UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FERNANDO JORGE PEREIRA JÚNIOR

AGILE ETEASY SOFTWARE: UM SISTEMA PARA GUIAR O USO DE TESTES EXPLORATÓRIOS NO CONTEXTO DE METODOLOGIAS ÁGEIS

CAMPINA GRANDE - PB

2022

FERNANDO JORGE PEREIRA JÚNIOR

AGILE ETEASY SOFTWARE: UM SISTEMA PARA GUIAR O USO DE TESTES EXPLORATÓRIOS NO CONTEXTO DE METODOLOGIAS ÁGEIS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Wilkerson de Lucena Andrade

CAMPINA GRANDE - PB

2022

FERNANDO JORGE PEREIRA JÚNIOR

AGILE ETEASY SOFTWARE: UM SISTEMA PARA GUIAR O USO DE TESTES EXPLORATÓRIOS NO CONTEXTO DE METODOLOGIAS ÁGEIS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Wilkerson de Lucena Andrade

Orientador – UASC/CEEI/UFCG

Patricia Duarte de Lima Machado

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

RESUMO

Organizações que trabalham com desenvolvimento de sistemas, sejam elas grandes instituições ou pequenos grupos de desenvolvedores, lidam com o desafio de preservar a qualidade do software, enquanto criam ou evoluem o mesmo. Porém o ritmo com que os requisitos e regras de negócio se transformam, torna essa tarefa árdua e repetitiva, resultando em desperdício de tempo e retrabalho. Esse trabalho tem como objetivo o desenvolvimento de um sistema que ajude as organizações e equipes a realizarem e manterem testes exploratórios em um contexto ágil de maneira consistente, através da aplicação das técnicas e tarefas propostas pelo método Agile ETeasy.

Agile ETeasy Software: Um sistema para guiar o uso de Testes Exploratórios no contexto de Metodologias Ágeis

Fernando Jorge Pereira Júnior
fernando.junior@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

RESUMO

Organizações que trabalham com desenvolvimento de sistemas, sejam elas grandes instituições ou pequenos grupos de desenvolvedores, lidam com o desafio de preservar a qualidade do software, enquanto criam ou evoluem o mesmo. Porém o ritmo com que os requisitos e regras de negócio se transformam, torna essa tarefa árdua e repetitiva, resultando em desperdício de tempo e retrabalho. Esse trabalho tem como objetivo o desenvolvimento de um sistema que ajude as organizações e equipes a realizarem e manterem testes exploratórios em um contexto ágil de maneira consistente, através da aplicação das técnicas e tarefas propostas pelo método Agile ETeasy.

Palavras chave

Desenvolvimento de software, testes, testes exploratórios, metodologias ágeis, scrum.

1 INTRODUÇÃO

O processo de execução de testes exploratórios é uma jornada orientada à descoberta, onde o conhecimento sobre o sistema, suas funcionalidades e suas limitações cresce ao longo da mesma. Este tipo de excursão é um modo de verificar e validar as expectativas do sistema e das funcionalidades. Em razão disso, essa categoria de testes combina perfeitamente com os ambientes ágeis, que necessitam que em um período curto de tempo as funcionalidades sejam exploradas e que suas qualidades sejam garantidas.

Nesta jornada vivenciada por equipes de desenvolvimento ágil, é comum que as equipes não consigam fazer com que os testes acompanhem o ritmo de desenvolvimento, devido à quantidade de esforço necessário para se realizar testes exploratórios e automatizados em ambiente que sofre alterações continuamente [1]. Dessa forma, é de extrema relevância o desenvolvimento de uma ferramenta que permita aos times realizar testes que se aplicam ao contexto dinâmico do mundo ágil e que possibilite às empresas e times de desenvolvimento garantir a qualidade dos softwares que estão sendo desenvolvidos.

2 PROBLEMA

No contexto atual de desenvolvimento de softwares temos que atender demandas cada vez maiores em intervalos de tempo cada vez mais curtos e ainda assim garantir que o trabalho atenda as necessidades estratégicas e técnicas. Uma das soluções para que empresas e times de desenvolvimento continuem atendendo as necessidades do mercado é aplicar metodologias ágeis de gerenciamento de projetos e testes. No entanto, o panorama real é que poucas instituições ou equipes conseguem aplicar metodologias ágeis e testes em conjunto para garantir que o software cumpra os critérios de qualidade e atinja o esperado pelo cliente.

