

Universidade Federal do Estado do Rio de Janeiro Centro de Ciências Exatas e Tecnologia Escola de Informática Aplicada

TesterQuiz: Ferramenta de Apoio ao Aprendizado de Testes de Software

Lucas Ciarlini Guilhon

Orientador

Gleison dos Santos Souza

RIO DE JANEIRO, RJ – BRASIL FEVEREIRO DE 2020

TesterQuiz: Ferramenta de Apoio ao Aprendizado de Testes de Software

Lucas Ciarlini Guilhon

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado	por:
I I p I o , a a c	POI.

Prof. Gleison dos Santos Souza, DSc. (UNIRIO)

Prof. Jobson Luiz Massollar da Silva, DSc. (UNIRIO)

Prof. Fabrício Raphael Silva Pereira, DSc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL Fevereiro de 2020

Catalogação informatizada pelo autor

Guilhon, Lucas Ciarlini

G953 TesterQuiz: Ferramenta de Apoio ao Aprendizado
de Testes de Software / Lucas Ciarlini Guilhon. -Rio de Janeiro, 2020.
73 f.

Orientador: Gleison dos Santos Souza. Trabalho de Conclusão de Curso (Graduação) -Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2020.

Teste de software. 2. Quiz. 3. Aprendizado.
 Souza, Gleison dos Santos, orient. II. Título.

Agradecimentos

Agradeço primeiramente aos meus pais e irmãos pelo apoio, amor e cumplicidade imprescindíveis para a minha vida.

A minha amada esposa, Olívia, pelo amor, carinho, companheirismo e força que expandem minha vida todos os dias. Por ter me dado suporte e ficado ao meu lado em um período tão conturbado, como sempre fez.

Aos docentes pela atenção, pelo zelo, pelo ensino de qualidade e pela excelente formação, aos quais serei eternamente grato.

Aos técnicos da secretaria do BSI por toda a ajuda que me proveram ao longo dos anos, sempre prestativos e atenciosos.

Aos colegas discentes, pelo apoio e companheirismo ao longo do curso.

Ao meu orientador, o professor Gleison dos Santos, pela paciência, dedicação e orientação. Este trabalho não seria possível sem sua imprescindível ajuda.

Aos professores da banca e em especial ao professor Jobson Luiz, pelo apoio durante o projeto.

RESUMO

Este trabalho apoia o aprendizado dos alunos de cursos de Tecnologia da Informação,

visando aprimorar a absorção e retenção do conhecimento sobre testes de software adquirido

ao longo do curso. O trabalho apresenta o TesterQuiz que possibilita a realização de um quiz

dividido em três tópicos importantes da literatura sobre testes: Tipos e Técnicas de Teste, Níveis

de Teste e Ciclo de Vida de Teste. A aplicação do TesterQuiz pode permitir ao professor

analisar o conhecimento dos alunos e das turmas, procurando apontar as possíveis dificuldades

na absorção do conhecimento.

Para a criação do TesterQuiz, foi feito um levantamento das funcionalidades desejadas,

tanto para o aluno quanto para o professor, a partir da análise de ferramentas disponíveis para

elaboração de quiz. A aplicação foi construída utilizando-se a infraestrutura do Google Forms.

Foram definidas, inicialmente, 60 perguntas, sendo 19 sobre o Tipos e Técnicas de Teste, 21

sobre Níveis de Teste e 20 sobre Ciclo de Vida de Teste.

Espera-se que o trabalho ajude alunos a revisar e reforçar os conhecimentos na área e

auxilie professores a identificar pontos que necessitem de reforço teórico na disciplina,

contribuindo positivamente para a formação de profissionais de tecnologia da informação.

Palavras-chave: Teste de software, Quiz, Aprendizado

ii

ABSTRACT

This work supports learning of Information Technology undergraduate students,

seeking to improve the absorption and retaining of knowledge about software testing, acquired

throughout their undergraduate program. We present TesterQuiz, which allows the execution

of a quiz divided into three important topics from the software testing literature: Test Types and

Techniques, Test Levels and Test Life Cycle. By using TesterQuiz teachers can analyze

students' knowledge on software test and identify possible difficulties in knowledge absorption.

To create TesterQuiz, we analyzed some quiz tools to identify functionalities adequate

to both students and teachers. TesterQuiz was built using the Google Forms infrastructure.

Initially, we defined 60 questions, being 19 about Test Types and Techniques, 21 about the Test

Levels and 20 about the Test Life Cycle.

We expect that this work will help students to review and reinforce their knowledge on

software test and teachers to identify knowledge gaps that need reinforcement, positively

contributing to the education of Information Technology professionals.

Keywords: Software test, Quiz, Learning

iii

Índice

1. INTRODUÇÃO	1
1.1. Contexto	1
1.2. Motivação	1
1.3. Objetivo	2
1.4. Avaliação do trabalho	2
1.5. Estrutura do Texto	2
2. Revisão da Literatura	3
2.1. Testes de Software	3
2.2. Educação em Engenharia de Software	4
2.3. Trabalhos Relacionados	6
2.4. Considerações Finais	6
3. Tecnologias Avaliadas	7
3.1. Levantamento dos Requisitos	7
3.2. Análise das Ferramentas	7
3.2.1. Quiz Maker	7
3.2.2. TriviaMaker	Ç
3.2.3. Kahoot!	10
3.2.4. Google Forms	11
3.2.5. Trivia Quiz	12
3.2.6. Django School	14
3.3. Levantamento das Funcionalidades	15
3.3.1. Características Desejáveis	16
3.4. Seleção da Ferramenta	17
3.5. Considerações Finais	17
4. Ferramenta TesterQuiz	18
4.1. Objetivo da Implementação	18

	4.2.	Identificação das Questões	18
	4.3.	Implementação	20
	4.4.	Considerações Finais	39
5	. A	valiação do TesterQuiz	40
	5.1.	Planejamento	40
	5.2.	Definição da métrica de avaliação	41
	5.3.	Execução	41
	5.4.	Avaliação com base no TAM	43
	5.5.	Considerações Finais	45
6	. C	CONCLUSÃO	46
	6.1.	Considerações Finais	46
	6.2.	Contribuições	46
	6.3.	Limitações e Trabalhos Futuros	46
7	. R	EFERÊNCIAS	48
8	. A	pêndice I	52

1. INTRODUÇÃO

1.1. Contexto

A cultura de teste de software é muito importante para o controle de qualidade no desenvolvimento de um sistema. Um bom processo de teste deve assegurar que o sistema atenda a todos os requisitos solicitados pelo cliente. Sob o ponto de vista do negócio, os testes de software são recursos utilizados para minimizar os custos de desenvolvimento e manutenção, além de evitar futuros prejuízos decorrentes do mau funcionamento do software.

A demanda por analistas de testes no mercado de T.I. tem crescido no Brasil em anos recentes. Empresas operando no país tem demandado maior qualidade no desenvolvimento de softwares nacionais. Uma evidência desse aumento é o surgimento de certificações emitidas por órgãos como o BSTQB¹ e o ALATS² que visam habilitar os profissionais que desejam seguir a carreira de analista de testes.

Além do aprendizado prático, adquirido ao longo da vida profissional de um analista de testes, o conhecimento é passado na formação universitária de cursos como Sistemas de Informação, Ciências da Computação e Engenharia da Computação. No curso de Bacharelado em Sistemas de Informação da Unirio, apenas uma disciplina obrigatória do curso (TIN0121 - Programação Modular) e uma disciplina optativa recém-criada (TIN0166 - Tópicos Avançados em Engenharia de Software II) abordam explicitamente testes de software. Devido à ementa da disciplina Programação Modular envolver diversos outros conteúdos além de testes de software que precisam ser juntamente lecionados e avaliados, não há uma avaliação específica do conhecimento em testes de software na disciplina.

1.2. Motivação

A motivação deste trabalho é a ausência de apoio a docentes de disciplinas que abordem Testes de Software na identificação de potenciais deficiências no conhecimento do assunto por parte dos alunos. Espera-se que o trabalho possa auxiliar o aprendizado dos graduandos e o aprimoramento de profissionais na área de testes.

¹ Disponível em: https://www.bstqb.org.br/> Acesso em: 06 fev, 2020.

² Disponível em: <<u>http://www.alats.org.br/portal/</u>> Acesso em: 06 fev, 2020.

1.3. **Objetivo**

O objetivo deste trabalho é desenvolver um *quiz* sobre testes de software, visando auxiliar o aprendizado dos alunos que cursem disciplinas que abordem testes de software. Como resultado da aplicação do quiz, objetiva-se, também, indicar ao professor os tópicos onde os alunos possuem mais dificuldade e sugerir aos alunos algum tipo de leitura complementar. Como resultado, foi criada uma ferramenta denominada TesterQuiz.

1.4. Avaliação do trabalho

Para avaliação do trabalho proposto, o *quiz* foi submetido aos alunos cursando a disciplina Técnicas Avançadas em Engenharia de Software II do Bacharelado em Sistemas de Informação da Unirio (http://bsi.uniriotec.br/bsi/index.html) no período de curso de férias do primeiro semestre de 2020. Ao fim da aplicação do quiz, os dados coletados foram utilizados para gerar um relatório. Foi também utilizado um formulário baseado no TAM (*Technology Acceptance Model*) (DAVIS et al., 1989) visando avaliar a aceitação do TesterQuiz em relação à percepção de utilidade e de facilidade de uso pela perspectiva dos alunos e do professor ministrando a disciplina.

1.5. Estrutura do Texto

O presente trabalho está estruturado em capítulos e, além desta Introdução, será apresentado da seguinte forma:

- Capítulo II: Apresenta a fundamentação da importância de testes de *software*, fundamentada na literatura existente sobre o tema.
- Capítulo III: Apresenta o levantamento dos requisitos e funcionalidades do Tester*Q*uiz a partir da análise de ferramentas relacionadas existentes.
- Capítulo IV: Apresenta a descrição da ferramenta desenvolvida, seu processo de criação e as questões elaboradas.
- Capítulo V: Aborda o planejamento e a execução do TesterQuiz em sala de aula, assim como sua avaliação pelo TAM (*Technology Acceptance Model*) (DAVIS et al., 1989), realizada pelos alunos e por um docente.
- Capítulo VI: Agrupa as considerações finais, as limitações encontradas, assinala as contribuições do projeto e sugere possíveis trabalhos futuros.

2. Revisão da Literatura

Neste capítulo será apresentada uma visão geral sobre testes de software e educação em engenharia de software.

2.1. Testes de Software

Segundo Sommerville (2011), o teste de software é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso. Quando se testa o software, o programa é executado usando dados fictícios. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa.

Ainda segundo o autor, o processo de teste tem dois objetivos distintos:

- Trazer mais confiança ao desenvolvedor e ao cliente de que o software atende aos requisitos. Para softwares desenvolvidos sob encomenda, isso significa que deve haver pelo menos um teste para cada requisito do documento de requisitos. Para softwares de prateleira, isso significa que deve haver testes para todas as características do sistema, além de suas combinações, que serão incorporadas ao release do produto.
- Descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma diferente das especificações. Essas são consequências de defeitos de software.

Segundo Eliza e Lagares (2012), o ciclo de vida de teste consiste em uma série de etapas dependentes, consideradas como o esqueleto do Processo de Teste, que visam estruturar as atividades definindo como os testes serão conduzidos no projeto de software. Sobre o ciclo de vida do teste de software, Farias (2014) descreve que ele é composto de 5 etapas. Na etapa de planejamento, que acontece paralelamente às atividades de levantamento de requisitos do projeto de desenvolvimento do software, ocorrem a elaboração da Estratégia de Teste, do Plano de Teste, e a análise de risco do projeto de testes. Na próxima etapa, ocorre a preparação do ambiente (equipamentos, hardware e software, ferramentas para automação de testes, entre outros) para a execução dos testes. Na etapa seguinte, ocorre a elaboração e revisão dos casos

e roteiros de teste. Em seguida, ocorre a etapa de *execução* com o registro dos resultados obtidos durante a execução. Por fim, ocorre a etapa de *entrega*, com o arquivamento da documentação referente ao projeto e o relato de todas as ocorrências encontradas a fim de melhorar o processo.

Os testes de software são fundamentais para o bom desenvolvimento e facilidade de manutenção do software. Segundo Bernardo e Kon (2008), os sistemas de software devem não só fazer corretamente o que o cliente precisa, mas também fazê-lo de forma segura, eficiente e escalável e serem de fácil manutenção e evolução. Não é sempre que os profissionais envolvidos no processo de desenvolvimento compreendem a necessidade de um processo adequado de teste de software. Um fator comum de estímulo a uma mudança de prioridades é o fator financeiro, visto que a ausência de um processo adequado de testes pode acarretar em grande prejuízo financeiro.

Um caso famoso relatado na literatura é referente ao Therac-25, um dispositivo computadorizado para tratamento por radiação para câncer. Entre junho de 1985 e janeiro de 1987, o Therac-25 teve diversos equipamentos ministrando doses elevadas a pacientes, pelo menos 6 pacientes receberam doses elevadas, alguns foram mortos e outros incapacitados. O FDA (Food and Drug Administration) investigou o caso e descobriu um programa mal documentado, sem especificação e nem plano de testes. Frank Houston do FDA escreveu em 1985 "Uma quantidade significativa de software para sistemas críticos para a vida vêm de pequenas empresas, especialmente na indústria de equipamentos médicos; empresas que enquadram no perfil daquelas que resistem aos princípios tanto da segurança de sistemas quanto da engenharia de software ou os desconhecem".

