INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS –
CÂMPUS DE URUAÇU
DEPARTAMENTO DE ÁREAS ACADÊMICAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

KATIANE DE SOUZA SANTOS

LEVANTAMENTO E ANÁLISE DOS MÉTODOS DE TESTAGEM DE SOFTWARE UTILIZADOS POR EMPRESAS DESENVOLVEDORAS DE SOFTWARE EM URUAÇU-GO E REGIÃO

URUAÇU
2023

INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[X] TCC - Graduação	[ ] Trabalho Apresentado em Evento

[ ] Produto Técnico e Educacional - Tipo: _____

Nome Completo do Autor: Katiane de Souza Santos
Matrícula: 20161050100243
Título do Trabalho: LEVANTAMENTO E ANÁLISE DOS MÉTODOS DE TESTAGEM DE SOFTWARE UTILIZADOS POR EMPRESAS DESENVOLVEDORAS DE SOFTWARE EM URUAÇU-GO E REGIÃO

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ___/___/_____ (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

( ) O documento está sujeito a registro de patente.
( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Uruaçu, 10/07/2023.
Local e Data

Katiane de Souza Santos
Assinatura do Autor e/ou Detentor dos Direitos Autorais

KATIANE DE SOUZA SANTOS

LEVANTAMENTO E ANÁLISE DOS MÉTODOS DE TESTAGEM DE SOFTWARE UTILIZADOS POR EMPRESAS DESENVOLVEDORAS DE SOFTWARE EM URUAÇU-GO E REGIÃO

Projeto de pesquisa apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas ao Departamento de Áreas Acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás – *Câmpus* Uruaçu.
Orientador: Prof. Me. Alexandre Martins Ferreira Bueno

URUAÇU
2023

Dados Internacionais de Catalogação na Publicação (CIP)

Santos, Katiane de Souza.

S237l Levantamento e análise dos métodos de testagem de software utilizados por empresas desenvolvedoras de software em Uruaçu-GO e região [manuscrito] / Katiane de Souza Santos. - 2023.

XLVIII, 48 f.: il.

Orientador: Prof. Me. Alexandre Martins Ferreira Bueno.

Trabalho de Conclusão de Curso (Tecnólogo) - Instituto Federal de Educação, Ciência e Tecnologia de Goiás: Campus Uruaçu, Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2023.

Bibliografia.

Inclui lista de figuras, lista de quadros, lista de tabelas e lista de siglas.

1. Software - testes. 2. Software - desenvolvimento. 3. Informatização. 4. Padrões de software. I. Bueno, Alexandre Martins Ferreira (orientador). II. Instituto Federal de Educação, Ciência e Tecnologia de Goiás. III. Título.

CDD: 004.6

Ficha catalográfica elaborada pela Bibliotecária Sabrina Gisele da Silva Felix CRB1/2561

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS URUACU

Katiane de Souza Santos

LEVANTAMENTO E ANÁLISE DOS MÉTODOS DE TESTAGEM DE SOFTWARE UTILIZADOS POR EMPRESAS DESENVOLVEDORAS DE SOFTWARE EM URUAÇU-GO E REGIÃO

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas ao Departamento de Áreas Acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Campus Uruaçu.

Aprovado em 20 de junho de 2023.

(Assinado eletronicamente)
Prof. Me. Alexandre Martins Ferreira Bueno
Instituto Federal de Goiás - Campus Uruaçu - Orientador

(Assinado eletronicamente)
Prof. Me. Davi Taveira Alencar Alarcao
Instituto Federal de Goiás - Campus Uruaçu

(Assinado eletronicamente)
Profa. Ma. Thiane Marques Torquato
Instituto Federal de Goiás - Campus Uruaçu

Documento assinado eletronicamente por:
- **Davi Taveira Alencar Alarcao**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 30/06/2023 22:32:33.
- **Thiane Marques Torquato**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 30/06/2023 20:14:20.
- **Alexandre Martins Ferreira Bueno**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 30/06/2023 16:27:01.

Este documento foi emitido pelo SUAP em 30/06/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifg.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 426071
Código de Autenticação: 2cbc2ae54b

Dedico este trabalho ao meu pai Benedito Cardoso dos Santos, à minha mãe Iracema Bispo de Souza Santos, à minha irmã Kátia de Souza Santos e ao meu orientador Me. Alexandre Martins Ferreira Bueno.

AGRADECIMENTOS

Agradeço primeiramente à Deus, que permitiu que tudo acontecesse ao longo da minha vida (e não somente nestes anos como universitária), e que é o maior Mestre que alguém pode conhecer. Agradeço à minha família, por acreditarem e investirem em mim. À minha mãe, pelo seu cuidado e dedicação. Você foi, em alguns momentos, a minha esperança para seguir. Ao meu pai, a sua presença significou segurança e a certeza de que não estou sozinha nessa caminhada. À minha irmã que sempre esteve presente e me auxiliou na minha formação. Aos meus amigos e colegas, pelo incentivo e pelo apoio constante. Agradeço ao prof. Me. Alexandre Martins pela paciência durante a orientação e pelo incentivo que tornou possível a conclusão deste TCC. A todos os professores do Curso, que foram tão importantes na minha vida acadêmica e no desenvolvimento deste trabalho. Às pessoas com quem convivi no IFG-Câmpus Uruaçu ao longo desses anos. A experiência de uma produção compartilhada com amigos nesse Câmpus foi a melhor experiência da minha formação acadêmica. Agradeço aos egressos e concluintes do Curso que disponibilizaram seu tempo para serem entrevistados, permitindo que este trabalho fosse realizado. Por fim, agradeço a todos que diretamente ou indiretamente fizeram parte da minha formação, o meu muito obrigada!

“Deus nunca disse que a jornada seria fácil, mas Ele disse que a chegada valeria a pena.”

Max Lucado

RESUMO

Este trabalho de pesquisa propôs investigar quais eram as principais técnicas de testagem de software utilizadas por profissionais desenvolvedores de software em Uruaçu-GO e região. Ele também se propôs a entender como funcionavam os processos de desenvolvimento de software nos quais esses profissionais estavam envolvidos, além de aspectos envolvendo gestão/qualidade de software. Para isso, foram realizadas entrevistas com seis profissionais que atuam em Uruaçu. Cada entrevista seguiu um roteiro definido, representado por um formulário (composto por 77 questões). A partir da coleta, tabulação e análise dos dados, observou-se que cada empresa possuía um processo de desenvolvimento único, organizado em equipes pequenas e que faziam uso de métodos ágeis para o desenvolvimento de software em incrementos. Observou-se que as equipes possuíam profissionais dedicados à gestão de projetos e de garantia da qualidade, e que vários tipos de teste eram empregados pelos desenvolvedores, como os testes de sistema, de release e de usuário. Também foram identificados pontos de melhoria nos processos que merecem atenção, como a adoção de forma ampla de testes unitários automatizados e maior participação dos usuários durante as etapas de desenvolvimento dos softwares. A partir dessas análises, entendeu-se que os principais objetivos desejados com esse trabalho foram alcançados. O planejamento e execução desse trabalho de pesquisa foi desafiador. Muito esforço foi necessário para a elaboração do formulário, sua aplicação, tabulação e análise dos dados.

Palavras-chave: teste de software, processo de desenvolvimento de software, qualidade de software, gestão de software.

ABSTRACT

This research work proposed to investigate what were the main software testing techniques used by professional software developers in Uruaçu-GO and region. It also proposed to understand how the software development processes in which these professionals were involved worked, in addition to aspects involving software management/quality. For this, interviews were conducted with six professionals who work in Uruaçu. Each interview followed a defined script, represented by a form (composed by 77 questions). From the collection, tabulation and analysis of the data, it was observed that each company had a unique development process, organized in small teams and that made use of agile methods for the development of software in increments. It was observed that the teams had professionals dedicated to project management and quality assurance, and that various types of tests were used by developers, such as system, release and user tests. Points of improvement were also identified in the processes and deserve attention, such as the widespread adoption of automated unit tests and a greater user participation during the software development stages. From these analyses, it was understood that the main objectives desired with this work were achieved. The planning and execution of this research work was challenging. A lot of effort was required in the elaboration of the form, its application, tabulation and data analysis.

