

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

**Implementação e Análise de um Método de
Priorização de Testes Baseado em Análise de
Risco: Um Estudo de Caso**

Autor: Philippe Rosa Serafim
Orientadora: Dra. Elaine Venson

Brasília, DF
2024



Philippe Rosa Serafim

Implementação e Análise de um Método de Priorização de Testes Baseado em Análise de Risco: Um Estudo de Caso

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dra. Elaine Venson

Brasília, DF

2024

Philippe Rosa Serafim

Implementação e Análise de um Método de Priorização de Testes Baseado em Análise de Risco: Um Estudo de Caso/ Philippe Rosa Serafim. – Brasília, DF, 2024-65 p. : il. (algumas color.) ; 30 cm.

Orientador: Dra. Elaine Venson

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2024.

1. Priorização de Testes de software. 2. Qualidade de Software. I. Dra. Elaine Venson. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Implementação e Análise de um Método de Priorização de Testes Baseado em Análise de Risco: Um Estudo de Caso

Philippe Rosa Serafim

Implementação e Análise de um Método de Priorização de Testes Baseado em Análise de Risco: Um Estudo de Caso

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 13 de setembro de 2024:

Dra. Elaine Venson
Orientadora

Dra. Cristiane Soares Ramos
Convidado 1

Dr. Ricardo Ajax Dias Kosloski
Convidado 2

Brasília, DF
2024

Agradecimentos

Concluir este Trabalho de Conclusão de Curso representa o culminar de anos de esforço, aprendizado e crescimento. Durante essa jornada, muitas pessoas estiveram ao meu lado, oferecendo apoio, orientação e inspiração. É com imensa gratidão que dedico algumas palavras a essas pessoas, que foram fundamentais para que eu pudesse chegar até aqui.

Primeiramente, gostaria de expressar minha profunda gratidão à minha mãe, Thelma, por todo o apoio e torcida incansável ao longo de toda a minha jornada acadêmica. Sua força e dedicação foram uma fonte constante de inspiração e motivação.

Ao meu namorado, Pedro Ruela, agradeço por todo o apoio emocional e pelos sábios conselhos que sempre me incentivaram a seguir em frente, mesmo nos momentos mais desafiadores. Sua presença foi essencial para que eu pudesse manter o equilíbrio e a determinação.

Aos meus amigos, Alexandre, Emily, Écio, Ithalo, Livia e João Gomes, sou extremamente grato pelos momentos e ensinamentos que compartilhamos. Vocês tornaram essa jornada mais leve e significativa, e levo cada lembrança com muito carinho.

À família que pude escolher, Damaso Junior, Edvan Junior, Letícia Meneses, Lety-cia Pedroza e Pedro Lima, meu mais sincero agradecimento por todo o apoio emocional e pelos momentos vividos durante a graduação. Vocês foram uma verdadeira fortaleza em minha vida.

Agradeço, de forma especial, à minha orientadora, Elaine Venson, por sua paciência e orientação ao longo de todo o processo de elaboração deste trabalho.

A todos que, de alguma forma, contribuíram para esta conquista, meu mais profundo agradecimento.

Minha doce *Perdrinha*, recebi com emoção
Suas palavras, que tocaram meu coração
Aceito com alegria, sem qualquer hesitação
Cuidaremos desse jardim, com muito amor e dedicação

Caminhando lado a lado, em plena harmonia
Nossa vida será sempre, uma bela sinfonia
Entre sonhos e verdades, a vida se alinha
No nosso caminho, serás sempre minha *Perdrinha*

Resumo

A crescente demanda pela qualidade em produtos de software torna essencial e obrigatória a realização de testes, embora o processo de teste de software possa se tornar oneroso e demorado, podendo representar até 50% do custo total do projeto. Para mitigar esses desafios, a automação de testes tem sido amplamente adotada, apesar de trazer consigo novos custos e desafios que precisam ser gerenciados de forma eficiente. Neste contexto, este Trabalho de Conclusão de Curso tem como objetivo principal executar e analisar um método de priorização de casos testes em um projeto real, especificamente no projeto MEC-Energia, um sistema de recomendação de energia elétrica para instituições de ensino superior. O método de priorização de testes aplicado baseia-se na análise de risco dos métodos do sistema, levando em consideração fatores como a complexidade, o tamanho dos métodos e os requisitos alterados. A metodologia do estudo é qualitativa, com uma abordagem exploratória e descritiva, sendo conduzida através de um estudo de caso único. Os resultados obtidos fornecem uma análise detalhada sobre a eficácia do método de priorização aplicado, contribuindo para a redução dos custos e esforços associados aos testes de software, sem comprometer a qualidade do produto final.

Palavras-chave: Casos de testes; Priorização de casos de testes; Qualidade de software.

Abstract

The increasing demand for quality in software products makes testing essential and mandatory, although the software testing process can become costly and time-consuming, potentially representing up to 50% of the total project cost. To mitigate these challenges, test automation has been widely adopted, despite bringing new costs and challenges that need to be efficiently managed. In this context, this Undergraduate Thesis aims to implement and analyze a test case prioritization method in a real project, specifically in the MEC-Energia project, a recommendation system for electrical energy for higher education institutions. The applied test prioritization method is based on risk analysis of the system's methods, taking into account factors such as complexity, method size, and modified requirements. The study methodology is qualitative, with an exploratory and descriptive approach, conducted through a single case study. The results provide a detailed analysis of the effectiveness of the applied prioritization method, contributing to the reduction of costs and efforts associated with software testing, without compromising the quality of the final product.

Key-words: Casos de testes; Priorização de casos de testes; Qualidade de software.

Lista de ilustrações

Figura 1 – Etapas do Modelo de Priorização de Testes	28
--	----

Lista de tabelas

Tabela 1 – Análise de Requisitos Alterados	33
Tabela 2 – Análise de Complexidade dos Componentes	36
Tabela 3 – Análise de Tamanho dos Componentes	39
Tabela 4 – Probabilidade de Falha do Método	43
Tabela 5 – Impacto de Falha dos Métodos	46
Tabela 6 – Valor de Risco do Método	50
Tabela 7 – Cenários de Teste Priorizados	53
Tabela 7 – Cenários de Teste Priorizados	54
Tabela 8 – Cadastrar Distribuidora	60
Tabela 9 – Editar Distribuidora	60
Tabela 10 – Tentativa de Cadastrar Distribuidora	60
Tabela 11 – Tentativa de Editar Distribuidora	61
Tabela 12 – Acessar Página de Instituições	61
Tabela 13 – Cadastro de Instituição	61
Tabela 14 – Tentativa de Cadastro de Instituição	61
Tabela 15 – Login bem sucedido	61
Tabela 16 – Tentativa de login	62
Tabela 17 – Acessar Painel	62
Tabela 18 – Acessar Página de Pessoas	62
Tabela 19 – Cadastro de Pessoa pelo adm geral	62
Tabela 20 – Tentativa de Cadastro de Pessoa pelo adm geral	62
Tabela 21 – Cadastro de Pessoa pelo gestor	63
Tabela 22 – Tentativa de Cadastro de Pessoa pelo gestor	63
Tabela 23 – Adição de Tarifa	63
Tabela 24 – Tentativa de Adição de Tarifa	63
Tabela 25 – Acessar Análise da Unidade Consumidora	63
Tabela 26 – Editar Contrato de Unidade Consumidora	64
Tabela 27 – Acessar Contrato da Unidade Consumidora	64
Tabela 28 – Tentativa de Editar Contrato de Unidade Consumidora	64
Tabela 29 – Acessar Unidade Consumidora	64
Tabela 30 – Análise Detalhada	64
Tabela 31 – Metodologias Avaliadas	65

Lista de abreviaturas e siglas

CR	<i>Changed Requirements</i>
LAPPIS	Laboratório Avançado de Produção, Pesquisa e Inovação de <i>Software</i>
LOC	<i>Lines of Code</i>
MC	<i>Methods Complexity</i>
MFL	<i>Methods Failure Likelihood Computation</i>
MRV	<i>Method Risk Value</i>
ROI	<i>Return on investment</i>
SUT	<i>Software under test</i>
TCC1	Trabalho de Conclusão de Curso 1
TCC2	Trabalho de Conclusão de Curso 2
XT	<i>Extreme Programming</i>

Sumário

1	INTRODUÇÃO	13
1.1	Contexto	13
1.2	Objetivos	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	15
1.3	Metodologia	15
1.3.1	Plano Metodológico	15
1.4	Organização do Trabalho	15
2	REFERENCIAL TEÓRICO	17
2.1	Testes de Software	17
2.1.1	Teste funcional	17
2.1.2	Teste de sistema	18
2.1.3	Testes regressivos e execução de testes	20
2.1.4	Automação de testes regressivos	21
2.2	Abordagens para priorização da automação de teste de sistema	21
3	MÉTODOS	24
3.1	Seleção do Método de Priorização	24
3.2	Estudo de Caso	25
3.2.1	Contexto	25
3.2.2	Design	25
3.2.3	Seleção de Casos	25
3.2.4	Procedimentos e Papéis do Estudo de Caso	26
3.2.5	Coleta de Dados	26
3.2.6	Análise	26
3.2.7	Validade do Plano	26
3.2.8	Limitações do Estudo	26
3.2.9	Relatório	26
3.2.10	Anexos	27
4	PRIORIZAÇÃO DE CASOS DE TESTE BASEADA EM RISCO	28
4.1	Etapa Inicial	28
4.2	Fatores Indicadores de Risco	28
4.2.1	Requisitos Alterados	28
4.2.2	Complexidade do Método	29

4.2.3	Tamanho do Método	29
4.3	Probabilidade de Falha do Método (MFL)	30
4.4	Impacto de Falha do Método (MFI)	30
4.5	Cálculo do Valor de Risco do Método (MRV)	31
4.6	Priorização dos Casos de Teste	31
5	MEC-ENERGIA	32
5.1	Aplicação Método de Priorização	32
5.1.1	Considerações Iniciais	32
5.1.2	Probabilidade de Falha do Método	33
5.1.2.1	Coleta dos Fatores Indicadores de Riscos	33
5.1.2.1.1	Requisitos Alterados	33
5.1.2.1.2	Complexidade do Método	36
5.1.2.1.3	Tamanho do Método	39
5.1.3	Coleta dos Impactos de Falha dos Métodos	46
5.1.4	Cálculo do Valor de Risco do Método	50
5.1.5	Priorização dos Casos de Teste	53
5.1.6	Análise dos Resultados Obtidos	54
6	CONCLUSÃO	56
	REFERÊNCIAS	57
	ANEXOS	59
	ANEXO A – CENÁRIOS DE TESTES	60
	ANEXO B – COMPARAÇÃO DAS METODOLOGIAS IDENTIFI- CADAS	65

1 Introdução

1.1 Contexto

Um produto de software com qualidade é definido como aquele que cumpre um conjunto de regras e acordos que fornecem ao produto final a capacidade de atender às exigências do usuário e satisfazer suas necessidades (MATEEN; ZHU; AFSAR, 2018). Ao realizar testes de software, busca-se adicionar valor ao produto, implicando na elevação da qualidade ou confiabilidade do SUT (*software under test*) (MYERS; SANDLER; BADGETT, 2011).

O teste de software é definido como o processo, ou uma série de processos, que possuem o propósito de garantir que o software execute suas funcionalidades conforme planejado e que não realize ações não intencionais. Espera-se que o software seja previsível e coerente, evitando qualquer tipo de comportamento não mapeado ou prejudicial para os usuários, conforme destacado por Myers, Sandler e Badgett (2011). O papel fundamental desse processo é garantir que a maioria dos erros seja identificada durante a fase de desenvolvimento, como ressaltado por Oliinyk e Oleksiuk (2019), evitando que os problemas e suas consequências afetem o usuário final.

Em contrapartida à necessidade e importância dos testes para o produto de software, é essencial destacar os recursos necessários envolvidos em tal atividade, onde os custos atrelados ao exercício de testes de um produto de software chegam a pelo menos 50% do custo total do projeto (RAMLER; WOLFMAIER, 2006). É importante adotar uma gestão eficaz dos *custos × recursos* levando em consideração os custos atrelados à verificação do produto, preservando sua qualidade. Uma abordagem possível é identificar quais tipos de testes são necessários para cada fase do projeto, bem como sua forma de execução.

Para uma boa relação custo-benefício, torna-se essencial encontrar meios de assegurar a qualidade do software (por meio da realização de testes) de maneira mais econômica, eliminando a necessidade de um esforço constante e permitindo que esse esforço seja progressivamente otimizado e utilizado de acordo com o ciclo de vida do produto. Uma possibilidade é utilizar a automação de testes associada à priorização de casos de teste, uma estratégia amplamente utilizada para reduzir o custo dos testes manuais (AMANEJAD et al., 2014a), que permitem reduzir o esforço dos testes sem comprometer sua efetividade e a qualidade do produto (SAHAF et al., 2014).

Tendo se tornado amplamente adotada na indústria de software, a automação de testes necessita de investimentos significativos na sua concepção e implementação

(SAHAF et al., 2014), sendo crucial realizar uma análise do ROI (*return on investment*) em relação aos testes manuais para medir e assegurar que a automação dos testes de software reduzirão, efetivamente, os custos de testes a longo prazo.

Ainda que a automação de testes possa reduzir os custos atrelados à verificação do software aumentando a velocidade e cobertura de testes executados, ela traz consigo novos custos que devem ser levados em consideração (KARHU et al., 2009). Para introduzir a automação de testes de software em um projeto, é necessário que os membros envolvidos adaptem-se às novas práticas de trabalho e que seja fornecido treinamento em linguagens de programação, *frameworks* e bibliotecas voltadas para testes automatizados (KARHU et al., 2009).

