**UNIVERSIDADE VEIGA DE ALMEIDA**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**LISTA DE COMPRAS INTELIGENTE**

**GABRIEL RAMOS DE SOUSA**

**RIO DE JANEIRO**

**2021**

**UNIVERSIDADE VEIGA DE ALMEIDA**

**GABRIEL RAMOS DE SOUSA**

Trabalho de Conclusão de Curso apresentado no curso de Sistemas

de Informação da Universidade Veiga de Almeida como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: André Lucio de Oliveira

**LISTA DE COMPRAS INTELIGENTE**

**RIO DE JANEIRO**

**2021**

Assinatura do aluno

Gabriel Ramos de Sousa – 20151112918

Trabalho acadêmico apresentado em / /

Orientador Prof. André Lucio de Oliveira

1º. Examinadora Prof. Dra. Manoela dos Santos

2º. Examinador Prof. Msc. Marcos dos Anjos

Coordenador Prof. Dr. Claudio Fico Fonseca

*Este trabalho acadêmico é dedicado à Deus e a minha família.*

**AGRADECIMENTOS:**

Agradeço à Deus, aos meus familiares, professores e amigos pela conclusão deste trabalho acadêmico. Agradecimentos especiais aos meus orientadores Profª. Adriana Aparicio Sicsú Ayres do Nascimento e Prof. André Lucio de Oliveira.

**SUMÁRIO**

[1. INTRODUÇÃO 8](#_Toc87711465)

[1.1. Nome do Sistema 8](#_Toc87711466)

[1.2. Cenário Atual 8](#_Toc87711467)

[1.3. Objetivos 8](#_Toc87711468)

[1.4. Justificativa 9](#_Toc87711469)

[2. REQUISITOS DO SISTEMA 10](#_Toc87711470)

[2.1. Descrição Geral do Sistema 10](#_Toc87711471)

[2.2. Partes Interessadas 13](#_Toc87711472)

[2.3. Atores 13](#_Toc87711473)

[2.4. Especificação dos Requisitos Funcionais 14](#_Toc87711474)

[2.4.1. Especificação de Casos de Uso 17](#_Toc87711475)

[2.4.2. Regras de Negócio 27](#_Toc87711476)

[2.5. Especificação dos Requisitos Não Funcionais 30](#_Toc87711477)

[3. MODELO DE DADOS 31](#_Toc87711478)

[4. GESTÃO DE PROJETO 35](#_Toc87711479)

[5. SOLUÇÃO ARQUITETURAL E TECNOLÓGICA 37](#_Toc87711480)

[6. PROJETO E EXECUÇÃO DOS TESTES 38](#_Toc87711481)

[6.1. Método de Versionamento 38](#_Toc87711482)

[6.2. Testes 39](#_Toc87711483)

[6.2.1. Testes de Sistemas 39](#_Toc87711484)

[6.2.2. Teste de integridade 39](#_Toc87711485)

[6.2.3 Teste de compatibilidade 40](#_Toc87711486)

[6.2.4 Teste de Funcionalidade 40](#_Toc87711487)

[7. CONCLUSÕES E TRABALHOS FUTUROS 40](#_Toc87711488)

[REFERÊNCIAS 43](#_Toc87711489)

**LISTA DE FIGURAS**

Figura 1 –Diagrama de caso de uso 15

Figura 2 – Modelo de dados 32

Figura 3 – Quadro Kanban no Trello 36

Figura 4 – Modelo de Arquitetura 38

**LISTA DE TABELAS**

Tabela 1 – Visão do produto 13

Tabela 2 – Partes interessadas 14

Tabela 3 – Atores 14

Tabela 4 – RF01 – Realizar cadastro 16

Tabela 5 – RF02 – Efetuar login 16

Tabela 6 – RF03 – Alterar dados cadastrais 16

Tabela 7 – RF04 – Manter ingrediente 16

Tabela 8 – RF05 – Manter receita 16

Tabela 9 – RF06 – Manter receita ingrediente 16

Tabela 10 – RF07 – Manter modo de preparo 17

Tabela 11 – RF08 – Planejar cardápio semanal 17

Tabela 12 – RF09 – Gerar lista de compras 17

Tabela 13 – Descrição de caso de uso – Realizar cadastro 18

Tabela 14 – Descrição de caso de uso – Efetuar login 19

Tabela 15 – Descrição de caso de uso – Alterar dados cadastrais 20

Tabela 16 – Descrição de caso de uso – Manter ingredientes 21

Tabela 17 – Descrição de caso de uso – Manter receita 23

Tabela 18 – Descrição de caso de uso – Manter receita ingrediente 24

Tabela 19 – Descrição de caso de uso – Manter modo preparo 25

Tabela 20 – Descrição de caso de uso – Manter cardápio semanal 26

Tabela 21 – Descrição de caso de uso – Gerar lista de compras 27

Tabela 22 – RN01 – Validações na tela de criar conta de usuário 27

Tabela 23 – RN02 – Validações na tela de login 27

Tabela 24 – RN03 – Editar perfil de usuário 28

Tabela 25 – RN04 – Validações na tela de ingrediente 28

Tabela 26 – RN05 – Exclusão ingrediente 28

Tabela 27 – RN06 – Categoria de receita 28

Tabela 28 – RN07 – Validações na tela de receita 28

Tabela 29 – RN08 – Validação do cadastro de receita completo 28

Tabela 30 – RN09 – Validações na tela de receita ingrediente 29

Tabela 31 – RN10 – Unidade de medida 29

Tabela 32 – RN11 – Tipos de unidade de medida 29

Tabela 33 – RN12 – Quantidade de ingredientes por receita 29

Tabela 34 – RN13 – Duplicidade de ingredientes na receita 29

Tabela 35 – RN14 – Validações na tela de modo de preparo 29

Tabela 36 – RN15 – Validações na tela cardápio semanal 30

Tabela 37 – RN16 – Listagem dos cardápios semanais 30

Tabela 38 – RN17 – Calculo de conversão de unidade de medida 30

Tabela 39 – RN18 – Lista de compras 30

Tabela 40 – RN19 – Duplicidade cardápio semanal 30

Tabela 41 – RNF01 – Usabilidade 31

Tabela 42 – RNF02 – Tecnologia 31

Tabela 43 – RNF03 – Segurança de acesso 31

Tabela 44 – RNF04 – Ambiente 31

Tabela 45 – Dicionário de dados 34

Tabela 46 – Gráfico das demandas no Trello 36

# 1. INTRODUÇÃO

## 1.1. Nome do Sistema

Lista de Compras Inteligente

## 1.2. Cenário Atual

Diversas pessoas já se encontram na situação de ir ao mercado comprar os ingredientes para o preparo de alguma refeição que iria preparar para o almoço, janta ou até mesmo um lanche e quando chega em casa lembra que faltou comprar algo ou até mesmo não comprou porque não sabia que precisa do ingrediente para a receita e com as diversas responsabilidade diárias temos cada vez menos tempo de ficar indo ao mercado diversas vezes, até mesmo pelo gasto exagerado quando acabamos não comprando algo ou compramos em excesso e acabamos tendo que jogar os alimentos fora, seja por serem perecíveis como frutas, legumes ou por data de validade vencida.

Atualmente no mercado existem diversas aplicações e sites que disponibilizam receitas, porém sua grande maioria não possui nenhuma forma de exportar os ingredientes das receitas para uma lista de compras.

## 1.3. Objetivos

A Lista de Compras Inteligente tem como objetivo otimizando gastos com compras, diminuir o desperdício de alimentos, poupar o tempo com compras diárias e planejar um cardápio de refeições mais saudáveis.

O sistema deve abranger:

* As rotinas relacionadas a login e logout.
* As rotinas relacionadas à criação e atualização dos usuários.
* As rotinas relacionadas a criação, consulta, atualização e destruição de dados (CRUD) dos ingredientes.
* As rotinas relacionadas a criação, consulta, atualização e destruição de dados (CRUD) das receitas.
* As rotinas relacionadas a criação, consulta, atualização e destruição de dados (CRUD) dos cardápios semanais.
* Gerar relatórios das listas de compras contendo todos os ingredientes necessários para o preparo de todas as receitas descritas no cardápio semanal selecionado.

## 1.4. Justificativa

Em aplicativos similares a lista de compras é gerada para atender a uma receita especifica, assim não fazendo a otimização de produtos presentes em mais de uma receita. Em outros casos, as receitas estão vinculadas a marca. E com isso gerando uma série de problemas, tais como:

* Tempo gasto com compras diárias
* Aumento do gasto com compras diárias
* Comprar alimentos demasiado que encarecem o carrinho
* Desperdícios de alimentos em caso de perecíveis
* Falta de planejamento alimentar saudável

# 2. REQUISITOS DO SISTEMA

## 2.1. Descrição Geral do Sistema

O serviço Lista de Compras Inteligente busca otimizar as compras para o preparo das refeições semanais.

