

CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

**DEFINIÇÃO DE UM ROTEIRO DE TESTE DE INSPEÇÃO COM
MÉTODOS DE VERIFICAÇÃO PARA A FÁBRICA DE TECNOLOGIAS
TURING (FTT)**

JACKSON HENRIQUE MOTA TEIXEIRA
KELLY KIANNY MORAIS CAIXÊTA

ANÁPOLIS
2020-01

**JACKSON HENRIQUE MOTA TEIXEIRA
KELLY KIANNY MORAIS CAIXÊTA**

**DEFINIÇÃO DE UM ROTEIRO DE TESTE DE INSPEÇÃO COM
MÉTODOS DE VERIFICAÇÃO PARA A FÁBRICA DE TECNOLOGIAS
TURING (FTT)**

Trabalho de Conclusão de Curso apresentado como requisito parcial para a conclusão da disciplina de Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Prof. Ma. Walquíria Fernandes Marins

**ANÁPOLIS
2020-01**

RESUMO

A inspeção atua como um importante filtro de detecção de defeitos em todas as fases do desenvolvimento de *software*, sendo que nas fases iniciais exerce maior efeito. As empresas de desenvolvimento têm se atentado cada vez mais para esse tipo de teste, entretanto, uma inspeção não controlada muitas vezes pode ser pior do que não realizar nenhuma inspeção. Assim, objetivou-se estabelecer um roteiro de teste de inspeção utilizando um modelo de processo, técnicas e ferramentas de auxílio importantes para identificar, tratar e corrigir erros, omissões e inconsistências quando ainda é fácil e barato resolvê-los. Para obter as informações necessárias para a avaliação, foi realizado entrevistas com a equipe de gerência e equipe de verificação e validação, observação do ambiente de trabalho e questionários sobre a aplicação do estudo. A partir dos dados obtidos foi realizada a aplicação do roteiro elaborado com tarefas e técnicas de inspeção na Fábrica de Tecnologias Turing (FTT), um ambiente acadêmico de desenvolvimento de *software* do Centro Universitário de Anápolis – UniEVANGÉLICA. Por fim, após identificar falhas no processo atual e realizar a intervenção com medidas corretivas que incluem atividades apropriadas para cada fase deste tipo de teste e ferramentas de apoio aos inspetores, foram apresentadas sugestões de melhoria do processo de inspeção dos documentos de requisitos de software.

Palavras-chave: Inspeção de *Software*. Fábrica de *Software*. Desenvolvimento de *Software*. Teste de *Software*. Requisitos de *Software*.

Abstract

Software inspection acts as an important failure detection filter in all stages of software development, and in the early stages it has the biggest effect. Development companies are increasingly focusing on this type of testing; however, an uncontrolled inspection can often be worse than not conducting any type of inspection. In this way, the aim was to establish an inspection test script using a process model, techniques and auxiliary tools which are important to identify, handle and correct errors, omissions, and inconsistencies when it is still easy and inexpensive to resolve them. To gather the necessary information for the evaluation, interviews were conducted with the management and verification and validation teams. It has also made observations about the workplace and it has been applied questionnaires about the application of this study. Based on the data obtained, the script developed with inspection tasks and techniques was applied at the Fábrica de Tecnologias Turing (FTT), an academic software development environment at the Centro Universitário de Anápolis - UniEVANGÉLICA. Finally, after identifying failures in the current process and carrying out the intervention with corrective measures which include proper activities for each phase of this type of testing, and also tools to support inspectors, suggestions were made to improve the inspection process of the software requirements documents.

Keywords: *Software inspection. Software factory. Software development. Software testing. Software requirements.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Metodologia do trabalho	3
Figura 2-Processo do Scrum.....	6
Figura 3 - Processo de Desenvolvimento da FTT	16
Figura 4 - Subprocesso da equipe de análise de requisitos da FTT	17
Figura 5 - Sub-processo da inspeção de requisitos	21
Figura 6 - Validação dos artefatos	21
Figura 7 - Função do entrevistado na empresa.....	25
Figura 8 - Período cursado pelo entrevistado	25
Figura 9 - Tempo que o entrevistado atua na área de TI	26
Figura 10 - Conhecimento do entrevistado sobre inspeção	26
Figura 11 - Experiência do entrevistado em inspeção.....	27
Figura 12 - Sub-processo de inspeção na FTT	28
Figura 13 - Necessidades da FTT em relação a inspeção.....	28
Figura 14 - Ferramenta de auxílio a leitura	29
Figura 15 - Eficiência das ferramentas de auxílio a inspeção	29
Figura 16 - Métricas de inspeção	30
Figura 17 - Contribuição do roteiro de inspeção na FTT	32
Figura 18 - Entendimento dos inspetores em relação ao roteiro.....	33
Figura 19 - Conhecimento adquirido a partir da pesquisa.....	33
Figura 20 - Utilização da técnica em outros ambientes.....	34
Figura 21- Relatório de inspeção pag.1	38
Figura 22- Relatório de inspeção pag.2	39
Figura 23- Relatório de inspeção pag.3	40
Figura 24-Relatório de inspeção pag.4	41
Figura 25-Relatório de inspeção pag.5	42
Figura 26- Relatório de inspeção pag.6	43
Figura 27- Relatório de inspeção pag.7	44
Figura 28- Relatório de inspeção pag.8	45
Figura 29- Relatório de inspeção pag.9	46
Figura 30- Relatório de inspeção pag.10	47
Figura 31- Relatório de inspeção pag.11	48
Figura 32- Relatório de inspeção pag.12	49
Figura 33- Relatório de inspeção pag.13	50

LISTA DE QUADROS

Quadro 1- Métricas de Revisão	12
Quadro 2 - Classificação de defeitos	14
Quadro 3 - Comparação dos modelos de inspeção na FTT	23
Quadro 4 - Defeitos detectados com técnica checklist.....	31
Quadro 5 - Defeitos detectados com técnica <i>AdHoc</i>	32

Sumário

1	INTRODUÇÃO.....	1
2	METODOLOGIA	3
3	FUNDAMENTAÇÃO TEÓRICA	5
3.1	Metodologia Ágil.....	5
3.2	Qualidade de <i>Software</i>	7
3.3	Verificação e Validação.....	7
3.4	Inspeção	9
3.4.1	Roteiro de Inspeção de Software	10
3.4.2	Participantes.....	11
3.4.3	Técnicas de Leitura de Software.....	11
3.4.4	Métricas	12
3.4.5	Classificação de Defeitos.....	14
3.5	Fábrica de Tecnologias Turing	15
4	ANÁLISE/DESENVOLVIMENTO	17
4.1	Necessidades na inspeção de documento de requisitos de <i>software</i>	18
4.2	Etapas de aplicação	18
4.3	Questionário de diagnóstico.....	20
4.4	Proposta de processo de inspeção dos artefatos de requisitos de <i>software</i>	20
4.5	Comparação dos modelos de inspeção	22
4.6	Questionário de avaliação	24
4.7	Entrevista	24
4.8	Resultados obtidos	24
5	Conclusão	34
5.1	Trabalhos futuros	35
6	REFERÊNCIAS BIBLIOGRÁFICAS	36
	APÊNDICE A – Documento de Relatório de Inspeção	38
	APÊNDICE B – Documentos de requisitos de software para inspeção.....	51
	APÊNDICE C – Checklist de inspeção.....	79
	APÊNDICE D – Gabarito de defeitos dos documentos de requisitos.....	80

1 INTRODUÇÃO

Segundo Sommerville (2011), em ambientes de desenvolvimento que adotam metodologias ágeis, espera-se que os membros do time possuam uma compreensão do sistema sem que seja necessário analisar a documentação, então, quando uma equipe muda, o conhecimento adquirido com a experiência durante o processo pode ser perdido, tornando difícil para os novos membros adquirir o mesmo entendimento do sistema, nessa situação a alta rotatividade das equipes pode tornar-se um problema relevante para o projeto. “A experiência do time é utilizada para avaliar as entradas necessárias para executar o trabalho planejado para o projeto. Esse julgamento e experiência são aplicados a todos os aspectos técnicos e de gerenciamento do projeto durante o processo de criar os entregáveis” (SCRUMSTUDY, 2016).

O objeto de estudo escolhido para esta pesquisa foi a Fábrica de Tecnologias Turing (FTT), um espaço dedicado aos acadêmicos dos cursos superiores de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA. Essa fábrica de *software* acadêmica fornece aos alunos a oportunidade de aplicar em projetos reais de desenvolvimento de *software*, o conhecimento adquirido em sala de aula, utilizando técnicas, ferramentas, processos e metodologias atuais, além de estimular em cada indivíduo o aperfeiçoamento das competências pessoais valorizadas pelas empresas (SOUSA et al., 2018).

Romanha, Muniz Jr. e Dale Luche (2019) em uma análise de fábricas de software em universidades brasileiras, identificaram que devido ao fato da mão de obra nesse ambiente ser em boa parte composta por discentes em formação e a curva de aprendizagem dos mesmos nas metodologias e ferramentas adotadas ser maior que a apresentada por profissionais experientes, a taxa de rotatividade é alta e os principais motivos disto é a conclusão de curso ou propostas de emprego no mercado, sendo assim, as fábricas de software acadêmicas adotam medidas para minimizar esse problema, como: constante realização de seleção, treinamento contínuo e inspeção do trabalho realizado pelos novos membros.

Na FTT a responsabilidade de avaliar a qualidade do que foi elaborado pelas equipes de análise de requisitos e equipe de desenvolvimento é da equipe de verificação e validação. Essa equipe atua desde o início do processo com testes de inspeção na documentação produzida, até as fases de desenvolvimento e entrega ao cliente com testes de validação do produto manuais e automatizados (SOUSA et al., 2018).

São grandes os efeitos causados por um erro de requisito descoberto nas fases de

implementação e operação do sistema, uma situação como essa exige a revisão de todos os artefatos como, código fonte, artefatos de teste, descrição de arquitetura e documentos de requisitos. Corrigir erros de requisitos nessas fases finais do processo implica custos significativos (OLIVEIRA, 2014).

Na Fábrica de Tecnologias Turing, foi observado que o teste de inspeção de requisitos é uma atividade dentro do processo macro, que ocorre após a finalização da especificação do requisito. Neste cenário o resultado pode ser uma grande demanda de retrabalho e falhas no cumprimento dos prazos de entrega, principalmente em sistemas complexos. Isto acontece porque, especificamente neste tipo de revisão, a equipe de verificação e validação não possui um fluxo de atividades definido, como: planejamento de teste de inspeção, relatórios para acompanhamento dos resultados das inspeções, gerenciamento e controle das inspeções realizadas. (SOUSA, 2018).

Neste contexto, definir um modelo padronizado de inspeção de requisitos de *software*, com técnicas que atendam às necessidades do ambiente de desenvolvimento e métricas que permitam o acompanhamento dos resultados, podem auxiliar a tomar medidas corretivas no processo e/ou artefatos e aumentar a qualidade do produto final. Em razão disso, a partir deste estudo será possível gerar um roteiro de teste de inspeção de requisitos de *software*, que apresenta técnicas e ferramentas que contribuem com a melhoria da qualidade do *software*, fundamentando em métodos de inspeção aplicáveis ao ambiente acadêmico da Fábrica de Tecnologias Turing.

Objetivo geral: Desenvolver um roteiro de testes de inspeção de requisitos de software aplicável a todos os projetos da Fábrica de Tecnologias Turing, apresentando técnicas e ferramentas de inspeção.

Objetivos específicos:

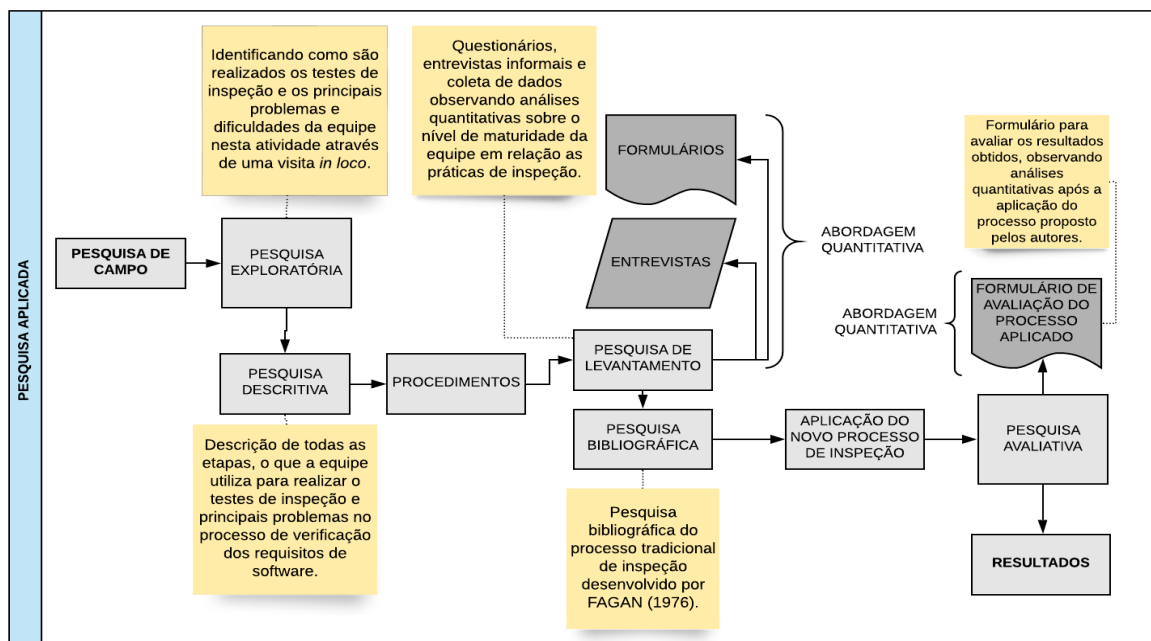
- Analisar o processo de teste de inspeção empregado na FTT e os principais problemas atuais;
- Apresentar um roteiro de teste de inspeção que se adapte ao contexto da FTT e reduza os problemas encontrados;
- Comparar a utilização do roteiro adaptado com a atividade de inspeção realizada até o momento.

Nos próximos capítulos serão abordados tópicos essenciais para fundamentar este trabalho: Na metodologia será descrito o processo utilizado na pesquisa e os procedimentos utilizados para a coleta e análise dos dados levantados, o tópico fundamentação teórica realiza uma revisão de artigos e livros relacionados ao assunto da pesquisa e é importante para orientar a interpretação dos dados obtidos, na seção de análise/desenvolvimento será apresentado os dados obtidos através do instrumento de pesquisa e a análise dos resultados, a última seção será a conclusão onde será apontado o desfecho do assunto levantado na pesquisa e a sugestão de trabalhos futuros.

2 METODOLOGIA

A finalidade deste trabalho é realizar uma pesquisa aplicada, isto é, avaliar como são realizadas as inspeções nos documentos de *software* da Fábrica de Tecnologias Turing através da pesquisa exploratória e descritiva. Propor uma melhoria no processo realizando pesquisas de levantamentos, pesquisas bibliográficas, aplicando um novo processo de teste de inspeção com métricas e métodos de verificação e identificando os resultados como mostrado no fluxograma da Figura 1.

Figura 1 - Metodologia do trabalho



Fonte: Autores, 2020

Para a realização deste trabalho foi feita uma pesquisa de campo do tipo exploratória e descritiva, com o intento de aprofundar o conhecimento sobre inspeção de requisitos de

software afim de facilitar a elaboração de um roteiro deste tipo de teste. Para a coleta de dados foi realizada a observação do ambiente de fábrica de *software* acadêmica, os envolvidos nos processos e as atividades desempenhadas, com o objetivo de definir o processo de teste de inspeção (CARNEVALLI; AUGUSTO; MIGUEL, 2001).

Durante o estudo, o teste de inspeção realizado na FTT foi assistido pelos pesquisadores. Este processo ocorreu através de visita informal, possibilitando uma análise desde a fase inicial quando a equipe de testes recebe os artefatos produzidos, até o momento em que finalizam as inspeções nos documentos.

Para identificar as dificuldades, foram realizadas pesquisas de levantamento, desenvolvendo questionários e entrevistas de análise quantitativa com os membros de todas as equipes (Equipes de Levantamentos de Requisitos de *Software*, Equipe de Testes e Equipe de Desenvolvimento), afim de identificar pontos fracos, minimizar os problemas encontrados no processo de verificação da qualidade dos requisitos de *softwares* desenvolvidos na FTT e aumentar a qualidade do produto.

Na definição do roteiro de teste, foi feito uma pesquisa do tipo bibliográfica, estudando o processo tradicional de inspeção de *software* desenvolvido por FAGAN (1976) o qual segundo (KALINOWSKI; SPÍNOLA, 2014) “possui uma estrutura rígida, com aspectos colaborativos, onde papéis, atividades e os relacionamentos entre atividades estão bem definidos”, sendo “um excelente meio de transferência de tecnologia e pode servir como mecanismo de *backup*, caso as pessoas chave sejam repentinamente removidas do projeto” (DOOLAN, 1992).

Para apoiar a fase de revisão individual foi realizada uma pesquisa das técnicas de leitura que melhor orientam os inspetores nas tarefas que desempenham. O roteiro de inspeção foi aplicado durante uma *sprint* em artefatos preparados para o experimento, como a equipe possui integrantes com pouca ou nenhuma experiencia em inspeção foram selecionadas técnicas de leitura como *Ad Hoc* e *Checklist*, com complexidade baixa para auxiliar os inspetores.

Para a aplicação desse novo processo de inspeção de *software*, foram realizadas intervenções com a equipe de testes da FTT. Artefatos foram produzidos pelos autores deste trabalho para a melhor análise dos resultados, esses documentos foram disponibilizados juntamente com a orientação dos pesquisados para a inspeção de acordo com o modelo de processo proposto. No entanto, após essa intervenção e aplicação do novo processo foram

realizadas pesquisas avaliativas com abordagem quantitativa para a obtenção dos resultados do processo aplicado.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados tópicos essenciais para fundamentar este trabalho, sendo eles: Fábrica de *Software*, Metodologia Ágil, Qualidade de *Software*, Verificação e Validação, Inspeção de Requisitos de *Software*, Técnicas de Leitura e Métricas.

3.1 Metodologia Ágil

Com a crescente necessidade de atender as mudanças repentinas, as novas oportunidades do mercado, o surgimento de produtos e serviços concorrentes e a grande insatisfação com processos de desenvolvimentos pesados, um grupo de desenvolvedores propuseram em 1990, novos métodos ágeis que focariam mais no *software* em si em vez de sua concepção e documentação (SOMMERVILLE, 2011).

Agile Manifesto (2001) descreve os doze princípios que regem o Manifesto Ágil: 1) A maior prioridade é satisfazer o cliente através da constante e antecipada entrega de *software* com valor agregado; 2) Mudanças nos requisitos são bem aceitas, mesmo tardiamente no desenvolvimento; 3) Entregar constantemente *software* funcionando, na menor escala de tempo possível; 4) Administradores e desenvolvedores devem trabalhar juntos durante todo o projeto; 5) Os indivíduos devem ser motivados com ambiente e suporte necessário; 6) Comunicação face a face é o melhor método para transmitir informações; 7) *Software* funcionando é o indicador primário do progresso; 8) Os processos ágeis promovem um ritmo constante do desenvolvimento; 9) A excelência técnica contínua e o *design* satisfatório eleva a agilidade; 10) Simplicidade para potencializar a quantidade de trabalho desnecessário; 11) A auto organização das equipes aperfeiçoa arquiteturas, requisitos e *designs*; 12) Em intervalos regulares, as equipes fazem uma análise do que deve ser melhorado e então ajustam seu comportamento. “Como todos os outros processos profissionais de desenvolvimento de *software*, o desenvolvimento ágil tem de ser gerenciado de modo que se faça o melhor uso com o tempo e os recursos disponíveis para a equipe” (SOMMERVILLE, 2011). Entre as metodologias ágeis existentes, o *Scrum* é a mais conhecida e usada, apesar de ter sido criado antes do manifesto ágil, ele se enquadra nos princípios ágeis.

3.1.1 Scrum

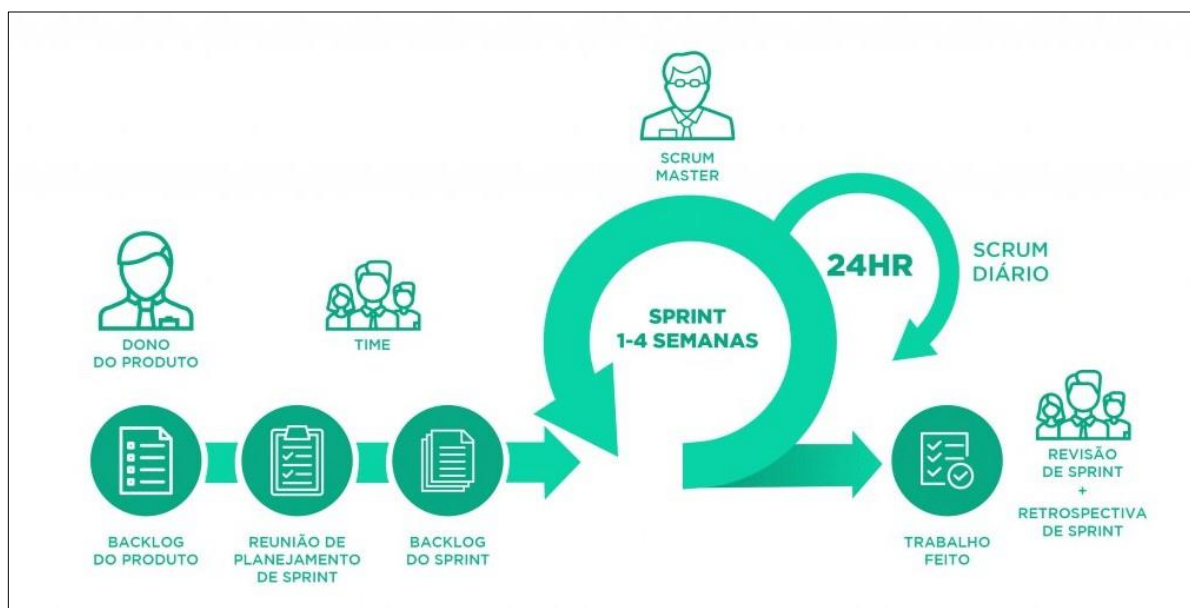
O *Scrum* é “um *framework* dentro do qual pessoas podem tratar e resolver problemas

complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível” (SCHWABER; SUTHERLAND, 2013). Segundo Schwaber e Sutherland (2013) o *framework scrum* estabelece quatro eventos formais dentro dos limites da *Sprint*, são eles: a) Reunião de planejamento da *sprint*; b) Reunião diária; c) Reunião de revisão da *sprint*; d) Retrospectiva da *sprint*.

O Time *Scrum* é formado por: a) *Product Owner* ou dono do produto, é o profissional responsável por gerenciar o *backlog* do produto e agregar valor ao trabalho; b) Time de Desenvolvimento, profissionais com diversas habilidades para entregar versões do produto ao final de cada *sprint*; c) *Scrum Master*, profissional responsável por garantir a aplicação do *scrum* de forma correta. O time é auto organizável e multifuncional, e por isso escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do time (SCRUMSTUDY™, 2016).

No *scrum* o desenvolvimento do projeto é dividido por *sprints*, um *time-boxed* de um mês ou menos, que se inicia com a reunião de planejamento, nesta reunião é definido o que pode ser entregue como resultado, após esta, para inspecionar o trabalho realizado e sincronizar as atividades do time é realizada a reunião diária, ao final de cada *sprint* o time apresenta ao *product owner* o produto “pronto” na reunião de revisão e para finalizar realiza uma análise pessoal e do time na retrospectiva da *sprint* (SCRUMSTUDY™, 2016), esse ciclo é mostrado na Figura 2.

Figura 2-Processo do Scrum



Fonte: Adaptado de Barbosa, 2019.

Usado desde o início de 1990, o *scrum* é fundamentado nas teorias empíricas de controle de processo e é baseado em três pilares que apoiam o empirismo: transparência, inspeção e adaptação (SCHWABER; SUTHERLAND, 2013). Transparência: A transparência assegura que todo o processo *Scrum* esteja visível para qualquer pessoa do time. Inspeção: Os artefatos *scrum* devem ser frequentemente inspecionados, afim de detectar variações no processo. Adaptação: Se um ou mais aspectos de um processo se desvia para fora dos limites aceitáveis e o produto se torna inaceitável, o processo ou o produto deve ser ajustado (SCHWABER; SUTHERLAND, 2013). Estes três pilares responsabilizam todo o time em relação a qualidade do produto, pois proporciona a ciência de todo o processo realizado.

3.2 Qualidade de Software

Visto que a qualidade de *software* não possui um valor fixo, ela pode ser mensurada relacionando a conformidade dos artefatos com as necessidades do cliente (ZANIN et al., 2018). (PRESSMAN; MAXIM, 2016) define qualidade de *software* como: “uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam”.

