

**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

Documentação e realização dos casos de teste do modulo 2

Aluno: Filipe Henrique Muniz

Prontuário: BV:1620436

São João da Boa Vista – SP

2019

Resumo

A ideia de se desenvolver um software pode envolver muito mais etapas e caminhos lógicos do que se imagina tão comumente. Por trás de todo o desenvolvimento torna-se evidente a importância do planejamento prévio, planejamento este que definirá ou o sucesso da empreitada ou seu terrível fracasso em meio ao caos. Com isso em mente torna-se indispensável um documento que consiga definir a priori como o projeto deve ser e criar com isso um roteiro rígido a ser seguido pelo desenvolvedor, garantindo assim um projeto entregue com sua totalidade de requisitos cumpridos. Para a resolução deste problema foi criado o chamado documento de caso de teste, objeto de nosso estudo nesta pesquisa. Dentro deste contexto demonstraremos neste estudo sua importância e as consequências de uma má aplicação dos casos de teste, além disso, será demonstrado a correlação dos casos de teste com todos os documentos desenvolvidos até o final do projeto Gerações tendo as duas ligações mais fortes os documentos de caso de uso e o de levantamento de requisitos.

Palavras-chave: Planejamento, Software, Caso de teste.

SUMÁRIO

1	Introdução	6
1.1	Contextualização / Motivação	6
1.2	Objetivo Geral da Pesquisa	8
1.3	Objetivos Específicos	8
1.4	Estrutura do Documento.....	8
2	Desenvolvimento	9
2.1	Levantamento Bibliográfico.....	9
2.2	Engenharia de Software.....	9
2.3	Casos de teste e sua importância.	9
2.4	Tipos de teste.....	10
2.5	Desenvolvimento dos objetivos específicos.....	11
2.5.1	Etapas para o Desenvolvimento da Pesquisa.....	11
2.5.2	Diagrama de Casos do Uso do Módulo 02	11
2.5.3	Projetar os Casos de Teste do Módulo 02	13
2.5.4	Executar os Casos de Teste do Módulo 02.....	17
3	Conclusões e Recomendações	19
4	Referências Bibliográficas	20

LISTA DE FIGURAS

Figura 1: Posição de São João entre as melhores pequenas cidades para envelhecer [3]	6
Figura 2: Instituto Federal do município de São João da Boa Vista	7
Figura 3 : Diferentes interpretações ao longo do ciclo de desenvolvimento de um software	10
Figura 4- Diagrama caso de uso Módulo 2	12
Figura 5-Protótipo caso de uso	13
Figura 6 Testes execução	17
Figura 7 Falhas histórico caso de teste	18

LISTA DE TABELAS

Tabela 1-Levantamento de requisitos	12
Tabela 2 Tabela caso de uso.....	14
Tabela 3 Caso de teste.....	16

1 Introdução

1.1 Contextualização / Motivação

O estado de São Paulo oferece à sua população uma das melhores condições de vida do Brasil. Isso reflete no aumento dos indicadores de longevidade e escolaridade dos municípios paulistas [1]. Nesse contexto, insere-se a cidade de São João da Boa Vista localizada no interior do estado, com uma área de 517,47 km² e um Índice de Desenvolvimento Humano (IDHM) de 0,797, em 2010, classificado como alto, correspondendo as estatísticas estaduais. Desta forma, entre 2000 e 2010 a população local aumentou cerca de 0,78%, o que totalizou, segundo o Instituto Brasileiro de Geografia e Estatística, 83 639 habitantes. Neste mesmo período de tempo, a taxa de envelhecimento passou de 8,41% para 10,43%, resultado da boa qualidade de vida oferecida pelo município [2].

Em princípio, baseado nas condições de vida oferecidas à população, São João da Boa Vista foi considerada uma das 40 melhores pequenas cidades para envelhecer. O aumento da expectativa de vida fez com que algumas mudanças fossem realizadas para atender a população na terceira idade. Dessa maneira, utilizando sete variáveis para essa classificação, Instituto de Longevidade Mongeral Aegon em parceria com a Fundação Getúlio Vargas realizou esta pesquisa entre 348 municípios. Entre essas variáveis estão Cuidados de Saúde; Bem-Estar; Finanças; Habitação; Educação e Trabalho e Cultura e Engajamento, que receberam pesos com base nas principais necessidades da população na terceira idade[10].