Keywords: software testing, software development process, software quality, software management.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ADS – Análise e Desenvolvimento de Sistemas.

CTO – *Chief Technology Officer* – Chefe de Operação de Tecnologia.

QA – *Quality Assurance* – Garantia da Qualidade.

SUMÁRIO

1 INTRODUÇÃO

A demanda por softwares e sistemas de melhor qualidade é uma realidade a nível mundial. Usuário e empresam cobram cada vez mais softwares que sejam úteis, seguros e que agreguem valor. Existe uma cobrança por softwares com cada vez mais funcionalidades e que sejam entregues dentro de cronogramas “apertados”.

Sabe-se que dentro do processo de construção de um software, a testagem dos seus artefatos desempenha um papel essencial, colaborando para que possíveis erros/inconsistências sejam encontrados antes da entrega do software aos seus clientes. Isto permite com que estes defeitos sejam “consertados” e que o software entregue aos clientes possua maior garantia quanto à sua qualidade.

Sabe-se também que um software pode ser testado nas diversas fases do seu desenvolvimento e que muitas são as técnicas de desenvolvimento que podem ser utilizadas em cada uma destas fases.

Devido à importância deste tema e à pouca informação relativa disponível, este trabalho propõe coletar e organizar aspectos relativos à testagem de software realizada por empresas/profissionais em Uruaçu-GO e região.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Diante do que já foi apresentado, este trabalho propõe investigar quais são os principais métodos/técnicas de testagem de software utilizados por empresas e profissionais desenvolvedores de software em Uruaçu e região.

1.1.2 Objetivos específicos

A partir do objetivo geral apresentado, este trabalho possui os seguintes objetivos específicos:

- Identificar os métodos e procedimentos utilizados por empresas e profissionais para se testar software durante o seu processo de desenvolvimento;

- Identificar outros aspectos utilizados relativos à gestão/qualidade de software;
- Realizar a análise dos dados coletados e evidenciar os principais pontos encontrados;
- Identificar possíveis lacunas e propor melhorias nos processos/métodos encontrados.

1.2 JUSTIFICATIVA

A preocupação com a qualidade de sistemas de software cresceu à medida que o software passou a se tornar cada vez mais integrado em cada aspecto da vida cotidiana. Testar um software desde o início do seu processo de construção/desenvolvimento é uma etapa essencial para garantir a sua qualidade. Diante desta importância, empresas desenvolvedoras e profissionais fazem uso dos testes nos seus processos de desenvolvimento.

Vale destacar que devido à grande variedade de tipos de testes existentes e de softwares que podem ser construídos, em geral, as empresas e desenvolvedores customizam as práticas de teste de software de acordo com a sua necessidade/realidade.

Em uma tentativa por parte da autora deste trabalho em investigar quais eram os principais métodos de testagem de software empregados por estas empresas e profissionais na cidade de Uruaçu-GO e região, pouco material foi encontrado. Não foi possível identificar uma “visão geral” sobre como os processos de testagem estão sendo empregados.

Diante desta lacuna, este trabalho propõe a coleta de dados e análise dos principais processos de testagem de software empregados em Uruaçu e região. Espera-se que, a partir desta análise, estes processos sejam identificados, detalhados e documentados.

2 REFERENCIAL TEÓRICO

Neste capítulo são detalhados os tipos de software existentes, além de alguns aspectos envolvendo qualidade de software.

2.1 QUALIDADE DE SOFTWARE

Esta seção foi escrita tendo como referência Sommerville (2011).

Os problemas com a qualidade de software foram inicialmente descobertos na década de 1960 com o desenvolvimento do primeiro grande sistema de software e continuaram a incomodar a engenharia de software ao longo do século XX. O software entregue era lento e pouco confiável, difícil de manter e de reusar. Em resposta à insatisfação com aquela situação, passaram a ser adotadas técnicas formais de gerenciamento de qualidade do software, as quais foram desenvolvidas a partir dos métodos usados na indústria manufatureira. Essas técnicas de gerenciamento de qualidade, em conjunto com novas tecnologias de software e melhores testes de software, conduziram a melhorias significativas no nível geral de qualidade de software.

Na indústria de software, são adotados processos e padrões que visam reforçar que a qualidade de software seja atingida (esse conjunto de boas práticas está inserido no contexto de uma subárea do gerenciamento da qualidade denominada Garantia da Qualidade de Software).

Segundo Presmann (2016), o controle de qualidade de software engloba um conjunto de ações de engenharia de software que ajudam a garantir que cada produto resultante atinja suas metas de qualidade. Os modelos são revistos de modo a garantir que sejam completos e consistentes. O código poderia ser inspecionado de modo a revelar e corrigir erros antes de os testes começarem. Aplica-se uma série de etapas de teste para descobrir erros na lógica de processamento, na manipulação de dados e na comunicação da interface. Uma combinação de medições e realimentação (*feedback*) permite a uma equipe de software ajustar o processo quando qualquer um desses produtos resultantes deixe de atender às metas estabelecidas para a qualidade.

Devido às características peculiares envolvidas no processo de desenvolvimento de software, a avaliação da qualidade de software é um processo

subjetivo, em que uma equipe precisa usar seu julgamento para decidir se foi alcançado um nível aceitável de qualidade. Ela precisa considerar se o software é adequado para sua finalidade ou não. Trata-se de responder a perguntas sobre as características do sistema, tais como: durante o processo de desenvolvimento os padrões de programação e documentação foram seguidos? O software foi devidamente testado? O software é suficientemente confiável para ser colocado em uso? O desempenho do software é aceitável para uso normal? O software é útil? O software é bem estruturado e compreensível?.

Um pressuposto do gerenciamento de qualidade de software é que a qualidade do software é diretamente relacionada à qualidade do processo de desenvolvimento de software. Não há dúvida de que o processo de desenvolvimento usado tenha uma influência significativa sobre a qualidade de software e que bons processos são mais suscetíveis de conduzir o software de boa qualidade. O gerenciamento e a melhoria de qualidade de processo podem gerar softwares com menos defeitos.

Uma importante etapa em um processo de desenvolvimento de software são os mecanismos de testes que esse processo utiliza. A próxima seção detalha os principais tipos de teste de software existentes e que são utilizados nos processos de desenvolvimento.

2.2 TESTES DE SOFTWARE

Esta seção foi escrita tendo como referência Sommerville (2011).

O teste 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.

Segundo Presmann (2016), o objetivo do teste de software é descobrir erros. O teste de software absorve a maior porcentagem do esforço técnico em um processo de software. Independentemente do tipo de software criado, uma estratégia para planejamento sistemático de teste, execução e controle começa considerando pequenos elementos do software e se encaminha para fora no sentido de abranger o programa como um todo.