De acordo com Pontes (2009), falar de testes de software é muito mais do que falar de execução de testes. Testar um software e relatar impressões e não conformidades é fornecer um diagnóstico do estado da aplicação, e é muito importante que este diagnóstico seja o mais completo e preciso possível, porque provavelmente ele vai servir de base para tomadas de decisões em relação ao projeto que está sendo analisado. Segundo a autora, devido à sua importância e complexidade, testes de software podem ser responsáveis por uma parcela considerável dos custos de um projeto, por isso, este assunto merece atenção por parte das organizações que desejam sucesso com a condução das atividades do processo de testes.

Sobre os níveis de teste, segundo Pressman (2011), considerando o processo de software de um ponto de vista procedimental, o teste dentro do contexto de engenharia de software é na realidade uma série de quatro etapas que são implementadas sequencialmente. De acordo com o autor, inicialmente, os testes focalizam cada componente individualmente, garantindo que ele

funcione adequadamente como uma unidade, daí o nome teste de unidade. O teste de unidade usa intensamente técnicas de teste com caminhos específicos na estrutura de controle para garantir a cobertura completa e a máxima detecção de falhas. Em seguida, o componente deve ser montado ou integrado para formar o pacote completo de software. O teste de integração cuida de problemas associados com aspectos duais de verificação e construção de programa. O teste de validação proporciona a garantia final de que o software satisfaz a todos os requisitos informativos, funcionais, comportamentais e de desempenho. O teste de sistema verifica se todos os elementos se combinam corretamente e se a função/desempenho global do sistema é conseguida.

Quanto às técnicas de teste, de acordo com Sommerville (2011), quando você usa a especificação de um sistema para identificar os casos de teste, isso é chamado 'teste de caixa-preta'. Nesse caso, você não precisa de nenhum conhecimento de como funciona o sistema. No entanto, pode ser útil suplementar os testes de caixa-preta com 'testes de caixa-branca', nos quais você pode consultar o código do programa para encontrar outros casos de teste.

2.2. Educação em Engenharia de Software

Tendo em vista a importância do entendimento dos processos adequados de teste de software pelos profissionais, torna-se de importância ainda maior a análise das metodologias de ensino em engenharia de software em busca de melhorias no aprendizado dos futuros profissionais da área. Como apontado por Lemos *et al.* (2019), diante do volume de conteúdo relacionado à Engenharia de Software e das exigências de um mercado cada vez mais dinâmico, alguns pesquisadores têm identificado as dificuldades no processo de ensino-aprendizagem para compreender e buscar estratégias que preparem os estudantes para as demandas da indústria de software.

As metodologias modernas, diferentemente das mais tradicionais, por vezes procuram adotar uma abordagem mais familiar aos alunos, visando aprimorar a forma de ensino a fim de obter melhores resultados na retenção do conhecimento. Pelo levantamento de Cunha et al. (2018) referente à literatura do ensino de engenharia de software (mas não específico sobre testes de software) no Brasil, pode-se ver a recorrência do uso de jogos no ensino de engenharia de software. Foram encontrados 113 artigos categorizados em metodologias de ensino, jogos, ferramentas e integração entre disciplinas. Os resultados demonstram a predominância de trabalhos relacionados a jogos e metodologias de ensino.

2.3. Trabalhos Relacionados

A ideia de utilizar um *quiz* como ferramenta de apoio no meio acadêmico não é pioneira do trabalho presente. Diversos trabalhos foram propostos utilizando conceitos similares para fins similares. No trabalho de Cassetari (2015), foi utilizada a ferramenta *Kahoot!* para a revisão dos conhecimentos de gerência de projetos com alunos da disciplina Gerência de Projetos do curso de Bacharelado em Sistemas de Informação/INE/UFSC e com os alunos da disciplina Planejamento e Gestão de Projetos do curso de Bacharelado em Ciência da Computação/INE/UFSC.

Na avaliação do trabalho pelos alunos que participaram do *quiz*, pode-se observar uma boa aceitação da metodologia, com 31 alunos de 33 respondendo que concordam com a afirmação "Me diverti com o jogo", qualificando a abordagem utilizada.

Em outro trabalho, também em um contexto acadêmico, Sande e Sande (2018) utilizaram o *Kahoot!*³ como ferramenta de avaliação e ensino-aprendizagem no ensino de microbiologia industrial. A percepção dos alunos participantes foi analisada utilizando um questionário criado na ferramenta *Google Forms* sobre as competências da ferramenta *Kahoot!* como método avaliativo. Na avaliação, 80% dos alunos afirmaram que a ferramenta possui a capacidade de avaliar todo o conteúdo teórico da disciplina de forma mais atraente e competitiva, criando um grande estímulo.

Tendo em mente os trabalhos analisados, a utilização de um *quiz* para auxílio do ensino de teste de software se apresenta como uma abordagem promissora, reforçando a relevância do trabalho presente.

2.4. Considerações Finais

Neste capítulo foi realizada uma revisão da literatura sobre os tópicos relevantes ao trabalho em questão, além de uma análise de trabalhos relacionados. No capítulo seguinte, é apresentada uma análise de ferramentas de *quiz* disponíveis com o objetivo de identificar requisitos e funcionalidades para a ferramenta TesterQuiz.

-

³ Disponível em: <<u>https://kahoot.com/</u>> Acesso em: 04 fev, 2020.

3. Tecnologias Avaliadas

3.1. Levantamento dos Requisitos

Para concepção do *quiz* a ser desenvolvido, foi levantada a lista dos requisitos para cumprimento do objetivo estabelecido neste trabalho. São condições para a elaboração da ferramenta os requisitos propostos a seguir:

- Ser uma tecnologia popular entre professores e alunos;
- Persistência dos dados gerados por sua execução, sendo possível armazenar perguntas realizadas e os resultados das respostas dos alunos em um banco de dados ou planilha;
- Ser uma tecnologia acessível a alunos de cursos na área.

3.2. Análise das Ferramentas

Diversas tecnologias e aplicativos que poderiam ser utilizados na elaboração do *quiz* foram analisados, buscando observar os requisitos do *quiz* a ser elaborado. Foram selecionados entre os primeiros itens da busca pelo termo *quiz* no site de repositórios GitHub (https://github.com/), na loja de aplicativos Google Play Store (https://play.google.com/store) e outros conhecidos previamente (*Kahoot!* e *TriviaMaker*), além do *Google Forms*, que é uma ferramenta muito popular para elaboração de questionários on-line. Assim, as ferramentas pesquisadas foram:

- *Kahoot!* (<u>https://kahoot.com</u>)
- Quiz Maker (https://play.google.com/store/apps/details?id=com.devup.qcm.maker)
- *TriviaMaker* (https://www.triviamaker.com)
- Google Forms (https://docs.google.com/forms)
- TriviaQuiz (https://github.com/sarveshchavan7/Quiz-Game)
- Django School Application (https://github.com/sibtc/django-multiple-user-types-example)

A seguir são apresentadas características dessas ferramentas. Na Seção 3.3, é apresentado um sumário das suas funcionalidades.

3.2.1. Quiz Maker

A ferramenta Quiz Maker é um aplicativo gratuito para celulares com o sistema operacional Android que permite a criação, execução e o compartilhamento de um *quiz* entre

seus usuários. Na Figura 1 pode ser vista a página frontal da ferramenta no Google Play Store, contendo algumas telas do software.

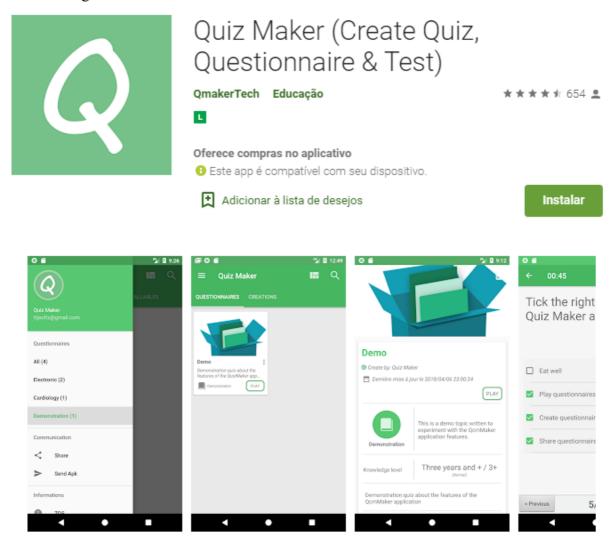


Figura 1 - Página do Quiz Maker na Google Play Store

A seguir são apresentadas algumas das características observadas no Quiz Maker:

- Permite o salvamento e carregamento de perguntas em arquivo formato ".qcm", que pode ser lido pelo aplicativo, permitindo um armazenamento e mobilidade de questões fora do contexto do aplicativo.
- Possui a funcionalidade de temporizador para o teste como um todo, mas não para as questões individuais.
- Após a execução do quiz, apresenta os resultados, mas os descarta assim que retorna para a tela inicial, não permitindo construir nenhum relatório a partir dos dados de execução.

 Permite a criação de uma explicação de cada resposta do quiz, a ser(em) exibida(s) ao término do quiz.

3.2.2. TriviaMaker

A ferramenta TriviaMaker é um aplicativo, com versões gratuita e paga para celulares com os sistemas operacionais Android e iOS, que permite a criação e a execução de diversos tipos de *trivia game*, dentre os quais, um *quiz game*. Na Figura 2 pode ser visto um recorte da página oficial do aplicativo, contendo exemplos dos formatos de jogos.

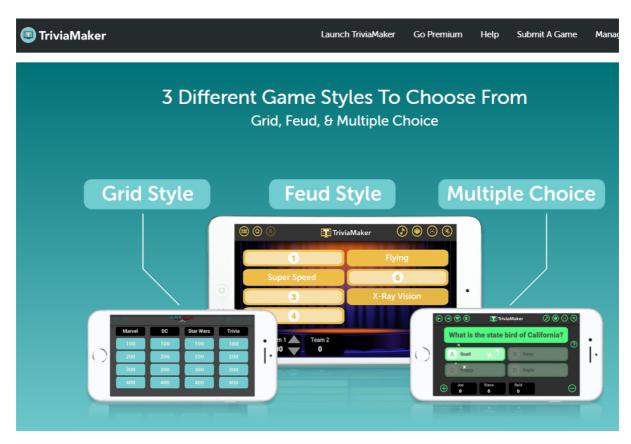


Figura 2 - Página oficial do TriviaMaker

A seguir são apresentadas algumas das características observadas no TriviaMaker:

- No modo básico e gratuito, permite apenas 2 respostas por pergunta.
- Possui compra vitalícia (sem mensalidade) com valor acessível, que amplia a
 funcionalidade da ferramenta, permitindo que 4 equipes realizem o quiz ao invés de 2,
 4 repostas por quiz ao invés de 2 e opções customizadas de layout, incluindo cor/figura
 de fundo, cores, logotipos e música de fundo.

 Permite apenas ser jogado em equipes e no máximo 4 equipes em um quiz na versão paga.

3.2.3. **Kahoot!**

O Kahoot! é um software voltado para a criação e execução de *quiz games* com versões gratuita e paga. A ferramenta utiliza um navegador para apresentar o *quiz* desenvolvido, enquanto os participantes devem realizar um check-in no aplicativo que deve ser instalado em seus celulares com os sistemas operacionais Android ou iOS e responder às perguntas apresentadas no navegador. Na Figura 3 pode ser visto um recorte da página de download do aplicativo na Google Play Store contendo algumas telas da ferramenta.

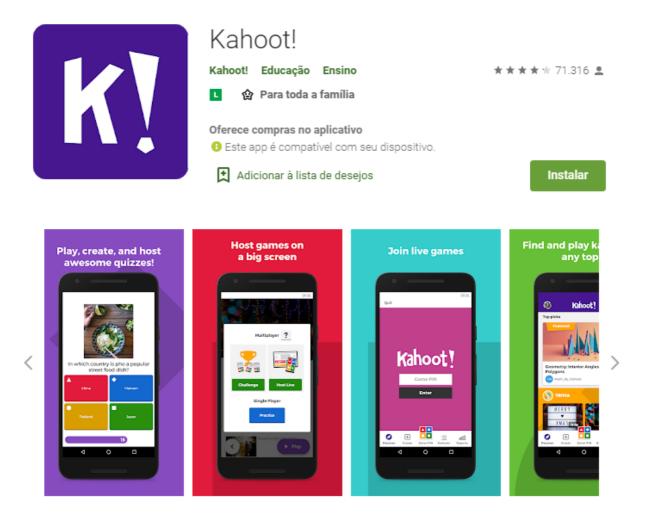


Figura 3 - Página do Kahoot! na Google Play Store

Algumas das características observadas no Kahoot! são listadas a seguir:

- Exporta um relatório em formato ".xls" que compreende mais pontos que as demais ferramentas, como pergunta mais acertada e mais errada e desempenho geral de uma turma, além da evolução individual.
- Possui uma versão Premium e uma Pro (para o professor) a um preço acessível, não sendo necessário aos alunos pagarem, que permitem a geração de um relatório avançado com o progresso do aluno.
- Permite, nas versões Premium e Pro, a inserção de slides de explicação da questão.

3.2.4. Google Forms

O Google Forms é uma ferramenta gratuita que permite a criação e o compartilhamento de formulários entre contas Google. Os formulários podem ser acessados em celulares e através de navegadores, tornando o sistema operacional um fator não impeditivo na utilização da ferramenta. Na Figura 4 é exibida uma parte da página de criação de um questionário utilizando a ferramenta.

