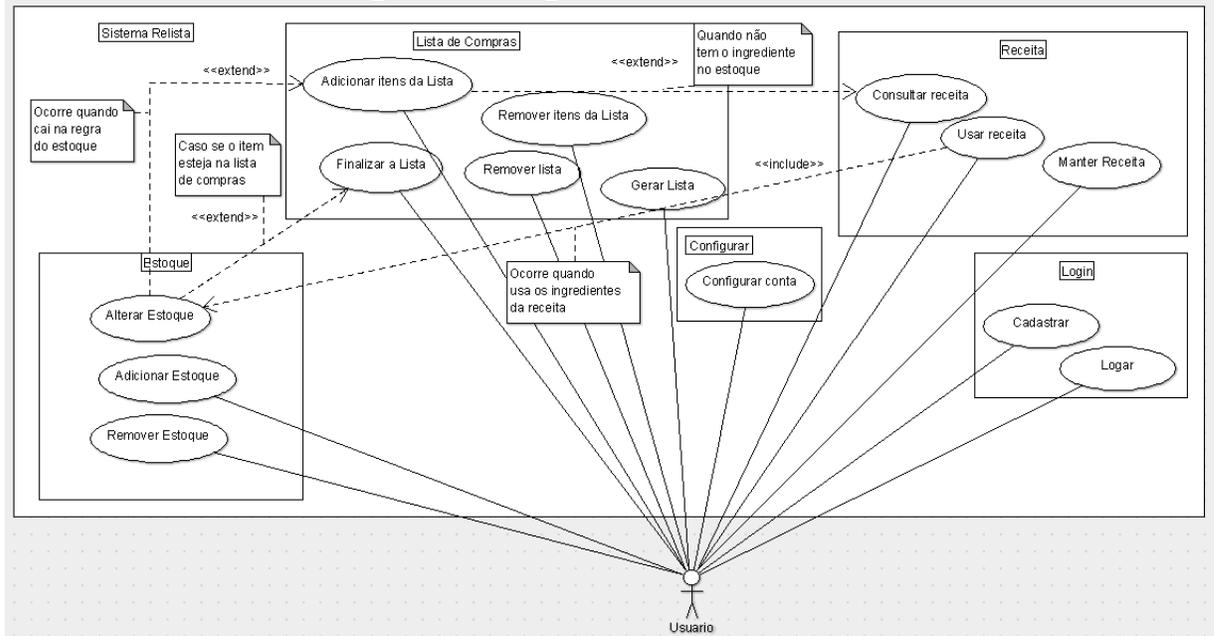
2.2 DIAGRAMA DE CASO DE USO

Este diagrama oferece uma visão mais simples do sistema do ponto de vista do usuário e outros atores envolvidos (Figura 1).

De acordo com GUEDES, o objetivo do diagrama de caso de uso é:

“[...] apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar muito com a questão de como tais funcionalidades serão implementadas. Costuma ser utilizado principalmente nas fases de elicitação e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para diversos outros diagramas.” (GUEDES, p.32-33, 2018).

Figura 1 - Diagrama de Caso de Uso



Fonte: Elaborado pelo Autor

O usuário poderá tanto criar uma conta, quanto efetuar *login* com uma já existente. Após o *login*, o usuário poderá acessar qualquer um dos menus disponíveis: Receita, Configurações, Lista de Compras, e Estoque.

Acessando o menu Receita, o usuário poderá criar uma receita, ou alterar uma já criada.

Acessando o menu Lista de Compras, o usuário poderá gerar uma nova lista, adicionar ou remover itens dela, ou também deletá-la.

Acessando o menu Estoque, o usuário poderá adicionar, remover, ou alterar as quantidades e/ou propriedades de um item no estoque.

É também possível, no menu Lista de compras, finalizar a lista, o que irá modificar as quantidades do estoque de acordo com os itens da lista.

No menu receita, existe a opção Usar Receita, que irá descontar do estoque os itens que foram usados na receita.

Ao criar a lista de compras, é possível relacionar uma ou mais receitas a ela, preenchendo automaticamente a lista com os itens necessários para completar a receita, com a quantidade sendo deduzida a partir da diferença entre a receita e o estoque.

No menu Configurar, é possível alterar as configurações da conta.

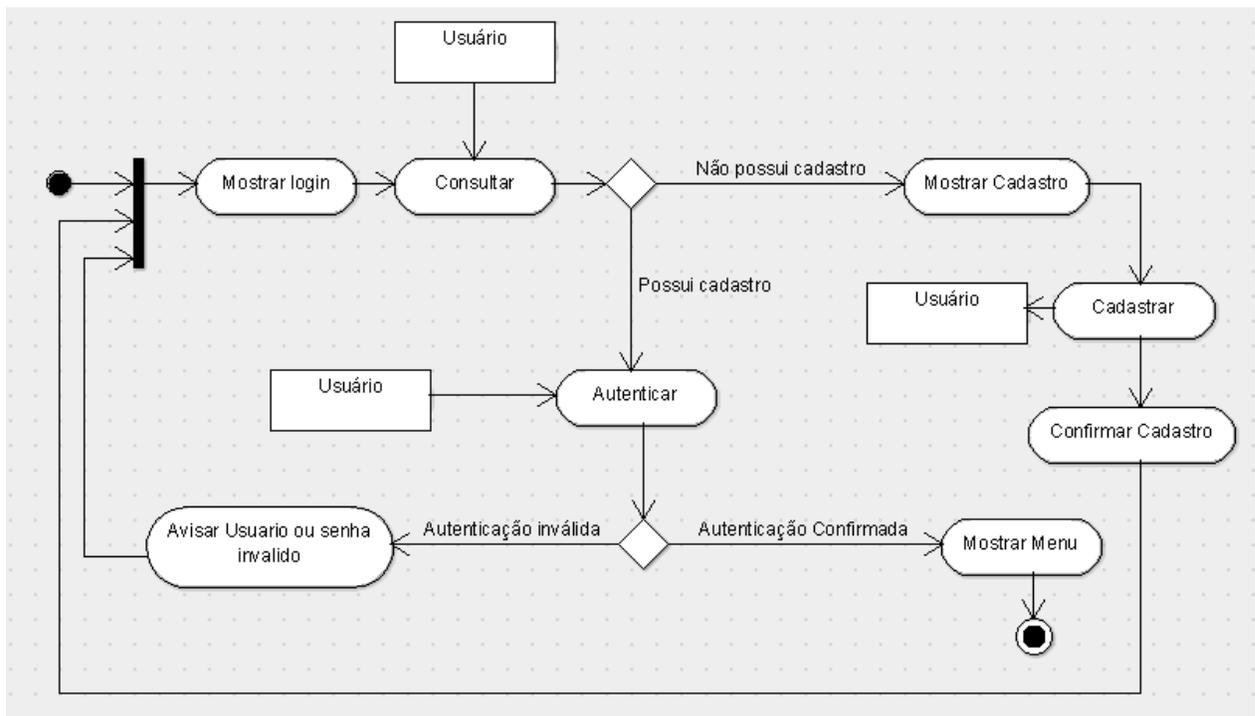
2.3 DIAGRAMA DE ATIVIDADE

O Objetivo do Diagrama de Atividades é detalhar individualmente as diversas operações realizadas pelo sistema.

A respeito de uma atividade descrita neste tipo de diagrama, GUEDES atesta que ela pode:

“[...] ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle e de objetos de uma atividade.” (GUEDES, p.39-40, 2018).

Figura 2 – Autenticação

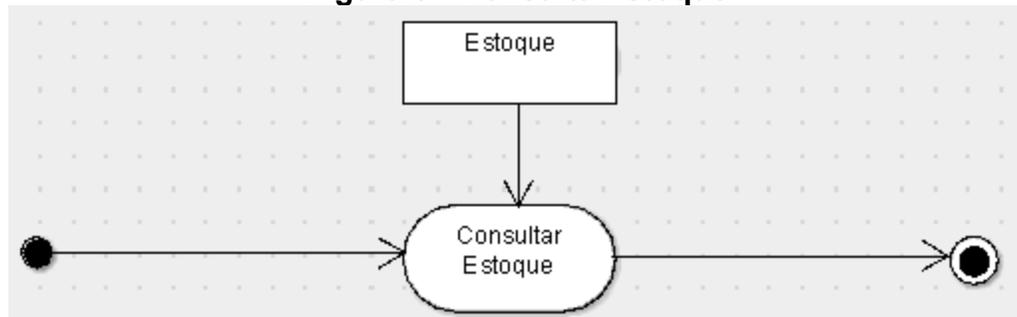


Fonte: Elaborado pelo Autor

Ao acessar a tela de *login*, caso o usuário não possua uma conta, ele irá cadastrá-la, e então confirmar o cadastro e acessar os menus.

Caso ele já possua um cadastro, ele deverá informar seus dados, que serão verificados. Caso sejam válidos, ele prosseguirá para os menus, caso contrário, retornará à tela de login como pode visualizar na (Figura 2).

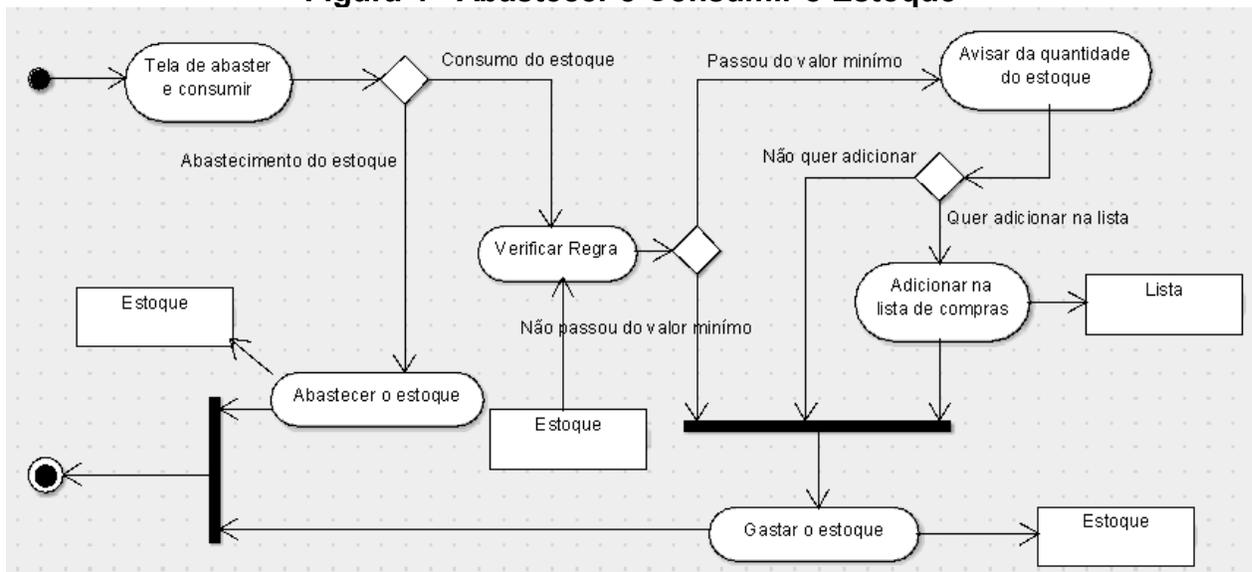
Figura 3 - Consulta Estoque



Fonte: Elaborado pelo Autor

Na (Figura 3) mostra-se que ao selecionar a opção de consultar o estoque, o sistema acessa o banco de dados do estoque, e usa os dados ali contidos como base para realizar as operações requisitadas que exijam as quantidades nele contidas.