Cidade	Posição	Nota
São João da Boa Vista, SP	1	99.82
Vinhedo, SP	2	99.49
Lins, SP	3	96.61
Fernandópolis, SP	4	95.70
Tupã, SP	5	94.78

Figura 1: Posição de São João entre as melhores pequenas cidades para envelhecer [3]

Na cidade, existem 13 unidades de saúde distribuídas pelos bairros, 2 hospitais e uma Unidade de Pronto Atendimento (UPA). Além disso, o município está construindo um centro para acolher idosos em situação de vulnerabilidade social [3].

Ademais, as Instituições de Longa Permanência, que abrigam idosos de 60 anos ou mais, visam garantir ao morador situações de convivência e cuidados que, por vezes, não pode ser oferecido pela família, seja por falta de condições ou quebra de vínculos familiares [4]. As organizações existentes no município necessitam de um sistema para gerenciar seu dia-a-dia, para contribuir com o trabalho dos enfermeiros, gestores, dos internos e demais funcionários.

Outrossim, o Instituto Federal de São Paulo, presente na localidade desde 2007, oferta cursos técnicos, tecnólogos, graduações e pós-graduações. Voltado à ciência e tecnologia, oferece, dentro dos cursos técnicos, os integrados ao ensino médio, tanto em informática quanto em eletrônica. Em relação ao curso de informática, que possui ao todo 4 anos de estudo, visa formar cidadãos que se preocupem com as relações sociais e humanas, além de profissionais capacitados a atender os diversos ramos da tecnologia da informação. O Curso está organizado em núcleo comum e áreas técnicas, onde no primeiro efetiva-se o ensino voltado ao ensino médio do estudante, e no segundo preocupa-se com a formação técnica do aluno [5].



Figura 2: Instituto Federal do município de São João da Boa Vista

No último ano, a matéria de Prática de Desenvolvimento de Sistemas, ministrada em 2019 pelos professores Breno Lisi Romano e Luiz Angelo Valota Francisco, busca reunir os conhecimentos obtidos previamente para a realização de um software, voltado principalmente às necessidades

sociais da comunidade externa. Nesse ano, o contexto do projeto é voltado às Instituições de Longa Permanência, importantes para a cidade por conta do crescimento da população idosa, nomeado Gerações, possuindo 09 módulos responsáveis por cada questão do sistema. Entre eles, está o módulo 02, responsável pelo prontuário dos idosos, possuindo dois analistas, dois DBA's e dois desenvolvedores. Dentro desse módulo, foram realizadas diversas documentações, entre elas o documento de casos de uso do módulo, que será abordado nessa pesquisa.

1.2 Objetivo Geral da Pesquisa

O objetivo geral desta pesquisa é mostrar o processo de elaboração dos casos de teste do Módulo 02 – Prontuário dos Idosos, assim como a sua importância para o projeto Gerações de modo geral.

1.3 Objetivos Específicos

- Etapas para o Desenvolvimento da Pesquisa.
- Diagrama de Casos do Uso do Módulo 02
- Projetar os Casos de Teste do Módulo 02
- Executar os Casos de Teste do Módulo 02

1.4 Estrutura do Documento

No primeiro capítulo está contida a contextualização desse trabalho, assim como sua motivação. Em sequência, estão o objetivo geral e o específico da pesquisa, a fim de introduzir as informações a serem exploradas.

No segundo capítulo será demonstrada a forma como foi realizada a documentação dos casos de teste, assim como a sua importância.

No terceiro capítulo está inserida a conclusão e recomendações, com base nas pesquisas realizadas a fim de explorar os casos de teste como um todo.

No quarto capítulo encontram-se as referências bibliográficas utilizadas para a realização deste documento.