Além do custos iniciais atrelados à automação, vale ressaltar aspectos intrínsecos à prática dos testes automatizados que, embora ofereçam eficiência a longo prazo, podem acabar criando uma ilusão de qualidade do produto. Os testes automatizados necessitam de evoluções constantes que acompanham a evolução do SUT e perdem sua eficácia ao longo do tempo caso não possuam a manutenção adequada (AMANNEJAD et al., 2014b).

Mesmo que a execução do testes regressivos seja necessária e crucial, a múltipla execução de uma suíte de testes regressivos se torna inviável em situações de cronograma reduzido ou orçamento limitado, sendo necessária uma priorização dos testes que serão executados em cada suíte de testes regressivos (JAHAN; FENG; MAHMUD, 2020).

Com o passar dos anos, diversos métodos de priorização de testes têm sido propostos e avaliados com intuito de identificar defeitos em um produto de software, como por exemplo: diretrizes que indicam os testes a serem priorizados (HAAS et al., 2021), recomendação de testes baseado em marcações feitas a partir do conhecimento da funcionalidade e do código fonte (AZIZI, 2021), tipos de processos e componentes passíveis de serem automatizados (OLIINYK; OLEKSIUK, 2019), etc.

1.2 Objetivos

1.2.1 Objetivo Geral

Tendo conhecimento dos aspectos mencionados e dos variados métodos de priorização de testes, este trabalho tem como objetivo *selecionar e analisar um método de priorização de testes*. Para isso, um método será escolhido a partir do estudo da literatura e será implementado em um projeto real com o intuito de analisar sua eficácia através dos resultados obtidos.

1.2.2 Objetivos Específicos

Para atingir o objetivo estabelecido, foram determinados os seguintes objetivos específicos:

- Pesquisar na literatura as abordagens existentes para priorização dos casos de teste;
- Selecionar um método de priorização de testes;
- Aplicar o método de priorização de testes em um projeto real;
- Analisar os resultados obtidos.

1.3 Metodologia

A pesquisa possui natureza aplicada, pois visa resolver um problema específico e prático dentro do contexto de priorização de casos de testes voltados para automação em projetos de desenvolvimento de software.

A abordagem da pesquisa será qualitativa, focando na análise detalhada e interpretativa dos dados coletados durante a implementação do método de priorização.

A tipologia da pesquisa será exploratória e descritiva. Exploratória pois busca identificar e analisar diferentes metodologias de priorização de casos de testes, e descritiva porque detalhará o processo de aplicação e os resultados obtidos, proporcionando uma visão clara e detalhada do estudo.

A técnica empregada incluirá a revisão literária para a seleção das metodologias de priorização, a avaliação de conteúdo para a definição dos critérios de seleção, e o estudo de caso para a aplicação do método selecionado em um projeto real. O estudo de caso permitirá avaliar o processo e os resultados, permitindo uma validação sobre a eficácia do método de priorização.

1.3.1 Plano Metodológico

Para realizar a validação do método, o plano metodológico será baseado em uma adaptação do Protocolo de Estudo de Caso proposto por [Brereton et al. \(2008\)](#). A adaptação se fará necessária principalmente para adequar o trabalho ao cenário enfrentado. As etapas serão detalhadas no [Capítulo 3 - Métodos](#).

1.4 Organização do Trabalho

O trabalho está dividido em 7 capítulos. Neste, [Capítulo 1 - Introdução](#), é realizada a introdução ao trabalho, contemplando sua contextualização, revisão bibliográfica,

objetivos e metodologia aplicada. Os demais capítulos são estruturados da seguinte forma:

- **Capítulo 2 - Referencial Teórico:** Nesta etapa foi realizada uma pesquisa focada em especificidades da automação de testes e em definições de testes de software, algumas das bibliografias utilizadas para contextualização na etapa de identificação do problema foram reaproveitadas nesta etapa. Com auxílio da pesquisa, foi possível condensar definições sobre os tipos de testes existentes, bem como sua forma de execução.
- **Capítulo 3 - Métodos:** Retoma-se o objetivo e o plano metodológico, sendo apresentado o detalhamento do plano metodológico que será utilizado no trabalho.
- **Capítulo 4 - Priorização de Casos de Testes Baseado em Risco:** Neste capítulo é apresentado, de forma aprofundada, a Priorização de Casos de Teste Baseada em Risco descrita por [Jahan, Feng e Mahmud \(2020\)](#).
- **Capítulo 5 - MEC-Energia** - Neste capítulo apresenta-se o projeto MEC-Energia bem como a aplicação do método de priorização ao projeto.
- **Capítulo 6 Conclusão** - Retoma-se o objetivo identificado na introdução e apresenta as percepções e resultados identificados no decorrer trabalho. Também são apresentadas ideias de trabalhos futuros que podem acrescentar na discussão do tema.

2 Referencial Teórico

Antes de tudo, faz-se necessário estabelecer uma contextualização adequada do conteúdo. Nesse momento, ressalta-se a importância de estabelecer uma compreensão sólida, e fundamentada, acerca das definições que serão utilizadas.

2.1 Testes de Software

Segundo [MYERS; SANDLER; BADGETT \(2011\)](#), é possível adicionar valor a um produto de software através da execução de testes provando sua qualidade ou confiabilidade sendo, neste contexto, o processo de executar o programa com a intenção de encontrar erros. A execução dos testes também auxilia na identificação de erros passíveis de correção antes do contato com usuário final, permitindo uma experiência mais adequada e sólida.

Há diversos tipos e níveis de testes de software, cada um voltado para partes específicas ou granularidades do programa, que atendem a diferentes níveis de verificação e validação. Este trabalho concentrará sua atenção nos testes funcionais, testes de sistema, testes regressivos e na execução dos testes, buscando relacioná-los e elucidar questões importantes para o modelo a ser criado. Acredita-se que esses aspectos sejam cruciais não apenas em termos de custo de investimento durante a implementação, mas também em relação ao potencial retorno, que por muitas vezes surge a longo prazo.

2.1.1 Teste funcional

O teste funcional busca determinar se o programa testado atende às funcionalidades, ou seja, determinar se o comportamento do programa testado está de acordo com a especificação descrita previamente ([MYERS; SANDLER; BADGETT, 2011](#)). De certo modo, o teste funcional visa assegurar que o programa cumpre aquilo que foi descrito e acordado na história de usuário, ou na forma que esteja sendo utilizada para descrição de funcionalidades e requisitos, possibilitando uma avaliação do ponto de vista do usuário ([DELAMARO, 2016](#)).

Semanticamente, o teste funcional atua em um nível menor se comparados aos testes de sistema, essa diferença no escopo do teste deve ser levada em consideração na tomada de decisão no que diz respeito aos critérios de priorização de casos de testes. Esse fator de generalização e especificidades serão fatores cruciais na determinação de quais testes devem ser automatizados.

Pelo motivo dos testes funcionais basearem-se apenas nas especificações dos requisitos levantados, os critérios utilizados na realização do teste dependem, basicamente, da qualidade das especificações dos requisitos (DELAMARO, 2016). Além disso, o fator de generalização permite que sejam empregadas técnicas previamente estabelecidas de acordo com a especificidade da funcionalidade a ser testada. Segundo MYERS; SANDLER; BADGETT, podem-se destacar:

- Particionamento de equivalência - consiste na divisão dos possíveis valores de entrada em faixas de valores específicas e utiliza um valor, ou uma porção de valores, para executar os testes;
- Análise de valores limites - consiste na seleção dos valores a serem utilizados baseados nas limitações impostas pela funcionalidade, sendo necessário utilizar o valor de corte de determinada entrada e seus valores imediatamente anterior e imediatamente posterior;
- Grafos de causa-efeito - consiste na representação lógica das combinações de entradas possíveis em determinado teste, possibilitando a visualização da maior quantidade de cenários possíveis;
- Adivinhação de erros - consiste na criação de planos de testes focado nas áreas que o profissional acredita, de forma intuitiva ou não, que haverão mais erros.

2.1.2 Teste de sistema

Diferente do teste funcional, o objetivo do teste de sistema atua comparando o software produzido com seus requisitos levantados, não se limitando à sistemas ou subsistemas, buscando verificar a existência de discrepâncias entre o produto e seus objetivos medidos e descritos, sendo uma etapa crucial na verificação de qualidade do produto e o mais propenso a erros (MYERS; SANDLER; BADGETT, 2011).

Dado o objetivo por muitas vezes considerado vago, torna-se difícil determinar abordagens para criação de casos de testes de sistema, de modo a direcionar o foco e facilitar elaboração das suítes de testes, MYERS; SANDLER; BADGETT (2011) elenca 15 categorias de testes, ou objetivos de testes, que devem ser levadas em consideração no momento da concepção dos testes de sistemas:

- **Funcionalidade** - o teste de funcionalidade, dentro do contexto de testes de sistema, busca verificar se os recursos mencionados nos objetivos foram implementados no programa, analisando minuciosamente as especificações, a documentação disponibilizada e o que foi implementado. Assegura a consistência dos testes seguintes;

- **Volume** - busca garantir que a implementação tem capacidade de suportar grandes volumes de dados, exigindo recursos significativos de hardware e de pessoas para preparar, realizar e analisar os resultados obtidos;
- **Estresse** - verifica se o programa tem capacidade de lidar com cargas intensas em um curto período de tempo, diferente do teste de volume que considera apenas grandes volumes de dados. O teste de estresse analisa o programa sob picos de atividade, simulando situações extremas;
- **Usabilidade** - envolve o usuário em um teste aplicado no mundo real, onde problemas reais podem ser revelados, problemas esses que não seriam identificados nos testes manuais;
- **Segurança** - testes que possuem especificidades voltadas para a segurança das informações e privacidade do usuário, confirmar verificações de segurança explorando problemas conhecidos em sistemas similares;
- **Desempenho** - indica valores relacionados a desempenho, como *throughput* sob condições específicas de carga e configuração;
- **Armazenamento** - testa a capacidade do software de gerenciar a quantidade de memória do sistema utilizada para execução e armazenamento de suas funcionalidades;
- **Configuração** - avalia o software em diferentes combinações de hardware e software, mesmo que seja impossível testar em todas as combinações disponíveis, testar nas mais utilizadas pelo usuários garante que boa partes dos usuários terão experiências similarmente positivas;
- **Compatibilidade/Conversão** - verifica se o software desenvolvido é compatível com os sistemas e procedimentos existentes. Esse tipo de teste é essencial na atualização e/ou migração de softwares, visto que é ele que vai garantir que o software tem a capacidade de adequar-se à nova combinação de softwares e hardware;
- **Instalação** - testar os procedimentos de instalação é parte importante da implantação de um software, uma instalação mal sucedida pode implicar em diversos problemas ao usuário desde baixo desempenho até mau funcionamento. Também é importante verificar se a documentação de instalação disponibilizada tem todas as informações necessárias para o processo;
- **Confiabilidade** - sendo essencial em todo programa, determinar a confiabilidade de um software é um desafio ímpar, uma das formas que pode ser adotada para medir a confiabilidade de um software é realizar afirmações indutivas para provar, dentro da realidade, a ausência de erros;

- **Recuperação** - verifica se as funções de recuperação funcionam de acordo com o esperado, possibilitando a retomada da jornada que estava sendo executada;
- **Manutenção/Serviço** - verifica a capacidade do programa de realizar diagnósticos, manutenções e logs sobre si próprio;
- **Documentação** - verifica se a documentação que será disponibilizada ao usuário está em sintonia com o que programa que será liberado. Uma forma de realizar essa verificação é usar a própria documentação para especificar representações dos casos de teste;
- **Procedimento** - verifica todos os procedimentos externos ao programa que serão realizados por usuários reais.

2.1.3 Testes regressivos e execução de testes

Os testes regressivos são de suma importância tendo a função de validar o software modificado, garantindo que as alterações adicionadas estão corretas e não afetam de forma negativa a implementação anterior, bem como outras partes do software (XU; DIAS; RICHARDSON, 2004). Sendo executado na fase de desenvolvimento e manutenção, os testes regressivos são essenciais para garantir que as modificações implementadas estão corretas, no que diz respeito à código implementado, e que as adições realizadas não alteraram o fluxo de funcionamento correto do software (DELAMARO, 2016).

Os esforços gerados pelos testes regressivos tendem a acompanhar o desenvolvimento do produto, onde as adições realizadas ao longo do tempo aumentam de acordo com a quantidade de funcionalidades presente no SUT. Cada iteração do ciclo de manutenção, possuirá mais testes regressivos que a anterior, o que implica dizer que a suíte de testes regressivos executada em uma $entrega_n$ tende a ser maior que a suíte de testes regressivos executadas em uma $entrega_{n-1}$, desde que a quantidade de funcionalidades presentes na $entrega_n$ seja maior que quantidade de funcionalidades presente na $entrega_{n-1}$. Desse modo, nota-se um crescente esforço no custo de execução da suíte de testes regressivos, sendo necessário estabelecer formas de priorizar os testes a serem executados.

Do ponto de vista da agilidade, para o teste ágil é necessário possuir uma abordagem colaborativa que envolva todos os processos do teste de software, do design à execução, desse modo os desenvolvedores também podem realizar testes (testes unitários) de modo a garantir a qualidade dos componentes desenvolvidos, favorecendo o desenvolvimento contínuo (MYERS; SANDLER; BADGETT, 2011). Levando em consideração o tempo dos ciclos de desenvolvimento ágeis, percebe-se que o tempo disponível para execução dos testes é reduzido, fazendo com que os testes manuais possam gerar um gargalo nas entregas, bem como na execução dos testes regressivos.

