

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**Elaboração do Documento de Casos de Teste referente ao Módulo de
Ferramentas Nutricionais do Projeto Mais Saúde São João**

Aluno: Alisson Dan de Oliveira Lopes

Prontuário: 1520903

São João da Boa Vista – SP

2018

Resumo

Por falta de atendimento de educação nutricional e de educação física para com a população de São João da Boa Vista, os alunos integrantes do quarto ano de informática, tem como objetivo atender essa necessidade através do projeto Mais Saúde São João, que foi iniciado em fevereiro de 2018, afim de promover um maior contato entre os nutricionistas e educadores físicos para com a população. O documento apresentado nas páginas seguintes exhibe como foi elaborado os casos de teste no módulo de ferramentas nutricionais do projeto Mais Saúde São João. No mesmo, será exibido a sua importância em um desenvolvimento de software, e como foi aplicado no projeto, mostrando as análises dos casos de uso das principais funções do módulo, a elaboração de um caso de teste, e por fim analisar o projeto afim de observar se cada protótipo realmente segue o caso de teste.

Sumário

1	Introdução	6
1.1	Objetivo geral	9
1.2	Objetivo específico	9
2	Desenvolvimento	10
2.1	Levantamento Bibliográfico	10
2.1.1	Caso de Uso	10
2.1.2	Compreensão sobre Casos de Teste	12
2.1.3	Qualidade de Teste	13
2.1.4	Teste de software	14
2.2	Etapas para o desenvolvimento da pesquisa	18
2.2.1	Tabelas do Casos de Uso	18
2.2.2	Tabela de casos de teste	24
2.2.3	Tabela de sucesso/falha	40
2.2.4	Análise dos protótipos feitos	42
3	Conclusões e Recomendações	43
4	Referências Bibliográficas	44

Lista de Figuras

Figura 1 - Subsistemas do projeto Mais Saúde São João[5].	7
Figura 2 - módulos pertencentes ao subsistema Geral com as suas respectivas funções[5]. ...	7
Figura 3 - módulos pertencentes ao subsistema Atividades Físicas com as suas respectivas funções[5].....	8
Figura 4- módulos pertencentes ao subsistema Nutrição com as suas respectivas funções[5].	8
Figura 5 - Exibição do cenário com seus atores e seus respectivos casos de uso.	11
Figura 6 - Exemplificação de funcionamento do ciclo PDCA.	16

Lista de Tabelas

Tabela 1 - Caso de uso Pesquisar Caloria de Alimentos	18
Tabela 2 – Caso de uso Gerenciar Meus Alimentos	20
Tabela 3 – Caso de uso Inserir Meus Alimentos.....	21
Tabela 4 - Caso de uso Excluir meus Alimentos	22
Tabela 5 – Caso de uso Atualizar meus Alimentos.....	23
Tabela 6 – Caso de teste Pesquisar Caloria de Alimento	25
Tabela 7 – Caso de teste Gerenciar Meus Alimentos.....	27
Tabela 8 – Caso de teste Inserir Meus Alimentos	28
Tabela 9 – Caso de teste Excluir Meus Alimentos.....	33
Tabela 10 – Caso de teste Editar Meus Alimentos.....	34
Tabela 11 – Tabela de Sucesso/Falha	40

1 Introdução

O Instituto Federal de Educação, Ciência e Tecnologia é um sistema de ensino que é reconhecido por sua qualidade, e pelo destaque de seus alunos que ingressam nas melhores faculdades do Brasil. O Instituto Federal se tornou muito reconhecido pelo seu método de ensino, que dá a total autonomia para os professores ensinar o que acham mais importante, sem precisar seguir fixamente um cronograma, e é isso que faz essa rede de ensino se propagar por todo o Estado-nação, onde é possível notar a presença de outros câmpus em diversas cidades ao longo do território brasileiro [1].

O Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus São João da Boa Vista contém em sua grade a disciplina de Prática e Desenvolvimento de Sistemas, que basicamente tem como objetivo pôr em prática todas as matérias técnicas que foram ensinadas e aprendidas ao longo dos 4 anos de curso, assim criando um projeto que é apresentado pelo professor no início do ano letivo, e que tem o objetivo de criar programas em benefício da cidade. O docente Breno Lisi Romano apresenta todos os anos, desde a criação dessa disciplina, projetos para os estudantes do Técnico Integrado em informática. E para 2018, o professor responsável, apresentou aos alunos desse mesmo ensino técnico o projeto chamado Mais Saúde São João [2].

Analisando o avanço da tecnologia, é possível observar que a área da saúde, e as demais ligadas a ela carece no meio tecnológico, pois é visível a falta de programas e aplicativos que ajudam a população a ter uma vida saudável, tanto na parte de alimentação e atividades físicas.

Observando que na cidade de São João da Boa Vista - SP, é notável a ausência de programas e aplicativos para fornecer serviços como os de nutrição, para auxiliar ou aproximar a população a ter uma vida saudável, o Instituto Federal de Educação Ciências e Tecnologia de São Paulo – campus São João da Boa Vista, se viu na necessidade de planejar com os alunos do técnico em informática a contribuir na criação deste projeto denominado Mais Saúde São João afim de ajudar a sociedade a ter maior qualidade de vida.

Esse projeto contém nove módulos específicos, com o intuito de ampliar praticas habituais saudáveis no município de São João da Boa Vista, que segundo o IBGE de 2017 (Instituto Brasileiro de Geografia e Estatística) contem cerca de 90.080 (noventa mil e oitenta) habitantes [3]. Esses nove módulos foram divididos em grupos que contém: analistas, DBA's, desenvolvedores e testadores, onde cada equipe é responsável por um módulo, sendo eles: Módulo 01 – Usuários; Módulo 02 – Rede Social; Módulo 03 – Checkups; Módulo 04 – Treinos; Módulo 05 – Resultado dos Treinos; Módulo 06 – Ferramentas Esportivas; Módulo 07 – Plano Alimentas/Cardápio;

Módulo 08 – Diário de Bordo e Módulo 09 – Ferramentas Nutricionais. Abaixo podemos ver as figuras que exibem mais especificadamente o que cada módulo terá por completo, até o dia da entrega e também a divisão de sistemas, denominadas subsistemas [4].

Figura 1 - Subsistemas do projeto Mais Saúde São João[5].

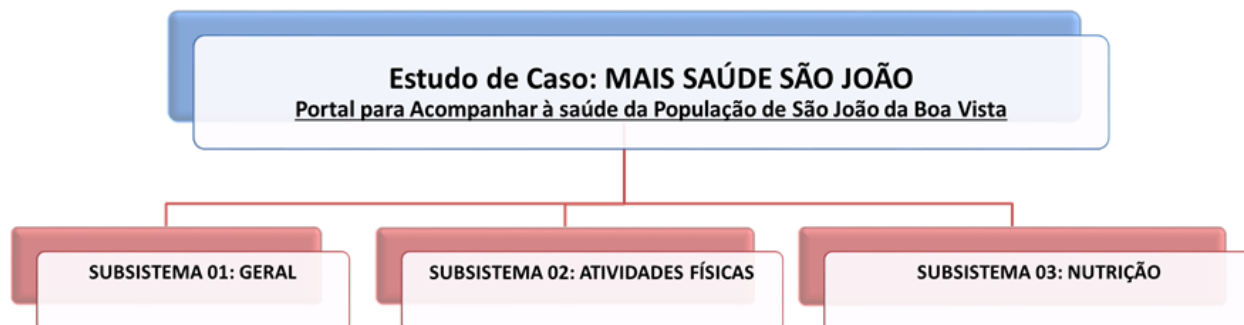


Figura 2 - módulos pertencentes ao subsistema Geral com as suas respectivas funções[5].