O usuário deverá estar cadastrado e autenticado no sistema para utilizar qualquer funcionalidade.

Assim que o usuário acessar o serviço pelo site ele deve se autenticar na tela de login informando o nome de usuário e senha, caso ele não seja cadastrado haverá a opção abaixo para “cadastrar-se” informando os seguintes dados: nome, e-mail, nome de usuário e senha todos os campos são obrigatórios e a senha do usuário deve ser criptografada. Ao finalizar o cadastro será direcionado para tela inicial do sistema.

Na tela inicial do sistema haverá no menu lateral listada as funcionalidades: Ingredientes Receitas, Cardápio Semanal e Configurações.

Em Ingredientes o usuário poderá cadastrar os ingredientes que serão utilizados em futuras receitas, para isso deve informar apenas o nome do ingrediente, não podendo haver duplicidade de ingredientes. Na funcionalidade de ingredientes o usuário poderá listar todas os ingredientes e editar. Para excluir um ingrediente é necessário antes excluir todas as receitas onde ele foi utilizado.

Em Receita o usuário poderá cadastrar as receitas que deseja realizar para isso deve informar: nome da receita e uma breve descrição do prato em seguida informar os ingredientes utilizados na receita com os dados: ingrediente (previamente cadastrado), quantidade e unidade de medida. As unidades de medidas de um ingrediente são preestabelecidas é podem ser:

* XCR - Xícara
* COL - Colher
* KG - Quilograma
* G – Grama
* MG - Miligrama
* L - Litro
* ML - Mililitro
* M - Metro
* CM - Centímetro
* MM – Milímetro

Caso o usuário não selecione nenhuma das unidades acima o sistema deverá entender que se trata de uma unidade, como por exemplo: 1 ovo.

Toda receita deve possuir ao menos 1 ingrediente e no máximo 10, o mesmo ingrediente não pode ser inserido duas vezes. O mínimo da quantidade do ingrediente é um (1) e no máximo dez mil (10.000).

Após informar todos os ingredientes que serão utilizados a última etapa do cadastro da receita é onde será informado o modo de preparo passando uma descrição do que será feito essa descrição deve possuía no máximo quinhentos (500) caracteres.

Após informar o modo de preparo e confirmar os dados o cadastro da receita estará finalizado. Na funcionalidade de receita o usuário poderá listar todas as receitas e editar ou excluir.

A funcionalidade de Cardápio Semanal é onde o usuário poderá planejar seu cardápio semanal. Na tela haverá uma tabela com os cinco dias das semanas (segunda-feira, terça-feira, quarta-feira, quinta-sexta e sexta-feira) e as quatro refeições (café da manhã, almoço, lanche e jantar) dos dias respectivos. O usuário deve selecionar uma receita para cada uma dessas refeições. É obrigatório preencher as refeições almoço e janta de todos os dias da semana (segunda-feira a sexta-feira). Na funcionalidade de cardápio semanal o usuário poderá listar todas os cardápios e editar ou excluir. Nessa listagem deve aparecer um campo que indique o percentual de refeições preenchidas em cada cardápio.

Também nessa listagem da funcionalidade Cardápio Semanal terá a opção de gerar a lista de compras de cada um dos cardápios listados. O sistema realizará um cálculo que deve totalizar a quantidade (fazendo as devidas conversões das unidades de medidas) dos ingredientes das receitas que se encontram no cardápio semanal, desse modo o usuário comprará apenas a quantidade que necessita para confeccionar todas as receitas nele contidas.

A lista de compras gerada não será gravada em banco, sendo sempre gerada a partir do cardápio selecionado, pois tanto o cardápio semanal, como as receitas podem sofrer alterações pelo usuário do sistema.

Na tela da lista de compras haverá uma tabela listando os ingredientes necessários e sua devida quantidade e unidade de medida e um botão para imprimir a lista e também a data de criação do cardápio semanal. Será possível mudar a ordem dos itens na lista apenas arrastando e soltando e também ao clicar no checkbox ao lado do item ele deverá ser riscado para o usuário identificar que já possui o tal item.

Exemplo de uma lista de compras gerada pelo sistema: Segunda e quinta as receitas selecionadas pelo usuário utilizam o ingrediente leite. Segunda 500ml e quinta 500ml na geração da lista de compras o sistema deverá totalizar e imprimir na tela algo como: “Comprar: 1 litro de leite”.

Por fim na funcionalidade Configurações o usuário poderá atualizar as informações da sua conta criada anteriormente na tela de cadastro do sistema. Podendo alterar informações como: nome, e-mail, nome de usuário e senha. Também poderá escolher entre 3 imagens (masculino, feminino e neutro) para utilizar como foto em seu perfil que aparecerá junto ao menu lateral ao lado de seu nome e no menu superior. Ao clicar em salvar o usuário é redirecionado para a página inicial.

A seguir é apresentado a visão do produto através da técnica elevator pitch (MOORE, Geofrey A., 2014).

|  |  |
| --- | --- |
| **Para** | Publico em geral |
| **Cujo** | necessitam de um serviço para otimizar suas compras no mercado. |
| **A** | Lista de Compras Inteligente |
| **É um** | serviço |
| **Que** | possui as principais funcionalidades para otimizar, agilizar a compras e também incentivar uma alimentação saudável. |
| **Diferentemente do** | da maioria dos serviços oferecidos que não integram todos os ingredientes necessários para a alimentação da semana inteira. |
| **O nosso produto** | é focado no mercado nacional e visa acabar com o desperdício e economizar dinheiro e tempo, sabendo exatamente o que precisará comprar durante a semana. |

**Tabela 1 – Visão do produto**

## 2.2. Partes Interessadas

Nesta seção são apresentadas as partes interessadas do sistema.

|  |  |
| --- | --- |
| **Nome** | **Usuário** |
| **Papel no desenvolvimento** | Contribuindo com feedback no uso do sistema e possíveis melhoramentos. |
| **Insumos ao sistema** | Essa parte interessada será capaz de utilizar as seguintes funcionalidades:   * Criar uma conta * Efetuar o login * Alterar seus dados cadastrais * Cadastrar ingredientes * Cadastrar receitas * Planejar cardápio seminal * Gerar a lista de compras |
| **Representante** | Pessoas que buscam perder peso, uma alimentação mais saudável, ser mais sustentáveis (evitando desperdícios de alimentos), variar as refeições e economizar dinheiro e tempo. |

**Tabela 2 – Partes interessadas**

## 2.3. Atores

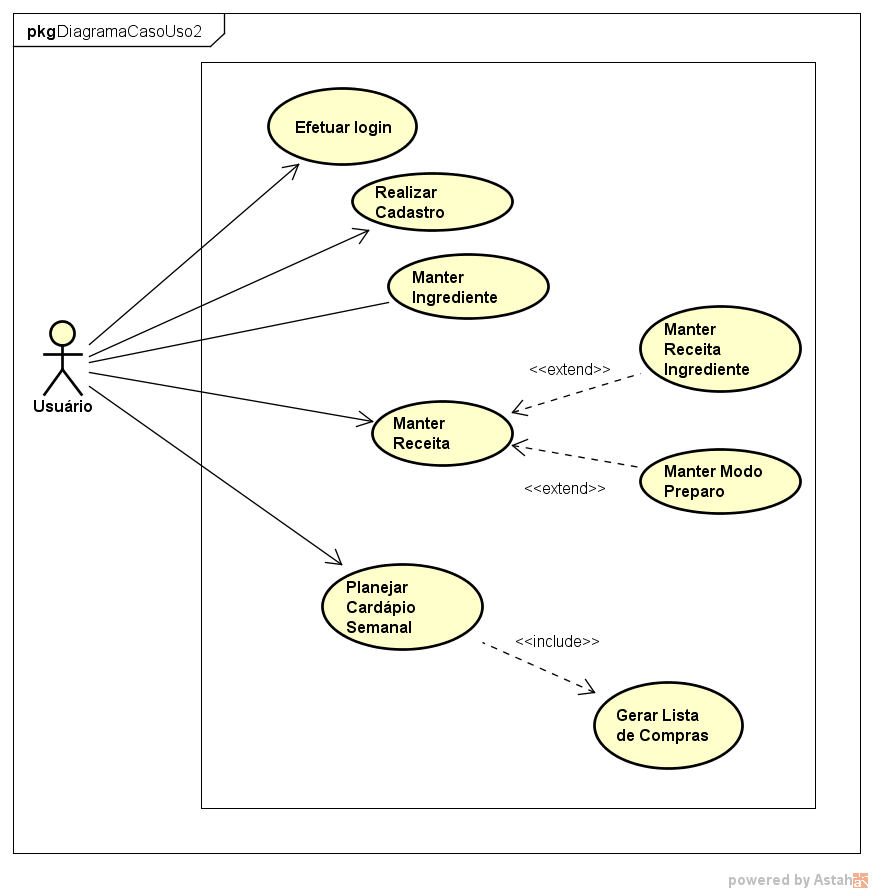
Nesta seção são apresentados os atores do sistema.

|  |  |
| --- | --- |
| **Nome** | **Usuário** |
| **Papel** | Destinando-se a qualquer pessoa individual que, por sua iniciativa procure agilizar suas compras planejando seu cardápio semanal. |
| **Insumos ao sistema** | Esse ator será capaz de utilizar as seguintes funcionalidades:   * Criar uma conta * Efetuar o login * Alterar seus dados cadastrais * Cadastrar ingredientes * Cadastrar receitas * Planejar cardápio semanal * Gerar a lista de compras |
| Representante | Público em geral |

**Tabela 3 – Atores**

## 2.4. Especificação dos Requisitos Funcionais

Nesta seção veremos o diagrama de caso de uso do sistema. O diagrama de caso de uso é uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado (Fowler M, 2005).