Figura 4 - Abastecer e Consumir o Estoque

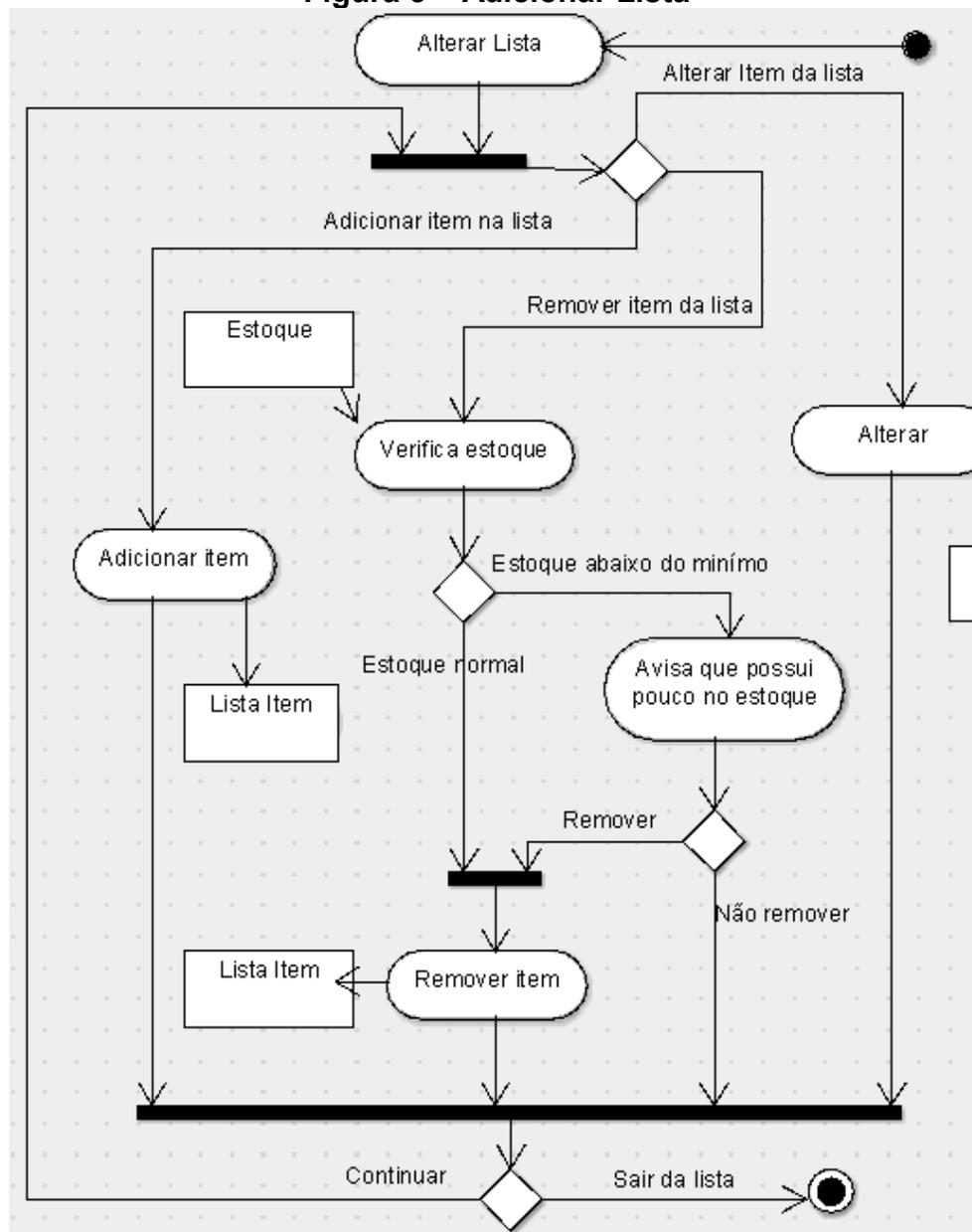


Fonte: Elaborado pelo Autor

Acessando o menu de estoque, a (Figura 4) mostra que se deve escolher entre Abastecer ou Consumir o estoque. Caso seja selecionado Abastecer, devem ser informados os itens a serem abastecidos.

Caso seja selecionado o Consumo, deve-se informar a quantidade a ser consumida. Em seguida, a quantidade informada é removida do estoque. Após o consumo, caso a quantidade em estoque do item em questão fique abaixo de algum limite pré-estabelecido pelo usuário, será exibido um aviso informando tal situação. Após o aviso, o usuário pode escolher entre adicionar este item à lista de compras.

Figura 5 – Adicionar Lista



Fonte: Elaborado pelo Autor

Ao selecionar o menu de listas como mostra na (Figura 5), pode-se tanto adicionar um novo item à lista, remover um item, ou alterar algum item já presente na lista.

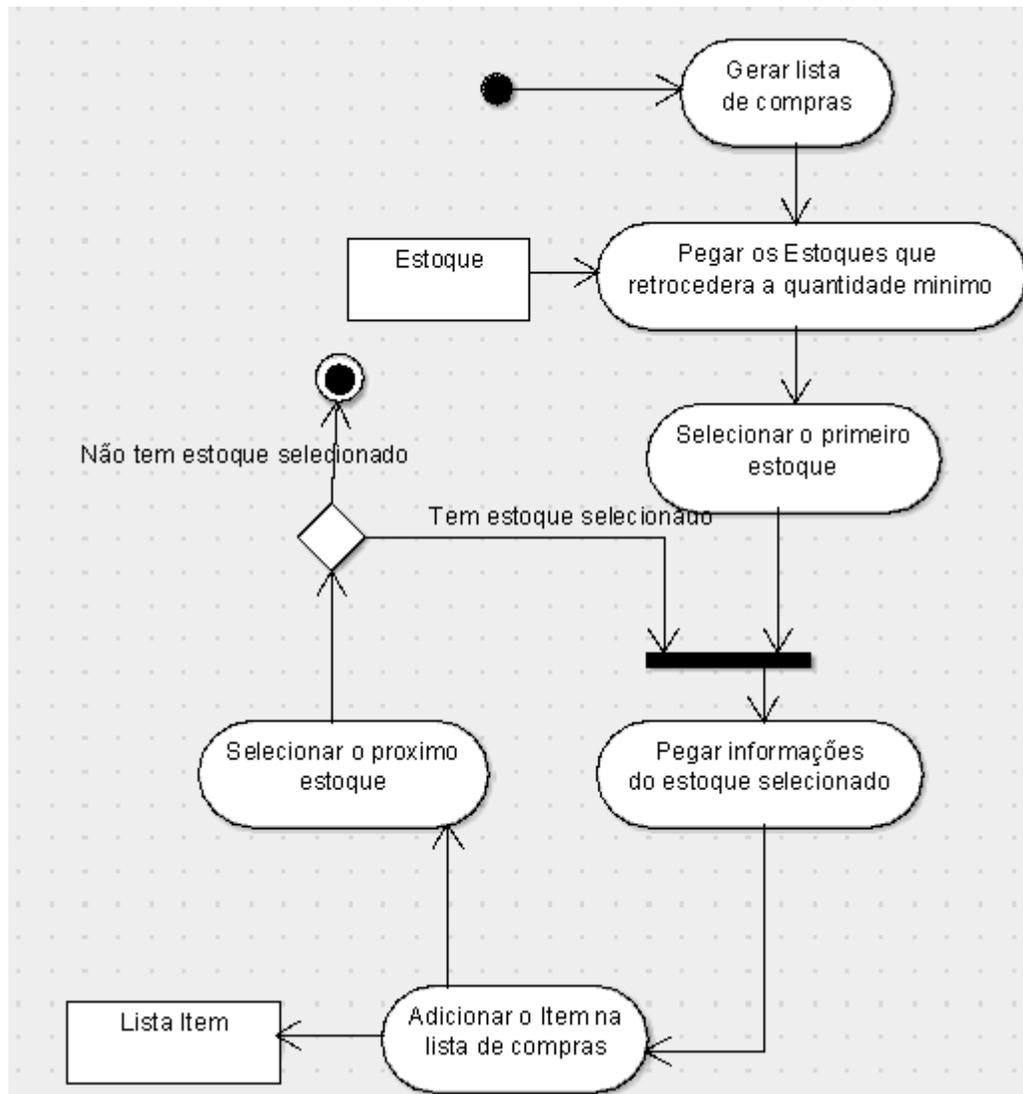
Caso decida adicionar, deve ser informada a quantidade a ser comprada.

Caso decida remover, e a quantidade em estoque não estiver sob nenhuma regra, o item será removido. Caso o item removido esteja abaixo da quantidade determinada pela regra do estoque, será exibido um aviso, e será pedida a confirmação da remoção do item da lista.

Caso decida Alterar um item, deve ser informada a nova quantidade a ser comprada.

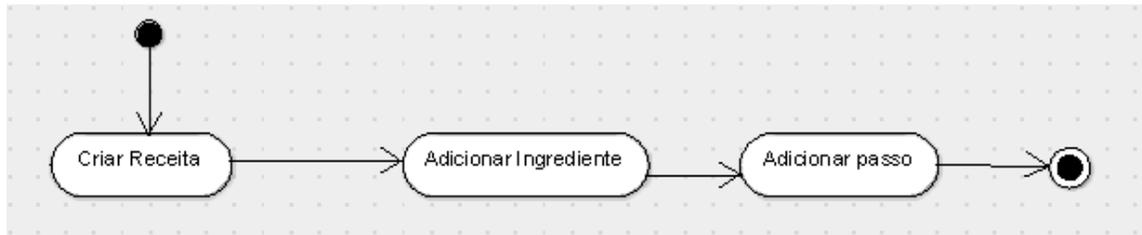
Após qualquer um dos caminhos anteriores, deve-se escolher entre prosseguir com a modificação da lista, retornando à seleção entre Adicionar, Remover ou Alterar, ou sair da lista, retornando ao menu principal.