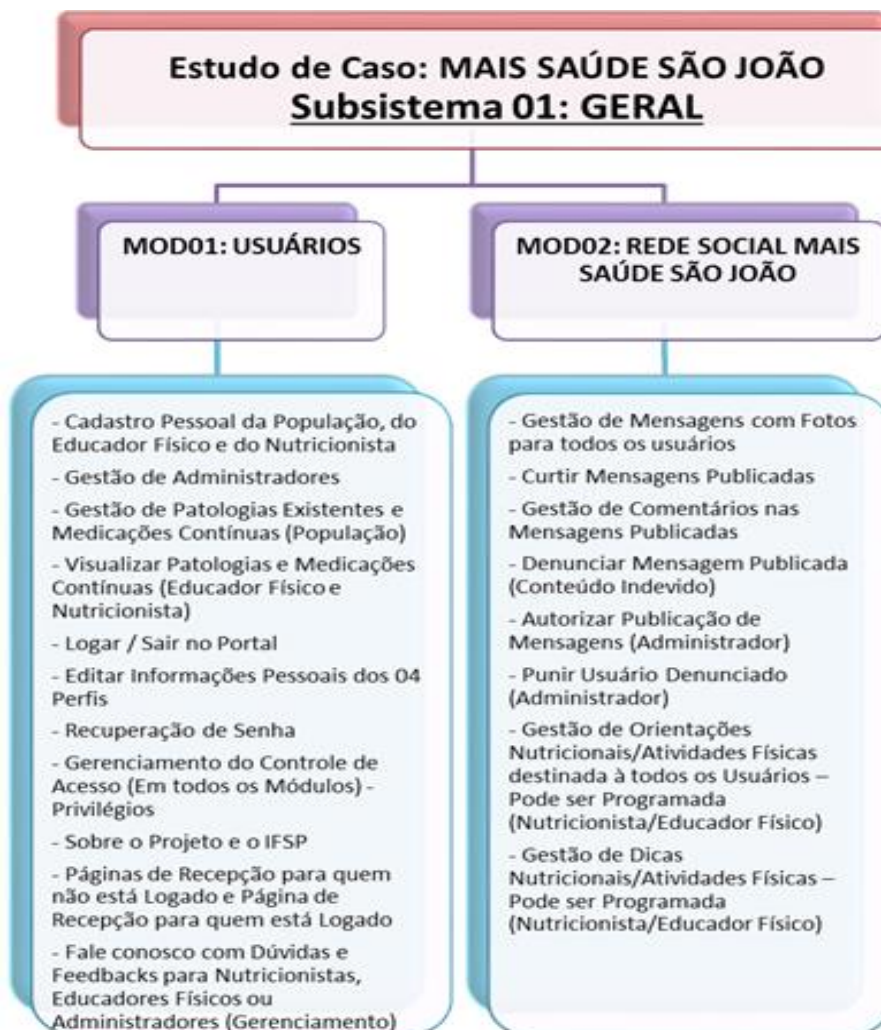
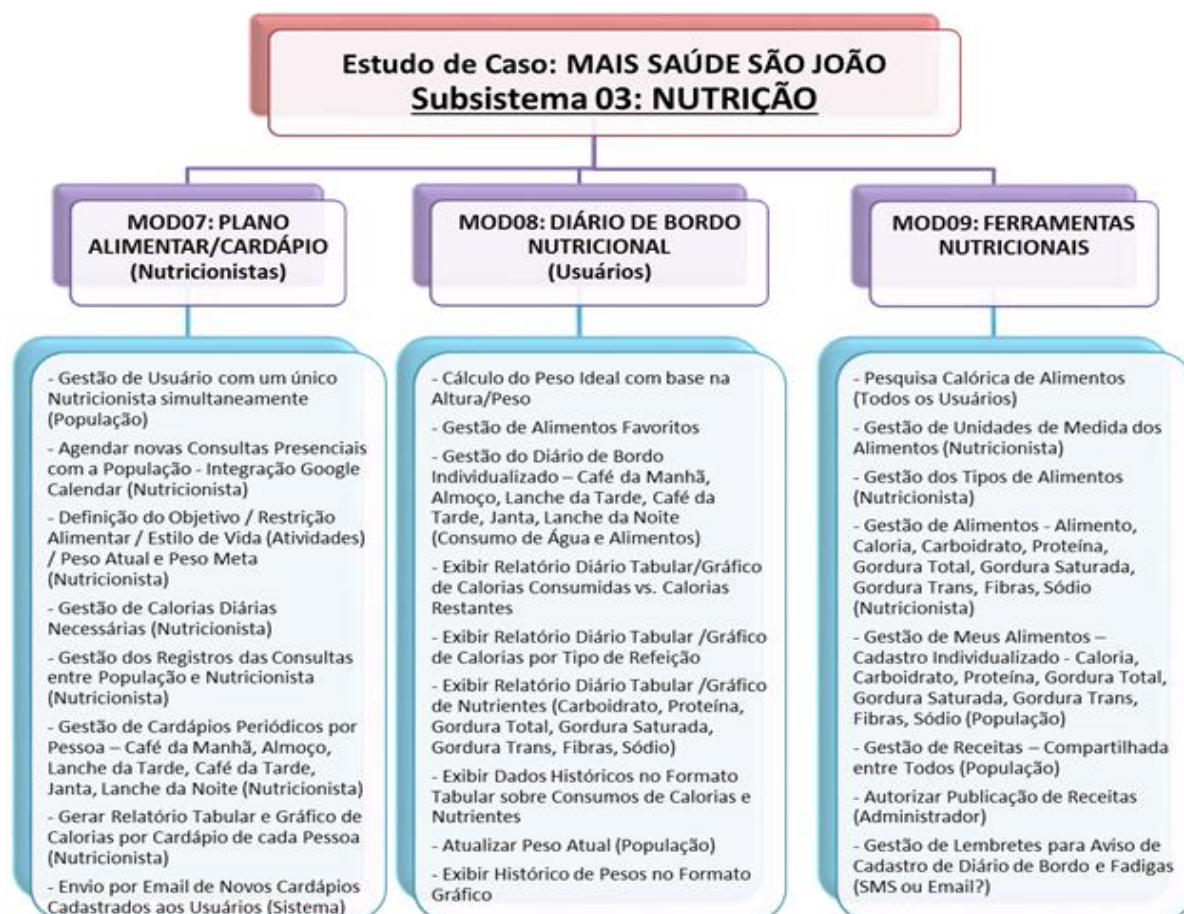


Figura 3 - módulos pertencentes ao subsistema Atividades Físicas com as suas respectivas funções[5].



Figura 4- módulos pertencentes ao subsistema Nutrição com as suas respectivas funções[5].



A figura 1, exibida acima mostra como foi dividido o projeto. Onde pode-se notar as 3 divisões, onde a primeira é a Geral, e nela está integrado os módulos de Usuários e Rede Social que também é exibido na figura 2. Na mesma figura 1, ao lado do primeiro subsistema podemos ver o segundo, que é a Atividades Físicas, e nela está integrado os módulos de Checkups, Treinos, Resultado dos Treinamentos e Ferramentas Esportivas que também é exibido na figura 3. E por fim, como último subsistema da figura 1 temos a Nutrição, e nela está integrado os módulos de Plano Alimentar/Cardápio, Diário de Bordo Nutricional e por último Ferramentas Nutricionais que também é exibido na figura 4.

O projeto Mais Saúde São João é um portal que teve como objetivo também facilitar a comunicação entre os subsistemas que são o geral, atividades físicas e nutrição visando acompanhar a saúde da população de São João da Boa Vista, que foi desenvolvido pelos alunos do quarto ano do curso Técnico Integrado em Informática do Instituto Federal.

1.1 Objetivo geral

Sendo assim, teve-se como objetivo geral a criação e aplicação dos casos de teste no projeto Mais Saúde São João com o intuito de minimizar os erros e deixar mais fácil utilização do projeto pelos usuários de todos os subsistemas.

Casos de teste são componentes muito importantes para o êxito das atividades de teste em um projeto de software, que incluem dados de entrada, dados esperados, fluxos alternativos e condições de execução, criadas para um determinado intuito ou condição de teste. Os testes que definem as entradas que devem ser informadas pelo testador, seja essas entradas feitas manualmente ou com apoio ferramental, e os resultados esperados a partir desta ação. Assim, eles nos permitem fazer análises qualitativas e quantitativas a respeito do software [6].

1.2 Objetivo específico

Os objetivos específicos dessa pesquisa são:

- Analisar os Casos de uso do modulo.
- Elaborar um caso de teste para cada caso de uso.
- Criar fluxos principais e alternativos.
- Analisar o projeto para ver se cada protótipo está realmente seguindo o caso de teste definido para o mesmo.
- Analisar a tabela de sucesso/falha

2 Desenvolvimento

Para atingir os objetivos desta pesquisa, foi feito um levantamento bibliográfico para uma maior compreensão acerca de casos de uso, casos de teste, e as análises qualitativas e quantitativas com base no caso de teste.

2.1 Levantamento Bibliográfico

Para poder alcançar os termos destacados acima, é necessária uma breve compreensão sobre casos de uso e casos de teste. Neste capítulo contém subcapítulos abordando os temas casos de uso e casos de teste.

2.1.1 Caso de Uso

UML cuja sigla tem o significado de *Unified Modeling Language*. Pelo que é definido em seu nome, é visto que a linguagem UML define os artefatos que nos auxilia na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos. A UML possui um total de nove diagramas diferentes entre si que são utilizados para a documentação e modelação de diversos aspectos dos sistemas. Grande parte dos problemas que são encontrados em sistemas orientados a objetos origina-se na construção do modelo, no desenho do sistema. Algumas vezes os profissionais e as empresas acabam não dando muita ênfase à essa parte do projeto, e acabam cometendo diversos erros de modelagem e análise [7].

Neste documento, para o desenvolvimento completo dos casos de teste do Módulo 9, houve a necessidade de utilizar, dentre os nove tipos de diagramas contidos no UML, o mais utilizável quando se trata de sistemas orientados a objetos, que é o Caso de Uso. Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema [7].

Diagramas de Casos de Uso são compostos basicamente por quatro partes [7]:

- Cenário: Eventos que acontecem quando um usuário interage com o sistema.
- Ator: Usuário do sistema, ou melhor, um tipo de usuário.
- Caso de Uso: É uma tarefa ou uma funcionalidade realizada pelo usuário.
- Comunicação: é o que liga um ator com um caso de uso.

Na criação do cenário do módulo 9 obtivemos os seguintes atores

- População: Pessoas com interesse em mudar sua alimentação e melhorar sua saúde
- Nutricionistas: Usuários que irão ajudar a população para melhorar sua saúde

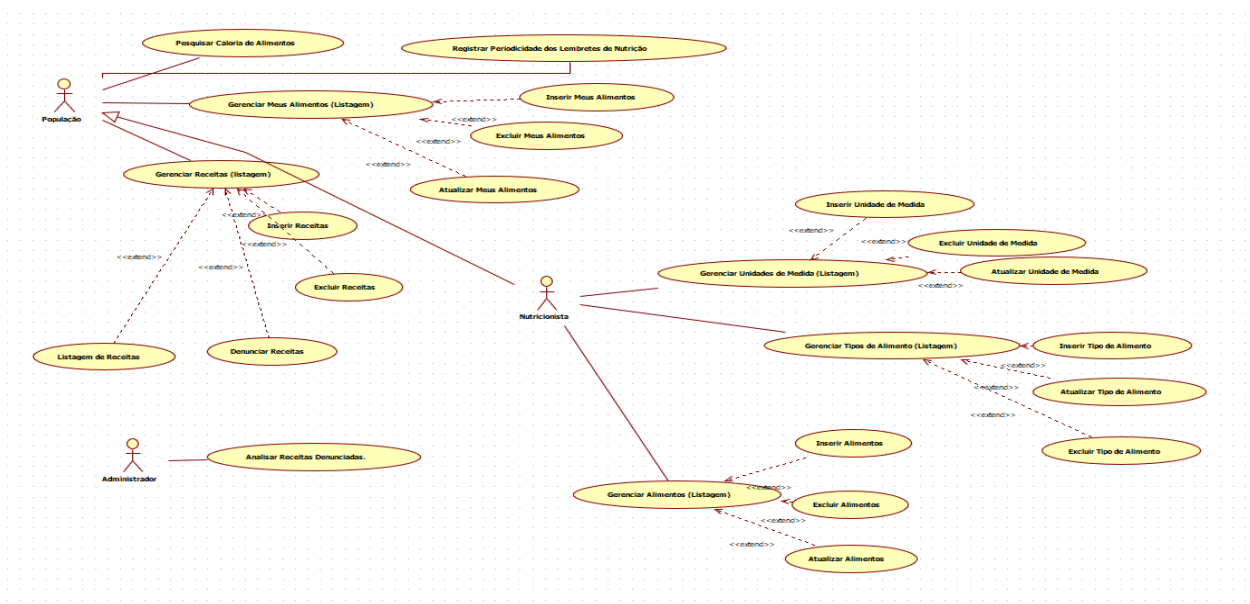
- Administrador: Pessoas que irão gerenciar o site Mais Saúde São João.