******

**Figura 1 – Diagrama de caso de uso**

Abaixo serão apresentadas as descrições dos requisitos funcionais da aplicação:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF01 | **Requisito:** | Realizar cadastro |
| **Descrição:** | Usuário deverá cadastrar os dados para criar uma conta com as seguintes informações: nome, e-mail, nome de usuário e senha. | | |

**Tabela 4 – RF01 – Realizar cadastro**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF02 | **Requisito:** | Efetuar login |
| **Descrição:** | Usuário deverá ser capaz de logar no sistema informando um nome de usuário e senha cadastrados. | | |

**Tabela 5 – RF02– Efetuar login**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF03 | **Requisito:** | Alterar dados cadastrais |
| **Descrição:** | Usuário deverá ser capaz de alterar os dados informados no cadastro como nome, e-mail, nome de usuário e senha, além pode alterar a imagem de perfil. | | |

**Tabela 6 – RF03 – Alterar dados cadastrais**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF04 | **Requisito:** | Manter ingrediente |
| **Descrição:** | Usuário deve poder incluir, alterar e excluir os ingredientes que serão utilizados nas futuras receitas que devem conter apenas o nome. | | |

**Tabela 7 – RF04 – Manter ingrediente**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF05 | **Requisito:** | Manter receita |
| **Descrição:** | Usuário deve poder incluir, alterar e excluir as receitas que devem conter nome, categoria e descrição. | | |

**Tabela 8 – RF05 – Manter receita**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF06 | **Requisito:** | Manter receita ingrediente |
| **Descrição:** | Usuário deve poder incluir, alterar e remover os ingredientes vinculado a receita que devem conter o nome do ingrediente já cadastrado anteriormente, quantidade e unidade de medida. | | |

**Tabela 9 – RF06 – Manter receita ingrediente**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF07 | **Requisito:** | Manter modo de preparo |
| **Descrição:** | Usuário deve poder incluir, alterar e excluir o modo de preparo que deve conter a descrição. | | |

**Tabela 10 – RF07 – Manter modo de preparo**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF08 | **Requisito:** | Planejar cardápio semanal |
| **Descrição:** | Usuário deve poder incluir, alterar e excluir o cardápio que deve conter uma receita para o café da manhã, almoço, lanche e jantar para cada dia da semana (segunda, terça, quarta, quinta e sexta). | | |

**Tabela 11 – RF08 – Planejar cardápio semanal**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RF09 | **Requisito:** | Gerar lista de compras |
| **Descrição:** | Sistema deverá gerar uma lista de compras a partir do cardápio semanal selecionado. | | |

**Tabela 12 – RF09 – Gerar lista de compras**

### **2.4.1. Especificação de Casos de Uso**

Nesta seção serão apresentadas as especificações dos casos de uso do sistema.

|  |  |  |
| --- | --- | --- |
| **Nome** | **Realizar Cadastro** | **Id: CSU01** |
| Sumário: | O sistema deverá ser capaz de permitir que usuário crie uma conta para acessar o sistema. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário ter acesso a internet. | |
| Requisito Funcional: | RF01 | |
| Regra de Negócio: | RN01, RN02 | |
| **Fluxo principal** | | |
| 1. O usuário acessa o site do sistema. 2. O sistema exibe a tela de login. 3. O usuário escolhe a opção “Criar uma conta” 4. O sistema apresenta a interface com os dados a serem preenchidos. 5. O usuário informa os dados e clica em Gravar. 6. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02] [FE03]** 7. O sistema exibe a tela de login. 8. Caso de uso é encerrado. | | |
| **Fluxo de exceção: [FE01] – Validar dados** | | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | | |
| 1. Caso retorne o código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal | | |
| Pós-Condição: | Conta criada | |

**Tabela 13 – Descrição de caso de uso – Realizar cadastro**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Efetuar login** | **Id: CSU02** |
| Sumário: | O sistema deverá ser capaz de permitir que o usuário entre no sistema. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário ter acesso a internet. | |
| Requisito Funcional: | RF02 | |
| Regra de Negócio: | RN02 | |
| **Fluxo principal** | | |
| 1. O usuário acessa o site do sistema. 2. O sistema exibe a tela de login. 3. O usuário escolhe a opção **[FA01] – Efetuar Login** ou **[FA02] – Realizar Cadastro.** 4. O sistema exibe a tela inicial. 5. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Efetuar Login** | | |
| 1. O sistema apresenta a interface com os dados a serem preenchidos. 2. O usuário informa o nome de usuário e senha e clica em “Login”. 3. O sistema valida os dados informados. **[FE01] [FE02] [FE03]** 4. Retornar para o passo 4 do fluxo principal. | | |
| **Fluxo alternativo: [FA02] – Realizar Cadastro** | | |
| 1. Passos apresentados no **[CSU01] – Realizar Cadastro** 2. Retornar para o passo 4 do fluxo principal. | | |
| **Fluxo de exceção: [FE01] – Validar dados** | | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | | |
| 1. Caso retorne o código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | | |
| Pós-Condição: | Dados atualizados | |

**Tabela 14 – Descrição de caso de uso – Efetuar login**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Alterar dados cadastrais** | **Id: CSU03** |
| Sumário: | O sistema deverá ser capaz de permitir o usuário atualizar suas informações cadastrais. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário está logado no sistema. | |
| Requisito Funcional: | RF03 | |
| Regra de Negócio: | RN03 | |
| **Fluxo principal** | | |
| 1. O usuário seleciona na tela inicial a opção de menu “Configurações > Editar Perfil” ou clica em “Editar perfil” localizado abaixo do nome de usuário. 2. O sistema exibe a tela de Editar Perfil. 3. O sistema apresenta a interface com os dados: nome, e-mail, nome de usuário e senha (em formato: \*\*\*\*) preenchidos e três imagens de perfil, sendo a primeira já selecionada. 4. O usuário altera os dados desejados e clica em “Salvar”. **[FE01] [FE02] [FE03]** 5. Caso de uso é encerrado. | | |
| **Fluxo de exceção: [FE01] – Validar dados** | | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | | |
| 1. Caso retorne o código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | | |
| Pós-Condição: | Dados do usuário atualizados | |

**Tabela 15 – Descrição de caso de uso – Alterar dados cadastrais**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Manter ingrediente** | **Id: CSU04** |
| Sumário: | O sistema deverá ser capaz de manter atualizadas as informações cadastrais sobre os ingredientes dos usuários. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário está logado no sistema. | |
| Requisito Funcional: | RF04 | |
| Regra de Negócio: | RN04, RN05 | |
| **Fluxo principal** | | |
| 1. O usuário seleciona na tela inicial a opção de menu “Ingredientes”. 2. O usuário escolhe a opção de menu **[FA01] – Lista** ou **[FA02] – Novo Ingrediente.** 3. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Lista** | | |
| 1. O sistema lista os ingredientes cadastrados. **[FE01]** 2. O usuário escolhe um dos ingredientes listados e utiliza uma das opções: **[FA03] – Editar; [FA04] – Excluir.** 3. Retornar para o passo 2 do fluxo principal. | | |
| **Fluxo alternativo: [FA02] – Novo Ingrediente** | | |
| 1. O sistema apresenta a interface com os dados a serem preenchidos. **[FE03]** 2. O usuário informa os dados e clique em “Gravar”. 3. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02]** 4. Retornar para o **[FA01] – Lista.** | | |
| **Fluxo alternativo: [FA03] – Editar** | | |
| 1. O sistema apresenta a interface com os dados recuperados na base de dados do registro selecionado. **[FE03]** 2. O usuário informa os dados e clique em “Gravar”. 3. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02] d.** Retornar para o **[FA01] – Lista.** | | |
| **Fluxo alternativo: [FA04] – Excluir** | | |
| 1. O sistema exibe uma popup com uma mensagem para confirmar a exclusão. 2. Caso o usuário confirme a exclusão do dado e exibido uma mensagem de sucesso. **[FE03]** 3. Retornar para o **[FA01] – Lista.** | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | Dados dos ingredientes atualizados |