O controle da qualidade em um ambiente de desenvolvimento de *software* pode contribuir em diversos aspectos. Zanin et al. (2018) lista uma série de benefícios que a qualidade de *software* pode oferecer para usuários e empresas, entre elas se destacam: a) Economia de dinheiro ao detectar defeitos nas primeiras fases do projeto; b) Evita emergências corporativas; c) Eleva a confiança do cliente; d) Aumenta o nível da experiência do usuário, e) Proporciona mais lucro, f) Aumenta a satisfação do cliente, g) Promove organização, produtividade e eficiência.

Pela visão da engenharia de software, a qualidade é um tema amplo e possui abordagens para cada fase do ciclo de vida do desenvolvimento. O processo de qualidade de *software* deve estar incorporado ao processo de desenvolvimento de *software*, de forma que para cada fase do desenvolvimento seja realizada uma etapa do processo de qualidade, estratégia conhecida como verificação e validação (ZANIN et al., 2018).

3.3 Verificação e Validação

Para Pressman e Maxim (2016) o teste de *software* é um tema amplo, conhecido como verificação e validação (V&V). “Verificação refere-se ao conjunto de tarefas que garantem que o *software* implementa corretamente uma função específica. Validação refere-se a um conjunto

de tarefas que asseguram que o *software* foi criado e pode ser rastreado segundo os requisitos do cliente” (PRESSMAN; MAXIM, 2016). “A motivação que está por trás do teste de programas é a confirmação da qualidade do *software* com métodos que podem ser econômica e efetivamente aplicados a todos os sistemas, de grande e pequena escala” (MILLER, 1977).

Uma ampla gama de atividades compõem a verificação e validação, tais como: revisões técnicas, auditorias de qualidade e configuração, monitoramento de desempenho, simulação, estudo de viabilidade, revisão de documentação, revisão de base de dados, análise de algoritmo, teste de desenvolvimento, teste de usabilidade, teste de qualificação, teste de aceitação e teste de instalação (PRESSMAN; MAXIM, 2016).

Kalinowski e Spínola (2014) destacam as seguintes atividades que podem ser utilizadas para verificar a qualidade de *software*:

- Revisões de *software*;
- Testes;
- Padrões e procedimentos formais;
- Controle de mudanças;
- Métricas de *software*;
- Procedimentos para coleção e disseminação de informações, (KALINOWSKI; SPÍNOLA, 2014).

Segundo Pressman e Maxim (2016), a revisão técnica é importante para avaliar a qualidade do *software* antes da implementação, quando ainda é fácil e barato corrigir erros, omissões e inconsistências. “Uma revisão técnica (RT) é o filtro mais eficiente do ponto de vista de controle da qualidade” (PRESSMAN; MAXIM, 2016). A “Inspeção de *software* é um tipo particular de revisão que pode ser aplicado a todos os artefatos de *software* e possui um processo de detecção de defeitos rigoroso e bem definido” (KALINOWSKI; SPÍNOLA, 2014). Entre os benefícios de se utilizar a inspeção de software pode ser citado, a detecção de defeitos nas fases iniciais do processo de desenvolvimento, isso porque, a redução no esforço e custo para corrigir nesta fase é significativo, levando em consideração que o produto ainda não foi implementado.

A expressão defeito é comumente utilizada de forma generalizada. Em testes de software é seguida a terminologia padrão para Engenharia de Software do IEEE (IEEE 610.12, 1990), onde: erro é aquilo que foi cometido por ação humana, gerando um código defeituoso.

Defeito se refere a uma funcionalidade implementada de forma incorreta. Falha acontece quando um código defeituoso é executado e interfere no funcionamento do sistema. No entanto, no contexto deste trabalho, os termos defeito e discrepância, estarão relacionados às faltas nos artefatos que estão sendo revisados, podendo caracterizar um defeito de fato ou um falso positivo como é chamado defeitos que após análise são classificados como interpretação incorreta (ZANIN et al., 2018).

3.4 Inspeção

Pressman e Maxim (2016) afirma que revisões técnicas são como filtros aplicados durante todo o desenvolvimento de *software* para apontar erros e defeitos, contribuindo para a qualidade do produto. “Uma revisão é uma forma de usar a diversidade de um grupo de pessoas para apontar aperfeiçoamentos necessários no produto de uma única pessoa ou de uma equipe” (PRESSMAN; MAXIM, 2016). Zanin et al. (2018) relatam que a inspeção é um tipo de revisão técnica formal, onde os artefatos analisados são distribuídos para equipes distintas em reuniões de revisão, mencionando os erros encontrados para o autor do artefato.

Pressman e Maxim (2016) ressaltam que durante a reunião, o produto deve ser revisado e não o seu autor para não causar desconforto entre as equipes, a reunião não deve se desviar do seu objetivo e prazo estabelecidos, os debates e discussões devem ser registrados e deixados para depois da reunião para que não haja perda de tempo, os problemas devem ser expostos e não as soluções, o número de participantes deve ser limitado e preparado com antecedência.

Em 1976 foi desenvolvido por Fagan o processo tradicional de inspeção de *software*, a partir deste processo, é possível encontrar na literatura diversas variações que apoiam a inspeção, como: Parnas e Weiss (1987), Humphrey (1989), Bisant e Lyle (1989), Johnson (1994) e Sauer et al. (2000) entre outros. Alguns autores defendem que no processo não é necessário realizar a etapa de reunião, pois o número de defeitos encontrados nela é relativamente baixo, outros tratam questões como equipes assíncronas, onde a dispersão geográfica reduz a comunicação, outros autores decidem por incluir outras etapas que em alguns casos servem para tratar outras questões como sugestões de melhoria.

Apesar de ser um modelo tradicional o processo produzido por (FAGAN, 1976) adere aos pilares dos métodos ágeis, o que torna possível seu emprego na FTT. O processo melhora a colaboração, pois todos testem os artefatos e não somente o especialista de teste, a comunicação e transparência do time aumentam através da etapa de reunião, a documentação

se torna mais enxuta pois são descartados defeitos inconsistentes.

3.4.1 Roteiro de Inspeção de Software

Desenvolvido por (FAGAN, 1976) o roteiro ou processo tradicional de inspeção de *software* é composto por cinco atividades:

I. Planejamento - O moderador define o contexto da inspeção, seu objetivo, técnica de detecção de defeitos utilizada, documentos a serem inspecionados, participantes da reunião e a distribuição do material necessário (FAGAN, 1976).

II. Preparação - Individualmente os inspetores realizam uma operação de detecção de erros, examinam os artefatos usando uma técnica de leitura para identificar os possíveis defeitos e elaborar uma lista de discrepâncias que será usada na reunião (FAGAN, 1976). “Um erro é definido como qualquer condição que causa mau funcionamento ou que impede a obtenção de resultados esperados ou especificados anteriormente. Portanto, os desvios das especificações são claramente denominados erros” (FAGAN, 1976).

III. Reunião - Na reunião todos os que foram escolhidos pelo moderador devem participar, as discrepâncias encontradas pelos inspetores na fase anterior devem ser discutidas neste momento, fica a cargo do moderador classificar o tipo do defeito e sua severidade, caso haja necessidade, outra sessão pode ser iniciada após um período de atividade diferente, pois somente duas sessões de duas horas por dia são aceitáveis (FAGAN, 1976).

IV. Retrabalho - Nesta fase os autores dos artefatos devem corrigir os defeitos encontrados pelos inspetores e validados pelo moderador (FAGAN, 1976).

V. Continuação - Os artefatos corrigidos são avaliados pelo moderador, ele decide se é necessário ocorrer uma nova inspeção ou não (FAGAN, 1976).

Filho (2009) apresenta como a reunião de inspeção é classificada ao seu término:

I. Revisão completa - A reunião foi completamente realizada e seu objetivo foi cumprido;

II. Revisão incompleta - A revisão não terminou por algum motivo;

III. Revisão cancelada - A revisão não foi iniciada por algum motivo.

No mesmo texto Filho (2009) demonstra como deve ser feita a classificação dos artefatos ao final da reunião:

I. Aprovado sem modificações - quando não são encontrados defeitos;

II. Aprovado com modificações - quando são encontrados pequenos defeitos que devem ser corrigidos;

III. Aprovado com reinspeção - quando são encontrados defeitos que exigem uma

nova inspeção;

Reprovado - quando são encontrados defeitos críticos e o artefato tem que ser refeito.

3.4.2 Participantes

Segundo Fagan (1976) a equipe de inspeção desempenha um melhor trabalho quando seus membros exercem funções específicas, são elas:

I. Moderador - O moderador deve ser a pessoa chave no processo de inspeção pois é ele quem conduzirá a reunião, ele precisa possuir habilidades técnicas que possibilitem seu entendimento das discrepâncias encontradas e de como elas podem afetar o sistema. Durante a reunião ele conduz as atividades promovendo a harmonia entre o grupo e usando os pontos fortes de cada participante. Também é dever do moderador, agendar o local e horário para a reunião, relatar os resultados da inspeção e acompanhar o retrabalho e oferecer treinamento a equipe que fará parte da inspeção (FAGAN, 1976).

II. Autor - Analista responsável por produzir o documento. Também é o responsável por corrigir os defeitos apontados na reunião (FAGAN, 1976).

III. Inspetor - Responsável por encontrar os erros, omissões e inconsistências dos artefatos (FAGAN, 1976).

IV. Relator - Alguns autores como Sommerville (2011) defendem ainda que a reunião deve ter um relator, responsável por registrar o andamento e os resultados da reunião.

3.4.3 Técnicas de Leitura de Software

Em um estudo comparativo das técnicas de inspeção de *software* Bertini (2006) afirma que a utilização de técnicas de leitura eficientes na etapa de inspeção individual, é o que torna a inspeção eficiente. As técnicas de leitura são classificadas como não-sistemáticas quando não há instruções ao inspetor ou sistemáticas quando há instruções do que deve ser procurado e como encontrar. As técnicas de leitura mais comuns são descritas por Bertini (2006) da seguinte forma:

I. Técnica de Leitura *Ad Hoc* - A técnica mais usada pelas empresas de desenvolvimento de *software* é considerada como não-sistemática, os inspetores não recebem nenhum tipo de orientação e dependem altamente da sua experiência e conhecimento, como não há nenhuma instrução os inspetores devem analisar todo o aspecto do artefato, um grande defeito desta técnica é o fato de não ser possível realizar melhorias pois não há passos a serem seguidos (BERTINI, 2006).

II. Técnica de Leitura baseada em *Checklist* - Esta técnica apresenta uma boa melhoria se comparada a técnica *Ad Hoc*. Cada inspetor recebe uma lista de questões que orientam a encontrar os erros mais comuns nos artefatos que estão sendo analisados. O *checklist* é personalizável, podendo ser modificado conforme a maior ocorrência de defeitos (BERTINI, 2006). A fragilidade desta técnica é que, ao utilizá-la em várias inspeções, os testadores tendem a conferir somente os pontos de defeitos frequentes e os pontos não considerados pelo *checklist* tendem a ser deixados de lado pelo inspetor (BERTINI, 2006).

3.4.4 Métricas

Segundo Pressman e Maxim (2016) uma série de métricas podem ser usadas como indicativos importantes no processo de revisão.

“Uma métrica de *software* relaciona as medidas individuais de alguma maneira (por exemplo, o número médio de erros encontrados por revisão ou o número médio de erros encontrados por teste de unidade)” (PRESSMAN; MAXIM, 2016).

Uma métrica de *software* é uma característica de um sistema de *software*, documentação de sistema ou processo de desenvolvimento que pode ser objetivamente medido. Exemplos de métricas incluem: o tamanho de um produto em linhas de código; o índice Fog, que é uma medida da legibilidade de uma passagem de texto escrito; o número de defeitos relatados em um produto de *software* entregue, e o número de pessoas/dia requerido para desenvolver um componente de sistema (SOMMERVILLE, 2011).

(PRESSMAN; MAXIM, 2016) afirmam que “um elemento-chave de qualquer processo de engenharia é a medição. Você pode usar medidas para compreender melhor os atributos dos modelos criados e para avaliar a qualidade dos produtos ou sistemas construídos”. Algumas métricas que podem ser usadas na inspeção de documento de requisitos de *software* estão apresentadas no Quadro 1.

Quadro 1- Métricas de Revisão

MÉTRICAS	DEFINIÇÃO
Esforço de preparação, E_p	Esforço (em pessoas/hora) exigido para revisar um artefato antes da reunião de revisão.

Esforço de avaliação, E_a	Esforço (em pessoas/hora) que é despendido durante a revisão em si.
Esforço de reformulação, E_r	Esforço (em pessoas/hora) dedicado à correção dos erros revelados durante a revisão.
Tamanho do artefato de <i>software</i> , TAS	Medida do tamanho do artefato de <i>software</i> que foi revisto (por exemplo, o número de modelos UML ou o número de páginas de documento).
Erros secundários encontrados, Err_{sec}	Número de erros encontrados que podem ser classificados como secundários (exigindo menos para ser corrigidos do que algum esforço pré-especificado).
Erros graves encontrados, Err_{graves}	Número de erros encontrados que podem ser classificados como graves (exigindo mais para ser corrigidos do que algum esforço pré-especificado).
Esforço total de revisão, $E_{revisão}$	$E_{revisão} = E_p + E_a + E_r$
Valor total de erros encontrados, Err_{tot}	$Err_{tot} = Err_{sec} + Err_{graves}$
Densidade de Erros = Err_{tot} / TAS	Densidade de erros encontrados em um documento ou diagrama (auxilia na estimativa da possibilidade de erros em outros projetos).

Fonte: Adaptado de (PRESSMAN; MAXIM, 2016)

Sommerville (2011) ressalta a importância das medições de processo para avaliar a evolução da eficiência de um processo. As melhorias no processo de teste reduzem o esforço e/ou tempo de teste. Entretanto, não podem ser usadas para determinar se a qualidade de produto melhorou. Enquanto as métricas do produto são atributos do *software* em si, tal como tamanho ou complexidade.

3.4.5 Classificação de Defeitos

Bertini (2006) apresenta no Quadro 2 o detalhamento da classificação de defeitos obtida a partir de Kirner e Abib (1997) e abragem os tipos:

- Omissão – Funcionalidade omitida, Performance omitida Ambiente omitido e Interface omitida;
- Comissão – Informação Ambígua, Informação inconsistente, Funcionalidade incorreta e Seção Incorreta.

Quadro 2 - Classificação de defeitos

CLASSE	TIPO	DESCRIÇÃO
Omissão	Funcionalidade Omitida (FO)	Alguma informação relativa à descrição do comportamento esperado do sistema, não aparece no documento.
	Performance Omitida (PO)	Alguma informação, relativa à descrição da performance desejada, não aparece no documento, ou aparece de forma inaceitável.
	Ambiente Omitido (AO)	Alguma informação, relativa à descrição do hardware, do software, do banco de dados e do pessoal envolvido, não aparece no documento.
	Interface Omitida (IO)	Alguma informação, relativa à forma como o sistema interagirá ou se comunicará com componentes que estão fora do escopo do sistema, não aparece no documento.
Comissão	Informação Ambígua (IA)	Um termo importante, uma frase ou uma sentença, essenciais para o entendimento do sistema não foi definido no documento, ou foi definido de forma que possa causar confusão.

	Informação Inconsistente (II)	Duas sentenças contradizem-se mutuamente ou expressam ações de que não estão corretas ou não podem ser executadas.
	Funcionalidade Incorreta (FI)	Alguma sentença expressa um fato que não pode ser verdade de acordo com as condições especificadas.
	Seção Incorreta (SI)	Alguma informação está em um local errado dentro do documento.
Outros (O)		Defeitos que não se enquadram nos tipos acima.

Fonte: Adaptado de (BERTINI, 2006).

3.5 Fábrica de Tecnologias Turing

Buscando fornecer a acadêmicos do ensino superior, um ambiente que simule empresas de desenvolvimento de software e seus principais desafios, instituições de ensino tem optado por oferecer instalações educacionais que apesar de ser localizadas em sala de aula, são mobiliadas tal com empresas modernas de desenvolvimento. O objetivo principal é criar fábricas de *software* que estimulem alunos a vivenciar o processo de concepção de sistemas reais. A expressão *software factory* ou fábrica de *software* tem como objetivo aplicar o conceito de indústria no desenvolvimento de *software*, provendo uma linha de produção de soluções que atendam às necessidades de cada cliente, esse processo aumenta a produtividade e diminui prazos e custos, tornando-o mais independente do fator humano (XAVIER, 2008).

“O termo fábrica sinaliza um compromisso de longo prazo, esforços integrados - acima do nível de projetos individuais - para melhorar operações de *software*” (AAEN; BOTCHER; MATHIASSEN, 1997).

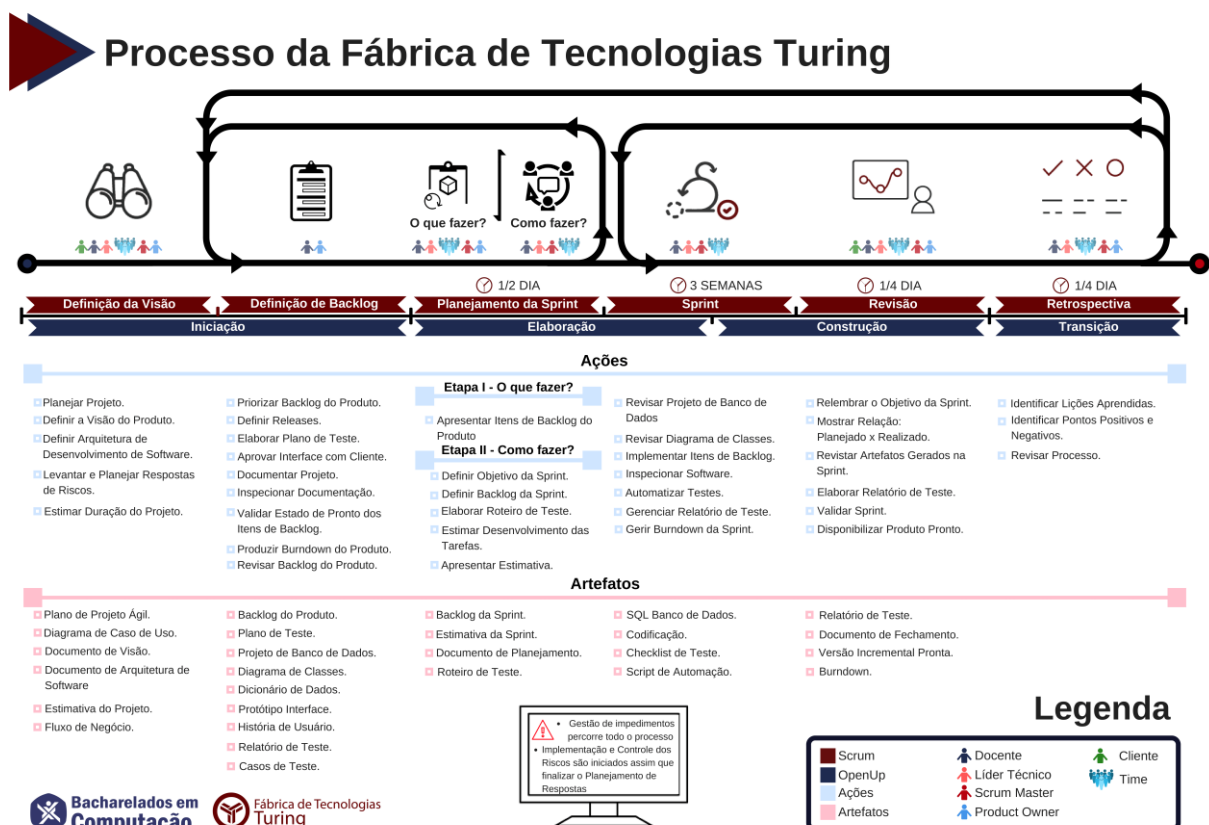
No meio acadêmico, as fábricas de *software* visam atender às demandas internas e externas por *softwares* personalizados e têm como objetivo promover a interdisciplinaridade nos cursos de TI, além de complementar a formação dos alunos através da participação destes em projetos reais desenvolvidos em laboratórios que são frequentados também por pesquisadores e docentes (ROMANHA, 2015).

A Fábrica de Tecnologias Turing (FTT) foi fundada em 2006 e se tornou um grande diferencial na formação dos acadêmicos dos cursos Bacharelados de Computação da UniEvangélica, proporcionando uma visão de mercado de trabalho, experiência com tecnologias atuais e orientação especializada em várias áreas de conhecimento

(UNIEVANGÉLICA, 2017). Os alunos que ingressam na FTT através de processo seletivo adquirem além de competências técnicas, competências comportamentais, através do trabalho em equipe e da hierarquia que a compõe (FARIA et al., 2018).

Faria et al. (2018) demonstra como a Fábrica de Tecnologias Turing está em conformidade com as tecnologias mais atuais, o processo utiliza uma metodologia ágil híbrida composta pelo *Scrum*, que tem foco no processo gerencial e o *OpenUp*, que tem foco no processo produtivo tornando o desenvolvimento de *software* iterativo, incremental e empírico, conforme mostrado na Figura 3.

Figura 3 - Processo de Desenvolvimento da FTT



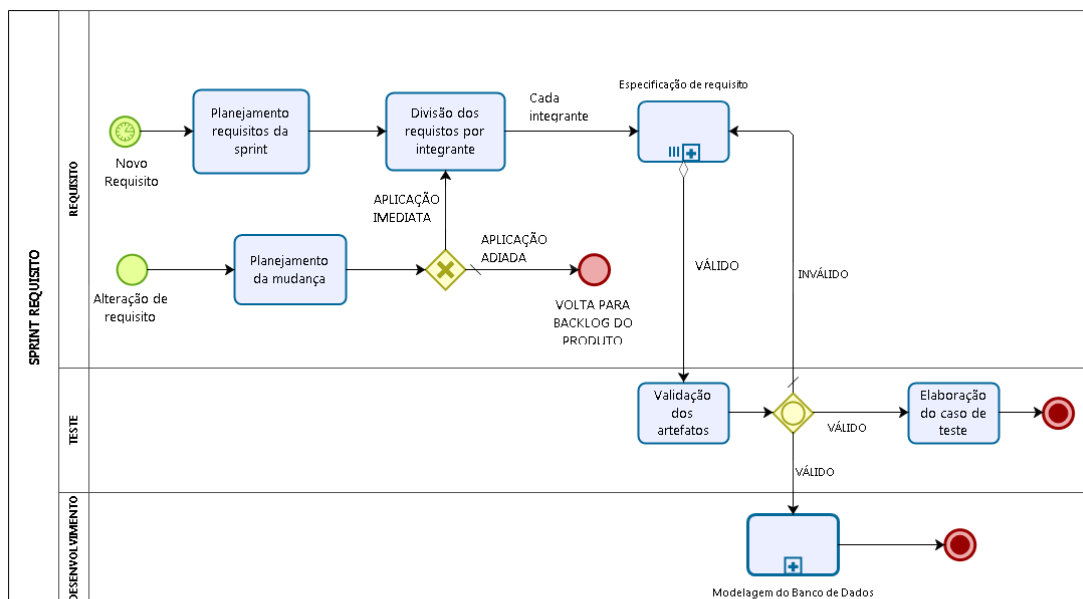
Fonte: Repositório FTT, 2018.

Na Figura 3 é possível notar que o processo da FTT está estruturado de forma que contempla todas as etapas de desenvolvimento: a) Definição da Visão – Nesta etapa é realizado todo o estudo e planejamento de novos projetos, é realizada a definição da arquitetura, estimativa de esforço e tempo e o fluxo de negócio; b) Definição de Backlog - Etapa em que os acadêmicos começam a ser mais atuantes, é realizado a priorização das funcionalidades necessárias ao funcionamento do sistema; c) Planejamento - Essa etapa se inicia após a

Definição do Backlog e se repete antes do início de cada Sprint; d) Sprint - Nesta etapa ocorre a especificação o desenvolvimento e a inspeção do produto; e) Revisão - Na etapa de revisão os incrementos produzidos na fase anterior são validados e seus resultados analisados; f) Retrospectiva - É a fase em que o processo é avaliado sob a ótica do time, o que foi bom e o que precisa ser melhorado (FARIA et al., 2018).

Cada equipe que compõe o time da FTT possui um subprocesso em que apresenta as etapas que devem ser realizadas até a finalização de um entregável. A inspeção de requisitos de *software* está inserida ao final do subprocesso de análise de requisitos, como uma atividade simples onde não constam informações como: quais técnicas devem ser usadas, quais ferramentas serão necessárias, quais passos ou etapas devem ser seguidos ou quais pessoas deverão participar desta tarefa, como mostra a Figura 4.

Figura 4 - Subprocesso da equipe de análise de requisitos da FTT



Fonte: Autores, 2020

4 ANÁLISE/DESENVOLVIMENTO

Conforme os objetivos propostos e método de pesquisa planejado para o desenvolvimento da pesquisa foi realizada a análise do ambiente, Fábrica de Tecnologias Turing, a rotina de trabalho dos colaboradores e os artefatos produzidos na inspeção dos requisitos de *software*.