E cada um desses atores terá os seus casos de uso, que basicamente é a tarefa que o ator deve executar, e na montagem do cenário, esses casos de uso se conectam com os seus atores utilizando setas, para que fique mais fácil a visualização do que cada ator deve fazer. Portanto, para os seguintes atores, ficaram os seguintes casos de uso:

- População: Pesquisar Calorias de Alimentos; Registrar Periodicidade dos Lembretes de Nutrição; Visualizar seus alimentos cadastrados; Excluir seus alimentos cadastrados; Editar seus alimentos cadastrados; Cadastrar seus alimentos; Visualizar suas receitas cadastradas; Excluir suas receitas cadastradas; Editar suas receitas cadastradas; Cadastrar uma nova receita; Denunciar receitas impróprias. Além disso, a População poderá visualizar os alimentos cadastrados pelos nutricionistas, sem poder executar qualquer tipo de ação, ou seja, apenas visualizar.
- Nutricionista: Visualizar as unidades de medidas cadastradas; Excluir as unidades de medidas cadastradas; Editar as unidades de medidas cadastradas; Cadastrar uma nova unidade de medida; Visualizar os tipos de Alimentos; Excluir os tipos de alimentos cadastrados; Editar os tipos de alimentos cadastrados; Cadastrar um novo tipo de alimento; Visualizar os alimentos cadastrados; Excluir os alimentos cadastrados; Editar os alimentos cadastrados; Cadastrar um novo alimento.
- Administrador: Analisar as receitas denunciadas.

Utilizando o Caso de Uso, com os atores e cenários descritos acima, obtivemos a seguinte figura:

Figura 5 - Exibição do cenário com seus atores e seus respectivos casos de uso.



A partir da figura acima, foi possível gerar um documento, com todas as descrições e detalhes de cada uma das ações, contendo quando deve ser executada, o que deve conter em cada uma dessas ações para que sejam todas realizadas com êxito, os campos para preenchimento dos atores, tudo com o maior detalhe e a maior clareza possível, e também protótipos, exibidos com detalhe, de como deve ser apresentado, onde contém quais estilizações a serem utilizadas, e tudo isso foi feito e verificado para que houvesse a solução de possíveis problemas como falta de clareza ou falta de especificação.

E com esse documento gerado, o caso de teste é de suma importância para que tudo seja feito dentro do planejado, pois a partir desse grande detalhamento feito no documento de casos de uso, acontece os testes para ver se tudo que foi feito na parte de prototipagem está seguindo o documento, para que seja possível corrigir as falhas que surgiram por não seguir objetivamente a descrição feita no caso de uso.

2.1.2 Compreensão sobre Casos de Teste

Em meados da década de 70 a aplicação do teste nos softwares era feita pelos desenvolvedores dos sistemas, essa atividade era vista como algo secundário, com muita pouca importância, era feita somente se o prazo da entrega e custo do produto permitisse tal ação. Ao passar do tempo, devido à grande concorrência existente no mercado e ao crescimento da complexidade dos sistemas, o grau de exigência por qualidade teve um crescimento e com isso a necessidade de testes mais eficazes. O teste de software é uma das fases mais caras do processo de desenvolvimento de software, pois muitas vezes acaba por envolver uma quantidade alta dos recursos de um projeto. O trabalho e o preço associado a esta fase dependem principalmente da parte crítica da aplicação a ser desenvolvida. Categorias de aplicações distintas acabam por requerer uma preocupação diferente com as atividades de teste. Aplicar testes não consiste somente na geração e execução de casos de teste, mas engloba também questões de gerenciamento, planejamento e análise de resultados [8].

Pode ser definido teste algo como uma atividade que tem como objetivo analisar se o software produzido está de acordo com sua documentação e se satisfaz as expectativas do cliente. A atividade de teste é um elemento consideravelmente crítico para garantia da qualidade. Um erro na atividade de teste de software está associado a como verificar se o programa foi testado suficientemente e se pode ser liberado para os usuários com uma confiança razoável de que funcionará de modo aceitável [9]. A “fundura” do teste é tem ligação direta ao número de casos de teste. O acréscimo do número de casos de teste acaba gerando uma maior confiança na qualidade do produto e na etapa de teste, já que um caso de teste reflete um cenário, um fluxo ou a uma condição

diferente através do produto. A avaliação principal da abrangência do teste é a que faz a cobertura baseada em requisitos, de acordo com a quantidade de casos de teste identificados, executados e/ou implementados [10].

A parte de identificar os casos de teste pode ser demorada e, às vezes, é necessário que o teste seja repetido. É por isso que necessariamente deve ser documentado. Um caso de teste ao ser preparado, é necessário a descrição de qual requisito é testado, e um esclarecimento de como o sistema deve ser testado e os princípios de teste, como: versão do aplicativo que será testado, arquivos de dados, sistema operacional, acesso de segurança, data física ou lógica, hora do dia, requisitos como outros testes e outras referências de configuração interligadas à requisitos que são testados e também as entradas e saídas de testes. A aplicação dos casos de uso como modelo para os testes vai permitir que o testador tenha uma visão mais aprofundada do software e ter uma ideia mais clara da necessidade dos clientes. Também permite que, se o caso de uso estiver detalhado adequadamente e consistente, a garantia de ter uma noção completa dos entregáveis desenvolvidos. A parte de sistematizar e padronizar do processo de teste acaba gerando uma confiança mais adequada por parte do cliente e decresce a sinuosidade de treinamento ou adaptação de novos testadores [11].

2.1.3 Qualidade de Teste

A palavra Qualidade é usado em diversas ocasiões, mas o seu significado não é sempre de definição clara e objetiva. Há muitas definições para o termo qualidade, do ponto de vista de pessoas diferentes, ou seja, o termo é geralmente utilizado para expressar a excelência de um produto ou serviço. Quando é acompanhado a atual realidade dos textos ligados à informática em geral, não é incomum se deparar com termos do tipo: “Controle da Qualidade”, “Garantia da Qualidade” e “Gerência da Qualidade” e isso se dá ao fato de que hoje são considerados conceitos quase que simples, seja nas empresas, comércios ou serviços. Esses conceitos são usados em muitas áreas, inclusive mais forte em “Qualidade de Software”. Gerência da qualidade é um processo de controle e gerencia do processo de qualidade na criação ou suporte de algum produto ou serviço. Garantia da qualidade são as ações tomadas para que os defeitos sejam diminuídos. Controle da qualidade são as atividades relacionadas ao ato de medir a qualidade, para diagnosticar se o resultado está sendo alcançado [12].

Em empresas, o termo qualidade é algo examinado há tempos, diversas metodologias foram criadas, sempre focado em melhorar a qualidade ao mesmo tempo em que diminui o custo, pois o custo do defeito pode ser alto demais, não só para as finanças, mas para o nome da empresa também, há diversos casos de defeitos que poderia ser mencionado, mas um conhecido e desastroso é o incidente que ocorreu em junho de 1985 a janeiro de 1987, o Therac-25, dispositivo

computadorizado para tratar através de radiação para câncer, teve alguns equipamentos ministrando doses mais altas a pacientes, um total de 6 pacientes tiveram doses elevadas e alguns acabaram morrendo e outros foram incapacitados. Com o acontecido acima vindo a importância da documentação de casos de teste, ou seja, realizar uma boa documentação, com um teste embasado em encontrar erros é primordial para se obter um software de qualidade. Há três dimensões de qualidade que precisam ser consideradas. A primeira é a confiança, onde o sistema é resistente a falhas durante a execução, isto é, não entra em loop, não interrompe a execução por falta de recursos. A segunda é a funcionalidade, onde o sistema se comporta conforme o esperado e definido em seus requisitos. E por último a performance, onde o sistema tem um tempo de resposta adequado e aceitável, mesmo quando é forçado a processar um volume de dados próximo de situações reais ou de pico [13].

Quando é seguido uma metodologia para desenvolver um produto de software é esperado um produto final que agrade tanto aos clientes quanto a própria empresa. Observando então, não faz sentido começar a construção de um software sem ter uma metodologia de trabalho bem firme e que seja do parecer de todos os envolvidos no processo de desenvolvimento. Pois segundo Bartié, por volta de 30% dos projetos acabam cancelados antes de serem concluídos e algo por volta de 70% dos projetos acabam falhando nas entregas de funcionalidade. No entanto, além de uma grande demanda por softwares de qualidade, as indústrias de desenvolvimento de software acabam sofrendo cada vez mais pressão por parte dos clientes para que o software seja concluído num período de tempo pequeno. Este fato pode fazer com que uma firme metodologia de trabalho acabe por oscilar e prejudicar de alguma forma. Independente da metodologia de trabalho que é utilizada para o desenvolvimento de um software, para que se alcance um produto final com um certo nível de qualidade é de extrema importância a melhoria dos processos de engenharia de software [12].

Contudo, é notório a importância de se criar um teste de qualidade para que não se torne um problema progressivo que acabe afetando o nome da empresa, ou vidas, dependendo da funcionalidade. Pois isso resultaria em um gasto que poderia ter sido evitado na fase de casos de teste, e custaria muito menos para realizar os testes, tanto utilizando ferramentas quanto manualmente.

2.1.4 Teste de software

Teste de Software faz parte do desenvolvimento de software, e representa a etapa que é extrema importância na fase de desenvolvimento de software, pois propõe-se validar se a aplicação está funcionando de acordo e se atinge o que foi especificado e requisitado. Sendo assim, existem diversas práticas que podem ser utilizadas em diversos momentos e de diversas formas para analisar

os pontos mais importantes do software. Usualmente é a última etapa no desenvolvimento de um software, aplicativo ou jogo a ser lançado, que pode ser um novo produto ou uma nova versão. Por mais que algumas empresas e equipes ainda construam softwares sem a interação de um testador ou analista de testes, essa atuação tende a ser cada vez mais difícil de se ver, pois ter alguém focado na qualidade do software é de extrema necessidade e pode comprometer o software e a imagem da empresa.