**Tabela 16 – Descrição de caso de uso – Manter ingrediente**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Manter receita** | **Id: CSU05** |
| Sumário: | O sistema deverá ser capaz de manter atualizadas as informações cadastrais sobre as receitas dos usuários. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário está logado no sistema. | |
| Requisito Funcional: | RF04 | |
| Regra de Negócio: | RN06, RN07, RN08, RN09 | |
| **Fluxo principal** | | |
| 1. O usuário seleciona na tela inicial a opção de menu “Receitas”. 2. O usuário escolhe a opção de menu **[FA01] – Lista** ou **[FA02] – Nova Receita.** 3. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Lista** | | |
| 1. O sistema lista as receitas cadastradas. **[FE01]** 2. O usuário escolhe uma das receitas listados e utiliza uma das opções: **[FA03] – Editar; [FA04] – Excluir.** 3. Retornar para o passo 2 do fluxo principal. | | |
| **Fluxo alternativo: [FA02] – Nova Receita** | | |
| 1. O sistema apresenta a interface com os dados a serem preenchidos. **[FE03]** 2. O usuário informa o nome, categoria e descrição e clica em “Continuar”. 3. O sistema redireciona para o **[CSU06] – Manter Receita Ingrediente.** | | |
| **Fluxo alternativo: [FA03] – Editar** | | |
| 1. O sistema apresenta a interface com os dados recuperados na base de dados do registro selecionado. **[FE03]** 2. O usuário informa os dados e clique em “Continuar”. 3. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02]** 4. O sistema redireciona para o **[CSU06] – Manter Receita Ingrediente.** | | |
| **Fluxo alternativo: [FA04] – Excluir** | | |
| 1. O sistema exibe uma popup com uma mensagem para confirmar a exclusão. 2. Caso o usuário confirme a exclusão do dado e exibido uma mensagem de sucesso. **[FE03]** 3. Retornar para o **[FA01] – Lista.** | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | Dados das receitas atualizados |

**Tabela 17 – Descrição de caso de uso – Manter receita**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Manter receita ingrediente** | **Id: CSU06** |
| Sumário: | Extensão do caso de uso "Manter receita" para permitir adicionar os ingredientes e suas medidas na receita. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário iniciar o caso de uso “Manter receita”. | |
| Requisito Funcional: | RF06 | |
| Regra de Negócio: | RN09, RN10, RN11, RN12, RN13 | |
| **Fluxo principal** | | |
| 1. O sistema exibe a tela “Adicione os Ingredientes”. 2. O sistema apresenta a interface com uma linha de campos Ingrediente, Quantidade e Unidade a serem preenchidos. 3. O usuário preenche os campos conforme critérios definidos. **[FE01] [FE03]** 4. O usuário pode escolhe entre as opções apresentadas na tela **[FA01] – Adicionar** ou **[FA02] – Remover.** 5. O usuário clica em “Continuar”. 6. O sistema redireciona para o **[CSU07] – Manter Modo Preparo.** 7. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Adicionar** | | |
| 1. O sistema adiciona uma nova linha com os campos de Ingrediente, Quantidade e Unidade. 2. O usuário preenche os campos conforme critérios definidos. **[FE01] [FE03]** 3. Retornar para o passo 4 do fluxo principal. | | |
| **Fluxo alternativo: [FA02] – Remover** | | |
| 1. O sistema remove a linha com todos os dados nela contidos. **[FE03]** 2. Retornar para o passo 4 do fluxo principal. | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | Dados dos ingredientes contidos na receita atualizados |

**Tabela 18 – Descrição de caso de uso – Manter receita ingrediente**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Manter modo preparo** | **Id: CSU07** |
| Sumário: | Extensão do caso de uso "Manter receita" para permitir adicionar o modo de preparo dos ingredientes selecionados para a receita. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário iniciar o caso de uso “Manter receita”. | |
| Requisito Funcional: | RF07 | |
| Regra de Negócio: | RN14 | |
| **Fluxo principal** | | |
| 1. O sistema exibe a tela “Modo de Preparo”. 2. O sistema apresenta a interface com um campo do tipo textarea “Descrição do Modo de Preparo da Receita”. 3. O usuário preenche o campo conforme critérios definidos. **[FE01] [FE03]** 4. O usuário clica em “Gravar”. 5. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02]** 6. Retornar para o **[CSU06] - Manter receita; Fluxo: [FA01] – Lista**. 7. Caso de uso é encerrado. | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | Dados do modo de preparo da receita atualizados |

**Tabela 19 – Descrição de caso de uso – Manter modo preparo**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Planejar cardápio semanal** | **Id: CSU08** |
| Sumário: | O sistema deverá ser capaz de manter atualizadas as informações cadastrais sobre os cardápios semanais do usuário. | |
| Ator Primário: | Usuário | |
| Ator Secundário: |  | |
| Pré-Condição: | Usuário está logado no sistema. | |
| Requisito Funcional: | RF08 | |
| Regra de Negócio: | RN15, RN16, RN19 | |
| **Fluxo principal** | | |
| 1. O usuário seleciona na tela inicial a opção de menu “Cardápio Semanal”. 2. O usuário escolhe a opção de menu **[FA01] – Lista** ou **[FA02] – Novo Cardápio Semanal.** 3. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Lista** | | |
| 1. O sistema lista os cardápios semanais cadastrados. **[FE01]** 2. O usuário escolhe um dos cardápios semanais listados e utiliza uma das opções: **[FA03] - Gerar lista de compras; [FA04] – Editar; [FA05] – Excluir.** 3. Retornar para o passo 2 do fluxo principal. | | |
| **Fluxo alternativo: [FA02] – Novo cardápio semanal** | | |
| 1. O sistema apresenta a interface com os dados a serem preenchidos. **[FE03]** 2. O usuário informa os dados e clique em “Gravar”. 3. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02]** 4. Retornar para o **[FA01] – Lista.** | | |
| **Fluxo alternativo: [FA03] – Gerar lista de compras** | | |
| 1. O sistema redireciona para o caso de uso **[CSU09] – Gerar lista de compras**. | | |
| **Fluxo alternativo: [FA03] – Editar** | | |
| 1. O sistema apresenta a interface com os dados recuperados na base de dados do registro selecionado. **[FE03]** 2. O usuário informa os dados e clique em “Gravar”. 3. O sistema valida os dados informados e atualiza na base de dados. **[FE01] [FE02]** 4. Retornar para o **[FA01] – Lista**. | | |
| **Fluxo alternativo: [FA04] – Excluir** | | |
| 1. O sistema exibe uma popup com uma mensagem para confirmar a exclusão. 2. Caso o usuário confirme a exclusão do dado e exibido uma mensagem de sucesso. **[FE03]** 3. Retornar para o **[FA01] – Lista**. | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | Dados dos cardápios semanais atualizados |

**Tabela 20 – Descrição de caso de uso – Manter cardápio semanal**

|  |  |  |
| --- | --- | --- |
| **Nome** | **Gerar Lista de Compras** | **Id: CSU09** |
| Sumário: | Include do caso de uso "Planejar cardápio semanal", onde o sistema deverá ser capaz de gerar uma lista de compras com a combinação dos dados do cardápio semanal (receitas e ingredientes). | |
| Ator Primário: | Sistema | |
| Ator Secundário: | Usuário | |
| Pré-Condição: | Usuário iniciar o caso de uso “Planejar cardápio semanal”. | |
| Requisito Funcional: | RF09 | |
| Regra de Negócio: | RN17, RN18 | |
| **Fluxo principal** | | |
| 1. O usuário clica na opção “Gerar lista de compras”. 2. O sistema exibe a tela de lista de compras com a listagem de todos os ingredientes e quantidades necessários para compra. **[FE02][FE03]** 3. O usuário pode escolher a opção: **[FA01] – Imprimir.** 4. Caso de uso é encerrado. | | |
| **Fluxo alternativo: [FA01] – Imprimir** | | |
| 1. O sistema exibe tela impressão nativa do navegador com as opções de “Imprimir” ou “Cancelar”. 2. Retornar para o passo 2 do fluxo principal. | | |

|  |  |
| --- | --- |
| **Fluxo de exceção: [FE01] – Validar dados** | |
| 1. Validar os dados informados, conforme critérios definidos. 2. Caso seja encontrado algum erro, enviar mensagem informando o problema. 3. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE02] – Erro de acesso à base de dados** | |
| 1. Caso retorno código referente à erros no acesso à base de dados, enviar mensagem informando o problema. 2. Retornar ao passo que originou o erro. | |
| **Fluxo de exceção: [FE03] – Erro de regra de negocio** | |
| 1. Caso haja erros nos critérios de validação da regra de negócio, enviar mensagem com o erro. 2. Retornar para o passo 2 do fluxo principal. | |
| Pós-Condição: | PDF da lista de compras contendo os ingredientes e quantidade |