Foram realizadas entrevistas informais, questionários, *workshop* e coleta de dados conforme disponibilidade da equipe, sendo assim, participaram da pesquisa de forma que o experimento não interferisse no projeto em desenvolvimento de forma negativa, desta forma, os artefatos utilizados na pesquisa foram adaptações feitas em artefatos de projetos em andamento, somente após a aplicação do roteiro de inspeção e a obtenção de seus resultados o subprocesso de inspeção foi formalizado entre a equipe de verificação e validação.

4.1 Necessidades na inspeção de documento de requisitos de *software*

Em visita exploratória na Fábrica de Tecnologias Turing, percebeu-se que apesar do processo de cada equipe da FTT sofrer alterações constantes se adequando ao contexto do ambiente e as necessidades de seus colaboradores, ainda apresentava impasses em demonstrar resultados satisfatórios na inspeção de requisitos de *software*, tais como: dificuldade em acompanhar a evolução dos artefatos e da equipe, contribuir com a transferência de conhecimento entre os membros, identificar os principais tipos de erros e gerar relatórios que contribuem nas estimativas de projetos futuros. Outra dificuldade encontrada está relacionada ao material utilizado para auxiliar nas inspeções, por se tratar de um *checklist* extenso e com muitos detalhes se tornou cansativo e obsoleto na percepção dos integrantes com mais experiência, em uma equipe nova e com pouco conhecimento, isto pode impossibilitar a detecção dos erros nos artefatos.

Buscando atenuar os efeitos da alta rotatividade, pouca experiência dos acadêmicos que ingressam na FTT e a falta de um processo formalizado de inspeção, foi realizada a aplicação e avaliação de um modelo de processo de inspeção executado em quatro etapas, sendo: 1) Identificação de pontos vulneráveis na inspeção de artefatos de *software*; 2) Adaptação de um processo de inspeção ao contexto da FTT; 3) Aplicação do modelo de inspeção proposto; 4) Avaliação dos resultados da aplicação do modelo de inspeção.

4.2 Etapas de aplicação

A primeira etapa foi identificar os pontos vulneráveis da inspeção de requisitos de *software* realizadas na FTT. Nesta fase, foi desenvolvida uma pesquisa, através do questionário de diagnóstico, com os integrantes da equipe de testes, *scrum master* e *product owner*, afim de identificar possíveis dificuldades em relação ao conhecimento do processo de inspeção e técnicas utilizadas na FTT. Como resultado foi possível notar que por se tratar de uma equipe nova e em fase de adaptação, alguns não conseguiram opinar de forma satisfatória em relação a esse tipo de teste, pois ainda não haviam executado o trabalho em sua nova função. Neste

caso, foi necessário acrescentar um novo passo, um treinamento em formato de *workshop*, abordando os conceitos de processo de inspeção, sua necessidade, como realiza-lo e as ferramentas que o auxiliam para nivelar o conhecimento necessário. Após o treinamento foi realizada novamente a pesquisa com o questionário de diagnóstico, que identifica o conhecimento dos colaboradores em relação ao formato atual de inspeção, nesta última, pôde-se obter resultados mais concretos e efetivos.

A segunda etapa consistiu em adaptar a atividade de inspeção realizada nos documentos de requisitos ao modelo de processo de inspeção proposto para a equipe de verificação e validação. Foi selecionado para esta fase o processo tradicional de inspeção desenvolvido por FAGAN (1976), o qual segundo (KALINOWSKI; SPÍNOLA, 2014) “possui uma estrutura rígida, com aspectos colaborativos, onde papéis, atividades e os relacionamentos entre atividades estão bem definidos”, sendo “um excelente meio de transferência de tecnologia e pode servir como mecanismo de *backup*, caso as pessoas-chave sejam repentinamente removidas do projeto” (DOOLAN, 1992). No segundo passo desta etapa foram selecionadas as técnicas de leitura que auxiliam os inspetores na detecção dos defeitos, as técnicas escolhidas foram a *Leitura Ad Hoc* e *Leitura Baseada em Checklist (LBCh)*, a escolha destas duas técnicas foi definida considerando-se a rotina de trabalho da equipe, a experiência dos integrantes e o nível de complexidade das técnicas. O material de apoio às inspeções foi produzido posteriormente de acordo com essas definições.

Na terceira etapa foi realizada a aplicação do modelo de inspeção, que se iniciou com a preparação dos artefatos a serem inspecionados. Os documentos de requisitos de software selecionados para sofrer a inspeção foram estudados e obtiveram defeitos adicionados ao seu escopo, para fins de análise da quantidade de defeitos em que cada inspetor seria capaz de identificar e qual técnica de leitura fosse mais eficaz individualmente. Durante a pesquisa os autores deste trabalho ocuparam o papel de moderadores do processo de inspeção para realizar a análise dos resultados, em seguida para dar início à fase de inspeção individual dos artefatos, que seria realizada em sua tradicional forma presencial, porém face ao cenário atual da pandemia de COVID-19, foi realizada uma reunião em videoconferência com os participantes do experimento para dar início à intervenção. Nesta reunião os inspetores receberam para realizar as inspeções individuais, os artefatos a serem inspecionados e as orientações referentes ao uso das técnicas de leitura, a classificação dos defeitos e o preenchimento do formulário de inspeção do documento de requisitos.

Após a reunião em videoconferência com as instruções, cada inspetor realizou a inspeção nos documentos que lhe foram designados. Ao ser finalizada as inspeções individuais, os moderadores realizaram uma auditoria para analisar os defeitos identificados, classificando-os em: defeitos válidos os quais foram inseridos nos documentos, falso positivo que são erros de interpretação do inspetor, defeitos extras que dizem respeito a erros que já estavam no documento e não foram inseridos para o experimento, defeito inconsistente onde o inspetor não deixou claro o defeito e defeitos não encontrados que são aqueles inseridos no documento, porém não identificados pelo inspetores. Como os documentos de requisitos de software foram utilizados exclusivamente no experimento e não seriam incluídos nos projetos em andamento da FTT, não foram realizadas as atividades finais do processo de correção e acompanhamento.

A etapa quatro, foi a última, nela foi realizada a análise dos resultados obtidos na aplicação do processo proposto para a FTT. Os resultados foram apresentados aos participantes da pesquisa e a orientadora da equipe de testes da FTT, para que pudesse avaliar a melhoria em relação ao formato antigo de inspeção. A partir das observações feitas pelos participantes o roteiro de teste de inspeção foi formalizado para a equipe.

4.3 Questionário de diagnóstico

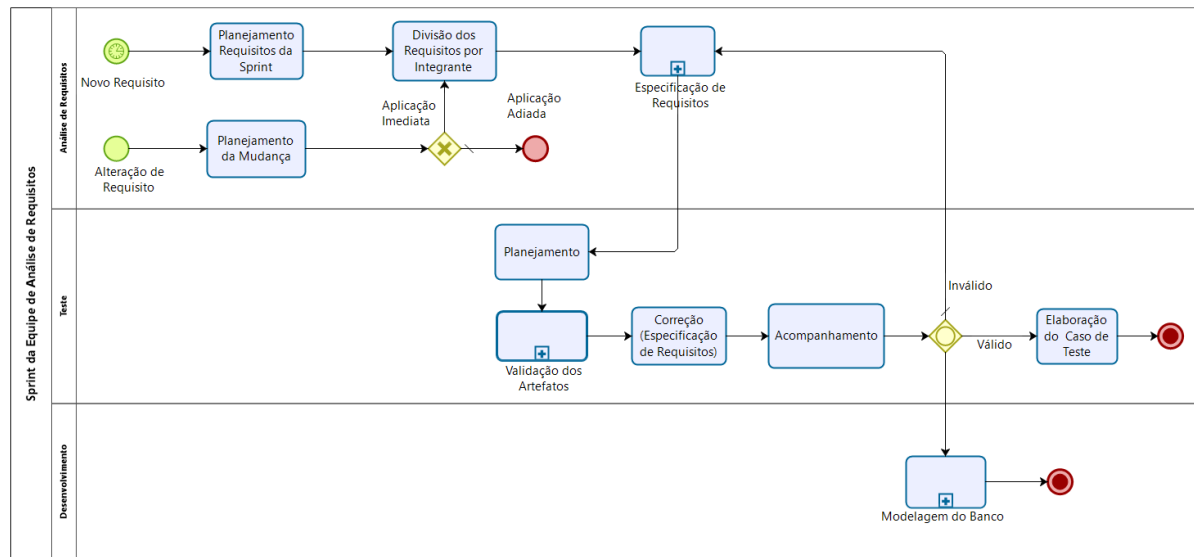
No início da pesquisa, foi aplicado um questionário com o objetivo de avaliar aspectos gerais do estudo. Para responder o questionário participaram o *scrum* master, o *product owner*, e os integrantes da equipe de testes da FTT, cada um possuindo um nível de conhecimento e experiência diferente, o que contribui para a análise do processo em um ambiente com alta rotatividade como a Fábrica de Tecnologias Turing.

O questionário de aspectos gerais do estudo revelou informações que serão analisadas na seção de resultados obtidos, nele foi avaliado o nível de maturidade da equipe em relação as práticas de inspeção usadas anteriormente à intervenção, técnicas utilizadas, métricas adotadas e quais os resultados que elas produziam.

4.4 Proposta de processo de inspeção dos artefatos de requisitos de *software*

O roteiro de teste de inspeção proposto foi modelado na ferramenta Bizagi Modeler, utilizando a notação BPMN para que fosse possível ser formalizado e integrado ao processo macro da FTT. Baseado no processo tradicional de (FAGAN, 1976), o qual é utilizado por equipes síncronas, e se divide em cinco atividades que se encaixam com as etapas do processo atual e com as práticas de inspeção vigentes, como demonstram as Figura 5 e Figura 6.

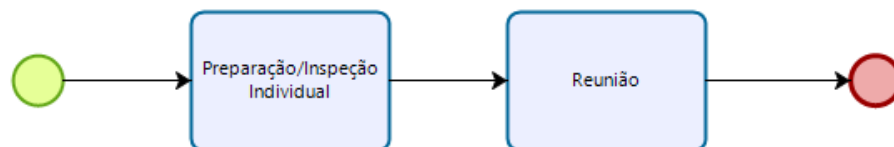
Figura 5 - Sub-processo da inspeção de requisitos



Powered by
bizagi
Modeler

Fonte: Autores, 2020

Figura 6 - Validação dos artefatos



Powered by
bizagi
Modeler

Fonte: Autores, 2020

Cada símbolo de tarefa da notação BPMN representa uma atividade do processo. O fluxo foi modelado considerando a finalização da *sprint* da equipe de análise de requisitos, com a execução da inspeção dentro da *sprint* da equipe de testes para que seja possível realizar a elaboração dos casos de teste ao seu final.

A atividade de planejamento pode acontecer na reunião de planejamento do *scrum*, ou fora dela caso seja necessário incluir um novo requisito para inspeção, nesta atividade o

moderador informará quem participará das inspeções, quais artefatos serão inspecionados, quais técnicas serão utilizadas nas detecções, qual o objetivo da inspeção e realiza a distribuição do material a ser inspecionado.

A próxima atividade a ser desenvolvida é a preparação, cada inspetor realizará individualmente a inspeção dos artefatos de sua responsabilidade, neste momento o inspetor utiliza técnicas de leitura para auxiliar na busca por erros, utilizando a definição apresentada no item 3.3, os defeitos encontrados no artefato são anotados, produzindo uma lista de discrepâncias, dúvidas e sugestões que devem ser classificadas utilizando a definição obtida a partir de Kirner e Abib (1997).

Após a inspeção individual, será realizada a reunião com a presença do moderador, o relator, o inspetor e o autor do documento, nesta atividade ocorrerá a discussão e classificação das discrepâncias como defeitos ou falso positivo. Em cada caso o moderador tem a decisão final, também deve garantir que não será discutida a solução dos defeitos para não exceder o tempo da reunião. O papel do relator nesta reunião é documentar as decisões que foram tomadas.

A atividade seguinte a ser realizada será a correção, nesta fase os artefatos retornam para a equipe de análise de requisitos, onde o autor do documento realiza as correções identificadas na inspeção.

O acompanhamento é a última atividade do roteiro, quando o documento corrigido pelos autores é finalizado e então o moderador reavalia a qualidade do artefato, ele define se o requisito está atendendo as condições necessárias de desenvolvimento ou se é necessário ocorrer uma nova inspeção.

Esse fluxo possibilita que além da equipe de verificação e validação, outros membros em funções de gerencia e supervisão tenham conhecimento do estado do requisito e sua complexidade, facilita a análise de conhecimento dos envolvidos, tornando possível identificar ferramentas ou estudos direcionados a dificuldades individuais. A formalização do roteiro de inspeção ainda contribui com a transferência de conhecimento em um ambiente com alta rotatividade de membros, pois tem mais indivíduos envolvidos nas decisões tomadas.

4.5 Comparação dos modelos de inspeção

A comparação dos modelos de inspeção de requisitos de *software* realizadas na FTT

antes e depois da intervenção foi possível através de observações *in loco*, análise de relatório dos testes e aplicação de questionário.

Os dados coletados foram analisados e comparados para então definir se houve melhoria após a intervenção e formalizar o fluxo do processo de inspeção. Na comparação entre os dois modelos foi possível notar pontos importantes que são sanados pela aplicação de um roteiro de inspeção e que não são atendidos pela inspeção simples realizada anteriormente, como mostra o Quadro 3.

O Quadro 3 evidencia que atividades essenciais do processo de inspeção já existiam dentro da FTT como: a) Auxílio à leitura dos documentos com ferramenta checklist; b) Reunião para solucionar dúvidas; c) inspeção individual; d) classificação de defeitos; e) métricas de inspeção. Apesar disso, essas atividades eram realizadas sem um controle formal e de acordo com a necessidade e experiência do inspetor e, portanto, não eram satisfatórias como pode ser observado no item 4.8. Com a aplicação do roteiro proposto, outros benefícios são incluídos na inspeção de requisitos de software, isso porque, todas as fases da inspeção passam a ser monitoradas e controladas, a responsabilidade dividida e os resultados compartilhados.

Quadro 3 - Comparação dos modelos de inspeção na FTT

	INSPEÇÃO DE REQUISITOS	ROTEIRO DE INSPEÇÃO DE REQUISITOS
Detalhamento das atividades de inspeção	-	X
Planejamento da inspeção	-	X
Ferramentas de auxílio à leitura	X	X
Reunião para solucionar dúvidas	X	X
Inspeção individual minuciosa	X	X
Reunião de verificação de defeitos	-	X
Classificação de defeitos	X	X
Métricas de inspeção	X	X
Acompanhamento de desempenho do inspetor	-	X
Transferência de conhecimento	-	X
Colaboração de outras equipes	-	X
Diminuição do tempo de retrabalho	-	X
Controle do fluxo do requisito	-	X

Fonte: Autores, 2020

4.6 Questionário de avaliação

O segundo questionário aplicado foi a avaliação do estudo, este foi respondido ao final da pesquisa, suas questões se referem a avaliação do processo proposto, seus benefícios ao ambiente e aos colaboradores e a complexidade das técnicas utilizadas. Na seção resultados obtidos, as informações colhidas através dele serão discriminadas.

4.7 Entrevista

Para realizar o mapeamento do teste de inspeção e identificar os desafios enfrentados pela equipe foram realizadas entrevistas com os integrantes que tem relação direta ou indireta com este tipo de teste, em visitas *in loco*.

As entrevistas mostraram que apesar da equipe possuir grande comunicação e boas práticas como solicitar uma segunda opinião entre os membros da equipe e até mesmo com membros de outras equipes, a FTT ainda não possui um processo de inspeção formalizado, isso faz com que ao mudar os integrantes da equipe as informações e boas práticas sejam facilmente perdidas.

O processo de inspeção proposto teve grande aceitação entre os entrevistados, que o julgaram como necessário, alguns informaram dificuldade em algum aspecto, como classificação de defeitos ou uso de técnicas de leitura, o que poderá ser superado através de treinamentos e a prática diária.

Parte dos entrevistados demonstraram apreensão em relação ao tempo que será gasto na inspeção, o roteiro torna o teste organizado e gerenciável, a formalização do processo revela atividades que antes não eram percebidas, mas de maneira informal já eram realizadas e facilita o treinamento de novos membros, desta forma as inspeções não serão mais extensas e sim mais controladas.

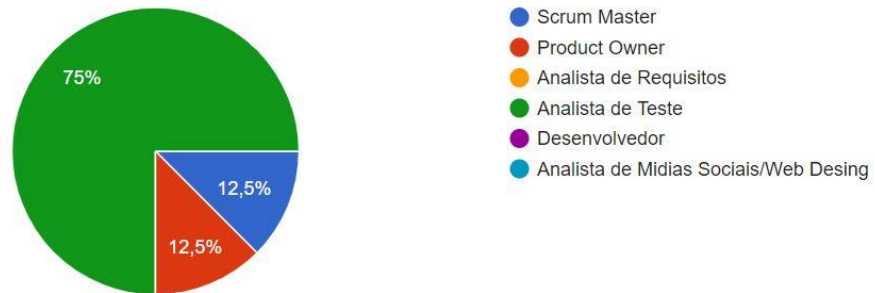
4.8 Resultados obtidos

Após a aplicação do processo proposto, foram coletados os dados obtidos para que fosse realizada a análise e interpretação dos mesmos. Com a análise das informações foi possível notar mudanças positivas no controle e acompanhamento das inspeções. Os resultados revelam que dos oito integrantes da Fábrica de Tecnologias Turing que participaram da pesquisa, 70% o equivalente a seis pessoas faz parte da equipe de testes, 12,5% ou 1 pessoa é *Scrum Master* e 12,5% ou 1 pessoa é *Product Owner*, como é mostrado na Figura 7.

Figura 7 - Função do entrevistado na empresa

Função na empresa

8 respostas



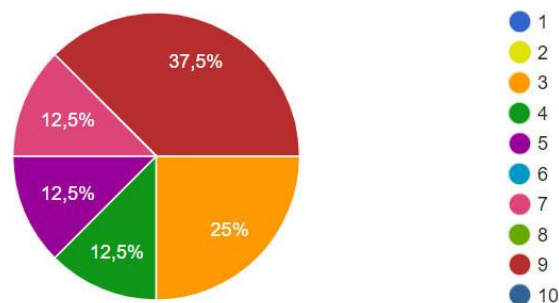
Fonte: Autores, 2020

A Figura 8 mostra que destes oito integrantes entrevistados, 37,5% ou 3 pessoas estão cursando o 9º período do curso de engenharia da computação/software, 25% ou 2 pessoas cursam o 3º período, 12,5% ou 1 pessoa cursa o 4º período, 12,5% cursa o 5º período e 12,5% cursa o 7º período, Sendo assim, 50% dos entrevistados já estão finalizando o curso, enquanto os outros 50% ainda não passaram por disciplinas fundamentais como a verificação e validação de software.

Figura 8 - Período cursado pelo entrevistado

Qual período está cursando?

8 respostas



Fonte: Autores, 2020

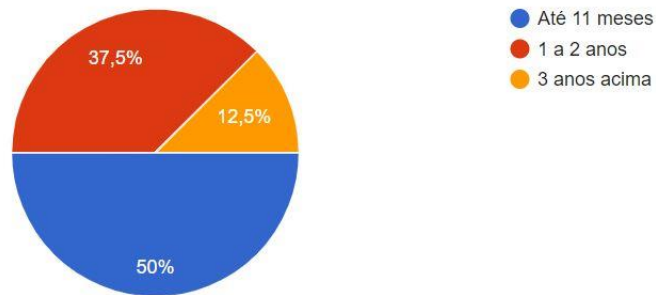
Observando a Figura 9 é possível concluir que 50% dos participantes possui menos de

um ano de atuação na área de TI, outros 37,5% possui de um a dois anos de atuação e somente 12,5% atuam de três anos acima. Aqui são considerados como tempo de atuação na área de TI tanto a vivência em fábrica de desenvolvimento acadêmica como a atuação em empresas da área.

Figura 9 - Tempo que o entrevistado atua na área de TI

Quanto tempo atua na área de TI:

8 respostas



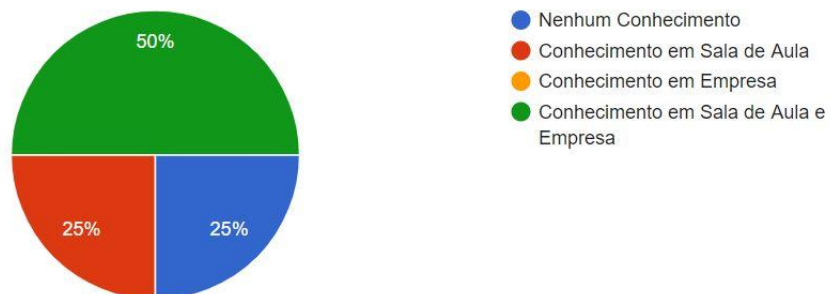
Fonte: Autores, 2020

Para compreender o nível de conhecimento dos entrevistados em relação a inspeção de artefatos de software, foi apontado que 50% alegaram possuir conhecimento sobre inspeção em sala de aula e empresa, 25% afirmaram ter conhecimento somente em sala de aula e 25% alegaram possuir nenhum conhecimento de inspeção, como mostrado pela Figura 10.

Figura 10 - Conhecimento do entrevistado sobre inspeção

Conhecimento sobre as Técnicas de Inspeção:

8 respostas



Fonte: Autores, 2020

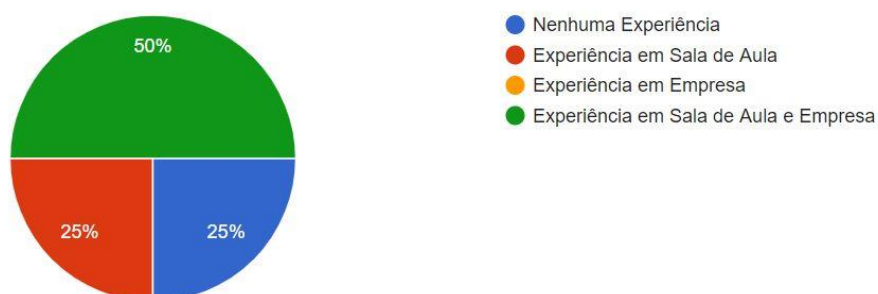
Quando perguntados sobre a experiência prática, os dados mostram igualdade em relação a teoria, sendo 50% dos entrevistados alegando possuir experiência sobre inspeção em sala de aula e empresa, 25% afirmando ter conhecimento somente em sala de aula e 25% considerando possuir nenhuma experiência e conhecimento de inspeção, como é possível observar na Figura 11.

Para confirmar a veracidade dos dados, vale frisar que todos os alunos para ingressar na Fábrica de Tecnologias Turing, passam por um processo de treinamento em sala de aula, esse período é composto por capacitação teórica e prática, abordando os vários processos e práticas executados na FTT. Então, mesmo que não tenham cursado disciplinas pertinentes no período do curso, eles possuem uma visão do tema. Neste caso, o conhecimento de inspeção é válido e serve de base para o entendimento e aplicação do roteiro de inspeção

Figura 11 - Experiência do entrevistado em inspeção

Experiência sobre Inspeção de Software:

8 respostas

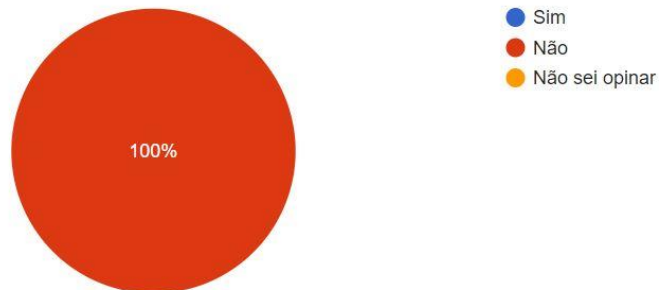


Fonte: Autores, 2020

Na investigação realizada no início da pesquisa, com o intento de identificar se a FTT possuía um processo de inspeção de requisitos de software, e se caso existisse, esse processo estaria atendendo as necessidades, foram obtidas as seguintes respostas: apesar de ser executado o teste de inspeção na Fábrica de Tecnologias Turing, todos os entrevistados consideram que a FTT não possui um subprocesso de inspeção de requisitos de *software*, como pode ser analisado na Figura 12, onde os papéis e etapas são bem definidos. Cada inspetor segue orientações de boas práticas, porém não existia um roteiro formalizado e sim uma atividade de inspeção no processo macro.

Figura 12 - Sub-processo de inspeção na FTT

A FTT possui um subprocesso de teste de inspeção?