3 SOLUÇÃO

Esse projeto tem por finalidade desenvolver uma plataforma que contribua para a disseminação de conhecimentos, métodos e técnicas de testes exploratórios aplicados ao contexto de equipes ágeis. Disponibilizando funcionalidades baseadas no método Agile ETeasy [2] que permitem o usuário aplicar os testes exploratórios em um ambiente ágil com intuito de melhorar a qualidade dos softwares testados.

O método Agile ETeasy foca na definição de etapas e atividades que facilitam a operacionalização dos testes exploratórios por equipes ágeis. De modo simples, o método tem como principais objetivos:

- Propiciar que Testes Exploratórios sejam realizados no contexto de equipes ágeis,

de modo a facilitar a execução dos testes em um cenário documentado de maneira informal e cuja mudança nos requisitos é frequente.
- Estabelecer uma estratégia de uso comum de Testes Exploratórios entre profissionais e pesquisadores da área.
- Alinhar a compreensão dos pesquisadores na discussão sobre a influência dos Testes Exploratórios durante o desenvolvimento de software.

A definição, artefatos, atividades e informações completas do método podem ser encontradas no site oficial do mesmo.

3.1 Trabalhos Relacionados

No mercado existem algumas ferramentas que buscam reduzir os desafios encontrados ao se executar testes exploratórios, algumas dessas ferramentas são TestRail [3], TestPad [4] e PractiTest [5]. Essas ferramentas disponibilizam funcionalidades básicas na sua versão gratuita mas que tem um certo grau de complexidade e que em alguns casos não conseguem ser implantadas em equipes ágeis. Portanto, o sistema desenvolvido parte de um método validado para times ágeis e busca definir funcionalidades que propiciem os times executarem testes exploratórios para ampliar a qualidade do software desenvolvido.

3.2 Funcionalidades

As principais funcionalidades propostas para o Agile ETeasy Software foram definidas a partir dos artefatos apresentados pelo método e podem ser encontrados no site oficial. As demais funcionalidades foram propostas para simplificar o processo de gerenciamento.

3.2.1 Funcionalidades principais: Gerenciamento de planejamento e Gerenciamento de encerramento.

Gerenciar planejamento: O usuário pode criar, visualizar, editar ou excluir artefatos de planejamentos.

Figura 1: Tela: Cadastrar artefato de planejamento

Figura 2: Tela: Visualização do artefato de planejamento

Gerenciar encerramento: O usuário pode criar, visualizar, editar ou excluir artefatos de encerramento.

Figura 3: Tela: Cadastrar artefato de encerramento.

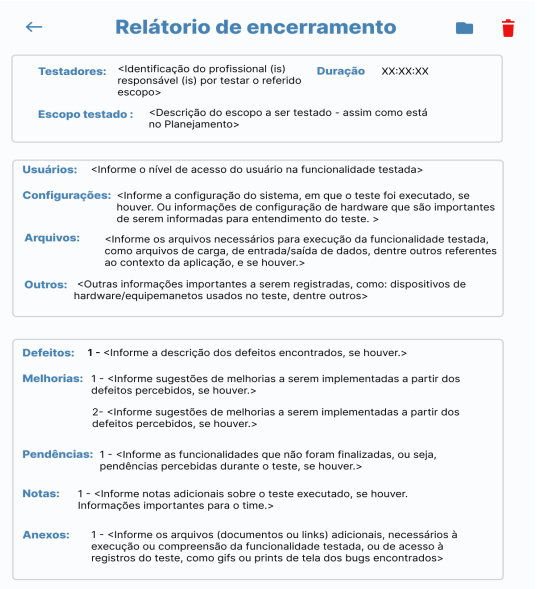

Figura 4: Tela: Visualização do artefato de encerramento

3.2.2 Outras funcionalidades:

Visualizar projetos: O usuário tem acesso a todos projetos criados por ele ou que está vinculado e as informações referentes a esse projeto.

Visualizar membros*:* O usuário tem acesso a todos os usuários vinculados ao seu projeto.

Vincular membro*:* O usuário gera um código de vinculação e disponibiliza para que um usuário cadastrado possa ter acesso ao seu projeto.

Figura 5: Tela: Login

Figura 6: Tela: Cadastro de usuário

4 ARQUITETURA

A arquitetura adotada nesta aplicação foi o modelo cliente-servidor, onde o cliente faz uma requisição ao servidor que retorna uma resposta com os dados solicitados. A aplicação é dividida em duas partes: O frontend, que é a interface da aplicação e assume o papel de cliente da arquitetura, e o backend, que é responsável por aplicar as regras de negócio e persistir os dados e assume o papel de servidor da arquitetura. A comunicação entre ambas é dada pelo modelo API REST.