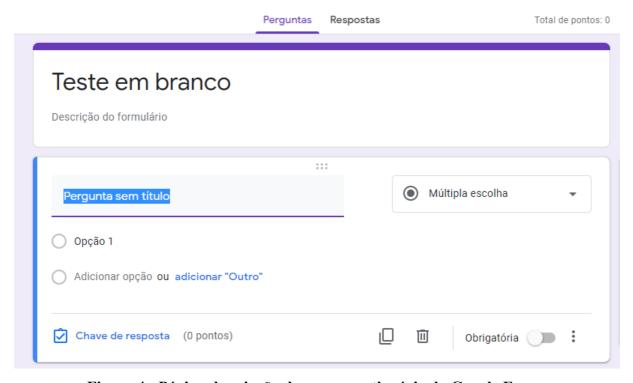


Figura 4 - Página de criação de novo questionário do Google Forms

Abaixo são listadas algumas das características identificadas na utilização do Google Forms:

- Assim como o QuizMaker, permite a criação de uma explicação detalhada da resposta, exibida ao término do quiz.
- Permite uma quantidade alta de caracteres na explicação detalhada da resposta.
- Possui formas de temporização do quiz através de extensões, como o Timify⁴ ou o formLimiter⁵.
- O resultado é salvo em uma planilha com data e hora, a pontuação do aluno e as respostas dadas para cada pergunta, sendo preciso tratar esses dados para se obter um relatório mais abrangente.

3.2.5. Trivia Quiz

O Trivia Quiz é um aplicativo de código aberto para celulares com sistema operacional Android, que permite a criação e execução de um *quiz* no próprio dispositivo em que foi criado. Na Figura 5 é exibida uma coleção de telas apresentadas na página do repositório da ferramenta.

٠

⁴ Disponível em: <<u>https://timify.me/</u>> Acesso em: 04 fev, 2020.

⁵ Disponível em: https://gsuite.google.com/marketplace/app/formlimiter/538161738778 Acesso em: 04 fev, 2020.

Screenshots

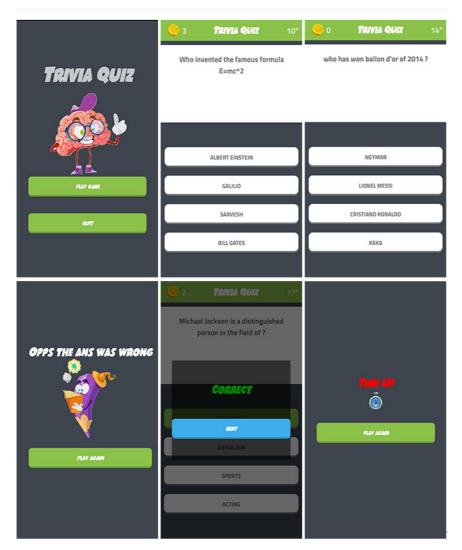


Figura 5 - Telas apresentadas na página do repositório da ferramenta Trivia Quiz

A seguir, uma lista das características observadas na versão presente no repositório da ferramenta:

- Desenvolvido para a plataforma Android.
- Permite temporização por questão.
- Não salva os registros da realização do quiz.

3.2.6. Django School

O Django School é um aplicativo de código aberto desenvolvido em Python, com o framework Django, que permite a criação e execução de um *quiz*. Na Figura 6 é apresentada a página da ferramenta hospedada.

Django School



This is an example project to illustrate an implementation of multiple user types. In this Django app, teachers can create quizzes and students can sign up and take quizzes related to their interests.

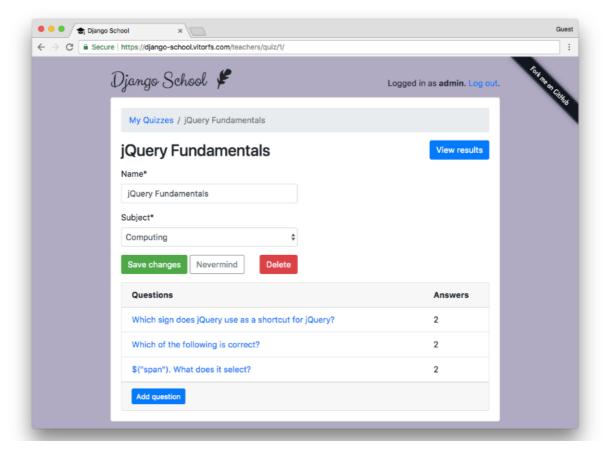


Figura 6 - Página da ferramenta Django School hospedada

A seguir, uma lista das características observadas na versão disponibilizada no repositório:

- Desenvolvida em linguagem Python.
- Possui uma versão já hospedada no site do desenvolvedor, sendo necessário aos alunos apenas uma criação de conta básica (usuário e senha) para acessar o quiz gerado pelo professor.

 Na versão hospedada, os dados não são exportados para um arquivo acessível ao professor, sendo apenas exibidas em tela. É necessário hospedar e utilizar o quiz localmente e fazer alterações no código para gerar um relatório persistente ao final.

3.3. Levantamento das Funcionalidades

Para atender à finalidade do trabalho proposto, foi realizado um levantamento das funcionalidades que o quiz deve possuir, visando construir uma base para a escolha da ferramenta a ser utilizada em sua elaboração. As funcionalidades levantadas foram divididas entre funcionalidades do professor e do aluno, sendo elas:

Aluno:

- o Realizar o quiz.
- o Permitir que o aluno veja o resultado da aplicação do quiz.
- Permitir exibir uma sugestão de leitura para uma questão errada.

Professor:

- Permitir que o professor crie um *quiz* para avaliar o aprendizado dos alunos sobre teste de software.
- Permitir que o professor aplique o *quiz* para os alunos do curso.
- Permitir que as respostas ao *quiz* sejam analisadas por meio de um relatório.
- Permitir a manutenção de um banco de questões.
- Analisar os dados salvos da realização do quiz.
- o Permitir acompanhamento da evolução dos alunos.

Tendo em vista as funcionalidades levantadas, foi elaborada a Tabela 1, contendo o sumário de funcionalidades presentes nas ferramentas e aplicativos pesquisados. Das ferramentas analisadas, as únicas que possuíam todas as funcionalidades levantadas foram Kahoot! e Google Forms, considerando que a Kahoot! apenas cobriu todas as funcionalidades no plano pago.

Tabela 1 - Funcionalidades das ferramentas pesquisadas

	QuizMaker	TriviaMaker	Kahoot!	Google Forms	Trivia Quiz	Django School
Criar Perguntas	X	X	X	X	X	X
Criar Quiz	X	X	X	X	X	X
Exibir Sugestão de Leitura após a Pergunta	I X	Não há	Apenas nas versões pagas	X	Não há	Não há
Gerar Relatório	Não há	Não há	X	X	Não há	X
Salvar Perguntas	X	X	X	X	X	X
Salvar Resultado do Quiz	Não há	X	X	X	Não há	X
Acompanhar Aluno	Não há	Não há	X	X	Não há	X

3.3.1. Características Desejáveis

Após o levantamento das funcionalidades presentes, foi preciso estabelecer os parâmetros utilizados como critérios de desempate para a escolha da ferramenta a ser utilizada na criação do *quiz* proposto, tendo em vista o objetivo do *quiz e* seus requisitos. Os critérios foram:

- Portabilidade a possibilidade de ser executado em diferentes plataformas.
- Interface intuitiva, que dispense treinamento ou conhecimento prévio.
- Aderência às funcionalidades especificadas.
- Capacidade de criar um quiz de, ao menos, 30 questões.
- Possibilidade de configuração do tempo de resposta das perguntas ou do *quiz*.

A Tabela 2 apresenta o sumário dos critérios atendidos por cada uma das ferramentas analisadas.

Tabela 2 - Critérios atendidos pelas ferramentas pesquisadas

	QuizMaker	TriviaMaker	Kahoot!	GoogleForms	TriviaQuiz	Django School
Portabilidade (Web+App)	Apenas App	X	X	Apenas Web	Apenas App	Apenas Web
Interface intuitiva	X	X	X	X	X	X
Funcionalidades (Free)	Parcial	Parcial	Parcial	Todas	Parcial	Parcial
Funcionalidades (Premium/Pro)	Não possui	Parcial	Todas	Não possui	Não possui	Não possui
Capacidade de Questões	30+	30+	30+	30+	30+	30+
Temporização	X	Não há	X	X	X	Não há
Persistência de Dados	Não há	Não há	X	X	Não há	X

3.4. Seleção da Ferramenta

Como discutido anteriormente, algumas ferramentas encontradas não atendiam a critérios importantes para o trabalho a ser realizado:

- As ferramentas TriviaMaker, QuizMaker e TriviaQuiz não emitem relatório após a realização do quiz.
- As ferramentas TriviaMaker, TriviaQuiz e Django School não permitem a exibição de uma explicação elaborada das respostas e a ferramenta Kahoot! só o permite nas versões pagas Pro e Plus.

Assim sendo, a ferramenta escolhida para a realização do trabalho foi o Google Forms, por atender todas as funcionalidades. No entanto, para apoiar a totalidade dos requisitos identificados foi necessário criar uma planilha para fazer o tratamento dos resultados exportados e produção de relatórios.

3.5. Considerações Finais

Neste capítulo, foi mostrado o levantamento de requisitos do *quiz*, a análise das ferramentas disponíveis e o levantamento de funcionalidades das ferramentas. No capítulo seguinte, será apresentado o quiz implementado, as ementas de disciplinas pesquisadas, a identificação das questões, e um tutorial de uso do *quiz*.

4. Ferramenta TesterQuiz

Este capítulo apresenta a ferramenta TesterQuiz, desenvolvida no âmbito deste trabalho, seu processo de criação, a escolha da tecnologia utilizada, a identificação das questões a serem colocadas e a implementação do *quiz* em si.

4.1. Objetivo da Implementação

O objetivo deste trabalho é prover uma ferramenta de apoio ao ensino de testes de *software*, que seja de fácil utilização e que possa servir de apoio ao aprendizado, indicando material de estudo complementar ao aluno na área em que seja identificada deficiência de aprendizagem. Ao professor, a ferramenta pode fornecer dados de desempenho dos alunos, além de apontar seus pontos de dificuldade no aprendizado.

4.2. Identificação das Questões

Para elaboração do banco de questões foram utilizadas diversas referências, tentando manter como foco o conteúdo das ementas de disciplinas de cursos de computação (Ciência da Computação, Sistemas de Informação e Engenharia da Computação) e mantendo o trabalho em um contexto de análise do conhecimento acadêmico absorvido pelos alunos. Na totalidade dos cursos pesquisados para a realização do trabalho, foram encontradas disciplinas de engenharia de software e afins que continham testes de *software* em suas respectivas ementas, o que reflete a importância da disciplina.

Os cursos analisados foram Sistemas de Informação (UNIRIO, UFRRJ, UFF e UEMS), Ciência da Computação (UERJ, UFRJ, UFRRJ, UFF, USP e Univap) e Engenharia da Computação (UERJ e UFRJ). Na Tabela 3 é apresentado um levantamento das disciplinas encontradas que possuem testes de software em suas ementas.

Tabela 3 - Disciplinas que possuem Testes de Software na ementa

Universidade	Curso	Disciplina	Fonte
UNIRIO	Bacharelado em Sistemas de Informação	Programação Modular (TIN0121)	http://bsi.uniriotec.br/disciplinas/ementas/progmod.html
		Tópicos Avançados em Engenharia de Software II (TIN0166)	http://bsi.uniriotec.br/disciplinas/ementas/taes2.html
UERJ	Bacharelado em Ciência da Computação	Engenharia de Software (Código: IME04-10830)	http://www.ementario.uerj.br/ement a.php?cdg_disciplina=10830
UERJ	Engenharia da Computação	Princípios de Engenharia de Software (Código: IPRJ-01-10768)	http://www.iprj.uerj.br/images/arquivos/eng-comp/manual/UERJ_IPRJ_EngComp_ManualEstudante_2012.pdf (página 8)
UFRJ	Ciência da Computação	Fundamentos da Engenharia de Software	https://dcc.ufrj.br/ensino/graduacao/disciplinas-computacao-2
UFRJ	Engenharia da Computação	Engenharia de Software (Código: EEL873)	https://siga.ufrj.br/sira/temas/zire/fr ameConsultas.jsp?mainPage=/repos itorio-curriculo/61AD45DD-92A4- F79B-3D87-7A444052DF9B.html
UFRRJ	Ciência da Computação	Engenharia de Software (Código: IM866)	https://www.cc.ufrrj.br/grad- ccomp/disciplinas/im866/
UFRRJ	Sistemas de Informação	Engenharia de Software	http://cursos.ufrrj.br/grad/sistemas/e mentas-e-bibliografias-das- disciplinas/#es
UFF	Sistemas de Informação e Ciência da Computação	Engenharia de Software II (Código: TCC00293)	https://app.uff.br/graduacao/quadro dehorarios#ementa_100000011249
USP	Bacharelado em Ciências de Computação	Engenharia de Software (Código: SSC0130)	https://uspdigital.usp.br/jupiterweb/ obterDisciplina?sgldis=SSC0130&c odcur=55041&codhab=0
UEMS	Sistemas de Informação	Engenharia de Software	http://www.uems.br/dourados
			https://slideplayer.com.br/slide/295 434/
Univap	Ciência da Computação	Engenharia de Software I	https://www1.univap.br/prado/disci plina/engsoft- I/notasdeaula/curso/part1/cap2/cap2 .htm

4.2.1. Análise de Ementas

A análise das ementas encontradas foi limitada ao conteúdo das ementas divulgados pelos respectivos cursos. Visto que o material didático com os tópicos ensinados não era acessível a pessoas externas, somente disponibilizados nos sistemas internos dos respectivos cursos, não foi possível um aprofundamento no conteúdo das ementas divulgadas. Os 3 tópicos escolhidos para nortear a elaboração das questões do *quiz* foram Tipos e Técnicas de Teste (por vezes chamado de Estratégia de Teste), Níveis de Teste e Ciclo de Vida de Teste, abordados no Capítulo 2. O banco de questões se encontra no Apêndice I.