A automação de testes vem sendo empregada na indústria como uma maneira de reduzir o gargalo gerado pela execução dos testes, de forma manual, a cada entrega do produto, possibilitando a redução do esforço direto na execução dos testes sem comprometer a qualidade do produto final (SAHAF et al., 2014).

2.1.4 Automação de testes regressivos

A automação de testes de software é mais empregada no contexto dos testes regressivos (OLIINYK; OLEKSIUK, 2019), presente em produtos de software que possuem várias versões, permitindo uma redução de custos nas subseqüentes execuções de testes regressivos realizadas ao longo das evoluções (GAROUSI; FELDERER, 2016).

A automação de testes de software é uma forma de garantir a qualidade de software diretamente ligada à relação *tamanho da suíte de testes* × *tempo de execução* (KARHU et al., 2009). Ainda que os testes possuam, por natureza, custos e complexidades elevadas, a automação possibilita uma forma de reduzir custos e complexidade sem comprometer a qualidade do produto (SAHAF et al., 2014). Entretanto a etapa do projeto na qual será implementada e o custo do investimento inicial necessário faz com que os próprios testadores ponderem essa decisão (SAHAF et al., 2014).

2.2 Abordagens para priorização da automação de teste de sistema

A análise de casos de teste é uma etapa crucial do processo de testes de software, garantindo a qualidade e confiabilidade do software. Por isso, é fundamental entender os diversos métodos de análise de casos de teste existentes. Ao longo dos anos, vários métodos foram propostos, cada um com seus próprios pontos fortes, fracos e características. Antes de aplicar um método de priorização, é essencial compreender o panorama atual e como ele se compara às alternativas. Neste contexto, este estudo se propõe a analisar abrangentemente diversos métodos de análise de casos de teste, explorando suas características, vantagens e desvantagens. Através de uma comparação, o objetivo é fornecer uma base sólida para avaliar a implementação do método, permitindo uma decisão baseada na sua efetividade e adequação ao contexto específico do projeto.

Haas et al. (2021) estabelecem diretrizes focadas na otimização do processo de testes manuais focando em técnicas de automação de testes, sintetizadas a partir de uma entrevista semi-estruturada realizada de forma online com engenheiros de testes, testadores, desenvolvedores, arquitetos de testes, *test leads* e gerentes de testes. Das diretrizes especificadas, destacam-se:

- Monitoramento da qualidade do teste;
- Criação de casos de testes com passos reutilizáveis;

- Documentação do resultado dos testes executados para análise e utilização posteriores;
- Análise de linguagem natural para seleção de casos de testes relevantes;
- Refatoração e redução de casos testes obsoletos ou de baixa cobertura;
- Priorização focada em testes que já encontraram defeitos anteriormente; e
- Testes focado nas funcionalidades principais.

Azizi (2021) recomendam uma técnica de priorização de testes que realiza uma recomendação automática de casos de testes baseado na sua cobertura em relação ao trecho do programa alterado utilizando o código fonte e as informações relacionadas à mudança do programa. A técnica é dada através das etapas:

- Geração de marcas:
 - Construção do documentos: construção de uma estrutura de dados que facilite a identificação das funcionalidades;
 - Criação de marcações: cria marcações de forma manual e utilizando um modelo estatístico;
- Geração de consultas: identifica as alterações realizadas, faz a higienização das alterações (remove espaços em branco, quebras de linhas, palavras e símbolos reservados) e gera as consultas que serão utilizadas;
- Marcação do sistema: associa marcas aos casos de teste, de modo que cada caso de teste possa ser representado por uma lista de marcas;
- Recomendação de teste:
 - Avaliação dos pesos de cada marca: realiza uma interpolação para correção dos pesos de cada marca;
 - Avaliação de similaridade: calcula a similaridade dos marcas baseado na frequência em que aparecem nas consultas e no documento de geração de marcas.

E em posse dos artefatos produzidos é possível realizar o ranqueamento, e consequentemente, a priorização dos testes.

Oliinyk e Oleksiuk (2019) determinam que existem alguns tipos de processos e componentes de código que são passíveis de serem automatizados, sendo eles:

- Locais de difícil acesso no sistema, como processos em segundo plano;

- Funcionalidades críticas com alto risco de erros, como sistemas de pagamento;
- Testes de carga para validar funcionalidades sob um grande volume de requisições;
- Validação de mensagens;
- Verificação de dados que envolvam cálculos matemáticos;

Jahan, Feng e Mahmud (2020) propõem uma abordagem baseada na detecção de riscos, priorizando testes que, quando reexecutados, terão maior probabilidade de encontrar defeitos, onde essa probabilidade é dada através de cálculos e co-relações sobre indicadores extraídos análises manuais e das modificações realizadas no código fonte do projeto:

- Mudança dos requisitos de um método: indicador da quantidade de requisitos alterados pela quantidade total de requisitos;
- Complexidade de um método: indicador da quantidade de requisitos cobertos por um método;
- Tamanho da mudança: indicador da quantidade de linhas de código modificadas em relação ao total de linhas de código do projeto;
- Estimativa da probabilidade de falha do método: indicador que co-relaciona a mudança dos requisitos, complexidade e tamanho da mudança de um método;
- Estimativa do impacto da falha do método: indicador que relaciona a importância e impacto de um método com o sua utilização dentro do projeto; e
- Estimativa do risco do método: indicador do risco atrelado ao método.

Com base nesses dados, Jahan, Feng e Mahmud (2020) demonstram que existe uma correlação positiva entre o risco de um método e a probabilidade de falhas. Como consequência, aumenta a probabilidade de um teste que cobre esse método identificar defeitos. Logo, a priorização dos testes deve ser feita por aqueles que cobrem os métodos de maior risco.

3 Métodos

Este trabalho tem como foco a aplicação de um método de priorização de casos de testes em um projeto real. Para isso, foram estabelecidos critérios para a seleção do método de priorização e um projeto adequado às necessidades deste estudo. Com a aplicação do método, espera-se obter dados suficientes para determinar a eficácia do processo de priorização de casos de testes.

3.1 Seleção do Método de Priorização

Para a seleção de um método de priorização de testes, foi realizada uma busca na literatura acadêmica, cujos resultados estão detalhados no [Capítulo 2](#). Em seguida, foram estabelecidos critérios para orientar a escolha do método mais adequado. Esses critérios incluem:

- Aplicação do método: verificação da aplicabilidade do método em projetos reais;
- Metodologia empregada: priorização de metodologias que demonstrem a aplicação prática do método;
- Data de publicação: priorização de metodologias mais recentes;
- Quantidade de citações: número de vezes que a metodologia foi citada em outras publicações;
- Índice de relevância: pontuação do periódico onde o artigo da metodologia foi publicado.

Das diversas formas de priorização identificadas, foi selecionada a metodologia que, de acordo com os critérios estabelecidos, possuía um melhor aproveitamento em termos de aplicação, implementação e validação. A comparação, descrita no [Apêndice B](#), evidencia que o melhor método para ser aplicado no contexto deste trabalho é o método de Priorização de Casos de Testes Baseado em Risco.

A escolha da Priorização de Casos de Testes Baseado em Risco proposta por ([JAHAN; FENG; MAHMUD, 2020](#)) se deu porque, de acordo com os critérios estabelecidos, demonstrou um bom aproveitamento em termos de aplicação, implementação e validação. Diferente de outros métodos que dependem unicamente de estimativas baseadas na experiência de quem aplica, o método selecionado utiliza dados extraídos diretamente do código fonte para calcular parte dos seus fatores de priorização. Essa característica

foi decisiva para sua seleção, pois alinha-se com a necessidade de uma abordagem mais quantitativa e menos subjetiva, visando diminuir o impacto da inconsistência humana. O método está detalhado no Capítulo 4.

3.2 Estudo de Caso

Nesta seção será tratada a forma como o estudo de caso será conduzida, baseando-se em literatura. Para isso, faz-se necessário compreender as etapas descritas por Brereton et al. (2008) no seu *Modelo de Protocolo de Estudo de Caso*:

3.2.1 Contexto

Este item do modelo, Contexto ou *Background*, é resultante das pesquisas realizadas nos capítulos Capítulo 1 e Capítulo 2, que viabilizaram a compreensão dos conceitos de Testes e alguns dos diversos métodos de priorização de testes.

3.2.2 Design

No que tange o Design da pesquisa, este estudo caracterizou-se como um estudo de caso único que será aplicado a um projeto de software.

3.2.3 Seleção de Casos

Para a seleção do projeto, foi necessário atentar-se a determinadas condições essenciais para a aplicação do método de priorização de casos de testes. Os critérios estabelecidos foram:

- Acesso ao código fonte da aplicação: necessidade de avaliação das alterações realizadas;
- Contato com mantenedores: necessidade de estabelecer contato para auxiliar na identificação e no impacto das funcionalidades do projeto;
- Projetos acadêmicos: preferência por projetos reais que foram implementados por alunos da Universidade de Brasília, mais especificamente da Campus Gama.

O projeto MEC-Energia¹ foi selecionado com base na disponibilidade e facilidade de contato com os mantenedores e *stakeholders*. A descrição deste projeto, e objeto de estudo, será melhor detalhada no Capítulo 5 - *MEC-Energia*.

¹ <https://gitlab.com/lappis-unb/projetos-energia/mec-energia/mec-energia-web>

3.2.4 Procedimentos e Papéis do Estudo de Caso

O autor deste trabalho tem vínculo com alguns colaboradores e ex-colaboradores do projeto, em sua grande maioria estudantes de *Engenharia de Software* que atuam, ou atuavam, como desenvolvedores. No que tange os procedimentos, serão realizadas as seguintes atividades baseadas no Objetivos identificados no [Capítulo 1](#):

1. Determinar qual metodologia será aplicada para realizar a priorização de casos de testes;
2. Analisar e selecionar os casos de testes que devem ser priorizados baseado no método de priorização de casos de testes descrito no [Capítulo 4](#);
3. Escrever, caso necessário, os casos de testes para priorização;

3.2.5 Coleta de Dados

A coleta de dados deu-se pelas atividades realizadas com intuito de realizar os Procedimentos identificados no [subseção 3.2.4](#).

3.2.6 Análise

Seguindo de forma sequencial os procedimentos identificados no [subseção 3.2.4](#), a análise será realizada ao final da avaliação da implementação com apoio dos desenvolvedores do objeto de estudo.

3.2.7 Validade do Plano

O trabalho foi validado através de uma validação interna, mostrando a relação dos critérios estabelecidos para priorização de testes e a identificação de defeitos na implementação.

3.2.8 Limitações do Estudo

Considerando que os colaboradores são estudantes de *Engenharia de Software*, a curva de aprendizados deve ser considerada, bem como a deficiência de rastreabilidade nos itens relacionados a testes e correções de defeitos.

3.2.9 Relatório

Este Trabalho de Conclusão de Curso será o Relatório da pesquisa do estudo de caso.

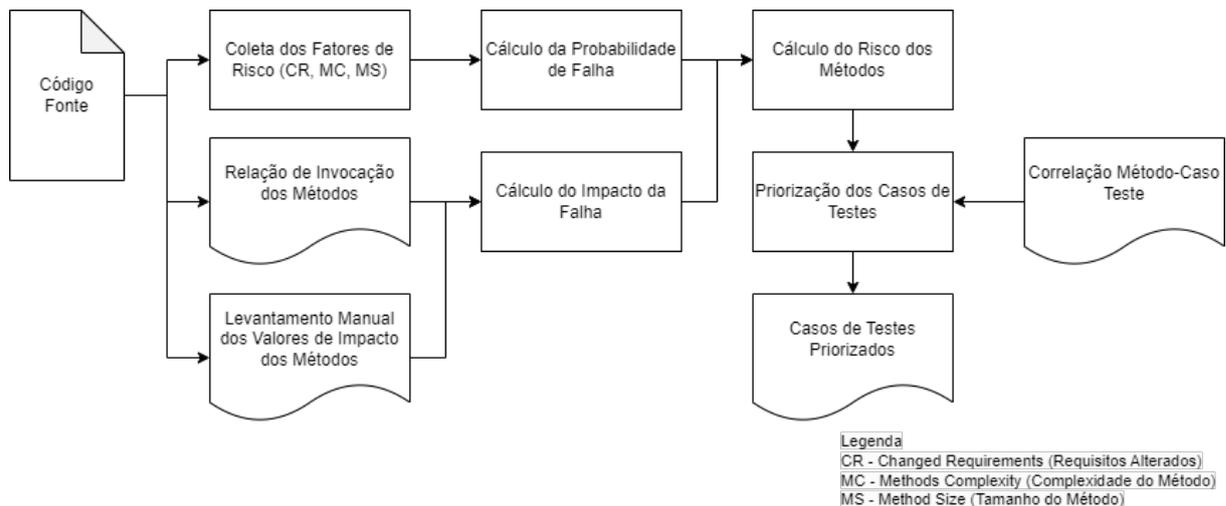
3.2.10 Anexos

No anexo [Apêndice A](#) estão descritos os testes utilizados na priorização. Todos os testes foram escritos em linguagem natural baseado em *Gherkin* ([Cucumber](#), [SmartBear Software](#), 2019).