Figura 6 - Gerar Lista



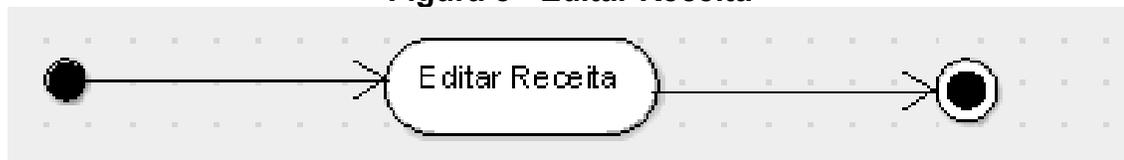
Fonte: Elaborado pelo Autor

Na (Figura 6), mostra que ao selecionar a geração de uma lista, o sistema verifica o estoque e, caso haja algum item, cuja quantidade em estoque for inferior à de alguma regra, ele será automaticamente adicionado à lista. Após o item ser adicionado, é verificado se há outro item em falta. Caso haja, ele será adicionado. Caso contrário, sairá da lista.

Figura 7 - Adicionar Receita

Fonte: Elaborado pelo Autor

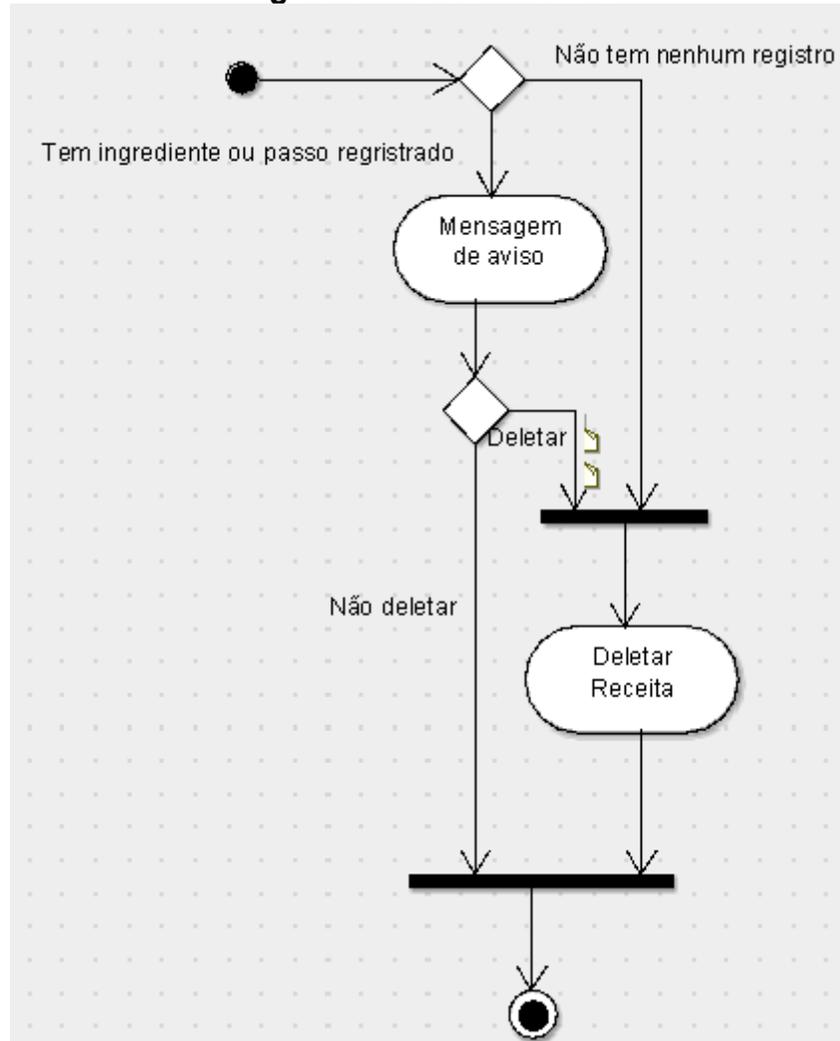
Ao selecionar Criar Receita, deverão ser inseridos os ingredientes necessários, seguidos então pelas etapas do preparo, antes de finalizar a criação da receita como mostra na (Figura 7).

Figura 8 - Editar Receita

Fonte: Elaborado pelo Autor

Na (Figura 8), ao selecionar editar a receita, será exibida a tela com as informações da receita, que poderão ser alteradas pelo usuário.

Figura 9 - Deletar Receita

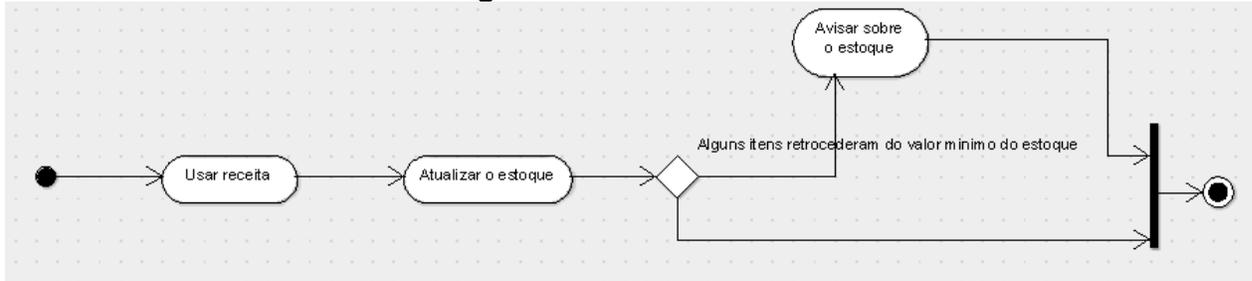


Fonte: Elaborado pelo Autor

Na (Figura 9), ela indica que ao selecionar a receita a ser deletada, caso seja uma receita vazia, sem registro, será deletada imediatamente.

Caso haja registros na receita, será pedida a confirmação da deleção. Caso confirme, a receita será deletada. Caso contrário, a receita será mantida.

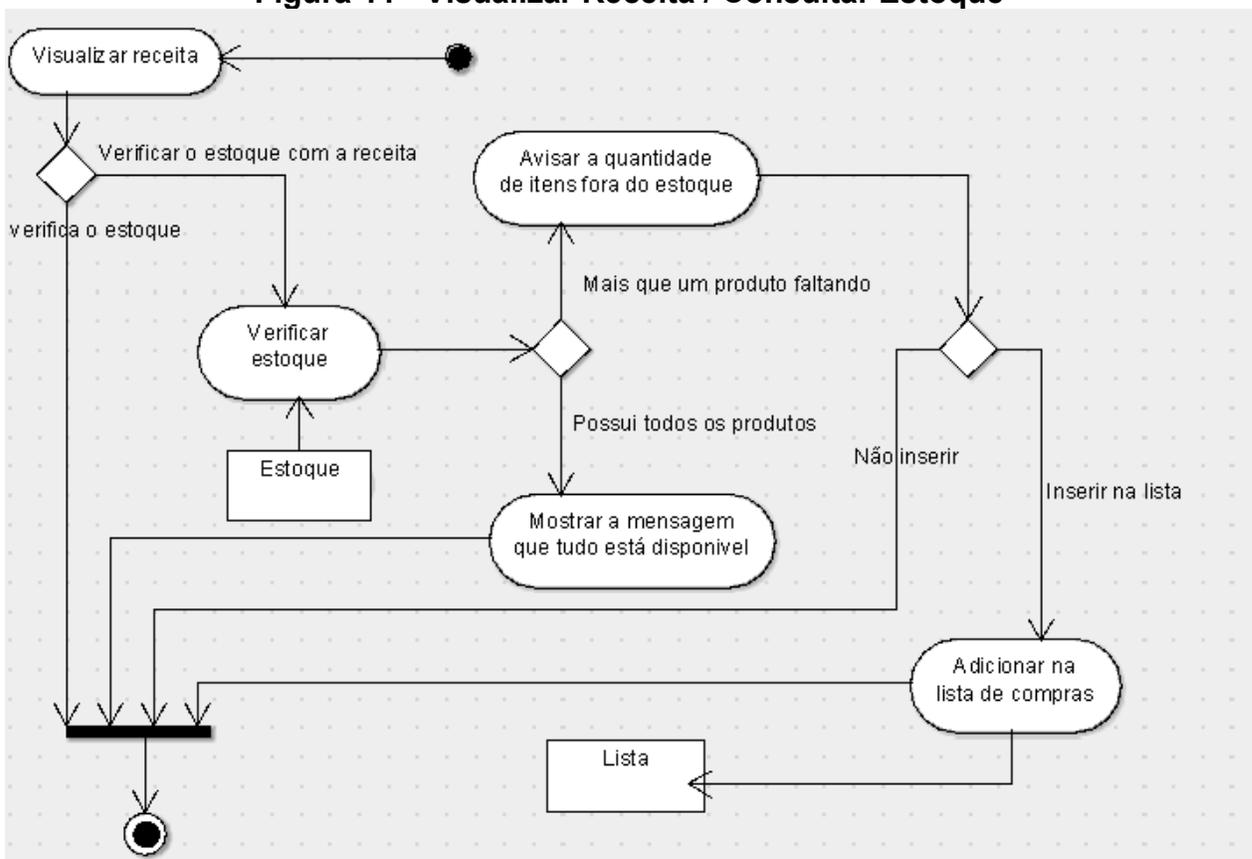
Figura 10 - Usar Receita



Fonte: Elaborado pelo Autor

Na (Figura 10), ao selecionar a receita e usá-la, as quantidades descritas nela serão reduzidas do estoque. Caso, após isto, algum item fique abaixo de algum valor mínimo, será exibido um aviso.