4.3. Implementação

Esta seção tem a finalidade de explicar a implementação e utilização do TesterQuiz por meio de um tutorial de uso com ilustrações das telas. Será mostrada a criação do *quiz* pelo professor, a resposta por parte de um aluno, a geração e a exibição do relatório ao final.

4.3.1. Adicionando questões ao banco de questões do quiz

Para adicionar questões ao banco de questões, o professor deve abrir a planilha 'TesterQuiz - Relatório' (ou uma cópia da mesma) encontrada no endereço https://docs.google.com/spreadsheets/d/1F58 KgzwQi6kPtkCTzp8DsblXjOwmgT3T3M8qpUonC Y//. Com a planilha aberta, é preciso acessar a aba 'Banco' e adicionar uma linha ao final da aba, na parte inferior, como mostra a Figura 7.

59		58	Sobre as relações entre as etapas do Ciclo de Vida				
60		59	As etapa	As etapas do ciclo de vida de testes Ciclo de Vida			
61		60	Afirmar q	ue ao término da	etapa de	Ci	iclo de Vida
62							
Adicionar mais 1 linhas ao fim.							
	+ ≣	Respo	stas 🕶	Acertos ▼	Relatório	o *	Banco ▼

Figura 7 - Parte inferior da aba 'Banco' da planilha 'TesterQuiz - Relatório'

Na linha adicionada, preencher as informações da questão (ver Figuras 8, 9, 10 e 11) da seguinte forma:

- Na coluna B, preencher o ID da próxima questão respeitando a numeração (ver Figura 8);
- Na coluna C, preencher a questão (ver Figura 8);
- Na coluna D, opcionalmente, preencher a fonte e o material adicional (ver Figura 8);
- Na coluna E, preencher o tópico da questão (ver Figura 8);
- Na coluna F, preencher a resposta correta (ver Figura 8);
- Na coluna G, preencher a fonte ou a literatura sugerida (ver Figura 9);
- Nas colunas H, I, J e K, preencher as opções de resposta da questão (ver Figuras 10 e 11).

C •	▶ E	F
Questão	Tópico	Resposta
São considerados testes caixa branca	Técnicas de Teste	no ambiente de desenvolvimento, pelos desenvolvedores
Um sistema precisa ser testado	Técnicas de Teste	Teste de Stress
Uma empresa bancária precisa	Técnicas de Teste	Teste de Carga
Sobre testes do tipo caixa branca,	Técnicas de Teste	Se preocupam apenas com a implementação.
São testes caixa branca que ajudam a	Técnicas de Teste	Testes de Carga, Testes de Desempenho e Testes de
Após a criação de um módulo do	Técnicas de Teste	Testes de Unidade
Testes de fumaça são os testes	Técnicas de Teste	Caixa Branca
Uma equipe de testes precisa realizar	Técnicas de Teste	Teste Caixa Branca
A afirmação "Testes caixa branca são	Técnicas de Teste	Correta.
Sobre testes de caixa preta, afirmar	Técnicas de Teste	Errado
	Questão São considerados testes caixa branca Um sistema precisa ser testado Uma empresa bancária precisa Sobre testes do tipo caixa branca, São testes caixa branca que ajudam a Após a criação de um módulo do Testes de fumaça são os testes Uma equipe de testes precisa realizar A afirmação "Testes caixa branca são	Questão Tópico São considerados testes caixa branca Técnicas de Teste Um sistema precisa ser testado Técnicas de Teste Uma empresa bancária precisa Técnicas de Teste Sobre testes do tipo caixa branca, Técnicas de Teste São testes caixa branca que ajudam a Técnicas de Teste Após a criação de um módulo do Técnicas de Teste Testes de fumaça são os testes Técnicas de Teste Uma equipe de testes precisa realizar Técnicas de Teste A afirmação "Testes caixa branca são Técnicas de Teste

Figura 8 - Colunas 'ID', 'Questão', 'Tópico' e 'Resposta' da aba 'Banco' da planilha 'TesterQuiz - Relatório'

G
Fonte e Material Adicional
https://medium.com/assertqualityassurance/teste-unit%C3%A1rio-e-qualidade-de-software-acce7b9c537
http://mds.cultura.gov.br/core.base_rup/guidances/concepts/performance_testing_37A31809.html
https://www.devmedia.com.br/testes-de-desempenho-carga-e-stress/26546
https://pt.wikipedia.org/wiki/Teste_de_caixa-branca
https://www.devmedia.com.br/testes-de-desempenho-carga-e-stress/26546
http://carlosschults.net/pt/testes-unitarios-iniciantes-parte1
Fonte: https://qualityagile.wordpress.com/tag/teste-de-fumaca/
https://blog.geekhunter.com.br/tecnicas-de-caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-preta-e-branca-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste-de-software/#O_que_e_teste_de_caixa-para-teste_de-software/#O_que_e_teste_de-software/#O

Figura 9 - Coluna 'Fonte e Material Adicional' da aba 'Banco' da planilha 'TesterQuiz - Relatório'

I I
Opção B
no ambiente de desenvolvimento, pelos
Teste de Contenção
Teste de Carga
Tem como objetivo garantir os requisitos funcionais
Testes de Carga, Testes de Regressão e Testes de
Testes de Carga
Caixa Branca
Teste Caixa Preta

Figura 10 - Colunas 'Opção A' e 'Opção B' da aba 'Banco' da planilha 'TesterQuiz - Relatório'

J	К
Opção C	Opção D
no ambiente de homologação, pelos usuários finais.	no ambiente de produção, pelos usuários finais.
Teste de Contingência	Teste de Stress
Teste de Stress	Teste de Regressão
São testes realizados pelo usuário final.	São executados apenas por automação.
Testes de Carga, Testes de Desempenho e Testes de	Testes de Regressão, Testes de Segurança e Testes de
Testes de Unidade	Testes de Aceitação

Figura 11 - Colunas 'Opção C' e 'Opção D' da aba 'Banco' da planilha 'TesterQuiz - Relatório'

4.3.1.1. **Criando o quiz**

O próximo passo é criar um *quiz* com as perguntas desejadas. Para isso, deve-se criar um novo formulário do tipo 'Teste em branco' na página principal do Google Forms, no endereço eletrônico http://forms.google.com/, como mostra a Figura 12.

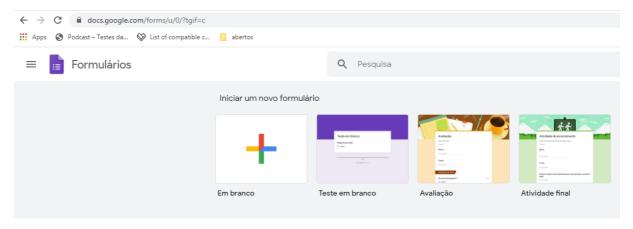


Figura 12 - Tela principal do Google Forms

A seguir, é possível renomear o *quiz* na parte superior central e na parte superior esquerda da página, que aparecem nomeadas como 'Teste em branco' como na Figura 13. Para adicionar questões novas ao banco de questões para serem utilizadas no TesterQuiz, o professor precisa clicar no botão 'Adicionar pergunta', marcado em vermelho na Figura 13.

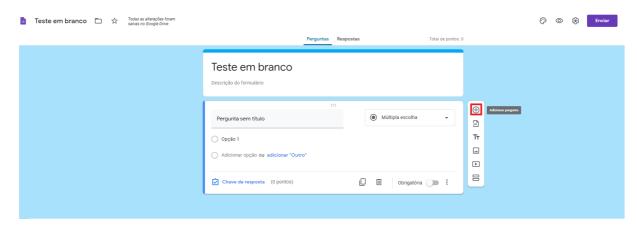


Figura 13 - Tela de criação de um novo formulário com botão 'Adicionar perguntas' em destaque

Na pergunta adicionada, marcar a opção 'Múltipla escolha' na *combobox* do tipo de pergunta, marcar a opção 'Obrigatória' e adicionar as respostas clicando no botão 'Adicionar opção'. Preencher a questão e as respostas como inseridas no banco de questões (ver Seção 4.3.1.1) e clicar no botão 'Chave de resposta', como exibido na Figura 14.

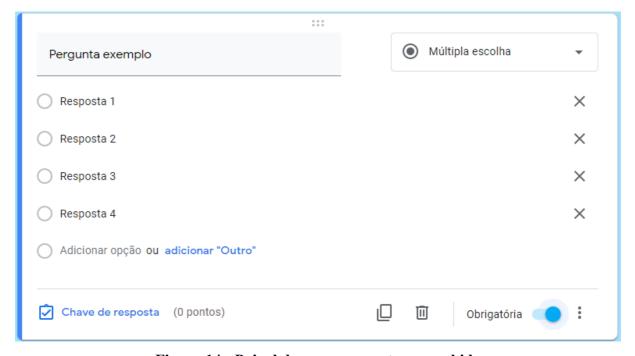


Figura 14 - Painel de nova pergunta preenchido

No painel aberto, escrever a pontuação da questão adicionada, selecionar a resposta correta da questão e clicar em 'Adicionar feedback da resposta', como mostra a Figura 15.

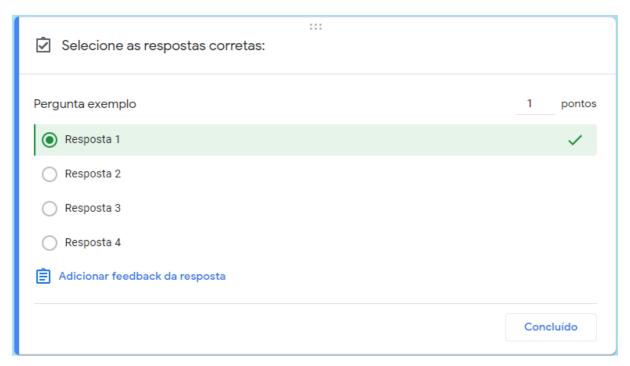


Figura 15 - Painel de resposta da nova pergunta

Na caixa de texto aberta, digitar a sugestão de leitura para a questão e clicar em 'Salvar' como exibido na Figura 16.

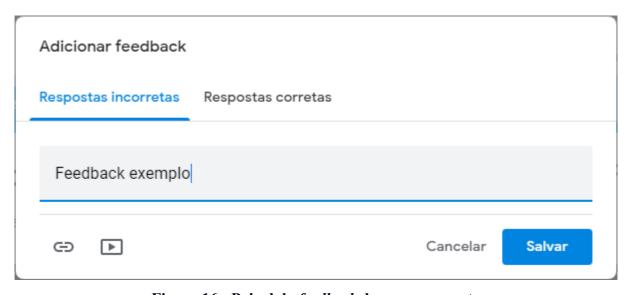


Figura 16 - Painel de feedback da nova pergunta

Com as perguntas já preenchidas, o próximo passo da criação do *quiz* é a exclusão da pergunta vazia que é criada junto ao novo formulário, como mostra a Figura 17, a menos que ela tenha sido utilizada na criação do *quiz*.

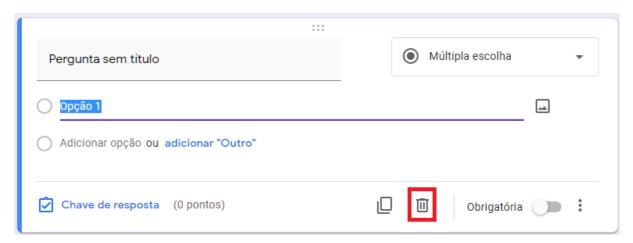


Figura 17 - Pergunta gerada na criação do formulário com botão de exclusão em destaque

Por último, o professor deve clicar no botão de 'Configurações' (ícone de engrenagem no canto superior direito da página do formulário) e marcar as opções 'Coletar endereços de email', 'Limitar a 1 resposta' e a opção 'Recibos de respostas' no modo 'Sempre', como mostra a Figura 18.



Figura 18 - Aba de Configurações

4.3.1.2. Importando perguntas

Para auxiliar o trabalho, foi criado um formulário para ser usado como base para importar perguntas já inseridas no banco, com o objetivo de poupar tempo do professor. Para

importar as questões, o professor precisa selecionar o formulário TesterQuiz, no endereço https://docs.google.com/forms/d/1ikwEGxq5XCdUcLhDtmFzL6gGir4R1fjpwARsC-Z3COo/ e clicar no botão 'Selecionar', como mostra a Figura 19.

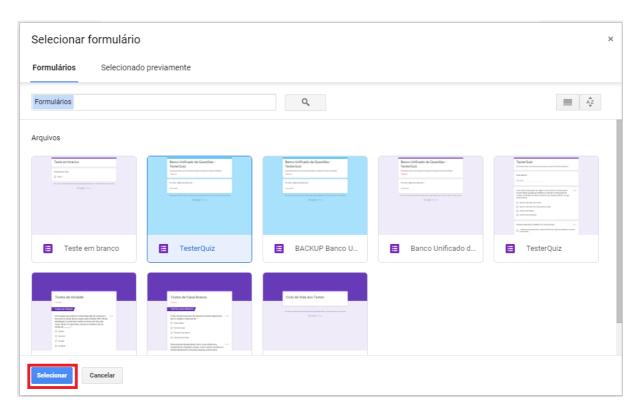


Figura 19 - Tela de seleção do formulário para importação de perguntas

No próximo passo, o professor deve selecionar, na aba de seleção de perguntas, a pergunta obrigatória do campo matrícula e as demais perguntas que deseja colocar no *quiz*, com o cuidado de não importar as seções e títulos, que se fazem presentes no banco de questões apenas para organização dos tópicos. Ao terminar de selecionar as perguntas, como ilustrado na Figura 20, deve clicar no botão 'Importar perguntas' na parte inferior da aba, e as perguntas serão adicionadas ao *quiz* sendo criado.