4.1 Backend

4.1.1 Tecnologias do Backend

O backend foi desenvolvido utilizando o NodeJS, uma plataforma que permite a execução de códigos JavaScript do lado do servidor de forma escalável e eficiente. Esta tecnologia foi escolhida por sua alta flexibilidade e fácil manutenção, garantindo um processo de desenvolvimento mais prático e produtivo.

Para armazenar os dados, foi escolhido o MongoDB, um banco de dados NoSQL que armazena e gerencia informações que são orientadas a documentos, tornando o gerenciamento mais prático pela aplicação.

4.1.2 Estrutura do Backend

Figura 7: Estrutura do backend

A organização das pastas do backend foi feita tendo como base boas práticas adquiridas em experiências de projetos passados, de modo que são separadas em 3 camadas principais: *config, modules e middlewares.*

Os *modules* definem as estruturas das entidades da aplicação que ficam armazenadas no banco de dados.

Os *middlewares* são um conjunto de métodos utilizados como interceptores, eles têm acesso aos objetos de requisição e resposta da chamada, facilitando o tratamento das requisições.

4.2 Frontend

4.2.1 Tecnologias do Frontend

O frontend foi implementado utilizando o React, uma biblioteca JavaScript para criar interfaces de usuário. O React é baseado em componentes, o que torna o desenvolvimento do sistema bastante flexível e modularizado, proporcionando boa reutilização de código. Essa tecnologia foi escolhida por possuir grande afinidade com a mesma.

Para realizar a comunicação com o servidor foi utilizado o Axios, uma biblioteca de código aberto, que pode ser utilizada tanto no navegador quanto no NodeJS. Ela é um cliente HTTP que consegue se comunicar com servidores REST através de requisições HTTP.

4.2.2 Estrutura do Frontend

A estrutura do frontend foi dividida em 4 pastas principais: *components, main, screens e services,* e em 2 pastas auxiliares: *assets e utils*.

Os *components* incluem os componentes presentes nas *pages*, eles foram desenvolvidos a finalidade de reutilizar o código de elementos que se repetem ao longo da aplicação.

As screens incluem todas as páginas presentes na interface da aplicação, onde a cada uma é atribuída uma rota em uma das *routes*.

O *services* incluem os responsáveis por fazer a interação entre a aplicação e os serviços fornecidos pela API da aplicação.

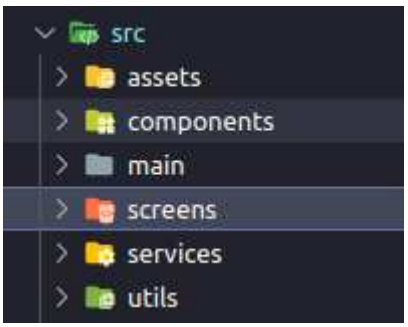

Figura 8: Estrutura do frontend

4.3 Autenticação

A autenticação do sistema, foi implementada por meio do modelo JWT(Json Web Token), que possibilita compartilhar informações de forma compacta e segura através de um objeto JSON. Essa biblioteca codifica um token com as informações desejadas, no contexto da autenticação é utilizado para codificar os dados do usuário que está realizando a requisição.

4.3.1 Fluxo de autenticação

O usuário submete o email e a senha, que faz com que o backend valide a existência de um usuário que possua os mesmos dados, em caso de sucesso, é gerado um token de acesso com informações do usuário e retornado como resposta para o frontend, que armazena o token recebido para inserir no campo Authorization dos headers das requisições que necessitam de autenticação.

5 AVALIAÇÃO

5.1 Deploy

Para dar início ao processo de testes, a aplicação foi hospedada no Heroku, uma plataforma que permite hospedagem, configuração, testagem e publicação de projetos virtuais na nuvem. Por se tratar de uma versão de teste do sistema, essa plataforma foi escolhida por disponibilizar recursos gratuitos que atendem as necessidades da aplicação mesmo possuindo algumas limitações de espaço de armazenamento e tempo de resposta.

5.2 Validação

A validação foi dividida em duas partes, na primeira o processo foi apresentado a duas duplas de um time de desenvolvimento ágil voluntário, onde o método, as técnicas e as tarefas foram explicadas para que cada dupla executasse o método. Para essa avaliação uma dupla executou o método utilizando o sistema e a outra os artefatos padrões, nos permitindo portanto coletar feedback por meio de um formulário contendo perguntas baseadas nas métricas estabelecidas.