2 Desenvolvimento

2.1 Levantamento Bibliográfico

2.2 Engenharia de Software

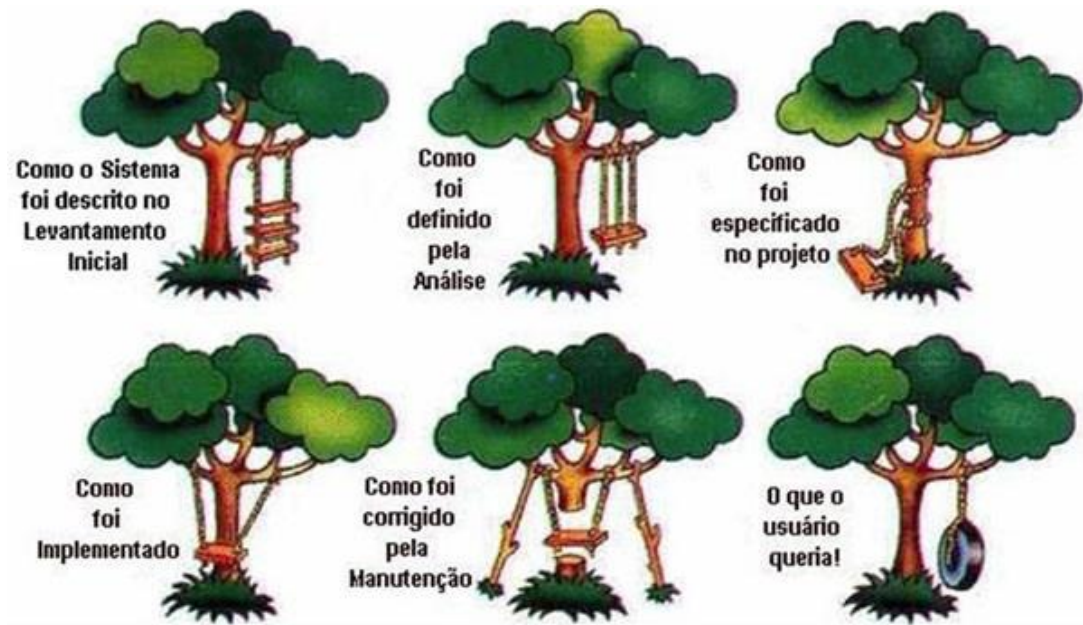
A Engenharia de Software é uma disciplina de engenharia relacionada a todos os aspectos de produção de software desde a especificação até a manutenção. O engenheiro de software aplica teorias métodos e ferramentas onde for apropriado e de forma seletiva mesmo quando não existem teorias e métodos aplicáveis, e fazem as coisas funcionarem. Trabalham também sobre restrições organizacionais e financeiras. Em geral utiliza-se uma abordagem sistemática e organizada, mas pode ser extremamente eficaz selecionar uma abordagem alternativa e menos formal para uma determinada circunstância. [6]

2.3 Casos de teste e sua importância.

Testar um software é o processo de execução de um produto para determinar se ele conseguiu seguir suas especificações e funcionou corretamente no ambiente pré-estabelecido a o qual foi projetado. O seu objetivo é revelar erros em um projeto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final. [7]

Ao se desenvolver um projeto com uma grande quantidade de pessoas e com isso diversas interpretações e ainda por se tratar de um software relativamente extenso torna-se mais provável a frequência de defeitos e falhas fugindo do padrão especificado, tornando com isso a realização de testes ainda mais importante no projeto gerações.

Figura 3: Diferentes interpretações ao longo do ciclo de desenvolvimento de um software



2.4 Tipos de teste

Duas das técnicas de teste mais comuns são a Técnica estrutural conhecida também como teste caixa branca ou Teste funcional comumente conhecido com teste de caixa preta. Iniciaremos nosso estudo falando sobre os Testes funcionais.

Teste Funcional: Técnica de teste em que o componente de software a ser testado é abordado como se fosse uma caixa-preta, ou seja, não se considera o comportamento interno do mesmo. Dados de entrada são fornecidos, o teste é executado e o resultado obtido é comparado a um resultado esperado previamente conhecido. Haverá sucesso no teste se o resultado obtido for igual ao resultado esperado. [6]

Técnica estrutural: Técnica de teste que avalia o comportamento interno do componente de software. Essa técnica trabalha diretamente sobre o código fonte do componente de software para avaliar aspectos tais como: teste de condição, teste de fluxo de dados, teste de ciclos e teste de caminhos lógicos [8]. Nesse tipo de técnica o testador tem acesso ao código fonte.