Analistas de testes, técnicos de testes, ou simplesmente testadores estão acostumados a realizar uma quantidade significativa de testes de diferentes propósitos, envolvendo não só os testes funcionais da aplicação, mas também muitas outras atividades como: Avaliação da especificação de requisitos, Avaliação de projeto técnico, Verificações em outros documentos, Testes de performance e capacidade, Avaliação de interface, etc [14].

O maior problema das empresas é desenvolver softwares com qualidade, em um pequeno espaço de tempo, com custo razoável e atender a demanda do cliente com o produto construído, ou seja, atender aos requisitos prescritos pelo mesmo. Efetivar testes dentro de um processo com metodologia própria é o grande problema da questão. O teste tem a finalidade acrescentar qualidade ao software podendo também fazer uma análise qualitativa em relação aos defeitos descobertos, pois, caso sejam achados poucos defeitos, o software será mais transparente, com os testes é capaz também antecipar a descoberta de erros e incompatibilidades, reduzindo assim o custo do produto.

Para obter um software de qualidade, o ciclo PDCA que significa *Plan, Do, Check, Act* (Planejar, Fazer, Verificar e Agir) é muito útil, pois esse método tem a função de garantir que a empresa organize seus processos, não importando a sua natureza. E como esse ciclo serve para organizar processos de qualquer natureza, ele pode ser muito bem utilizado em um desenvolvimento de software para conseguir garantir uma qualidade desejável e que satisfaça as necessidades do cliente. E no desenvolvimento, antes de que tudo seja entregue, tem a parte de teste, onde tudo o que foi feito no desenvolvimento está realmente funcionando e sem possíveis falhas, que atrapalhem na usabilidade do software. Então o teste de software é de extrema importância para a verificação dos desenvolvimentos, onde se ocorrer alguma falha, ele volta na fase de desenvolvimento para que o erro seja solucionado, e é por isso que o ciclo é importante, pois algumas vezes os erros podem ser por conta de uma má documentação na parte de planejamento, e que foi seguido o que estava descrito, porém quando foi executado a parte de teste apresentou falhas que precisavam ser solucionadas, e nesse ciclo, se for algum problema de documentação, é feita toda uma revisão para poder encontrar a falha, até chegar na fase de testes novamente. Abaixo podemos ver a figura do ciclo PDCA.

Figura 6 - Exemplificação de funcionamento do ciclo PDCA.



Na fase de planejamento é onde acaba sendo definido tudo que é importante para que se tenha uma documentação adequada, e assim, podendo conseguir um software bom que atinja as expectativas do cliente.

Depois da parte de planejamento, entra-se na fase em que o plano será executado, ou seja, é onde tudo que foi definido na fase de planejamento é colocado em prática, onde começa a criação do software, com todas as especificações, com tudo que deve conter e como deve funcionar, entre outros fatores importantes.

Então, depois que o software é criado, os processos são analisados através de ferramentas próprias, para verificar se cada processo cumpre aquilo que foi proposto no planejamento. É nessa fase que poderão ser encontrados erros ou falhas no processo. Ou seja, é nesta parte que entra o teste de software, portanto, se for feito uma documentação bem feita também para realizar os testes de software, essa parte será de extrema importância para que consiga atingir todos os parâmetros exigidos pelos clientes, e para isso, como já foi dito, é necessário uma boa documentação a respeito da parte de testes desse software.

Conforme o resultado na fase Check, será observada as falhas e se os objetivos foram atingidos, caso contrário, o ciclo deve ser reiniciado para que haja uma melhoria. E é onde o ciclo se torna a repetir até que todos os erros sejam solucionados, e quando forem solucionados, o ciclo é terminado nessa última parte [15].

Como o teste de software é de extrema importância para que um bom programa seja construído, ele como qualquer outro processo deve ser revisto continuamente, de forma a ampliar sua atuação e possibilitar aos profissionais uma maior visibilidade e organização dos seus trabalhos, o que resulta numa maior agilidade e controle operacional dos projetos de testes. Além disso, esse processo de teste representa uma estruturação de etapas, atividades, artefatos, papéis e

responsabilidades que buscam a padronização dos trabalhos e ampliar a organização e controle dos projetos de testes.

No processo de identificação de problema nos testes de software, eles podem ser definidos como erros, defeitos ou falhas. Vendo essas três palavras citadas atrás, é comum pensar que são iguais, porém entre elas há uma diferença que acaba diferindo de uma para a outra e por isso não são consideradas iguais. Qualificar os problemas de teste encontrados com uma das três opções anteriores é de extrema importância, pois as vezes o próprio desenvolvedor não consegue resolver porque pode não ter sido causado por ele, e então o problema pode ser da ferramenta, que é um problema diferente do problema causado por um desenvolvedor.

Na Engenharia de Software, o processo de testes de software tem uma nomenclatura muito confusa: erro, defeito e falha podem até parecer a mesma coisa, mas neste contexto possuem diferentes significados. O *International Software Testing Qualifications Board* (Conselho Internacional de Qualificações para Testes de Software), diz que a aplicação correta destes termos são [16]:

- Erro: é fruto da ação humana, que produz um resultado incorreto, como uma falha na escrita de um código.
- Defeito: É qualquer imperfeição ou inconsistência no produto do software ou em seu processo, também conhecido como bug, é o resultado de um erro no código, gerando uma anomalia no funcionamento no sistema.
- Falha: Uma falha é resultado da execução de um defeito no código, por exemplo, houve um retorno de um valor não esperado, como null, isso é um erro, e por causa desse null ocasionou uma falha no sistema.

De certa forma, podemos dizer que toda falha depende de um defeito, mas nem todo o defeito gera uma falha. Da mesma forma, todo defeito é fruto de um erro, entretanto não são todos os erros que resultam em defeito. Toda falha tem origem num erro [16].

Portanto, podemos notar com tudo o que foi destacado nos tópicos anteriores, desde a criação até a sua grande importância na criação de um software, e que é notório que muitas empresas não dão a devida atenção a esta parte da criação, e também o que pode acontecer caso algo não tenha sido detalhado conforme o esperado, ou tenha havido uma interpretação errada por conta de analistas, e isso pode acabar acarretando problemas inimagináveis que poderiam ter sido solucionados anteriormente utilizando ferramentas, como o ciclo PDCA que cria um ciclo de verificação para ver se tudo está feito conforme o documento de testes, afim de encontrar erros e solucioná-los da melhor maneira possível.

2.2 Etapas para o desenvolvimento da pesquisa

Para um melhor aproveitamento do capítulo 2.1, este tem como objetivo exibir na prática todos os detalhes a respeito de casos de teste e, de como o módulo de ferramentas nutricionais fez uso do mesmo, explicando o estilo de documentação, tanto de casos de uso como casos de teste, e estão descritos nos subcapítulos abaixo.

2.2.1 Tabelas do Casos de Uso

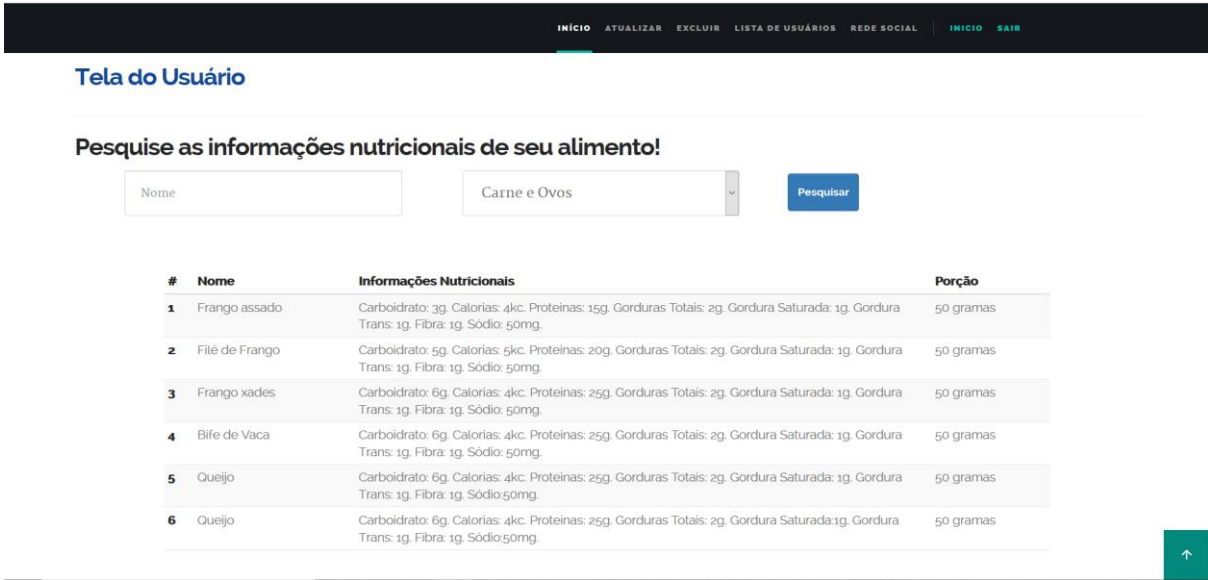
Para que o caso de teste fosse criado, necessitou-se de um outro documento que é o de casos de uso. Esse documento, teve como objetivo documentar todos os possíveis fluxos de uma página, ou seja, documentar quem estaria executando a ação, quando essa ação deveria ser executada, qual condição necessária antes de executar a ação, quais campos deveriam aparecer, quais os nomes dos campos, quais eram obrigatórios, nomes dos botões, e também a ação do sistema, que inclui o que deveria acontecer após a execução dessa ação pelo usuário, como o que aparecer caso tudo estivesse preenchido corretamente e também o que aparecer caso nada estivesse preenchido corretamente, e com fluxos alternativos, como eventuais situações.