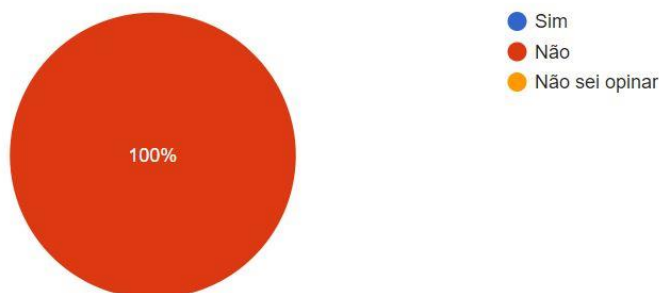
Fonte: Autores, 2020

Neste contexto, apesar das inspeções estarem sendo realizadas e ter contribuído nos diversos projetos em andamento, as necessidades da fábrica de tecnologias não têm sido supridas de forma satisfatória. O formato de teste executado no ambiente de desenvolvimento não tem sido favorável para fornecer resultados consistentes ou avaliar o desempenho dos testadores, e pode sofrer mudanças que contribuem positivamente com os resultados esperados, como mostrado na Figura 13.

Figura 13 - Necessidades da FTT em relação a inspeção

(Se houver) O subprocesso de teste de inspeção atual, está atendendo as necessidades da FTT?

8 respostas



Fonte: Autores, 2020

De acordo com as Figura 14, no questionário de diagnóstico do estudo, os entrevistados afirmam ainda que, a equipe de testes possui como ferramentas de inspeção um *checklist* para

auxiliar na identificação dos defeitos e a ferramenta Mantis Bug Tracker para relatar os defeitos encontrados.

Figura 14 - Ferramenta de auxílio a leitura

A equipe de teste possui ferramentas para auxílio da leitura dos requisitos?

8 respostas

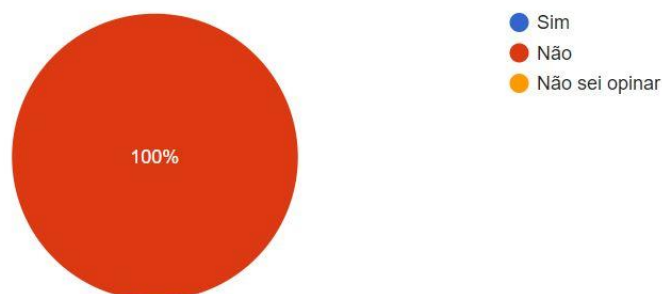


Fonte: Autores, 2020

Por ser um questionário extensa e muito detalhado, o *checklist* se tornou cansativo para os testadores que optam pela não utilização. O sistema Mantis, usado para relatar as discrepâncias encontradas, embora possua muitas funcionalidades próprias para testes, não tem sido totalmente explorado para auxiliar nas inspeções, por falta de familiaridade da equipe com a ferramenta, como pode ser observado na Figura 15.

Figura 15 - Eficiência das ferramentas de auxílio a inspeção

(Se houver) As ferramentas auxiliam efetivamente os testadores?



Fonte: Autores, 2020

A ferramenta Mantis, possibilita que sejam utilizadas métricas para acompanhar os resultados de cada inspeção, no entanto, apenas um relatório de quantidade de erros encontrados

é gerado pela equipe, como mostrado na Figura 16. Para os testes de inspeção existem uma série de métricas que podem ser aplicadas para fornecer resultados importantes para tomadas de decisões no projeto em desenvolvimento e também nas estimativas de projetos futuros, como descrito no item 3.4.4.

Figura 16 - Métricas de inspeção

A equipe de teste, adota métricas de avaliação e análise dos resultados dos testes de inspeção?

8 respostas



Fonte: Autores, 2020

Para avaliar o desempenho dos inspetores durante a etapa de inspeção individual prevista no subprocesso, foram utilizados três documentos de requisitos de um dos processos em andamento da FTT. Esses documentos já haviam sido especificados, inspecionados e finalizados, deste modo, para a realização do experimento foram inseridos defeitos no requisito e em seus artefatos complementares, de acordo com o (APÊNDICE D). Nesta fase foi possível notar que, mesmo após a ferramenta *checklist* (APÊNDICE C), sofrer mudanças e se adaptar a principal queixa dos inspetores, tornando-se reduzido e funcional, houve dificuldade em encontrar defeitos simples e classifica-los corretamente, conforme o gabarito de defeitos destes mesmos documentos de requisitos (APÊNDICE D), deixando evidente a pouca experiência da equipe neste tipo de teste mesmo com ferramenta de auxílio à leitura. Diante dos resultados negativos apresentados nas detecções, o estudo se mostrou eficaz na identificação das deficiências ocorridas na equipe, como mostra o (APÊNDICE A).

No Quadro 4 e Quadro 5 é possível notar que a quantidade de defeitos encontrados pelos inspetores está muito abaixo da quantidade de defeitos inseridos nos documentos de requisitos de *software* e a classificação destes defeitos tem resultados ainda menores, as técnicas de leitura não apresentaram grande diferença na quantidade de defeitos encontrados. A média de tempo

gasto nas inspeções individuais foram de 2 horas, tanto em documentos menores como nos maiores.

No Quadro 4, usando a técnica *checklist* no documento UC032 - Definir cronograma de elaboração o inspetor 1 encontrou um total de 3 defeitos válidos e o inspetor 5 encontrou 5 defeitos válidos de 14 que haviam sido inseridos no requisito. No UC019 – Elaborar plano de ensino foi encontrado pelo inspetor 2 a quantidade de 12 erros, pelo inspetor 3 foi encontrado 6 erros de um total de 74 erros inseridos. No UC008 – Manter competências e habilidades o inspetor 4 encontrou 7 erros e o inspetor 6 encontrou 8 erros de 20 inseridos no documento. Esses resultados podem significar que os inspetores não possuem muita experiência com a ferramenta e ela não é utilizada adequadamente.

Quadro 4 - Defeitos detectados com técnica checklist

TÉCNICA CHECKLIST	INSPECTOR 1		INSPECTOR 2		INSPECTOR 3		INSPECTOR 4		INSPECTOR 5		INSPECTOR 6		
	QUANTIDADE DE DEFEITOS INSERIDOS	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA
UC032 - DEFINIR CRONOGRAMA DE ELABORAÇÃO	14	3	1	-	-	-	-	-	-	5	0	-	-
UC019 - ELABORAR PLANO DE ENSINO	74	-	-	12	11	6	2	-	-	-	-	-	-
UC008 - MANTER COMPETÊNCIAS E HABILIDADES	20	-	-	-	-	-	-	7	3	-	-	8	4

Fonte: Autores, 2020

No Quadro 5, usando a técnica *AdHoc* no documento UC032 - Definir cronograma de elaboração o inspetor 2 encontrou um total de 4 defeitos válidos, o inspetor 4 encontrou 4 defeitos válidos e o inspetor 6 também encontrou 4 defeitos válidos de 14 que haviam sido inseridos no requisito. No UC019 – Elaborar plano de ensino foi encontrado pelo inspetor 1 a quantidade de 6 erros de um total de 74 erros inseridos. No UC008 – Manter competências e habilidades o inspetor 3 encontrou 7 erros e o inspetor 5 encontrou 13 erros de 20 inseridos no documento. Esses resultados podem indicar que os inspetores possuem mais facilidade em

identificar defeitos mais comuns e que tem maior incidência, como defeitos no padrão do documento e ortografia, enquanto defeitos de estrutura do sistema e negócio são menos encontrados.

Quadro 5 - Defeitos detectados com técnica AdHoc

TÉCNICA ADHOC	INSPECTOR 1		INSPECTOR 2		INSPECTOR 3		INSPECTOR 4		INSPECTOR 5		INSPECTOR 6		
	QUANTIDADE DE DEFEITOS INSERIDOS	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA	DEFEITOS ENCONTRADOS	CLASSIFICAÇÃO CORRETA
UC032 - DEFINIR CRONOGRAMA DE ELABORAÇÃO	14	-	-	4	2	-	-	4	1	-	-	4	2
UC019 - ELABORAR PLANO DE ENSINO	74	6	2	-	-	-	-	-	-	-	-	-	-
UC008 - MANTER COMPETENCIAS E HABILIDADES	20	-	-	-	-	7	5	-	-	13	0	-	-

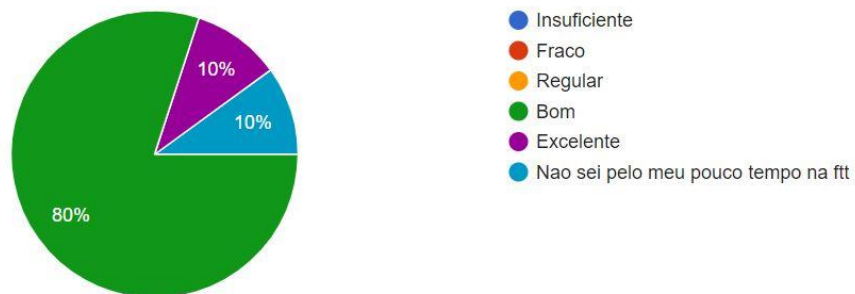
Fonte: Autores, 2020

O último questionário aplicado, foi a avaliação do estudo, 90% dos entrevistados classificaram de forma positiva o subprocesso de teste de inspeção e sua contribuição para a Fábrica de Tecnologias Turing como mostra a Figura 17.

Figura 17 - Contribuição do roteiro de inspeção na FTT

De modo geral, o Subprocesso de Inspeção contribui de forma positiva na rotina da FTT?

10 respostas



Fonte: Autores, 2020

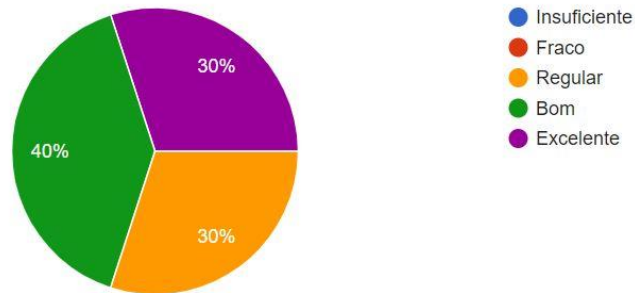
Os artefatos e técnicas disponibilizados na intervenção tiveram a aprovação dos entrevistados, porém foram feitas considerações pelos inspetores que acreditam que mesmo

sendo de fácil entendimento será necessário mais tempo de utilização para se habituarem ao processo, técnicas e ferramentas, como mostrado na Figura 18.

Figura 18 - Entendimento dos inspetores em relação ao roteiro

De modo geral, o Subprocesso de Inspeção é de fácil entendimento?

10 respostas



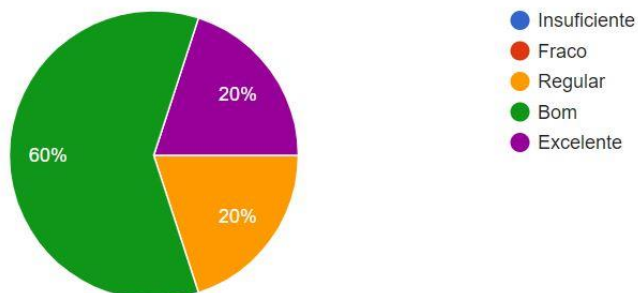
Fonte: Autores, 2020

Através de relatos dos entrevistados e dados da Figura 19 é possível notar que os participantes adquiriram algum conhecimento a partir do estudo realizado. Mesmo com alguma dificuldade os entrevistados relataram que o estudo contribuiu para o entendimento dos conceitos e benefícios do teste de inspeção.

Figura 19 - Conhecimento adquirido a partir da pesquisa

A partir do estudo realizado com o documento de requisitos de software, você considera ter adquirido conhecimento sobre a técnica de inspeção?

10 respostas



Fonte: Autores, 2020

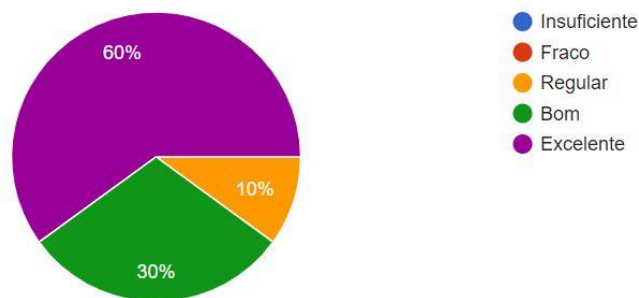
Os entrevistados mostram ainda grande disposição em aplicar o processo de inspeção em outros ambientes de desenvolvimento. Eles acreditam que podem usar o processo de

inspeção proposto em projetos futuros e até mesmo fora da Fábrica de Tecnologias Turing, como mostra a Figura 20.

Figura 20 - Utilização da técnica em outros ambientes

Você utilizaria a Técnica de Inspeção em um projeto de desenvolvimento de software no seu ambiente de trabalho?

10 respostas



Fonte: Autores, 2020

5 Conclusão

A realização desta pesquisa revelou a importância de um roteiro de inspeção com técnicas, ferramentas e etapas detalhadas em uma fábrica de desenvolvimento de *software*. Na FFT foi identificado a partir dos relatos e observações a atividade de inspeção realizada pela equipe de verificação e validação, essa atividade serviu de ponto de partida para o estudo das práticas realizadas neste tipo de teste. A partir da identificação dos pontos principais na inspeção de documentos de requisitos, foi definido um novo modelo de teste de inspeção.

A partir da aplicação e comparação com o modelo anterior, o novo modelo com roteiro de inspeção formalizado gerou melhoria no acompanhamento do progresso dos requisitos, a identificação das dificuldades vivenciadas pela equipe tornou-se mais acessível, facilitou a identificação das falhas em etapas específicas como a inspeção individual, aumentou a colaboração de profissionais de outras equipes na inspeção e conseqüentemente a transferência de conhecimento e a rotina de inspeção realizada em cada documento ficou mais clara.

Através dos dados gerados pelas detecções foi possível entender que por se tratar de uma equipe com novos integrantes, a maioria em períodos iniciais do curso e com pouca ou nenhuma experiência profissional, os participantes possuem dificuldades em encontrar defeitos relacionados as regras de negócio e arquitetura do sistema, enquanto possuem mais facilidades

para identificar defeitos relacionados a padronização e estrutura da documentação.

Em razão do momento de pandemia em que o país está vivenciando, a intervenção e aplicação do roteiro de inspeção, precisou ser realizada virtualmente, o que dificultou o entendimento completo dos resultados e impactos gerados pelo trabalho. A dificuldade em aplicar e sanar todas as dúvidas dos participantes durante a pesquisa, tornou inviável a utilização de métricas e medições de inspeção, porém serão deixadas a disposição para novas pesquisa e utilização futura.

5.1 Trabalhos futuros

A partir dos resultados dessa pesquisa, foi possível identificar possíveis melhoria no roteiro de inspeção proposto e sua aplicação, como: a aplicação das métricas de inspeção, treinamento da equipe em relação a detecção e classificação de defeitos, configuração e uso de ferramentas como o Mantis em benefício do teste de inspeção.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AAEN, I.; BOTCHER, P.; MATHIASSEN, L. The Software Factory: Contributions and Illusions. **Proceedings of the Twentieth information Systems Research Seminar in Scandinavia**, p. 407–433, 1997.

AGILE MANIFESTO. **Manifesto for Agile Software Development**. Disponível em: <<http://agilemanifesto.org/principles.html>>. Acesso em: 29 mar. 2020.

BARBOSA, M. **O Que É Scrum E Como Executar Suas Tarefas Na Metade Do Tempo?** Disponível em: <<https://outboundmarketing.com.br/o-que-e-scrum/>>. Acesso em: 29 mar. 2020.

BERTINI, L. A. **Técnicas de Inspeção Aplicadas à Avaliação de Requisitos de Sistemas de Software: Um Estudo Comparativo**. Piracicaba: Universidade Metodista de Piracicaba – UNIMEP, 2006.

DOOLAN, E. P. Experience with Fagan’s inspection method. **Software: Practice and Experience**, v. 22, n. 2, p. 173–182, 1992.

FAGAN, M. E. Design and code inspections to reduce errors in program development. **IBM System Journal**, v. 15, n. 3, p. 182–211, 1976.

FARIA, A. D. A. et al. **Guia prático de funcionamento da Fábrica de Tecnologias Turing (FTT)**Anápolis, 2018.

FILHO, W. DE P. P. **ENGENHARIA DE SOFTWARE: FUNDAMENTOS, MÉTODOS E PADRÕES**. 3. ed. Rio de Janeiro: XXI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO – ENEGEP, 2009.

KALINOWSKI, M.; SPÍNOLA, R. O. Introdução à Inspeção de Software. **Engenharia de Software Magazine**, n. 1, p. 68–74, maio 2014.

KIRNER, T. G.; ABIB, J. C. Inspection of software requirements specification documents: A pilot study. **ACM SIGDOC Annual International Conference on Computer Documentation, Proceedings**, n. April, p. 161–171, 1997.

MILLER, E. The Philosophy of Testing, in Program Testing Techniques. **IEEE Computer Society Press**, p. 1–3, 1977.

OLIVEIRA, C. **Utilização de checklist para validação de requisitos de software _ iMasters**. Disponível em: <<https://imasters.com.br/devsecops/utilizacao-de-checklist-para-validacao-de-requisitos-de-software>>. Acesso em: 4 out. 2019.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software UMA ABORDAGEM PROFISSIONAL**. 8. ed. Porto Alegre: AMGH, 2016.

ROMANHA, S. **Fábrica de Software: Vantagens da Implementação do Conceito em Instituição de Ensino Superior. XII Simpósio de Excelência em Gestão e Tecnologia**, 2015.

ROMANHA, S. D.; MUNIZ JR., J.; DALE LUCHE, J. R. **Fábrica de software em instituições de ensino superior: análise de universidades brasileiras. Revista Produção Online**, v. 19, n. 2, p. 408–429, 2019.

SCHWABER, K.; SUTHERLAND, J. Guia do SCRUM™. **Harvard Business Review, Boston**, v. IV, p. 163–179, 2013.

SCRUMSTUDY™. **SCRUM Body of Knowledge (SBOK™ Guide) - 2016 Edition**. Disponível em: <<https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide->

2016.pdf>. Acesso em: 30 jan. 2020.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice, 2011.

SOUSA, F. C. et al. Preparação dos Acadêmicos dos Cursos de Computação da UniEVANGÉLICA para o Mercado de Trabalho através de uma Fábrica de Software Acadêmica. **Revista Ada Lovelace**, v. 2, p. 16–26, 2018.

SOUSA, J. V. M. Inspeção de Software - Experiência na Fábrica de Tecnologias Turing (FTT). **Revista Ada Lovelace**, v. 2, p. 57–60, 2018.

UNIEVANGÉLICA. **Engenharia de Computação - Diferenciais**. Disponível em: <<http://www.unievangelica.edu.br/curso.engenharia-de-computacao/diferenciais/>>. Acesso em: 28 jan. 2020.

XAVIER, C. D. **FÁBRICA DE SOFTWARE: ATÉ QUE PONTO FORDISTA?**Rio de Janeiro, 2008.

ZANIN, A. et al. **Qualidade de Software**. Porto Alegre: SAGAH, 2018.

APÊNDICE A – Documento de Relatório de Inspeção

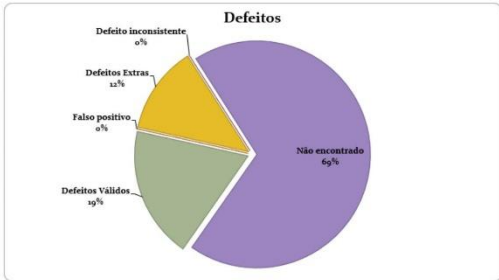
Figura 21- Relatório de inspeção pag.1



Figura 22- Relatório de inspeção pag.2

REGISTRO DE INSPEÇÃO		
Número da Inspeção: 001	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 1	
Requisito	UC032 Definir Cronograma de elaboração	
Técnica	Checklist	


OBJETIVO	
Avaliar a situação do teste de inspeção de documento de requisitos de software na FTT	

RESULTADOS													
<table border="1"> <thead> <tr> <th>Tipo de defeitos</th> <th>Quantidade</th> </tr> </thead> <tbody> <tr> <td>Defeitos Válidos</td> <td>3</td> </tr> <tr> <td>Falso positivo</td> <td>0</td> </tr> <tr> <td>Defeitos Extras</td> <td>2</td> </tr> <tr> <td>Defeito inconsistente</td> <td>0</td> </tr> <tr> <td>Não encontrado</td> <td>11</td> </tr> </tbody> </table>	Tipo de defeitos	Quantidade	Defeitos Válidos	3	Falso positivo	0	Defeitos Extras	2	Defeito inconsistente	0	Não encontrado	11	
Tipo de defeitos	Quantidade												
Defeitos Válidos	3												
Falso positivo	0												
Defeitos Extras	2												
Defeito inconsistente	0												
Não encontrado	11												
<p>Obs: Dos 3 defeitos válidos encontrados de 14, apenas 1 foi classificado corretamente.</p>													


PRIORIDADE	ALTA
------------	------

INFORMAÇÕES COMPLEMENTARES	
Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 23- Relatório de inspeção pag.3



**Bacharelados em
Computação**



**Fábrica de Tecnologias
Turing**

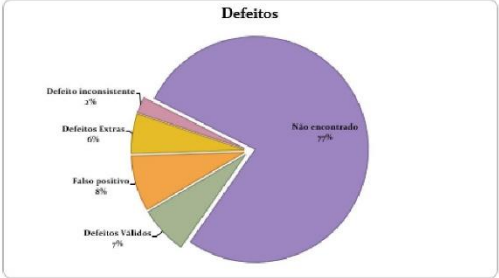
Número da Inspeção: 002	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 1	
Requisito	UC019 Elaborar Plano de Ensino	
Técnica	Ad Hoc	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FTT

RESULTADOS

Tipo do defeito	Quantidade
Defeitos Válidos	6
Falso positivo	7
Defeitos Extras	5
Defeito inconsistente	2
Não encontrado	88



Defeitos

Obs: Dos 6 defeitos válidos encontrados de 74, apenas 2 foram classificados corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 24-Relatório de inspeção pag.4

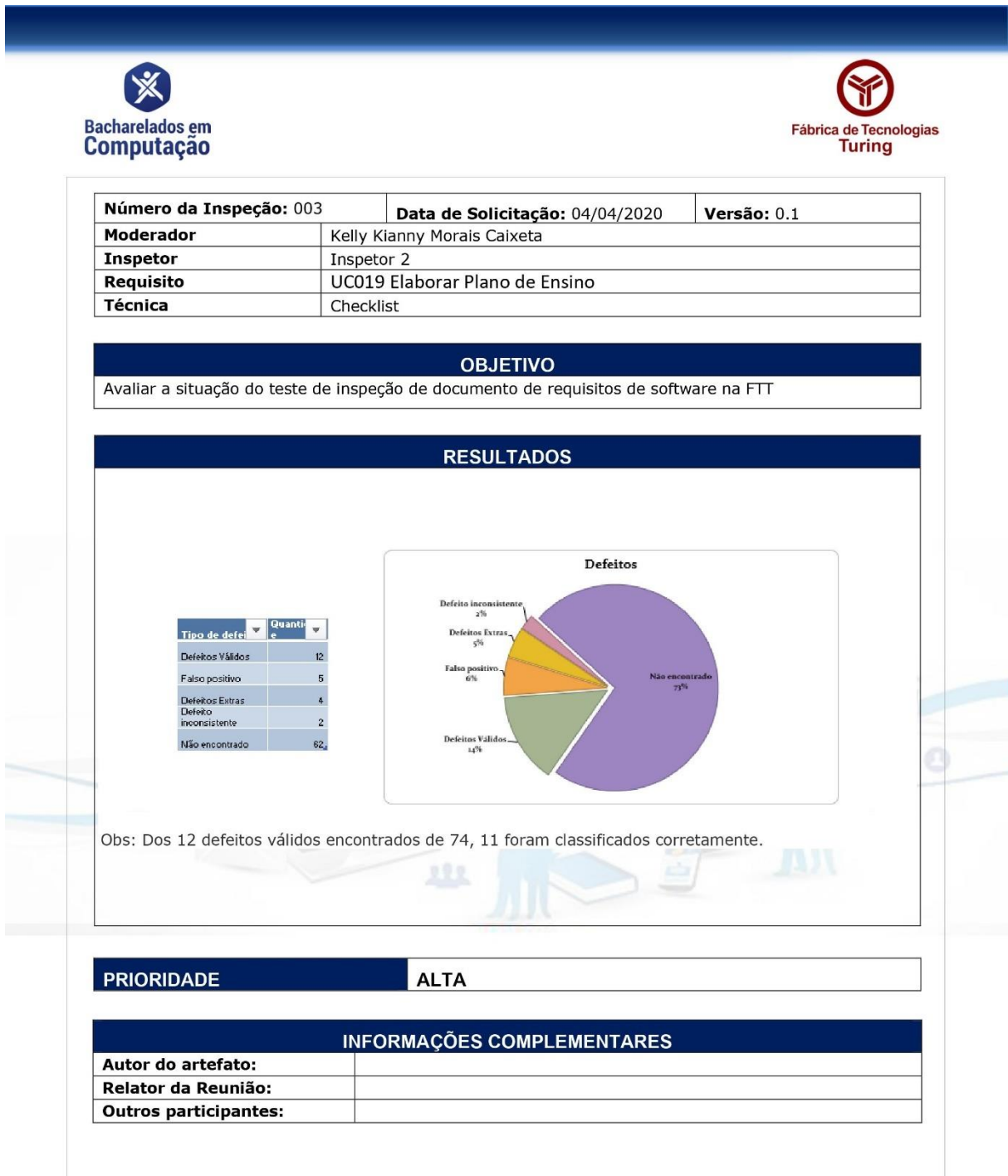


Figura 25-Relatório de inspeção pag.5

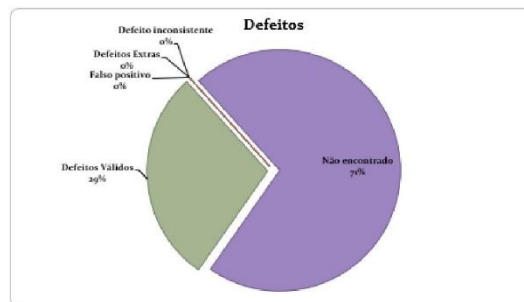
Número da Inspeção: 004	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 2	
Requisito	UC032 Definir Cronograma de elaboração	
Técnica	Ad Hoc	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FFT