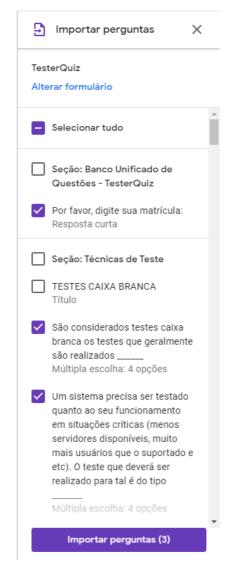


Figura 20 - Aba de seleção de perguntas

4.3.1.3. Enviando o quiz

Para enviar o *quiz*, o professor pode enviar o link do formulário para os alunos, clicando no botão 'Enviar' no canto superior direito, como mostra a Figura 21.



Figura 21 - Botão 'Enviar' do Google Forms

Na janela de envio, o professor pode optar por enviar por e-mail, acrescentando os e-mails dos alunos e clicando em 'Enviar' como mostrado na Figura 22. Alternativamente, o professor pode optar por compartilhar um link com os alunos, clicando na aba com um símbolo de corrente enviando aos alunos o link gerado no campo 'Link' da forma que preferir, como mostra a Figura 23.



Figura 22 - Janela de envio de formulário por e-mail

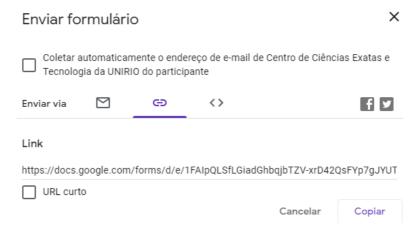


Figura 23 - Janela de envio de formulário por link

4.3.1.4. Respondendo o quiz

Ao receber o *quiz*, o aluno deve preencher a matrícula na primeira pergunta e responder as demais perguntas marcando uma única resposta, como mostrado na Figura 24.

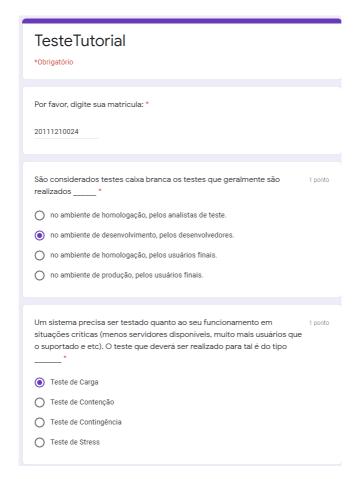


Figura 24 - Tela de preenchimento do quiz

Após preencher todas as questões, o aluno deve clicar no botão 'Enviar', no canto inferior da página, como ilustra a Figura 25.

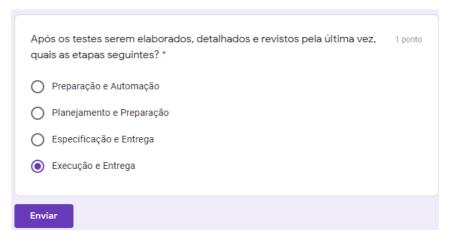


Figura 25 - Canto inferior da página de preenchimento do quiz

Após a mensagem de confirmação de envio das respostas ser exibida, como mostra a Figura 26, o aluno deve clicar no botão 'Ver Pontuação' para ver seu resultado individual.

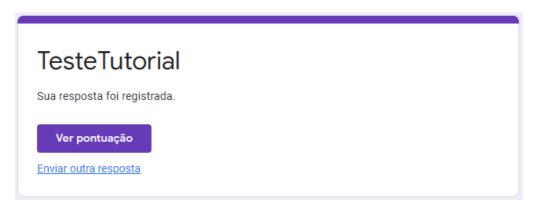


Figura 26 - Mensagem de confirmação de envio com botão 'Ver Pontuação'

Na tela de resultado individual, serão exibidas as questões acertadas e erradas, além de uma sugestão de leitura para as questões erradas, como ilustrado pela Figura 27. Este relatório será disponibilizado aos alunos posteriormente pelo professor (ver Figura 33).

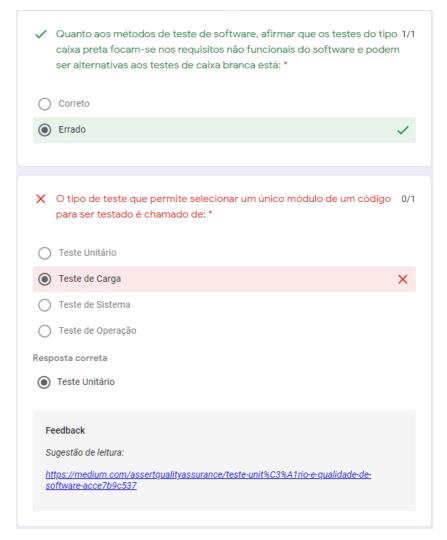


Figura 27 - Tela de resultado individual

4.3.1.5. Visualizando o resultado geral

Ao término da execução do *quiz*, o professor poderá visualizar o resultado geral da turma acessando a página do formulário gerado e clicando na aba 'Respostas', marcada em vermelho na Figura 28.

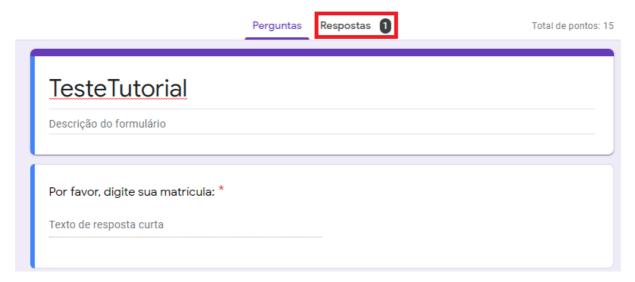


Figura 28 - Tela do formulário com a aba 'Respostas' destacada.

Na aba 'Respostas', serão exibidos dados gerais da execução, como a média (que no gráfico gerado, é chamada de Mediano) e a mediana da turma, além do intervalo de pontos dos alunos, como mostrado na Figura 29.

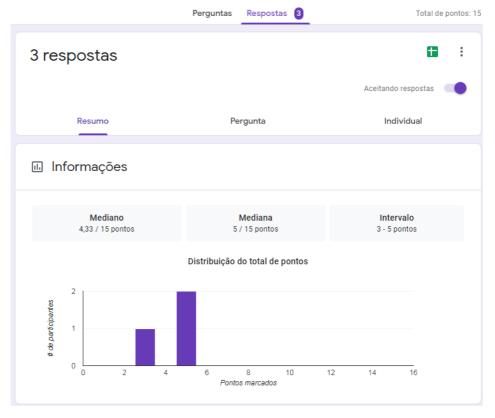


Figura 29 - Aba 'Respostas' com dados gerais da execução do quiz

Mais abaixo, são exibidos dados referentes às respostas assinaladas, como as questões erradas com maior frequência (com taxa de acerto inferior a 50%), como exibido na Figura 30, além da frequência com que os alunos assinalaram cada resposta, como exibido na Figura 31.

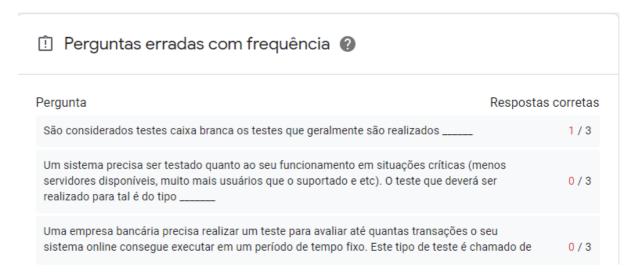


Figura 30 - Seção de perguntas erradas com frequência



Figura 31 - Tela de respostas com a frequência das respostas assinaladas

Na seção 'Pontuações', abaixo da seção 'Perguntas erradas com frequência', o professor deve clicar em 'Liberar Pontuações', como mostra na Figura 32.



Figura 32 - Seção 'Pontuações' com botão 'Liberar pontuações'

Na tela 'Liberar pontuações', deve marcar 'Todos os participantes' e clicar no botão 'Enviar e-mails e liberar', como exibido na Figura 33.

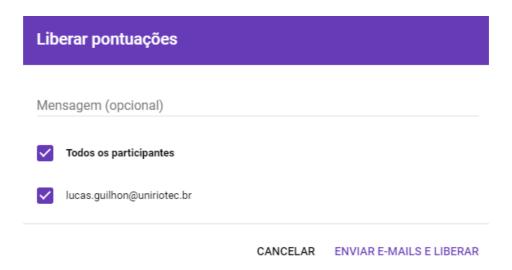


Figura 33 - Tela 'Liberar pontuações'

4.3.1.6. Gerando o relatório

Para gerar o relatório individual do *quiz*, a fim de visualizar o desempenho individual dos alunos, o professor deve abrir o *quiz* aplicado, acessar a aba de 'Respostas' e clicar no botão 'Criar planilha' destacado em vermelho na Figura 34.

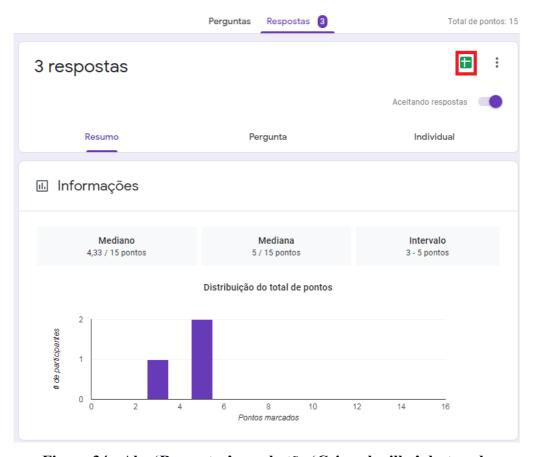


Figura 34 - Aba 'Respostas' com botão 'Criar planilha' destacado

Na janela que se abre, o professor deve selecionar a opção 'Selecionar planilha existente' e clicar no botão 'Selecionar', como mostra a Figura 35.

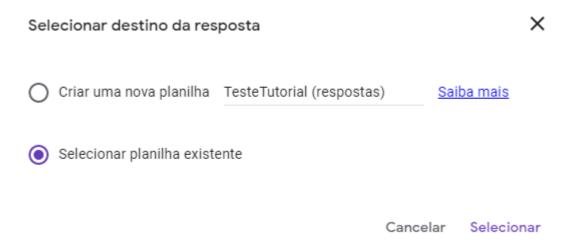


Figura 35 - Janela de seleção do destino da planilha

Na tela de escolha de planilha, o professor deve selecionar a planilha 'TesterQuiz - Relatório', presente no link https://docs.google.com/spreadsheets/d/1F58_KgzwQi6kPtkCTzp8DsblXjOwmgT3T3M8qpUonC Y/edit?usp=sharing ou uma cópia idêntica da planilha e clicar no botão 'Selecionar', como mostra a Figura 36.

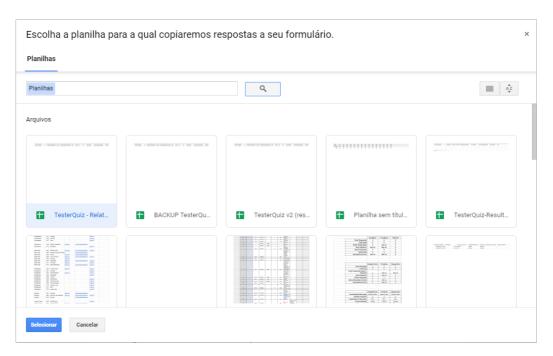


Figura 36 - Tela de escolha da planilha

Após a planilha selecionada ser aberta na nova aba com os resultados da execução, o professor deve selecionar todos os dados da aba, clicando na área entre a marcação de colunas e linhas, marcada em vermelho na Figura 37.

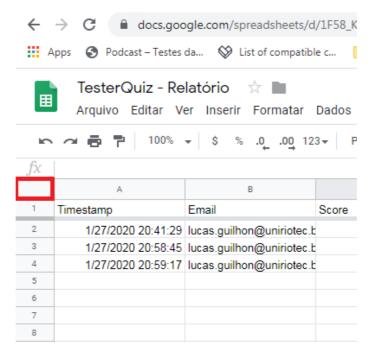


Figura 37 - Aba criada com área de seleção de dados destacada em vermelho

O professor deve copiar os dados, abrir a aba 'Respostas', apagar quaisquer dados que encontre na aba, selecionar a célula 'A1' e colar os dados copiados, como na Figura 38.

fx	Timestamp			
	A	В	С	
1	Timestamp	Email Address	Score	
2	2/2/2020 20:26:04	lucas.guilhon@uniriotec.b		5 / 15
3	2/2/2020 20:26:38	lucas.guilhon@uniriotec.b		7 / 15
4	2/2/2020 20:27:06	lucas.guilhon@uniriotec.b		2 / 15

Figura 38 - Aba 'Respostas'

A aba 'Acertos' do relatório é usada como planilha auxiliar para a geração do relatório e nela constam as respostas marcadas pelos participantes, os tópicos das questões, as respostas corretas e os identificadores das questões como mostra a Figura 39.

f_X	ID			
	A	В	С	D
1	ID	1	2	3
2	Tópico	Técnicas de Tes	Técnicas de Test	Técnicas de Test
3	Questão	São considerado	Um sistema prec	Uma empresa ba
4	Resposta Correta	no ambiente de	Teste de Stress	Teste de Carga
5	20111210024	Errado	Errado	Errado
6	20121210222	Errado	Errado	Certo
7	20132210333	Errado	Errado	Errado
8				
9				
10				

Figura 39 - Aba 'Acertos' da planilha "TesterQuiz - Relatório"

Após este passo, o relatório estará disponível na aba 'Relatório', após a seleção do aluno pela matrícula na lista dinâmica, como mostra a Figura 40. Os gráficos indicam, por aluno selecionado na lista da célula 'B6', os erros e acertos por tópico, os erros e acertos gerais e a proporção de acertos por tópico, em gráficos de pizza e de rosca, para auxiliar a comparação entre os tópicos e para servir de apoio visual.