Tivemos um total de 23 casos de uso, ou seja, um total de 23 tabelas descrevendo as ações que deveriam ser tomadas pelo usuário, e a resposta retornada pelo sistema, porém, não são necessárias todas as tabelas já que todas seguem um padrão, então foi feita uma seleção das mais importantes.

A Tabela 1 apresenta a documentação do caso de uso Pesquisar Caloria de Alimentos.

Tabela 1 - Caso de uso Pesquisar Caloria de Alimentos

Nome do Caso de Uso: Pesquisar Caloria de Alimentos	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre que o usuário desejar pesquisar as calorias dos alimentos.
Ator Principal:	População.
Pré-Condição:	Acessar o site e logar como usuário normal.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Poderá pesquisar a caloria de seus alimentos por duas vias, alimento e categoria.	2. O sistema deverá direcionar e exibir o nome, porção do alimento e os seguintes valores nutricionais: carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio.

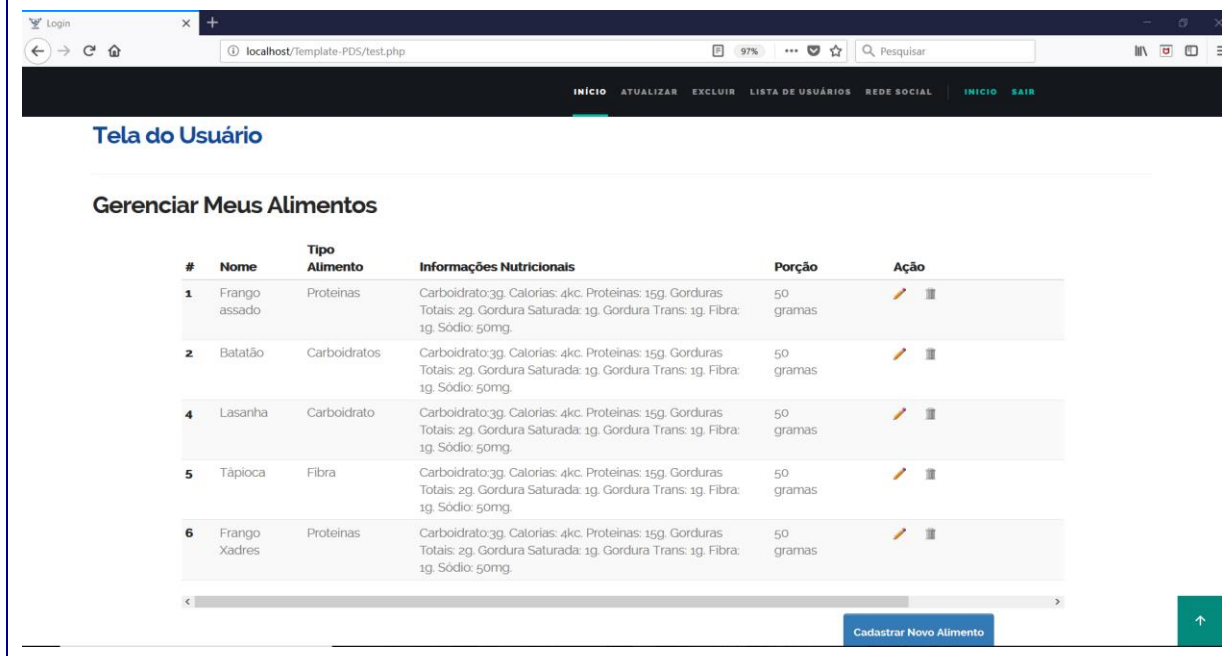
3.Fim do Caso de Uso	
Fluxo Alternativo A: Pesquisa utilizando filtro simples (Nome)	
2. Caso o usuário deseje apenas consultar o valor nutricional do alimento específico, o mesmo poderá pesquisar utilizando a caixa de texto dos alimentos, utilizando apenas o nome do alimento.	3. Atualizar tabela e exibir nome do alimento, porção e os seguintes valores nutricionais: Carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio. Através de uma lista dos alimentos existentes no banco, sendo específico dos alimentos que possuem o dado nome, exibindo todos os que o possuem.
Fluxo Alternativo B: Pesquisa utilizando filtro simples (Categoria)	
2. Caso o usuário deseje apenas consultar o valor nutricional dos alimentos da categoria sem uma pesquisa muito específica sobre um alimento só, o mesmo utilizará a o campo da caixa de seleção.	3. Atualizar tabela e exibir nome do alimento, porção e os seguintes valores nutricionais: Carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio. Através de uma lista dos alimentos existentes no banco sobre a categoria que foi pesquisada, sendo ampla a busca por categoria.
Fluxo Alternativo C: Pesquisa de alimento não existente no banco	
2. Caso o usuário queira consultar o valor nutricional de um alimento não existente no banco.	3. Direcionar o usuário para um modal em que haverá um alerta explicando que o sistema não poderá realizar tal ação.
Protótipo de Interface Homem-Máquina:	
 <p>The screenshot displays a web application interface for searching food items. At the top, there is a navigation bar with links: INÍCIO, ATUALIZAR, EXCLUIR, LISTA DE USUÁRIOS, REDE SOCIAL, INÍCIO, and SAIR. Below the navigation bar, the title 'Tela do Usuário' is shown. The main heading is 'Pesquise as informações nutricionais de seu alimento!'. There is a search form with a text input field labeled 'Nome', a dropdown menu labeled 'Carne e Ovos', and a 'Pesquisar' button. Below the search form, a table lists food items with their nutritional information. The table has four columns: #, Nome, Informações Nutricionais, and Porção. The items listed are: 1. Frango assado, 2. Filé de Frango, 3. Frango xades, 4. Bife de Vaca, 5. Queijo, and 6. Queijo. Each item has a detailed list of nutritional values. A green button with an upward arrow is located at the bottom right of the table.</p>	

A Tabela 2 apresenta a documentação do caso de uso Gerenciar Meus Alimentos.

Tabela 2 – Caso de uso Gerenciar Meus Alimentos

Nome do Caso de Uso: Gerenciar Meus Alimentos	
Breve Descrição:	Esse caso de uso ocorre sempre que o usuário desejar cadastrar algum alimento
Ator Principal:	População
Pré-Condição:	Acessar o site e logar como usuário normal.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Poderá cadastrar seu alimento desejado e informando seus valores nutricionais, bem como excluir e atualizar. Além disso poderá visualizar os alimentos já presentes no banco de dados.	2. Na listagem dos alimentos, o usuário poderá visualizar os alimentos, onde apareceram, em uma tabela, os campos: nome, porção do alimento, unidade de medida, tipo do alimento e as informações nutricionais que são: Carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio. Além dessa funcionalidade será possível que o usuário exclua um alimento, por meio do ícone de lixeira. Bem como poderá atualizar por meio do ícone de lápis.
3. Fim do caso de Uso.	

Protótipo de Interface Homem-Máquina:



A Tabela 3 apresenta a documentação do caso de uso Inserir Meus Alimentos.

Tabela 3 – Caso de uso Inserir Meus Alimentos

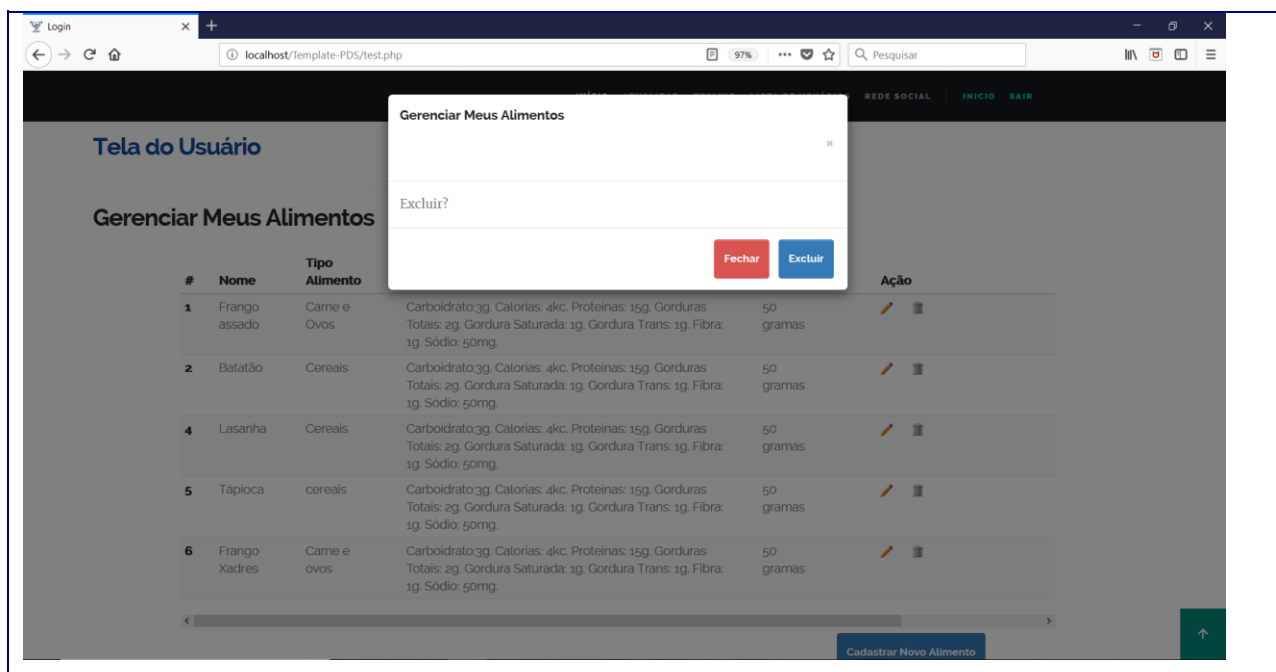
Nome do Caso de Uso: Inserir Meus Alimentos	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre que o usuário desejar inserir o seu próprio alimento.
Ator Principal:	População
Pré-Condição:	Acessar o site e estar logado como usuário normal.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Poderá inserir dados sobre seus próprios alimentos, ou seja, nome, porção do alimento e as informações nutricionais que são: Carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio.	2. Ao querer adicionar um devido alimento, o sistema terá a opção de clicar em um botão, presente no canto inferior direito, que o levará a uma nova janela que permitirá que ele consiga inserir o novo alimento, informando seu nome, porção do alimento, unidade de medida, tipo do alimento e as informações nutricionais.
3. Fim do Caso de Uso.	
Fluxo Alternativo A: Inserir Alimentos já existentes	
2. Caso a População tente inserir um alimento já existente no banco.	3. O sistema deverá exibir um dialog com a informação de que o alimento já está inserido no banco, e voltar a tela de listagem.
Fluxo Alternativo B: Não Preencher nenhum campo	
2. Caso a População, no momento da inserção de alimentos deixar todos os campos em branco.	3. O sistema deverá exibir um dialog com a informação de que nenhum campo foi preenchido, voltando a tela de listagem.