RESULTADOS

Tipo de defeito	Quantidade
Defeitos Válidos	4
Falso positivo	0
Defeitos Extras	0
Defeito inconsistente	0
Não encontrado	10



Obs: Dos 4 defeitos válidos encontrados de 14, apenas 2 foram classificados corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 26- Relatório de inspeção pag.6

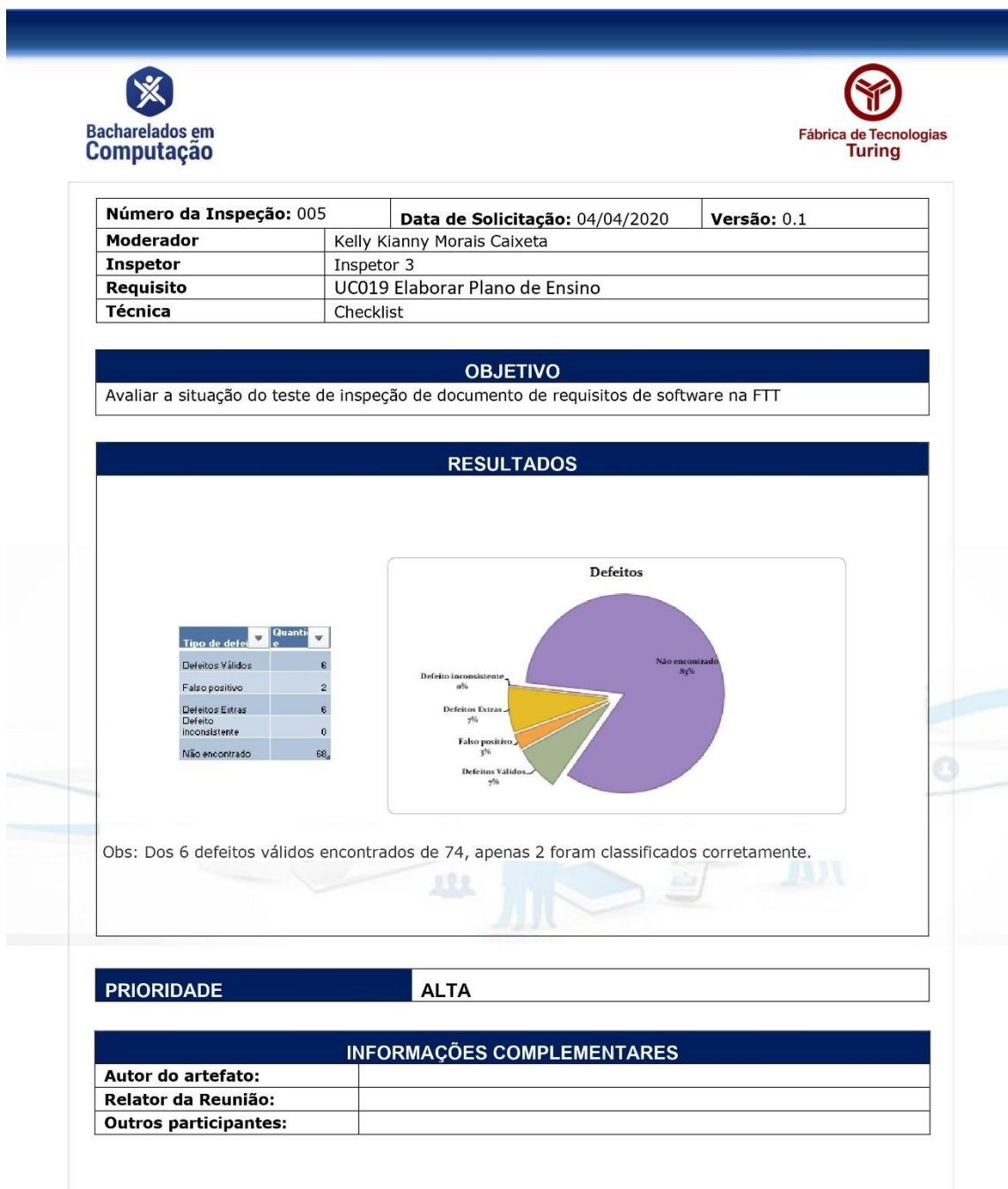


Figura 27- Relatório de inspeção pag.7

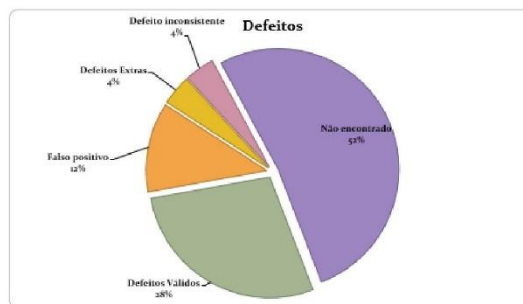
Número da Inspeção: 006	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Moraes Caixeta	
Inspetor	Inspetor 3	
Requisito	UC008 Manter Competências e Habilidades	
Técnica	Ad Hoc	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FFT

RESULTADOS

Tipo de defeito	Quantidade
Defeitos Válidos	7
Falso positivo	3
Defeitos Extras	1
Defeito inconsistente	1
Não encontrado	13



Obs: Dos 7 defeitos válidos encontrados de 20, apenas 5 foram classificados corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 28- Relatório de inspeção pag.8

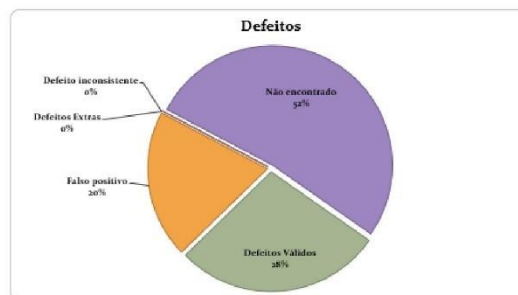
Número da Inspeção: 007	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 4	
Requisito	UC008 Manter Competências e Habilidades	
Técnica	Checklist	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FTT

RESULTADOS

Tipo de defeito	Quantidade
Defeitos Válidos	7
Falso positivo	5
Defeitos Extras	0
Defeito inconsistente	0
Não encontrado	13



Obs: Dos 7 defeitos válidos encontrados de 20, apenas 3 foram classificados corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 29- Relatório de inspeção pag.9

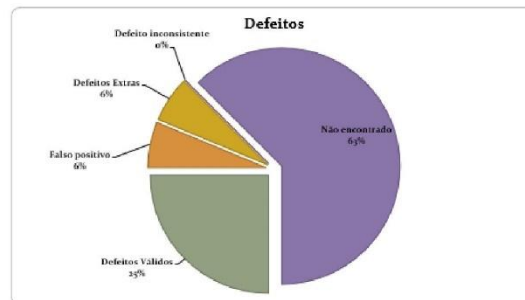
Número da Inspeção: 008	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 4	
Requisito	UC032 Definir Cronograma de elaboração	
Técnica	Ad Hoc	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FTT

RESULTADOS

Tipo de defeito	Quantidade
Defeitos Válidos	4
Falso positivo	1
Defeitos Extras	1
Defeito Inconsistente	0
Não encontrado	10



Obs: Dos 4 defeitos válidos encontrados de 14, apenas 1 foi classificado corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 30- Relatório de inspeção pag.10

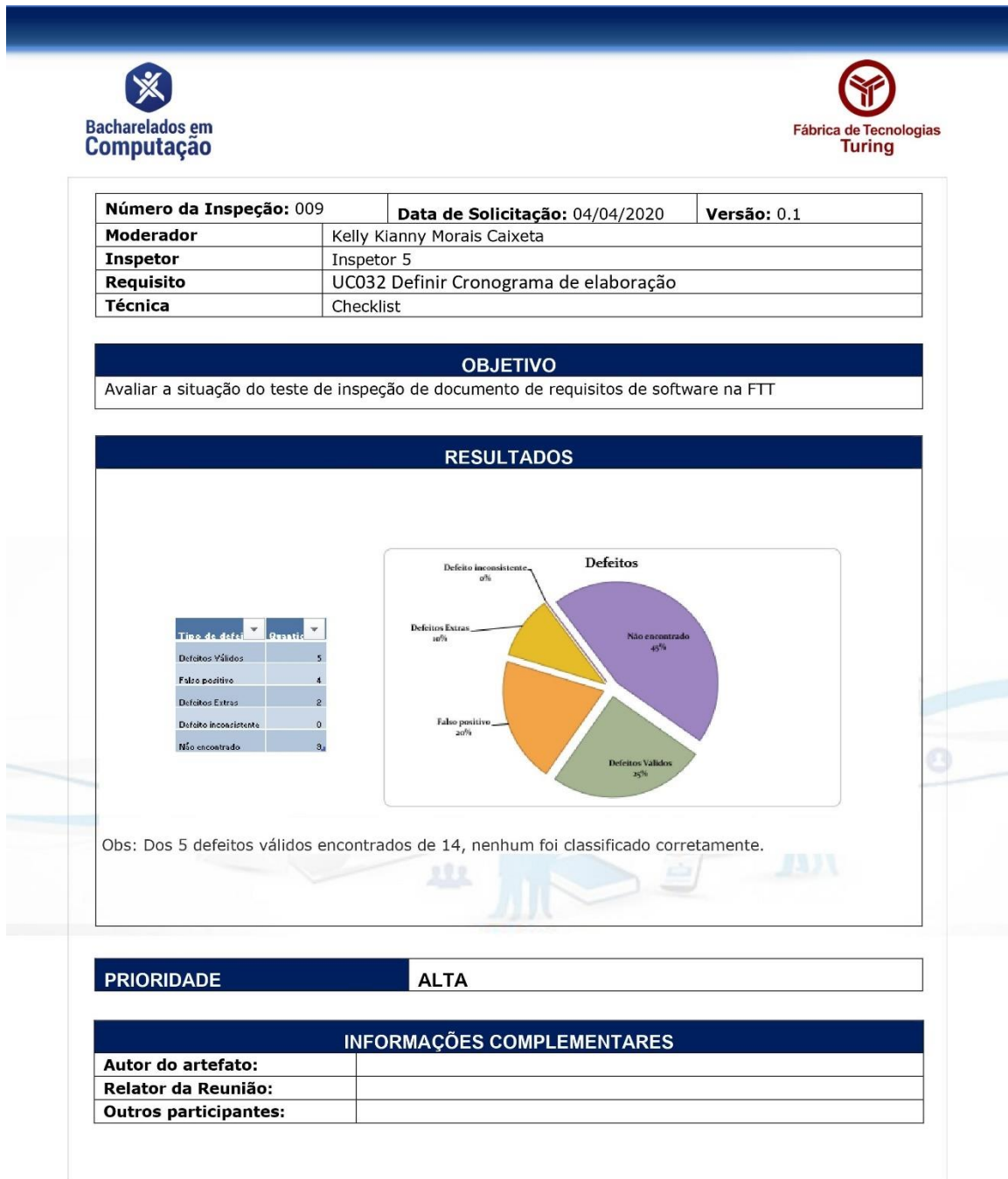


Figura 31- Relatório de inspeção pag.11



Bacharelados em
Computação



Fábrica de Tecnologias
Turing

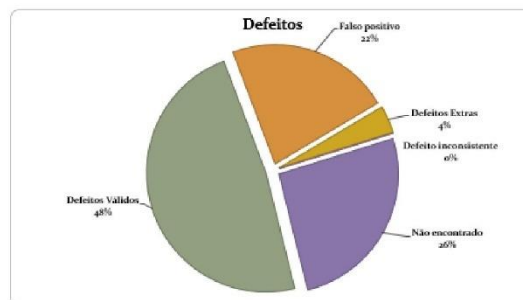
Número da Inspeção: 010	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 5	
Requisito	UC008 Manter Competências e Habilidades	
Técnica	Ad Hoc	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FFT

RESULTADOS

Tipo de defeito	Quantidade
Defeitos Válidos	13
Falso positivo	6
Defeitos Extras	1
Defeito inconsistente	0
Não encontrado	7



Obs: Dos 13 defeitos válidos encontrados de 20, nenhum foi classificado corretamente.

PRIORIDADE

ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Av. Universitária Km 3,5 - Cidade Universitária - Anápolis/GO, CEP 75083-515
Secretaria Setorial - Bloco I
Fone: (62) 3310 6658 | FAX: (62) 3310 6607

Figura 32- Relatório de inspeção pag.12

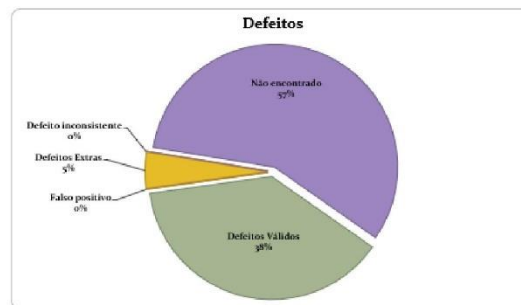
Número da Inspeção: 011	Data de Solicitação: 04/04/2020	Versão: 0.1
Moderador	Kelly Kianny Morais Caixeta	
Inspetor	Inspetor 6	
Requisito	UC008 Manter Competências e Habilidades	
Técnica	Checklist	

OBJETIVO

Avaliar a situação do teste de inspeção de documento de requisitos de software na FTT

RESULTADOS

Tipo de defeitos	Quantidade
Defeitos Válidos	8
Falso positivo	0
Defeitos Extras	1
Defeito inconsistente	0
Não encontrado	12



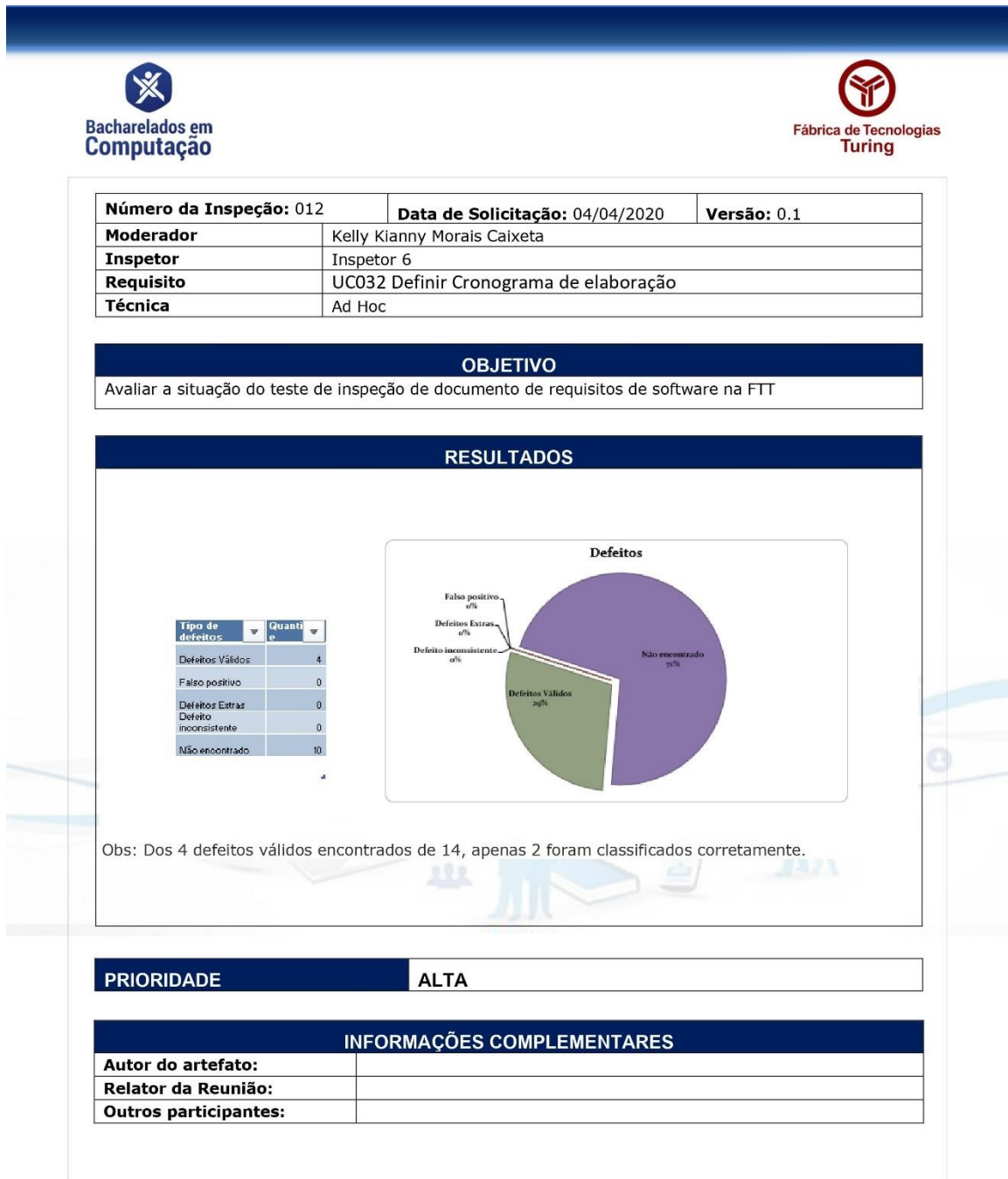
Obs: Dos 8 defeitos válidos encontrados de 20, apenas 4 foram classificados corretamente.

PRIORIDADE ALTA

INFORMAÇÕES COMPLEMENTARES

Autor do artefato:	
Relator da Reunião:	
Outros participantes:	

Figura 33- Relatório de inspeção pag.13



APÊNDICE B – Documentos de requisitos de software para inspeção

UC032 Definir Cronograma de elaboração

Histórico de Versão

Versão	Data	Responsável	Descrição
0.1	10/09/2019	Charley JJ	Início da documentação
0.2	18/11/2019	Rafael Martins de Lima	Adição dos campos de "Turno" e "Período Letivo".
0.3	19/03/2020	Marcella Canedo Tristão	Refatoração do requisito e remoção de campos desnecessários

Personas

N/A

Pré-Condições

A persona deverá estar cadastrado na base de dados.

A persona deverá realizar login para acessar as funcionalidades do caso de uso no sistema.

A persona deverá possuir as funcionalidades deste caso de uso atribuídas ao seu perfil de acesso.

Como Acessar

Acessar Menu Principal -> Planos de Ensino -> [UC032] - Cronograma de Elaboração

História de Usuário 1

COMO persona **POSSO** definir prazos **PARA** liberar o acesso aos planos de ensino.

Cenários

[CG-007] - Filtrar

[CG-002] - Listar

[CG-0012] - Cadastrar em Modal

[CG-0013] - Alterar em Modal

[CG-004] - Excluir/Remover

[CG-008] - Pesquisar sem botão

Mockups

[M1] - Filtro / Listagem

The screenshot shows the EPSTUDY logo interface. On the left is a dark sidebar menu with the following items: Home, Instituições de Ensino, Informações Acadêmicas, Calendários, Plano de Ensino, Cronograma, Gerenciar Planos de Ensino, and Meus Planos de Ensino. At the bottom of the sidebar are logos for Engenharia de Software, Engenharia de Computação, and Fábrica de Tecnologias Turing. The main content area is titled 'Cronograma de Elaboração de Planos de Ensino' and features a search bar with the placeholder 'Faça uma pesquisa!'. Below the search bar is a dropdown menu for 'Curso' with the text 'Selecione uma opção'. There is a '+ Cronograma' button, a 'Pesquisar...' button, and a 'Mostrando 3 itens' dropdown. A table displays three records:



Curso	Início	Fim	Ações
Engenharia de Computação	01/06/2019	03/07/2019	[Link] [Excluir]
Engenharia de Software	01/06/2019	04/07/2019	[Link] [Excluir]
Engenharia Civil	01/06/2019	01/07/2019	[Link] [Excluir]

At the bottom of the table, it says 'Mostrando 3 de 3 registros' and navigation arrows for '<< Anterior' and 'Próximo >>'.

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Curso	Input Text	Esse campo possuirá as seguintes opções: - Cursos cadastrados nas bases de dados.	Sim	Não	- Ao selecionar o curso a tabela é filtrada
Pesquisar	Input Text	Alfanumérico(Ilimitado)	Sim	Não	- O sistema realiza a pesquisa à medida que a persona preenche o campo. - A busca será feita por todos os campos que são exibidos na listagem.

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Filtrar	Realiza o filtro dos cursos de acordo com o selecionado no select.	Button	Não	N/A
+Cronograma	Irà redirecionar a persona para o modal de cadastro de novos cronogramas	Button	Não	N/A
[] Alterar	Irà redirecionar a persona para o modal de Alteração do respectivo cronograma	Button	Sim	N/A
[] Excluir	Excluirá o respectivo cronograma na base de dados.	Button	Sim	[MSG006.1]

CRITÉRIOS DE ACEITAÇÃO

Na coluna de cursos, serão mostrados todos os cursos que foram cadastrados juntos com o prazo de início e fim do cronograma.

[M2] - Cadastrar Cronograma

The screenshot displays the 'Cadastro de Cronograma' (Course Registration) form within the EPSTUDY system. The form is titled 'Cadastro de Cronograma' and includes the following fields:

- Curso***: A dropdown menu with the placeholder text 'Selecione uma opção'.
- Período Letivo***: A dropdown menu with the placeholder text 'Selecione uma opção'.
- Data de Início***: A date picker field with the placeholder text 'dd/mm/aaaa'.
- Data de Fim***: A date picker field with the placeholder text 'dd/mm/aaaa'.