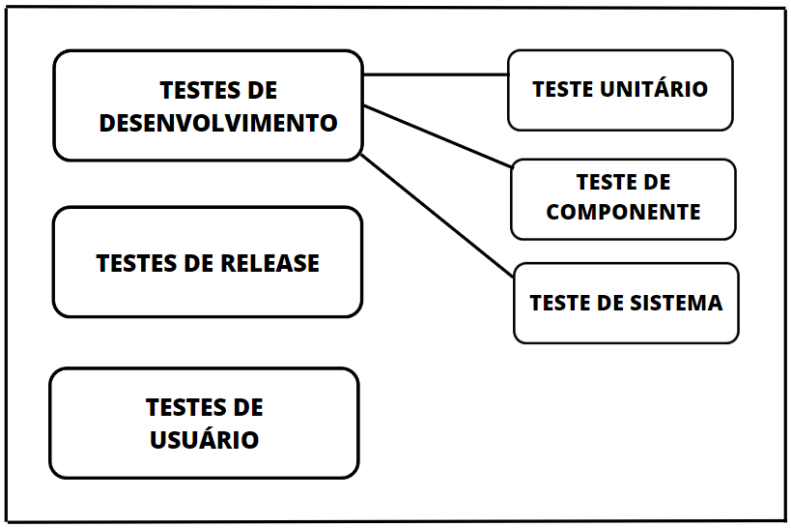
O processo de teste tem dois objetivos distintos: demonstrar ao desenvolvedor e ao cliente que o software atende a seus requisitos (testes de validação) e descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma diferente das especificações (testes de defeitos).

Os testes não podem demonstrar se o software é livre de defeitos ou se ele se comportará conforme especificado em qualquer situação. É sempre possível que um teste que você tenha esquecido seja aquele que poderia descobrir mais problemas no sistema. Os testes podem mostrar apenas a presença de erros, e não sua ausência.

O teste é parte de um amplo processo denominado Verificação e Validação. Este processo também inclui inspeções e revisões. Estas técnicas analisam e verificam os requisitos de sistema, modelos de projeto, o código-fonte de programa e até mesmo os testes de sistema propostos. Essas são chamadas técnicas “estáticas” de verificação e validação, em que você não precisa executar o software para verificá-lo (ao contrário dos testes de software, que ocorrem a partir da execução do software ou de partes dele).

Em geral, um software passa por três estágios de teste (ilustrados na Figura 1): testes em desenvolvimento, em que o sistema é testado durante o desenvolvimento para descobrir bugs e defeitos; testes de release, em que uma equipe de teste independente testa uma versão completa do sistema antes que ele seja liberado para os usuários; e testes de usuário, em que os usuários ou potenciais usuários de um sistema testam o sistema em seu próprio ambiente.

Figura 1 - Tipos de teste de software



Fonte: Próprio autor (2023).

As seções a seguir descrevem melhor cada um destes tipos de testes.

2.3 TESTES DE DESENVOLVIMENTO

Os Testes de desenvolvimento incluem todas as atividades de testes que são realizadas pela equipe de desenvolvimento do sistema. O testador do software geralmente é o programador que o desenvolveu. Este tipo de teste é essencialmente um processo de teste de defeitos, em que o objetivo do teste é descobrir *bugs* no software.

Durante o desenvolvimento de um software, o teste pode ocorrer em três níveis: teste unitário, teste de componentes e teste de sistema. Estes tipos são descritos nas seções a seguir.

2.3.1 Teste unitário

O teste unitário é o processo de testar os pequenos componentes de um programa, como métodos ou classes de objetos. As funções individuais ou métodos são o tipo mais simples de componente de um software. Testes unitários são programados chamando esses métodos com parâmetros diferentes de entrada e observando as saídas apresentadas.

Quando você está testando as classes de objetos, deve projetar os testes para fornecer uma cobertura de todas as características do objeto. Isso significa que você deve: testar todas as operações associadas ao objeto; definir e verificar o valor de todos os atributos associados ao objeto; e colocar o objeto em todos os estados possíveis, o que significa simular todos os eventos que causam mudanças de estado. Sempre que possível, deve-se automatizar os testes unitários. Em testes unitários automatizados, pode-se usar um framework de automação de teste (como JUnit) para escrever e executar testes de seu programa. Frameworks de testes unitários fornecem classes de teste genéricas que você pode estender para criar casos de teste específicos. Eles podem, então, executar todos os testes implementados e informar sobre o sucesso ou o fracasso dos testes.

Testes são custosos e demorados, por isso é importante a escolha casos efetivos de teste unitário. A depender da complexidade do software, ele pode ser centenas de testes unitários implementados.

Existem muitas estratégias que podem ser eficazes para auxiliar na escolha de casos de teste, como: teste de partição, em que se identifica os grupos de entradas que possuem características comuns e devem ser tratados da mesma maneira (escolhe-se, então, os testes dentro de cada um desses grupos); e testes baseados em diretrizes, em que são usadas diretrizes de testes para escolher casos de teste (essas diretrizes refletem a experiência anterior dos tipos de erros que os programadores cometem frequentemente no desenvolvimento de componentes).

Outras diretrizes mais gerais que podem ser utilizadas para a criação de casos de testes são: escolha entradas que forcem o sistema a gerar todas as mensagens de erro; projete entradas que causem *overflow* de *buffers* de entrada; repita a mesma entrada ou uma série de entradas inúmeras vezes; obrigue a geração de saídas inválidas; e obrigue os resultados de cálculos a serem muito grandes ou muito pequenos.

2.3.2 Teste de componente

Os componentes do software são compostos de diversos objetos que interagem. Testes de componentes centram-se em mostrar que a interface de um componente se comporta de acordo com sua especificação (assume-se que os testes unitários sobre os objetos individuais dentro do componente já foram concluídos). Os casos de teste nestes casos não são aplicados aos componentes individuais/objetos, mas sim à interface de componente criada pela combinação desses componentes. Erros de interface no componente podem não ser detectáveis por meio de testes em objetos individuais, pois esses erros resultam de interações entre os objetos do componente.

Podem existir diferentes tipos de interface entre os componentes de um programa o que pode ocasionar na ocorrência de diferentes tipos de erros de interface. Entre eles estão: interfaces de parâmetro, interfaces de memória compartilhada, interfaces de procedimento e interface de passagem de mensagem.

Geralmente, erros de interface são classificados em três classes:

- Mau uso de interface - Um componente chamador chama outro componente e comete um erro no uso de sua interface. Esse tipo de erro é comum com interfaces de parâmetro, em que os parâmetros podem ser de tipo errado

ou ser passados na ordem errada, ou o número errado de parâmetros pode ser passado;

- Mau-entendimento de interface - Um componente chamador desconhece a especificação da interface do componente chamado e faz suposições sobre seu comportamento. O componente chamado não se comporta conforme o esperado, causando um comportamento inesperado no componente de chamada;
- Erros de *timing* - Eles ocorrem em sistemas em tempo real que usam uma memória compartilhada ou uma interface de passagem de mensagens. O produtor e o consumidor de dados podem operar em velocidades diferentes podendo, por exemplo, o consumidor acessar uma informação desatualizada, porque o produtor da informação não atualizou as informações da interface compartilhada.

2.3.3 Teste de sistema

O teste de sistema, durante o desenvolvimento, envolve a integração de componentes para criação de uma versão do sistema e, em seguida, o teste do sistema integrado. O teste de sistema verifica se os componentes são compatíveis, se interagem corretamente e transferem os dados certos no momento certo, por suas interfaces.

Nesse estágio, componentes desenvolvidos por diferentes membros da equipe ou grupos podem ser integrados. O teste de sistema é um processo coletivo, não individual. Em algumas empresas, o teste de sistema pode envolver uma equipe independente, sem participação de projetistas e programadores.

Quando se integra componentes para criar um sistema, obtém-se um comportamento emergente. Isso significa que alguns elementos da funcionalidade do sistema só se tornam evidentes quando colocados juntos. Esse comportamento emergente pode ser planejado e precisa ser testado. É preciso desenvolver testes que verifiquem se o sistema está fazendo apenas o que ele supostamente deve fazer. Esses testes devem descobrir *bugs* de componente que só são revelados quando um componente é usado por outros componentes do sistema.