4 Priorização de Casos de Teste Baseada em Risco

O método selecionado para este estudo é a *Priorização de Casos de Teste Baseada em Risco por Meio da Correlação Entre Métodos do Sistema e Seus Riscos Associados* proposto por [Jahan, Feng e Mahmud \(2020\)](#). Esta abordagem combina etapas automáticas e semi-automáticas para calcular as probabilidades e os impactos que a inserção de um defeito, causado por alterações do código existente ou inserção de novos trechos de código, e com isso classificar quais trechos possuem mais riscos e são passíveis de serem priorizados. A [Figura 1](#) ilustra todas as etapas presentes no modelo:

Figura 1 – Etapas do Modelo de Priorização de Testes



Autoria: Própria. Fonte: Adaptado de [Jahan, Feng e Mahmud \(2020\)](#)

4.1 Etapa Inicial

Na etapa inicial, realiza-se uma análise textual automática dos arquivos-fonte, de modo a extrair os requisitos alterados, o tamanho e a complexidade das alterações realizadas. Esses fatores serão utilizados para calcular a probabilidade de falha dos métodos.

4.2 Fatores Indicadores de Risco

4.2.1 Requisitos Alterados

O fator Requisitos Alterados (*Changed Requirements* - CR) baseia-se no número de requisitos alterados dentro de um método. Em produtos de *software*, os defeitos geralmente

surtem a partir de alterações realizadas para suprir as mudanças realizadas nos requisitos do produto. Portanto, o número de requisitos alterados [Equação 4.1](#) em um método pode ser um indicador eficaz para identificar possíveis defeitos. Cada CR deve ser normalizado entre 0 e 1, onde 0 indica poucas alterações nos requisitos e 1 indica muitas alterações nos requisitos.

$$CR_i = \frac{NR_i}{\text{Max}_{\forall i}\{NR_i\}} \quad (4.1)$$

Onde NR_i representa a quantidade de requisitos alterados em um método M_i .

4.2.2 Complexidade do Método

A Complexidade dos Métodos (*Methods Complexity* - MC) é calculada com base no número total de requisitos presentes nesse método, ou seja, quantos requisitos aquele método contempla. Um método pode contemplar um ou mais requisitos. Entretanto, a presença de muitos requisitos torna um método complexo e, assim, mais suscetível a induzir implementações incorretas, ou ainda mais complexas, o que pode aumentar os riscos relacionados aos métodos. Portanto, métodos que contemplam muitos requisitos ou possuem alta complexidade são considerados mais complexos. Para contabilizar a complexidade de um método, deve-se contabilizar o número de requisitos contemplados por cada método. Cada MC deve ser normalizado entre 0 e 1, onde o 0 indica o método menos complexo e 1 indica o método mais complexo, conforme [Equação 4.2](#).

$$MC_i = \frac{NC_i}{\text{Max}_{\forall i}\{NC_i\}} \quad (4.2)$$

Onde NC_i representa o número de condicionais ou requisitos contemplados em um método M_i .

4.2.3 Tamanho do Método

Para computar esse fator, o total de Linhas de Código (LOC) para cada método deve ser considerado, visto que o Tamanho dos Métodos (*Methods Size* - MS) pode afetar a quantidade de defeitos encontrados em produtos de *software*. Faz-se necessário normalizar o fator MS entre 0 e 1, onde 0 indica um menor método e 1 indica um método extenso, conforme [Equação 4.3](#).

$$MS_i = \frac{LOC_i}{\text{Max}_{\forall i}\{LOC_i\}} \quad (4.3)$$

Onde LOC_i representa o total de linhas de código em um método M_i .

4.3 Probabilidade de Falha do Método (MFL)

Considerando que as atividades realizadas na fase de desenvolvimento de um *software* influenciam fortemente na sua probabilidade ao risco, multiplica-se os fatores descritos em [Equação 4.1](#), [Equação 4.2](#) e [Equação 4.3](#) para calcular a probabilidade de falha de um método, conforme [Equação 4.4](#):

$$MFL_i = CR_i \times MC_i \times MS_i \quad (4.4)$$

4.4 Impacto de Falha do Método (MFI)

Antes de calcular o Impacto de Falha do Método (*Methods Failure Impact* - MFI), faz-se necessário considerar dois pontos e sua relação com o cenário de priorização de testes:

- Relação de Chamada dos Métodos - dependência funcional de um software. Se um método contém defeitos, esses defeitos podem afetar, de forma direta ou indireta, outros métodos dependentes dele. Um método que é chamado diversas vezes por outros métodos pode ter a detecção de seus defeitos facilitada pela quantidade de vezes que ele foi invocado e executado, facilitando a identificação de defeitos em um método;
- Valor de Impacto dos Métodos - o valor de impacto de um método é levantado de forma manual baseado na relevância do método para o sistema. A relevância do método no sistema pode ser estipulada pela análise da importância dos requisitos dos métodos ou pela experiência de alguém experiente no projeto;

Tendo os valores da relação dos métodos e do valor de impacto, que deve ser definido com valores entre 1 (método de baixa importância e baixo impacto) e 5 (alta importância e alto impacto), é possível calcular o MFI_i conforme [Equação 4.5](#):

$$MFI_i = \frac{\sum_{j=1}^x MI_{ij}}{|NMC_i| \times \max\{MI_i\}} \quad (4.5)$$

Onde $\sum_{j=1}^x MI_{ij}$ representa a soma dos impactos de todos os métodos em que o MI_i é invocado, incluindo o impacto da sua própria chamada; NMC_i representa a quantidade total de métodos pelos quais o método MI_i é chamado, incluindo ele mesmo; e $\max\{MI_i\}$ é o maior valor de impacto de qualquer método entre todos os métodos.

4.5 Cálculo do Valor de Risco do Método (MRV)

Em posse dos valores de MFL [Equação 4.4](#) e MFI [Equação 4.5](#), é possível estabelecer o valor de risco do método, que indicará a probabilidade de ocorrência de falhas. Valores de Risco do Método (*Method Risk Value* - MRV) mais elevados indicam uma maior probabilidade do método conter erros que possam ser detectados. Os valores para MRV podem ser obtidos através da [Equação 4.6](#):

$$MRV_i = MFL_i \times MFI_i \quad (4.6)$$

4.6 Priorização dos Casos de Teste

Para realizar a priorização dos casos de testes, calcula-se o valor dos riscos de casos de testes baseando-se na soma dos MRV_i cobertos por eles. Faz-se necessário conhecimento da correlação entre os métodos implementados, as funcionalidades e os casos de testes. Caso a correlação não possa ser estabelecida, a documentação com especificações funcionais, histórias de usuários, podem ser utilizados para auxiliar a estabelecer a correlação entre os métodos e os casos de testes.

O valor de risco de cada caso de teste pode ser calculado conforme:

$$T_k = \sum_{i=1}^y MRV_{ik} \quad (4.7)$$

Onde o k indica o *késimo* caso de teste, o y indica o número de métodos cobertos pelo *késimo* caso de teste.

Após estabelecer os valores de T_K para os casos de testes, a priorização dar-se-á pela ordenação decrescente dos valores de T_k .

5 MEC-Energia

O aplicativo MEC-Energia é um sistema de recomendação de energia elétrica, voltado para instituições de ensino superior, coordenado pelos professores Renato Coral e Loana Velasco sediado no laboratório LAPPIS presente na Universidade de Brasília. O sistema tem como foco auxiliar na gerência e avaliação dos contratos de contas de energia elétrica a partir do registro das faturas mensais de energia, permitindo a geração de relatórios de consumo e recomendações de ajustes nos contratos visando redução de custos (VELASCO; CORAL, 2023).

O sistema desenvolvido será utilizado na aplicação do método de Priorização de Casos de Testes Baseado em Risco, objeto de estudo deste trabalho.

5.1 Aplicação Método de Priorização

Retomando o objetivo específico descrito no [Capítulo 1](#), neste capítulo serão apresentados os dados obtidos através do método de priorização descrito no [Capítulo 4](#).

5.1.1 Considerações Iniciais

Definiu-se que a parte do sistema a ser utilizada como insumo para aplicação do método de priorização seria o código fonte presente no repositório MEC Energia Web¹. Considerando o caráter incremental ao qual o método de priorização de testes se enquadra, foi definido uma versão inicial e uma versão final do código fonte que seria avaliado:

- Versão inicial: "v1.0.0 - Versão final da entrega do projeto"²
- Versão final: "develop - fix: change date picker views"³

Ressalta-se a necessidade de adaptar as análises realizadas de acordo com a natureza do código fonte analisado. Deve-se levar em consideração que código fonte trata da implementação do *frontend* da aplicação. Com isso foi necessário abstrair a análise de métodos para análise dos arquivos de componentes presente no *atomic design*, onde os componentes são pensados e implementados de modo a ter o maior reaproveitamento (FROST, 2016).

¹ <https://gitlab.com/lappis-unb/projetos-energia/mec-energia/mec-energia-web>

² <https://gitlab.com/lappis-unb/projetos-energia/mec-energia/mec-energia-web/-/tags/v1.0.0>

³ <https://gitlab.com/lappis-unb/projetos-energia/mec-energia/mec-energia-web/-/commits/13edc44b>

5.1.2 Probabilidade de Falha do Método

5.1.2.1 Coleta dos Fatores Indicadores de Riscos

De modo a ter uma maior confiança dos dados coletados, foi utilizada a ferramenta de análise de código estático *SonarCloud* para realizar a coleta de dados quantitativos como quantidade de linhas de código e complexidade dos componentes.

5.1.2.1.1 Requisitos Alterados

Na [Tabela 1](#) é possível visualizar os dados coletados e o valor do CR_i , de acordo com a [Equação 4.1](#).

Tabela 1 – Análise de Requisitos Alterados

Arquivo	NR	$\max \forall NR_i$	CR_i
src/components/Card/index.tsx	2	24	0,08333333
src/components/Card/Wrapper.tsx	0	24	0,00000000
src/components/ConsumerUnit/Card.tsx	8	24	0,33333333
src/components/ConsumerUnit/Content/Analysis AndRecommendation.tsx	6	24	0,25000000
src/components/ConsumerUnit/Content/Dropdown Section.tsx	1	24	0,04166666
src/components/ConsumerUnit/Form/Create.tsx	11	24	0,45833333
src/components/ConsumerUnit/Form/Edit.tsx	20	24	0,83333333
src/components/ConsumerUnit/Form/Renew Contract.tsx	16	24	0,66666666
src/components/ConsumerUnit/Form/Warning Dialog.tsx	3	24	0,12500000
src/components/Distributor/DistributorCard.tsx	6	24	0,25000000
src/components/Distributor/Form/CreateForm.tsx	2	24	0,08333333
src/components/Distributor/Form/Distributor CreateForm.tsx	12	24	0,50000000
src/components/Distributor/Form/Distributor EditForm.tsx	7	24	0,29166666
src/components/Drawer/index.tsx	8	24	0,33333333
src/components/Drawer/ListItem.tsx	1	24	0,04166666
src/components/ElectricityBill/Form/CreateEdit ElectricityBillForm.tsx	24	24	1,00000000
src/components/Footer/index.tsx	1	24	0,04166666
src/components/Form/DrawerV2.tsx	1	24	0,04166666

Arquivo	NR	$max\forall NR_i$	CR_i
src/components/Institution/Form/CreateInstitutionForm.tsx	2	24	0,0833333
src/components/Institution/Form/EditInstitutionForm.tsx	3	24	0,1250000
src/components/Person/Form/CreatePersonForm.tsx	4	24	0,1666666
src/components/Person/Form/EditPersonForm.tsx	3	24	0,1250000
src/components/Tariff/Form/TariffCreateForm.tsx	22	24	0,9166666
src/pages/_document.tsx	1	24	0,0416666
src/pages/api/auth/[...nextauth].ts	1	24	0,0416666
src/pages/distribuidoras/[distributorId].tsx	2	24	0,0833333
src/pages/distribuidoras/index.tsx	1	24	0,0416666
src/pages/pessoas.tsx	1	24	0,0416666
src/pages/uc/[id].tsx	2	24	0,0833333
src/pages/uc/index.tsx	2	24	0,0833333
src/templates/Analysis/BaseCostComparisonCard.tsx	6	24	0,2500000
src/templates/Analysis/BaseCostComparisonPlot.tsx	4	24	0,1666666
src/templates/Analysis/CurrentBaseCostPlot.tsx	4	24	0,1666666
src/templates/Analysis/DetailedAnalysisDrawer/BaseCostComparisonTable.tsx	6	24	0,2500000
src/templates/Analysis/DetailedAnalysisDrawer/ConsumerUnitInfo.tsx	1	24	0,0416666
src/templates/Analysis/DetailedAnalysisDrawer/ConsumptionHistoryTable.tsx	4	24	0,1666666
src/templates/Analysis/DetailedAnalysisDrawer/ContractsComparisonTable.tsx	5	24	0,2083333
src/templates/Analysis/DetailedAnalysisDrawer/CurrentContractTable.tsx	5	24	0,2083333
src/templates/Analysis/DetailedAnalysisDrawer/DetailedBaseCostsComparisonPlot.tsx	3	24	0,1250000
src/templates/Analysis/DetailedAnalysisDrawer/index.tsx	12	24	0,5000000
src/templates/Analysis/DetailedAnalysisDrawer/Logos.tsx	1	24	0,0416666
src/templates/Analysis/DetailedAnalysisDrawer/RecommendedContractDemandPlot.tsx	4	24	0,1666666