2.5 Desenvolvimento dos objetivos específicos

2.5.1 Etapas para o Desenvolvimento da Pesquisa.

Para a realização desta pesquisa foram realizados inúmeros passos que a precederam fazendo com que cada parte dela seja estruturada de maneira constante e bem definida. Podemos definir as etapas da formulação deste estudo como:

- Contextualização do projeto e sua motivação: Nesta fase foram colocado em perspectiva o local onde o projeto será implementado e suas características populacionais/socioeconômicas, além de ser lembrado a motivação inicial desta empreitada que é implementar um sistema para casas de longa permanência auxiliando o administrador e melhorando a vida dos idosos que lá vivem.
- Definição de objetivos: Demonstrar a área do projeto em que irei focar neste estudo e definir metas para a exibição com sucesso dos casos de teste do modulo 2.
- Referencias bibliográficas: Aqui reunimos conhecimento de grandes autores para o fortalecimento de nosso desenvolvimento, fazendo com que o trabalho carregue com ele uma bagagem teórica bem pautada.
- Desenvolvimento: Nesta etapa serão incrementados os conhecimentos retirados da pesquisa bibliográfica formulando de maneira concreta o desenvolvimento de ideias envolvendo casos de teste.
- Conclusão: Síntese dos pontos vitais para a pesquisa.

2.5.2 Diagrama de Casos do Uso do Módulo 02

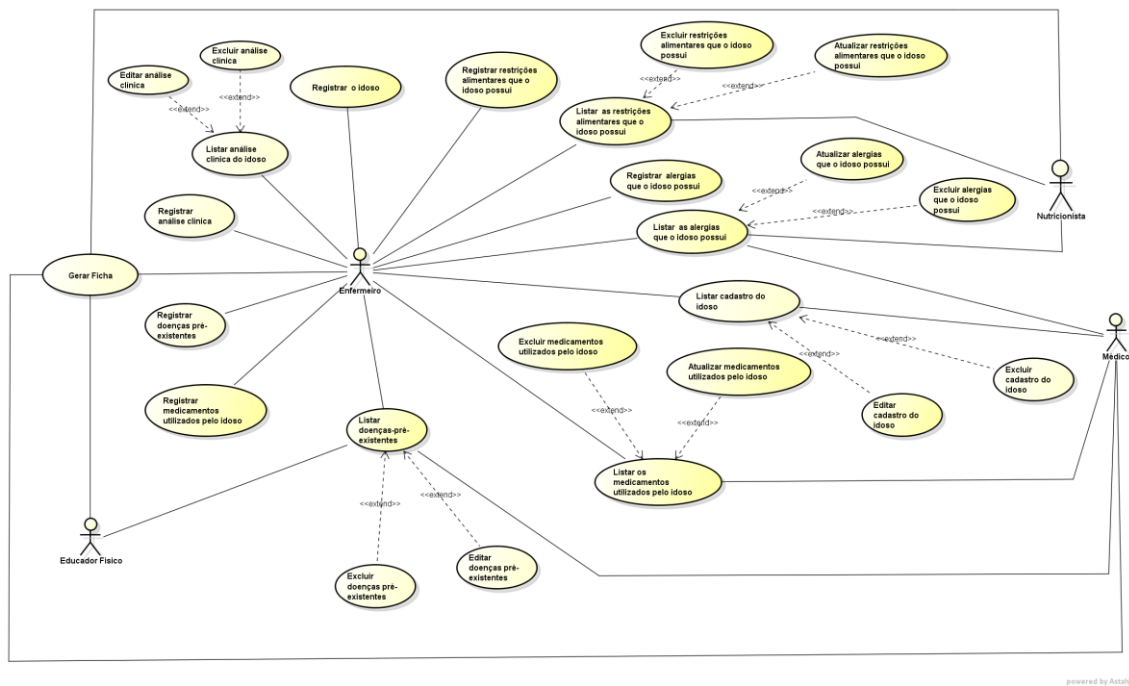
Para a formulação de casos de testes e seus protótipos se faz necessário a priori a observação dos casos de uso. Em suma a observação dos atores e suas funções dentro do projeto são essenciais para a criação de um sistema logico e bem estruturado. Para a melhor organização e visualização dos atores foram realizados diagramas facilitando, por exemplo, a formulação do banco de dados.