Figura 11 - Visualizar Receita / Consultar Estoque



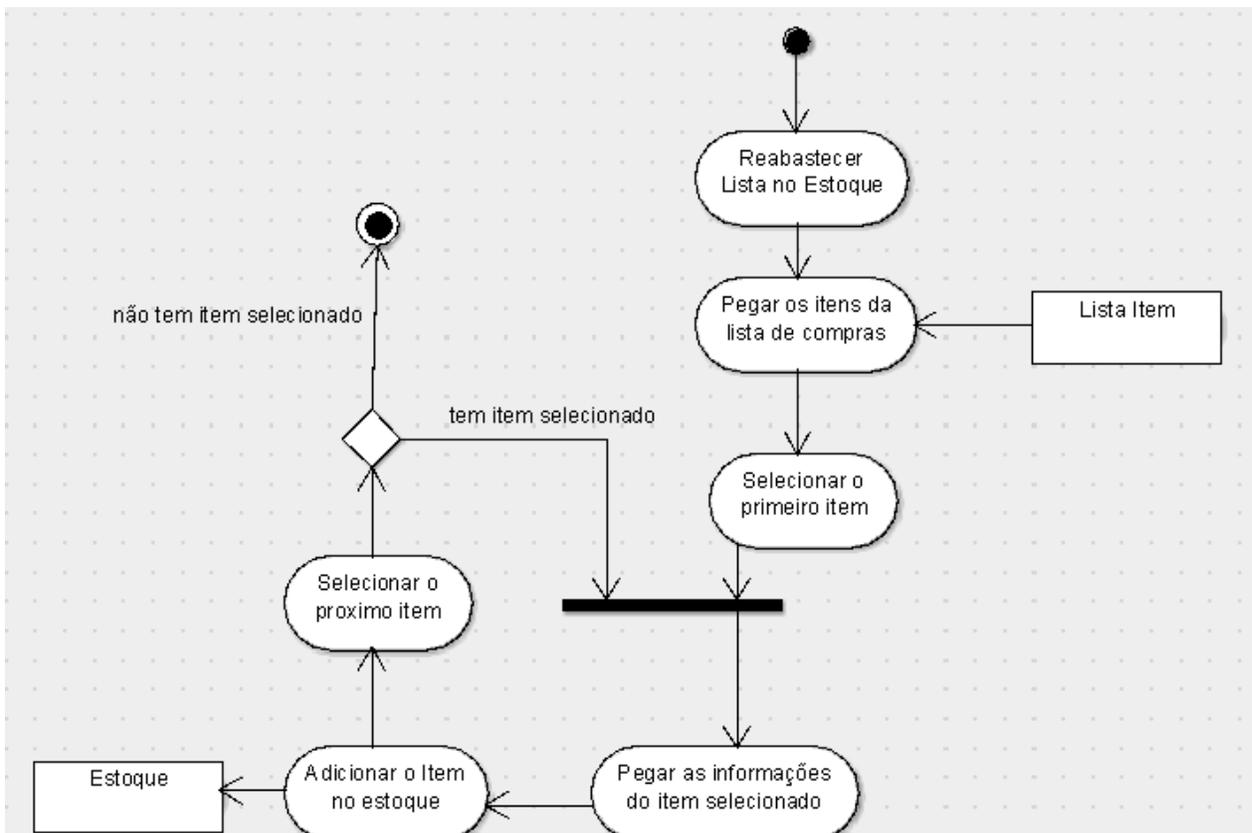
Fonte: Elaborado pelo Autor

Ao seleccionar a receita a ser consultada, pode-se apenas visualizar, ou realizar uma verificação automática. Caso realize a verificação, as quantidades descritas na receita serão comparadas com os itens em estoque.

Caso a quantidade em estoque seja inferior à descrita na receita, será exibido o aviso e a opção de adicionar os itens automaticamente a uma lista de compras. Se optar por adicionar, o item será adicionado à lista de compras.

Caso o estoque contenha a quantidade necessária para o preparo, será exibida uma mensagem informando que tudo está disponível como se mostra na (Figura 11).

Figura 12 - Reabastecer Estoque



Fonte: Elaborado pelo Autor

A (Figura 12) mostra que para reabastecer o estoque a partir de uma lista de compras, o sistema verifica os itens da lista, seleciona o primeiro, adiciona a quantidade descrita nele ao estoque, segue para o próximo item, o adiciona ao estoque, até que a lista termine.

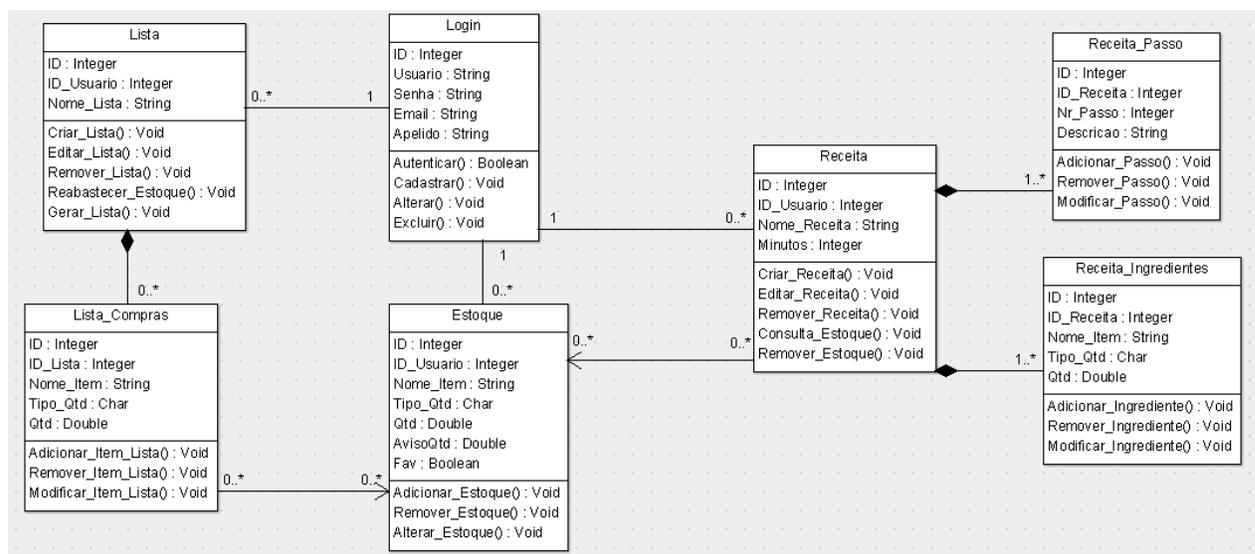
2.4 DIAGRAMA DE CLASSES

O Diagrama de Classes tem o objetivo de demonstrar as relações entre as classes e métodos de um Projeto, segue abaixo na (Figura 13) o diagrama de classe do protótipo.

De acordo com GUEDES, o Diagrama de Classes:

[...] Apresenta uma visão estática de como as classes estão organizadas, preocupando-se em como definir a estrutura lógica delas. O diagrama de classes serve ainda como apoio para a construção da maioria dos outros diagramas da linguagem UML. (GUEDES, p.34, 2018).

Figura 13 - Diagrama de Classes



Fonte: Elaborado pelo Autor

2.5 BANCO DE DADOS

O Banco de dados é essencial para que as informações sejam armazenadas de forma simples e direta. O que este Protótipo apresenta é o *MySQL*, com o painel de controle *XAMPP* e executado a partir do *PHPMyAdmin*. A (Figura 14) demonstra como será a arquitetura completa do SQL, modelado pelo próprio *PHPMyAdmin*.

Figura 14 - Estrutura do banco de dados



Fonte: Elaborado pelo Autor

As Tabelas inseridas dentro do *PHPMyAdmin* foram projetadas com base no desenvolvimento dos diagramas, e necessitam umas das outras para que os dados saiam completos.

2.6 DICIONÁRIO DE DADOS

Tabela 3 - Tabela de dados do *login*

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_login_ID	INT	PK	Não	Identidade Única do usuário
rel_login_Usuario	VARCHAR(30)	-	Não	Usuário para Login
rel_login_Senha	VARCHAR(128)	-	Não	Senha do Usuário em HASH512
rel_login_Email	VARCHAR(256)	-	Não	Email do Usuário
rel_login_Nome	VARCHAR(100)	-	Não	Apelido do Usuário

Fonte: Elaborado pelo Autor

Na (Tabela 3), a Tabela de Login é onde são armazenados os dados do cadastro do usuário, podendo ser tanto alterados quanto consultados para fins de autenticação.

Tabela 4 - Tabela de dados do estoque

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_estoque_ID	INT	PK	Não	Identidade Única do Estoque
rel_estoque_Login_ID	INT	FK	Não	Identificador do Usuário que criou o Estoque
rel_estoque_Nome_Item	VARCHAR(255)	-	Não	Nome do estoque
rel_estoque_Tipo_Qtd	CHAR(1)	-	Não	Unidade de Medida Ex: Kg, g, L, Ml...
rel_estoque_Qtd	FLOAT	-	Não	Quantidade que tem no estoque
rel_estoque_Avisoqtd	FLOAT	-	Sim	Quantidade mínima para ser avisado caso que o valor retrocedeu o valor mínimo
rel_estoque_Fav	TINYINT(1)	-	Sim	Dado que guarda se foi selecionado como favorito

Fonte: Elaborado pelo Autor

A Tabela do estoque (Tabela 4) armazena os dados dos itens em estoque, e é a peça central de toda a estrutura, pois grande parte das funções do sistema se baseiam nesta tabela, tanto que é nela que fica definido se uma parte do estoque ficará destacável (rel_estoque_Fav), além de ser o (rel_estoque_Avisoqtd) o responsável por armazenar a quantidade mínima que terá no estoque.

Tabela 5 - Tabela de dados da lista

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_lista_ID	INT	PK	Não	Identidade Única da Lista
rel_lista_login_ID	INT	FK	Não	Identificador do Usuário que criou a Lista
rel_lista_nome	VARCHAR(30)	-	Não	Nome da Lista

Fonte: Elaborado pelo Autor

A Tabela da lista de compras (Tabela 5), armazena os dados das listas de compras, e seu principal uso é para a listagem delas no menu de listas de compras, e ela depende da tabela de Itens da lista, onde estão localizadas as informações dos itens da lista de compras.

Tabela 6 - Tabela de dados dos itens da lista

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_lista_compras_ID	INT	PK	Não	Identidade Única do Item
rel_lista_compras_lista_ID	INT	FK	Não	Identificador da Lista em que foi criado o Item
rel_lista_compras_nomeitem	VARCHAR(255)	-	Não	Nome do Item da Lista de Compras
rel_lista_compras_tipoqtd	CHAR(1)	-	Não	Unidade de medida Ex: Kg, g, L, Ml...
rel_lista_compras_qtd	FLOAT	-	Não	Quantidade que tem o Item da Lista de Compras

Fonte: Elaborado pelo Autor

A (Tabela 6) é a tabela que armazena os itens da lista, e descreve os itens que compõem as listas de compras. É usada principalmente para consulta e inserção no estoque.

Tabela 7 - Tabela de dados da receita

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_receita_ID	INT	PK	Não	Identidade Única da Receita
rel_receita_login_ID	INT	FK	Não	Identificador do Usuário que criou a Receita
rel_receita_Nome	VARCHAR(255)	-	Não	Nome da Receita

rel_receita_Minuto	INT	-	Não	Quantidade de minutos da Receita
---------------------------	------------	----------	------------	---

Fonte: Elaborado pelo Autor

A Tabela de dados da receita (Tabela 7) descreve as informações básicas das receitas, que são o nome e o tempo em minutos de preparo, e é usada para catalogar as receitas.

Tabela 8 - Tabela de dados do passo da receita

Nome da Tabela	Tipo	Chave	Nulo	Comentário
rel_receita_passo_ID	INT	PK	Não	Identidade Única do Passo
rel_receita_passo_rel_receita_ID	INT	FK	Não	Identificador da Receita que foi criada o Passo
rel_receita_passo_Nr_Passo	INT	-	Não	O número do Passo da Receita
rel_receita_passo_Descricao	VARCHAR(1000)	-	Sim	Descrição do passo da Receita

Fonte: Elaborado pelo Autor

A Tabela dos passos da receita (Tabela 8) armazena as etapas do preparo da receita, incluindo a descrição e a ordem numérica da etapa.