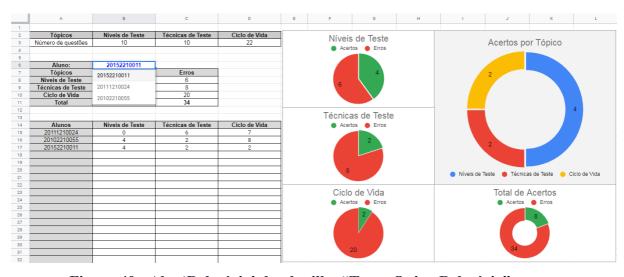


Figura 40 - Aba 'Relatório' da planilha "TesterQuiz - Relatório"

4.4. Considerações Finais

Neste capítulo foram apresentados os aspectos da elaboração e aplicação do TesterQuiz, bem como a análise das ementas de disciplinas usadas como base. O capítulo seguinte será sobre a aplicação e execução do *quiz* para os alunos do curso, bem como a avaliação por parte dos mesmos e do professor.

5. Avaliação do TesterQuiz

Após a aplicação do *quiz*, foi utilizado um formulário para avaliar a possibilidade da ferramenta ser utilizada no contexto acadêmico. Para esse fim, foi utilizado um formulário baseado no TAM (*Technology Acceptance Model*) (DAVIS, 1989) (DAVIS et al., 1989) segundo o paradigma Goal/Question/Metric – GQM (Basili et al., 1994).

Devido ao escopo do trabalho, é importante ressaltar que foi a avaliação do TesterQuiz foi feita com base em um estudo exploratório, sem significância estatística dos dados coletados.

5.1. Planejamento

O planejamento da avaliação consistiu no estabelecimento dos objetivos da avaliação, tendo como base os conceitos básicos de usabilidade e facilidade de uso percebidas. Além destes objetivos, foi adicionado o objetivo da medição de futuro uso da ferramenta. A estrutura mostrada aqui é baseada na apresentada em (BONELLI, 2014). O objetivo principal ficou dividido em 3 objetivos, mostrados na Tabela 4.

Tabela 4 - Objetivos e perguntas da avaliação da ferramenta

G0: Avaliar a facilidade de uso e utilidade do TesterQuiz percebida pelos alunos e pelo professor.					
G1: Avaliar a facilidade de uso da ferramenta.	G2: Avaliar a utilidade percebida da ferramenta.	G3: Avaliar o interesse no uso futuro da ferramenta por parte dos participantes.			
Q1: Durante a aplicação do <i>quiz</i> não tive nenhum problema de compreensão quanto a sua utilização.	Q3: Considero positiva a aplicação do quiz no auxílio do aprendizado de testes de software.	Q8: Considero positiva a aplicação futura do quiz em um futuro semestre.			
Q2: O método de aplicação do <i>quiz</i> foi de fácil compreensão.	Q4: As perguntas do <i>quiz</i> são pertinentes ao aprendizado de testes de software.	Q9: Considero positiva a aplicação do quiz em disciplinas que possuem testes de <i>software</i> em suas ementas.			
	Q5 - O conteúdo coberto pelas perguntas do quiz é adequado.	Q10: Você teria sugestões de melhoria para o <i>quiz</i> aplicado?			
	Q6 - O material adicional indicado é útil.				
	Q7: A tecnologia utilizada ajuda a manter o foco no <i>quiz</i> .				

5.2. Definição da métrica de avaliação

Para responder à avaliação, os alunos devem preencher o formulário gerado com as questões da Tabela 4, utilizando a métrica ilustrada na Tabela 5 para responder às questões de 1 a 9, podendo justificar as respostas que achar necessário em um campo de texto ao final do formulário, como na Figura 52. A questão 10 é uma questão discursiva de caráter subjetivo, com o propósito de permitir ao aluno indicar uma análise mais pessoal.

Tabela 5 - Organização das respostas ao formulário de avaliação da ferramenta

Valor	Opção	Significado	
1	Concordo plenamente	O participante concorda completamente com a afirmação no que diz respeito a facilidade, utilidade e interesse na utilização futura do <i>quiz</i> .	
2	Concordo parcialmente	O participante tem dúvidas, ressalvas ou exceções em relação a afirmação.	
3	Indiferente	O participante não enxerga relevância na afirmação mas também não discorda da mesma.	
4	Discordo parcialmente	O participante discorda da afirmação mas enxerga que ela pode ser válida em outros contextos ou para outras pessoas.	
5	Discordo plenamente	O participante discorda completamente da afirmação no que diz respeito a facilidade, utilidade e interesse na utilização futura do quiz.	

5.3. Execução

Para seleção das perguntas foi consultado o professor ministrante da disciplina Técnicas Avançadas de Software II (TAES-II). O professor optou por selecionar 7 questões de cada tópico, totalizando 21 questões (que podem ser vistas no Apêndice I). Foram elas:

- Tipos e Técnicas de Teste: 3, 5, 6, 12, 14, 15 e 19;
- Níveis de Teste: 21, 25, 26, 27, 29, 31 e 39;
- Ciclo de Vida: 41, 42, 44, 46, 49, 57 e 58.

Os links do TesterQuiz e do formulário de avaliação foram enviados para os 11 alunos inscritos na disciplina, por e-mail com uma breve explicação. Ao final do prazo dado, 6 alunos responderam o *quiz* e, destes, 4 alunos preencheram o formulário de avaliação. Ao professor foi exibido o relatório e enviado o formulário de avaliação. A Figura 41 ilustra a média (Mediano), mediana (Mediana) e o intervalo de pontos entre os alunos (Intervalo).



Figura 41 - Resultados gerais da aplicação do TesterQuiz

Pela análise da Figura, observa-se que 5 dos 6 alunos participantes obtiveram média superior a 6,0.

A Figura 42 ilustra os acertos dos 6 alunos participantes.

Alunos	Níveis de Teste	Tipos e Técnicas de Teste	Ciclo de Vida
20161210019	5	6	3
201512	3	2	2
20161210027	7	6	3
20181210012	5	7	6
20181210013	6	6	2
20471210037	6	7	2

Figura 42 - Resultados individuais por tópico

Pela análise dos dados presentes na Figura 42, sabemos que de sete questões por tópico, a média dos alunos participantes foi a seguinte:

• Tipos e Técnicas de Teste: 5,66

• Níveis de Teste: 5,33

• Ciclo de Vida: 3

Em média, o tópico em que os alunos participantes encontraram maior dificuldade foi Ciclo de Vida. Individualmente, apenas um aluno obteve pontuação menor em Níveis de Teste e outro aluno obteve a pontuação menor em Tipos e Técnicas de Teste, equiparado com Ciclo de Vida.

A Figura 43 exibe as perguntas com menos de 50% de acertos. Um fato que merece destaque é que a Questão 46, sobre Ciclo de Vida, não foi acertada por nenhum participante.

Pergunta	Respostas correta
Um analista de testes, durante a fase de levantamento dos requisitos precisa, para realizar qual etapa do processo da criação de teste?	lelamente, 2 / 6
Após o planejamento, qual a ordem correta das etapas do ciclo de vida de um teste	e? 0/6
Após a execução de um ciclo de testes, percebeu-se que os testes não cobriam tor funcionalidades que foram pedidas. Ao analisar o Plano de Teste, foi constatado q constavam, mas os testes simplesmente não foram elaborados. Em qual parte do isso deveria ter acontecido?	ue elas lá
Sobre as relações entre as etapas do ciclo de vida de um teste, é correto afirmar qu	ue: 1/6

Figura 43 - Lista de perguntas com menos de 50% de acerto

5.4. Avaliação com base no TAM

A Figura 44 ilustra o resultado da avaliação preenchida pelos alunos.

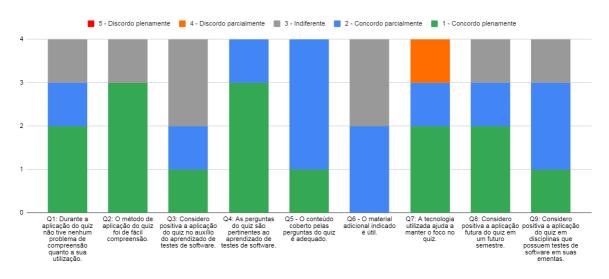


Figura 44 - Resultado do formulário de avaliação dos alunos

- Pela análise dos resultados na Figura 44, constataram-se os seguintes pontos:
- Como ilustrado na avaliação da Questão 1, não foi observado nenhum problema de compreensão na utilização do *quiz* por parte dos alunos: três participantes concordaram plena ou parcialmente e um foi indiferente.
- Pode-se constatar pelos resultados da Questão 2 que o método de aplicação foi, em geral, de fácil compreensão: três participantes concordaram plenamente e um foi indiferente.
- Pela análise da Questão 3, pode-se constatar que a aplicação do *quiz* foi relativamente positiva: dois participantes concordaram parcial ou plenamente e outros dois afirmaram ter sido indiferente.
- Pelas respostas da Questão 4, pode-se constatar que as perguntas do *quiz* foram pertinentes ao aprendizado de testes de software: três participantes concordaram plenamente e um concordou parcialmente.
- No resultado da Questão 5, pode-se constatar que o conteúdo coberto pelas perguntas do quiz é adequado: três participantes concordaram parcialmente e um concordou plenamente.
- Pelas respostas da Questão 6, percebeu-se que o material indicado é relativamente útil: dois participantes concordaram parcialmente e outros dois foram indiferente.
- Pela análise da Questão 7, pode-se afirmar que em termos gerais, a tecnologia utilizada ajudou a manter o foco no *quiz*: três participantes concordaram plena ou parcialmente e um discordou parcialmente.
- Pelo resultado da Questão 8, pode-se inferir que a aplicação do quiz em um futuro semestre seria considerada positiva: três participantes concordaram plena ou parcialmente e um foi indiferente.
- Pode-se constatar pelas respostas da Questão 9, que a aplicação do *quiz* em disciplinas que possuem testes de software em suas ementas foi considerada positiva: três participantes concordaram plena ou parcialmente e um foi indiferente.
- Nas respostas da Questão 10, apenas um participante avaliou que o quiz deveria ser mais curto ou ter perguntas mais objetivas. O espaço opcional disponibilizado para justificativas das questões não foi utilizado por nenhum participante.

Questionado se haveria uma razão para as notas dos alunos não serem mais altas, o professor afirmou que a causa seria falta de estudo e reforçou que o conteúdo coberto pelas questões do *quiz* foi ministrado integralmente ao longo do curso.

A Figura 45 ilustra o resultado da avaliação preenchida pelo professor.



Figura 45 - Resultado do formulário de avaliação do professor

Pela avaliação do professor, podemos constatar que a aplicação do TesterQuiz foi positiva: o professor concordou parcial ou plenamente com as questões 1, 2, 3, 4, 5, 7, 8 e 9.

Na avaliação da Questão 6, o professor avaliou como indiferente a utilidade do material adicional indicado.

Na Questão 10, o professor sugeriu que, ao final do *quiz*, o participante pudesse ter acesso a algum tipo de avaliação que indicasse os pontos de melhoria, ou seja, em que tópicos ele deveria focar os seus estudos para cobrir as falhas de conhecimento identificadas.

No espaço opcional disponibilizado para justificativa das questões, o professor afirmou que avaliou a Questão 6 como "Indiferente" porque não viu o material adicional que é apresentado aos alunos.

5.5. Considerações Finais

Neste capítulo foi apresentada a execução e subsequente avaliação do TesterQuiz por parte dos alunos, a avaliação por parte do professor e a análise dos resultados das avaliações. No capítulo seguinte serão apresentadas as conclusões e trabalhos futuros.

6. CONCLUSÃO

Este capítulo reúne as considerações finais, as limitações, assinala as contribuições do trabalho e sugere possibilidades de aprofundamento posterior.

6.1. Considerações Finais

O objetivo deste trabalho foi desenvolver um *quiz* sobre testes de software, visando auxiliar o aprendizado dos alunos que cursam disciplinas que abordem testes de *software*. Como resultado da aplicação do quiz, objetiva-se, também, indicar os tópicos onde os alunos possuem mais dificuldade e indicar sugestão de leitura complementar. Para a criação do TesterQuiz foi utilizada a infraestrutura do Google Forms e uma planilha que gerou um relatório da execução do *quiz*, com acertos por tópico dos alunos participantes.

6.2. Contribuições

Com a criação do TesterQuiz, professores e alunos possuem mais um recurso didático nas disciplinas que abordam testes de *software*, dando mais visibilidade a lacunas no aprendizado, aplicando o TesterQuiz e utilizando os dados gerados pelo relatório para identificar em quais tópicos se encontram as dificuldades dos alunos cursando a disciplina.

