

**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 e Prof. Luiz Angelo Valota Francisco

**O processo de elaboração dos casos de teste do módulo de nutrição do
projeto Gerações**

Aluno: Aline Faustino

Prontuário: 1620126

São João da Boa Vista – SP

2019

Resumo

Este trabalho descreve o processo de criação da parte de documentação do módulo de nutrição do projeto do *software* Gerações, que tem como objetivo automatizar o sistema de gerenciamento de uma instituição de longa permanência. Dessa forma, será descrito cada etapa da parte dos documentos, contendo os requisitos, casos de uso e casos de teste, focando nos dados referentes ao módulo de nutrição e nos casos de teste que serão realizadas após toda a parte de programação, sendo fundamentais para evitar que erros cheguem até o usuário final. Por conseguinte, o documento trará a teoria por trás de cada etapa e como foram realizados dentro da equipe, contendo imagens reais do que foi produzido pelos alunos.

Assim sendo, posterior a todo as atividades realizadas serem apresentadas haverá também o levantamento dos pontos positivos e negativos do trabalho, além de conter também recomendações para que ocorram mudanças nos próximos anos.

Palavras-chave: *software*, instituições de longa permanência, módulo de nutrição, requisitos, casos de uso, casos de teste.

Sumário

1	Introdução	5
1.1	Contextualização/Motivação	5
1.2	Objetivo Geral da Pesquisa	7
1.3	Objetivos Específicos	8
1.4	Estrutura do Documento	8
2	Desenvolvimento	9
2.1	Levantamento Bibliográfico	9
2.1.1	Especificação de requisitos	9
2.1.2	Levantamento de requisitos	9
2.1.3	Problemas na elaboração de requisitos	10
2.1.4	Relação entre cliente e analista	10
2.1.5	Dependência dos casos de uso	11
2.1.6	Definição de casos de uso em diagrama	11
2.1.7	Documentação dos casos de uso	13
2.1.8	Casos de teste	13
2.1.9	Tipos de teste de <i>software</i>	14
2.1.10	Testes de caixa preta	14
2.2	Etapas para o desenvolvimento da pesquisa	15
2.2.1	Requisitos do módulo de nutrição	15
2.2.2	Diagrama de casos de uso do módulo de nutrição	16
2.2.3	Elaboração dos casos de teste referentes aos casos de uso do módulo de nutrição	17
2.2.4	Execução dos Casos de Teste do Módulo Nutrição	19
3	Conclusões e Recomendações	21
4	Referências Bibliográficas	22

Índice de figuras

Figura 1: Requisitos dos módulos 1, 2 e 3	6
Figura 2: Requisitos dos módulos 4, 5 e 6	6
Figura 3: Requisitos dos módulos 7, 8 e 9	7
Figura 4: Exemplo de má comunicação entre cliente e analista	11
Figura 5: Exemplo de diagrama de casos de uso	12
Figura 6: Exemplo com <i>include</i> e <i>extend</i>	12
Figura 7: Exemplo de tabela dos casos de uso	13
Figura 8: Exemplo de casos de teste	14
Figura 9: Exemplo de testes realizados	15
Figura 10: Exemplo de falhas documentadas.....	15
Figura 11: Requisitos RF #01, RF #02, RF #03, RF #04, RF #05, RF #06 e RF #07 do módulo de nutrição.....	16
Figura 12: Caso de teste do montar refeição	17
Figura 13: Caso de teste do cadastrar tipo de alimento.....	18
Figura 14: Caso de teste do cadastrar alimento.....	18
Figura 15: Caso de teste do cadastrar padrão alimentar.....	18
Figura 16: Caso de teste do agendar consulta	19
Figura 17: Tabela de testes efetuados pelo módulo de nutrição	19
Figura 18: Tabela de falhas nos testes efetuados no módulo de nutrição	20

1 Introdução

1.1 Contextualização/Motivação

Se for analisado a realidade presente em São João da Boa Vista, cidade situada no interior de São Paulo com uma população de aproximadamente 84 mil pessoas segundo dados do IBGE Do ano de 2010, é notório a inexistência de um elevado número de meios tecnológicos voltados unicamente para idosos, todavia há medidas sociais voltadas para esse grupo de pessoas, havendo como exemplo os Centros de Convivência dos Idosos (CICs) e o Centro de Integração do Idoso, por constituírem uma parcela significativa dentro da população do município. [1][2]

Ademais, pode ser constatado esse fator ao observar que mesmo havendo diversas instituições de longa permanência na cidade, que se caracterizam por seguirem as normas contidas na Agência Nacional de Vigilância Sanitária (ANVISA), sendo indispensável que os direitos humanos dos residentes sejam respeitados, garantindo sua liberdade e privacidade, incentivando a participação em atividades locais, a convivência dos idosos entre si e com as suas famílias, promovendo momentos de lazer e tendo uma rotina pré-estabelecida, nenhuma delas possui um sistema de gerenciamento especializado para as necessidades dessas organizações que visam trazer comodidade para os indivíduos que moram dentro de seus domínios. [3]

Com o intuito de alterar essa realidade trazendo mais confiabilidade em relação ao gerenciamento dessas fundações os alunos do 4º ano de informática do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo do campus de São João da Boa Vista, que possui como missão dentro da sociedade formar cidadãos que visam enriquecer o cenário ao qual fazem parte através de seu conhecimento adquirido, dessa forma, tendo projetos voltados para fins sociais. Em suma, o software que será criado pelas turmas 419 e 420 na matéria de Prática de Desenvolvimento de Sistemas (PDS), com os professores Breno Lisi Romano e Luiz Angelo Valota Francisco da disciplina, será voltado unicamente para poder fazer com que casas de longa permanência possam cada vez mais atender as necessidades dos idosos de uma forma mais satisfatória e organizada. [4]

Por consequência, devido à complexidade existente no programa e ao número relativamente elevado de alunos para sua produção foi necessário separá-los em grupos com 6 ou 7 pessoas, onde 2 deles são Analistas responsáveis por toda a parte de documentação e pelos testes realizados nas funcionalidades do sistema, 2 DBAs que realizam a criação e manutenção das informações referentes ao banco de dados, e 2 ou 3 Desenvolvedores que produzem todas as telas do projeto. Assim sendo, todas as funcionalidades criadas deverão estar integradas fazendo com que o programa funcione

corretamente e para que isso ocorra de forma satisfatória serão utilizadas ferramentas para o controle de atividades o *Kanbam*, *Redmine* e o *SVN*.