Arquivo	NR	$max\forall NR_i$	CR_i
src/templates/Analysis/DetailedAnalysisDrawer/RecommendedContractTable.tsx	6	24	0,2500000
src/templates/Analysis/DetailedAnalysisDrawer/TariffsTable.tsx	2	24	0,0833333
src/templates/Analysis/DetailedAnalysisHeader.tsx	1	24	0,0416666
src/templates/Analysis/MeasuredConsumptionPlot.tsx	4	24	0,1666666
src/templates/Analysis/MeasuredDemandPlot.tsx	4	24	0,1666666
src/templates/Analysis/RecommendationCard.tsx	1	24	0,0416666
src/templates/Auth/SignIn.tsx	5	24	0,2083333
src/templates/ConsumerUnit/Content/Contract.tsx	4	24	0,1666666
src/templates/ConsumerUnit/Content/Header.tsx	8	24	0,3333333
src/templates/ConsumerUnit/Content/index.tsx	3	24	0,1250000
src/templates/ConsumerUnit/Content/Invoice/Filter.tsx	4	24	0,1666666
src/templates/ConsumerUnit/Content/Invoice/Table.tsx	7	24	0,2916666
src/templates/ConsumerUnit/Grid.tsx	2	24	0,0833333
src/templates/Dashboard/Grid.tsx	3	24	0,1250000
src/templates/Dashboard/index.tsx	1	24	0,0416666
src/templates/DefaultV2.tsx	1	24	0,0416666
src/templates/Distributor/Content/ConsumerUnitsList.tsx	2	24	0,0833333
src/templates/Distributor/Content/index.tsx	3	24	0,1250000
src/templates/Distributor/Content/TariffsTable.tsx	3	24	0,1250000
src/templates/Distributor/Header/index.tsx	9	24	0,3750000
src/templates/Institution/index.tsx	2	24	0,0833333
src/templates/Profile/index.tsx	1	24	0,0416666
src/templates/Profile/ResetPasswordButton.tsx	2	24	0,0833333
src/templates/UserList/index.tsx	10	24	0,4166666
src/templates/UserList/PasswordResetButton.tsx	2	24	0,0833333
src/theme/index.ts	0	24	0
src/types/app.ts	1	24	0,0416666
src/types/auth.ts	2	24	0,0833333
src/types/consumerUnit.ts	1	24	0,0416666
src/types/energyBill.ts	2	24	0,0833333

Arquivo	NR	$max\forall NR_i$	CR_i
src/types/institution.ts	1	24	0,0416666
src/types/person.ts	2	24	0,0833333
src/types/recommendation.ts	2	24	0,0833333
src/types/tariffs.tsx	1	24	0,0416666
src/api/index.tsx	0	24	0
src/middleware.ts	1	24	0,0416666
src/store/appSlice.ts	7	24	0,2916666
src/utills/date.ts	8	24	0,3333333
src/utills/head.ts	1	24	0,0416666
src/utills/number.ts	1	24	0,0416666

Alguns arquivos tiverem apenas sua indentação alteradas, não possuindo mudança de requisitos e nem CR_i .

5.1.2.1.2 Complexidade do Método

Na [Tabela 2](#) é possível visualizar os dados coletados e o valor do MC_I , de acordo com a [Equação 4.2](#).

Tabela 2 – Análise de Complexidade dos Componentes

Arquivo	NC	$max\forall NC_i$	MC_i
src/components/Card/index.tsx	13	132	0,0984848
src/components/Card/Wrapper.tsx	7	132	0,0530303
src/components/ConsumerUnit/Card.tsx	20	132	0,1515151
src/components/ConsumerUnit/Content/Analysis AndRecommendation.tsx	29	132	0,2196969
src/components/ConsumerUnit/Content/Dropdown Section.tsx	7	132	0,0530303
src/components/ConsumerUnit/Form/Create.tsx	86	132	0,6515151
src/components/ConsumerUnit/Form/Edit.tsx	106	132	0,8030303
src/components/ConsumerUnit/Form/Renew Contract.tsx	87	132	0,6590909
src/components/ConsumerUnit/Form/Warning Dialog.tsx	3	132	0,0227272
src/components/Distributor/DistributorCard.tsx	11	132	0,0833333
src/components/Distributor/Form/CreateForm.tsx	17	132	0,1287878

Arquivo	NC	$\max \forall NC_i$	MC_i
src/components/Distributor/Form/DistributorCreateForm.tsx	37	132	0,2803030
src/components/Distributor/Form/DistributorEditForm.tsx	44	132	0,3333333
src/components/Drawer/index.tsx	30	132	0,2272727
src/components/Drawer/ListItem.tsx	10	132	0,0757575
src/components/ElectricityBill/Form/CreateEditElectricityBillForm.tsx	120	132	0,9090909
src/components/Footer/index.tsx	1	132	0,0075757
src/components/Form/DrawerV2.tsx	3	132	0,0227272
src/components/Institution/Form/CreateInstitutionForm.tsx	19	132	0,1439393
src/components/Institution/Form/EditInstitutionForm.tsx	28	132	0,2121212
src/components/Person/Form/CreatePersonForm.tsx	35	132	0,2651515
src/components/Person/Form/EditPersonForm.tsx	28	132	0,2121212
src/components/Tariff/Form/TariffCreateForm.tsx	132	132	1
src/pages/_document.tsx	6	132	0,0454545
src/pages/api/auth/[...nextauth].ts	9	132	0,0681818
src/pages/distribuidoras/[distributorId].tsx	11	132	0,0833333
src/pages/distribuidoras/index.tsx	6	132	0,0454545
src/pages/pessoas.tsx	1	132	0,0075757
src/pages/uc/[id].tsx	11	132	0,0833333
src/pages/uc/index.tsx	6	132	0,0454545
src/templates/Analysis/BaseCostComparisonCard.tsx	8	132	0,0606060
src/templates/Analysis/BaseCostComparisonPlot.tsx	9	132	0,0681818
src/templates/Analysis/CurrentBaseCostPlot.tsx	13	132	0,0984848
src/templates/Analysis/DetailedAnalysisDrawer/BaseCostComparisonTable.tsx	4	132	0,0303030
src/templates/Analysis/DetailedAnalysisDrawer/ConsumerUnitInfo.tsx	2	132	0,0151515
src/templates/Analysis/DetailedAnalysisDrawer/ConsumptionHistoryTable.tsx	3	132	0,0227272
src/templates/Analysis/DetailedAnalysisDrawer/ContractsComparisonTable.tsx	4	132	0,0303030

Arquivo	NC	$\max \forall NC_i$	MC_i
src/templates/Analysis/DetailedAnalysisDrawer/CurrentContractTable.tsx	15	132	0,1136363
src/templates/Analysis/DetailedAnalysisDrawer/DetailedBaseCostsComparisonPlot.tsx	13	132	0,0984848
src/templates/Analysis/DetailedAnalysisDrawer/index.tsx	22	132	0,1666666
src/templates/Analysis/DetailedAnalysisDrawer/Logos.tsx	1	132	0,0075757
src/templates/Analysis/DetailedAnalysisDrawer/RecommendedContractDemandPlot.tsx	10	132	0,0757575
src/templates/Analysis/DetailedAnalysisDrawer/RecommendedContractTable.tsx	8	132	0,0606060
src/templates/Analysis/DetailedAnalysisDrawer/TariffsTable.tsx	14	132	0,1060606
src/templates/Analysis/DetailedAnalysisHeader.tsx	1	132	0,0075757
src/templates/Analysis/MeasuredConsumptionPlot.tsx	11	132	0,0833333
src/templates/Analysis/MeasuredDemandPlot.tsx	11	132	0,0833333
src/templates/Analysis/RecommendationCard.tsx	8	132	0,0606060
src/templates/Auth/SignIn.tsx	15	132	0,1136363
src/templates/ConsumerUnit/Content/Contract.tsx	11	132	0,0833333
src/templates/ConsumerUnit/Content/Header.tsx	22	132	0,1666666
src/templates/ConsumerUnit/Content/index.tsx	11	132	0,0833333
src/templates/ConsumerUnit/Content/Invoice/Filter.tsx	15	132	0,1136363
src/templates/ConsumerUnit/Content/Invoice/Table.tsx	45	132	0,3409090
src/templates/ConsumerUnit/Grid.tsx	10	132	0,0757575
src/templates/Dashboard/Grid.tsx	23	132	0,1742424
src/templates/Dashboard/index.tsx	5	132	0,0378787
src/templates/DefaultV2.tsx	2	132	0,0151515
src/templates/Distributor/Content/ConsumerUnitsList.tsx	9	132	0,0681818
src/templates/Distributor/Content/index.tsx	13	132	0,0984848
src/templates/Distributor/Content/TariffsTable.tsx	27	132	0,2045454
src/templates/Distributor/Header/index.tsx	27	132	0,2045454

Arquivo	NC	$\max \forall NC_i$	MC_i
src/templates/Institution/index.tsx	6	132	0,0454545
src/templates/Profile/index.tsx	6	132	0,0454545
src/templates/Profile/ResetPasswordButton.tsx	2	132	0,0151515
src/templates/UserList/index.tsx	50	132	0,3787878
src/templates/UserList/PasswordResetButton.tsx	4	132	0,0303030
src/theme/index.ts	0	132	0
src/types/app.ts	0	132	0
src/types/auth.ts	0	132	0
src/types/consumerUnit.ts	0	132	0
src/types/energyBill.ts	0	132	0
src/types/institution.ts	0	132	0
src/types/person.ts	0	132	0
src/types/recommendation.ts	0	132	0
src/types/tariffs.tsx	0	132	0
src/api/index.tsx	57	132	0,4318181
src/middleware.ts	0	132	0
src/store/appSlice.ts	70	132	0,5303030
src/utills/date.ts	12	132	0,0909090
src/utills/head.ts	2	132	0,0151515
src/utills/number.ts	5	132	0,0378787

Alguns arquivos não tiveram modificações lógicas, implicando em valores zerados para complexidade e conseqüentemente para seus respectivos MC_i .

5.1.2.1.3 Tamanho do Método

Na [Tabela 3](#) é possível visualizar os dados coletados e o valor do MS_i , de acordo com a [Equação 4.3](#).

Tabela 3 – Análise de Tamanho dos Componentes

Arquivo	LOC	$\max \forall LOC_i$	MS_i
src/components/Card/index.tsx	96	1020	0,0941176
src/components/Card/Wrapper.tsx	45	1020	0,0441176
src/components/ConsumerUnit/Card.tsx	158	1020	0,1549019
src/components/ConsumerUnit/Content/AnalysisAndRecommendation.tsx	206	1020	0,2019607

Arquivo	LOC	$max\forall LOC_i$	MS_i
src/components/ConsumerUnit/Content/DropdownSection.tsx	94	1020	0,0921568
src/components/ConsumerUnit/Form/Create.tsx	771	1020	0,7558823
src/components/ConsumerUnit/Form/Edit.tsx	876	1020	0,8588235
src/components/ConsumerUnit/Form/RenewContract.tsx	671	1020	0,6578431
src/components/ConsumerUnit/Form/WarningDialog.tsx	43	1020	0,0421568
src/components/Distributor/DistributorCard.tsx	102	1020	0,1000000
src/components/Distributor/Form/CreateForm.tsx	219	1020	0,2147058
src/components/Distributor/Form/DistributorCreateForm.tsx	233	1020	0,2284313
src/components/Distributor/Form/DistributorEditForm.tsx	312	1020	0,3058823
src/components/Drawer/index.tsx	272	1020	0,2666666
src/components/Drawer/ListItem.tsx	81	1020	0,0794117
src/components/ElectricityBill/Form/CreateEditElectricityBillForm.tsx	810	1020	0,7941176
src/components/Footer/index.tsx	55	1020	0,0539215
src/components/Form/DrawerV2.tsx	104	1020	0,1019607
src/components/Institution/Form/CreateInstitutionForm.tsx	213	1020	0,2088235
src/components/Institution/Form/EditInstitutionForm.tsx	238	1020	0,2333333
src/components/Person/Form/CreatePersonForm.tsx	341	1020	0,3343137
src/components/Person/Form/EditPersonForm.tsx	247	1020	0,2421568
src/components/Tariff/Form/TariffCreateForm.tsx	994	1020	0,9745098
src/pages/_document.tsx	53	1020	0,0519607
src/pages/api/auth/[...nextauth].ts	81	1020	0,0794117
src/pages/distribuidoras/[distributorId].tsx	77	1020	0,0754901
src/pages/distribuidoras/index.tsx	35	1020	0,0343137
src/pages/pessoas.tsx	14	1020	0,0137254
src/pages/uc/[id].tsx	77	1020	0,0754901
src/pages/uc/index.tsx	33	1020	0,0323529
src/templates/Analysis/BaseCostComparisonCard.tsx	118	1020	0,1156862