Por causa de seu foco na interação, o teste baseado em caso de uso é uma abordagem eficaz para testes de sistema. Normalmente, cada caso de uso é implementado por vários componentes ou objetos do sistema. Testar os casos de uso força essas interações a ocorrerem. Se foi desenvolvido um diagrama de sequência para modelar a implementação dos casos de uso, nele poderão ver os objetos ou componentes envolvidos na interação (podendo-se identificar também quais são as entradas necessárias e que saídas são criadas).

Para a maioria dos sistemas, é difícil saber o quanto o teste de sistema é essencial e quando você deve parar de testar. É impossível fazer testes exaustivos, em que cada sequência possível de execução do programa seja testada. Assim, os testes precisam ser baseados em um subconjunto possível de casos de teste. Como alternativa, os testes podem basear-se na experiência de uso do sistema e centrar-se em testes de características como, por exemplo: todas as funções do sistema acessadas por meio de menus devem ser testadas; combinações de funções (por exemplo, a formatação do texto) acessadas por meio de menu devem ser testadas; e, nos casos em que a entrada do usuário é fornecida, todas as funções devem ser testadas com entradas corretas e incorretas.

2.4 TESTES DE RELEASE

O Teste de release é o processo de testar um release/versão particular de um sistema que se destina para uso fora da equipe de desenvolvimento. Geralmente, o release de sistema é testado e liberado para uso dos clientes e usuários.

Existem duas diferenças importantes entre o teste de release e o teste de sistema durante o processo de desenvolvimento: uma equipe separada, que não esteve envolvida no desenvolvimento do sistema, deve ser responsável pelo teste de release; e testes de sistema feitos pela equipe de desenvolvimento devem centrar-se na descoberta de *bugs* no sistema (teste de defeitos), já o objetivo do teste de release é verificar se o sistema atende a seus requisitos e é bom o suficiente para uso externo (teste de validação) e, se assim for, o sistema poderá ser lançado como um produto ou entregue aos clientes.

Portanto, o teste de release precisa mostrar que o sistema oferece a funcionalidade, o desempenho e a confiança especificados e que não falhará durante o uso normal. Ele costuma ser um processo de teste de caixa-preta, no qual os testes

são derivados da especificação de sistema. O sistema é tratado como uma caixa-preta cujo comportamento só pode ser determinado por meio do estudo das entradas e saídas relacionadas (não havendo preocupação sobre como o sistema funciona internamente).

Entre as abordagens para criação de testes de release estão:

- Testes baseados em requisitos – em geral, os requisitos de um sistema são testáveis. Um testador pode, então, verificar se cada requisito foi satisfeito, projetando casos de teste em que se considera cada requisito e, então, deriva um conjunto de testes para ele. Testes baseados em requisitos são mais uma validação do que um teste de defeitos (está se tentando demonstrar que o sistema implementou adequadamente seus requisitos);
- Testes de cenário – nesta abordagem são considerados cenários típicos de uso para o sistema e, então, eles são usados para o desenvolvimento de casos de teste para o sistema. Um cenário é uma história que descreve uma maneira de usar o sistema. Cenários devem ser realistas e abrangentes, e usuários reais do sistema devem ser capazes de se relacionar com eles. Na abordagem por cenários, normalmente testa-se vários requisitos dentro do mesmo cenário;
- Testes de desempenho – estes testes precisam ser projetados para assegurar que o sistema possa processar a carga a que se destina. Isso normalmente envolve a execução de uma série de testes em que se aumenta a carga até que o desempenho do sistema se torne inaceitável (“estressando” o sistema com demandas que estejam fora dos limites de projeto do software). Estes testes estão interessados tanto em demonstrar que o sistema atende seus requisitos quanto em descobrir problemas e defeitos do sistema decorrentes de cargas elevadas.

2.5 TESTES DE USUÁRIO

Teste de usuário ou de cliente é um estágio no processo de teste em que os usuários ou clientes fornecem entradas e conselhos sobre o teste que fizeram com o sistema. Isso pode envolver o teste formal de um sistema que foi testado por um fornecedor externo ou processo informal em que os usuários experimentam um

produto de software novo para ver se gostam e verificar se faz o que eles precisam. O teste de usuário é essencial, mesmo em sistemas abrangentes ou quando testes de release tenham sido realizados. A razão para isso é que as influências do ambiente de trabalho do usuário têm um efeito importante sobre a confiabilidade, o desempenho, a usabilidade e a robustez de um sistema.

Usualmente, existem três tipos de testes de usuário:

- Teste alfa – em que os usuários do software trabalham com a equipe de desenvolvimento para testar o software no local do desenvolvedor, ou seja, este teste é realizado em um ambiente controlado pelo desenvolvedor. Este tipo de teste geralmente é feito em softwares do tipo de prateleira;
- Teste beta – em que um release do software é disponibilizado aos usuários para que possam experimentar e levantar os problemas que eles descobriram com os desenvolvedores do sistema. Este teste é realizado no ambiente do usuário, com o desenvolvedor coletando o *feedback* do usuário. Geralmente, este tipo de teste é feito em softwares do tipo de prateleira e realizado após a conclusão dos testes alfa;
- Teste de aceitação – em que os clientes testam um sistema para decidir se ele está ou não pronto para ser aceito e implantado no ambiente do cliente. Ele é parte inerente ao processo de desenvolvimento de sistemas customizados (ao contrário dos softwares de prateleira, onde normalmente este tipo de teste não é feito), ocorrendo após o teste de release. Engloba o teste formal de um sistema pelo cliente para decidir se ele deve ou não ser aceito. A aceitação designa que o pagamento pelo sistema deve ser feito.

3 MÉTODOS E TÉCNICAS

Utilizando o acervo bibliográfico do IFG e materiais da Internet, foram estudas e compreendidas técnicas que auxiliaram o desenvolvimento do projeto. A partir da leitura de livros de Engenharia de Software, Metodologia Científica e pesquisas na Internet, foram identificados e documentados os principais métodos e ferramentas de teste de software existentes, dentre outros aspectos de qualidade de software e de pesquisa científica importantes.

A partir desta identificação, criou-se um formulário a ser aplicado durante entrevistas realizadas com profissionais desenvolvedores de software. Este formulário investiga como se dá o processo de desenvolvimento e de testagem de software, através da análise de aspectos como, por exemplo, equipe, tipo de processo de desenvolvimento e de gestão utilizados, os principais testes utilizados e fases em que eles são aplicados, e ferramentas auxiliares para testagem/gestão utilizadas.

A etapa de coleta de dados foi feita através da aplicação de entrevistas online e do formulário criado. Com esta etapa concluída passou-se, então, para a tabulação e análise dos dados, onde buscou-se tirar conclusões sobre como se dava o processo de desenvolvimento de software nas empresas, quais eram os principais métodos de testagem utilizados e as lacunas que poderiam ser preenchidas.

Neste trabalho, optou-se pelo preenchimento do formulário criado durante as entrevistas (e não de forma assíncrona) devido à grande variedade de tipos de processos de desenvolvimento de software, de tipos de gestão e de testes de software que podem ser utilizados/adaptados. Certas nuances podem ser melhor detectadas e tratadas durante as entrevistas.

Diante do apresentado, percebe-se que este trabalho propôs uma investigação de caráter descritivo e de levantamento de dados. Segundo Wazlawick 2014, a pesquisa descritiva busca obter dados mais consistentes sobre determinada realidade, ela tenta descrever os fatos como eles são e a pesquisa de levantamento consiste em buscar os dados diretamente no ambiente onde eles estão. Ambos os métodos de pesquisa podem fazer uso de técnicas como observações, entrevistas e questionários/formulários para realizar a coleta de dados.

Uma entrevista é um encontro geralmente entre duas pessoas, um entrevistado e um entrevistador, com a finalidade de que o entrevistador obtenha informações a respeito de determinado assunto ou problema, mediante uma conversação de

natureza profissional. É um procedimento muito utilizado para investigação e coleta de dados, auxiliando no diagnóstico ou tratamento de um problema (LAKATOS, MARCONI; 2006).