Por conseguinte, cada grupo está voltado para a criação de uma área específica que compõe o software, mantendo-se módulos referentes aos usuários, prontuário dos idosos, acompanhamento de familiares, cuidado diário dos idosos, médicas e controle de incidentes, nutrição, atividades físicas e recreativas, controle administrativos e relatórios especializados. Sendo descrito nas imagens abaixo cada um dos requisitos que compõem as atividades realizadas pelas equipes:

Figura 1: Requisitos dos módulos 1, 2 e 3



Figura 2: Requisitos dos módulos 4, 5 e 6



Figura 3: Requisitos dos módulos 7, 8 e 9



Com o propósito de focar nesse documento mais especificamente no módulo de nutrição, possuindo os requisitos de gerenciamento de tipos de alimentos, alimentos, do padrão alimentar por refeição de cada idoso e do plano alimentar semanal, que se refere a todo o planejamento dos alimentos que serão consumidos pelos idosos dentro da casa de longa permanência durante a semana e o controle de amostras de alimentos consumidos na instituição e registro de consultas periódicas nutricionais por idosos, ocorrendo todas essas atividades através dos nutricionistas e permitindo que os enfermeiros visualizem essas informações inseridas.

Como resultado, cada uma dessas funcionalidades deverá ser realizada de determinada maneira que será documentada e posteriormente deverão ser respeitadas em todo o processo de criação da página, sendo necessário a realização de casos de teste dos casos de uso do sexto módulo para que sejam respeitadas todas as informações descritas na documentação que será realizada.

1.2 Objetivo Geral da Pesquisa

O objetivo geral desse trabalho é demonstrar como será realizado os casos de teste dos casos de uso do módulo 6, que se refere a tudo relacionado com a nutrição dos idosos dentro das instituições, e sua respectiva importância para que posteriormente erros nas funcionalidades sejam inexistentes.

1.3 Objetivos Específicos

Devido a erros existentes em programas após todo o processo de criação que acarretaram em maiores problemas para seus usuários foi notado que medidas deveriam ser tomadas para mudar esse cenário, dessa forma os casos de teste dos métodos de caixa branca e caixa preta foram criados, assim sendo, neste projeto serão utilizados os métodos de caixa preta.

Por conseguinte, será explicado como irá ocorrer a documentação de cada caso de teste se baseando no que foi realizado posteriormente nos casos de uso do módulo de nutrição que descreverá cada passo realizado, desde a descrição de cada ação nas páginas, até o momento em que elas serão testadas para validá-las, existindo uma tabela para que êxitos e fracassos sejam documentados, para ocorrer um controle de qualidade no programa. Dessa forma, erros encontrados serão corrigidos no processo de criação evitando que surtem maiores problemas quando for utilizado pelo usuário final.

1.4 Estrutura do Documento

No primeiro capítulo haverá a introdução contendo a textualização do tema, objetivos do documento de forma geral e específica, no segundo terá o desenvolvimento do tema do trabalho, descrevendo a parte teórica ao que se refere os requisitos, casos de uso e casos de teste, e após isso os requisitos, casos de uso e de casos de teste realizados pelo módulo de nutrição do projeto Gerações, no terceiro a conclusão que será composta por uma recapitulação dos assuntos tratados no desenvolvimento de maneira sucinta para descrever o que pode ser observado de fundamental no documento e no quarto capítulo estão as referências bibliográficas.

2 Desenvolvimento

2.1 Levantamento Bibliográfico

2.1.1 Especificação de requisitos

Segundo os padrões estabelecidos no glossário do *Standard Glossary of Software Engineering Terminology* (IEEE) os requisitos se caracterizam nas necessidades demonstradas pelos clientes e que serão analisadas pela empresa para que seja observado sua importância para fazer parte da documentação e que posteriormente será uma funcionalidade ou restrição no programa, porém se observado sua irrelevância ou impossibilidade de realização devido a algum fator apresentado será descartada. [5]

Assim sendo, são existentes 3 diferentes tipos de requisitos, existindo os explícitos que se caracterizam por estarem descritos nos documentos de requisitos do *software*, os normativos são responsáveis por seguirem leis, normas, padrões e regulamentos já pré-estabelecidos, e há também os implícitos que são as expectativas que os clientes possuem em relação sistema operacional que está em processo de criação, mas que não são documentados. [6]

Analogamente, cada requisito possui pontos que o torna diferente e indispensável dentro da criação do aplicativo, dessa forma, ele possui características consideradas especiais que se subdividem em requisitos funcionais (RF) e requisitos não funcionais (RNF), em que RF são os comportamentos que o sistema deve apresentar quando realizadas determinadas ações pelos usuários e RNF devem determinar aspectos de desempenho e comportamento. [6]

2.1.2 Levantamento de requisitos

O processo inicial do projeto ocorre com o levantamento de requisitos, pois todas as necessidades do cliente em relação ao software nesse primeiro momento deverão ser devidamente apresentadas e estruturadas para que assim sejam minimizadas as chances de futuros erros nas capacidades criadas no programa. Por conseguinte, um analista deve realizar entrevistas com o seu consumidor visando trazer pontos que serão importantes, estrutura-los e documenta-los, sendo fundamental para a criação do sistema, pois há outras atividades que necessitarão dessas informações, como é o caso dos casos de uso e dos casos de teste que serão especificados nos tópicos seguintes do desenvolvimento deste arquivo.