**Tabela 21 – Descrição de caso de uso – Gerar lista de compras**

### **2.4.2. Regras de Negócio**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN01 | **Regra:** | Validações na tela de criar conta de usuário |
| **Descrição:** | Para o usuário criar uma conta deverá ser obrigatório informar os campos: nome, email. nome de usuário e senha e o campo email deve possuir uma validação especial para o tipo e-mail. | | |

**Tabela 22 – RN01 – Validações na tela de criar conta de usuário**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN02 | **Regra:** | Validações na tela de login |
| **Descrição:** | Para o usuário logar no sistema deve ser obrigatório informar os campos nome de usuário e senha. | | |

**Tabela 23 – RN02 – Validações na tela de login**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN03 | **Regra:** | Editar perfil de usuário |
| **Descrição:** | Após o usuário estar logado deverá ser permitido alterar as informações: nome, e-mail, nome de usuário e senha. Também será possível escolher entre três figuras predefinidas para ser exibida como imagem de perfil do usuário. | | |

**Tabela 24 – RN03 – Editar perfil de usuário**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN04 | **Regra:** | Validações na tela de ingrediente |
| **Descrição:** | O campo nome é único e obrigatório no cadastro do ingrediente, sendo assim não podendo haver dois ingredientes com o mesmo nome. | | |

**Tabela 25 – RN04 – Validações na tela de ingrediente**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN05 | **Regra:** | Exclusão de ingrediente |
| **Descrição:** | Não será permitido excluir um ingrediente que já esteja associado há alguma receita, sendo necessário primeiro excluir a(s) receita(s). | | |

**Tabela 26 – RN05 – Exclusão ingrediente**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN06 | **Regra:** | Categoria de receita |
| **Descrição:** | As categorias devem estar previamente listadas em um objeto enum (conjunto fixo de valores) para serem usados nas receitas. | | |

**Tabela 27 – RN06 – Categoria de receita**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN07 | **Regra:** | Validações na tela de receita |
| **Descrição:** | Os campos nome, categoria são obrigatórios e o sistema deve registrar automaticamente o usuário que criou a receita. | | |

**Tabela 28 – RN07 – Validações na tela de receita**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN08 | **Regra:** | Validação do cadastro de receita completo |
| **Descrição:** | Caso o usuário não complete todas as etapas (tela de receita, tela de ingredientes na receita e tela modo de preparo) necessárias para cadastrar uma receita o registro é deletado automaticamente na próxima listagem de receitas. | | |

**Tabela 29 – RN08 – Validação do cadastro de receita completo**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN09 | **Regra:** | Validações na tela de receita ingrediente |
| **Descrição:** | Os campos ingrediente e quantidade são obrigatórios. A quantidade deve ser maior do que zero (0) e menor ou igual a dez mil (10.000). | | |

**Tabela 30 – RN09 – Validações na tela de receita ingrediente**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN10 | **Regra:** | Unidade de Medida |
| **Descrição:** | Caso o usuário não informe a unidade de medida o sistema deverá entender que se trata de uma unidade. Como por exemplo: 1 ovo. | | |

**Tabela 31 – RN10 – Unidade de medida**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN11 | **Requisito:** | Tipos de unidade de medida |
| **Descrição:** | As unidades de medidas de um ingrediente são:   * UNI – Unidade (caso o campo seja vazio) * XCR – Xícara * COL - Colher * KG - Quilograma * G - Grama * MG - Miligrama * L - Litro * ML - Mililitro * M - Metro * CM - Centímetro * MM - Milímetro | | |

**Tabela 32 – RN11 – Tipos de unidade de medida**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN12 | **Regra:** | Quantidade de ingredientes por receita |
| **Descrição:** | Toda receita deve possui ao menos um (1) ingrediente e no máximo dez (10). | | |

**Tabela 33 – RN12 – Quantidade de ingredientes por receita**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN13 | **Requisito:** | Duplicidade de ingredientes na receita |
| **Descrição:** | O mesmo ingrediente não pode ser inserido duas vezes em uma receita. | | |

**Tabela 34 – RN13 – Duplicidade de ingredientes na receita**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN14 | **Requisito:** | Validações na tela de modo de preparo |
| **Descrição:** | O campo descrição é de preenchimento obrigatório e deve possuir no máximo 500 caracteres. | | |

**Tabela 35 – RN14 – Validações na tela de modo de preparo**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN15 | **Requisito:** | Validações na tela cardápio semanal |
| **Descrição:** | Os campos almoço e jantar de todos os dias (segunda a sexta) são obrigatórios. O sistema deverá gravar também data atual no momento da criação do cardápio. | | |

**Tabela 36 – RN15 – Validações na tela cardápio semanal**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN16 | **Requisito:** | Listagem dos cardápios semanais |
| **Descrição:** | Na tela onde será listado os cardápios deverá ser exibido junto uma porcentagem do total de receitas preenchidas, sendo 100% o máximo. Também deverá estar disponível um botão para gerar uma lista de compras. | | |

**Tabela 37 – RN16 – Listagem dos cardápios semanais**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN17 | **Requisito:** | Calculo de conversão de unidade de medida |
| **Descrição:** | O sistema realiza um cálculo que deverá totalizar a quantidade (juntamente com a unidade de medida) dos ingredientes das receitas que se encontram no cardápio, desse modo o usuário não irá correr o risco de não comprar o que precisa nem desperdiçar perecíveis.  Por exemplo: Segunda e quinta as receitas selecionadas pelo usuário utilizam o ingrediente leite. Segunda 500ml e quinta 500ml, na geração da lista de compra o sistema deverá totalizar e imprimir na tela algo como: “Comprar: 1 litro de leite”. | | |

**Tabela 38 – RN17 – Calculo de conversão de unidade de medida**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN18 | **Requisito:** | Lista de compras |
| **Descrição:** | Na tela da lista de compras gerada deverá ser possível mudar a ordem dos itens apenas arrastando e soltando. Existirá um checkbox ao lado dos itens para risca-los, além disso a tela deverá possuir um botão de imprimir. | | |

**Tabela 39 – RN18 – Lista de compras**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RN19 | **Requisito:** | Duplicidade cardápio semanal |
| **Descrição:** | É permitido criar apenas um cardápio por dia (data atual), caso o usuário tente criar um segundo cardápio será apresentada uma tela direcionando para editar o cardápio já criado. | | |

**Tabela 40 – RN19 – Duplicidade cardápio semanal**

## 2.5. Especificação dos Requisitos Não Funcionais

Nesta seção serão apresentados os requisitos não funcionais do sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RNF01 | **Requisito:** | Usabilidade |
| **Descrição:** | Em alguns campos há notas explicativas (hint) identificados por um ícone redondo com uma interrogação no meio ao passar o mouse sobre o ícone são exibidas informações adicionais para auxiliar o usuário no preenchimento do mesmo. | | |

**Tabela 41 – RNF01 – Usabilidade**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RNF02 | **Requisito:** | Tecnologia |
| **Descrição:** | O software deverá possuir interface web e utilizar Java SpringBoot para o back-end e Angular para o front-end com acesso ao banco de dados relacional MySQL. | | |

**Tabela 42 – RNF02 – Tecnologia**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RNF03 | **Requisito:** | Segurança de acesso |
| **Descrição:** | O sistema deverá permitir o acesso apenas aos usuários cadastrados e a senha deve ser criptografada com a hash SHA-1. | | |