Dentre as contribuições do trabalho pode-se destacar:

- A criação do próprio TesterQuiz, incluindo as instruções para uso do Google Forms e também da planilha auxiliar criada;
- A disponibilização do banco de questões, com 60 questões cobrindo os tópicos Tipos e
 Técnicas de Teste, Níveis de Teste e Ciclo de Vida.

6.3. Limitações e Trabalhos Futuros

Para trabalhos futuros, é necessário conhecer as limitações do TesterQuiz. Dentre as limitações podemos destacar o escopo das questões ser limitada a Testes de Software e aos tópicos Tipos e Técnicas de Software, Níveis de Teste e Ciclo de Vida. Outra limitação é o número de questões (60) que, apesar de ser grande, não esgota os assuntos abordados.

Existem várias possibilidades de continuidade deste trabalho. Uma possível evolução seria armazenar as execuções por turma e projetar a evolução dos resultados por semestres para avaliar se eventuais mudanças no ensino da disciplina tiveram efeito positivo no desempenho dos alunos ou se diminuiu lacunas no conhecimento sobre algum dos tópicos.

Outra possível evolução seria utilizar extensões do Google Forms para temporizar a execução, permitindo a aplicação do TesterQuiz como prova ou simulado de prova à distância na disciplina ministrada.

Outros possíveis trabalhos de continuidade seriam os de ampliar o escopo do TesterQuiz para englobar mais tópicos para identificar lacunas em um contexto de ensino maior, acrescentar mais questões ou aproveitar a estrutura do TesterQuiz para aplicar um *quiz* em outro assunto ou disciplina.

Outro trabalho de continuidade possível seria disponibilizar ao aluno o relatório para indicar os pontos de melhoria, ou seja, em que tópicos ele deveria focar os seus estudos para cobrir as falhas de conhecimento identificadas.

Por fim, outras avaliações do TesterQuiz com mais turmas e alunos devem ser realizadas para avaliar melhor sua aceitação e aplicabilidade.

7. REFERÊNCIAS

Basili, V.R., Caldiera, G., and Rombach, H.D., 1994, Goal Question Metric Paradigm, Encyclopedia of Software Engineering, JJ Marciniak:

Bernardo, P. C. e Kon, F. A Importância dos Testes Automatizados. Engenharia de Software Magazine, 1(3), pp. 54-57. 2008.

Bonelli, S., (2014) "Uma abordagem para registro de indicadores de medição em organizações de desenvolvimento de software", Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro - UNIRIO, Rio de Janeiro, Brasil.

Cassettari, F. T. Estudo de caso: uso de um quiz game para revisão de conhecimentos em gerenciamento de projetos. Dissertação (Dissertação em Sistemas de Informação) – UFSC. Rio de Janeiro, 2015.

Cunha, J. A. O., Marques, G. A., Lemos, W. L., Câmara Jr, U. D., & Vasconcellos, F. J. (2018). Software engineering education in Brazil: a mapping study. In Proceedings of the XXXII Brazilian Symposium on Software Engineering (pp. 348-356). ACM.

Davis, F.D., 1989, Perceived usefulness, perceived ease of use, and user acceptance of information technology: MIS Quarterly, v. 13, no. 3, p. 319–340.

Davis, F.D., Bagozzi, R.P., and Warshaw, P.R., 1989, User Acceptance of Computer Technology: a Comparison of Two Theoretical Models *: Management, v. 35, no. 8, p. 982–1003.

Eliza, R. e Lagares, V., Processo de Teste de Software. Disponível em: https://www.devmedia.com.br/processo-de-teste-de-software/23795. Acesso em: 10 de jan. 2020

Farias, Ester., Ciclo de Vida do Teste de Software. Disponível em: https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/. Acesso em: 10 de jan. 2020

Lemos, Wellington; Cunha, José; Saraiva, Juliana. Ensino de Engenharia de Software em um Curso de Sistemas de Informação: Uma Análise dos Problemas e Soluções na Perspectiva de Professores e Alunos. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27., 2019, Belém. Anais do XXVII Workshop sobre Educação em Computação. Porto Alegre:

Sociedade Brasileira de Computação, july 2019 . p. 305-318. ISSN 2595-6175. DOI: https://doi.org/10.5753/wei.2019.6638.

Pontes, M.B., Introdução a testes de software, Engenharia de Software Magazine, pp. 50-54. 2009.

Pressman, Roger S., Engenharia de Software - Uma Abordagem Profissional - 7ª Edição (2011), AMGH Editora Ltda.

Rocha, F.G., A importância dos testes para a qualidade do software. Disponível em: https://www.devmedia.com.br/a-importancia-dos-testes-para-a-qualidade-do-software/28439. Acesso em: 10 de jan. 2020

Sande, Denise; Sande, Danilo. USO DO KAHOOT COMO FERRAMENTA DE AVALIAÇÃO E ENSINO-APRENDIZAGEM NO ENSINO DE MICROBIOLOGIA INDUSTRIAL. HOLOS, [S.l.], v. 1, p. 170-179, fev. 2018. ISSN 1807-1600. Disponível em: http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/6300>. Acesso em: 14 fev. 2020. doi:https://doi.org/10.15628/holos.2018.6300.

Sommerville, Ian., Engenharia de Software - 9ª Edição (2011), Editora Pearson.

Apêndice I - Banco de Questões do TesterQuiz

As perguntas registradas no banco de questões do TesterQuiz podem ser vistas na Tabela I.1.

Tabela I.1 - Perguntas do banco de questões do TesterQuiz

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
1	Técnica s de Teste	Os testes caixa branca geralmente são realizados	no ambiente de homologação, pelos analistas de teste. no ambiente de desenvolvimento, pelos desenvolvedores. no ambiente de homologação, pelos usuários finais. no ambiente de produção, pelos usuários finais.	no ambiente de desenvolvimento , pelos desenvolvedores	https://medium.com/assertqualityassurance/teste-unit%C3%A1rio-e-qualidade-desoftware-acce7b9c537 e http://www.nti.ufpb.br/~caroline/curso/Aula01-Curso%20de%20Testes%20de%20Software%20-%20NTI.pdf (slide 20)
2	Técnica s de Teste	Um sistema precisa ser testado quanto ao seu funcionamento em situações críticas (menos servidores disponíveis, muito mais usuários que o suportado e etc). O teste que deverá ser realizado para tal é do tipo	Teste de Carga Teste de Contenção Teste de Contingência Teste de Stress	Teste de Stress	http://mds.cultura.gov.br/core.base_rup/guid_ances/concepts/performance_testing_37A31_809.html
3	Técnica s de Teste	Uma empresa bancária precisa realizar um teste para avaliar até quantas transações o seu sistema online consegue executar em um período de tempo fixo. Este tipo de teste é chamado de	Teste de Contingência Teste de Carga Teste de Stress	Teste de Carga	https://www.devmedia.com.br/testes-dedesempenho-carga-e-stress/26546

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			Teste de Regressão		
4	Técnica s de Teste	Sobre testes do tipo caixa branca, assinale a opção verdadeira:	São projetados a partir da implementação. Tem como objetivo garantir os requisitos funcionais levantados durante a modelagem do sistema. São testes realizados pelo usuário final. Nunca são executados por automação.	São projetados a partir da implementação.	https://pt.wikipedia.org/wiki/Teste_de_caix_a-branca
5	Técnica s de Teste	São testes que ajudam a analisar se determinada infraestrutura de hardware é adequada para um sistema:	Testes de Segurança, Testes de Controle e Testes de Conformidade Testes de Carga, Testes de Regressão e Testes de Stress Testes de Carga, Testes de Desempenho e Testes de Stress Testes de Regressão, Testes de Segurança e Testes de Contingência	Testes de Carga, Testes de Desempenho e Testes de Stress	https://www.devmedia.com.br/testes-dedesempenho-carga-e-stress/26546
6	Técnica s de Teste	Após a criação de um módulo do sistema, qual tipo de teste deve ser executado, primeiramente, pelo programador?	Testes de Integração Testes de Carga Testes de Unidade Testes de Aceitação	Testes de Unidade	http://carlosschults.net/pt/testes-unitarios- iniciantes-parte1
7	Técnica s de Teste	Testes de fumaça são os testes mínimos que procuram validar a parte central do negócio de um produto. Dos testes de	Caixa Preta Caixa Branca	Caixa Branca	Fonte: https://qualityagile.wordpress.com/tag/teste- de-fumaca/

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
		fumaça existentes, os Testes de Regressão NÃO são testes do tipo			Material adicional: http://jkolb.com.br/teste-fumaca/
8	Técnica s de Teste	Uma equipe de testes precisa realizar testes que buscam avaliar um sistema a partir da sua implementação. Este tipo de teste é chamado de	Teste Caixa Branca Teste Caixa Preta	Teste Caixa Branca	https://blog.geekhunter.com.br/tecnicas-de- caixa-preta-e-branca-para-teste-de- software/#O que e teste de caixa branca
9	Técnica s de Teste	A afirmação "Testes caixa branca são testes baseados em um exame rigoroso do detalhe procedimental. Caminhos lógicos e colaborações entre componentes são testadas." é:	Correta Errada	Correta.	Fonte: http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 47)
10	Técnica s de Teste	Sobre testes de caixa preta, afirmar que: "São testes realizados a partir dos casos de teste especificados durante a fase de modelagem do sistema, que necessitam acesso ao código fonte do projeto" seria:	Correto Errado	Errado	https://pt.wikipedia.org/wiki/Teste_de_caix a-preta
11	Técnica s de Teste	Dentre as técnicas de teste, assinale a alternativa correta:	testes de caixa branca somente são realizados após os testes de integração. testes de caixa branca e preta tem como objetivo encontrar os mesmos tipos de defeitos no sistema. testes de caixa preta podem avaliar o funcionamento da interface com o usuário. testes de caixa branca não são executados após a execução de testes de caixa preta.	testes de caixa preta podem avaliar o funcionamento da interface com o usuário.	https://pt.wikipedia.org/wiki/Teste_de_caix a-preta

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
12	Técnica s de Teste	Quanto aos métodos de teste de software, afirmar que os testes do tipo caixa preta validam os requisitos não funcionais do software e podem ser executados no lugar dos testes de caixa branca está:	Correto Errado	Errado	http://qualidade-de- software.blogspot.com/2010/01/teste-de- caixa-preta.html
13	Técnica s de Teste	Com a entrega de uma parte codificada de um projeto a um analista de testes, o mesmo executará testes do tipo caixa preta com o propósito de	validar o código-fonte do projeto. analisar as estruturas de dados do sistema. encontrar funcionalidades incompletas ou que não funcionem corretamente. testar os módulos individuais de um sistema.	encontrar funcionalidades incompletas ou que não funcionem corretamente.	http://qualidade-de-software.blogspot.com/2010/01/teste-de-caixa-preta.html
14	Técnica s de Teste	Um sistema apresentou diversos tipos de problema na versão colocada no ambiente de produção. A opção que corretamente relaciona o tipo de problema com o tipo de teste que visa evitá-lo é:	O sistema não aguentou um alto número de conexões simultâneas e teve grande queda de performance. (Testes de Requisitos) Os usuários gastam muito tempo procurando entender como funciona o sistema. (Testes de Usabilidade) Uma nova versão do sistema apresentou defeito em uma funcionalidade que funcionava normalmente na anterior (Testes de Stress) Um usuário conseguiu realizar um ataque de DDoS e derrubar o serviço (Testes de Stress)	Os usuários gastam muito tempo procurando entender como funciona o sistema. (Testes de Usabilidade)	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 27)

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
15	Técnica s de Teste	Dentre os tipos de teste de software, o que realiza o reteste, visando verificar se uma alteração recente causou	Operação Controle	Regressão	https://www.devmedia.com.br/teste-de- regressao/23038
		defeitos em funcionalidades que já estavam funcionando é o teste de	Regressão Contingência		
16	Técnica s de Teste	Visando poupar tempo de teste, uma empresa irá utilizar um software de integração contínua para executar testes automatizados, buscando validar se funcionalidades já implementadas e testadas continuam funcionando. Os testes descritos são do tipo	Teste de Continuidade Teste de Regressão Teste de Contingência Teste de Unidade	Teste de Regressão	https://www.devmedia.com.br/teste-de-regressao/23038
17	Níveis de Teste	Assinale a alternativa correta: "Os testes de aceitação podem ser"	Ou automáticos ou manuais (mutuamente exclusivos) Apenas automáticos Apenas manuais Automáticos ou manuais	Automáticos ou manuais	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 28)
18	Técnica s de Teste	A técnica de teste que visa encontrar funções incorretas ou omitidas, erros de interface com o usuário, erros de comportamento ou desempenho, erros de iniciação e erros de término, é chamada de teste de	Caixa Preta Caixa Branca	Caixa Preta	http://qualidade-de- software.blogspot.com/2010/01/teste-de- caixa-preta.html
19	Técnica s de Teste	Qual o tipo de teste que reflete, de certa forma, a óptica do usuário, que está interessado em se servir do programa sem considerar os detalhes de sua	Teste Funcional Teste Não Funcional	Teste Funcional	http://qualidade-de- software.blogspot.com/2010/01/teste-de- caixa-preta.html