Por consequência, a parte de desenvolvimento passará a ter que respeitar os critérios documentados, porque somente assim o que é necessário será criado e pontos irrelevantes não serão abordados em nenhum momento, isso ocorre devido ao fato que somente quem realizou o diálogo o

usuário sabe mais claramente as necessidades existentes que deverão ser supridas pelo novo aplicativo.

Embora, se todos os envolvidos no projeto, como cliente, analista e desenvolvedor, não se empenharem devidamente e possuírem uma comunicação falha isso acabará resultando em problemas futuros, pois essa atividade reflete no empenho final do *software* dado que ela que define quais funcionalidades o sistema operacional deverá conter e como serão realizadas.

2.1.3 Problemas na elaboração de requisitos

Um dos principais problemas na Engenharia de *Software* referente aos requisitos está relacionado a existência de um desprezo por parte de desenvolvedores para com esse processo, pois o consideram um fator que pode ser descartado. Dessa forma, erros são decorrentes já que aspectos importantes como relevância e necessidade não foram levados em consideração para que o programa fosse produzido. [6]

Igualmente, clientes também passam a desconsiderar que todos os pontos importantes no sistema devem ser devidamente informados e sem alterações futuras, por consequência, isso ressalta uma forte instabilidade para quem está trabalhando e para a eficiência do aplicativo, já que constantemente novas capacidades são inseridas ou descartadas, e muitos pontos importantes não são abordados, acarretando em um elevado número de erros que dificilmente serão arrumados devido ao teor de dificuldade que será encontrado para fazê-lo. [6]

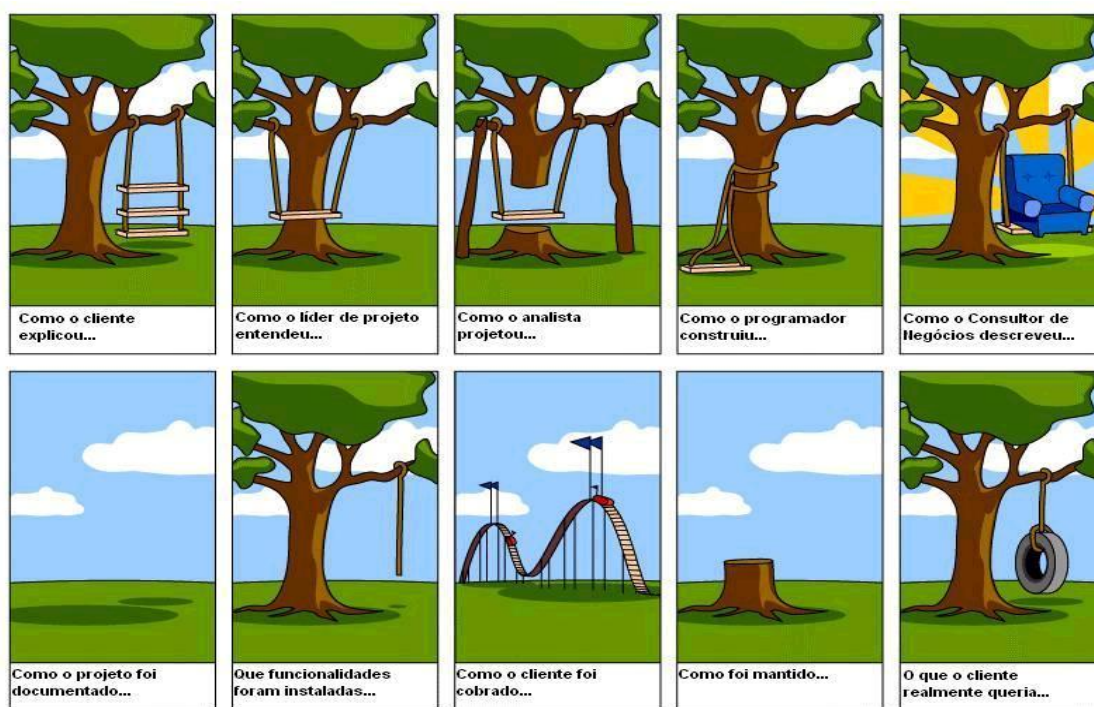
Certamente, se houver uma boa comunicação entre clientes e analistas, boas especificações de requisitos, comprometimento de todos e um respeito pelo que foi documentado problemas ocorrerão em menor escala e poderão ser resolvidos facilmente, tendo menores custos e maior satisfação para todas as partes envolvidas.

2.1.4 Relação entre cliente e analista

Um fator de extrema importância para a criação do *software* é o momento inicial, onde analistas irão conversar com os clientes, pois é nesse momento que diversos pontos serão abordados, ou seja, é o que define as características que deverão conter o programa. Porém, se não houver uma boa comunicação entre analistas e clientes problemas aparecerão, já que em alguns casos acabam sendo projetados e criados aplicativos completamente diferentes do que se era esperado pelos usuários, ocorrendo devido a uma má interpretação do que foi falado por ambas as partes.

Por consequência, tensões desnecessárias apareceram e o produto final não será satisfatório, sendo notório os pontos citados a imagem abaixo, onde ocorre claramente divergência de ideias:

Figura 4: Exemplo de má comunicação entre cliente e analista



[7]

2.1.5 Dependência dos casos de uso

O processo de criação dos casos de uso só pode ocorrer quando todos os requisitos do *software* sejam devidamente documentados e especificados, pois essa etapa de documentação deverá se basear no que foi levantado posteriormente, por consequência, todas as atividades que são realizadas para a produção do sistema estão interligadas e são dependentes entre si.