Neste trabalho foi utilizada uma entrevista do tipo padronizada, na qual o entrevistador seguiu o formulário previamente elaborado para a coleta de dados. Segundo Lakatos e Marconi (2006), uma entrevista padronizada ou estruturada é aquela em que o entrevistador segue um roteiro previamente estabelecido, onde as perguntas feitas ao entrevistado são predeterminadas. Ela se realiza de acordo com um formulário elaborado. O motivo da padronização é obter dos entrevistados respostas às mesmas perguntas, permitindo que todas elas sejam comparadas com o mesmo conjunto de perguntas.

Para o desenvolvimento desta pesquisa foi utilizado um computador pessoal próprio, com sistema operacional Windows 10 e acesso à Internet. A partir desse equipamento foi utilizada a ferramenta online Google Docs para a hospedagem do formulário planejado e tabulação inicial dos dados coletados nas entrevistas.

4 RESULTADOS

O Quadro 1 apresenta o resultado tabulado da coleta de dados. O questionário criado contou com 77 questões que buscaram explorar a descrição do profissional e da empresa na qual trabalha, como era composta a sua equipe de desenvolvimento, o processo de desenvolvimento e os métodos de gestão/qualidade utilizados, e quais eram as técnicas de testagem de software utilizadas.

A coleta de dados foi realizada individualmente através de entrevistas online (via Meet) com profissionais que atuam em Uruaçu e região. Elas foram realizadas entre 23 e 29 de maio e tiveram duração média de 1h e 20min. O perfil dos entrevistados é jovem, com idade média inferior a 30 anos, e com tempo médio de atuação na área de desenvolvimento de cerca de 4,5 anos. Foram realizadas seis entrevistas e o professor orientador também participou de todas elas. A quantidade de amostras é considerada pequena, mas condizente com a realidade de poucas empresas e profissionais desenvolvedores de software sediados/residentes na região.

Todos os entrevistados são egressos ou concluintes do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) do IFG-Câmpus Uruaçu. Vale ressaltar que apenas um profissional atua em empresa local (o Gabriel Veloso atua na Singular Consultoria e Soluções, empresa sediada de Campinorte-GO), os demais profissionais atuam de forma remota para empresas sediadas em outras partes do país.

A ordem dos profissionais presentes no Quadro 1 obedece a uma ordem crescente (da esquerda para a direita) da complexidade do processo de desenvolvimento de software no qual o profissional está envolvido.

Quadro 1 - Resultado tabulado da coleta de dados

Nº	**1.IDENTIFICAÇÃO**	**EMPRESA/PROFISSIONAL**					
1	Nome*: *(nomes fictícios são mostrados)*	João*	Pedro*	Marcos*	Maria*	Eduardo*	Alessandro*
2	Qual empresa representa?	CLT como Programador na Singular Consultoria e Soluções (empresa brasileira criada em 2009, com cerca de 5 funcionários e sediada em Campinorte-GO) desde março/2021	Contratado PJ como Desenvolvedor Sênior na Smark CRM desde setembro/2022 (empresa brasileira sediada em Porto Alegre-MG, com cerca de 26 funcionários)	CLT como Desenvolvedor Pleno na Interação Tecnologias (empresa brasileira sediada em Uberlândia-MG, com cerca de 35 funcionários) desde janeiro/2022	PJ como Desenvolvedora Júnior na Voraz Consultoria (empresa brasileira sediada em Pato Branco-PR, com cerca de 46 funcionários) de janeiro/2022 a março/2023	CLT Fullstack web na Easy Pallet (Startup brasileira sediada em Brasília, com cerca de 10 funcionários e controlada pela LinkedBy) desde 2021	CLT como Engenheiro Sênior Líder na NTT DATA desde 2022 (Multinacional japonesa com cerca de 500 mil funcionários, sede Japão e filial de Uberlândia-MG)
3	Qual é o seu telefone e e-mail de contato?	******@gmail.com	******@gmail.com	******@hotmail.com	******@outlook.com	******@icloud.com	******@gmail.com
4	Qual é a cidade onde trabalha?	Uruaçu-GO	Uruaçu-GO	Uruaçu-GO	Uruaçu-GO	Uruaçu-GO	Uruaçu-GO
5	Trabalha em regime de home-office?	Sim, dedicação exclusiva	Sim, por demanda	Sim, 40h semanais	Sim, por demanda	Sim, por demanda	Sim, 40h semanais
6	Em quais setores atua/presta serviço?	Mantém softwares para gestão em mineradoras, atuando nas várias áreas destas empresas (a mineradora Maracá é seu principal cliente)	Sofware próprio de gestão de produtos para varejo e gestão/relacionamento com clientes (maior cliente da empresa é a Unimed)	Serviço de bot para cobranças bancárias e app para suporte ao produtor rural	Empresa com software ERP e software para gestão do agronegócio	Logística: Sistema de gestão de picking (separação de produtos). Presta serviço para distribuidores como os da Coca-Cola	Consultoria para desenvolvimento de softwares dos clientes. Também atua na área de gestão do negócio, RH, etc. O Diogo atua no desenvolvimento Web de soluções bancárias para Pessoa Jurídica do banco ****** (gestão

							financeira, pagamentos, empréstimos, etc.)
7	Presta serviços para quais cidades/regiões?	Todo o país e em países do exterior como México, Honduras, Equador etc.	Todo o país	Todo o país	Todo o país	Todo o país e iniciando expansão para o exterior	Todo o país e a nível global
8	Qual é a sua formação acadêmica/profissional?	Concluinte do curso ADS no IFG-Câmpus Uruaçu	Graduado em ADS (egresso IFG-Câmpus Uruaçu)	Graduado em ADS (egresso IFG-Câmpus Uruaçu)	Graduada em ADS (egressa IFG-Câmpus Uruaçu)	Concluinte do curso ADS no IFG-Câmpus Uruaçu. Possui certificados da Alura e Onebitcode	Graduado em ADS (egresso IFG-Câmpus Uruaçu)
9	Há quanto tempo atua na área de desenvolvimento?	Desde 2021	Desde 2017	Desde 2019	Desde 2022	Desde 2021	Desde 2017
	2.EQUIPE DE DESENVOLVIMENTO						
10	Qual é o tamanho dela?	5	8	6	8	6	11(mas faz parte de uma equipe bem maior que desenvolve serviços para PJ, com mais de 400 pessoas)
11	Descreva melhor como está organizada a sua equipe:	4 Devs Fullstack e 1 Gerente de Projeto(Leonardo)	6 Devs Fullstack, 1 Quality Assurance e 1 CTO (Chefe de Operação de Tecnologia)	2 Devs Mobile, 2 Devs Fullstack, 1 Gerente de projetos e 1 QA (Quality Assurance)	1 Gerente de Projeto e 7 Devs Fullstack	2 Devs Fullstack, 2 Devs Mobile, 1 CTO(Chefe de Operação de Tecnologia) e 1 QA (Quality Assurance)	6 Devs Fullstack, 2 QA (Quality Assurance), 1 Gerente de Projetos, 1 Product Owner e 1 Tech Lead
12	Quais são as principais linguagens de programação e	C# com asp.net core, Javascript, Bootstrap e Jquery	C# com Dot.net e Javascript	C# com .net, Entity Framework, React(mobile)	C# com .net, Angular, Entity e Postman	Web: Ruby on rails(backend) e VUE.JS(frontend) Mobile: Kotlin	Java com Spring no backend, Framework Angular, Xcode(IOS) e Java para Android