At the bottom of the form, there are two buttons: 'Salvar' (Save) and 'Cancelar' (Cancel). A red asterisk indicates that all fields are mandatory. The background shows a table of courses with their respective start and end dates:

Curso	Data de Início	Data de Fim
Engenharia de Produção	01/06/2019	04/07/2019
Enfermagem	01/06/2019	01/07/2019

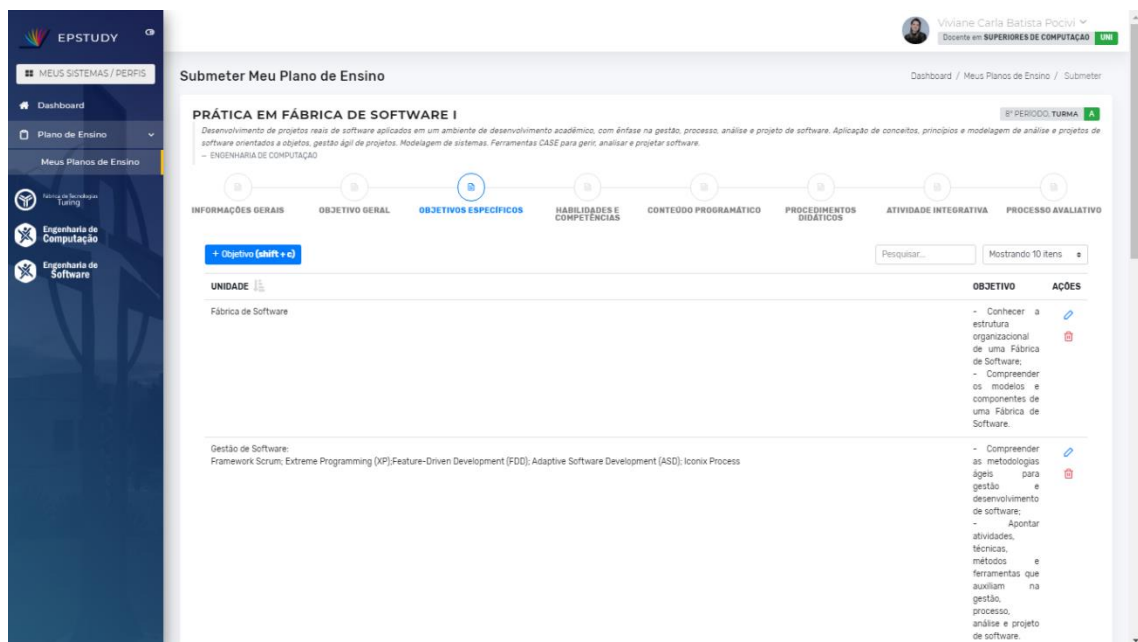
DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Curso	Select	Esse campo possui as seguintes opções: - Cursos cadastrados nas bases de dados.	Sim	Sim	-[MSG002].
Período Letivo	Select	Esse campo possui as seguintes opções: - Periodos Letivos cadastrados no requisito de "Períodos Letivos".	Sim	Sim	- [MSG002].
Data de Início	Date Picker	Date(10)	Sim	Sim	- Por padrão esse campo terá o limite de 100 (para mais e menos). - Este campo deve possuir a máscara de data: dd/mm/aaaa - [MSG003]
Data de Fim	Date Picker	Date(10)	Sim	Sim	-Por padrão esse campo terá o limite de 100 (para mais e menos)

					<p>- Este campo deverá ficar bloqueado até que seja selecionado a data de início.</p> <p>- Este campo não pode ter a data inferior a data de início.</p> <p>- Este campo deve possuir a máscara de data: dd/mm/aaaa</p> <p>-[MSG002].</p>
--	--	--	--	--	---

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar	Salva as informações na base de dados e retornar para a página de listagem	Button	Não	Caso já exista um cronograma criado com o curso selecionado, deve-se mostrar a -[MSG003]
Cancelar	Cancelar a ação e retornar para a página de listagem	Button	Não	- [MSG006.5]
 [X] Fechar Modal	Cancelar a ação e retornar para a página de listagem	Button	Não	- [MSG006.5]

[M3] - Wizard Cadastro Plano de Ensino – Objetivo Específico



Submeter Meu Plano de Ensino

PRÁTICA EM FÁBRICA DE SOFTWARE I





Desenvolvimento de projetos reais de software aplicados em um ambiente de desenvolvimento acadêmico, com ênfase na gestão, processo, análise e projeto de software. Aplicação de conceitos, princípios e modelagem de análise e projetos de software orientados a objetos, gestão ágil de projetos, Modelagem de sistemas. Ferramentas CASE para gerar, analisar e projetar software.

– ENGENHARIA DE COMPUTAÇÃO

INFORMAÇÕES GERAIS OBJETIVO GERAL **OBJETIVOS ESPECÍFICOS** HABILIDADES E COMPETÊNCIAS CONTEÚDO PROGRAMÁTICO PROCEDIMENTOS DIDÁTICOS ATIVIDADE INTEGRATIVA PROCESSO AVALIATIVO



+ Objetivo (Shift + C)

UNIDADE: Fábrcia de Software

UNIDADE	OBJETIVO	AÇÕES
Fábrcia de Software	- Conhecer a estrutura organizacional de uma Fábrcia de Software. - Compreender os modelos e componentes de uma Fábrcia de Software.	 
Gestão de Software: Framework Scrum; Extreme Programming (XP); Feature-Driven Development (FDD); Adaptive Software Development (ASD); Iconix Process	- Compreender as metodologias ágeis para gestão e desenvolvimento de software; - Aportar atividades, técnicas, métodos e ferramentas que auxiliam na gestão, processo, análise e projeto de software.	 

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Unidade	Input Text	Alfanumérico(1000)	Sim	Sim	N/A
Objetivos	Input Text	Alfanumérico(2500)	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
[+ Objetivo (Shift + C)]	Cadastra um novo objetivo específico	Button	Não	N/A
[ Alterar]	Habilita a opção de editar o objetivo específico cadastrado	Button	Sim	N/A
[ Excluir]	Exclui o objetivo específico cadastrado	Button	Sim	- MSG – 006.1

UC019 Elaborar Plano de Ensino

Histórico de Versão

Versão	Data	Responsável	Descrição
0.1	11/10/2019	Charley JJ	Criação do Documento
0.2	17/02/2020	Lucas Henrique	Atualização da documentação

Personas

Administrador do Sistema
Acadêmico
Docente e Coordenadora Pedagógica

Pré-Condições

A persona deverá estar cadastrado na base de dados.
A persona deverá realizar login para acessar as funcionalidades do caso de uso no sistema.
A persona deverá possuir as funcionalidades deste caso de uso atribuídas ao seu perfil de acesso.

Como Acessar

Acessar Menu Principal -> Plano de Ensino -> [UC019] - Elaborar Plano de Ensino.

Regras de Negócio

[RN001] - Após o cronograma de elaboração de planos de ensino chegar na data de fim todos os planos de ensino que não possuírem o status de "Cadastrado" serão enviados para a validação.

Épico

COMO pessoa **POSSO** elaborar os planos de ensino **PARA** gerencia-los.

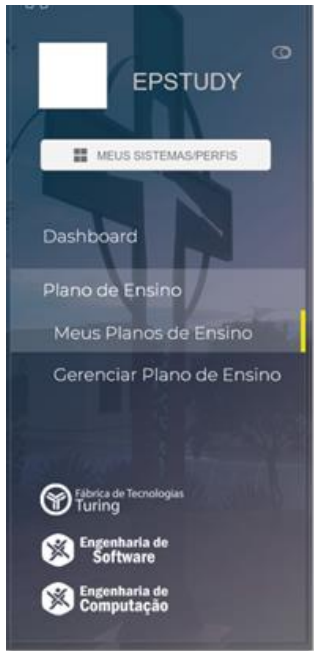
História de Usuário 1

COMO persona **POSSO** cadastrar planos de ensino **PARA** gerar os planos de ensino

Cenários


[CG-001] - Inativar
[CG-002] - Ativar
[CG-003] – Alterar
[CG-004] Excluir/Remover

[M1] - Wizard de Cadastro do Plano de Ensino - Informações Gerais



Submeter Meu Plano de Ensino

Dashboard / Planos de Ensino / Meus planos de ensino

 Nome Usuário ▾
 Disciplina em ENGENHARIA DA COMPUTAÇÃO

PROGRAMAÇÃO ORIENTADA A OBJETOS

ENGENHARIA DE SOFTWARE

1º PERÍODO, TURMA A

INFORMAÇÕES GERAIS

OBJETIVO GERAL

OBJETIVO ESPECÍFICO

HABILIDADES E COMPETÊNCIAS

CONTEÚDO PROGRAMÁTICO

PROCEDIMENTOS DIDÁTICOS

ATIVIDADE INTEGRATIVA

PROCESSO AVALIATIVO

PRÁTICA EM FÁBRICA DE SOFTWARE I

Desenvolvimento de projetos reais de software aplicados em um ambiente de desenvolvimento acadêmico, com ênfase na gestão, processo, análise e projeto de software. Aplicação de conceitos, princípios e modelagem de análise e projeto de software orientados a objetos, gestão Ágil de projetos, Modelagem de sistemas, Ferramentas CASE para gerar, analisar e projetar software. – Engenharia de Computação

INFORMAÇÕES GERAIS

OBJETIVO GERAL

OBJETIVOS ESPECÍFICOS

HABILIDADES E COMPETÊNCIAS

CONTEÚDO PROGRAMÁTICO

PROCEDIMENTOS DIDÁTICOS


ATIVIDADE INTEGRATIVA

PROCESSO AVALIATIVO

CARACTERIZAÇÃO DA DISCIPLINA

Curso: Engenharia de Computação	Turma: Retorno
Matriz: 2016.1	Da 03 semana: Segunda-feira
Nome da disciplina: Prática em Fábrica de Software I	Ano/Semestre: 2020/1
Código da disciplina: 05849	Período: 1º Turma: A
Carga horária total: 80h/a	Carga horária prática: 40h/a Carga horária teórica: 40h/a
Pré-Requisito: Programação Orientada a Objetos	Co-Requisito: Prática em Tópicos de Software I

PROFESSOR(ES)

 Viviane Carla Batista Pooivi
<https://www.linkedin.com/in/vivianepooivi>
<http://lattes.cnpq.br/3730689178330202>

EMENTA

Desenvolvimento de projetos reais de software aplicados em um ambiente de desenvolvimento acadêmico, com ênfase na gestão, processo, análise e projeto de software. Aplicação de conceitos, princípios e modelagem de análise e projeto de software orientados a objetos, gestão Ágil de projetos, Modelagem de sistemas, Ferramentas CASE para gerar, analisar e projetar software.

BIBLIOGRAFIAS

BIBLIOGRAFIA BÁSICA	BIBLIOGRAFIA COMPLEMENTAR
SCHACH, Stephen R. <i>Engenharia de Software: os paradigmas clássico e orientado a objetos</i> . 7 ed. São Paulo: McGraw-Hill, 2008. 616 p.	
KOSIOJANSKI, André; SGARES, Michel dos Santos. <i>Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software</i> . 2. ed. São Paulo, SP: Brasili Novatec, 2007. 395 p.	
VALERIANO, Dalton L. <i>Gerência em Projetos: pesquisa, desenvolvimento e engenharia</i> . São Paulo: Pearson Education, 2004.	

> Próximo
Cancelar

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Próximo	Volta para a tela anterior.	Button	Não	N/A
Cancelar	Cancela a ação	Button	Não	N/A

CRITÉRIOS DE ACEITAÇÃO


Os campos deverão ser validados.

[M2] - Wizard Cadastro Plano de Ensino – Objetivo Geral

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Objetivo Geral	Plugin de Texto	Alfa numérico(1000)	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar e continuar	Salva e avança para a próxima wizard	select	Não	N/A
Anterior	Volta para a wizard anterior e mantém os dados preenchidas na mesma	Button	Não	N/A
	O avaliador poderá adicionar comentários ao plano de ensino	Button	Não	N/A
Próximo	Avança para próxima Wizard	Button	Não	N/A

[M3] - Wizard Cadastro Plano de Ensino – Objetivo Específico

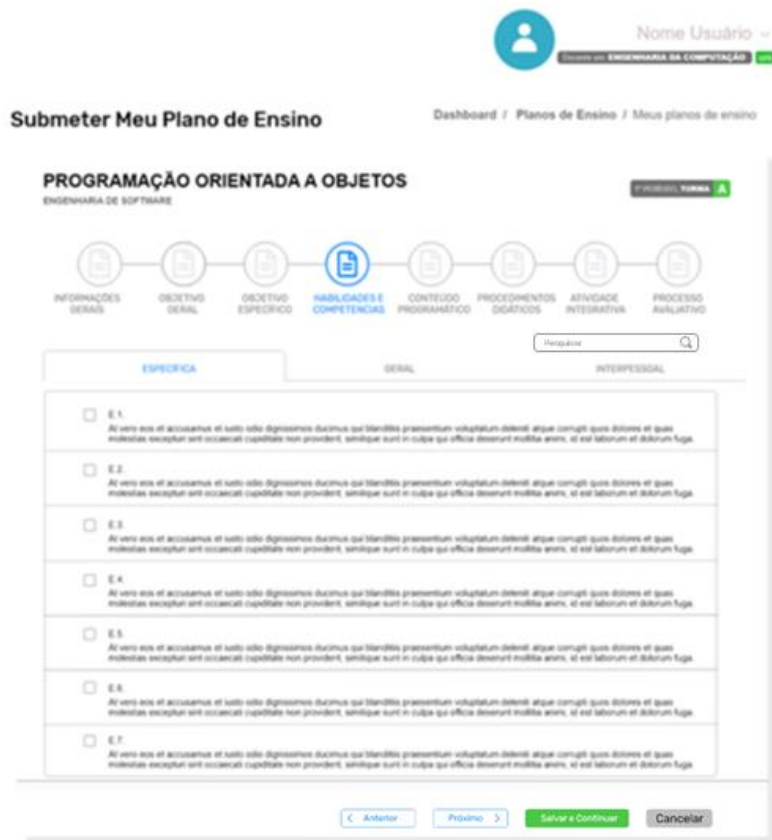
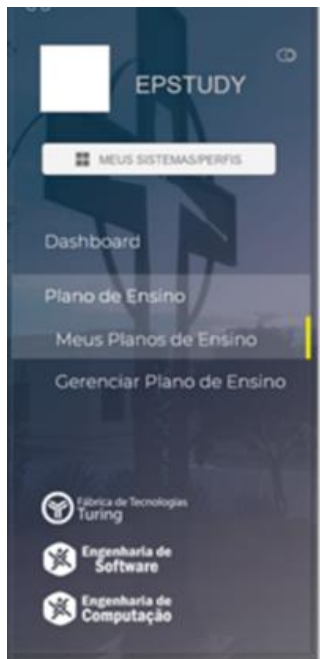
DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Unidade	Input Text	Alfanumérico(1000)	Sim	Sim	N/A
Objetivos	Input Text	Alfanumérico(2500)	Sim	Sim	N/A


DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
[+ Objetivo (Shift + C)]	Caduta um novo objetivo específico	Button	Não	N/A
[Alterar]	Habilita a opção de editar o objetivo específico cadastrado	Button	Sim	N/A
[Excluir]	Exclui o objetivo específico cadastrado	Button	Sim	- MSG – 006.1

[M4] - Wizard Cadastro Plano de Ensino - Habilidades e Competências

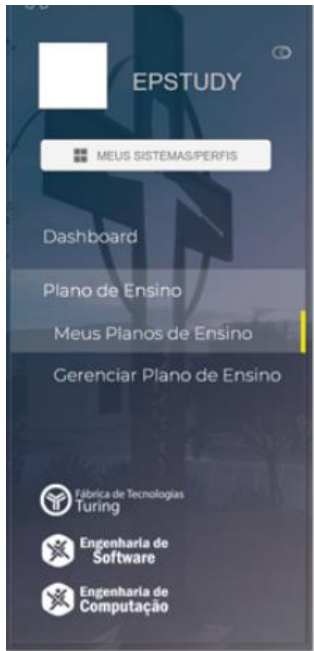


DESCRIÇÃO DE CAMPOS					
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Habilidade e Competências	Checkbox	Selecionar qual habilidade competência o usuário deseja usar.	Sim	Não	N/A

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
	Poderei marcar ou desmarcar a habilidade e competências cadastrada.	Input	Não	- Ser habilitado apenas quando clicar no botão editar.

CRITÉRIOS DE ACEITAÇÃO
O Wizard deverá ser capaz de manter as informações ao prosseguir para as próximas etapas.
Ao clicar em "Voltar", o Wizard deverá manter as informações preenchidas na etapa atual além de apresentar as informações das etapas anteriores de forma integral e sem alterações.

[M5] - Wizard Cadastro Plano de Ensino - Conteúdo Programático



Nome Usuário ▾
Iniciado em ENGENHARIA DE COMPUTAÇÃO

Submeter Meu Plano de Ensino Dashboard / Planos de Ensino / Meus planos de ensino

PROGRAMAÇÃO ORIENTADA A OBJETOS
ENGENHARIA DE SOFTWARE 1º PERÍODO, TURMA A

INFORMAÇÕES GERAIS OBJETIVO GERAL OBJETIVO ESPECÍFICO HABILIDADES E COMPETÊNCIAS **CONTEÚDO PROGRAMÁTICO** PROCEDIMENTOS DIDÁTICOS ATIVIDADE INTEGRATIVA PROCESSO AVALIATIVO

Pesquisar em: Mostrar 20 itens 2

ENCONTRO	ESTRATÉGIA DE ENSINO	AÇÕES
1º ENCONTRO - 02/02/2020 Contrary to popular belief, Lorem Ipsum is not simply random text, Lorem Ipsum is not simply	Técnicas Sala de Aula	Apresentação por grupo Atividade Avaliativa
2º ENCONTRO - 09/02/2020 Contrary to popular belief, Lorem Ipsum is not simply random text, Lorem Ipsum is not simply	Prática Assíncrono	Apresentação por grupo Atividade Avaliativa
3º ENCONTRO - 16/02/2020 Contrary to popular belief, Lorem Ipsum is not simply random text, Lorem Ipsum is not simply	Técnicas & Teóricas Laboratório	Apresentação por grupo Atividade Avaliativa
4º ENCONTRO - 23/02/2020 Nenhuma descrição informada!		
5º ENCONTRO - 01/03/2020 Nenhuma descrição informada!		
6º ENCONTRO - 08/03/2020 Nenhuma descrição informada!		
7º ENCONTRO - 15/03/2020 Nenhuma descrição informada!		
8º ENCONTRO - 22/03/2020 Nenhuma descrição informada!		
9º ENCONTRO - 29/03/2020 Nenhuma descrição informada!		
10º ENCONTRO - 05/04/2020 Nenhuma descrição informada!		
11º ENCONTRO - 12/04/2020 Nenhuma descrição informada!		
12º ENCONTRO - 19/04/2020 Nenhuma descrição informada!		
13º ENCONTRO - 26/04/2020 Nenhuma descrição informada!		
14º ENCONTRO - 03/05/2020 Nenhuma descrição informada!		
15º ENCONTRO - 10/05/2020 Nenhuma descrição informada!		
16º ENCONTRO - 17/05/2020 Nenhuma descrição informada!		
17º ENCONTRO - 24/05/2020 Nenhuma descrição informada!		
18º ENCONTRO - 31/05/2020 Nenhuma descrição informada!		
19º ENCONTRO - 07/06/2020 Nenhuma descrição informada!		
19º ENCONTRO - 13/06/2020 Nenhuma descrição informada!		
20º ENCONTRO - 19/06/2020 Nenhuma descrição informada!		

* Campo obrigatório < Anterior Próximo > Salvar e Continuar Cancelar

DESCRIÇÃO DE CAMPOS					
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Encontro	Smart Table (Int)	Alfanumérico (120)	Não	Sim	- [MSG002] - Já virá pré determinado de 1 à 20 , seguindo a carga horária de aulas da disciplina

Estratégia Ensino-aprendizado	Smart Table (Input Text)	Alfanumérico(120)	Sim	Sim	- [MSG002]
-------------------------------	--------------------------	-------------------	-----	-----	------------

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

[M6] Wizard Cadastro Plano de Ensino - Procedimentos Didáticos

Submeter Meu Plano de Ensino

PRÁTICA EM FÁBRICA DE SOFTWARE I

Desenvolvimento de projetos reais de software aplicados em um ambiente de desenvolvimento acadêmico, com ênfase na gestão, processo, análise e projeto de software. Aplicação de conceitos, princípios e modelagem de análise e projeto de software orientados a objetivos, gerada a partir de projetos. Modelagem de sistemas. Ferramentas CASE para gerar, analisar e projetar software.

ENDINHEIRA DE COMPUTAÇÃO

INFORMAÇÕES GERAIS OBJETIVO GERAL OBJETIVOS ESPECÍFICOS HABILIDADES E COMPETÊNCIAS CONTEÚDO PROGRAMÁTICO **PROCEDIMENTOS DIDÁTICOS** ATIVIDADE INTEGRATIVA PROCESSO AVALIATIVO

PROCEDIMENTOS DIDÁTICOS*

As metodologias empregadas e os recursos educacionais adotados na presente disciplina incluem:

- Exposição verbal e dialogada dos conteúdos técnicos, com o apoio de recursos multimídia;
- Estudo de texto em que os estudantes deverão analisar criticamente artigos ou textos científicos que abordem os conceitos técnicos sobre análise, projeto, gestão e processo de software;
- Peer instruction (mencão aos Pares) e Sala de Aula Invertida para estudo dos conteúdos sobre as metodologias Ágile e conceitos de formação geral, com resolução comentada de questões usando a ferramenta Socrative. As questões de formação geral são as três últimas aulas do ENADE serão disponibilizadas, previamente, no Locus;
- Painel, discussão informal dos estudantes, apresentação e vídeo geral, conceitos, princípios e técnicas de metodologia e processos ágeis usando mapa mental elaborado pelos acadêmicos;
- Seminários sobre a UAC, 2.5 com aplicação em um projeto real;
- Retomada de conteúdos;
- Desenvolvimento de um software real (projeto real) envolvendo problemas com os quais os alunos possam utilizar técnicas para gerar, analisar e projetar uma solução, associando-os teoria e prática trabalhadas nas aulas;
- Desenvolvimento de portfólio, que consistirá na construção, análise e reflexão dos artefatos produzidos sobre a gestão, a análise e o projeto de software do projeto real.

* Campo obrigatório

< Anterior Próximo > Salvar e Continuar Cancelar

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Procedimentos Didáticos	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

[M7] Wizard Cadastro Plano de Ensino – Atividade Integrativa

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Atividade integrativa	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]

[M8] Wizard Cadastro Plano de Ensino - Processos Avaliativos

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Atividade integrativa	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]

1° Avaliação de Aprendizagem	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]
2° Avaliação de Aprendizagem	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]
3° Avaliação de Aprendizagem	Plugin de Texto	Alfanumérico(2500)	Sim	Sim	- [MSG002]

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
	Envia e salva o plano de ensino para validação	Button	Não	N/A
	Salva os dados preenchidos no plano de ensino.	Button	Não	- Os dados não serão perdidos, semelhante ao template do Apex.

CRITÉRIOS DE ACEITAÇÃO
Os campos deverão ser validados.

História Usuário 2

COMO persona **POSSO** gerenciar conteúdo programático **PARA** gerenciar estratégias

Cenários

[CG-001] - Cadastrar

[CG-002] - Listar

[CG-003] - Alterar

[CG-004] – Excluir/Remover

[M1] - Alterar Conteúdo Programático

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Estratégias de Ensino	Multi select	Neste campo poderá selecionar as estratégias cadastradas em: Gerenciar estratégias.	Sim	Sim	N/A
Tipo de aula	Select	Este campo possui as seguintes opções: Teórica, Prática, Prática e Teórica	Sim	Sim	N/A
Local	Multi select	Neste campo poderá selecionar os locais cadastrados em: Gerenciar Locais	Sim	Sim	N/A
Conteúdo	Plugin de Texto	Este campo possui as informações sobre o conteúdo programático	Sim	Sim	N/A

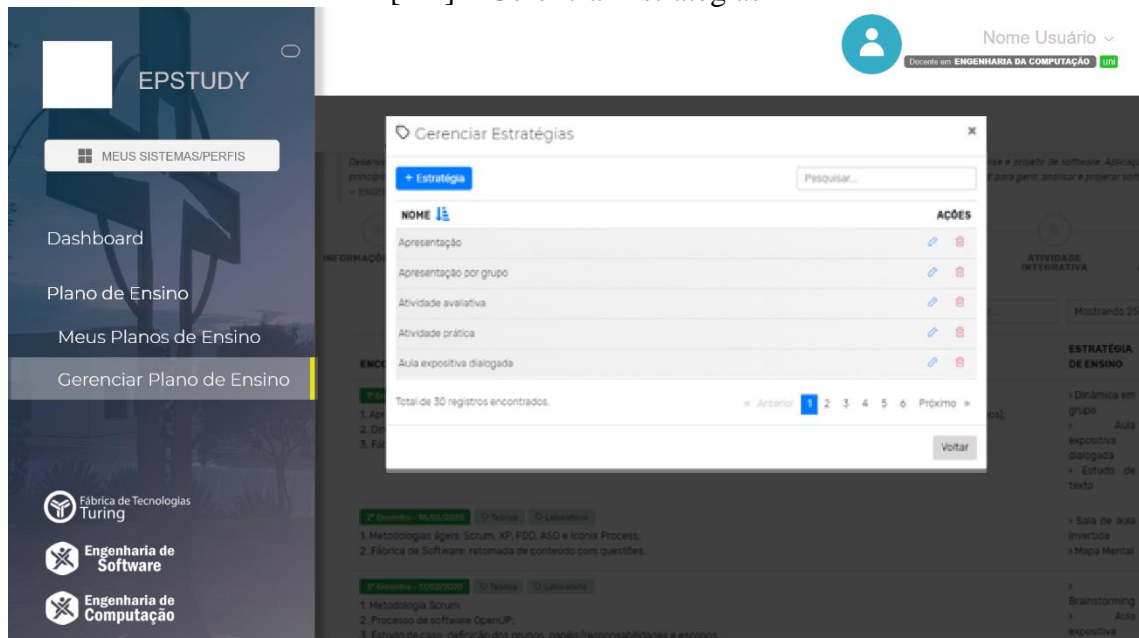
DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar	Salva os dados alterados	Button	Não	N/A
Cancelar	Cancela a ação de criar um plano de ensino retornando a persona para a tela de listagem.	Button	Não	- [MSG006.5]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

[M2] – Gerenciar Estratégias





DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Lista	Alfanumérico(200)	Sim	Sim	N/A
Pesquisar	Input Text	Alfanumérico(100)	Sim	Não	N/A

DESCRIÇÃO DOS COMANDOS

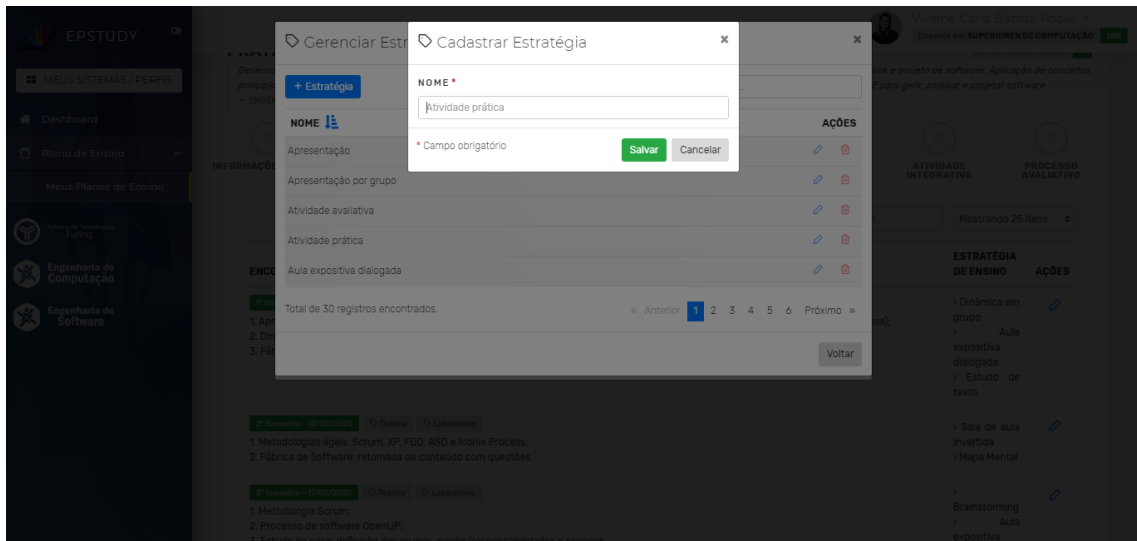
Nome do comando	Ação	Tipo	Tooltip	Restrições
Próximo	Avança para a próxima página onde irá conter mais estratégias	Button	Não	Apenas será possível ir para o próximo se existir mais informações a serem exibidas
Anterior	Volta para a página anterior de listagens de estratégias	Button	Não	Apenas será possível clicar em pagina anterior se o mesmo não estiver na primeira página.
Voltar	Volta pra tela inicial da wizard – Conteúdo programático	Button	Não	N/A
+ Estratégia	Cadastra uma nova estratégia	Button	Não	N/A

 Alterar	Altera uma estratégia já cadastrada	Button	Sim	- Estará inativo para estratégias pré-cadastradas.
 Excluir	Exclui uma estratégia já cadastrada	Button	Sim	- Estará inativo para estratégias pré-cadastradas.
Voltar	Redireciona o usuário para a tela de cadastro dos "Objetivos" do plano de ensino	Button	Não	- Os dados ao voltar não serão perdidos.