Tabela 9 - Tabela de dados do ingrediente da receita

Nome da Tabela	Tipo	Chave	Nulo	Comentário
-----------------------	-------------	--------------	-------------	-------------------

rel_receita_ingredientes_ID	INT	PK	Não	Identidade Única do Ingrediente
rel_receita_ingredientes_rel_receita_ID	INT	FK	Não	Identificador da Receita que foi criada o Ingrediente
rel_receita_ingredientes_Nome_Item	VARCHAR(255)	-	Não	Nome do Item da Receita
rel_receita_ingredientes_Tipo_Qtd	CHAR(1)	-	Não	Unidade de medida da Receita Ex: Kg, g, L, MI...
rel_receita_ingredientes_Qtd	FLOAT	-	Não	Quantidade do Item da Receita

Fonte: Elaborado pelo Autor

A Tabela de ingredientes da receita (Tabela 9) descreve quais itens compõem a receita, incluindo seu nome, quantidade, e unidade de medida.

3 TECNOLOGIAS A SEREM EMPREGADAS

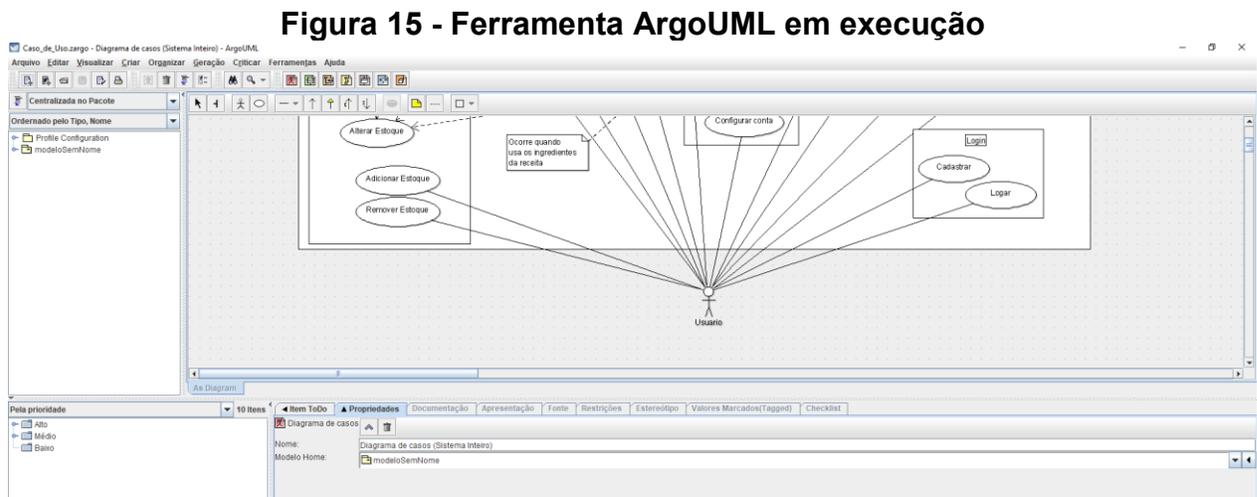
Antes de descrever as ferramentas utilizadas, é importante relatar que foram cogitados outros tipos de Tecnologias, tais como ASP.NET, entre outros.

A conclusão foi a escolha do PHP como plataforma de base do projeto, por ser uma linguagem simples e já conhecida pelos integrantes do grupo, facilitando assim o seu desenvolvimento.

Essa escolha facilitou, tanto para o início do Projeto, quanto ao seu gerenciamento de modo geral, e assim garantiu o bom desenvolvimento do Protótipo, graças aos inúmeros recursos que oferece.

3.1 ArgoUML

ArgoUML é uma ferramenta de modelagem UML *open-source* implementada em *Java*. Ele oferece suporte para os diagramas UML, e, por ser rodado em *Java*, é altamente compatível com diversos sistemas (Figura 15).



Fonte: Elaborado pelo Autor

O *ArgoUML* foi lançado em 1998, e foi finalista na premiação da *Software Development Magazine*, recebendo o prêmio Revelação em 2003.

3.2 CSS

O CSS (*Cascading Style Sheets*) é uma linguagem de estilo, ou seja, sua função é modificar a forma como o conteúdo de uma página é exibido, como espaçamento, fonte, cores, imagens, e outros elementos visuais, sendo compatível com as linguagens HTML e XML (Figura 16). O CSS possui, até o momento, três versões:

CSS1 – Foi a primeira versão, permitia a criação de *typefaces*, e cor de texto, planos de fundo, e a modificação de alguns elementos de texto como espaçamento entre palavras, letras e linhas.

CSS2 – Foi desenvolvido pela W3C em 1998, e a adição de ferramentas de posicionamento, que permitiram que fosse usada para a criação de layouts de página.

CSS3 – Diferente das versões anteriores, o CSS3 não é um bloco sólido de funções, e sim um conjunto de módulos, que foram sendo adicionados ao longo do tempo. Tais módulos são chamados de *Level 3*

CSS4 – Não existe nada com o nome oficial de CSS4. Entretanto, o que existe são módulos que aprimoram outros já existentes, também conhecidos como *Level 4*, e o conjunto de tais módulos pode, em teoria, ser chamado de CSS4.

A ideia do CSS foi lançada em 10 de outubro de 1994 por Håkon Wium Lie, juntamente com Tim Berners-Lee e a equipe da CERN, com a intenção de ser a linguagem de estilo padrão para a *World Wide Web*.

Figura 16 - Exemplo de linguagem em CSS

```

title {
margin: 0;
text-transform: uppercase;
font-weight: 900;
font-size: 90;
letter-spacing: 3px;
color: #1f145e;
text-shadow: -1px 1px 0px #aeb8fe,
             1px 1px 0 #aeb8fe,
             1px 1px 0 #aeb8fe,
             -1px -1px 0 #aeb8fe;
}
title span {
color: #aeb8fe;
}

```

Fonte: Elaborado pelo Autor

3.3 HTML

O HTML, conhecido por Linguagem de Marcação de Hipertexto (*HyperText Markup Language*) é o bloco de construção de páginas da web que pode ser acessado a partir de links. Pode-se conectar para páginas da web tanto para o próprio site quanto para outros sites. O HTML pode ser programado em qualquer bloco de texto, tanto que pode inserir texto e códigos, os códigos são utilizados para organizar estruturas dos textos, mudarem de tamanho, cor, imagem, botão, links, etc. Atualmente o HTML mais atual é a versão 5 em que foi inserido uma serie de melhorias e boas práticas na construção do site.

A origem do HTML foi em 1991 pelo físico Tim Berners-Lee na suíça, o foco na época do HTML foi conectar com outras instituições de pesquisas e para compartilhar os documentos entre outras entidades, um ano depois a WWW (*World Wide Web*) teve sua biblioteca liberada para alcance mundial, O HTML suporta o CSS, PHP, *JavaScript* e entre outros (Figura 17).

Figura 17 - Exemplo de programação em HTML

```
<html>
  <head>
  ..... Aqui fica o cabeçalho
  </head>
  <body>
  ..... <h1> Olá mundo </h1>
  </body>
</html>
```

Fonte: Elaborado pelo Autor

3.4 JavaScript

JavaScript é uma linguagem de programação com alguns recursos de orientação a objetos, e usada em sites para deixar a página de HTML mais dinâmica, ou seja, o *JavaScript* tem o poder de em tempo de execução criar *tags* HTML e exibir no navegador sem exigir a atualização da página (Figura 18).

JavaScript foi criado em 1995 por Brendan Eich, na *Netscape Communications*. O *JavaScript* foi criado para ser usado no navegador carro-chefe da empresa, o *Netscape*. Embora o nome seja *JavaScript*, a única relação com o *Java* são algumas similaridades da sintaxe, e o fato de a *Netscape* ser parceira da *Sun Microsystems*. No início, o *JavaScript* não era levado muito em consideração por ter sérios problemas de performance e segurança, mas isto mudou em 2008, com o lançamento do *Chrome V8*, que era uma ferramenta *JavaScript* de alta performance, possibilitando a criação de programas que competiam em desempenho com softwares desktop. Logo após isso, Ryan Dahl lançou o *Node.js*, permitindo que o *JavaScript* fosse executado fora do navegador (Figura 18).

Figura 18 - Exemplo de JavaScript que exibe um pop-up escrito “ReLista, 2021”

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <title>Exemplo JavaScript</title>
  <script>
    .....
    alert('ReLista, 2021');
  </script>
</head>
<body>
</body>
</html>
```

Fonte: Elaborado pelo Autor

3.5 MySQL

O *MySQL* é um Sistema Gerenciamento o Banco de Dados (SGBDs). Ele é bem conhecido por sua simplicidade, tanto que grandes empresas utilizam esse gerenciador em seus ambientes corporativos. A linguagem usada no gerenciador é SQL, que significa Linguagem de Consulta Estruturada (*Structured Query Language*), é o que opera com banco de dados que são baseadas em tabelas. Muitos desenvolvedores a utilizam para que tenham acesso aos dados que pode, tanto criar, editar ou removê-los. Na Figura 19 mostra-se como é estruturado um comando de seleção na linguagem SQL.

O MySQL foi criado por David Axmark, Allan Larsson e pelo Michael Widenius em 1980 na Suécia.

Figura 19 - Exemplo de seleção de registros em SQL

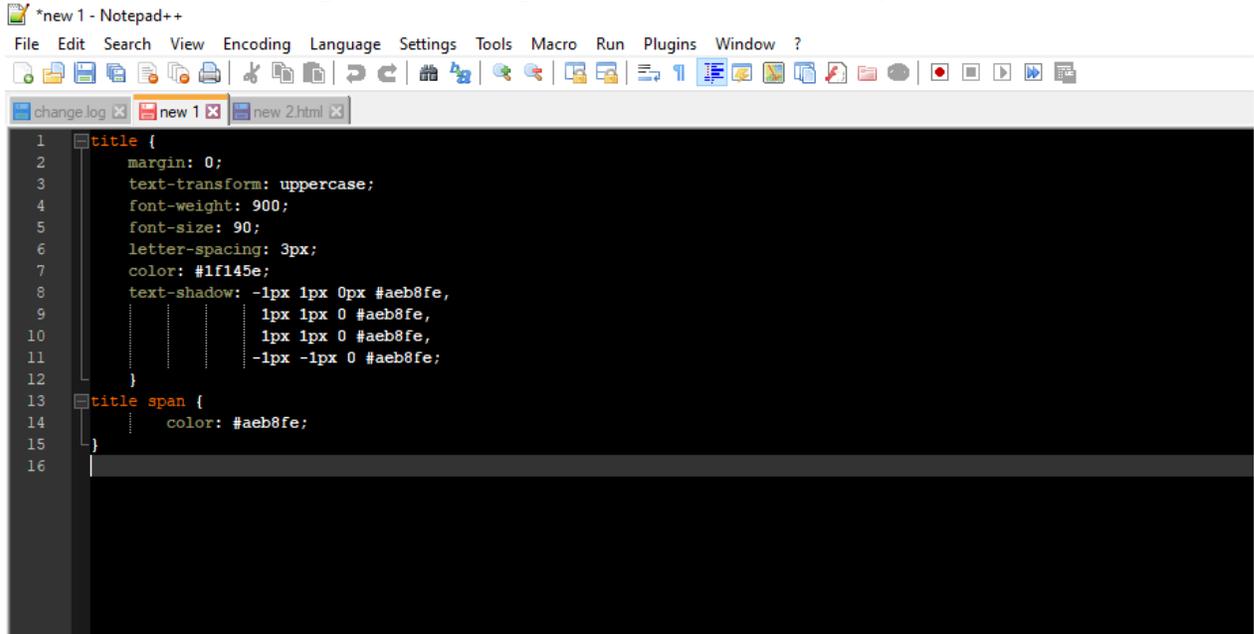
```
SELECT tabela_1.campo_1,  
       tabela_1.campo_2,  
       tabela_2.campo_3,  
       tabela_x.campo_n  
FROM   bd_1.tabela_1,  
       bd_2.tabela_2,  
       bd_x.tabela_x  
WHERE  condição_1,  
       condição_2
```

Fonte: https://bookdown.org/labxss/coorte_adm2/trabalhando-com-o-sgbd-mysql.html

3.6 Notepad++

O *Notepad++* é um editor de texto com uma funcionalidade bem simples, mas conta com uma gama de ferramentas úteis, especialmente para a programação Web.

Além de criar e editar textos e salvá-los em diversos formatos diferentes, o *Notepad++* conta com a função de determinar uma linguagem de programação. Ao selecionar uma linguagem, o sistema se adapta a ela, usando seu próprio esquema de padronização de cores e correções ortográficas para aquela linguagem específica, além de ferramentas de espaçamento que ajudam na organização do código. Entre as linguagens suportadas pelo *Notepad++* estão o HTML, CSS, *JavaScript* e PHP (Figura 20).

Figura 20 - Imagem da ferramenta Notepad++The image shows a screenshot of the Notepad++ text editor. The window title is '*new 1 - Notepad++'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations and editing. The editor has three tabs: 'change.log', 'new 1', and 'new 2.html'. The 'new 1' tab is active and contains the following CSS code:

```
1 title {
2   margin: 0;
3   text-transform: uppercase;
4   font-weight: 900;
5   font-size: 90;
6   letter-spacing: 3px;
7   color: #1f145e;
8   text-shadow: -1px 1px 0px #aeb8fe,
9               1px 1px 0 #aeb8fe,
10              1px 1px 0 #aeb8fe,
11              -1px -1px 0 #aeb8fe;
12 }
13 title span {
14   color: #aeb8fe;
15 }
16
```

Fonte: Elaborado pelo Autor

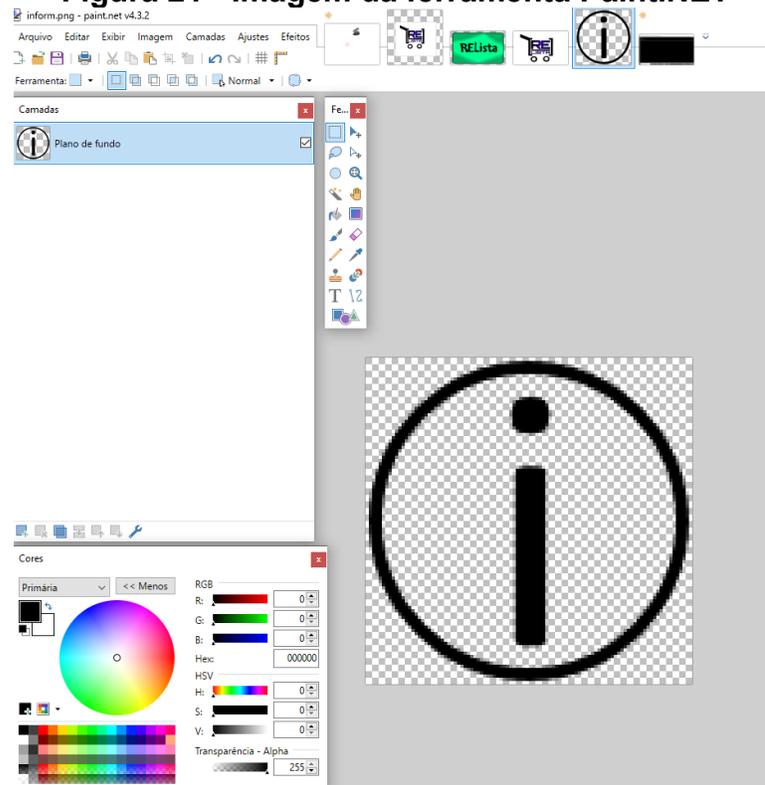
O *Notepad++* foi criado em 2003 por Don Ho, que estava insatisfeito com o editor de código fonte usado por sua empresa na época (o JEXT), e decidiu criar um por conta própria.

3.7 *Paint.NET*

Paint.NET é um editor *freeware* para fotos e imagens, que roda no sistema operacional *Windows*. Um dos pontos fortes é a configuração de camadas e transformação de imagens em png, o que ajuda no desenvolvimento das imagens que foram produzidas com ela.

Foi uma ferramenta essencial para criar ícones e imagens para o Projeto, como pode ser visto na Figura 21

Figura 21 - Imagem da ferramenta Paint.NET



Fonte: Elaborado pelo Autor

Foi desenvolvido por Rick Brewster que na época queria um software que podia substituir o *Paint* da *Microsoft*, tanto que foi comparado com outros softwares de editor de imagens, tanto pagos, quanto gratuitos.

3.8 PHP

O PHP (*Personal Home Page*) é uma linguagem de programação dinâmica para a produção de *websites*.

O PHP é processado no servidor, retornando para o usuário (pessoa que acessa o site) em apenas HTML.

O PHP não é uma linguagem de programação COMPILADA como C++ ou JAVA. É uma linguagem de programação interpretada pelo servidor, pois não gera um executável, nem códigos de máquina.

Um script PHP pode conter ou não *tags* de HTML. Essas *tags* não são processadas pelo servidor, são simplesmente passadas ao solicitante. Normalmente

utiliza-se HTML para fazer a estrutura e parte estática da página, e o PHP para a parte lógica, que exige processamento. A (Figura 22) demonstra um exemplo do PHP em que escreve a palavra dentro do parâmetro do HTML.

Figura 22 - Exemplo de PHP que puxa a data do dia do servidor

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>

    <?php
      echo "Olá, eu sou um script PHP!";
    ?>

  </body>
</html>
```

Fonte: Desenvolvendo Websites com PHP

Criado em 1994 por Rasmus Lerdorf, originalmente um conjunto de binários *Common Gateway Interface* (CGI), escritos em linguagem C, chamado de *Personal Home Page Tools*, ou *PHP Tools*. Seu propósito original era acompanhar o número de visitas do currículo on-line de Rasmus, mas com o passar do tempo e a implementação de novas ferramentas, ele reescreveu o PHP, permitindo agora interações com banco de dados e, com a nova estrutura, os usuários então eram capazes de criar aplicações web. Em junho de 1995, Rasmus tornou público o código fonte do *PHP Tools*.

3.9 *PHPMyAdmin*

PHPMyAdmin é um software gratuito que foi escrito na linguagem PHP, com a intenção de ser o gerenciador do *MySQL* no ambiente da Web, no intuito de gerenciar o banco de dados, criar tabelas, alterar campos, inserir registros e entre outras funções que podem ser executadas no *MySQL*.

O *PHPMYAdmin* será utilizado para administrar o banco de dados do Projeto ReLista. A Figura 23 apresenta um exemplo dele em funcionamento:

Figura 23 - Ferramenta phpMyAdmin em execução

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of databases and tables. The main panel displays the 'rel_login' table with the following data:

rel_login_ID	rel_login_Usuario	rel_login_Senha
5	admin	c7ad44cbad762a5da0a452f9e854fdc1e0e7a52

The interface also shows a SQL query editor with the query `SELECT * FROM `rel_login`` and various control buttons like 'Mostrar tudo', 'Número de registros', and 'Filtrar registros'.

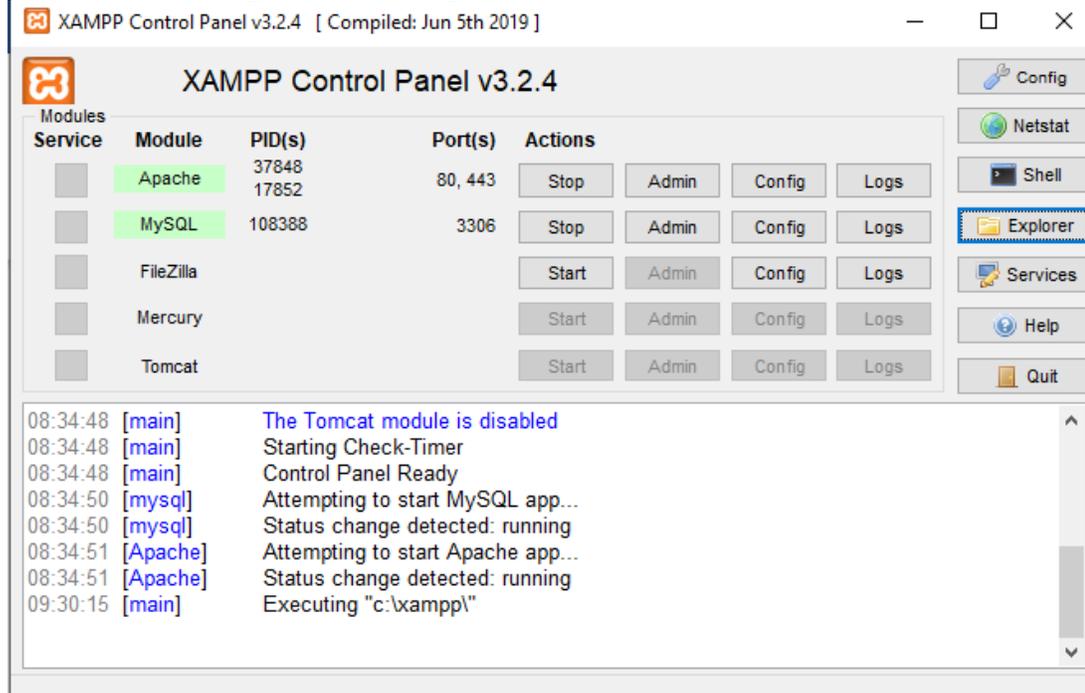
Fonte: Elaborado pelo Autor

O software foi desenvolvido por Tobias Ratschiller em 1998, e o próprio autor realizou a manutenção até 2002.

3.10 XAMPP

XAMPP é uma ferramenta essencial para o desenvolvimento dos arquivos PHP, tanto que nele vem o conteúdo *MySQL* e do PHP. Ele foi desenvolvido a partir dos *Apache Friends* que realizaram esse projeto sem fins lucrativos. O software é uma das ferramentas que ajuda na conexão do *MySQL* e do *PHPMYAdmin*, além da produção de páginas web. A Figura 24 mostra um exemplo do software em execução.