Protótipo de Interface Homem-Máquina:

A Tabela 4 apresenta a documentação do caso de uso Excluir Meus Alimentos.

Tabela 4 - Caso de uso Excluir meus Alimentos

Nome do Caso de Uso: Excluir meus Alimentos	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre que o usuário desejar excluir seu alimento.
Ator Principal:	População
Pré-Condição:	Acessar o site e estar logado como usuário normal.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Poderá excluir os alimentos que foram inseridos corretamente ou incorretamente, através do ícone da lixeira, presente ao lado do alimento listado.	2. O Sistema irá disponibilizar na tela um modal, quando clicar no “link” que possui um ícone de lixeira e pedirá a confirmação da exclusão.
3. Fim de Caso de Uso.	
Protótipo de Interface Homem-Máquina:	



A Tabela 5 apresenta a documentação do caso de uso Atualizar Meus Alimentos.

Tabela 5 – Caso de uso Atualizar meus Alimentos

Nome do Caso de Uso: Atualizar meus Alimentos	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre que o usuário desejar atualizar os dados de um alimento.
Ator Principal:	População
Pré-Condição:	Acessar o site e estar logado como usuário normal.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Poderá atualizar os dados dos alimentos que foram inseridos pelos usuários.	2. O sistema irá disponibilizar um modal com os campos nome, porção do alimento e as informações nutricionais que são: Carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio. Os quais já estarão preenchidos com os dados antigos, onde o usuário poderá editar ou manter igual, e um botão ao lado dos campos escrito “Salvar Mudanças” e ao lado desse outro escrito “Fechar”.
3. Fim do Caso de Uso.	
Fluxo Alternativo A: Apagar os campos preenchidos e deixar em branco.	

2. Caso a População, no momento da atualização de alimentos, apagar os dados dos campos nome, porção e nutrientes adjacentes, ou seja, deixando tudo em branco.	3. O sistema deverá exibir um pop-up com a informação de que os campos nome, porção e nutrientes adjacentes, não foram preenchidos, e por conseguinte, voltando a tela de atualização para que possa ser feito corretamente.
---	--

Fluxo Alternativo B: Atualizar com o mesmo nome de um já existente.

2. Caso a População tente atualizar um alimento com o mesmo nome de outro alimento que já está inserido no banco.	3. O sistema deverá exibir um pop-up com a informação de que o alimento já está inserido no banco, e por conseguinte, voltando a tela de atualização para que possa ser feito corretamente.
---	---

Protótipo de Interface Homem-Máquina:

The screenshot shows a web browser window with the URL `localhost/template-PDS/test.php`. The page has a dark header with 'Login' and navigation links 'REDE SOCIAL', 'INICIO', and 'SAIR'. The main content area is titled 'Tela do Usuário' and 'Gerenciar Meus Alimentos'. It contains a table with columns '#', 'Nome', and 'Tipo Alimento'. The table lists six items: 1. Frango assado (Carne e Ovos), 2. Batatão (Cereais), 4. Lasanha (Cereais), 5. Tapioca (cereais), and 6. Frango Xadres (Carne e ovos). A modal window titled 'Gerenciar Meus Alimentos' is open, showing a form for editing the selected item 'Frango Assado'. The form includes input fields for 'Nome' (containing 'Frango Assado'), 'Carboidratos' (3), 'Calorias' (4), 'Proteínas' (15), 'Gorduras T.' (2), 'Gordura Sat.' (1), 'G. Trans' (1), 'Fibra' (1), 'Sódio' (50), 'Valor da Porção.' (50), 'Un. Medida' (Gramas), and 'Tipo Alimento' (Carne e Ovos). There are 'Salvar Mudanças' and 'Fechar' buttons at the bottom of the modal.

2.2.2 Tabela de casos de teste

Os testes realizados no módulo de ferramentas nutricionais foram todos feitos manualmente, pois o nome dos campos, limite de caracteres do campo, estilo do campo, tipo do campo e o que deve conter no campo foi definido no documento de caso de teste. Além de ser definido tudo isso no documento, é definido também a ação esperada quando se executa os casos de teste, ou seja, como foi seguido especificadamente os documentos de casos de uso e de casos de teste, todas as ações previstas aconteceram. A definição dos limites e tipos do campo são muito importantes, pois na utilização é necessário que o que for digitado, corresponda ao que está sendo exibido. Nesse documento contém a descrição do caso de teste, que contém o que deve ser testado e como deve ser testado para que seja obtido o mesmo resultado na tabela ao lado, que é resultado esperado, onde é definido, caso a ação seja feita corretamente, o que deve acontecer, e se não acontecer conforme o

descrito nessa tabela, significa que está errado e que deve ser colocado em observação para que seja corrigido o mais rápido possível. E para que se tenha um controle na parte da tabela de sucesso e falha, é necessário a numeração de cada caso de teste, para que seja fácil de encontra-lo para definir seu status.

Seguindo o que foi descrito acima a respeito do documento de casos de teste do módulo de ferramentas nutricionais, é possível montar tabelas com as mesmas especificações. Porém, como nos casos de uso foi selecionado as mais importantes, abaixo está listada as tabelas de casos de teste correspondente as tabelas de casos de uso no item 2.2.1 deste documento.

A Tabela 6 apresenta a documentação do caso de teste Pesquisar Caloria de Alimentos.

Tabela 6 – Caso de teste Pesquisar Caloria de Alimento

Caso de teste: Pesquisar Caloria de Alimento		
Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #01	Verifica se a pesquisa dupla por meio do nome do alimento e da seleção de alimentos funciona de forma correta.	Caso o usuário queira a pesquisa dupla por nome e categoria, o sistema deve exibir o nome e porção do alimento bem como os seguintes valores nutricionais: carboidrato, calorias, proteínas, gordura total, gordura saturada, gordura trans, fibras e sódio.
CT #02	Verifica se a pesquisa simples por nome retorna os valores corretos.	Caso o usuário queira pesquisar de forma simples, por meio do nome do alimento, o sistema deverá retornar ao usuário todos os alimentos que possuam o nome informado, exibindo concomitantemente os atributos do alimento devolvido.
CT #03	Verifica se a pesquisa simples por categoria retorna os valores corretos.	Caso o usuário queira pesquisar de forma simples, por meio do nome da categoria, o sistema deverá permitir que o usuário escolha uma das categorias dos dropdown's

		possíveis. Sendo que ao selecionar uma delas, o sistema retorne todos os alimentos que façam parte daquela categoria, exibindo conjuntamente os atributos do alimento devolvido.
CT #04	Verificar se o parâmetro de alimento passado é existente, caso não for existente o sistema deverá tomar uma posição contra.	No caso do usuário querer pesquisar algum alimento não existente no banco, o sistema deve exibir um modal impedindo o usuário de realizar tal ação e redireciona-lo a mesma página, pedindo que informe um outro nome, pois aquele não se encontra cadastrado no sistema.
CT #05	Verifica se o nome informado no seu campo respectivo e sua categoria são pertencentes um ao outro, caso não seja, o sistema deve informar ao usuário a inviabilidade de sua ação.	Quando o nome digitado não for pertencente a sua categoria ou vice-versa, o sistema deve exibir um modal impedindo o usuário de realizar tal ação e redireciona-lo a mesma página, pedindo que informe um outro nome, ou, selecione outra categoria, pois aqueles dados não são associados um ao outro.
CT #06	Verifica se os campos tem algum valor digitado ou selecionado, para exibir as ações devidas, caso ambos vazios, o sistema deve informar ao usuário a inviabilidade de sua ação.	Se o usuário quiser pesquisar algo com os campos em branco, o sistema retornará um modal que interdirá sua ação, de modo redirecioná-lo a mesma página, pedindo que sejam passados os parâmetros necessários para a pesquisa.

A Tabela 7 apresenta a documentação do caso de teste Gerenciar Meus Alimentos.

Tabela 7 – Caso de teste Gerenciar Meus Alimentos

Caso de Teste: Gerenciar Meus Alimentos		
Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #07	Verifica se os alimentos cadastrados pelo usuário podem ser visualizados pelo mesmo, por meio de uma lista.	Caso o usuário acesse a página de Meus Alimentos, deverá ser possível visualizar os alimentos cadastrados por ele, com seus respectivos atributos designados na inserção.
CT #08	Verifica se há alimentos cadastrados pelo usuário.	Caso não haja alimentos cadastrados pelo usuário, o sistema deverá exibir um modal informando que o sistema não possui alimentos cadastrados, e, recomendará a inserção de alimentos por meio do botão “Cadastrar Novo Alimento”.
CT #09	Verifica se opção de “Excluir alimento” funciona corretamente.	Caso o usuário deseje excluir algum alimento, poderá clicar na opção “excluir” ao lado da descrição do alimento. Assim, o sistema direcionará o usuário a uma outra página em que será possível realizar excluir.
CT #10	Verifica se opção de “Editar alimento” funciona corretamente.	Caso o usuário deseje editar algum alimento, poderá clicar na opção “editar” ao lado da descrição do alimento. Assim, o sistema direcionará o usuário a uma outra página em que será possível realizar editar.
CT #11	Verifica se opção de “Cadastrar Novo Alimento” funciona corretamente.	Caso o usuário deseje inserir algum alimento, poderá clicar na opção “Cadastrar Novo Alimento” localizado no canto inferior direito. Assim, o sistema direcionará o usuário a uma outra

		página em que será possível realizar inserir um novo alimento.
--	--	--

A Tabela 8 apresenta a documentação do caso de teste Inserir Meus Alimentos.