O formulário contou com 6 perguntas, uma que coleta o tempo gasto na execução do métodos e outras 5 perguntas com opções de respostas definidas dentro de uma escala numérica de 1 á 5 que representam o intervalo de *Baixa* (1) á *Alta* (5), onde foi possível analisar as seguintes métricas:

- Cobertura de código.
- Cobertura de funcionalidades.
- Cobertura de requisitos.
- Cobertura geral.
- Relevância.

Na segunda parte, para avaliar o Agile ETeasy Software foi utilizado um formulário contendo perguntas baseadas no método da escala Likert, que é uma das metodologias mais populares do mundo quando se trata de pesquisas de satisfação. Essa metodologia oferece uma opção de resposta em uma escala de pontos com descrições verbais, com isso, foram utilizados 5 níveis de intensidade: *concordo totalmente, concordo*, *nem concordo nem discordo, discordo* e *discordo totalmente*.

O formulário contou com 6 perguntas, onde foi possível analisar os seguintes pontos da aplicação:

- Usabilidade.
- Utilidade.
- Interface.
- Comportamento.
- Desempenho.
- Satisfação geral.

5.3 Resultados

5.3.1 Resultados da aplicação do método.

Os resultados coletados apresentam a quantificação das principais métricas relacionadas a aplicação do método.

Tempo de execução: representa a faixa de tempo que os voluntários levam para aplicar o método nos cenários propostos.

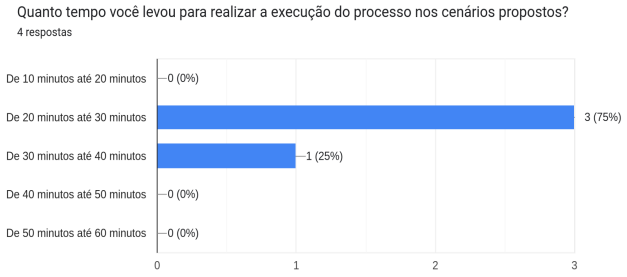

Figura 9: Resultado da pesquisa relacionado ao tempo gasto para aplicar o método.

Cobertura de código: representa a quantidade de código que os voluntários acreditam ter abrangido ao aplicar o método nos cenários propostos.

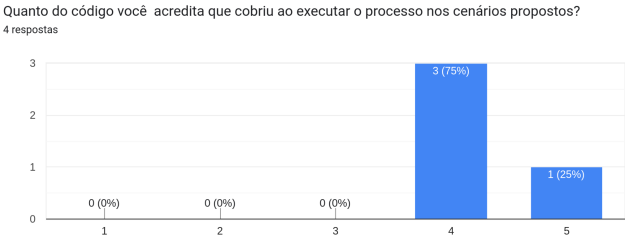

Figura 10: Resultado da pesquisa relacionado a cobertura de código.

Cobertura de requisitos: representa a quantidade de requisitos que os voluntários acreditam ter abrangido ao aplicar o método nos cenários propostos.

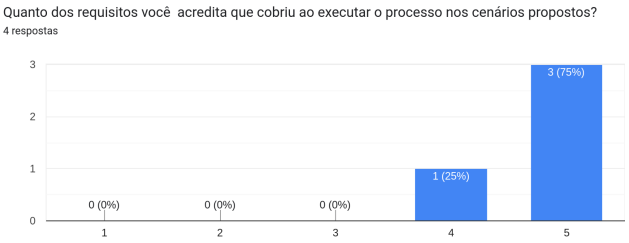

Figura 11: Resultado da pesquisa relacionado a cobertura de requisitos.

Cobertura de funcionalidades: representa a quantidade de requisitos que os voluntários acreditam ter abrangido ao aplicar o método nos cenários propostos.

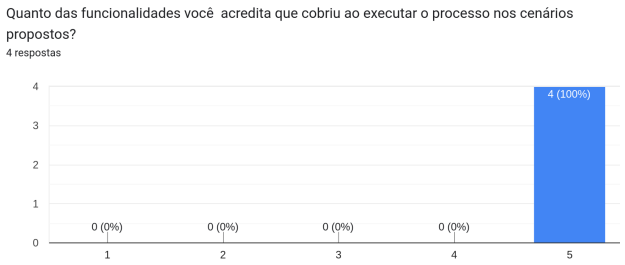

Figura 12: Resultado da pesquisa relacionado a cobertura de requisitos.