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
		construção?			
20	Técnica s de Teste	Tentar cadastrar valores inválidos, realizar login com credenciais inválidas e não preencher campos obrigatórios, são testes realizados pelos testes	Não funcionais Funcionais	Funcionais	http://qualidade-de- software.blogspot.com/2010/01/teste-de- caixa-preta.html
21	Níveis de Teste	Uma equipe que adota a metodologia ágil vai começar a escrever os testes de um projeto pelo método TDD. Nessa abordagem, os primeiros testes a serem escritos, que visam testar as subrotinas, classes e métodos são os testes de	Sistema Operação Unidade Aceitação	Unidade	https://pt.wikipedia.org/wiki/Teste_de_soft ware#Fases_ou_N%C3%ADveis
22	Níveis de Teste	O tipo de teste que permite selecionar um único módulo de um código para ser testado é chamado de:	Teste de Unidade Teste de Carga Teste de Sistema Teste de Operação	Teste de Unidade	https://medium.com/assertqualityassurance/teste-unit%C3%A1rio-e-qualidade-desoftware-acce7b9c537
23	Níveis de Teste	Quando uma nova funcionalidade é implementada, antes mesmo dos testes realizados pelo analista de teste, os primeiros testes a serem realizados pelo desenvolvedor, são os testes	de Unidade de Integração de Atualização de Regressão	de Unidade	http://carlosschults.net/pt/testes-unitarios- iniciantes-parte1
24	Níveis de Teste	Os testes que são realizados pelo Desenvolvedor, de maneira automática, que verificam a corretude das classes	de Atualização de Regressão	de Unidade	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
		individualmente são testes	de Unidade de Integração		%20-%20NTI.pdf (slide 32)
25	Níveis de Teste	Em "Testes unitários precisam ser automatizados e auxiliam na refatoração de código", a afirmação está:	Errada Certa	Certa	http://www.dsc.ufcg.edu.br/~jacques/cursos/apoo/html/impl/impl3.htm
26	Níveis de Teste	Marque a alternativa que corretamente define um nível de teste:	Teste de Sistema - Nesta fase, o objetivo é verificar o sistema em execução com seus módulos já integrados e testados, varrendo o código-fonte a procura de falhas e inconsistências. Teste de Regressão - Consiste em testar parte já desenvolvidas anteriormente, visando observar se os defeitos descobertos anteriormente foram corrigidos na nova versão. Teste de Aceitação - Testes que buscam confirmar internamente a implementação das funções esperadas. Teste de Integração - É a fase em que os módulos do software em questão são combinados e testados em um contexto da interação entre eles. Antecede a fase em que o sistema já integrado é testado num ambiente que simula o ambiente de produção.	Teste de Integração - É a fase em que os módulos do software em questão são combinados e testados em um contexto da interação entre eles. Antecede a fase em que o sistema já integrado é testado num ambiente que simula o ambiente de produção.	https://pt.wikipedia.org/wiki/Teste_de_integ_ra%C3%A7%C3%A3o

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
27	Níveis de Teste	Durante o desenvolvimento de um software de cadastro e consulta de multas de trânsito usando metodologia ágil, uma equipe pretende projetar testes a fim de encontrar possíveis falhas provenientes da interação entre os módulos de multas cadastradas e de condutores cadastrados. Assinale corretamente o nível dos testes a serem escritos:	Unidade Integração Sistema Regressão	Integração	https://www.devmedia.com.br/teste-de- integracao-na-pratica/31877
28	Níveis de Teste	Para testar as dependências entre os componentes de um software, devemos realizar testes de	Aceitação Unidade Sistema Integração	Integração	https://www.devmedia.com.br/teste-de- integracao-na-pratica/31877
29	Níveis de Teste	Além dos testes unitários, qual outro nível de testes é executado pelos desenvolvedores?	Sistema Integração Aceitação Regressão	Integração	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
30	Níveis de Teste	Afirmar que na metodologia TDD (Test-Driven Development), testes de integração são os últimos a serem executados pelos próprios desenvolvedores está:	Certo Errado	Certo	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
31	Níveis de Teste	Um software precisa ser testado, pela primeira vez, sob ponto de vista de seu usuário final, tendo como foco o	Integração Regressão	Sistema	https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/testes-de-release_v01.pdf

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
		comportamento do sistema em si, sem se preocupar com a parte interna, já testada. Esses testes a serem realizados são do nível de	Operação Sistema		
32	Níveis de Teste	Em relação aos níveis de teste e suas execuções, os testes de	aceitação são testes que executam as funções principais de um sistema, antes do produto final, para validá-las com o cliente. sistema devem ser executado em um ambiente o mais próximo possível do ambiente no qual o usuário utilizará o sistema. unidade são testes que necessitam serem executados após o desenvolvimento de todos os módulos. regressão necessitam acesso a versões anteriores do projeto.	sistema devem ser executado em um ambiente o mais próximo possível do ambiente no qual o usuário utilizará o sistema.	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
33	Níveis de Teste	Testes que podem ser executados em interfaces com o usuário reais ou simuladas são os testes de qual nível?	Unidade Sistema Regressão Aceitação	Sistema	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
34	Níveis de Teste	A afirmação: "O teste de sistema não necessita de interface, visto que as mesmas foram todas testadas pelos testes de Integração" está	Correta Errada	Errada	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
35	Níveis de Teste	Além dos testes de aceitação, qual outro nível que tem como objetivo testar o software como um todo e não apenas parte dele?	Testes de Unidade Testes de Integração Testes de Carga Testes de Sistema	Testes de Sistema	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
36	Níveis de Teste	Testes que envolvem o sistema como um todo e são realizados pelos usuários finais, são chamados de:	Testes de Sistema Testes de Aceitação Testes de Regressão Testes de Operação	Testes de Aceitação	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
37	Níveis de Teste	Testes que necessitam de interfaces com o usuário reais são os testes de qual nível?	Unidade Sistema Regressão Aceitação	Aceitação	http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slide 20)
38	Níveis de Teste	Uma equipe de desenvolvedores deseja garantir, junto ao cliente, que a qualidade das funcionalidades do software entregue esteja em conformidade com o que foi pedido. Para isso, devem realizar testes de	Integração / são testes que validam o comportamento externo e interno do sistema. Aceitação / tem como objetivo simular a utilização do sistema pelo usuário. Unidade / validam cada módulo das funcionalidades desejadas com o usuário, garantindo o funcionamento do sistema como um todo.	Aceitação / tem como objetivo simular a utilização do sistema pelo usuário.	http://shipit.resultadosdigitais.com.br/blog/5 -boas-praticas-para-se-aplicar-em-testes-de-aceitacao e https://medium.com/bionexo/teste-de-aceita%C3%A7%C3%A3o-uma-documenta%C3%A7%C3%A3o-objetiva-%C3%BAtil-e-execut%C3%A1vel-67675bafe96a/

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			Sistema / pois consistem dos testes do código já desenvolvido e os testes de interface combinados.		
39	Níveis de Teste	A MELHOR definição do objetivo do teste de aceitação é:	Garantir que o software entre sem erros na produção. Garantir que o grupo de testes fez um bom trabalho. Executar um teste funcional. Garantir que o software esteja fazendo exatamente aquilo que o usuário precisa.	Garantir que o software esteja fazendo exatamente aquilo que o usuário precisa.	Fonte: http://www.nti.ufpb.br/~caroline/curso/Aula 01- Curso%20de%20Testes%20de%20Software %20-%20NTI.pdf (slides 48 e 27)
40	Níveis de Teste	A fase de teste que busca validar se um software está pronto para ser utilizado pelos usuários e desempenhar as funções e tarefas que os usuários demandam é a fase dos testes de	Contingência Carga Aceitação Fumaça	Aceitação	http://mds.cultura.gov.br/core.base_rup/guid ances/concepts/acceptance_testing_12A0F1 52.html
41	Ciclo de Vida	Um analista de testes, durante a fase de levantamento dos requisitos precisa, paralelamente, realizar qual etapa do processo da criação de teste?	Especificação Preparação Planejamento Execução	Planejamento	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/ e http://analistadetestes.blogspot.com/2013/05/ciclo-de-vida-dos-testes-e-do.html
42	Ciclo de Vida	A configuração do ambiente de testes se dará na etapa de	Planejamento Especificação	Preparação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			Execução Preparação		e http://analistadetestes.blogspot.com/2013/05 /ciclo-de-vida-dos-testes-e-do.html
43	Ciclo de Vida	Uma equipe de desenvolvimento acaba de liberar os módulos a serem testados por um analista de testes. A etapa que depende desta disponibilidade dos módulos a serem testados é a(o) de testes.	Preparação Especificação Planejamento Execução	Especificação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/ e http://analistadetestes.blogspot.com/2013/05/ciclo-de-vida-dos-testes-e-do.html
44	Ciclo de Vida	Em qual fase se deve registrar as evidências de um teste?	Preparação Execução Especificação Planejamento	Execução	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/ e http://analistadetestes.blogspot.com/2013/05/ciclo-de-vida-dos-testes-e-do.html
45	Ciclo de Vida	Um analista de testes encontra-se no momento de relatar as ocorrências encontradas a fim de melhorar o processo de testes para os próximos ciclos de desenvolvimento. Em qual etapa o processo se encontra?	Preparação Execução Especificação Entrega	Entrega	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/ e http://analistadetestes.blogspot.com/2013/05/ciclo-de-vida-dos-testes-e-do.html
46	Ciclo de Vida	Após o planejamento, qual a ordem correta das etapas do ciclo de vida de um teste?	Preparação - Entrega - Especificação - Execução	Preparação - Especificação - Execução -	https://www.devmedia.com.br/processo-de- teste-de-software/23795

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			Especificação - Preparação - Entrega - Execução Especificação - Preparação - Execução - Entrega Preparação - Especificação - Execução - Entrega	Entrega	e http://analistadetestes.blogspot.com/2013/05 /ciclo-de-vida-dos-testes-e-do.html
47	Ciclo de Vida	A etapa do ciclo de vida de um teste de software na qual se configura o Hardware é chamada:	Configuração Preparação Planejamento Entrega	Preparação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
48	Ciclo de Vida	A etapa do ciclo de vida de um teste de software da qual se sai com a Estratégia de Testes pronta é chamada:	Entrega Planejamento Configuração Preparação	Planejamento	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
49	Ciclo de Vida	Após a execução de um ciclo de testes, percebeu-se que os testes não cobriam todas as funcionalidades que constavam no Plano de Teste. Os testes simplesmente não foram elaborados. Em qual parte do ciclo de vida isso deveria ter acontecido?	Especificação Planejamento Configuração Preparação	Especificação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
50	Ciclo de Vida	Após os testes serem elaborados, detalhados e revistos pela última vez, quais as etapas seguintes?	Preparação e Automação Planejamento e Preparação	Execução e Entrega	http://analistadetestes.blogspot.com/2013/05/ciclo-de-vida-dos-testes-e-do.html

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			Especificação e Entrega Execução e Entrega		
51	Ciclo de Vida	A configuração das ferramentas a serem utilizadas nos testes é realizada na etapa de:	Execução Planejamento Especificação Preparação	Preparação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
52	Ciclo de Vida	Após a etapa na qual o Plano de Testes é elaborado, qual etapa deve ser realizada:	Especificação Preparação Execução Validação	Preparação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
53	Ciclo de Vida	A documentação de um projeto foi liberada para uma equipe de testes e os mesmos se preparam para elaborar os casos de teste e escrever/programar os scripts de teste. Essa etapa é chamada de:	Preparação Execução Validação Especificação	Especificação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
54	Ciclo de Vida	Após qual etapa do ciclo de vida de um teste é que podemos rever as ocorrências a fim de melhorar o processo para os próximos ciclos?	Execução Entrega Validação Especificação	Entrega	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
55	Ciclo de Vida	Com o Plano de Testes e a Estratégia de Testes prontos, qual as próximas etapas no ciclo de vida de um teste de software, na ordem correta?	Especificação - Preparação - Execução - Entrega Preparação - Especificação - Execução - Entrega Preparação - Execução - Especificação - Entrega Especificação - Execução - Preparação - Entrega	Preparação - Especificação - Execução - Entrega	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
56	Ciclo de Vida	Após especificar os testes, a equipe de testes precisa aguardar o desenvolvimento das funcionalidades para realizar a de testes, coletar evidências e registrar defeitos encontrados. Assinale a alternativa que preenche corretamente a lacuna:	Preparação Execução Entrega Planejamento	Execução	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
57	Ciclo de Vida	Em qual etapa ocorre a verificação sobre os requisitos levantados do software?	Planejamento Preparação Execução Entrega	Planejamento	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
58	Ciclo de Vida	Sobre as relações entre as etapas do ciclo de vida de um teste, é correto afirmar que:	A especificação cuida da elaboração de novos testes e/ou revisão dos antigos testes que passarão pela execução. A preparação consiste em elaborar os testes que serão executados. O planejamento consiste em configurar	A especificação cuida da elaboração de novos testes e/ou revisão dos antigos testes que passarão pela execução.	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/

ID	Tópico	Questão	Opções	Resposta	Fonte e Material Adicional
			ferramentas e ambientes utilizados na execução do teste. Durante a especificação se busca encontrar os defeitos no software que serão corrigidos na execução.		
59	Ciclo de Vida	As etapas do ciclo de vida de testes que antecedem a execução ocorrem em qual ordem?	Preparação - Especificação - Planejamento Especificação - Planejamento - Preparação Planejamento - Preparação - Especificação Especificação - Preparação - Planejamento	Planejamento - Preparação - Especificação	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/
60	Ciclo de Vida	Afirmar que ao término da etapa de Preparação no ciclo de vida de teste de software, todo hardware e software que será utilizado deve estar configurado é:	Correto Errado	Correto	https://artedotestedesoftware.wordpress.com/2014/03/05/ciclo-de-vida-do-teste-de-software/