**Tabela 43 – RNF03 – Segurança de acesso**

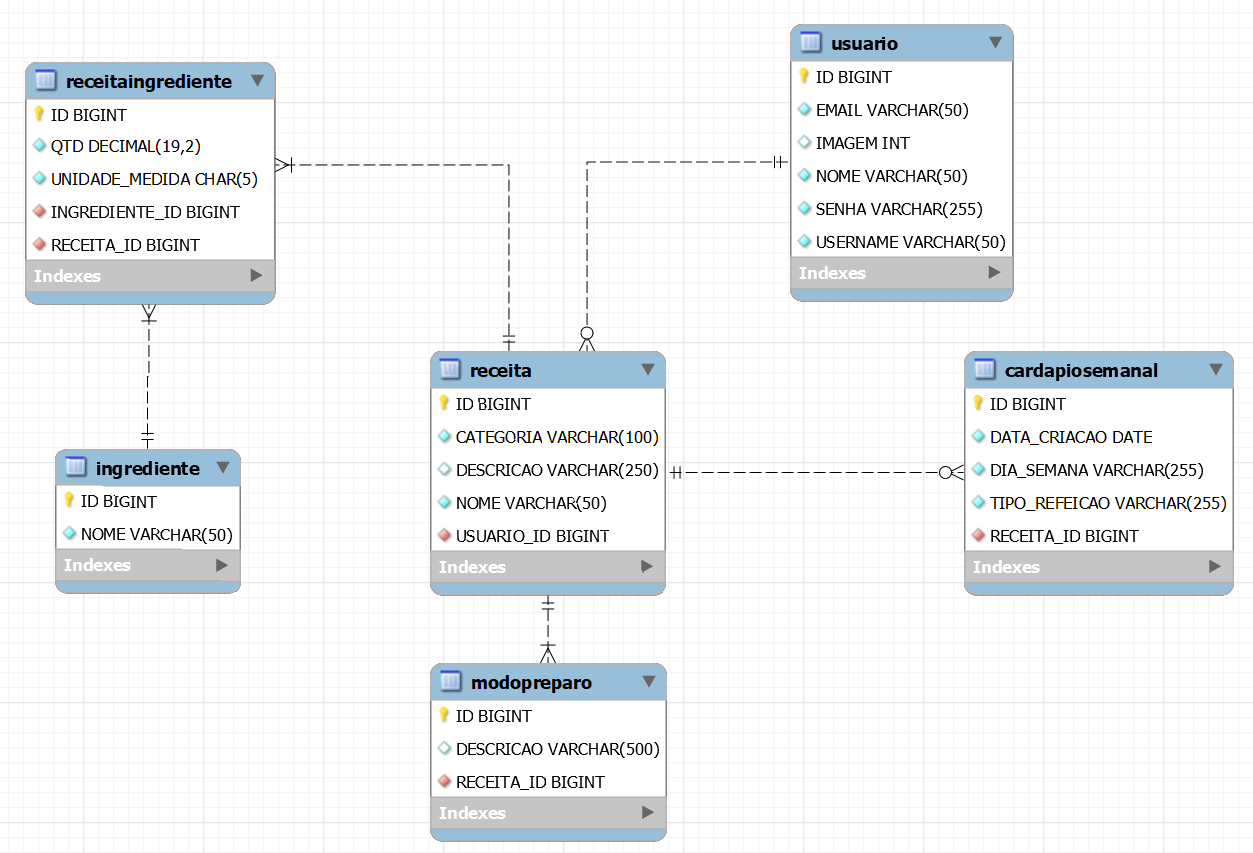
|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | RNF04 | **Requisito:** | Ambiente |
| **Descrição:** | O hardware necessário para o usuário é um computador com acesso à internet (velocidade igual ou superior ao 3G). Este computador deve ter um sistema operacional que suporte os navegadores: Google Chrome, Mozilla Firefox, Internet Explorer ou Microsoft Edge. | | |

**Tabela 44 – RNF04 – Ambiente**

# 3. MODELO DE DADOS

Nesta seção será apresentado o modelo de dados da aplicação Lista de Compras Inteligente. A Tabela 45 apresenta o dicionário de dados referente a este modelo.

Segue abaixo uma representação de Modelo de Relacionamento de Entidade utilizando a anotação Martin ou notação do pé de galinha como é mais conhecida, proposta por James Martin.

****

**Figura 2 – Modelo de dados**

|  |  |  |
| --- | --- | --- |
| **Dicionário de Dados** | | |
| **Entidade** | Usuário | |
| **Atributo** | **Tipo** | **Descrição** |
| Id | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| Email | VARCHAR (50) | E-mail do usuário |
| Imagem | INT | Um valor inteiro que representa umas das 3 imagens de perfil pré-definidas na aplicação. |
| Nome | VARCHAR (50) | Nome do usuário |
| Senha | VARCHAR (225) | Senha do usuário |
| Username | VARCHAR (50) | Nome de usuário utilizado para o login |
| **Entidade** | Ingrediente | |
| **Atributo** | **Tipo** | **Descrição** |
| ID | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| NOME | VARCHAR (50) | Nome do ingrediente |
| **Entidade** | Receita | |
| **Atributo** | **Tipo** | **Descrição** |
| Id | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| Categoria | VARCHAR (100) | Lista de categorias de receitas pré-definidas na aplicação |
| Descricao | VARCHAR (250) | Descrição da receita |
| Nome | VARCHAR (50) | Nome da receita |
| Usuario\_id | BIGINT | Chave estrangeira (foreign key) é o campo que estabelece o relacionamento entre a entidade Usuario. |
| **Entidade** | Receitaingrediente | |
| **Atributo** | **Tipo** | **Descrição** |
| Id | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| Qtd | DECIMAL (19,2) | Quantidade do ingrediente |
| Unidade\_medida | CHAR (5) | Lista fixa de unidades de medida pré-definidas na aplicação |
| Ingredient\_id | BIGINT | Chave estrangeira (foreign key) é o campo que estabelece o relacionamento entre a entidade Ingrediente. |
| Receita\_id | BIGINT | Chave estrangeira (foreign key) é o campo que estabelece o relacionamento entre a entidade Receita. |
| **Entidade** | Modopreparo | |
| **Atributo** | **Tipo** | **Descrição** |
| Id | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| Descricao | VARCHAR (500) | Descrição do passo a passo do modo de preparo da receita |
| Receita\_id | BIGINT | Chave estrangeira (foreign key) é o campo que estabelece o relacionamento entre a entidade Receita. |
| **Entidade** | Cardapiosemanal | |
| **Atributo** | **Tipo** | **Descrição** |
| Id | BIGINT | Chave primaria (primary key) seu valor é auto incrementado a cada novo registro inserido. |
| Data\_cricao | DATE | Data atual da criação do cardápio semanal |
| Dia\_semana | VARCHAR (255) | Lista fixa de dias da semana de segunda a sexta |
| Tipo\_refeicao | VARCHAR (255) | Lista fixa de refeições pré-definidas na aplicação |
| Receita\_id | BIGINT | Chave estrangeira (foreign key) é o campo que estabelece o relacionamento entre a entidade Receita. |