Figura 24 - Imagem do XAMPP em execução



Fonte: Elaborado pelo Autor

O XAMPP foi fundado por Kai 'Oswald' Seidler e Kay Vogelgesang em 22 de maio de 2002, com o intuito de facilitar na instalação dos pacotes para os outros usuários que estão começando o ramo.

4 PROJETO

Este capítulo demonstra quais serão principais funcionalidades do sistema, como elas foram projetadas, e como elas serão exibidas para o usuário.

4.1 PROBLEMÁTICA

Embora certas pessoas criem as suas listas, ainda existe a probabilidade de esquecer quais mercadorias deveriam ser compradas, por não terem a visão total do que tem em sua residência, e quando retornam depois das suas compras, descobrem no dia seguinte que precisam de mais uma mercadoria, o que leva novamente à locomoção ao mercado. O software apresentado neste Protótipo pode resolver esse problema de ir várias vezes ao mercado, e até ter uma lista de quais itens precisam ser comprados.

Tabela 10 - Comparativo de funcionalidades de outros aplicativos

Funcionalidades	ReLista	Vidder	Sortly	OrganizEat
Ambiente de cadastro de objetos	X	X	X	
Ambiente de gerar lista de compras	X			
Ambiente de guardar receitas	X			X
Consulta de estoque baseado em receitas	X			
Rastreamento de item baseado em QR code		X		

Fonte: Elaborado pelo Autor

4.2 PROTÓTIPO

Após a documentação do sistema, foi desenvolvido o Protótipo que apresentará as funcionalidades da aplicação.

O sistema possuirá as seguintes funcionalidades:

- Função de login: é verificado se o usuário existe para que não haja acesso de pessoas não autorizadas;
- Função de cadastro de receitas;
- Função de geração de lista de compras;
- Função de controle de estoque.

4.3 ARQUITETURA E TECNOLOGIA

Seguem abaixo as imagens do Protótipo da tela de adição de itens ao estoque e da lista de compras:

Figura 25 - Tela de Estoque



Fonte: Elaborado pelo Autor

Como pode ser visto, na Figura 25, ao pressionar o botão 'Adicionar item' em laranja, serão exibidas algumas caixas de texto com os campos: Nome do produto, Quantidade, Mínimo de Aviso e Unidade de Medida (kg, g, l, ml e uni.). Os campos Nome do produto e Quantidade são obrigatórios. Ao pressionar o botão verde, 'Adicionar', ele chamará o adicionarItemEstoque.php

Figura 26 – Código PHP de adicionar item do estoque

```

include_once dirname(__FILE__).'/bancoded.php'; // Chama o comandos do bancoded.php
$response=array();

if($_SERVER['REQUEST_METHOD']=='POST'){

    if ( isset($_POST['nome']) && isset($_POST['qtd']) && isset($_POST['tipoqtd'])) { // Verifica se foi inserido o nome do estoque, quantidade e
        $db=new bancoded();

        if ( $db->verificaEstoqueusu($_POST['nome'], $_SESSION['usuario'], $_SESSION['senha'])) { // Verifica se existe o estoque
            header('Location: index.php?pagina=estoque');
            $_SESSION['erro_estexiste'] = true;
            exit;
        }

        if ( $db->adicionarEstoque($_POST['nome'],$_POST['qtd'],$_POST['tipoqtd'],$_POST['minimo'], $_SESSION['usuario'], $_SESSION['senha'])) {
            $db->disconnectFromDB();
            header('Location: index.php?pagina=estoque');
            $_SESSION['adicionarestoque_sucesso']=true;
            exit;
        }
    }
}

```

Fonte: Elaborado pelo Autor

Logo de início, como mostrado na Figura 26, inclui-se o bancoded.php, que é responsável pela comunicação do banco de dados, e verifica-se se existe o estoque no banco de dados. Em caso verdadeiro, ele irá retornar à tela de estoque com um aviso falando que esse estoque já existe. Caso contrário, ele envia as informações na função do bancoded.php para cadastrar as informações inseridas no banco de dados, como pode ser visto na Figura 27.

Figura 27 - Função de adicionar no banco de dados

```

/^ Adiciona o estoque ^/
function adicionarEstoque($nome,$qtd,$tipoqtd,$avisoqtd,$usuario,$senha) {
    $id = $this->retornaID($usuario,$senha);
    $stmt=$this->con->prepare("insert into rel_estoque values (null,?, ?, ?, ?, ?, 0)");
    $stmt->bind_param("issdd", $id,$nome,$tipoqtd,$qtd,$avisoqtd);
    return $stmt->execute();
}

```

Fonte: Elaborado pelo Autor

Depois de inserido, ele retornará à página de estoque com a confirmação que o estoque foi inserido com sucesso.

Figura 28 - Tela da Lista de Compras

Fonte: Elaborado pelo Autor

A Figura 28 mostra a tela de criação da lista de compras. Logo na parte superior desta tela, vê-se a caixa de texto 'procurar', que é um filtro para procurar por itens específicos na lista. Logo abaixo, existe outra caixa de texto, desta vez escrito 'Nome da Lista', onde deve ser inserido um nome para a lista que está sendo criada no momento. Em seguida, abaixo da caixa 'Nome da Lista', à esquerda, há um botão 'Adicionar Item: +'. Pressionar este botão serve para criar mais caixas no campo de registro da lista, onde os itens são especificados, que fica logo abaixo deste botão. Na área de registro, devem ser descritos os itens que irão compor a lista, inserindo o nome, a quantidade e a unidade de medida dos itens, além de um 'X' ao lado de cada item, para facilitar a remoção deles. Abaixo do campo de registro, existem dois botões: Adicionar e Cancelar.

Adicionar: Transfere as informações da área de registro para a lista de compras, logo abaixo.

Cancelar: Deleta tudo que está na área de registro.

Abaixo destes botões está a lista em si, onde os itens adicionados da área de registro são armazenados, e com as seguintes opções ao lado: Visualizar Lista (Folha de Papel), Configurar Lista (Engrenagem) e Deletar Lista ('X' Vermelho).

Visualizar Lista: Abre uma janela que exibe o conteúdo da lista.

Configurar Lista: Exibe os itens da lista e permite que eles sejam alterados.

Deletar lista: Deleta a lista em questão.

Para que seja possível adicionar mais itens, foi criada uma tabela com identificador tabela como pode ser visto na Figura 29:

Figura 29 - Código HTML de adicionar item da lista de compras

```
<div>
  Adicionar Item:
  <span class="adicionar" onclick="AdicionarLista('tabela',0)"> </span>
</div>
<div class="lista">
  <center>
  <table id="tabela">
  </table>
  </center>
</div>
```

Fonte: Elaborado pelo Autor

Ao pressionar o botão de adicionar, que é identificado como o sinal de (+) verde, ele chama a função AdicionarLista que fica dentro do *JavaScript*: java.js (Figura 29).

Figura 30 - Código JS de adicionar uma linha na tabela

```
function AdicionarLista($tabela, $tipo) {
  $tabela = IDElemento($tabela);

  $newRow = $tabela.insertRow(lista);
  $coluna1 = $newRow.insertCell(0);
  $coluna2 = $newRow.insertCell(1);
  $coluna3 = $newRow.insertCell(2);
  $coluna4 = $newRow.insertCell(3);

  $coluna1.innerHTML = "<input name='listal["+lista+"]' required class='AdicionarEstoque' placeholder='Nome do Item' id='listal["+lista+"]' type='text'";
  $coluna2.innerHTML = "<input name='lista2["+lista+"]' required class='AdicionarEstoque' placeholder='Qtd' id='lista2["+lista+"]' type='text'";
  $coluna3.innerHTML = "<select name='lista3["+lista+"]' required id='lista3["+lista+"]' class='AdicionarEstoque' name='tipoqtd' required> <option";
  $coluna4.innerHTML = "<span class='remover' id='listaspan["+lista+"]' onclick='RemoverLista('tabela','+lista+')'> </span>";

  lista++;
}
```

Fonte: Elaborado pelo Autor

Ele pega o identificador tabela e insere uma nova linha na tabela (Figura 30), e depois cria-se as células. E para que sejam mostrados os campos de texto e a lista de tipo de quantidade, para inserir dentro do banco de dados, foi feita uma inserção diferente da do próprio estoque.

Figura 31 - Código PHP para adicionar uma lista de compras

```

if ( !isset($_POST['nome'])) { // Verifica se foi inserido o nome da lista
$db=new bancoded();

if ( $db->verificaListausu($_POST['nome'], $_SESSION['usuario'], $_SESSION['senha'])) { // Verifica se existe na lista
    header('Location: index.php?pagina=lista');
    $_SESSION['erro_estexiste'] = true;
    exit;
}

$db->adicionarLista($_POST['nome'],$_SESSION['usuario'], $_SESSION['senha']);

$idlista = $db->retornarIDLista($_POST['nome'],$_SESSION['usuario'], $_SESSION['senha']);

$contador = 0;
$total = count($_POST["contador"]);

/* OK */

while ($contador < $total) :

    $db->adicionarListaCompras($idlista,$_POST['lista1'][$contador],$_POST['lista2'][$contador],$_POST['lista3'][$contador],$_SESSION['usuario'],$_SESSION['senha']);
    $contador++;

endwhile;

$db->disconnectFromDB();
header('Location: index.php?pagina=lista');
$_SESSION['adicionarestoque_sucesso']=true;
exit;

```

Fonte: Elaborado pelo Autor

O que muda é que se captura o identificador (ID) da lista que foi criada e em seguida se captura a quantidade de linhas na tabela, e dentro de um *loop* é feita a contagem de 0 até o número total subtraído por 1, e então são inseridos um a um os itens da lista de compras (Figura 31).

4.4 TELAS

Serão exibidas as telas que compõem o projeto, através das quais o usuário poderá ter acesso às diversas funcionalidades oferecidas.

Figura 32 – Tela de Login do Protótipo

The image shows a login form with a blue header bar containing 'Entrar' and 'Cadastrar' buttons. Below the header is a grey box with the title 'Login'. The form includes two input fields: 'Usuário' with the placeholder text 'Usuario' and 'Senha' with the placeholder text 'Senha'. To the right of the 'Senha' field is an eye icon for toggling password visibility. Below the input fields is a red button labeled 'Login'. At the bottom of the form, there is a link: 'Não tem uma conta? [Cadastre já](#)'. The footer of the page contains the text: 'Projeto TCC - FATEC 2S/2021 desenvolvido por Gabriel Martins de Oliveira e Hugo Tavares Rodrigues'.

Fonte: Elaborado pelo Autor

A tela de Login (Figura 32) exibe dois campos: Usuário e Senha. No campo da senha, há o botão que permite que o conteúdo do campo Senha seja desmascarado. Abaixo destes campos há o botão de Login.