Arquivo	LOC	$max\forall LOC_i$	MS_i
src/templates/Analysis/BaseCostComparison Plot.tsx	112	1020	0,1098039
src/templates/Analysis/CurrentBaseCostPlot.tsx	126	1020	0,1235294
src/templates/Analysis/DetailedAnalysisDrawer/ BaseCostComparisonTable.tsx	111	1020	0,1088235
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumerUnitInfo.tsx	42	1020	0,0411764
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumptionHistoryTable.tsx	98	1020	0,0960784
src/templates/Analysis/DetailedAnalysisDrawer/ ContractsComparisonTable.tsx	123	1020	0,1205882
src/templates/Analysis/DetailedAnalysisDrawer/ CurrentContractTable.tsx	143	1020	0,1401960
src/templates/Analysis/DetailedAnalysisDrawer/ DetailedBaseCostsComparisonPlot.tsx	117	1020	0,1147058
src/templates/Analysis/DetailedAnalysisDrawer/ index.tsx	1020	1020	1
src/templates/Analysis/DetailedAnalysisDrawer/ Logos.tsx	28	1020	0,0274509
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractDemandPlot.tsx	165	1020	0,1617647
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractTable.tsx	123	1020	0,1205882
src/templates/Analysis/DetailedAnalysisDrawer/ TariffsTable.tsx	111	1020	0,1088235
src/templates/Analysis/DetailedAnalysisHeader.tsx	28	1020	0,0274509
src/templates/Analysis/Measured ConsumptionPlot.tsx	107	1020	0,1049019
src/templates/Analysis/Measured DemandPlot.tsx	178	1020	0,1745098
src/templates/Analysis/RecommendationCard.tsx	170	1020	0,1666666
src/templates/Auth/SignIn.tsx	210	1020	0,2058823
src/templates/ConsumerUnit/Content/Contract.tsx	194	1020	0,1901960
src/templates/ConsumerUnit/Content/Header.tsx	221	1020	0,2166666
src/templates/ConsumerUnit/Content/index.tsx	85	1020	0,0833333
src/templates/ConsumerUnit/Content/Invoice/ Filter.tsx	86	1020	0,0843137

Arquivo	LOC	$max\forall LOC_i$	MS_i
src/templates/ConsumerUnit/Content/Invoice/ Table.tsx	393	1020	0,3852941
src/templates/ConsumerUnit/Grid.tsx	79	1020	0,0774509
src/templates/Dashboard/Grid.tsx	126	1020	0,1235294
src/templates/Dashboard/index.tsx	37	1020	0,0362745
src/templates/DefaultV2.tsx	55	1020	0,0539215
src/templates/Distributor/Content/Consumer UnitsList.tsx	63	1020	0,0617647
src/templates/Distributor/Content/index.tsx	113	1020	0,1107843
src/templates/Distributor/Content/TariffsTable.tsx	311	1020	0,3049019
src/templates/Distributor/Header/index.tsx	310	1020	0,3039215
src/templates/Institution/index.tsx	80	1020	0,0784313
src/templates/Profile/index.tsx	46	1020	0,0450980
src/templates/Profile/ResetPasswordButton.tsx	31	1020	0,0303921
src/templates/UserList/index.tsx	307	1020	0,3009803
src/templates/UserList/PasswordResetButton.tsx	46	1020	0,0450980
src/theme/index.ts	251	1020	0,2460784
src/types/app.ts	105	1020	0,1029411
src/types/auth.ts	16	1020	0,0156862
src/types/consumerUnit.ts	115	1020	0,1127450
src/types/energyBill.ts	95	1020	0,0931372
src/types/institution.ts	41	1020	0,0401960
src/types/person.ts	63	1020	0,0617647
src/types/recommendation.ts	103	1020	0,1009803
src/types/tariffs.tsx	63	1020	0,0617647
src/api/index.tsx	494	1020	0,4843137
src/middleware.ts	10	1020	0,0098039
src/store/appSlice.ts	381	1020	0,3735294
src/utils/date.ts	62	1020	0,0607843
src/utils/head.ts	10	1020	0,0098039
src/utils/number.ts	23	1020	0,0225490

E com os valores de CR_i , MC_i e MS_i , foi possível estabelecer a Probabilidade de Falha do Método, de acordo com a [Equação 4.4](#) e ilustrado na [Tabela 4](#).

Tabela 4 – Probabilidade de Falha do Método

Arquivo	MFL_i
src/components/Card/index.tsx	0,0007724
src/components/Card/Wrapper.tsx	0
src/components/ConsumerUnit/Card.tsx	0,0078233
src/components/ConsumerUnit/Content/Analysis AndRecommendation.tsx	0,0110925
src/components/ConsumerUnit/Content/Dropdown Section.tsx	0,0002036
src/components/ConsumerUnit/Form/Create.tsx	0,2257148
src/components/ConsumerUnit/Form/Edit.tsx	0,5747177
src/components/ConsumerUnit/Form/Renew Contract.tsx	0,2890522
src/components/ConsumerUnit/Form/Warning Dialog.tsx	0,0001197
src/components/Distributor/DistributorCard.tsx	0,0020833
src/components/Distributor/Form/CreateForm.tsx	0,0023042
src/components/Distributor/Form/Distributor CreateForm.tsx	0,0320150
src/components/Distributor/Form/Distributor EditForm.tsx	0,0297385
src/components/Drawer/index.tsx	0,0202020
src/components/Drawer/ListItem.tsx	0,0002506
src/components/ElectricityBill/Form/CreateEdit ElectricityBillForm.tsx	0,7219251
src/components/Footer/index.tsx	0,0000170
src/components/Form/DrawerV2.tsx	0,0000965
src/components/Institution/Form/CreateInstitution Form.tsx	0,0025048
src/components/Institution/Form/EditInstitution Form.tsx	0,0061868
src/components/Person/Form/CreatePersonForm.tsx	0,0147739
src/components/Person/Form/EditPersonForm.tsx	0,0064208
src/components/Tariff/Form/TariffCreateForm.tsx	0,8933006
src/pages/_document.tsx	0,0000984
src/pages/api/auth/[...nextauth].ts	0,0002256
src/pages/distribuidoras/[distributorId].tsx	0,0005242
src/pages/distribuidoras/index.tsx	0,0000649

Arquivo	MFL_i
src/pages/pessoas.tsx	0,0000043
src/pages/uc/[id].tsx	0,0005242
src/pages/uc/index.tsx	0,0001225
src/templates/Analysis/BaseCostComparison Card.tsx	0,0017528
src/templates/Analysis/BaseCostComparison Plot.tsx	0,0012477
src/templates/Analysis/CurrentBaseCostPlot.tsx	0,0020276
src/templates/Analysis/DetailedAnalysisDrawer/ BaseCostComparisonTable.tsx	0,0008244
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumerUnitInfo.tsx	0,0000259
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumptionHistoryTable.tsx	0,0003639
src/templates/Analysis/DetailedAnalysisDrawer/ ContractsComparisonTable.tsx	0,0007612
src/templates/Analysis/DetailedAnalysisDrawer/ CurrentContractTable.tsx	0,0033190
src/templates/Analysis/DetailedAnalysisDrawer/ DetailedBaseCostsComparisonPlot.tsx	0,0014120
src/templates/Analysis/DetailedAnalysisDrawer/ index.tsx	0,0833333
src/templates/Analysis/DetailedAnalysisDrawer/ Logos.tsx	0,0000086
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractDemandPlot.tsx	0,0020424
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractTable.tsx	0,0018270
src/templates/Analysis/DetailedAnalysisDrawer/ TariffsTable.tsx	0,0009618
src/templates/Analysis/DetailedAnalysisHeader.tsx	0,0000086
src/templates/Analysis/Measured ConsumptionPlot.tsx	0,0014569
src/templates/Analysis/Measured DemandPlot.tsx	0,0024237
src/templates/Analysis/RecommendationCard.tsx	0,0004208
src/templates/Auth/SignIn.tsx	0,0048741

Archivo	MFL_i
src/templates/ConsumerUnit/Content/Contract.tsx	0,0026416
src/templates/ConsumerUnit/Content/Header.tsx	0,0120370
src/templates/ConsumerUnit/Content/index.tsx	0,0008680
src/templates/ConsumerUnit/Content/Invoice/ Filter.tsx	0,0015968
src/templates/ConsumerUnit/Content/Invoice/ Table.tsx	0,0383104
src/templates/ConsumerUnit/Grid.tsx	0,0004889
src/templates/Dashboard/Grid.tsx	0,0026905
src/templates/Dashboard/index.tsx	0,0000572
src/templates/DefaultV2.tsx	0,0000340
src/templates/Distributor/Content/Consumer UnitsList.tsx	0,0003509
src/templates/Distributor/Content/index.tsx	0,0013638
src/templates/Distributor/Content/TariffsTable.tsx	0,0077957
src/templates/Distributor/Header/index.tsx	0,0233121
src/templates/Institution/index.tsx	0,0002970
src/templates/Profile/index.tsx	0,0000854
src/templates/Profile/ResetPasswordButton.tsx	0,0000383
src/templates/UserList/index.tsx	0,0475032
src/templates/UserList/PasswordResetButton.tsx	0,0001138
src/theme/index.ts	0
src/types/app.ts	0
src/types/auth.ts	0
src/types/consumerUnit.ts	0
src/types/energyBill.ts	0
src/types/institution.ts	0
src/types/person.ts	0
src/types/recommendation.ts	0
src/types/tariffs.tsx	0
src/api/index.tsx	0
src/middleware.ts	0
src/store/appSlice.ts	0,0577744
src/utills/date.ts	0,0018419
src/utills/head.ts	0,0000061
src/utills/number.ts	0,0000355

Considerando o caráter multiplicativo da [Equação 4.5](#) e os valores nulos de alguns arquivos é possível compreender o motivo pelo qual alguns itens apresentam Probabilidade de Falha igual a zero.

5.1.3 Coleta dos Impactos de Falha dos Métodos

De modo a ter uma maior confiança nos dados coletados, foi utilizado uma ferramenta de análise de projetos feitos em *React: React-Scanner*⁴, para realizar a contagem de instanciações de cada componente. Considerando que alguns dos arquivos avaliados tiveram seus respectivos $MFL_i = 0$, foi descartado a análise de impacto de falha dos métodos para estes arquivos visando a otimização do processo.

E com os valores de MI_i e MI_{ij} foi possível calcular a MFI_i , de acordo com a [Equação 4.5](#) e ilustrado na [Tabela 5](#).

Tabela 5 – Impacto de Falha dos Métodos

Arquivo	MI_i	MI_{ij}	NMC	$\max \forall MI_i$	MFI_I
src/components/Card/index.tsx	2	6	3	5	0,4000000
src/components/Card/Wrapper.tsx	*	*	*	*	*
src/components/ConsumerUnit/Card.tsx	2	2	1	5	0,4000000
src/components/ConsumerUnit/Content/AnalysisAndRecommendation.tsx	5	5	1	5	1
src/components/ConsumerUnit/Content/DropdownSection.tsx	5	31	7	5	0,8857142
src/components/ConsumerUnit/Form/Create.tsx	4	14	2	5	1,4000000
src/components/ConsumerUnit/Form/Edit.tsx	4	4	1	5	0,8000000
src/components/ConsumerUnit/Form/RenewContract.tsx	3	3	1	5	0,6000000
src/components/ConsumerUnit/Form/WarningDialog.tsx	1	16	5	5	0,6400000
src/components/Distributor/DistributorCard.tsx	2	2	1	5	0,4000000
src/components/Distributor/Form/CreateForm.tsx	4	9	3	5	0,6000000

⁴ <https://www.npmjs.com/package/react-scanner>

Arquivo	MI_i	MI_{ij}	NMC	$\max \forall MI_i$	MFI_I
src/components/Distributor/Form/DistributorCreateForm.tsx	4	4	1	5	0,8000000
src/components/Distributor/Form/DistributorEditForm.tsx	4	14	2	5	1,4000000
src/components/Drawer/index.tsx	3	13	2	5	1,3000000
src/components/Drawer/ListItem.tsx	3	12	3	5	0,8000000
src/components/ElectricityBill/Form/CreateEditElectricityBillForm.tsx	4	14	2	5	1,4000000
src/components/Footer/index.tsx	1	18	5	5	0,7200000
src/components/Form/DrawerV2.tsx	3	50	12	5	0,8333333
src/components/Institution/Form/CreateInstitutionForm.tsx	4	4	1	5	0,8000000
src/components/Institution/Form/EditInstitutionForm.tsx	4	4	1	5	0,8000000
src/components/Person/Form/CreatePersonForm.tsx	4	4	1	5	0,8000000
src/components/Person/Form/EditPersonForm.tsx	4	4	1	5	0,8000000
src/components/Tariff/Form/TariffCreateForm.tsx	4	4	1	5	0,8000000
src/pages/_document.tsx	5	5	1	5	1
src/pages/api/auth/[...nextauth].ts	5	5	1	5	1
src/pages/distribuidoras/[distributorId].tsx	5	5	1	5	1
src/pages/distribuidoras/index.tsx	5	5	1	5	1
src/pages/pessoas.tsx	5	5	1	5	1
src/pages/uc/[id].tsx	5	5	1	5	1
src/pages/uc/index.tsx	5	5	1	5	1
src/templates/Analysis/BaseCostComparisonCard.tsx	5	5	1	5	1
src/templates/Analysis/BaseCostComparisonPlot.tsx	5	5	1	5	1
src/templates/Analysis/CurrentBaseCostPlot.tsx	5	15	2	5	1,5000000

Arquivo	MI_i	MI_{ij}	NMC	$\max \forall MI_i$	MFI_I
src/templates/Analysis/Detailed AnalysisDrawer/BaseCost ComparisonTable.tsx	5	5	0	5	0
src/templates/Analysis/Detailed AnalysisDrawer/ConsumerUnit Info.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/Consumption HistoryTable.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/Contracts ComparisonTable.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/CurrentContract Table.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/DetailedBase CostsComparisonPlot.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/index.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/Logos.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/Recommended ContractDemandPlot.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/Recommended ContractTable.tsx	5	5	1	5	1
src/templates/Analysis/Detailed AnalysisDrawer/TariffsTable.tsx	5	10	4	5	0,5000000
src/templates/Analysis/Detailed AnalysisHeader.tsx	5	5	1	5	1
src/templates/Analysis/Measured ConsumptionPlot.tsx	5	5	1	5	1
src/templates/Analysis/Measured DemandPlot.tsx	5	5	1	5	1