Diagramas de caso de uso podem ser explicados da seguinte maneira:

- Cenário: Sucessão de interações entre um usuário e o sistema.
- Ator: Aquele que ira utilizar o sistema seja este humano ou um sistema, representado por a figura de um boneco palito.

- Caso de uso: Funcionalidade realizada pelo ator podendo esta ser concreta quando feita diretamente pelo ator ou abstrata quando é transmitida através de outro caso de uso, geralmente representado por um eclipse.
- Relacionamento: Aquele que fornece sentido as ligações trazendo uma descrição dos casos de uso podendo ser relacionamentos por associação, inclusão, extensão ou herança, geralmente representado por uma seta ou linha dependendo do relacionamento.

Figura 4- Diagrama caso de uso Módulo 2



Após a formulação destes diagramas fica muito mais simples a visualização dos requisitos levantados que iriam ditar todo o funcionamento do módulo e consequentemente o sucesso do projeto.

Tabela 1-Levantamento de requisitos

Identificador	Descrição do Requisito
RF #01	O sistema deve gerar prontuários automaticamente gerando uma sequência numérica pré-definida, começando pelo número 01 para o cadastro do idoso. O prontuário deve aparecer em cima da tela de cadastro, se cancelarem o cadastro, esse prontuário não será atribuído a nenhum idoso.

2.5.3 Projetar os Casos de Teste do Módulo 02

Formular casos de teste sem duvida não configura uma tarefa fácil, tendo em vista o grande numero de rumos que um projeto pode tomar. Contudo é papel daquele que elabora este documento se manter sempre focado no objetivo do projeto e formular seus testes de maneira mais coerente e logica possível. Na elaboração dos casos de teste do módulo 2 recebeu foco os documentos antes formulados e suas especificações, fazendo com que os casos de teste seguissem o máximo possível suas indicações. Além disso, também foram consultadas as opiniões dos desenvolvedores que iriam efetuar a conversão dos casos de teste em código fonte, para que com isso pudesse estabelecer aquilo que era ou não possível realizar deixando o documento de testes cada vez mais completo e robusto em sua formação. O documento de caso de teste do modulo 2 conta com 62 casos de teste que configuram o desenvolvimento das funcionalidades do modulo de prontuário de idosos e sua aplicação com grande êxito. Vale ressaltar a conversão dos casos de uso em casos de teste que deviam ser feitas de uma maneira que conseguisse manter o máximo da integridade original dos protótipos previamente feitos.

Figura 5-Protótipo caso de uso

The screenshot displays a web application interface for 'Gerações'. The top navigation bar includes a search bar and user profile icons. The left sidebar lists various administrative and medical functions. The main content area is titled 'Painel Administrativo / Gerenciamento de Medicamentos' and contains a 'Cadastro de Medicamentos' section with a 'Cadastrar Medicamento' button. Below this is a table of medications with columns for 'Medicamento', 'Dosagem', 'Vezes ao dia', and 'Horário'. The table shows one entry for 'Saxagliptina' with a dosage of '5mg' and frequency of '2' times a day at '14h/20h'. The interface also includes pagination controls and a footer with copyright information.

Gerações

Search for...

Painel Administrativo / Gerenciamento de Medicamentos

Cadastro de Medicamentos

Cadastrar Medicamento

Tabela de Medicamentos

Show 10 entries Search:

Medicamento	Dosagem	Vezes ao dia	Horário
Saxagliptina	5mg	2	14h/20h

Showing 1 to 1 of 1 entries Previous 1 Next

Atualizado 10/05/2019

©2019 Todos os Direitos Reservados: PDS 2019

Abaixo se encontra um exemplo de caso de uso e seu respectivo caso de teste.