Validação geral: representa a quantidade geral de validação que os voluntários acreditam ter alcançado ao aplicar o método nos cenários propostos.

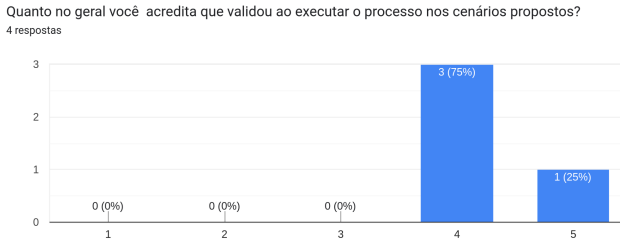

Figura 13: Resultado da pesquisa relacionado a cobertura de geral.

Relevância do método: representa o quanto os voluntários acreditam que o método contribui para a qualidade do software gerado nos cenários propostos.

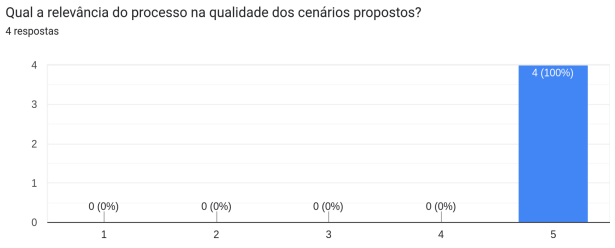

Figura 14: Resultado da pesquisa relacionado a relevância do método.

Em vista dos dados coletados, comprovasse que em pouco tempo e de maneira simples o método contribui de forma significativa para a qualidade dos softwares desenvolvidos e atinge os objetivos de disseminar e aplicar os conhecimentos, métodos e técnicas de testes exploratórios aplicados ao contexto de equipes ágeis.

5.3.2 Resultados da avaliação do sistema.

Os resultados obtidos representam o nível de satisfação dos voluntários que aprenderam o método e aplicaram através do sistema.

Usabilidade: representa a capacidade da aplicação em fazer com que o usuário consiga utilizar suas funcionalidades de maneira fácil e eficiente.

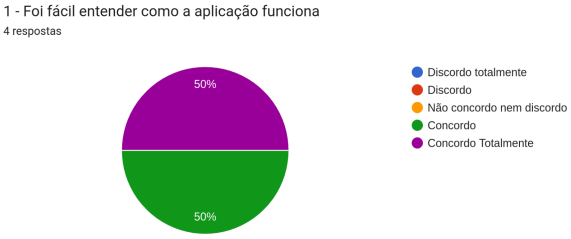

Figura 15: Resultado da pesquisa relacionado à usabilidade.

Utilidade: representa o nível de serventia da aplicação em meio ao contexto proposto.

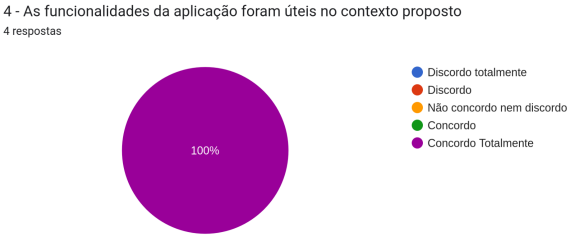

Figura 16: Resultado da pesquisa relacionado à utilidade.

Interface: representa o conjunto de elementos visuais da aplicação e a forma como são estruturados.

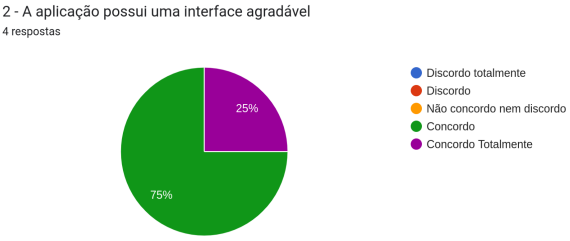

Figura 17: Resultado da pesquisa relacionado à interface.

Comportamento: representa a forma como a aplicação reage às ações do usuário.

Figura 18: Resultado da pesquisa relacionado ao comportamento.

Desempenho: representa a capacidade da aplicação em processar dados e realizar ações de forma rápida e eficiente.

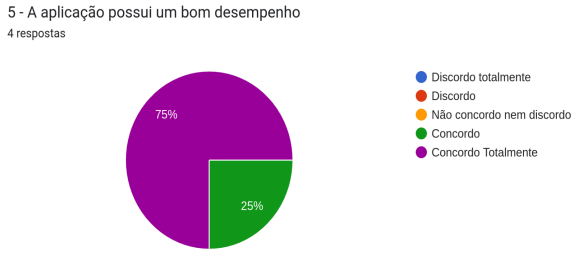

Figura 19: Resultado da pesquisa relacionado ao desempenho.