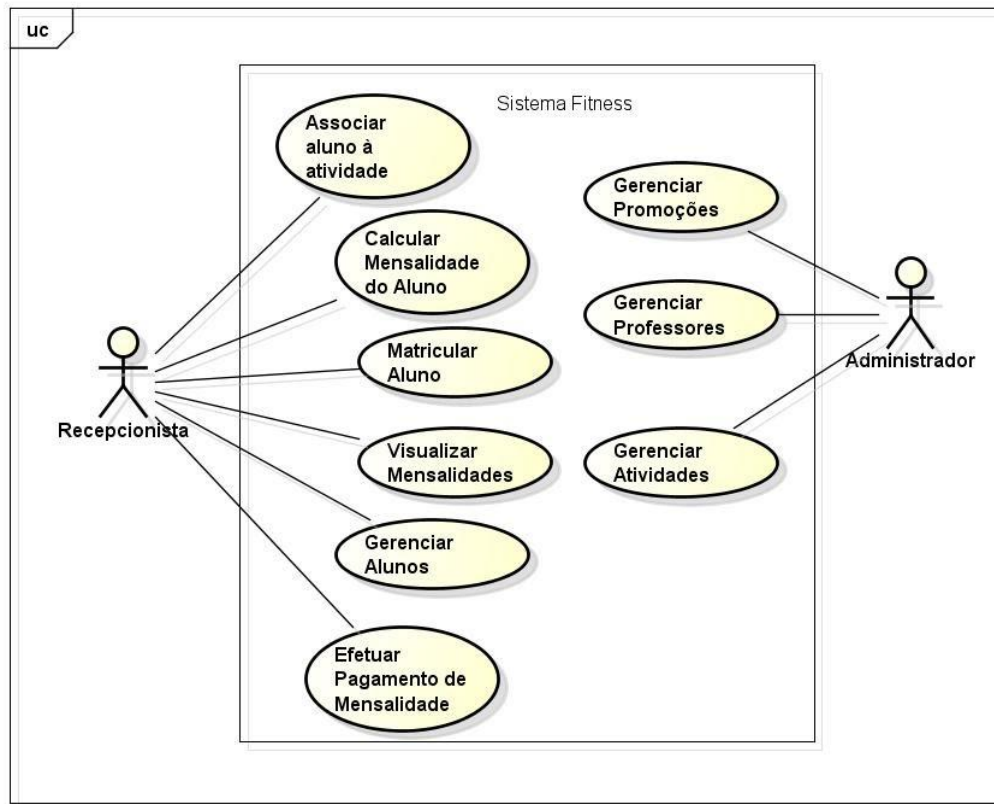
Dessa forma, essa segunda operação tem o intuito de demonstrar de maneira mais completa e simples as funcionalidades e os agentes que as realizarão, fazendo com que em alguns cenários a interpretação de como funcionará o aplicativo seja melhorada para todos os envolvidos nele minimizando os riscos de que ocorram erros significativos no produto final.

2.1.6 Definição de casos de uso em diagrama

O documento de casos de uso se caracteriza pela criação de um diagrama que utiliza atores para representar o agente que realizará determinada ação, sendo ele algo físico ou não, podendo ser clientes, sistema, entre outros, e os casos de uso que são as ações que deverão ser realizadas por um ator. Por conseguinte, cada um dos elementos citados possui relações entre si que possuem como função demonstrar de maneira clara quais são os requisitos do *software* que está sendo criado e como isso ocorrerá no cenário real, passando uma perspectiva mais palpável e segura para quem produz e quem receberá o aplicativo. [6][8]

Dessa forma, na imagem abaixo é possível observar o diagrama de um ambiente fictício que mostra como são realizados:

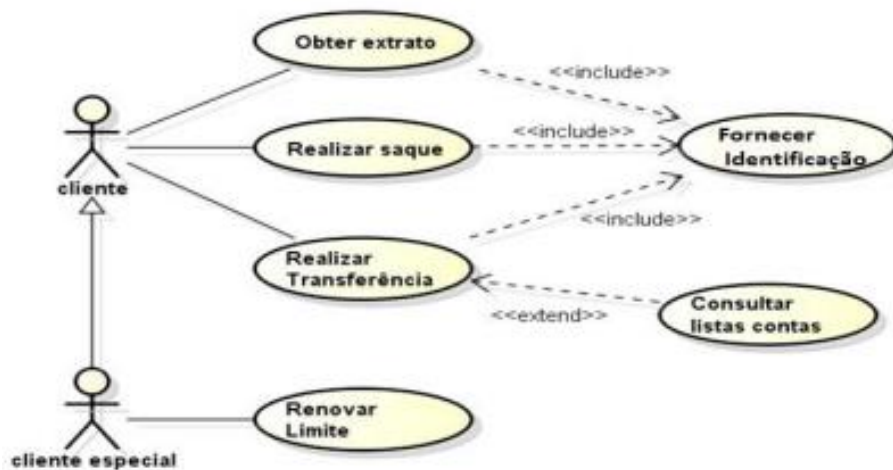
Figura 5: Exemplo de diagrama de casos de uso



powered by astah® [9]

Analogamente, existem dependências de ações entre si, ou seja, para que determinada tarefa ocorra é necessário que uma outra já tenha sido realizada, esse processo é chamado de *include*, e há também atividades que não são obrigatórias podendo ocorrer ou não, sendo chamadas essas ligações de *extend*. Assim sendo, na imagem abaixo é notório os fatos citados:

Figura 6: Exemplo com *include* e *extend*



[10]

2.1.7 Documentação dos casos de uso

A documentação de casos de uso contém os diagramas criados que já foram exemplificados no tópico 2.2.1 e também tabelas que possuem a função de documentar mais claramente cada ação realizada pelos atores que compõem o programa, ou seja, tudo que ocorre no *software* será especificado. Dessa maneira, pode-se observar todas as informações necessárias para o funcionamento satisfatório do sistema no exemplo abaixo:

Figura 7: Exemplo de tabela dos casos de uso

Nome	UC01- Efetuar Login
Objetivo	Efetuar Login.
Atores	Administrador; Usuários Cadastrados.
Dados Consumidos	Login e senha.
Dados Produzidos	Acesso ao sistema.
Prioridade	Alta.
Pré-condições	Possuir cadastro no sistema.
Pós-condições	N/A
Fluxo Principal	
Efetuar login.	
Ator	Sistema
Informar login e senha.	Valida as informações do usuário e senha.
	Se os dados estiverem OK, o sistema carrega a página inicial. Se os dados estiverem incorretos, a aplicação retorna uma mensagem de erro.
Receber mensagem	
Se a mensagem foi de erro, reinicie o processo do fluxo principal.	
Se a mensagem foi de sucesso, encerre o Fluxo	

[11]

2.1.8 Casos de teste

Ao analisar os capítulos anteriores é notório que cada processo na criação de um *software* está propenso a erros, e que todas as etapas estão ligadas, ou seja, uma única falha acarretará em diversas outras que determinaram a ineficácia do produto final.

Dessa forma, para evitar que o programa com características que não condizem com o que era esperado em relação as suas funcionalidades pelo cliente, podendo causar problemas drásticos para instituições, chegue aos seus usuários é necessário a realização de testes que validam as funções do sistema que são realizados por analistas depois que ocorre a desenvolvimento, para que erros possam ser reparados ainda no processo de criação.

Por consequência, essa atividade realizada pelos analistas é responsável por validar cada funcionalidade do *software* que preenche os requisitos estabelecidos pelo cliente e aumenta significativamente a qualidade do produto, sendo algo benéfico para empresa e consumidor.

2.1.9 Tipos de teste de *software*

Existem diversos métodos para a realização de testes de *software* que visam trazer um produto final de qualidade e que respeite os requisitos esperados pelos clientes, sendo que cada forma para ser implementada deverá ser analisada, pois cada uma delas possui aspectos que focam em determinado ponto e que são importantes para um tipo de programa. Ou seja, os testes realizados utilizam métodos que suprem as necessidades do sistema criado. [12]

Por consequência, o projeto Gerações utiliza os testes de caixa-preta por se encaixarem mais facilmente no que é esperado para a finalização desse trabalho.

2.1.10 Testes de caixa preta

Esse tipo de teste é considerado como o teste mais simples, pois ele analisa se os requisitos do programa estão sendo executados da maneira esperada, descartando assim a forma de funcionamento das atividades internamente, possuindo também o nome de teste funcional. Assim sendo, é verificado pelo analista como as ações dos usuários serão empenhadas no sistema, sem validar como está o código fonte. [13][14]

Dessa forma, o analista responsável deverá documentar todas as ações que ocorrerão referentes aos casos de uso previamente realizados e mostrarão o que se espera de cada uma delas, colocando todos os fluxos que existirem, podendo ser observado esses pontos na imagem abaixo:

Figura 8: Exemplo de casos de teste

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #01	Verificar se o nutricionista ao clicar na opção de "Enviar" na atualização dos dados da consulta preencheu todos os campos de preenchimento obrigatório.	O sistema deve validar os campos preenchidos, atualizar no banco de dados as informações que foram alteradas e enviar uma mensagem de "Dados alterados com sucesso".
CT #02	Verificar se o nutricionista ao clicar na opção de "Enviar" na atualização dos dados da consulta não preencheu todos os campos de preenchimento obrigatório.	O sistema deve não efetuar a atualização de dados e mostrar uma mensagem de "Um ou mais campos não foram preenchidos corretamente".

Por conseguinte, ao realizar essa tabela o analista adquire um controle de todas as tarefas a serem realizadas pelo *software* e quais respostas aparecerão na interface quando realizadas determinadas ações, fazendo com que erros possam ser corrigidos e aumentando o nível de qualidade de funcionamento do sistema. [12]

Posteriormente, no período onde os testes ocorrerão efetivamente uma tabela deverá ser criada contendo todas as validações realizadas com data de execução, número referente ao caso de teste e status dessa atividade, ou seja, se ela foi realizada com sucesso ou com falha, podendo ser observado isso no exemplo abaixo:

Figura 9: Exemplo de testes realizados

	02/08/2019
CT #24	Sucesso
CT #25	Sucesso
CT #34	Sucesso
CT #35	Sucesso

Logo após, deverá ser criada uma tabela com todas as falhas que foram observadas contendo o dia em que isso ocorreu, o porquê de ter dado esse erro e quando o sistema será testado novamente, determinando dessa forma um modelo de gerenciamento de status de atividades minucioso, por consequência com resultados de produto final mais satisfatórios. Segue-se abaixo uma tabela de falhas para exemplificação:

Figura 10: Exemplo de falhas documentadas

Casos de Teste	Data da Execução	Responsável da Atividade de Teste	Status do Teste	Motivo da Falha	Responsável pela Correção da Falha	Data Prevista para Novo Teste
CT # 17	23/08/2019	Aline	Falha	Não está ocorrendo a vinculação do tipo de alimento com alimento, permitindo assim que qualquer tipo de alimento seja excluído mesmo que haja a dependência de algum alimento.	Matheus B.	06/09/2019

2.2 Etapas para o desenvolvimento da pesquisa

2.2.1 Requisitos do módulo de nutrição

O módulo de nutrição para conseguir suprir todas as necessidades do *software* referentes a alimentação dos idosos dentro da instituição de longa permanência precisou realizar um levantamento de todos os requisitos necessários, sendo essa etapa realizada por toda a equipe e tendo como responsáveis os dois analistas. Na imagem abaixo há uma imagem com todos os requisitos efetuados dentro do grupo:

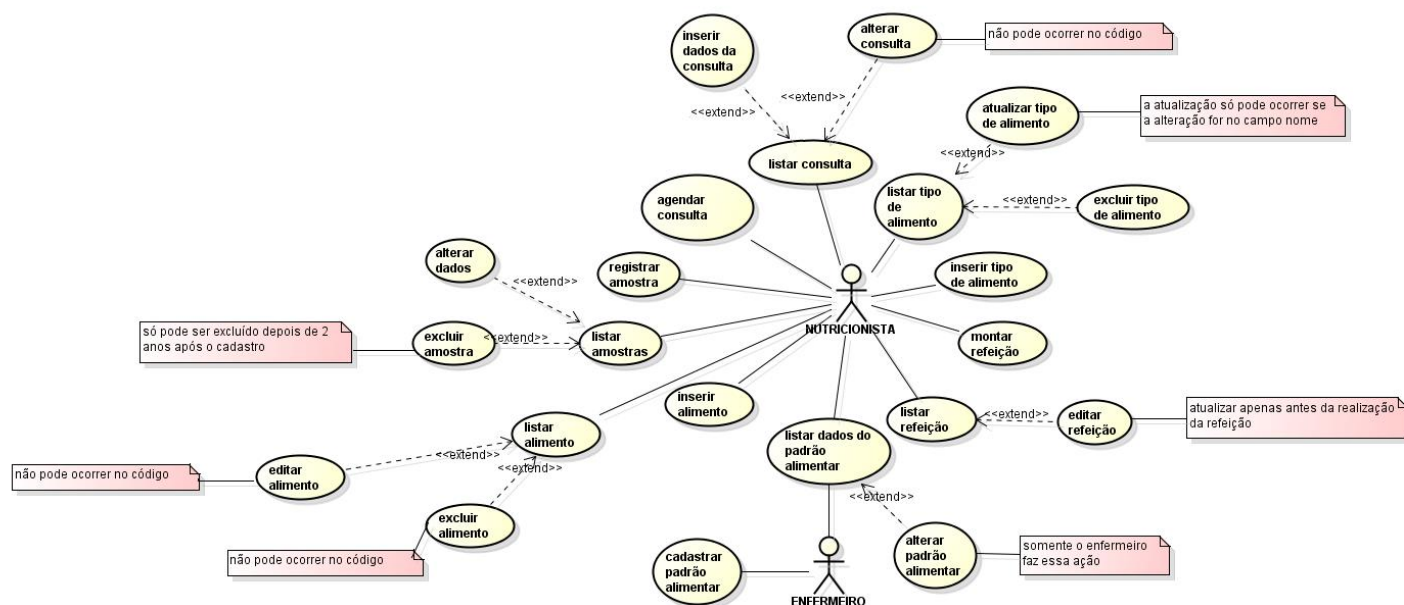
Figura 11: Requisitos RF #01, RF #02, RF #03, RF #04, RF #05, RF #06 e RF #07 do módulo de nutrição

Identificador	Descrição do Requisito
RF #01	O nutricionista irá cadastrar os tipos de alimentos, permitindo que posteriormente ocorra uma listagem onde é permitido alterar o nome e excluir o tipo de alimento se não houver nenhum alimento vinculado a ele.
RF #02	O nutricionista irá cadastrar os alimentos com todas as suas informações, ocorrendo uma listagem onde os dados poderão ser alterados com exceção do código e para serem excluídos não pode ocorrer uma vinculação com o plano alimentar semanal.
RF #03	O nutricionista irá cadastrar um plano alimentar semanal com 6 refeições por dia colocando a data de início e término do período, ocorrendo uma listagem onde só será permitido a alteração antes que ocorra a refeição e com uma semana de antecedência e não é permitido a exclusão de nenhum plano alimentar.
RF #04	O nutricionista irá cadastrar o agendamento da consulta nutricional que ocorrerá uma vez por semana para cada idoso, ocorrendo uma listagem onde só será permitido alterações nos dados da consulta antes que ela seja realizada.
RF #05	O nutricionista irá cadastrar os dados referentes as consultas realizadas juntamente com os dados do idoso onde não será permitido alteração e exclusão.
RF #06	O nutricionista irá cadastrar dados das amostras dos alimentos consumidos em cada refeição dentro da instituição que serão armazenadas durante 7 dias, ocorrendo uma listagem com opção de alterar os dados e só será permitido a exclusão após 2 anos que a amostra foi registrada.
RF #07	Os enfermeiros terão acesso aos dados referentes ao plano alimentar semanal e irão realizar uma cadastro referente a alimentação de cada idoso na instituição, ocorrendo uma listagem onde poderão alterar e excluir as informações, os nutricionista só poderão visualizar a listagem.

Assim sendo, cada uma dessas funcionalidades descritas possui um nível de importância, porém todas elas são fundamentais para a conclusão do projeto.

2.2.2 Diagrama de casos de uso do módulo de nutrição

O módulo de nutrição realizou um diagrama de casos de uso e posteriormente toda a parte documentada especificando as ações realizadas. Dessa forma, cada atividade e relação está sendo descrita permitindo que ocorra um controle de funcionalidades. Abaixo é possível observar os casos de uso criados pelo módulo de nutrição:



Dessa forma, como é notório na imagem acima o nutricionista é o ator principal que realiza todas as ações com exceção apenas do cadastro do padrão alimentar, alteração e exclusão que serão realizadas pelo ator secundário que são os enfermeiros, que poderão visualizar as informações referentes ao plano alimentar semanal.

2.2.3 Elaboração dos casos de teste referentes aos casos de uso do módulo de nutrição

Para cada caso de uso foi necessário realizar um caso de teste onde as ações foram documentadas juntamente com os seus fluxos, dessa forma esse processo se tornou bem extenso, para que posteriormente os testes fossem executados. Abaixo há uma imagem referente ao caso de teste do montar refeição se caracterizando por conter todos os dados da alimentação da semana dentro da instituição, que devido a sua importância e complexidade acabou se tornando a funcionalidade mais difícil de ser criada:

Figura 12: Caso de teste do montar refeição

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #22	Verificar se o nutricionista clica na opção “Salvar” com todos os campos de preenchimento obrigatório preenchidos referentes a montar refeição .	O sistema deve validar e armazenar no banco de dados as informações inseridas do plano alimentar semanal.
CT #23	Verificar se o nutricionista clica na opção “Salvar” com campos de preenchimento obrigatório vazios referentes a montar refeição.	O sistema deve não validar e mostrar a mensagem “Um ou mais campos não foram preenchidos corretamente”.