CRITÉRIOS DE ACEITAÇÃO

Não é permitido a exclusão ou alteração de estratégias pré-cadastradas no sistema

[M3] – Cadastrar Estratégias



DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Input Text	Alfanumérico(200)	Sim	Sim	N/A

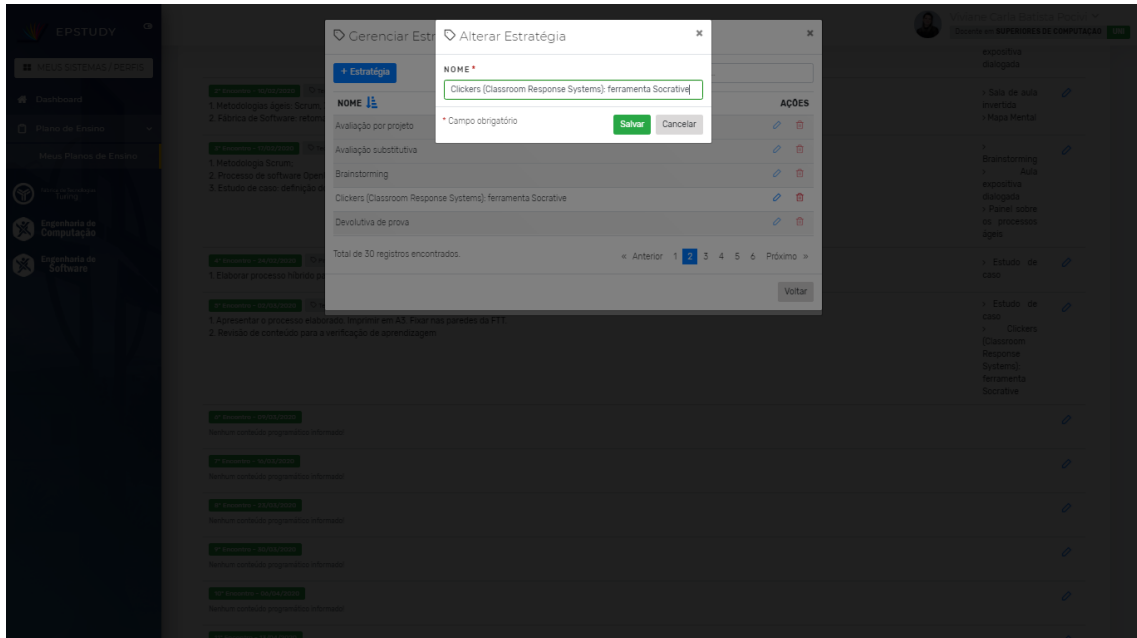
DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
	Salva a nova estratégia cadastrada	Button	Não	- [MSG005]
	Cancela a nova estratégia que seria cadastrada	Button	Não	- [MSG036]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

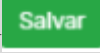
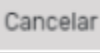
[M4] – Alterar Estratégias



DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Input Text	Alfanumerico (200)	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
	Salva a alteração realizada na estratégia	Button	Não	- [MSG005]
	Cancela a alteração realizada na estratégia	Button	Não	- [MSG036]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

História de Usuário 3

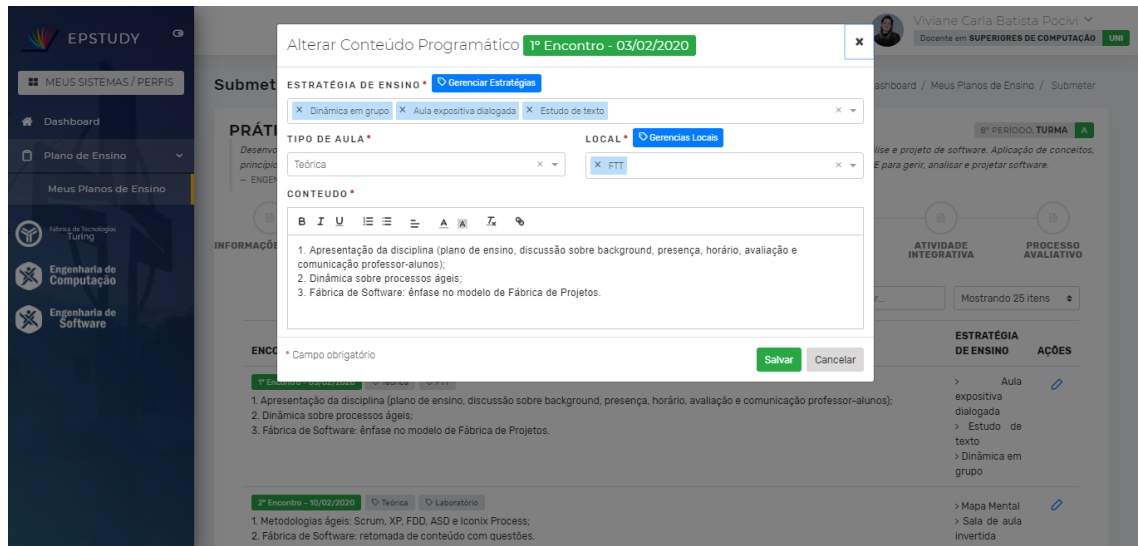
COMO persona **POSSO** gerenciar conteúdo programático **PARA** gerenciar estratégias

Cenários

[CG-001] - Cadastrar

[CG-002] - Listar
 [CG-003] - Alterar
 [CG-004] – Excluir/Remover

[M1] - Alterar Conteúdo Programático



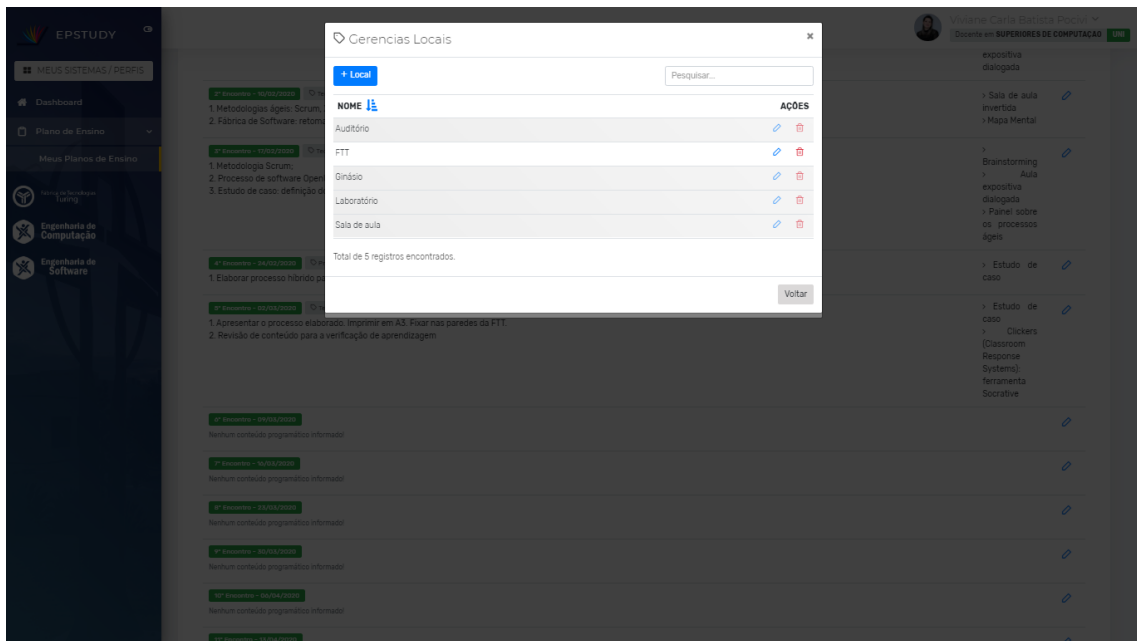
DESCRIÇÃO DE CAMPOS					
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Estratégias de Ensino	Multi select	Neste campo poderá selecionar as estratégias cadastradas em: Gerenciar estratégias.	Sim	Sim	N/A
Tipo de aula	Select	Este campo possui as seguintes opções: Teórica, Prática, Prática e Teórica	Sim	Sim	N/A
Local	Multi select	Neste campo poderá selecionar os locais cadastrados em: Gerenciar Locais	Sim	Sim	N/A
Conteúdo	Plugin de Texto	Este campo possui as informações sobre o conteúdo programático	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar	Salva os dados alterados	Button	Não	N/A
Voltar	Volta para a tela anterior.	Button	Não	- [MSG006.5]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

[M2] – Gerenciar Locais



DESCRIÇÃO DE CAMPOS

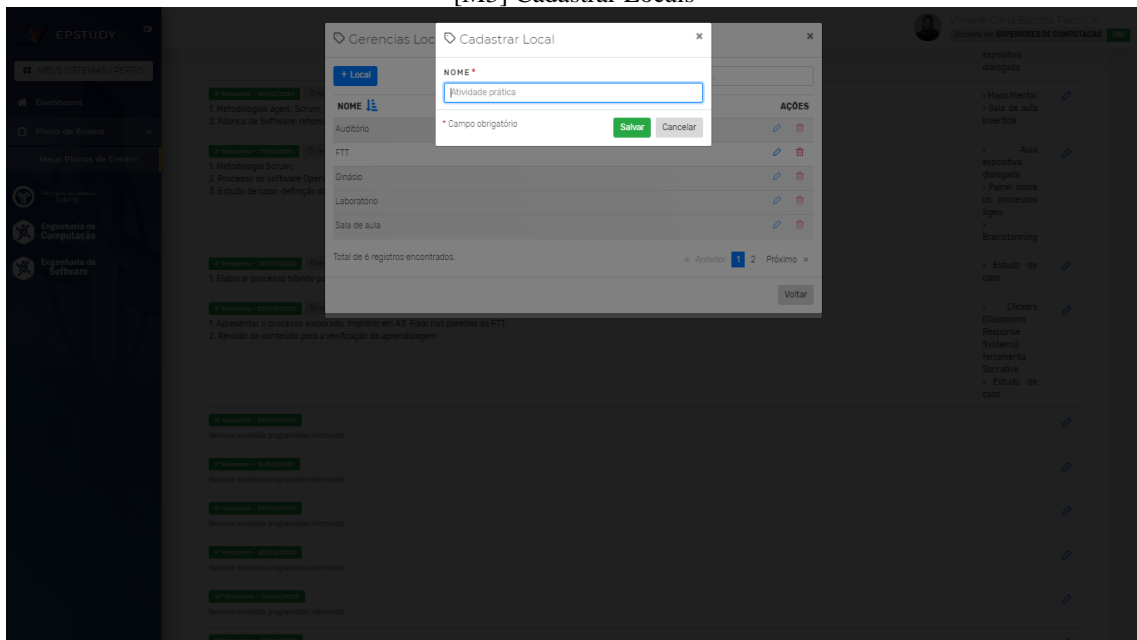
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Lista	Alfanumérico(200)	Sim	Sim	N/A
Pesquisar	Input Text	Alfanumérico(100)	Sim	Não	N/A

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Próximo	Avança para a próxima página onde irá conter mais locais cadastrados	Button	Não	Apenas será possível ir para o próximo se existir mais informações a serem exibidas
Anterior	Volta para a página anterior de listagens de locais	Button	Não	Apenas será possível clicar em pagina anterior se o mesmo não estiver na primeira página.

Voltar	Volta pra tela inicial da wizard – Conteúdo programático	Button	Não	N/A
[+ Local]	Cadastra um novo local	Button	Não	N/A
[] Alterar	Altera um local já cadastrado	Button	Sim	- Estará inativo para locais pré-cadastrados.
[] Excluir	Exclui um local já cadastrado	Button	Sim	- Estará inativo para locais pré-cadastrados.
Cancelar	Redireciona o usuário para a tela de cadastro dos "Objetivos" do plano de ensino	Button	Não	- Os dados ao voltar não serão perdidos.

[M3] Cadastrar Locais



DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Input Text	Alfanumérico(200)	Sim	Sim	N/A

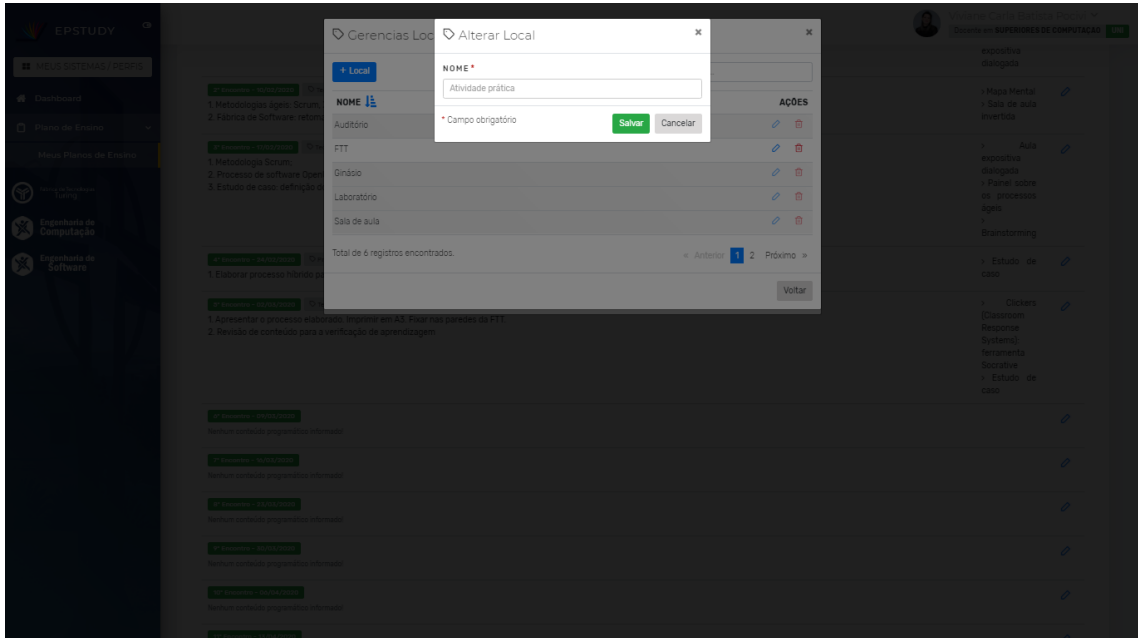
DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
[Salvar]	Salva o novo local cadastrada	Button	Não	- [MSG005]
[Cancelar]	Cancela o novo local que seria cadastrado	Button	Não	- [MSG036]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.


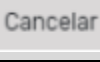
[M3] Alterar Locais



DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Nome	Input Text	Alfanumerico (200)	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
	Salva a alteração realizada no local	Button	Não	- [MSG005]
	Cancela a alteração realizada no local	Button	Não	- [MSG036]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

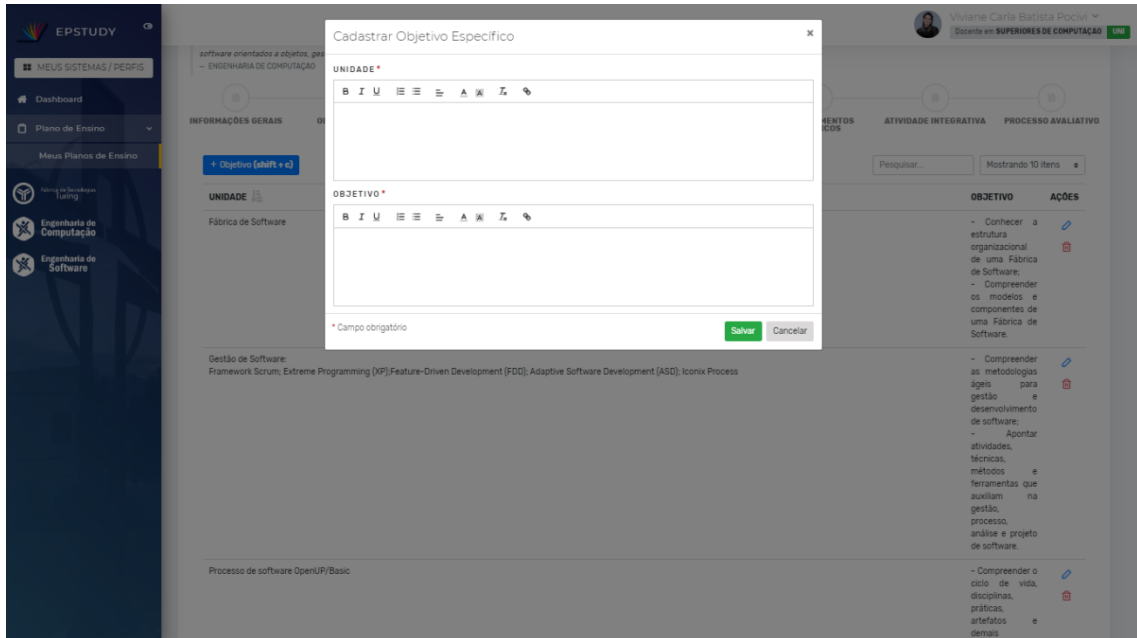
História Usuário 4

COMO persona POSSO cadastrar planos de ensino PARA gerar os planos de ensino

Cenários

[CG-001] - Cadastrar
 [CG-002] - Listar
 [CG-005] – Alterar

[M1] Cadastrar Objetivo Específico



DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Unidade	Plugin de Texto	Alfanumérico (5000)	Sim	Sim	N/A
Objetivo	Plugin de Texto	Alfanumérico (5000)	Sim	Sim	N/A

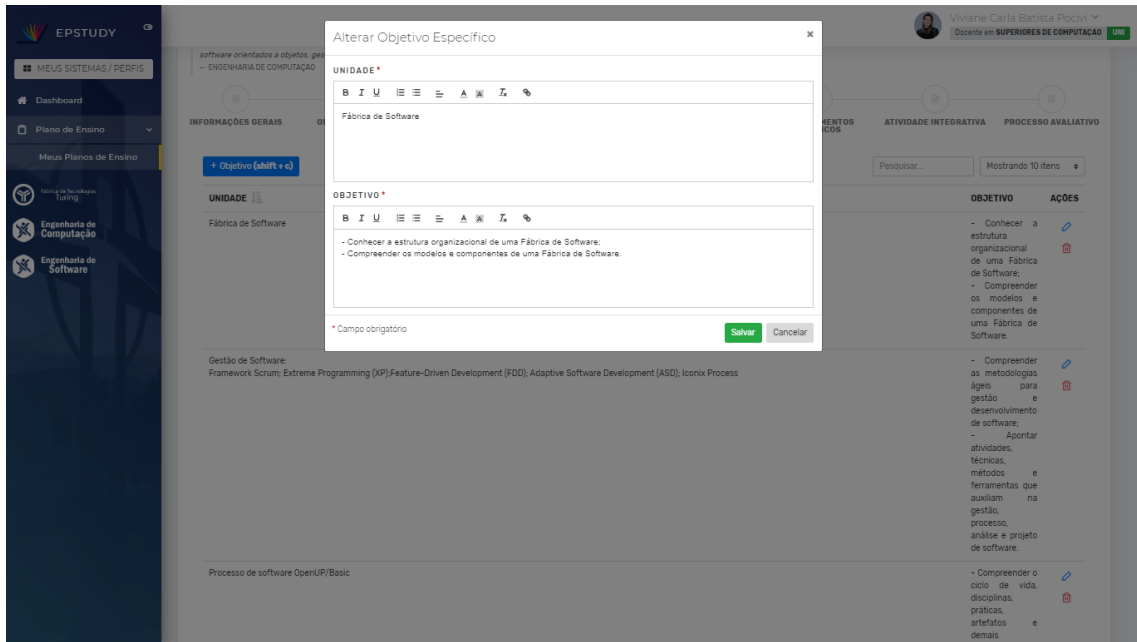
DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar	Salva a novo objetivo específico cadastrado	Button	Não	- [MSG005]
Cancelar	Cancela o novo objetivo específico cadastrado	Button	Não	- [MSG036]

CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

[M2] – Alterar Objetivo Especifica



CRITÉRIOS DE ACEITAÇÃO

Os campos deverão ser validados.

UC032 Definir Cronograma de elaboração

Histórico de Versão

Versão	Data	Responsável	Descrição
0.1	10/09/2019	Charley JJ	Início da documentação
0.2	18/11/2019	Rafael Martins de Lima	Adição dos campos de "Turno" e "Período Letivo".
0.3	19/03/2020	Marcella Canedo Tristão	Refatoração do requisito e remoção de campos desnecessários

Personas

N/A

Pré-Condições

A persona deverá estar cadastrado na base de dados.

A persona deverá realizar login para acessar as funcionalidades do caso de uso no sistema.

A persona deverá possuir as funcionalidades deste caso de uso atribuídas ao seu perfil de acesso.

Como Acessar

Acessar Menu Principal -> Planos de Ensino -> [UC032] - Cronograma de Elaboração

História de Usuário 1

COMO persona **POSSO** definir prazos **PARA** liberar o acesso aos planos de ensino.

Cenários

- [CG-007] - Filtrar
- [CG-002] - Listar
- [CG-0012] - Cadastrar em Modal
- [CG-0013] - Alterar em Modal
- [CG-004] - Excluir/Remover
- [CG-008] - Pesquisar sem botão

Mockups

[M1] - Filtro / Listagem

The screenshot shows the EPSTUDY application interface. On the left is a dark sidebar with the EPSTUDY logo and navigation menu items: Home, Instituições de Ensino, Informações Acadêmicas, Calendários, Plano de Ensino, Cronograma, Gerenciar Planos de Ensino, and Meus Planos de Ensino. At the bottom of the sidebar are logos for Engenharia de Software, Engenharia de Computação, and Fábrica de Tecnologias Turing. The main content area is titled 'Cronograma de Elaboração de Planos de Ensino' and features a search bar with the placeholder 'Faça uma pesquisa!'. Below the search bar is a dropdown menu for 'Curso' with the text 'Selecione uma opção'. There is a blue '+ Cronograma' button, a search input field with 'Pesquisar...', and a 'Mostrando 3 itens' dropdown. The table below has columns for 'Curso', 'Início', 'Fim', and 'Ações'. It lists three courses: Engenharia de Computação (01/06/2019 to 03/07/2019), Engenharia de Software (01/06/2019 to 04/07/2019), and Engenharia Civil (01/06/2019 to 01/07/2019). At the bottom, it says 'Mostrando 3 de 3 registros' and has navigation arrows for '<< Anterior' and 'Próximo >>'.