**Tabela 45 – Dicionário de dados**

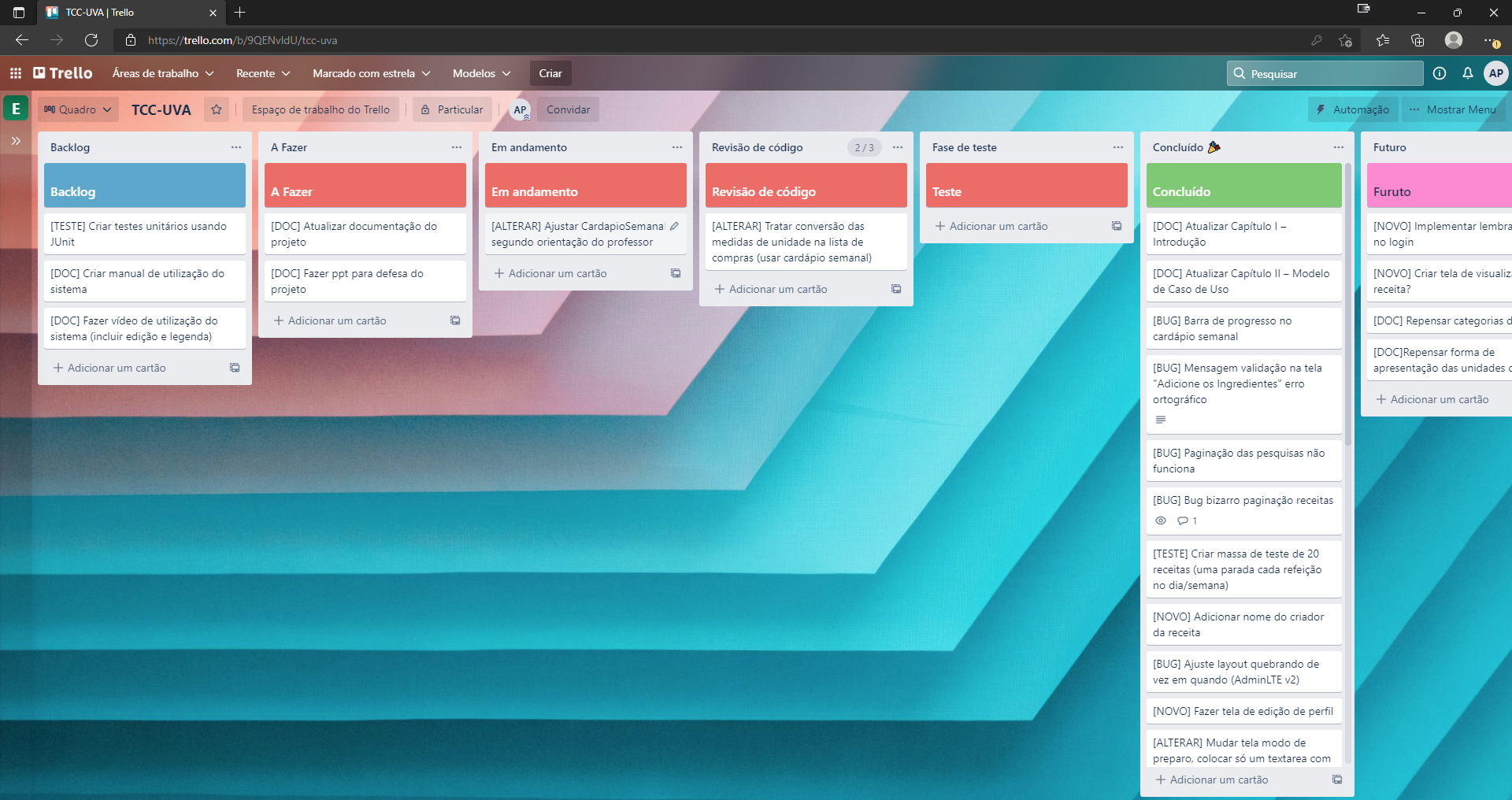
# 4. GESTÃO DE PROJETO

A gestão de projetos é uma união de práticas que serve de guia para trabalhar de forma mais produtiva. Ela inclui métodos e ferramentas que organizam as tarefas, identificam sua sequência de execução e dependências existentes, apoia a alocação de recursos e tempo, além de permitir o rastreamento do andamento das atividades e mensuração do progresso referente ao que foi definido no plano de projeto.

A ferramenta escolhida foi o Trello que é uma ferramenta flexível de gerenciamento de trabalho em que os times podem criar planos, colaborar em projetos, organizar fluxos de trabalho e acompanhar o progresso com visualização, produtividade e gratificação (TRELLO, 2021). A ferramenta é muito utilizada para trabalhos em equipe, mas pode ser usada também por um só individuo, como foi o caso desse trabalho.

No Trello montei um quadro Kanban que se trata de um método visual para gerenciar e conduzir o trabalho, nele é utilizado um quadro (físico ou digital) para planejar e acompanhar as tarefas. Segundo David J. Anderson, “O Kanban rapidamente elimina as questões que prejudicam o desempenho, e desafia uma equipe para se concentrar em resolver essas questões a fim de manter um fluxo constante de trabalho”.

As tarefas foram organizadas por funcionalidades na ordem em que o sistema seria utilizado, ou seja, começando pela funcionalidade de login e terminando na funcionalidade de gerar lista de compras. Como demostrado na imagem abaixo:



**Figura 3 – Quadro Kanban no Trello**

Segue abaixo também um gráfico gerado a partir das atividades colocadas no Trello para melhor visualização do andamento do projeto.

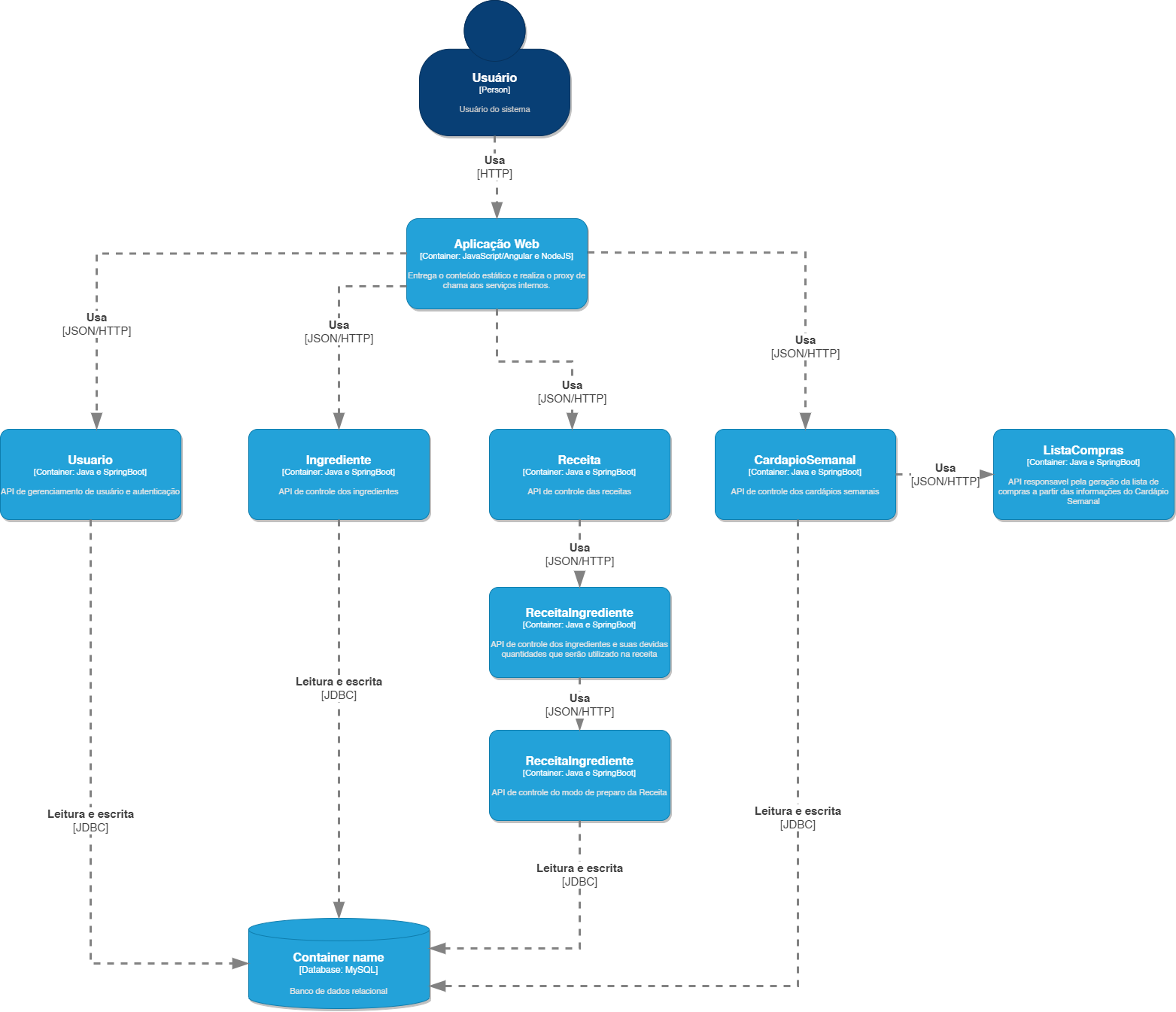
**Tabela 46 – Gráfico das atividades - Trello**

# 5. SOLUÇÃO ARQUITETURAL E TECNOLÓGICA

O modelo de arquitetural usado no sistema é do tipo cliente-servidor. O conceito por trás desse modelo é a comunicação por mensagem transportadas através de uma rede. Cada etapa de comunicação possui um membro rotulado como Cliente, e outro rotulado como Servidor. Se o usuário acessa uma aplicação pela Web, por exemplo, o navegador (mesmo sendo um navegador móvel) do qual ele acessa o sistema seria o processo Cliente, e o servidor no qual a aplicação está rodando seria o processo Servidor (KUROSE; ROSS, 2017).

Para esta aplicação, o processo considerado Cliente será o navegador do usuário, que troca mensagens com o processo Servidor sendo executado no servidor Web (para estudo será utilizado ambiente local). Quando uma página é carregada no lado do Cliente, o navegador as chamadas de cada um dos serviços através da aplicação principal, acionando o controlador de cada um dos serviços. No processo Servidor, que devolve o código requisitado, juntamente com os dados, para ser exibido pelo navegador do Cliente.

Para exemplificar a arquitetura descrita, utilizou-se o modelo C4, no nível 2, o Modelo C4 foi criado pelo consultor independente especializado em arquitetura de software e autor do livro "Arquitetura de Software para Desenvolvedores", Simon Brown. "O modelo C4 é uma maneira simples de comunicar a arquitetura de um software em diferentes níveis de abstração, para que você possa contar histórias diferentes para diferentes públicos. É também uma maneira de introduzir (muitas vezes, reintroduzir) algum rigor e modelagem leve para as equipes de desenvolvimento de software. (INFOQ.COM, 2018).



**Figura 4 – Modelo de Arquitetura**

# 6. PROJETO E EXECUÇÃO DOS TESTES

## 6.1. Método de Versionamento

O versionamento de software é um procedimento de controle de versões que se fundamentam no uso de numerações variadas.