Dessa maneira, todas as atividades possuem suas características e seus aspectos indispensáveis, e devido ao fato de que entre os requisitos existem dependências, ou seja, cada uma delas têm a sua importância dentro do módulo de nutrição, por consequência é fundamental para o projeto Gerações. Segue abaixo uma imagem do caso de teste do cadastrar de tipo de alimento que contém todas as informações dos tipos de alimentos:

Figura 13: Caso de teste do cadastrar tipo de alimento

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #24	Verificar se o nutricionista clica na opção “Cadastrar” com todos os campos de preenchimento obrigatório preenchidos referentes ao cadastro de tipo de alimento.	O sistema deve validar e armazenar no banco de dados as informações inseridas do tipo de alimento.
CT #25	Verificar se o nutricionista clica na opção “Cadastrar” com campos de preenchimento obrigatório vazios referentes ao cadastro de tipo de alimento.	O sistema deve não validar e mostrar a mensagem “Um ou mais campos não foram preenchidos corretamente”.

Analogamente, pode ser observado abaixo nas imagens os casos de teste referentes ao cadastrar alimento que se caracteriza na inserção dos dados dos alimentos que estão dentro da organização, cadastrar padrão alimentar que insere como ocorre a aceitação dos alimentos pelos idosos e agendar consulta que cadastra as consultas que serão realizadas pelos nutricionistas:

Figura 14: Caso de teste do cadastrar alimento

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #26	Verificar se o nutricionista clica na opção “Cadastrar” com todos os campos de preenchimento obrigatório preenchidos referentes ao cadastro de alimento.	O sistema deve validar e armazenar no banco de dados as informações inseridas do alimento.
CT #27	Verificar se o nutricionista clica na opção “Cadastrar” com campos de preenchimento obrigatório vazios referentes ao cadastro de alimento.	O sistema deve não validar e mostrar a mensagem “Um ou mais campos não foram preenchidos corretamente”.

Figura 15: Caso de teste do cadastrar padrão alimentar

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #32	Verificar se o nutricionista clica na opção “Cadastrar” com todos os campos de preenchimento obrigatório preenchidos referentes ao padrão alimentar.	O sistema deve validar e armazenar no banco de dados as informações inseridas do padrão alimentar.
CT #33	Verificar se o nutricionista clica na opção “Cadastrar” com campos de preenchimento obrigatório vazios referentes ao padrão alimentar.	O sistema deve não validar e mostrar a mensagem “Um ou mais campos não foram preenchidos corretamente”.

Figura 16: Caso de teste do agendar consulta

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #46	Verificar se o nutricionista clica na opção “Enviar” com todos os campos de preenchimento obrigatório preenchidos referentes aos dados da consulta.	O sistema deve validar e armazenar no banco de dados as informações inseridas do dados da consulta.
CT #47	Verificar se o nutricionista clica na opção “Inserir” com campos de preenchimento obrigatório vazios referentes aos dados da consulta.	O sistema deve não validar e mostrar a mensagem “Um ou mais campos não foram preenchidos corretamente”.

2.2.4 Execução dos Casos de Teste do Módulo Nutrição

Após a realização da descrição dos casos de teste do módulo de nutrição os testes foram efetivamente realizados pelos analistas que documentaram cada sucesso e falha da verificação em uma tabela, que continha as datas em que foram efetuadas. Podendo ser observado todas elas realizadas até o momento na imagem a seguir:

Figura 17: Tabela de testes efetuados pelo módulo de nutrição

	02/08/2019	23/08/2019	30/08/2019	06/09/2019	20/09/2019	02/10/2019
CT #24	<u>Sucesso</u>					
CT #25	<u>Sucesso</u>					
CT #34	<u>Sucesso</u>					
CT #35	<u>Sucesso</u>					
CT #26		<u>Sucesso</u>				
CT #27		<u>Sucesso</u>				
CT #36		<u>Sucesso</u>				
CT #37		<u>Sucesso</u>				
CT #06		<u>Sucesso</u>				
CT #07		<u>Sucesso</u>				
CT #15		<u>Sucesso</u>				
CT #16		<u>Sucesso</u>				
CT #17		<u>Falha</u>	<u>Sucesso</u>	<u>Falha</u>	<u>Sucesso</u>	
CT #46		<u>Sucesso</u>				
CT #47		<u>Sucesso</u>				
CT #41		<u>Sucesso</u>				
CT #42		<u>Sucesso</u>				
CT #43		<u>Sucesso</u>				
CT #30		<u>Sucesso</u>				
CT #31		<u>Sucesso</u>				
CT #01		<u>Sucesso</u>		<u>Falha</u>		<u>Sucesso</u>
CT #02		<u>Sucesso</u>		<u>Falha</u>		<u>Sucesso</u>
CT #18			<u>Sucesso</u>			
CT #19			<u>Sucesso</u>			
CT #08			<u>Sucesso</u>			
CT #09			<u>Sucesso</u>			

Posteriormente, foi elaborado uma tabela contendo as informações referentes aos testes que demonstraram erros nas funcionalidades executadas para que esse problema seja corrigido para a entrega de um *software* completo e satisfatório para os seus usuários. Embaixo pode-se observar a tabela de falhas do módulo de nutrição:

Figura 18: Tabela de falhas nos testes efetuados no módulo de nutrição

Casos de Teste	Data da Execução	Responsável da Atividade de Teste	Status do Teste	Motivo da Falha	Responsável pela Correção da Falha	Data Prevista para Novo Teste
CT # 17	23/08/2019	Aline	Falha	Não está ocorrendo a vinculação do tipo de alimento com alimento, permitindo assim que qualquer tipo de alimento seja excluído mesmo que haja a dependência de algum alimento.	Matheus B.	06/09/2019
CT # 17	06/09/2019	Aline	Falha	Não está aparecendo um aviso sobre a impossibilidade da exclusão do tipo de alimento por haver vinculação com alimento.	Matheus B.	20/09/2019
CT # 1	06/09/2019	Isabella	Falha	Os dados não estão sendo alterados.	Arthur A.	20/09/2019
CT # 2	06/09/2019	Isabella	Falha	É inexistente a conclusão dessa funcionalidade.	Arthur A.	20/09/2019

3 Conclusões e Recomendações

Esse trabalho teve o intuito de mostrar o projeto do *software* Gerações, sendo iniciado com o objetivo de criar um sistema de automação para o gerenciamento de instituições de longa permanência, que contou com a colaboração de 56 alunos, que se dividiram em 9 módulos para tornar possível a execução de todos os requisitos referentes ao programa, e 2 professores que participaram como orientados em todos os processos realizados. Por consequência, no decorrer deste documento houve o foco em trazer as etapas de documentação realizadas pelos analistas, e mais especificamente sobre os casos de teste do módulo de nutrição.

A princípio, foi necessário fazer um levantamento da parte teórica que compõe as características dos requisitos, dos casos de uso e dos casos de teste para que após isso cada uma dessas atividades fosse realizada com maior cuidado e domínio do que deveria ser documentado, criando assim um programa melhor estruturado.

Posteriormente, foi necessário a realização do levantamento de requisitos que foram devidamente descritos e considerando o seu grau de relevância, para que somente assim fosse possível trabalhar nas outras etapas do projeto. Logo após, houve a realização dos casos de uso que se caracterizavam em possuir um diagrama com todos os atores e ligações, e também pela parte escrita documentando pontos fundamentais.

Em seguida, a criação das interfaces fez-se relevante realizar testes em cada funcionalidade para que o usuário final não tenha que lidar com erros, ou seja, entregando assim um produto com maior qualidade. Por conseguinte, sendo o foco principal deste documento, tabelas descrevendo cada ação e resposta referentes aos casos de uso do módulo de nutrição tiveram que ser produzidas, os testes realizados foram descritos em uma tabela contendo o *status* de falha ou sucesso e as datas de efetivação, e os erros também possuíram uma tabela contendo informações referentes as suas características.

Analogamente, a cada tarefa realizada se tornou notório o comprometimento dos envolvidos no projeto que possibilitou que todas as etapas fossem produzidas com um número elevado de conclusões satisfatórias. Entretanto, houveram problemas significativos na queda do servidor em todo o ano que se tornou um grande empecilho para a conclusão do *software*, mas que conseguiu ser contornado através do esforço de todos, e a falta de comunicação entre os alunos em alguns aspectos acabou causando pequenas divergências que se solucionaram posteriormente.

Assim sendo, para o próximo ano medidas devem ser tomadas para evitar que os alunos não possam trabalhar no projeto por causa do servidor, e os alunos precisam estabelecer melhores relações para o bem do trabalho em equipe, logo, para que consigam criar um sistema de alta qualidade.

4 Referências Bibliográficas

- [1] IBGE. População, 2010. Disponível em: <<https://cidades.ibge.gov.br/brasil/sp/sao-joao-da-boa-vista/panorama>>. Acesso em: 23/09/2019.
- [2] São João em Notícias, 2017. Disponível em: <http://saojoao.sp.gov.br/home/ler_noticia.php?id=2312>. Acesso em: 23/08/2019.
- [3] Ministério da Saúde, 2005. Disponível em: <file:///C:/Users/aluno/Documents/SVN/documentacao/comum/Documentos%20de%20Instituições%20de%20Longa%20Permanência%20-%20Estudo/04%20-%20RDC_283_2005_COMP.pdf> Acesso em: 20/09/2019.
- [4] Instituto Federal de São Paulo, 2017. Disponível em: <<https://www.ifsp.edu.br/servidores/14-editoria-b/82-missao-visao-e-valores>>. Acesso em: 06/09/2019.
- [5] DEVMEDIA, 2008. Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>>. Acesso em: 08/10/2019.
- [6] Filho W.P.P., 2019. Disponível em: Engenharia de Software Fundamentos, Métodos e Padrões, Terceira Edição. Acesso em: 08/10/2019
- [7] Startup Sorocaba. Disponível em: <<https://startupsorocaba.com/importancia-requisitos-de-software/>>. Acesso em: 09/10/2019.
- [8] Disponível em: <http://www.hectordufau.com.br/openup/openup/guidances/concepts/use_case_model_CD178_AF9.html>. Acesso em: 08/10/2019.
- [9] Disponível em: <<https://sites.google.com/site/projetopsifitness/casos-de-uso>>. Acesso em: 08/10/2019.
- [10] Qconcursos.com, 2016. Disponível em: <<https://www.qconcursos.com/questoes-de-concursos/questoes/2e50ad65-91>>. Acesso em: 08/10/2019.
- [11] SlideShare, 2010. Disponível em: <<https://pt.slideshare.net/darrochella/sisdesk-sistema-de-servicedesk-voltado-para-desenvolvimento-de-software>>. Acesso em: 08/10/2019.
- [12] Molinari L., 2014. Disponível em: Inovação e Automação de Testes de Software. Acesso em: 08/10/2019.
- [13] Delamaro M. E., Maldonado J. C., Jino M., 2007. Disponível em: Introdução ao Teste de Software. Acesso em: 15/10/2019.
- [14] OneDayTestingBlog. Disponível em: <<https://blog.onedaytesting.com.br/teste-de-software/>>. Acesso em: 15/10/2019.