	frameworks utilizados pela sua equipe?						
13	Quais são a principais ferramentas de desenvolvimento utilizadas?	IDE VisualStudio, VisualStudioCode e Microsoft SQLServer	IDE VisualStudio com SQL Server e Redis	IDE VisualStudio, PostgreSQL e Firebase(não relacional)	IDE VisualStudio, HanaSQL e SQLServer	VisualStudioCode, PostgreSQL e Redis	IDEs InteliJ e Eclipse, Oracle e DínamoDB(AWS)
14	Sua equipe desenvolve projetos do tipo web, mobile e/ou desktop?	Web e partes Mobile (desenvolvimento terceirizado)	Web e Mobile (desenvolvimento terceirizado)	Web(o Edinei atua nele) e Mobile	Desktop, com algumas partes Web	Web(o Thiago atua nele) e Mobile	Web(o Diogo atua nele) e Mobile (ambos com foco em PJ)
	3.GESTÃO DE SOFTWARE						
15	Descreva qual é e como funciona o tipo de processo de desenvolvimento de software aplicado pela sua equipe:	Atividades gerenciadas com Trello(Kanban)	Scrum, com o líder sendo o CTO (um dos sócios da empresa)	Scrum, com líder sendo o Gerente de Projeto	Scrum, com o líder sendo o Gerente de Projeto	Scrum, com o líder sendo o CTO	Srum, sendo o líder o Gerente de Projetos
16	Descreva qual é e como funciona o método de governança de TI utilizada pela sua equipe:	Não se aplica, equipe pequena	Não se aplica, equipe pequena	Não se aplica, equipe pequena	Não se aplica, equipe pequena	Não se aplica, equipe pequena	São aplicadas partes da ITIL dentro da equipe, como gestão de portifólio e conhecimento através da plataforma Confluence, Sharepoint e GithubPages.
17	Descreva qual é e como funciona o método de gestão de projetos de desenvolvimento utilizados:	O gerente de projetos gerencia as atividades pelo Trello(Kanban)	Scrum, com o CTO organizando os sprints e as tasks	Scrum, o Gerente gerencia os projetos e cria as cards/tasks	Scrum, o Gerente gerencia os projetos e cria as cards/tasks	Scrum, o CTO gerencia os projetos e cria as cards/tasks	Scrum, com uso do Kanban para fazer o controle das sprints e das tasks. OKRs para definição dos

							objetivos-chave dos sprints.
18	Sua equipe possui um gerente de projetos?	Sim, 1 Gerente de Projetos	Sim, 1 CTO	Sim, 1 Gerente de Projetos	Sim, 1 Gerente de Projetos	Sim, 1 CTO	Sim, 1 Gerente de Projetos
19	Descreva qual é e como funciona o método de gestão da qualidade de software e de melhoria do processo de desenvolvimento utilizados:	Não se aplica	Não se aplica	Não se aplica	Não se aplica	Não se aplica	Utiliza, mas não soube informar qual
20	Descreva qual é e como funciona o método de gestão da configuração e de versionamento utilizados:	Git com Gitlab	Uma API do sistema é versionada no Git e o restante do código é versionado no TFS (Microsoft)	Git, via VisualStudio	Git para projetos internos e Azure para demandas que deveriam ser compartilhadas com as empresas clientes	Git, através da plataforma BitBucket	Github, com configurações na AWS e datacenters
21	Descreva como é a seção ou profissional específico para trabalhar com qualidade de software:	Não possui	1 profissional de QA	1 profissional de QA	Não possui	1 profissional de QA	2 profissionais de QA
22	Descreva como avalia a qualidade dos softwares entregues aos seus clientes:	Muito boa, em geral	Como são várias etapas para avaliação da funcionalidade desenvolvida, em geral, o software entregue satisfaz aos clientes	Muito bom, em geral	Boa, em geral	Houve problemas recentes com versões entregues, mas foi disponibilizado um QA para lidar com aspectos de qualidade	Muito boa, são vários filtros de qualidade até que o software seja entregue para produção
23	Descreva como avalia a satisfação dos seus clientes quanto à qualidade	Muito boa	Bastante satisfeitos	Não soube informar	Boa, em geral	Boa	Muito boa, o software é avaliado pelos próprios usuários

	dos softwares entregues:						
24	Com que frequência os softwares planejados são entregues dentro do cronograma e custos previamente definidos?	Na maioria das vezes	Nem sempre os prazos das sprints são cumpridas pois, como política da equipe, para se iniciar uma nova sprint, os erros pendentes e encontrados na versão anterior têm que ser corrigidos	Em geral, são cumpridos os prazos dos sprints	Na maior parte, as sprints eram finalizadas dentro do prazo estabelecido	Em geral, são cumpridos os prazos dos sprints	Em geral os sprints estouram os prazos previamente definidos
25	Descreva os principais pontos fortes do seu processo de desenvolvimento e de qualidade do software:	Responsabilidades bem definidas, atividades bem detalhadas, boa comunicação com o gerente (gerente como um facilitador do processo) e ferramentas que facilitam o desenvolvimento	Equipe com boa comunicação e colaboração, profissional de QA muito criterioso, tasks muito detalhadas	Equipe com boa comunicação, tasks bem detalhadas e responsabilidades bem divididas	Equipe com boa comunicação, responsabilidades e funções bem definidas	Equipe com ótima comunicação, com bom processo de testagem, tasks bem detalhadas e de fácil entendimento	Equipes bem definidas, com tasks detalhadas e processo bem definido e organizado
26	Descreva os principais pontos fracos (ou que precisam de melhorias) do seu processo de desenvolvimento e de qualidade do software:	Falta de testes unitários e de padrões de criação de código	A plataforma TFS dificulta os aspectos de versionamento de código	Os devs não possuem acesso ao ambiente de homologação para descobrirem quais foram os erros encontrados	Organização e distribuição das sprints e tasks não era eficiente, tasks com poucos detalhes. Falta de um servidor interno para realização de testes.	Profissional de QA sobrecarregado de funções (o que está sendo resolvido)	Processo burocrático, devido à necessidade do negócio
27	Descreva quão satisfeita é a sua equipe quanto ao processo de desenvolvimento utilizado:	Bem satisfeita	Bem satisfeita	Bem satisfeita	Bem satisfeita	Bem satisfeitos, bom ambiente de trabalho e processo bem organizado	50% satisfeita, devido à burocracia e qualidade exigida pelo negócio

28	Quais são as principais ferramentas utilizadas para gestão de software?	Trello(Kanban)	Plataforma AzureDEVOPS	Kanban	Jira	Plataforma de gestão de projetos Atlassian, que integra com a plataforma BitBucket	Kanban e IUclick(ferramenta própria do banco ******)
	4.TESTES DE SOFTWARE						
29	Descreva, em linhas gerais, como se dá o planejamento e execução dos testes de software nos projetos de desenvolvimento:	Devs realizam apenas testes manuais, não realizam testes automatizados	Não são codificados testes, são realizados testes manuais na medida em que o código é criado.	Não são utilizados testes automatizados, mas são realizados testes manuais pelos devs	Eram realizados apenas testes manuais pelos devs	Para as tasks novas são criados testes automatizados. Para correções, nem sempre são criados testes automatizados(caso eles não existam)	Para as tasks são realizados testes automatizados
30	Descreva como é visto pela equipe a importância de se testar o software que está sendo desenvolvido:	Importante e obrigatório	Importante e obrigatório	Muito importante e obrigatório	Muito importante e obrigatório	Primordial e obrigatória	Muito importante e obrigatório
31	Descreva qual é a estimativa média de tempo/custo necessários por projeto para se planejar, criar, executar e documentar testes:	Para cada task, entorno de 15% do tempo de desenvolvimento é gasto para executar os testes	Para cada task, entorno de 15% a 20% do tempo de desenvolvimento é gasto para executar os testes	Para cada task, entorno de 20% do tempo de desenvolvimento é gasto para executar os testes	Para cada task, entorno de 20% do tempo de desenvolvimento é gasto para executar os testes	Para cada task, entorno de 30% do tempo de desenvolvimento é gasto para criar os testes e testar	Para cada task, entorno de 30% do tempo de desenvolvimento é utilizado para criar os testes e executá-los
32	Descreva como e com que frequência ocorrem as inspeções e revisões de	Os devs de tempos em tempos revisam seus próprios códigos (por padrão, um dev não revisa o código do outro). Em	Não são realizadas inspeções e revisões	Não são realizadas inspeções e revisões	1 Fullstack mais experiente sempre revisava os códigos dos demais devs	Todo o código criado pelos devs é revisado pelo CTO antes que ele suba para o profissional de QA	Os devs líderes fazem inspeções e revisões do código antes do seu envio para o ambiente piloto