Tabela 8 – Caso de teste Inserir Meus Alimentos

Caso de Teste: Inserir Meus Alimentos		
Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #12	Verifica se os campos de inserção são apresentados de maneira apropriada.	<p>Ao clicar na opção “Cadastrar Novo Alimento”, é necessário que o sistema exiba com excelência, os campos de “dropdown’s” com todas as opções, nome como alfa numérico e todos os campos:</p> <ul style="list-style-type: none"> -Nome; -Carboidratos; -Calorias; -Proteínas; -Gorduras trans.; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio; -Valor da porção; -Unidade de media; -Tipo de alimentos.
CT #13	Verificar se os campos não estão em branco ao se tentar inserir um alimento.	Caso o usuário não digite/selecione as informações necessárias para a inserção, o sistema deve barrar o usuário através de um modal, sejam elas em totalidade ou em unidade, dizendo que é preciso preencher todos

		<p>os campos, redirecionando-o para o mesmo formulário. No caso de um só campo não for preenchido os outros devem manter o preenchimento. Sendo os campos obrigatórios:</p> <ul style="list-style-type: none"> -Nome; -Carboidratos; -Calorias; -Proteínas; -Gorduras trans.; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio; -Valor da porção; -Unidade de media; -Tipo de alimentos.
CT #14	Verifica se o alimento inserido já é existente no sistema.	Caso o usuário deseje inserir um alimento que já esteja no sistema, este deve exibir um modal informando que já existe esse alimento no sistema, redirecionando-o para página de formulário em que se encontrava, com as devidas informações preenchidas.
CT #15	Verifica se o campo “Carboidratos”, é exibido nulo quando há alguma inserção.	O campo “Carboidratos” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #16	Verifica se o campo “Calorias”, é exibido nulo quando há alguma inserção.	O campo “Calorias” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #17	Verifica se o campo “Proteínas”, é exibido nulo quando há alguma inserção.	O campo “Proteínas” deve ser exibido nulo, quando houver a inserção de algum alimento.

CT #18	Verifica se o campo “Gorduras Totais”, é exibido nulo quando há alguma inserção.	O campo “Gorduras Totais” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #19	Verifica se o campo “Gorduras Saturadas”, é exibido nulo quando há alguma inserção.	O campo “Gorduras Saturadas” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #20	Verifica se o campo “Gorduras trans.”, é exibido nulo quando há alguma inserção.	O campo “Gorduras trans.” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #21	Verifica se o campo “Fibras”, é exibido nulo quando há alguma inserção.	O campo “Fibras” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #22	Verifica se o campo “Sódio”, é exibido nulo quando há alguma inserção.	O campo “Sódio” deve ser exibido nulo, quando houver a inserção de algum alimento.
CT #23	Verifica se o usuário não excedeu o limite do campo de “Carboidratos”, caso houver, o sistema tomará as devidas ações.	O campo “Carboidratos” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #24	Verifica se o usuário não excedeu o limite do campo de “Calorias”, caso houver, o sistema tomará as devidas ações.	O campo “Calorias” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará

		ao campo, com os números preenchidos até onde é possível armazenar.
CT #25	Verifica se o usuário não excedeu o limite do campo de “Proteínas”, caso houver, o sistema tomará as devidas ações.	O campo “Proteínas” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #26	Verifica se o usuário não excedeu o limite do campo de “Gorduras Totais”, caso houver, o sistema tomará as devidas ações.	O campo “Gorduras Totais” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #27	Verifica se o usuário não excedeu o limite do campo de “Gorduras Saturadas”, caso houver, o sistema tomará as devidas ações.	O campo “Gorduras Saturadas” não deve ultrapassar o limite de dez números inteiros e duas casas decimais. Caso ultrapasse o limite, além de serem exclusivamente numéricos, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #28	Verifica se o usuário não excedeu o limite do campo de “Gorduras	O campo “Gorduras trans.” não deve ultrapassar o limite de dez números

	trans.”, caso houver, o sistema tomará as devidas ações.	inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #29	Verifica se o usuário não excedeu o limite do campo de “Fibra”, caso houver, o sistema tomará as devidas ações.	O campo “Fibras” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #30	Verifica se o usuário não excedeu o limite do campo de “Sódio”, caso houver, o sistema tomará as devidas ações.	O campo “Sódio” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #31	Verifica se o usuário não excedeu o limite do campo de “Valor da Porção”, caso houver, o sistema tomará as devidas ações.	O campo “Valor da Porção” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é

		possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #32	Verifica se o usuário não excedeu o limite do campo de “Nome”, caso houver, o sistema tomará as devidas ações.	O campo “Nome” não deve ultrapassar o limite 100 caracteres, podendo ser alfanumérico. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar uma palavra maior do que se encontra. E assim, retornará ao campo, com as palavras preenchidas até onde é possível armazenar.
CT #33	Verifica se o botão “Cadastrar” realmente salvou todas informações e as adicionou no banco de dados.	O botão “Cadastrar” deve cadastrar as informações no banco de dados, retornando a página de listagem com todos os alimento listados anteriormente, inclusive o previamente inserido.

A Tabela 9 apresenta a documentação do caso de teste Excluir Meus Alimentos.

Tabela 9 – Caso de teste Excluir Meus Alimentos

Caso de Teste: Excluir Meus Alimentos		
Casos de Teste	Descrição do Caso de Teste	Resultado Esperado

CT #34	Verifica se o alimento de “Meus Alimentos” é excluído com sucesso.	Caso o usuário deseje excluir algum alimento, poderá fazer isso clicando na opção excluir, ao lado da descrição do alimento a ser excluído. Ao clicar o sistema deve excluir o alimento, onde aparecerá um modal, após a confirmação do usuário, o sistema excluirá o alimento.
CT#35	Verificar se o botão “fechar” funcione corretamente.	Caso o usuário deseje cancelar a exclusão do seu alimento, poderá clicar na opção “fechar”. Assim, o sistema direcionará o usuário a página de origem.

A Tabela 10 apresenta a documentação do caso de teste Editar Meus Alimentos.

Tabela 10 – Caso de teste Editar Meus Alimentos

Caso de Teste: Editar Meus Alimentos		
Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #36	Verifica se os campos de edição aparecem corretamente preenchidos, conforme o usuário determinou.	<p>Caso o usuário selecione o botão editar, ao lado do alimento que ele deseja realizar a edição, devem aparecer corretamente preenchidos, segundo os dados que o usuário inseriu, os campos:</p> <ul style="list-style-type: none"> -Nome; -Carboidratos; -Calorias; -Proteínas; -Gorduras trans.; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio;

		<ul style="list-style-type: none"> -Valor da porção; -Unidade de media; -Tipo de alimentos.
CT #37	Verifica se os campos podem ser editados, conforme o querer do usuário, permitindo a inserção de novos dados.	<p>Caso o usuário acesse a edição dos dados inseridos, os mesmos devem ser passíveis de modificação, usando o teclado e mouse para alterar os seguintes campos:</p> <ul style="list-style-type: none"> -Nome; -Carboidratos; -Calorias; -Proteínas; -Gorduras trans.; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio; -Valor da porção; -Unidade de media; -Tipo de alimentos.
CT #38	Verifica se o usuário não excedeu o limite do campo de “Carboidratos”, caso houver, o sistema tomará as devidas ações.	O campo “Carboidratos” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #39	Verifica se o usuário não excedeu o limite do campo de “Calorias”, caso houver, o sistema tomará as devidas ações.	O campo “Calorias” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso

		ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #40	Verifica se o usuário não excedeu o limite do campo de “Proteínas”, caso houver, o sistema tomará as devidas ações.	O campo “Proteínas” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #41	Verifica se o usuário não excedeu o limite do campo de “Gorduras Totais”, caso houver, o sistema tomará as devidas ações.	O campo “Gorduras Totais” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #42	Verifica se o usuário não excedeu o limite do campo de “Gorduras Saturadas”, caso houver, o sistema tomará as devidas ações.	O campo “Gorduras Saturadas” não deve ultrapassar o limite de dez números inteiros e duas casas decimais. Caso ultrapasse o limite, além de serem exclusivamente numéricos, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará

		ao campo, com os números preenchidos até onde é possível armazenar.
CT #43	Verifica se o usuário não excedeu o limite do campo de “Gorduras Trans”, caso houver, o sistema tomará as devidas ações.	O campo “Gorduras Trans” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #44	Verifica se o usuário não excedeu o limite do campo de “Fibra”, caso houver, o sistema tomará as devidas ações.	O campo “Fibras” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #45	Verifica se o usuário não excedeu o limite do campo de “Sódio”, caso houver, o sistema tomará as devidas ações.	O campo “Sódio” não deve ultrapassar o limite de dez números inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #46	Verifica se o usuário não excedeu o limite do campo de “Valor da	O campo “Valor da Porção” não deve ultrapassar o limite de dez números

	Porção”, caso houver, o sistema tomará as devidas ações.	inteiros e duas casas decimais, além de serem exclusivamente numéricos. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar um número maior do que se encontra. E assim, retornará ao campo, com os números preenchidos até onde é possível armazenar.
CT #47	Verifica se o usuário não excedeu o limite do campo de “Nome”, caso houver, o sistema tomará as devidas ações.	O campo “Nome” não deve ultrapassar o limite 100 caracteres, podendo ser alfanumérico. Caso ultrapasse o limite, o sistema deve exibir um modal dizendo que não é possível armazenar uma palavra maior do que se encontra. E assim, retornará ao campo, com as palavras preenchidas até onde é possível armazenar.
CT #48	Verifica se os campos numerais sejam preenchidos somente por numerais.	Os campos numerais não devem ser preenchidos com caracteres diferentes de decimais e inteiros, esses campos são: <ul style="list-style-type: none"> -Carboidratos; -Calorias; -Proteínas; -Gorduras trans.; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio; -Valor da porção;
CT #49	Verifica se os campo nome pode ser alterado em caracteres e numerais.	O campo nome pode ser preenchido por caracteres e números.
CT #50	Verificar se os campos não estão	Caso o usuário não digite/selecione as

	em branco ao se tentar editar um alimento.	informações necessárias para a edição, o sistema deve barrar o usuário através de um modal, sejam elas em totalidade ou em unidade, dizendo que é preciso preencher todos os campos, redirecionando-o para o mesmo formulário. No caso de um só campo não for preenchido os outros devem manter o preenchimento. Sendo os campos obrigatórios: <ul style="list-style-type: none"> -Nome; -Carboidratos; -Calorias; -Proteínas; -Gorduras trans; -Gorduras saturadas; -Gorduras totais; -Fibra; -Sódio; -Valor da porção; -Unidade de media; -Tipo de alimentos.
CT #51	Verifica se os campo nome, não é alterado para o nome já existente no banco.	O campo nome não deve ser alterado para um nome já existente no banco. Caso o usuário informe um nome já existente, será exibido um modal para informar o erro e solicitar a troca do nome, voltando ao formulário de edição com as informações que já estavam inseridas.
CT #52	Verifica se a edição é salva com sucesso, por meio do botão “Salvar Mudanças”.	A edição deve ser salva e atualizada na listagem de Meus Alimentos, após a confirmação.