Satisfação geral: representa o nível de satisfação que a aplicação trouxe para os usuários que a utilizaram.

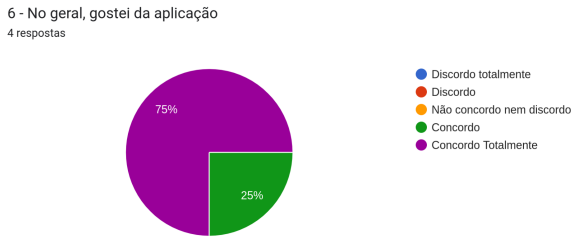

Figura 20: Resultado da pesquisa relacionado à satisfação geral.

Considerando os dados coletados na escala Likert conseguimos compreender de maneira geral a satisfação dos usuários ao aplicar o método através do sistema. Desta forma, podemos inferir que foi obtido um nível de satisfação positivo ao se utilizar o sistema.

6 EXPERIÊNCIA

6.1 Processo de desenvolvimento

No processo de desenvolvimento da aplicação foi adotado uma adaptação dos princípios do Kanban, um método de gestão de trabalho visual baseado em um quadro de atividades, onde cada cartão representa uma atividade que deve ser executada para que o objetivo final seja alcançado.

No contexto deste trabalho as funcionalidades do sistema foram transformadas em cartões em um quadro, permitindo que as mesmas fossem desenvolvidas de modo gradual de acordo com seu nível de prioridade. A cada duas semanas as atividades ainda existentes no quadro eram avaliadas para que uma nova ordem de priorização fosse estabelecida com intuito que o planejamento e o quadro estivesse alinhado com as necessidades e demandas.

Inicialmente, foi feita a prototipação das telas da aplicação através do Figma, uma ferramenta de design de interface. No qual, foi possível refinar algumas funcionalidades, além de facilitar o processo de desenvolvimento da interface.

O processo de desenvolvimento se iniciou no backend, com a implementação das regras de negócio e das funcionalidades definidas a partir do método, de tal forma que os serviços gerados permitissem o gerenciamento dos artefatos propostos.

Em seguida, foi iniciado a implementação do frontend, onde foi desenvolvida a interface do usuário responsável por consumir os serviços criados no backend.

6.2 Principais desafios

Alguns desafios foram enfrentados durante a construção do sistema, o inicial foi determinar quais funcionalidades deveriam ser desenvolvidas a partir dos processos e artefatos apresentados no método. Mas o desafio que mais se destacou foi a implementação do backend. Devido ao fato de ter mais afinidade com frontend, acabou sendo uma experiência única realizar todas as partes do processo desde a configuração inicial até deploy, proporcionando um crescimento pessoal.

7 TRABALHOS FUTUROS

A continuidade deste trabalho pode ser dividida em duas partes, uma que se trata do método e outra do sistema. No contexto deste trabalho o método Agile ETeasy foi aplicado por um pequeno time voluntário em poucas sessões e em poucos cenários, portanto, o método juntamente com o sistema poderiam ser aplicados uma quantidade maior de vezes e com equipes distintas, permitindo que o método fosse refinado e os seus resultados retificados. No contexto do sistema, o mesmo possui um escopo simples baseados nos artefatos gerados no

método, mas que representa uma base muito sólida e escalável para um ambiente com times de desenvolvimento ágil, sendo assim, em trabalhos futuros podem ser introduzidas novas funcionalidades, como:

- Dashboard, com a análise dos dados resultantes dos planejamentos e encerramentos de testes sendo utilizadas para apoiar a garantia de qualidade do software desenvolvido.
- Notificação, ao ser associado em alguns dos artefatos os usuários são notificados para simplificar o processo de gerenciamento do método por meio do sistema.

8 REFERÊNCIAS

Esta seção faz menção aos conteúdos estudados e citados durante o processo de construção deste trabalho, são eles:

[1] WHITTAKER, James A. Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design.

[2] COUTINHO, Jarbele. Método Agile ETeasy. Disponível em: https://sites.google.com/splab.ufcg.edu.br/agileeteasy/agile-eteasy.

[3] TestRail. Disponível em: https://www.gurock.com/testrail/.

[4] TestPad. Disponível em: https://ontestpad.com/.

[5] PractiTest. Disponível em: https://www.practitest.com/.