	documentos e de código dos projetos:	geral, uma atividade/task é realizada por um único dev.					
33	Descreva como é visto pela equipe a importância de se realizar inspeções e revisões:	Importante	Diante do processo estabelecido, eles não são vistos com importância	É importante, mas a empresa e a equipe não adotaram essa prática	Importante, eram sugeridas muitas otimizações de código	Primordial e obrigatória	É uma etapa importante, pois ajuda a garantir a qualidade do código
	4.1.Teste unitário						
34	Os testes unitários são sempre empregados nos projetos de desenvolvimento?	Nunca	Nunca	Nunca	Sempre, mas realizados de forma manual	Sempre. Em geral existe um este unitário para cada método mais importante	Sempre
35	Descreva como os testes unitários são empregados nos projetos de desenvolvimento:	Não se aplica	Não se aplica	Não se aplica	Os testes manuais eram feitos de acordo com a necessidade da task	Os testes automatizados ficam registrados na máquina dos devs e na plataforma BitBicket (sempre que um merge é feito na plataforma, os testes automatizados são executados)	Os testes tem que cobrir funções e 95% do código de todos os arquivos
36	Descreva quais são as diretrizes gerais utilizadas para a criação de casos de testes unitários:	Não se aplica	Não se aplica	Não se aplica	Os testes manuais eram feitos de acordo com a necessidade da task	Eles são criados para métodos mais críticos e com manipulação de dados	Os testes são criados de acordo com a necessidade da task
37	Os testes unitários criados são totalmente automatizados?	Não se aplica	Não se aplica	Não se aplica	Não	Sim	Sim

38	Quais são as ferramentas de automação de testes unitários utilizadas?	Não se aplica	Não se aplica	Não se aplica	Não	Rspec(biblioteca para teste automatizado do Ruby)	Jest(ferramenta que testa back e frontend)
39	Descreva como é visto pela equipe a importância de se testar software através de testes unitários:	Importante, entretanto não faz parte do processo de desenvolvimento	Não se aplica	Não se aplica	Importante	Muito importante	Essencial
	4.2.Teste de componente						
40	Os testes de componentes são sempre empregados nos projetos de desenvolvimento?	Nunca	Nunca	Nunca	Maioria das vezes	Nunca	Sempre
41	Descreva como os testes de componentes são empregados nos projetos de desenvolvimento:	Não se aplica	Não se aplica	Não se aplica	Eram feitos para partes mais críticas do sistema(gestão de nota fiscal, por exemplo)	Não se aplica	Todos os componentes, serviços e interfaces são testados
42	Descreva quais são as diretrizes gerais utilizadas para a criação de casos de testes de componentes:	Não se aplica	Não se aplica	Não se aplica	Os testes são criados de acordo com a necessidade da task	Não se aplica	Os testes são criados de acordo com a necessidade da task
43	Os testes de componentes criados são automatizados?	Não se aplica	Não se aplica	Não se aplica	Não, eram realizados de forma manual	Não se aplica	Sim

44	Quais são as ferramentas de automação de testes de componentes utilizadas?	Não se aplica	Não se aplica	Não se aplica	Não	Não se aplica	Jest
45	Descreva como é visto pela equipe a importância de se testar software através de testes de componentes:	Não considera importante, diante da necessidade do que é desenvolvido	Não se aplica	Não se aplica	Essencial, pois eram testados aspectos críticos do sistema	Não se aplica	Essencial
	4.3.Teste de sistema						
46	Os testes de sistema são sempre empregados nos projetos de desenvolvimento?	Sempre	Sempre	Sempre	Sempre	Sempre	Sempre
47	Descreva como os testes de sistema são empregados nos projetos de desenvolvimento:	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema	Sempre que uma feature é criada ou alterada, testes de sistema básicos são feitos pelos devs na interface do sistema
48	Descreva quais são as diretrizes gerais utilizadas para a criação de casos de testes de sistema:	Os testes são criados de acordo com a necessidade da task	Baseado na task que gerou a alteração no sistema	Baseado na task que gerou a alteração no sistema	Os testes são criados de acordo com a necessidade da task	Baseado na task que gerou a alteração no sistema	Os testes são criados de acordo com a necessidade da task
49	Os testes de sistema criados são automatizados?	Não, são feitos manualmente	Não	Não. Entretanto, existem objetos mock que simulam o comportamento das requisições mobile.	Não	Não	Parte manual, parte automatizada. Após a realização dos testes de sistema, o código é subido para o

				Esses objetos são implementados através do framework Insomnia e Swagger			ambiente de homologação, onde é feita uma nova rotina de testes de vulnerabilidade/segurança via Sonarcube e Fortify, e novo teste com o Jest. Se aprovado nessa esteira, é feita mais uma rodada de testes manuais pelo dev no ambiente de homologação.
50	Quais são as ferramentas de automação de testes de sistema/interfaces utilizadas?	Não	Não	Não	Não	Não	Jest
51	Descreva como é visto pela equipe a importância de se testar software através de testes de sistema:	Obrigatório, sendo este basicamente o tipo de teste feito pela equipe de desenvolvimento	Importante e obrigatório, pois evita que erros sejam passados adiante	Essencial	Essencial	Muito importante, pois permitem encontrar erros ainda pela equipe de desenvolvimento	Essencial
52	Quais são os procedimentos adotados no caso de não aprovação do software nos testes de sistema?	O próprio dev realiza as correções necessárias	O próprio dev realiza as correções necessárias	O próprio dev realiza as correções necessárias	O próprio dev realiza as correções necessárias	O próprio dev realiza as correções necessárias	O próprio dev realiza as correções necessárias. Caso problemas sejam encontrados no ambiente de homologação, o dev e o QA responsável atuam para resolvê-los.

	4.4.Teste de release						
53	Os testes de release são sempre empregados nos projetos de desenvolvimento?	Nunca	Sempre	Sempre	Metade das vezes	Sempre	Sempre
54	Possui um profissional ou equipe, fora da equipe de desenvolvimento, responsável por realizar testes de release?	Não existe	Sim, existe 1 profissional de QA	Sim, existe 1 profissional de QA	Sim, 1 consultor externo (que representa o cliente)	Sim, existe 1 profissional de QA	Sim, 2 profissionais de QA
55	Os testes de release são realizados por profissionais que participaram ativamente do desenvolvimento do sistema?	Não se aplica	Não	Não	Não	Não	Não
56	Descreva como os testes de release são empregados nas versões dos sistemas:	Não se aplica	O profissional de QA entende a task nova e realiza testes manuais vinculados a ela, criando também testes automatizados. Em geral, cada sprint gera uma versão do software. A cada versão nova todos os testes automatizados são executados.	O profisisonal de QA entende a task nova e realiza testes manuais vinculados a ela. Esse procedimento é feito no ambiente web de homologação. O app já se conecta nele para testes.	O consultor recebia o release em um ambiente de homologação do cliente(em geral, para cada sprint era gerado um release do sistema) e realizava os testes de verificação das funcionalidades	O profissional de QA entende a task nova e realiza testes vinculados a ela no ambiente de homologação e com o sistema de ponta a ponta (caso julgue necessário)	O profissional de QA recebia o release em um ambiente de homologação(em geral, para cada sprint era gerado um novo release) e realizava os testes de verificação das funcionalidades
57	Descreva quais são as diretrizes gerais utilizadas para a	Não se aplica	Baseado na task que gerou o release	Baseado na task que gerou o release	Baseado na task que gerou o release	Baseado na task que gerou o release	Baseado na task que gerou o release