Arquivo	MI_i	MI_{ij}	NMC	$\max \forall MI_i$	MFI_I
src/templates/Analysis/RecommendationCard.tsx	5	5	1	5	1
src/templates/Auth/SignIn.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Content/Contract.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Content/Header.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Content/index.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Content/Invoice/Filter.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Content/Invoice/Table.tsx	5	5	1	5	1
src/templates/ConsumerUnit/Grid.tsx	5	5	1	5	1
src/templates/Dashboard/Grid.tsx	5	5	1	5	1
src/templates/Dashboard/index.tsx	5	5	1	5	1
src/templates/DefaultV2.tsx	5	45	8	5	1,1250000
src/templates/Distributor/Content/ConsumerUnitsList.tsx	5	5	1	5	1
src/templates/Distributor/Content/index.tsx	5	5	1	5	1
src/templates/Distributor/Content/TariffsTable.tsx	5	5	1	5	1
src/templates/Distributor/Header/index.tsx	5	5	1	5	1
src/templates/Institution/index.tsx	5	5	1	5	1
src/templates/Profile/index.tsx	5	5	1	5	1
src/templates/Profile/ResetPasswordButton.tsx	5	5	1	5	1
src/templates/UserList/index.tsx	5	5	1	5	1
src/templates/UserList/PasswordResetButton.tsx	5	5	1	5	1
src/theme/index.ts	*	*	*	*	*
src/types/app.ts	*	*	*	*	*

Arquivo	MI_i	MI_{ij}	NMC	$max\forall MI_i$	MFI_I
src/types/auth.ts	*	*	*	*	*
src/types/consumerUnit.ts	*	*	*	*	*
src/types/energyBill.ts	*	*	*	*	*
src/types/institution.ts	*	*	*	*	*
src/types/person.ts	*	*	*	*	*
src/types/recommendation.ts	*	*	*	*	*
src/types/tariffs.tsx	*	*	*	*	*
src/api/index.tsx	*	*	*	*	*
src/middleware.ts	*	*	*	*	*
src/store/appSlice.ts	3	305	43	5	1,4186046
src/utills/date.ts	1	41	16	5	0,5125000
src/utills/head.ts	1	2	7	5	0,0571428
src/utills/number.ts	1	2	6	5	0,0666666

5.1.4 Cálculo do Valor de Risco do Método

Em posse de todos os valores analisados, torna-se possível calcular o valor de risco associado a cada componente, de acordo com [Equação 4.6](#) e ilustrado na [Tabela 6](#).

Tabela 6 – Valor de Risco do Método

Arquivo	MRV_i
src/components/Card/index.tsx	0,0003089
src/components/Card/Wrapper.tsx	0
src/components/ConsumerUnit/Card.tsx	0,0031293
src/components/ConsumerUnit/Content/Analysis AndRecommendation.tsx	0,0110925
src/components/ConsumerUnit/Content/Dropdown Section.tsx	0,0001803
src/components/ConsumerUnit/Form/Create.tsx	0,3160008
src/components/ConsumerUnit/Form/Edit.tsx	0,4597742
src/components/ConsumerUnit/Form/Renew Contract.tsx	0,1734313
src/components/ConsumerUnit/Form/Warning Dialog.tsx	0,0000766
src/components/Distributor/DistributorCard.tsx	0,0008333
src/components/Distributor/Form/CreateForm.tsx	0,0013825

Arquivo	MRV_i
src/components/Distributor/Form/Distributor CreateForm.tsx	0,0256120
src/components/Distributor/Form/Distributor EditForm.tsx	0,0416339
src/components/Drawer/index.tsx	0,0262626
src/components/Drawer/ListItem.tsx	0,0002005
src/components/ElectricityBill/Form/CreateEdit ElectricityBillForm.tsx	1,0106951
src/components/Footer/index.tsx	0,0000122
src/components/Form/DrawerV2.tsx	0,0000804
src/components/Institution/Form/CreateInstitution Form.tsx	0,0020038
src/components/Institution/Form/EditInstitution Form.tsx	0,0049494
src/components/Person/Form/CreatePersonForm.tsx	0,0118191
src/components/Person/Form/EditPersonForm.tsx	0,0051366
src/components/Tariff/Form/TariffCreateForm.tsx	0,7146405
src/pages/_document.tsx	0,0000984
src/pages/api/auth/[...nextauth].ts	0,0002256
src/pages/distribuidoras/[distributorId].tsx	0,0005242
src/pages/distribuidoras/index.tsx	0,0000649
src/pages/pessoas.tsx	0,0000043
src/pages/uc/[id].tsx	0,0005242
src/pages/uc/index.tsx	0,0001225
src/templates/Analysis/BaseCostComparison Card.tsx	0,0017528
src/templates/Analysis/BaseCostComparison Plot.tsx	0,0012477
src/templates/Analysis/CurrentBaseCostPlot.tsx	0,0030414
src/templates/Analysis/DetailedAnalysisDrawer/ BaseCostComparisonTable.tsx	0
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumerUnitInfo.tsx	0,0000259
src/templates/Analysis/DetailedAnalysisDrawer/ ConsumptionHistoryTable.tsx	0,0003639
src/templates/Analysis/DetailedAnalysisDrawer/ ContractsComparisonTable.tsx	0,0007612

Archivo	MRV_i
src/templates/Analysis/DetailedAnalysisDrawer/ CurrentContractTable.tsx	0,0033190
src/templates/Analysis/DetailedAnalysisDrawer/ DetailedBaseCostsComparisonPlot.tsx	0,0014120
src/templates/Analysis/DetailedAnalysisDrawer/ index.tsx	0,0833333
src/templates/Analysis/DetailedAnalysisDrawer/ Logos.tsx	0,0000086
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractDemandPlot.tsx	0,0020424
src/templates/Analysis/DetailedAnalysisDrawer/ RecommendedContractTable.tsx	0,0018270
src/templates/Analysis/DetailedAnalysisDrawer/ TariffsTable.tsx	0,0004809
src/templates/Analysis/DetailedAnalysisHeader.tsx	0,0000086
src/templates/Analysis/Measured ConsumptionPlot.tsx	0,0014569
src/templates/Analysis/Measured DemandPlot.tsx	0,0024237
src/templates/Analysis/RecommendationCard.tsx	0,0004208
src/templates/Auth/SignIn.tsx	0,0048741
src/templates/ConsumerUnit/Content/Contract.tsx	0,0026416
src/templates/ConsumerUnit/Content/Header.tsx	0,0120370
src/templates/ConsumerUnit/Content/index.tsx	0,0008680
src/templates/ConsumerUnit/Content/Invoice/ Filter.tsx	0,0015968
src/templates/ConsumerUnit/Content/Invoice/ Table.tsx	0,0383104
src/templates/ConsumerUnit/Grid.tsx	0,0004889
src/templates/Dashboard/Grid.tsx	0,0026905
src/templates/Dashboard/index.tsx	0,0000572
src/templates/DefaultV2.tsx	0,0000382
src/templates/Distributor/Content/Consumer UnitsList.tsx	0,0003509
src/templates/Distributor/Content/index.tsx	0,0013638
src/templates/Distributor/Content/TariffsTable.tsx	0,0077957
src/templates/Distributor/Header/index.tsx	0,0233121

Arquivo	MRV_i
src/templates/Institution/index.tsx	0,0002970
src/templates/Profile/index.tsx	0,0000854
src/templates/Profile/ResetPasswordButton.tsx	0,0000383
src/templates/UserList/index.tsx	0,0475032
src/templates/UserList/PasswordResetButton.tsx	0,0001138
src/theme/index.ts	0
src/types/app.ts	0
src/types/auth.ts	0
src/types/consumerUnit.ts	0
src/types/energyBill.ts	0
src/types/institution.ts	0
src/types/person.ts	0
src/types/recommendation.ts	0
src/types/tariffs.tsx	0
src/api/index.tsx	0
src/middleware.ts	0
src/store/appSlice.ts	0,0819590
src/utills/date.ts	0,0009439
src/utills/head.ts	0,0000003
src/utills/number.ts	0,0000023

5.1.5 Priorização dos Casos de Teste

Em posse dos valores de MRV_i e considerando os casos de testes descritos no Apêndice A, é possível determinar o valor de risco de cada caso de teste, conforme Equação 4.7, e através da ordenação decrescente baseada em T_k tem-se a priorização dos casos de testes baseado em risco, conforme ilustrado na Tabela 7.

Tabela 7 – Cenários de Teste Priorizados

Nome do Teste	T_k
TC18 - Acessar Análise da Unidade Consumidora	1,2422696
TC23 - Análise Detalhada	1,2422696
TC08 - Login bem sucedido	1,1592843
TC20 - Acessar Contrato da Unidade Consumidora	1,1401894
TC22 - Acessar Unidade Consumidora	1,1401894
TC16 - Adição de Tarifa	1,1401894
TC19 - Editar Contrato de Unidade Consumidora	1,1401894

Tabela 7 – Cenários de Teste Priorizados

Nome do Teste	T_k
TC17 - Tentativa de Adição de Tarifa	1,1401894
TC21 - Tentativa de Editar Contrato de Unidade Consumidora	1,1401894
TC01 - Cadastrar Distribuidora	0,9323492
TC02 - Editar Distribuidora	0,9323492
TC03 - Tentativa de Cadastrar Distribuidora	0,9323492
TC04 - Tentativa de Editar Distribuidora	0,9323492
TC11 - Acessar Página de Pessoas	0,1794210
TC12 - Cadastro de Pessoa pelo adm geral	0,1794210
TC14 - Cadastro de Pessoa pelo gestor	0,1794210
TC13 - Tentativa de Cadastro de Pessoa pelo adm geral	0,1794210
TC15 - Tentativa de Cadastro de Pessoa pelo gestor	0,1794210
TC05 - Acessar Página de Instituições	0,1245473
TC06 - Cadastro de Instituição	0,1245473
TC07 - Tentativa de Cadastro de Instituição	0,1245473
TC10 - Acessar Painel	0,0879478
TC09 - Tentativa de login	0,0879478

5.1.6 Análise dos Resultados Obtidos

Nota-se a repetição de alguns valores para T_k , essa repetição é causada pela estrutura arquitetural de implementações que utiliza o *framework React*, onde todos os componentes instanciados serão carregados inicialmente, mesmo que não sejam exibidos (Meta Open Source, 2023). Além de questões estruturais do código fonte, é possível verificar que as telas de cadastro e edição de dados, são modais instanciadas junto das telas principais das suas respectivas funcionalidades, implicando na repetição dos valores e conseqüentemente na repetição de T_k para testes que acessam uma página e realizam cadastros ou edição de dados.

As adaptações das formas de coleta de dados para complexidade e a abstração considerada para *métodos = componentes*, foram essenciais para garantir a assertividade dos valores coletados, e conseqüentemente das probabilidades. A utilização de softwares voltados para análise do código fonte foram cruciais para garantir a precisão dos dados quantitativos.

Os dados qualitativos coletados para a aplicação do método e o caráter multiplicativo das fórmulas empregadas, podem ocasionar em avaliações e priorizações incorretas caso sejam realizadas por alguém sem conhecimento prévio do sistema, de suas funcionalidades e dos requisitos a qual ele atende.

Observou-se através dos dados obtidos que os índices CR , MC e MS não estão diretamente relacionados, onde uma alteração de poucas linhas de código pode resultar em grandes mudanças nos valores de complexidade.

O teste $TC08$ - *Login bem sucedido* apresentou valores incomuns para T_k caso seja considerado apenas o ato de autenticar-se no sistema. Entretanto sua verificação acontece na página inicial após a renderização de todos os componentes associados à mesma, ocasionando no aumento das páginas cobertas por este teste e consequentemente seu valor associado.

É possível notar que os testes candidatos a serem priorizados, segundo o valor de T_k , coincidiram com os testes que verificam as funcionalidades principais do código fonte avaliado.

6 Conclusão

Este trabalho reforça a importância e a eficácia da priorização de casos de teste em um ambiente de desenvolvimento de software. O estudo de caso aplicado ao projeto MEC-Energia demonstrou que a priorização de testes baseados na análise de risco pode otimizar o processo de priorização de testes, reduzindo custos e esforços sem comprometer a qualidade do produto final. A metodologia adotada permitiu identificar quais cenários de teste eram mais relevantes no contexto de risco associado à implementações incrementais e, portanto, deveriam ser priorizados, garantindo que os defeitos fossem detectados de forma precoce e mais eficiente.

Além disso, o trabalho evidenciou a possibilidade de adaptações necessárias a cada aplicação do métodos de priorização em projetos reais, especialmente em contextos onde a automação de testes pode ser implementada.

Foi possível, através do estudo de caso realizado, verificar que o método de priorização de casos de testes foi capaz de identificar casos de testes importantes para a automação de testes de regressão, de modo que seja possível utilizá-lo como ferramenta de suporte na priorização de casos de teste. Vale ressaltar que o método de priorização de testes não deve substituir a análise de uma pessoa que possua experiência no sistema a ser testado.

Considerando a aplicabilidade e a possibilidade de adaptação, acredita-se que o método implementado possui viabilidade de utilização no contexto do desenvolvimento ágil, onde será possível executá-lo periodicamente de acordo com o tamanho da *sprint* e implementações realizadas.