Figura 33 – Tela de Cadastro de Conta do Protótipo

O protótipo da tela de cadastro apresenta uma interface com uma barra superior azul contendo os botões "Entrar" e "Cadastrar". O título "Cadastro" está centralizado na área principal. O formulário possui cinco campos de entrada:

- Apellido: Digite o seu apelido
- Email: Digite o email
- Usuário: Digite o usuario
- Senha: Digite a senha (com ícone de olho para alternar visibilidade)
- Senha Novamente: Digite a senha novamente (com ícone de olho para alternar visibilidade)

Abaixo dos campos, há um botão de "Cadastro" em um fundo vermelho. Na base da tela, há o texto "Já possui cadastro? [Entre já](#)" e o rodapé "Projeto TCC - FATEC 2S/2021 desenvolvido por Gabriel Martins de Oliveira e Hugo Tavares Rodrigues".

Fonte: Elaborado pelo Autor

A tela de Cadastro (Figura 33) possui os seguintes campos para preenchimento: Apellido, Email, Usuário, Senha, e Repetição da Senha. Abaixo destes campos há o botão de Cadastrar.

Figura 34 - Tela de criação de Receitas

The screenshot shows a mobile application interface for creating recipes. At the top, there is a button labeled "Adicionar Ingrediente: +" with a green plus icon. Below it is a green background area containing four rows of input fields. Each row has a text input field labeled "Nome do Item", a numeric input field labeled "Qtd", and a unit selection dropdown menu labeled "Kg" with a red "X" icon to its right. Below this area is another button labeled "Adicionar Passo: +" with a green plus icon. This is followed by a green background area containing seven rows of input fields. Each row is labeled "Passo 1" through "Passo 7" and contains a text input field labeled "Nome do Item" with a red "X" icon to its right.

Fonte: Elaborado pelo Autor

A tela de criação da lista de compras (Figura 34) é composta pelo botão “Adicionar Ingredientes: +”, que cria campos para a inserção de itens na lista abaixo, e “Adicionar Passo: +”, que cria campos para a inserção do texto que descreve o preparo, dividido em “Passos”.

Figura 35 – Tela da Lista de Receitas

The screenshot shows a mobile application interface for a recipe list. At the top left is a logo with the text "RECEITAS" and a shopping cart icon. At the top right is a user profile icon and a red arrow pointing to a door icon. Below the header is a navigation bar with four tabs: "Receitas", "Estoque", "Lista de Compras", and "Sobre". The main content area is titled "Receita" and features a search bar labeled "Procurar...". Below the search bar is a large orange button labeled "Adicionar Receita". Underneath this button are two text input fields: "Nome da Receita" and "Minutos". Below these fields is a button labeled "Adicionar Ingrediente: +" with a green plus icon. This is followed by a green background area containing one row of input fields: a text input field labeled "Nome do Item", a numeric input field labeled "Qtd", and a unit selection dropdown menu labeled "Kg" with a red "X" icon. Below this area is another button labeled "Adicionar Passo: +" with a green plus icon. This is followed by a green background area containing one row of input fields: a text input field labeled "Passo 1" and "Nome do Item" with a red "X" icon. At the bottom of the form are two large buttons: a green "Adicionar" button and a red "Cancelar" button. At the very bottom of the screen, there is a grey bar showing "Bolo de Chocolate 160" and three icons: a list icon, a gear icon, and a red "X" icon.

Fonte: Elaborado pelo Autor

A tela de Lista de Receitas (Figura 35) exibe as receitas já criadas, além do botão “Adicionar Receita, que exibe os campos Nome da Receita e Minutos, além dos outros campos descritos na Figura 34.

Figura 36 - Tela Sobre



Fonte: Elaborado pelo Autor

A tela Sobre (Figura 36) exibe as informações a respeito do projeto, tanto seus objetivos quanto seu papel acadêmico.

5 CONSIDERAÇÕES FINAIS

A proposta deste Projeto é apresentar o Protótipo de uma aplicação Web para o Controle de Estoque de Alimentos, que permitirá obter um acesso mais preciso a informações relacionadas a armazenamento e receitas do dia-a-dia.

Para o desenvolvimento deste Protótipo, foram averiguadas as ferramentas de desenvolvimento WEB disponíveis, tendo sido escolhidos o *Notepad++* para a escrita dos códigos. Por se tratar do Projeto de uma aplicação web, será necessário não somente um SGBD para o Banco de Dados, mas também um administrador intermediário entre o BD e a WEB, e, para estas funções, foram escolhidos o *MySQL* e o *PHPMyAdmin* respectivamente. Além disso, se fez necessário também o uso do *ArgoUML* para a criação dos diagramas, para estruturar o Projeto e permitir uma visão mais clara dos planos, não somente para os autores, como também para os leitores do mesmo.

Ao longo do desenvolvimento deste Protótipo, foram encontradas algumas dificuldades, mas a maior delas foi a adaptação para o desenvolvimento web, área com a qual nenhum dos autores era muito familiarizado.

No futuro, os planos envolvem a disponibilização do software na *AppStore*, a implementação do controle das datas de validade e do dinheiro gasto em relação ao estoque, o uso de *jQuery* e *Ajax* para permitir a atualização de páginas sem que elas precisem ser recarregadas, a possibilidade de tornar receitas como Públicas, permitindo que outros usuários as visualizem, e importação das listas de compras em arquivo PDF ou bloco de notas. Além disso, planeja-se o compartilhamento de estoques e listas de compras por meio de chaves únicas, com a possibilidade de permitir ou não que aqueles com quem o estoque e/ou lista foram compartilhados editem o conteúdo deles.

Em suma, o Protótipo descrito da introdução à conclusão deste documento foi elaborado com sucesso, e a criação do software aqui descrito, que será capaz de gerenciar estoques e gerar listas baseadas em tais estoques, além de registrar receitas, está plenamente elaborada e a possibilidade de sua criação foi atestada.

REFERÊNCIAS

Apache Friends team. **O que é o XAMPP?**. on-line, 2021, Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 14 nov. 2021.

Apache Friends team. **Sobre**. on-line, 2021, Disponível em: <https://www.apachefriends.org/pt_br/about.html>. Acesso em: 14 nov. 2021.

BACCA, Carolina Cozer. **O que é SQL e pra que ele serve?**. on-line, 2019, Disponível em: <[https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm#:~:text=SQL%20significa%20“Structured%20Query%20Language,relacional%20\(baseado%20em%20tabelas\)](https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm#:~:text=SQL%20significa%20%20Structured%20Query%20Language,relacional%20(baseado%20em%20tabelas))>. Acesso em 15 nov. 2021.

FERRÉ, Felipe. **Modelagem de coortes com dados administrativos**: Trabalhando com o SGBD MySQL. on-line, 2020, Disponível em: <https://bookdown.org/labxss/coorte_adm2/trabalhando-com-o-sgbd-mysql.html>. Acesso em: 15 nov. 2021.

GUEDES, Gilleanes T. A. **UML2: UMA ABORDAGEM PRÁTICA**. 3ª Edição. Editora: Novatec. São Paulo, 2018.

HERTEL, Rafael. **O Que é HTML5 e Quais Vantagens ele Traz para seu Site**. on-line, 2021, Disponível em: <<https://www.hostinger.com.br/tutoriais/diferenca-entre-html-e-html5>> Acesso em: 13 nov. 2021.

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 3ª Edição. Editora: Novatec. São Paulo, 2017.

ORIN, Andy. **Behind the App: The Story of Notepad++**. On-line, 2021, Disponível em <<https://lifelhacker.com/behind-the-app-the-story-of-notepad-1711936108>>. Acesso em: 16 nov. 2021.

ORTIZ, Alexandra. **How to Create Block Text Using the CSS Text-Shadow Property.** on-line, 2021, Disponível em: <<https://levelup.gitconnected.com/how-to-create-block-text-using-the-css-text-shadow-property-f43d4f5cdb84>>. Acesso em: 15 nov. 2021.

PACIEVITCH, Yuri. **HTML.** on-line, Disponível em: <<https://www.infoescola.com/informatica/html/#:~:text=O%20HTML%20foi%20criado%20em,e%20compartilhar%20documentos%20com%20facilidade>>. Acesso em: 13 nov. 2021.

RIBEIRO, Halyne D'angelo de Oliveira. **Consumo x consumismo: você sabe a diferença, as motivações?** on-line, 2019. Disponível em: <<https://procon.pb.gov.br/noticias/consumo-x-consumismo-voce-sabe-a-diferenca-as-motivacoes>>. Acesso em: 02 dez. 2020.

RODRIGUES, Paula. **Os desperdícios por trás do alimento que vai para o lixo.** on-line, 2017, Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/28827919/os-desperdicios-por-tras-do-alimento-que-vai-para-o-lixo>>. Acesso em: 15 nov. 2021.

SOMMERVILLE, I. **Engenharia de Software.** 8ª Edição. Editora: Pearson Addison-Wesley. São Paulo, 2019.

_____. **HTML: Linguagem de Marcação de Hipertexto:** Tutoriais para iniciantes. on-line, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 13 nov. 2021.

_____. **About.** on-line, 2021, Disponível em: <<https://www.phpmyadmin.net>>. Acesso em: 14 nov. 2021.

_____. **Credits:** Credits, in chronological order. on-line, 2012 – 2021, Disponível em: <<https://docs.phpmyadmin.net/en/latest/credits.html>>. Acesso em: 14 nov. 2021.

corelangs. **CSS Versions.** on-line, 2020, Disponível em: <<http://www.corelangs.com/css/basics/versions.html>>. Acesso em: 14 nov. 2021.

_____. **A Brief JavaScript History.** on-line, Disponível em: <<https://launchschool.com/books/javascript/read/introduction>>. Acesso em: 15 nov. 2021.

_____. **Origins and Overview of ArgoUML.** On-line, Disponível em: <<https://web.archive.org/web/20200123011308/http://argouml-stats.tigris.org/documentation/manual-0.32/ch01.html#d0e213>>. Acesso em: 17 de nov. 2021.

_____. **História do PHP.** on-line, Disponível em: <https://www.php.net/manual/pt_BR/history.php.php>. Acesso em: 15 nov. 2021.

_____. **O que é o PHP?.** on-line, Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php#example-1> Acesso em: 15 nov. 2021.

_____. **CSS: Tutoriais.** on-line, 2021, Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 14 nov. 2021.

_____. **CSS History, A Brief Overview.** on-line, Disponível em: <<https://simplecss.eu/css-history-brief-overview.html>>. Acesso em: 14 nov. 2021.

_____. **Interview with Kai Seidler from the XAMPP project.** on-line, 2006, Disponível em: <<https://web.archive.org/web/20090628133036/http://dev.mysql.com/tech-resources/interviews/kai-seidler-xampp.html>>. Acesso em: 14 nov. 2021.

_____. **About.** on-line, Disponível em: <<https://www.getpaint.net>>. Acesso em: 14 nov. 2021.