DESCRIÇÃO DE CAMPOS

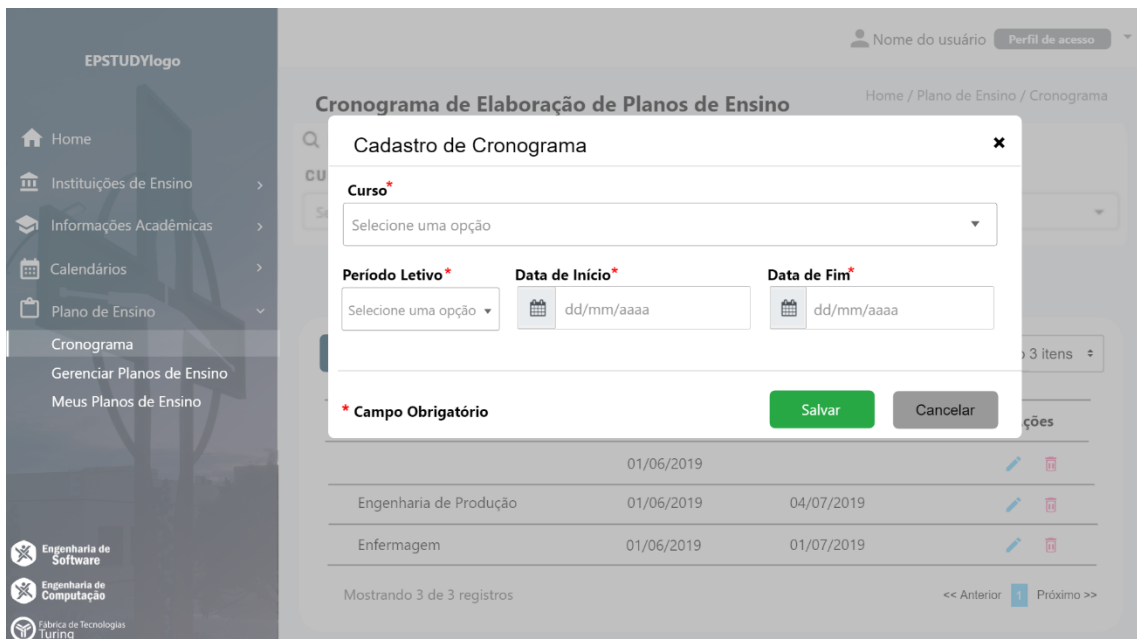
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Curso	Input Text	Esse campo possuirá as seguintes opções: - Cursos cadastrados nas bases de dados.	Sim	Não	- Ao selecionar o curso a tabela é filtrada
Pesquisar	Input Text	Alfanumérico(Ilimitado)	Sim	Não	- O sistema realiza a pesquisa à medida que a persona preenche o campo.

					- A busca será feita por todos os campos que são exibidos na listagem.
--	--	--	--	--	--

DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
Filtrar	Realiza o filtro dos cursos de acordo com o selecionado no select.	Button	Não	N/A
+Cronograma	Irá redirecionar a persona para o modal de cadastro de novos cronogramas	Button	Não	N/A
[] Alterar	Irá redirecionar a persona para o modal de Alteração do respectivo cronograma	Button	Sim	N/A
[] Excluir	Excluirá o respectivo cronograma na base de dados.	Button	Sim	[MSG006.1]

CRITÉRIOS DE ACEITAÇÃO
Na coluna de cursos, serão mostrados todos os cursos que foram cadastrados juntos com o prazo de início e fim do cronograma.

[M2] - Cadastrar Cronograma



The screenshot displays the 'Cadastro de Cronograma' modal in the EPSTUDY application. The modal contains the following fields and controls:

- Curso:** A dropdown menu with the placeholder text 'Selecione uma opção'.
- Período Letivo:** A dropdown menu with the placeholder text 'Selecione uma opção'.
- Data de Início:** A date picker field with the placeholder 'dd/mm/aaaa'.
- Data de Fim:** A date picker field with the placeholder 'dd/mm/aaaa'.
- Buttons:** 'Salvar' (green) and 'Cancelar' (grey).
- Legend:** '* Campo Obrigatório'.

The background shows a table of existing cronograms:

Curso	Data de Início	Data de Fim	Ações
Engenharia de Produção	01/06/2019	04/07/2019	[Alterar] [Excluir]
Enfermagem	01/06/2019	01/07/2019	[Alterar] [Excluir]

DESCRIÇÃO DE CAMPOS					
Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Curso	Select	Esse campo possui as seguintes opções:	Sim	Sim	-[MSG002].

		- Cursos cadastrados nas bases de dados.			
Período Letivo	Select	Esse campo possui as seguintes opções: - Periodos Letivos cadastrados no requisito de "Períodos Letivos".	Sim	Sim	- [MSG002].
Data de Início	Date Picker	Date(10)	Sim	Sim	- Por padrão esse campo terá o limite de 100 (para mais e menos). - Este campo deve possuir a máscara de data: dd/mm/aaaa - [MSG003]
Data de Fim	Date Picker	Date(10)	Sim	Sim	-Por padrão esse campo terá o limite de 100 (para mais e menos) - Este campo deverá ficar bloqueado até que seja selecionado a data de início. - Este campo não pode ter a data inferior a data de início. - Este campo deve possuir a máscara de data: dd/mm/aaaa -[MSG002].



DESCRIÇÃO DOS COMANDOS				
Nome do comando	Ação	Tipo	Tooltip	Restrições
Salvar	Salva as informações na base de dados e retornar para a página de listagem	Button	Não	Caso já exista um cronograma criado com o curso selecionado, deve-se mostrar a -[MSG003]
Cancelar	Cancelar a ação e retornar para a página de listagem	Button	Não	- [MSG006.5]
 [X] Fechar Modal	Cancelar a ação e retornar para a página de listagem	Button	Não	- [MSG006.5]

[M3] - Wizard Cadastro Plano de Ensino – Objetivo Específico

DESCRIÇÃO DE CAMPOS

Nome do campo	Tipo do campo	Tipo de dado(tamanho)	Editável	Obrigatório	Observações
Unidade	Input Text	Alfanumérico(1000)	Sim	Sim	N/A
Objetivos	Input Text	Alfanumérico(2500)	Sim	Sim	N/A

DESCRIÇÃO DOS COMANDOS

Nome do comando	Ação	Tipo	Tooltip	Restrições
[+ Objetivo (Shift + C)]	Cadastra um novo objetivo específico	Button	Não	N/A
[ Alterar]	Habilita a opção de editar o objetivo específico cadastrado	Button	Sim	N/A
[ Excluir]	Exclui o objetivo específico cadastrado	Button	Sim	- MSG – 006.1

APÊNDICE C – Checklist de inspeção

CHECKLIST TESTE DE INSPEÇÃO					
Autor:	Data:				
Funcionalidade:					
Observações:					
Classe – Omissão / Tipo – Funcionalidade Omitida (FO)					
	Sim	Não	Parcial	N/A	
As funções descritas são suficientes para se conhecer os objetivos do requisito?					
Todas as entradas para cada função são suficientes para a execução da função requerida?					
Eventos indesejáveis são considerados e suas respostas são especificadas?					
O estado inicial e os estados especiais (tais como: inicialização do sistema, término anormal, etc.) são considerados?					
Classe – Omissão / Tipo – Performance Omitida (PO)					
	Sim	Não	Parcial	N/A	
O sistema pode ser testado, demonstrado, analisado ou inspecionado para mostrar que satisfaz os requisitos?					
As características relativas à descrição dos itens de dados, como: tipo, taxa de variação unidade, exatidão, resolução, limites, média e valores críticos são definidos sempre que necessário?					
As características relativas à descrição das entradas e saídas de cada função (tais como precisão, média, frequência, volume, etc.) foram definidas?					
Classe – Omissão / Tipo – Ambiente Omitida (AO)					
	Sim	Não	Parcial	N/A	
A funcionalidade do hardware, software, banco de dados ou fluxo de atividades, relacionadas com o sistema estão descritos?					
Classe – Omissão / Tipo – Interface Omitida (IO)					
	Sim	Não	Parcial	N/A	
As entradas e as saídas definidas para cada uma das interfaces são suficientes?					
Classe – Comissão / Tipo – Informação Ambígua (IA)					
	Sim	Não	Parcial	N/A	
O requisito está definido de forma discreta, não ambíguo e testável?					
Todas as transições entre modos ou estados do sistema são especificadas claramente e corretamente?					
Classe – Comissão / Tipo – Informação Inconsistente (II)					
	Sim	Não	Parcial	N/A	
O requisito funcional é consistente entre os demais?					
O requisito funcional é consistente com a visão geral?					
O requisito funcional é consistente com o ambiente em que o sistema será executado?					
O requisito funcional é consistente com o ambiente de desenvolvimento?					
Classe – Comissão / Tipo – Funcionalidade Incorreta (FI)					
	Sim	Não	Parcial	N/A	
Todas as funções descritas são necessárias para se alcançar os objetivos do sistema?					
Todas as entradas para cada função são necessárias para implementar a função?					
As entradas e saídas, para todas as interfaces definidas, são necessárias?					




Todas as saídas produzidas, para cada função, são usadas por outras funções ou transferidas através de alguma interface externa?				
Classe – Comissão / Tipo – Seção Errada (SE)				
	Sim	Não	Parcial	N/A
Todos os requisitos, interfaces, restrições, etc, estão listados nas seções apropriadas?				
Classe / Tipo – Outros				
	Sim	Não	Parcial	N/A
O DRS segue o formato padrão de documentação?				
O DRS possui algum erro de gramática?				

APÊNDICE D – Gabarito de defeitos dos documentos de requisitos



FORMULÁRIO DE INSPEÇÃO DE DOCUMENTOS DE SOFTWARE			
Técnica:		Grupo:	
Documento Inspecionado: UC032 Definir Cronograma de elaboração			
Horário da Inspeção: Início () Término ()			
Nome do Participante:			
Observações:			
			
Nº Defeito	Tipo Defeito	Seção	Descrição do Defeito
01	SE	Mockups	[M3] - Wizard Cadastro Plano de Ensino – Objetivo Específico / Não faz parte deste documento
02	AO	Personas	Não consta as personas do documento
03	II	Mockups	Tipo de campo CURSO está incorreto no "[M1] - Filtro / Listagem"
04	II	Mockups	Mensagem do campo DATA DE INICIO está incorreta no "[M2] - Cadastrar Cronograma"
05	II	Mockups	Mascara da DATA DE FIM incorreta no "[M2] - Cadastrar Cronograma"
06	FI	Cenário Geral	CG12 – Ação “Alterar” e “Alteração” não condiz com a funcionalidade
07	O-Gramática	Cenário Geral	CG-002 – Listar
08	O-Gramática	Cenários	Mensagem [MSG003] do cenário [CG-004] Excluir/Remove possui erro “Registro cadastrado!”.
09	O-Gramática	Cenários	Mensagem [MSG001] do cenário [CG-004] Excluir/Remove possui erro “Nenhuma”.
10	O-Gramática	Cenários	Mensagem [MSG007] do cenário [CG-007] – Filtrar possui erro na palavra “nenhuma”.
11	O-Gramática	Cenários	Mensagem [MSG001] do cenário [CG-002] - Listar possui erro “Nenhuma”.

12	O- Gramática	Cenários	Mensagem [MSG005.3] do cenário [CG-012] - Cadastrar em Modal possui erro na palavra "informação".
13	O- Gramática	Cenários	Mensagem [MSG007] do cenário [CG-008] – Pesquisar sem botão possui erro na palavra "nenhuma".
14	O- Gramática	Cenários	Mensagem [MSG003] do campo "Data de início" possui erro "Registro cadastrado!".

TIPOS DE DEFEITOS	
OMISSÃO	COMISSÃO
FO – FUNCIONALIDADE OMITIDA	IA – INFORMAÇÃO AMBÍGUA
PO – PERFORMANCE OMITIDA	II – INFORMAÇÃO INCONSISTENTE
AO – AMBIENTE OMITIDO	FI – FUNCIONALIDADE INCORRETA
IO – INTERFACE OMITIDA	SE – SEÇÃO ERRADA
O – OUTROS	

FORMULÁRIO DE INSPEÇÃO DE DOCUMENTOS DE SOFTWARE			
Técnica:		Grupo:	
Documento Inspecionado: UC019 Elaborar Plano de Ensino			
Horário da Inspeção: Início () Término ()			
Nome do Participante:			
Observações:			
		   Bacharelados em Computação Fábrica de Tecnologias Turing	
Nº Defeito	Tipo Defeito	Seção	Descrição do Defeito
01	FO	Regras de Negocio	Status de plano de ensino não está correto como “Cadastrado” para ser enviado para validação.
02	II	Cenários	Nome do cenário [CG-001] está incorreto na HU1.
03	II	Cenários	Nome do cenário [CG-002] está incorreto na HU1.
04	FI	Mockups	Ação do comando “PRÓXIMO” está incorreta na HU1.
05	FI	Mockups	Ação do comando “CANCELAR” está incompleta na HU1.
06	FO	Critérios de aceitação	Falta C.A. na HU1 no [M1] “Não serão aceitos campos vazios”.
07	FO	Critérios de aceitação	Falta C.A. na HU1 no [M1] “Não serão aceitos campos preenchidos somente com espaço”.
08	FO	Critérios de aceitação	Falta C.A. na HU1 no [M1] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
09	FO	Critérios de aceitação	Falta C.A. na HU1 no [M1] “O Wizard deverá ser capaz de manter as informações ao prosseguir para as próximas etapas”.
10	FO	Critérios de aceitação	Falta C.A. na HU1 no [M1] “As bibliografias devem ser tragas da disciplina”.
11	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “Os wizards poderão ser selecionados”.
12	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “Não serão aceitos campos vazios”.
13	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “Não serão aceitos campos preenchidos somente com espaço”.
14	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
15	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “O Wizard deverá ser capaz de manter as informações ao prosseguir para as próximas etapas”.
16	FO	Critérios de aceitação	Falta C.A. na HU1 no [M5] “Ao clicar em "Voltar", o Wizard deverá manter as informações preenchidas na etapa atual além de apresentar as informações das etapas anteriores de forma integral e sem alterações”.
18	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “Não serão aceitos campos vazios”.

19	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “Não serão aceitos campos preenchidos somente com espaço”.
20	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
21	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “O Wizard deverá ser capaz de manter as informações ao prosseguir para as próximas etapas”.
22	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “Ao clicar em "Voltar", o Wizard deverá manter as informações preenchidas na etapa atual além de apresentar as informações das etapas anteriores de forma integral e sem alterações”.
23	FO	Critérios de aceitação	Falta C.A. na HU1 no [M6] “No plug-in de texto deve conter: Recursos de Acessibilidade disponíveis aos acadêmicos: (em negrito) O curso assegura acessibilidade metodológica, digital, comunicacional, atitudinal, instrumental e arquitetônica, garantindo autonomia plena do discente. O docente poderá alterar se necessário”.
24	FO	Critérios de aceitação	Falta C.A. na HU1 no [M8] “Não serão aceitos campos vazios”.
25	FO	Critérios de aceitação	Falta C.A. na HU1 no [M8] “Não serão aceitos campos preenchidos somente com espaço”.
26	FO	Critérios de aceitação	Falta C.A. na HU1 no [M8] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
27	FO	Critérios de aceitação	Falta C.A. na HU1 no [M8] “O Wizard deverá ser capaz de manter as informações ao prosseguir para as próximas etapas”.
28	FO	Critérios de aceitação	Falta C.A. na HU1 no [M8] “Ao clicar em "Voltar", o Wizard deverá manter as informações preenchidas na etapa atual além de apresentar as informações das etapas anteriores de forma integral e sem alterações”.
29	FO	Critérios de aceitação	Falta C.A. na HU2 no [M1] “Não serão aceitos campos vazios”.
30	FO	Critérios de aceitação	Falta C.A. na HU2 no [M1] “Não serão aceitos campos preenchidos somente com espaço”.
31	FO	Critérios de aceitação	Falta C.A. na HU2 no [M1] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
32	FO	Critérios de aceitação	Falta C.A. na HU2 no [M3] “Não serão aceitos campos vazios”.
33	FO	Critérios de aceitação	Falta C.A. na HU2 no [M3] “Não serão aceitos campos preenchidos somente com espaço”.
34	FO	Critérios de aceitação	Falta C.A. na HU2 no [M3] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
35	FO	Critérios de aceitação	Falta C.A. na HU2 no [M4] “Não serão aceitos campos vazios”.
36	FO	Critérios de aceitação	Falta C.A. na HU2 no [M4] “Não serão aceitos campos preenchidos somente com espaço”.
37	FO	Critérios de aceitação	Falta C.A. na HU2 no [M4] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
38	FO	Critérios de aceitação	Falta C.A. na HU3 no [M1] “Não serão aceitos campos vazios”.

39	FO	Critérios de aceitação	Falta C.A. na HU3 no [M1] “Não serão aceitos campos preenchidos somente com espaço”.
40	FO	Critérios de aceitação	Falta C.A. na HU3 no [M1] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
41	FO	Critérios de aceitação	Falta C.A. na HU3 no [M3] “Não serão aceitos campos vazios”.
42	FO	Critérios de aceitação	Falta C.A. na HU3 no [M3] “Não serão aceitos campos preenchidos somente com espaço”.
43	FO	Critérios de aceitação	Falta C.A. na HU3 no [M3] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
44	FO	Critérios de aceitação	Falta C.A. na HU3 no [M4] “Não serão aceitos campos vazios”.
45	FO	Critérios de aceitação	Falta C.A. na HU3 no [M4] “Não serão aceitos campos preenchidos somente com espaço”.
45	FO	Critérios de aceitação	Falta C.A. na HU3 no [M4] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
46	FO	Critérios de aceitação	Falta C.A. na HU4 no [M1] “Não serão aceitos campos vazios”.
47	FO	Critérios de aceitação	Falta C.A. na HU4 no [M1] “Não serão aceitos campos preenchidos somente com espaço”.
48	FO	Critérios de aceitação	Falta C.A. na HU4 no [M1] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
49	FO	Critérios de aceitação	Falta C.A. na HU4 no [M2] “Não serão aceitos campos vazios”.
50	FO	Critérios de aceitação	Falta C.A. na HU4 no [M2] “Não serão aceitos campos preenchidos somente com espaço”.
51	FO	Critérios de aceitação	Falta C.A. na HU4 no [M2] “Ao cancelar, os dados preenchidos serão automaticamente apagados”.
52	FO	Critérios de aceitação	Falta C.A. na HU4 no [M2] “Ao clicar em editar, deverá carregar o texto já preenchido nos campos anteriormente”.
53	II	Mockups	Tipo de Campo “SALVAR E CONTINUAR” está incorreto na HU1 no "[M2] - Wizard Cadastro Plano de Ensino – Objetivo Geral”.
54	FI	Mockups	Restrição do comando “  Comentário” está incorreto na HU1 no "[M2] - Wizard Cadastro Plano de Ensino – Objetivo Geral”.
55	II	Mockups	Tipo do comando “  CHECKBOX” está incorreto na HU1 no "[M4] - Wizard Cadastro Plano de Ensino - Habilidades e Competências”.
56	O-Gramática	Regras de Negócio	“Dato”
57	O-Gramática	Mockup	“Cadada” [M3] HU1
58	O-Gramática	Mockup	“Poderei” [M4] HU1
59	O-Gramática	Mockup	“informassões” [M4] HU1
60	O-Gramática	Mockup	“Didathicos” [M6] HU1
61	O	História 1	“pessoa” deveria ser “persona”
62	IA	Cenário Geral	CG-004 – 2 mensagens incorretas
63	FI	Cenário Geral	CG-003 – Ação “Cancelar/Voltar” não condiz com a funcionalidade
64	O-Gramática	Cenário Geral	CG-003 – Existentee

65	O-Gramática	Cenário Geral	CG-001 – cadastrar
66	O-Gramática	Cenário Geral	CG-002 – Listar
67	O-Gramática	Cenários	Mensagem [MSG005.3] do cenário [CG-001] – Cadastrar possui erro na palavra “informação”.
68	O-Gramática	Cenários	Mensagem [MSG001] do cenário [CG-002] - Listar possui erro “Nenhuma”.
69	O-Gramática	Cenários	Mensagem [MSG003] do cenário [CG-004] - Excluir/Remover possui erro “Registro cadastrado!”.
70	O-Gramática	Cenários	Mensagem [MSG001] do cenário [CG-004] - Excluir/Remover possui erro “Nenhuma”.
71	II	Cenários	Nome do cenário [CG-005] está incorreto na HU4.
72	O-Gramática	Cenários	Mensagem [MSG005.2] do cenário [CG-005] – Ativar possui erro na palavra “ativa”.
73	O-Gramática	Mockup	Mensagem [MSG005] do comando “Salvar” [M3] – Cadastrar Estratégias – HU2 possui erro no padrão (Não informa qual variação da msg deve ser mostrada).
74	O-Gramática	Mockup	Mensagem [MSG – 006.1] do comando “Excluir” [M3] - Wizard Cadastro Plano de Ensino – Objetivo Específico – HU1 possui erro no padrão de escrita.

TIPOS DE DEFEITOS	
OMISSÃO	COMISSÃO
FO – FUNCIONALIDADE OMITIDA	IA – INFORMAÇÃO AMBÍGUA
PO – PERFORMANCE OMITIDA	II – INFORMAÇÃO INCONSISTENTE
AO – AMBIENTE OMITIDO	FI – FUNCIONALIDADE INCORRETA
IO – INTERFACE OMITIDA	SE – SEÇÃO ERRADA
O – OUTROS	

FORMULÁRIO DE INSPEÇÃO DE DOCUMENTOS DE SOFTWARE			
Técnica:	Grupo:		
Documento Inspeccionado: UC008 Manter Competências e Habilidades			
Horário da Inspeção: Início () Término ()			
Nome do Participante:			
Observações:			
Nº Defeito	Tipo Defeito	Seção	Descrição do Defeito
01	IO	Mockup	Falta o campo “Categoria” no Mockup [M3] - Cadastrar Competências e Habilidades.
02	II	Como Acessar	O caminho para acessar a funcionalidade está incorreto.

03	IA	Mockup	O tipo de dado do campo "CURSO" está incorreto na descrição dos campos "[M3] – Cadastrar Competências e Habilidades" (Deveria ser cursos e não disciplinas).
04	II	Mockup	O nome do campo "Voltar" está incorreto, deveria ser "Cancelar" na descrição dos comandos na "[M3] - Cadastrar Competências e Habilidades".
05	FI	História	A descrição da história está incorreta.
06	FO	Mockup	A tabela de Descrição de Comandos não apresenta o comando "Salvar" no "[M3] - Cadastrar Competências e Habilidades".
07	AO	Personas	O documento não especifica as personas.
08	II	Mockup	O tipo do campo "Curso" está incorreto no "[M1] – Filtrar Competências e Habilidades".
09	O- Gramática	Mockup	Tipo de campo Curso erro na palavra "Esses" no [M1] - Filtrar Competências e Habilidades.
10	O- Gramática	Mockup	Observações do campo Categoria erro na palavra "categoria" no [M1] - Filtrar Competências e Habilidades.
11	O- Gramática	Mockup	Ação do comando Filtrar erro na palavra "habilidade" no [M1] - Filtrar Competências e Habilidades.
12	O- Gramática	Mockup	Ação do comando +Competência erro na palavra "cadastra" no [M2] - Listar Competências e Habilidades.
13	O- Gramática	Mockup	A observação do campo Código possui erro na palavra "automaticament" no [M3] - Cadastrar Competências e Habilidades.
14	O	Histórico de Versão	Sequencia de versão está incorreta.
15	FI	Cenário Geral	CG12 – Ação "Alterar" e "Alteração" não condiz com a funcionalidade
16	O- Gramática	Cenários	Mensagem [MSG005.3] do cenário [CG-012] - Cadastrar em Modal possui erro na palavra "informação".
17	O- Gramática	Cenários	Mensagem [MSG007] do cenário [CG-007] – Filtrar possui erro na palavra "nenhuma".
18	O- Gramática	Cenários	Mensagem [MSG003] do cenário [CG-004] - Excluir/Remove possui erro "Registro cadastrado!".
19	O- Gramática	Cenários	Mensagem [MSG001] do cenário [CG-004] - Excluir/Remove possui erro "Nenhuma".
20	O- Gramática	Cenários	Mensagem [MSG007] do cenário [CG-008] – Pesquisar sem botão possui erro na palavra "nenhuma".

TIPOS DE DEFEITOS	
OMISSÃO	COMISSÃO
FO – FUNCIONALIDADE OMITIDA	IA – INFORMAÇÃO AMBÍGUA
PO – PERFORMANCE OMITIDA	II – INFORMAÇÃO INCONSISTENTE
AO – AMBIENTE OMITIDO	FI – FUNCIONALIDADE INCORRETA
IO – INTERFACE OMITIDA	SE – SEÇÃO ERRADA
O – OUTROS	