É possível observar que todos os objetivos específicos propostos no [Capítulo 1](#) foram alcançados, sendo assim, é possível considerar que o trabalho cumpriu seu Objetivo Geral.

Em pesquisas futuras, pode-se implementar a priorização de casos de testes em múltiplas versões de um software e avaliar, com especialistas da área de qualidade de software, se o resultado obtido condiz com o esperado. A implementação de um método, ou sistema, que possibilite os cálculos e a análise de forma automatizada também podem ser considerados.

Referências

- AMANNEJAD, Y. et al. A search-based approach for cost-effective software test automation decision support and an industrial case study. In: *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops*. [S.l.: s.n.], 2014. p. 302–311. Citado na página 13.
- AMANNEJAD, Y. et al. A search-based approach for cost-effective software test automation decision support and an industrial case study. In: *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops*. [S.l.: s.n.], 2014. p. 302–311. Citado na página 14.
- AZIZI, M. A tag-based recommender system for regression test case prioritization. In: *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. [S.l.: s.n.], 2021. p. 146–157. Citado 3 vezes nas páginas 14, 22 e 65.
- BRERETON, P. et al. Using a protocol template for case study planning. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swindon, GBR: BCS Learning & Development Ltd., 2008. (EASE'08), p. 41–48. Citado 2 vezes nas páginas 15 e 25.
- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior. 2021. Disponível em: <https://www.gov.br/capes/pt-br/centrais-de-conteudo/documentos/avaliacao/09012022_RELATORIOQUALISEVENTOS20172020COMPUTACAO.PDF>. Citado na página 65.
- Cucumber, SmartBear Software. *Gherkin Reference*. 2019. Disponível em: <<https://cucumber.io/docs/gherkin/reference/>>. Citado na página 27.
- DELAMARO, M. *Introdução ao Teste de Software*. [S.l.]: Grupo GEN, 2016. Citado 3 vezes nas páginas 17, 18 e 20.
- FROST, B. *Atomic design*. [S.l.]: Brad Frost Pittsburgh, 2016. Citado na página 32.
- GAROUSI, V.; FELDERER, M. Developing, verifying, and maintaining high-quality automated test scripts. v. 33, n. 3, p. 68–75, 2016. ISSN 1937-4194. Conference Name: IEEE Software. Citado na página 21.
- HAAS, R. et al. How can manual testing processes be optimized? developer survey, optimization guidelines, and case studies. In: . New York, NY, USA: Association for Computing Machinery, 2021. (ESEC/FSE 2021), p. 1281–1291. ISBN 9781450385626. Disponível em: <<https://doi.org/10.1145/3468264.3473922>>. Citado 3 vezes nas páginas 14, 21 e 65.
- JAHAN, H.; FENG, Z.; MAHMUD, S. M. H. Risk-based test case prioritization by correlating system methods and their associated risks. *Arabian Journal for Science and Engineering*, v. 45, 04 2020. Citado 6 vezes nas páginas 14, 16, 23, 24, 28 e 65.

- KARHU, K. et al. Empirical observations on software testing automation. In: *2009 International Conference on Software Testing Verification and Validation*. IEEE, 2009. p. 201–209. ISBN 978-1-4244-3775-7. Disponível em: <<http://ieeexplore.ieee.org/document/4815352/>>. Citado 2 vezes nas páginas 14 e 21.
- MATEEN, A.; ZHU, Q.; AFSAR, S. Comparative analysis of manual vs automotive testing for software quality. In: *Proceedings of the 7th International Conference on Software Engineering and New Technologies*. ACM, 2018. p. 1–7. ISBN 978-1-4503-6101-9. Disponível em: <<https://dl.acm.org/doi/10.1145/3330089.3330121>>. Citado na página 13.
- Meta Open Source. *Renderizar e Confirmar*. 2023. Disponível em: <<https://pt-br.react.dev/learn/render-and-commit>>. Citado na página 54.
- MYERS, G. J.; SANDLER, C.; BADGETT, T. *The art of software testing*. 3. ed. Nashville, TN: John Wiley & Sons, 2011. Citado 4 vezes nas páginas 13, 17, 18 e 20.
- OLIINYK, B.; OLEKSIUK, V. Automation in software testing, can we automate anything we want? p. 11, 2019. Citado 5 vezes nas páginas 13, 14, 21, 22 e 65.
- RAMLER, R.; WOLFMAIER, K. Economic perspectives in test automation: Balancing automated and manual testing with opportunity cost. In: *Proceedings of the 2006 International Workshop on Automation of Software Test*. New York, NY, USA: Association for Computing Machinery, 2006. (AST '06), p. 85–91. ISBN 1595934081. Disponível em: <<https://doi.org/10.1145/1138929.1138946>>. Citado na página 13.
- SAHAF, Z. et al. When to automate software testing? decision support based on system dynamics: an industrial case study. In: *Proceedings of the 2014 International Conference on Software and System Process - ICSSP 2014*. ACM Press, 2014. p. 149–158. ISBN 978-1-4503-2754-1. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2600821.2600832>>. Citado 3 vezes nas páginas 13, 14 e 21.
- SpringerLink. *Arabian Journal for Science and Engineering Impact Factor*. 2023. Disponível em: <<https://link.springer.com/journal/13369>>. Citado na página 65.
- VELASCO, L.; CORAL, R. Mec-energia. <<https://lappis-unb.gitlab.io/projetos-energia/mec-energia/documentacao/>> [Acessado em: 25 de agosto de 2024], 2023. Citado na página 32.
- XU, L.; DIAS, M.; RICHARDSON, D. Generating regression tests via model checking. In: *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004*. [S.l.: s.n.], 2004. p. 336–341 vol.1. Citado na página 20.

Anexos

ANEXO A – Cenários de Testes

Cenários de Testes Identificados para o MEC - Energia

Tabela 8 – Cadastrar Distribuidora

TC01 - Cadastrar Distribuidora	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Distribuidora
Quando	clicar em "+ Distribuidora"
E	preencher o formulário com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Distribuidora adicionada com sucesso!" deve ser exibida

Tabela 9 – Editar Distribuidora

TC02 - Editar Distribuidora	
Dado	que esteja logado como "usuario@unb"
E	esteja na tela de Distribuidora
Quando	clicar em "Editar"
E	preencher com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Distribuidora modificada com sucesso!" deve ser exibida

Tabela 10 – Tentativa de Cadastrar Distribuidora

TC03 - Tentativa de Cadastrar Distribuidora	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Distribuidora
Quando	clicar em "+ Distribuidora"
E	preencher o formulário com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 11 – Tentativa de Editar Distribuidora

TC04 - Tentativa de Editar Distribuidora	
Dado	que esteja logado como "usuario@unb"
E	esteja na tela de Distribuidora
Quando	clicar em "Editar"
E	preencher com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 12 – Acessar Página de Instituições

TC05 - Acessar Página de Instituições	
Dado	que esteja logado como "admin@admin"
Quando	acessar a página "Instituições"
Então	a listagem de instituições deve ser exibida

Tabela 13 – Cadastro de Instituição

TC06 - Cadastro de Instituição	
Dado	que esteja logado como "admin@admin"
E	esteja na tela de Instituições
Quando	clicar em "+ Instituições"
E	preencher o formulário com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Instituição adicionada com sucesso!" deve ser exibida

Tabela 14 – Tentativa de Cadastro de Instituição

TC07 - Tentativa de Cadastro de Instituição	
Dado	que esteja logado como "admin@admin"
E	esteja na tela de Instituições
Quando	clicar em "+ Instituições"
E	preencher o formulário com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 15 – Login bem sucedido

TC08 - Login bem sucedido	
Dado	a existência usuário "admin@admin" válido
E	esteja na tela de login
Quando	preencher com credencias válidas
E	clicar em "Logar"
Então	o login deve ser realizado

Tabela 16 – Tentativa de login

TC09 - Tentativa de login	
Dado	a existência usuário "admin@admin" válido
E	esteja na tela de login
Quando	preencher com credencias inválidas
E	clicar em "Logar"
Então	a mensagem de erro deve ser exibida

Tabela 17 – Acessar Painel

TC10 - Acessar Painel	
Dado	que esteja logado como "admin@unb"
Quando	acessar a página "Painel"
Então	o painel com as Unidades Consumidoras e Distribuidoras deve ser exibido

Tabela 18 – Acessar Página de Pessoas

TC11 - Acessar Página de Pessoas	
Dado	que esteja logado como "admin@admin"
Quando	acessar a página "Pessoas"
Então	a listagem de pessoas deve ser exibida

Tabela 19 – Cadastro de Pessoa pelo adm geral

TC12 - Cadastro de Pessoa pelo adm geral	
Dado	que esteja logado como "admin@admin"
E	esteja na tela de Pessoas
Quando	clicar em "+ Pessoa"
E	preencher o formulário com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Pessoa cadastrada com sucesso!" deve ser exibida

Tabela 20 – Tentativa de Cadastro de Pessoa pelo adm geral

TC13 - Tentativa de Cadastro de Pessoa pelo adm geral	
Dado	que esteja logado como "admin@admin"
E	esteja na tela de Pessoas
Quando	clicar em "+ Pessoa"
E	preencher o formulário com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 21 – Cadastro de Pessoa pelo gestor

TC14 - Cadastro de Pessoa pelo gestor	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Pessoas
Quando	clicar em "+ Pessoa"
E	preencher o formulário com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Pessoa cadastrada com sucesso!" deve ser exibida

Tabela 22 – Tentativa de Cadastro de Pessoa pelo gestor

TC15 - Tentativa de Cadastro de Pessoa pelo gestor	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Pessoas
Quando	clicar em "+ Pessoa"
E	preencher o formulário com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 23 – Adição de Tarifa

TC16 - Adição de Tarifa	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Unidade Consumidora
Quando	clicar em Lançar Fatura
E	preencher o formulário com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Fatura lançada com sucesso!" deve ser exibida

Tabela 24 – Tentativa de Adição de Tarifa

TC17 - Tentativa de Adição de Tarifa	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Unidade Consumidora
Quando	clicar em Lançar Fatura
E	preencher o formulário com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 25 – Acessar Análise da Unidade Consumidora

TC18 - Acessar Análise da Unidade Consumidora	
Dado	que esteja logado como "admin@unb"
Quando	acessar a página de Análise de uma Unidade Consumidora
Então	as os cards Recomendação, Consumo Médio, Demanda Média e Comparativo de Consumo devem ser exibidos

Tabela 26 – Editar Contrato de Unidade Consumidora

TC19 - Editar Contrato de Unidade Consumidora	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Unidade Consumidora
Quando	clicar em "Contrato"
E	clicar em "Editar"
E	preencher com informações válidas
E	clicar em "Gravar"
Então	a mensagem "Unidade consumidora modificada com sucesso!" deve ser exibida

Tabela 27 – Acessar Contrato da Unidade Consumidora

TC20 - Acessar Contrato da Unidade Consumidora	
Dado	que esteja logado como "admin@unb"
Quando	acessar a página de uma Unidade Consumidora
E	clicar em "Contrato"
Então	o contrato da Unidade Consumidora deve ser exibido

Tabela 28 – Tentativa de Editar Contrato de Unidade Consumidora

TC21 - Tentativa de Editar Contrato de Unidade Consumidora	
Dado	que esteja logado como "admin@unb"
E	esteja na tela de Unidade Consumidora
Quando	clicar em "Contrato"
E	clicar em "Editar"
E	preencher com informações inválidas
E	clicar em "Gravar"
Então	a mensagem "Corrija os erros acima antes de gravar" deve ser exibida

Tabela 29 – Acessar Unidade Consumidora

TC22 - Acessar Unidade Consumidora	
Dado	que esteja logado como "admin@unb"
Quando	acessar a página de uma Unidade Consumidora
Então	a listagem das faturas deve ser exibida

Tabela 30 – Análise Detalhada

TC23 - Análise Detalhada	
Dado	que esteja logado como "admin@unb"
Quando	acessar a página de Análise de uma Unidade Consumidora
E	clicar em "Ver análise detalhada"
Então	os itens Recomendação, Consumo Médio, Demanda Média e Comparativo de Consumo devem ser exibidos

ANEXO B – Comparação das Metodologias Identificadas

Tabela 31 – Metodologias Avaliadas

Metodologia	Autores	Metodologia Empregada	Viabilidade da Aplicação	Data	Fator de Impacto	Citações
How Can Manual Testing Processes Be Optimized? Developer Survey, Optimization Guidelines, and Case Studies	HAAS et al.	Diretrizes para auxiliar na identificação de testes candidatos à priorização	Aplicável no objeto de estudo mas não produz insucessos para avaliar o sucesso da aplicação	2021	A1 ¹	14
A Tag-based Recommender System for Regression Test Case Prioritization	AZIZI	Estabelece marcações e realiza consultas baseado nas marcações para selecionar as marcações mais coerentes para a priorização	Aplicável mas a geração das marcas depende exclusivamente do esforço humano não sendo possível uma padronização	2021	B3 ¹	3
Automation in software testing, can we automate anything we want?	OLIINYK; OLEK-SIUK	Categoriza processos que podem ser automatizados e priorizados	Priorização superficial baseado no tipo do componentes testado	2019	-	20
Risk-Based Test Case Prioritization by Correlating System Methods And Their Associated Risks	JAHAN; FENG; MAH-MUD	Coleta de dados mensuráveis e subjetivos para geração de valores que serão responsáveis pela priorização	Aplicável e produz dados mensuráveis e verificáveis	2020	2.5 ²	16

¹ Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (2021)

² SpringerLink (2023)