Tabela 2 Tabela caso de uso

Nome do Caso de Uso: Cadastro do idoso	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre no primeiro acesso ao “Cadastro do idoso”.
Ator Principal:	Enfermeiro
Pré-Condição:	Estar logado no sistema.
Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Selecionar a opção “Cadastro do idoso” no menu inicial “Meu idoso”.	<p>2. Exibir uma tela de cadastro contendo os campos a serem preenchidos:</p> <ol style="list-style-type: none"> 1. Nome Completo (varchar 255) – Obrigatório 2. RG (varchar 9) – Obrigatório 3. CPF (char 12 XXX.XXX.XXX-XX) – Obrigatório – deve ser validado 4. Sexo (M ou F) – Obrigatório 5. Data de nascimento (DD/MM/AAAA) – Obrigatório 6. Estado do idoso (Ativo/Inativo), inicialmente ativo – Obrigatório 7. Fralda ou <u>forrinho</u> (S/N) - Obrigatório 8. Código do responsável (O código deve ser pesquisado a partir do nome do responsável e/ou cpf, se o responsável for cadastrado, será selecionado e será designado ao idoso em questão. Se não houver responsável cadastrado (Botão “Não tenho responsável” (direcionar para a funcionalidade do Módulo 01), o site deverá ir para a página de cadastro de responsável) – obrigatório, não pode ser alterado. <p>O preenchimento dos campos é obrigatório.</p>
3. Preencher todos os campos e selecionar a opção “Salvar”, abaixo dos campos de cadastro.	4. Validar os campos preenchidos, confirmando o cadastro com o botão “cadastrar” e registrando no banco de dados.
5. Fim do Caso de Uso.	
Fluxo Alternativo A: Não preenchimento dos campos obrigatórios	

3. Não preencher todos os campos e selecionar a opção “Cadastrar”, abaixo dos campos de cadastro.	4. Não irá validar o cadastro, exibindo a mensagem “Um ou mais campos não foram preenchidos corretamente”.
	5. Retornar à ação 2 do Fluxo Principal.

Protótipo de Interface Homem-Máquina:

Cadastrar idoso

Nome completo:

RG:

CPF:

Sexo:

☐ Feminino ☐ Masculino

Data de nascimento:

Estado do idoso:

Fralda:

☐ Sim ☐ Não

Forrinho:

☐ Sim ☐ Não

Código do responsável:

[Não possui responsável](#)

Foto:



[Adicionar](#)

[Cadastrar](#)

Tabela 3 Caso de teste

Casos de Teste	Descrição do Caso de Teste	Resultado Esperado
CT #01	Verifica se ao cadastrar um novo idoso o sistema gerou um novo prontuário ao clicar no botão “cadastrar idoso”.	O sistema deve gerar um novo prontuário para começar o cadastro do idoso
CT #02	Verifica se o usuário já é previamente cadastrado	O sistema deve procurar no banco de dados e verificar se já existe um idoso com o CPF cadastrado.
CT #03	Verifica se todos os campos foram preenchidos.	Caso o usuário não preencha os itens obrigatórios, não permitir que a ação seja finalizada e exibir a mensagem “Preencha este campo!”.
CT #04	Verifica se o código do responsável existe no cadastro do idoso	O sistema deve procurar no banco de dados e verificar se existe o código do responsável para o respectivo idoso.
CT #05	Validação do CPF.	O sistema deve verificar se o CPF é valido, conforme os parâmetros pré-estabelecidos, se não exibir mensagem “Informe um CPF valido”, se o CPF for valido, dar continuidade no fluxo normal do sistema.

Como pode ser notado o caso de uso inicial se transformou em 5 casos de testes divididos e orientados as necessidades e resultados esperados. Ainda podemos visualizar que os casos de teste obtiveram êxito em seguir com alto rigor as especificações passadas pelo caso de uso tendo em seu conteúdo nenhuma ou pequena alteração, mantendo sempre a essência do que fora planejado.

2.5.4 Executar os Casos de Teste do Módulo 02

Após todas as etapas de documentação e planejamento que ocorreram durante todo projeto Gerações chegamos a uma de suas etapas finais, a execução dos casos de teste. Para que chegássemos até aqui passamos por diversas especificações e por este motivo a execução dos testes devem estar em pleno acordo com os casos de testes e os documentos anteriores a ele como, por exemplo, o de caso de uso. Contudo foi criada uma serie de etapas na execução do teste, garantindo assim minimizar os erros e seguir o planejamento. Podemos definir as fases para a execução do teste como:

- Comparação das tabelas de caso de teste com o código fonte apresentado.
- Comparação visual do sistema com aquilo que era esperado nos casos de teste.
- Registro do sucesso ou falha de cada caso de teste analisado.
- Caso ocorra o erro definição de um novo prazo para a nova realização de testes.