No momento em que os programadores estão desenvolvendo um software, têm de lidar com códigos e realizar modificações ao longo do tempo que produzem diferentes versões dessa aplicação.

O versionamento é exatamente o que possibilita que esses profissionais possuam uma maneira de reconhecer as diferentes versões e conseguir identificar quais alterações foram feitas a cada fase.

Utilizado popularmente para o versionamento de arquivos à ferramenta de controle versão conhecida como Git que é uma excelente solução para o desenvolvimento de sistemas. Há várias ferramentas baseadas em Git para auxiliar no desenvolvimento, porém optei por utilizar o GitHub por estar mais habituado com a ferramenta.

## 6.2. Testes

Teste é o processo de executar um programa com o intuito específico de encontrar erros, antes de sua entrega ao cliente ou usuário final (MYERS, 1979).

### **6.2.1. Testes de Sistemas**

Na preparação dos testes da Lista de Compras Inteligente, foi adotado dois métodos de avaliação e ajuste. Testes ao longo do desenvolvimento que quando se deparava com alguma inconsistência ou erro/bug, o mesmo era adicionado ao Trello para assim ser ajustado e reparado imediatamente é só após isso era testado outra funcionalidade, poupando desta forma um retrabalho e trazendo mais agilidade no desenvolvimento. A fase seguinte de testes funcionais aconteceu na versão final do projeto.

### **6.2.2. Teste de integridade**

Nesses testes foram verificados a confiabilidade dos dados ao longo de seu ciclo de vida que são armazenados no banco de dados. Além disso é averiguado se os dados são precisos e funcionam de acordo com os requisitos. Também é usado para validar se os dados não são modificados ou corrompidos inesperadamente no decorrer do acesso ao banco de dados.

Foram realizados os seguintes testes:

* Verificado se é possível adicionar, excluir ou modificar todos os dados nas tabelas.
* Verificado se um valor em branco ou padrão pode ser recuperado do banco de dados.
* Verificado quando um conjunto de dados é salvo com sucesso no banco de dados (o truncamento não deve acontecer).
* Verificado se o valor padrão é atribuído quando a entrada do usuário não é fornecida.

No futuro será utilizada a framework JUnit para facilitar o desenvolvimento e execução de testes unitários em código Java.

### **6.2.3 Teste de compatibilidade**

Segundo (PRESMAM, 2006), aplicações web precisam operar em ambientes que diferem uns dos outros. Diferentes computadores, dispositivos de exibição, sistemas operacionais e navegadores, têm uma influência significativa na operação.

Portanto, para assegurar que o sistema opera em multiplataforma como proposto, foram feitos alguns testes usando diferentes navegadores (Google Chrome, Mozilla Firefox, Internet Explorer e Microsoft Edge), sistemas operacionais (Windows 10/11 e Ubuntu), computadores e dispositivos móveis (Android).

### **6.2.4 Teste de Funcionalidade**

Uma das maneiras mais práticos de teste de funcionalidade de sistemas é a aplicação de ferramentas para acessar os endpoints. Foi utilizado a ferramenta Postman para comunicação de todos os serviços da API executando os métodos GET, PUT, POST e DELETE.

Também foi realizado um teste de ponta a ponta da utilização do sistema, executando as funcionalidades a começar do cadastrar de usuário até gerar a lista de compras.

# 7. CONCLUSÕES E TRABALHOS FUTUROS

Desenvolver um sistema, independentemente do nível de conhecimento de ferramentas e linguagens, é um compromisso de entrega e responsabilidade com o futuro usuário do sistema. O desenvolvedor se compromete em oferecer a melhor experiência para que seu usuário tenha a melhor experiência de uso possível e comece a utilizar esse sistema para facilitar seu dia a dia. A principal meta ao desenvolver um software, logo, é a experiência que ele irá proporcionar aos seus usuários. E isso deixa a todos os desenvolvedores, arquitetos, designers e outras diversas funções que estejam envolvidas, a importante missão de estar a todo momento aprendendo e buscando inovar cada vez mais, já que o usuário apesar de não saber medir a importância de uma versão atualizada do sistema, ele sabe que o tempo é crucial e vai querer usar um produto que seja rápido, com usabilidade e acessível.

Desenvolver o sistema Lista de Compras Inteligente possibilitou a oportunidade de reconsiderar o desenvolvimento tradicional e transformar o produto final em algo que gostaria de usar diariamente. Com certeza há inúmeras funcionalidades e melhorias a serem desenvolvidas para o sistema futuramente, mas o aprendizado possibilitado tornou a minha experiência em algo valioso e crescente. A Lista de Compras Inteligente poderia ser disponibilizada para uso imediatamente e seria um produto funcional e que agregaria valor ao usuário final.

O resultado aguardado foi obtido, iniciei o projeto do zero e entreguei um produto. Uma das grandes dificuldades encontradas logo de início foi devido a estar realizando o trabalho de forma individual, portanto tive que assumir diversos papéis possíveis em uma equipe de desenvolvimento, ou seja, a função de analista de requisitos, analista de produto, analista de testes, arquiteto de sistema, designer, DBA e desenvolvedor back-end/front-end.

Também tive a oportunidade de trabalhar com o framework Angular que nunca antes havia utilizado, com isso fui capaz de aprender bastante sobre o funcionamento do framework que utiliza o Node.js como interpretador de JavaScript possibilitando assim utilizar JavaScript fora do browser e em conjunto usa a linguagem de programação TypeScript (superset do JavaScript) para criar a interface com o usuário, ou seja, o front-end do sistema web.

Na parte do desenvolvimento back-end pude também relembrar diversos conceitos de programação orientada a objetos ao usar a linguagem de programação Java em conjunto com o framework SpringBoot que reduz bastante o tempo gasto com as configurações iniciais do Spring, além de utilizar a biblioteca Spring Security que ajuda a proteger o sistema web utilizando o protocolo de autorização OAuth2 e também usando o Spring Data JPA para facilitar a comunicação e a persistência de dados com banco de dados, no caso o escolhido para o projeto foi o banco de dados relacional MySQL da Oracle ao qual já possuo mais familiaridade. Enfim fui capaz relembrar e consolidar diversos aprendizados ensinados durante todo período acadêmico na universidade, como gestão de projeto de software, arquitetura, modelagem do banco de dados e diagramas da UML.

A partir da realização deste trabalho pôde-se perceber a oportunidade de trabalhos a serem desenvolvidos futuramente que envolvem o aprimoramento do trabalho atual, como por exemplo:

* Implementar na combobox de ingredientes para adicionar na receita a função de pesquisar e ao ir digitando o nome do ingrediente encurtando a lista da combobox para assim agilizar a procura dos ingredientes para receita.
* Adicionar uma tela de visualização da receita completa contendo além das informações da receita, seus ingredientes e quantidades e seu modo de preparo para melhor experiência do usuário;
* Adicionar novas features referentes a tela de login, como: Recurso de lembrar senha (mantendo o usuário logado mesmo se fechar o navegador) e recuperar senha (uma forma de recuperar a senha utilizando o e-mail de cadastro);
* Desenvolver e implementar regras de negócios de permissão no que diz respeito a edição e exclusão das receitas, permitindo somente ao criador das mesmas executar tais funções. Estando disponível aos outros usuários apenas a visualização da receita.
* Implementar feature de compartilhar o PDF da lista de compras gerada para aplicativos como: WhatsApp, E-mail, Facebook e etc;
* Realizar uma pesquisa de campo com o intuito de melhorar a usabilidade do sistema para todos os públicos;
* Implementar testes unitários usando JUnit para garantir um código mais limpo e confiável;
* Disponibilizar a aplicação para smartphones (Android e iOS).

# REFERÊNCIAS

* Fowler M. **UML Essencial - Um breve guia para a linguagem-pdrão de modelagem de objetos.** 3ª edição. Porto Alegre: Bookman 2005
* MYERS, Glenford J. **The art of software testing.** New York: John Wiley & Sons, 1979.
* PRESSMAN, Roger S. **Engenharia De Software.** 6. ed. [S. l.]: Mcgraw-Hill Interamericana, 2006.
* Anderson, David J. Kanban: **Mudança Evolucionaria de Sucesso Para Seu Negócio de Tecnologia.** Blue Hole Press; 1ª edição, 2011
* KUROSE, J. F.; ROSS, K. W**. Redes de computadores: uma abordagem top-down.** [S.l.]: Pearson, 2017.
* INFOQ.COM. **O modelo C4 de documentação para Arquitetura de Software**, c2018. Disponível em: <<https://www.infoq.com/br/articles/C4-architecture-model/>>. Acesso em: 06 de nov. de 2021.
* TRELLO**. Sobre o Trello: O que está por trás dos quadros.** Disponível em: <<https://trello.com/about>>. Acesso em: 06 nov. 2021.