2.2.3 Tabela de sucesso/falha

Após a execução dos casos de teste descritos, da forma detalhada acima, foi necessária uma tabela para que houvesse o controle de todas as datas previstas para a entrega, e para que o cronograma fosse seguido corretamente e sem atrasos, os casos de testes eram executados um ou dois dias antes para que pudesse ser feito todos os ajustes. Nessa tabela continha o planejamento e o acompanhamento da execução dos testes, onde para cada caso de teste, era especificado a data que seria realizado o teste, e quais os possíveis dias que poderiam ser realizados outros testes caso houvesse algum erro que precisasse ser corrigido, onde essas outras datas para realizar os testes que obtiveram falhas eram definidas em um dia de comum acordo entre os desenvolvedores e os testadores. E como já foi especificado anteriormente, foi selecionado apenas os considerados importantes para esta listagem, ou seja, a tabela feita abaixo, descreve apenas os casos de teste correspondentes aos especificados no item 2.2.2.

A Tabela 11 apresenta a documentação dos testes executados e o que foi obtido.

Tabela 11 – Tabela de Sucesso/Falha

	16/08/18	23/08/18	27/08/18	30/08/18	17/09/18	25/09/18
CT #01					Sucesso	
CT #02					Sucesso	
CT #03					Sucesso	
CT #04					Sucesso	
CT #05					Sucesso	
CT #06					Sucesso	
CT #07					Sucesso	
CT #08					Sucesso	
CT #09					Sucesso	
CT #10					Sucesso	
CT #11					Sucesso	
CT #12					Sucesso	
CT #13					Sucesso	
CT #14					Sucesso	
CT #15					Sucesso	
CT #16					Sucesso	
CT #17					Sucesso	
CT #18					Sucesso	

CT #19					Sucesso	
CT #20					Sucesso	
CT #21					Sucesso	
CT #22					Sucesso	
CT #23					Sucesso	
CT #24					Sucesso	
CT #25					Sucesso	
CT #26					Sucesso	
CT #27					Sucesso	
CT #28					Sucesso	
CT #29					Sucesso	
CT #30					Sucesso	
CT #31					Sucesso	
CT #32					Sucesso	
CT #33					Sucesso	
CT #34					Sucesso	
CT #35					Sucesso	
CT #36					Sucesso	
CT #37					Sucesso	
CT #38					Sucesso	
CT #39					Sucesso	
CT #40					Sucesso	
CT #41					Sucesso	
CT #42					Sucesso	
CT #43					Sucesso	
CT #44					Sucesso	
CT #45					Sucesso	
CT #46					Sucesso	
CT #47					Sucesso	
CT #48					Falha	Sucesso
CT #49					Sucesso	
CT #50					Sucesso	

CT #51					Sucesso	
CT #52					Falha	Sucesso

2.2.4 Análise dos protótipos feitos

Como pode ser observado na tabela exibida no item 2.2.3, todos os protótipos apresentados no item 2.2.1 foram seguidos corretamente obtiveram sucesso ao serem testados.

3 Conclusões e Recomendações

Considerando o objetivo de criar e aplicar os casos de testes no projeto Mais Saúde São João para que fosse possível atingir a minimização dos erros e deixar mais fácil na questão de usabilidade em horários de pico, para que não houvesse problemas futuros que acarretassem problemas que poderiam ter sido solucionados na fase de casos de teste, foi necessário criar primeiramente uma documentação a respeito de casos de uso, com o mais profundo detalhamento, para que assim, no documento de casos de teste, houvesse uma documentação completa para que não houvesse problemas ao longo do desenvolvimento. E isso foi feito através da análise dos protótipos e dos fluxos principais e alternativos que estavam inseridos na documentação de casos de uso, para que na hora que estivesse sendo construído o oficial, não houvesse falhas.

Como houve um total de 23 casos de uso com fluxos principais e alternativos para que fosse desenvolvida, foi observado que mesmo com essa quantidade de casos de uso, todos funcionaram conforme o que foi especificado nos casos de teste em sua última versão.

Nos casos de teste do módulo de ferramentas nutricionais, houve um total de 172 casos de testes que tinham a função de diagnosticar problemas no software que impossibilitava a sua utilização adequada e sem conflitos. Todos os casos de testes foram definidos como sucesso. Depois de alcançar o sucesso nos 172 casos de teste, o módulo de ferramentas nutricionais foi integrado junto com outros 8 módulos distintos.

Durante a criação do documento, houve alguns problemas por conta da complexidade e conhecimento acerca de casos de teste para se produzir um documento satisfatório, e também o fato da escola não ter um ambiente que suportasse duas turmas com 49 alunos. Houve-se a vantagem em desenvolver em um ambiente acadêmico por conta do grande suporte que foi obtido através de professores qualificados e que exerciam atividades na área. E também a questão de se trabalhar com esse número de alunos, pois é adquirido muita experiência ao trabalhar com tantas pessoas com pensamentos distintos em algumas partes.

Como uma recomendação futura, deve-se haver uma atualização frequente dos documentos de teste e de casos de uso, para que isso não prejudique o desempenho. E como os testes são na maioria das vezes extensos, é sugere-se também cogitar a possibilidade de utilizar uma ferramenta de testes, para que não haja sobrecarga de trabalhos que poderiam ter sido diminuídos se fosse utilizado uma ferramenta própria para testes.

4. Referências Bibliográficas

- [1] INSTITUTO FEDERAL DE SÃO PAULO. Estude aqui. Disponível em: <<https://www.sbv.ifsp.edu.br/sobre-campus>>. Acesso em: 17 set. 2018.
- [2] INSTITUTO FEDERAL DE SÃO PAULO. Técnico Integrado em Informática. Disponível em: <<https://www.sbv.ifsp.edu.br/index.php/component/content/article/64-ensino/cursos/168-tecnico-integrado-informatica>>. Acesso em: 17 set. 2018.
- [3] BRASIL. IBGE. Instituto Brasileiro de Geografia e Estatística. **São João da Boa Vista**. Disponível em: <<https://www.ibge.gov.br/estatisticas-novoportal/por-cidade-estado-estatisticas.html?t=destaques&c=3549102>>. Acesso em: 27 ago. 2018.
- [4] ROMANO, BRENO. Termo de Abertura do Projeto Mais Saúde São João. Disponível em: <<https://svn.sbv.ifsp.edu.br/svn/pds2018vespertino/>>. Acesso em: 17 set. 2018.
- [5] ROMANO, BRENO. Macrorrequisitos dos Módulos do Projeto. Disponível em: <<https://sites.google.com/site/blromano/disciplinas/pds2014>>. Acesso em: 17 set. 2018.
- [6] NETO, ARILO. Casos de Teste: Aprimore seus casos e procedimentos de teste. Disponível em: <<https://www.devmedia.com.br/casos-de-teste-aprimore-seus-casos-e-procedimentos-de-teste/30526>>. Acesso em: 27 ago. 2018.
- [7] RIBEIRO, LEANDRO. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 13 set. 2018.
- [8] SOUZA, KARLA. GASPAROTTO, ANGELITA. A Importância da Atividade de Teste no Desenvolvimento de Software. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2013_TN_STO_177_007_23030.pdf>. Acesso em: 15 set. 2018.
- [9] DEVMEDIA. Casos de Teste: Importância da rastreabilidade entre requisitos. Disponível em: <<https://www.devmedia.com.br/casos-de-teste-importancia-da-rastreabilidade-entre-requisitos/30524>>. Acesso em: 15 set. 2018.
- [10] MERENDA, ESTHER. Definição de Casos de Teste Funcionais a partir de Casos de Uso. Disponível em: <<https://slideplayer.com.br/slide/326794/>>. Acesso em: 15 set. 2018.
- [11] HOOGENRAAD, WIM. Casos de teste, exemplos e melhores práticas. Disponível em: <<https://www.itpedia.nl/pt/2018/06/01/testcases-voorbeelden-en-best-practices/>>. Acesso em: 19 set. 2018.

[12] MELO, WALTER. Qualidade + Testes de Softwares = Qualidade de Software. Disponível em: <<https://www.tiespecialistas.com.br/qualidade-testes-de-softwares-qualidade-de-software/>>. Acesso em: 04 out. 2018.

[13] ROCHA, FABIO. A importância dos testes para a qualidade do software. Disponível em: <<https://www.devmedia.com.br/a-importancia-dos-testes-para-a-qualidade-do-software/28439>>. Acesso em: 04 out. 2018.

[14] PORTAL GSTI. O que é Teste de Software? Disponível em: <<https://www.portalgsti.com.br/testes-de-software/sobre/>>. Acesso em: 04 out. 2018.

[15] ALMEIDA, CARLA. Introdução ao Teste de Software. Disponível em: <<http://www.linhadecodigo.com.br/artigo/2775/introducao-ao-teste-de-software.aspx>>. Acesso em: 04 out. 2018.

[16] BARBOSA, JOSÉ. Testes: Diferença entre Erro, Defeito e Falha. Disponível em: <<http://josebarbosa.com.br/?p=463>>. Acesso em: 16 out. 2018.