Utilizando as etapas detalhadas acima torna se possível prever vários erros e alterar cenário que seriam problemáticos ao usuário comum. Abaixo um exemplo da tabela de planejamento e suas consequentes falhas e sucessos.

Figura 6 Testes execução

3. Testes Executados

a) Planejamento / Acompanhamento da Execução dos Testes

	02/08/2019	23/08/2019	06/09/2019	06/09/2019	20/09/2019	02/10/2019			
CT #01	Sucesso								
CT #02	Falha	Sucesso							
CT #03	Sucesso								
CT #04			Falha	Sucesso					
CT #05	Falha	Sucesso							
CT #06			Sucesso						
CT #07			Sucesso						
CT #08			Falha	Sucesso					
CT #09			Sucesso						
CT #10			Sucesso						
CT #11			Sucesso						
CT #12			Sucesso						
CT #13			Sucesso						
CT #14			Falha	Sucesso					
CT #15				Sucesso					
CT #16				Sucesso					
CT #17		Sucesso							
CT #18		Sucesso							
CT #19		Falha		Sucesso					
CT #20		Falha		Sucesso					
CT #21		Sucesso							
CT #22		Sucesso							
CT #23		Sucesso							
CT #24									
CT #25									

Temos ainda em caso de falha ou incoerência com o documento de testes um histórico de falhas que formam um registro facilitando tanto para o analista quanto para o desenvolvedor em um futuro erro ocasionado por uma correção não feita anteriormente.

Figura 7 Falhas histórico caso de teste

4. Histórico da Execução dos Testes

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#02	02/08/2019	Filipe Muniz	Falha	A verificação da existência do mesmo idoso não está sendo realizada.	Verônica	09/08/2019
CT#05	02/08/2019	Filipe Muniz	Falha	As funcionalidades através de JavaScript ainda não foram realizadas para a validação do CPF.	Pedro	09/08/2019
CT#04	06/09/2019	Filipe Muniz	Falha	Ao clicar no botão confirmar, não é realizada a edição como previsto no caso de teste 4.	Verônica	06/09/2019
CT#08	06/09/2019	Beatriz Basilio	Falha	O sistema confirma o registro mesmo se tiver caixas de texto em branco.	Pedro	06/09/2019
CT#14	06/09/2019	Beatriz Basilio	Falha	O sistema não edita/adiciona um novo medicamento.	Pedro	06/09/2019
CT#19	06/09/2019	Filipe Muniz	Falha	O sistema não realiza a edição dos dados do idoso.	Verônica	06/09/2019
CT#20	06/09/2019	Filipe Muniz	Falha	O sistema não verifica se a edição está sendo realizada de forma incorreta.	Verônica	06/09/2019

3 Conclusões e Recomendações

Em suma o projeto Gerações assim como outros desenvolvidos pelos 4º anos de informática segue esta voltado a causas de cunho social ligadas principalmente a população de São João da Boa Vista, com isso nada melhor do que falar de um tema tão presente em nossa cidade como o bem estar dos cidadãos de terceira idade. Toda a equipe deste projeto caracterizam 58 pessoas diretamente envolvidas, além é claro das outras muitas que participaram de maneiras indiretas e que ajudaram este projeto a ser concluído de diversas formas. Sendo assim o objetivo deste estudo é demonstrar apenas uma parte desta grande empreitada que são os casos de teste do módulo 2. Quando se fala de caso de teste instintivamente deve se lembrar de que esta é uma das ultimas fases de desenvolvimento de qualquer software e com isso apenas para chegar nesta etapa já se considera sucesso em uma grande parte do sistema. A primeira parte para formulação dos casos de teste é sem duvida a reflexão perante documentos posteriores e a avaliação por parte do Analista daquilo que é indispensável para o bom funcionamento do software, a partir disso ele começa a avaliar os casos de uso e seus protótipos transferindo eles para a realidade e o sabatinando com diversos cenários distintos. Como segundo passo o responsável pelos testes deve utiliza os dados recolhidos na primeira etapa e formular uma abordagem para os casos de teste que deveram sempre seguir uma modelo logico preciso, garantindo que com isso não ocorram falhas mais a frente. E como terceiro e ultima fase serão realizados os testes propriamente ditos, que deverão garantir a integridade dos raciocínios efetuados nos dois passos anteriores, estando sempre atendo em que rumo esta tomando o projeto podendo sempre corrigir sua direção para o caminho correto.

Destacando como ponto negativo temos a falta de contextualização de cada documento, com isso não se torna claro a maneira como aquele arquivo ira ser utilizado no futuro dificultando seu planejamento. Para experiências futuras se faz recomendável a maior integração das outras partes do projeto no desenvolvimento de documentos primordiais como caso de uso e caso de teste, tendo em vista sua complexidade. Enfim após todas essas considerações pode se dizer que o projeto Gerações atingiu o sucesso visto seu alto percentual de progresso em todos os módulos, fazendo assim que o sistema desenvolvido possa ser aplicado e constitua melhoras em casas de longa permanência como fora previsto em seu planejamento inicial.

4 Referências Bibliográficas

- [1] Qualidade de vida estado de São Paulo. **INVESTESP**, 2018. Disponível em: <<https://www.investe.sp.gov.br/por-que-sp/qualidade-de-vida/>> Acesso em: 08 de outubro de 2019.
- [2] Caracterização do território. **ATLAS**, 2013. Disponível em: <http://www.atlasbrasil.org.br/2013/pt/perfil_m/sao-joao-da-boa-vista_sp#demografia> Acesso em: 08 de outubro de 2019.
- [3] Melhores cidades para viver. **EXAME**, 2017. Disponível em: <<https://exame.abril.com.br/brasil/as-40-melhores-pequenas-cidades-para-envelhecer/>> Acesso em: 08 de outubro de 2019.
- [4] Instituições de longa permanência SP. **CAPITALSP**, 2019. Disponível em: <<http://www.capital.sp.gov.br/cidadao/familia-e-assistencia-social/centros-de-acolhida/centros-de-acolhida-especial/instituicao-de-longa-permanencia-para-idoso-ilpi>> Acesso em: 08 de outubro de 2019.
- [5] Sobre o campus IFSP. **IFSPSBV**, 2019. Disponível em: <<https://www.sbv.ifsp.edu.br/sobre-campus>> Acesso em: 01 de novembro de 2019.
- [6] Engenharia de software. **SOMMERVILLE, IAN**, 2011. Disponível em: <<https://www.di.ubi.pt/~sebastiao/Ensino/UBI/2017->>> Acesso em: 29 de outubro de 2019.
- [7] Introdução a Teste de software. **NETO, ARLINDO**, 2015. Disponível em: <https://www.researchgate.net/profile/Arilo_Neto/publication/266356473_Introducao_a_Testes_de_Software/links/5554ee6408ae6fd2d821ba3a/Introducao-a-Teste-de-Software.pdf> Acesso em: 29 de outubro de 2019.

[8] Roger S. **Pressman**; Bruce R. Maxim. **Engenharia De Software Uma Abordagem Profissional** - 2016. Disponível em: <https://books.google.com/books?hl=pt-BR&lr=&id=wexzCwAAQBAJ&oi=fnd&pg=PR1&dq=requisitos+de+software&ots=0N-LoJJA50&sig=khkvND0RTjXhiXfHzWaIWEiPGwE> Acesso em 29 de outubro de 2019

[9] SUMMERVILLE, Ian. **Engenharia de Software**. 9ª ed. São Paulo: Person Prentice Hall, 2011.

[10] **Pesquisa Aponta São João da Boa Vista Como Melhor Cidade para Idosos**. G1 – 2017. Disponível em: <http://g1.globo.com/sp/sao-carlos-regiao/noticia/2017/03/pesquisa-aponta-sao-joao-da-boa-vista-como-melhor-cidade-para-idosos.html>. Acesso em 28 de setembro.2019.