	criação de casos de testes de release:						
58	Os testes de release criados são automatizados?	Não se aplica	Parte manual, parte automatizada	Não	Não	Não	Parte manual, parte automatizada
59	Quais são as ferramentas de automação de testes de release/interfaces utilizadas?	Não se aplica	Selenium	Não	Não	Não	Jest e outras
60	Descreva como é visto pela equipe a importância de se testar software através de testes de release:	Não é importante, diante do processo de desenvolvimento estabelecido	Muito importante, pois ajuda a evitar que erros sejam colocados em produção	Muito importante, pois evita que erros sejam passados para o ambiente de produção	Muito importante, pois evita que erros sejam passados para o ambiente de produção	Muito importante, pois ajuda a evitar que erros sejam colocados em produção	Muito importante, pois ajuda a evitar que erros sejam colocados em produção
61	Quais são os procedimentos adotados no caso de não aprovação do software nos testes de release:	Não se aplica	Após a conclusão de cada task e/ou spint, o código é disponibilizado no ambiente de homologação. Nele, o QA, caso encontre um erro em uma task, ele notifica o dev através de um comentário na task. Quando ele encontra um erro envolvendo toda uma spint (várias tasks), ele abre uma task de bug/correção. Nesses casos, o software não é levado para produção.	O QA acrescenta na task as evidências de problemas e a devolve para o dev	O consultor retornava via e-mail os erros, o gerente verificava e solicitava aos devs as correções (via adição na task dos problemas encontrados)	O sotware não é liberado para produção e a task é devolvida para os devs, com as considerações sobre os problemas encontrados	O QA acrescenta na task as evidências de problemas e a devolve para o dev
62	Descreva como é feita a entrega do	Após os testes de sistema pelo dev, o	Após passar pela homologação e teste	Após aprovação pelo QA no ambiente de	Após a aprovação do consultor e do	Após a aprovação pelo QA, o release é	Após aprovação pelo QA no servidor de

	software/release aos clientes/usuários:	gerente sobe o release para o servidor web de homologação presente na Infra do cliente. Após aprovação pelo cliente, o release é homologado e configurado no servidor web de produção do cliente	do usuário, a nova versão do sistema é disponibilizada no servidor web de produção (em horários não críticos, geralmente às 6h da manhã).	homologação, o release é implantado no servidor web de produção	cliente(usuário final), o release era implantado no ambiente de produção do cliente	implantando no servidor web de produção (geralmente pela manhã, pois a maior parte do uso do sistema ocorre durante a noite)	homologação, o código é disponibilizado no servidor piloto, onde usuários testam o sistema de forma controlada. Se aprovado por eles, o código é subido para produção com 70% dos clientes tendo acesso à nova versão. Se não houver problemas no seu uso, a versão é liberada para 100% dos clientes.
	4.5.Teste de usuário						
63	Os testes de usuário são sempre empregados nos projetos de desenvolvimento?	Sempre	Na maioria das vezes. Entretanto, quando as alterações envolvem apenas melhorias internas ou afins, o software não é testado pelo usuário. Os testes são acompanhados pelo CTO e às vezes pelo QA.	Nunca	Sempre	Para funcionalidades de grande impacto/críticas são realizados testes com usuários	Sempre
64	Descreva como os testes de usuário são empregados nas versões dos sistemas:	Após os testes de sistema pelo dev, o gerente sobe o release para o servidor web de	Após passar pela homologação, a nova versão do sistema é disponibilizada em um servidor web de	Não se aplica. Entretanto, o gerente de projetos sempre apresenta protótipos de telas do sistema ao	Os usuários finais participavam apenas na etapa de homologação	Para funcionalidades de grande impacto/críticas, é disponibilizado ao usuário o link para o	Em um servidor piloto, usuários selecionados testam a nova versão do sistema de forma

		homologação presente na Infra do cliente. Após aprovação pelo cliente, o release é homologado e configurado no servidor web de produção do cliente	testes/pré-produção, onde os usuários testam o software por um período.	usuário após a definição da funcionalidade.		ambiente de homologação, onde ele testará a funcionalidade	controlada, como se fosse em um ambiente de produção. Se aprovada por eles, esta nova versão também é testada por uma equipe de testadores que apoiam a alta gestão do banco.
65	Descreva quais são as diretrizes gerais utilizadas para a realização de testes de usuário:	Depende da task realizada	Depende da task realizada	Não se aplica.	Depende da task realizada	Depende da task realizada	Depende da task realizada
66	Os testes do tipo alfa são sempre empregados nas versões do software?	Não	Não	Não	Não	Não	Não
67	Descreva como se dá a realização dos testes alfa:	Não	Não	Não	Não	Não	Não
68	Os testes do tipo beta são sempre empregados nas versões do software?	Não	Não se aplica, mas às vezes são feitas apresentações para o cliente da funcionalidade que está sendo desenvolvida (em geral necessária para funcionalidades maiores)	Não	Não	Não	Não
69	Descreva como se dá a realização dos testes beta:	Não	Não	Não	Não	Não	Não

70	Os testes do tipo de aceitação são sempre empregados nas versões do software?	Não	Não	Não	Não	Não	Não
71	Descreva como se dá a realização dos testes de aceitação:	Não	Não	Não	Não	Não	Não
72	Descreva como é visto pela equipe a importância de se testar software através de testes de usuário:	Fundamental e obrigatório, pois eles tinham que aceitar o release disponibilizado	Muito importante	Não se aplica	Fundamental, pois eles tinham que aceitar o release disponibilizado	Fundamental	Fundamental, pois eles tinham que aceitar o release disponibilizado
73	Quais são os procedimentos adotados no caso de não aprovação do software nos testes de usuário (principalmente no teste de aceitação):	O gerente é notificado, através das evidências fornecidas em um documento .doc e comunica o dev para que sejam feitas as adequações	Caso o problema seja um erro, o QA atua para simulá-lo e repassá-lo para os devs. Caso seja uma melhoria não solicitada, o CTO atua para rever a funcionalidade.	Não se aplica. Os usuários fazem uso do sistema já em ambiente de produção. Caso um erro seja reportado pelo usuário, o QA atua para simulá-lo e repassá-lo para os devs. Em caso de melhorias/sugestões, a diretoria atua para rever a funcionalidade, com possibilidade de novo orçamento/cronograma.	O consultor retornava via e-mail os erros repassados pelo cliente, o gerente verificava e solicitava aos devs as correções (via adição na task dos problemas encontrados)	O CTO se reúne com os usuários para detalhamento dos problemas encontrados e novas solicitações/orientações são repassadas para a equipe de devs (via adição na task dos problemas encontrados)	Caso algum problema seja encontrado, o QA que acompanha os testes dos usuários acrescenta na task as evidências de problemas e a devolve para o dev
	5.OUTROS TIPOS DE TESTES E CONSIDERAÇÕES						

74	Descreva, caso se aplique, como são realizados testes de recuperação:	Não se preocupavam com esses testes, pois a Infra de cada cliente é que deveria se preocupar com esses fatores	Uma equipe de Infra é responsável por aspectos de backup e restauração. O servidor web é hospedado na nuvem (AWS) com autoscaling e loadbalancer.	As rotinas de backup são feitam pela equipe de Infra da empresa	Não se preocupavam com esses testes, pois a Infra de cada cliente é que deveria se preocupar com esses fatores	São feitos backups diários do bando de dados e existe contingência do servidor web	Desconhece, devido ao tamanho do negócio
75	Descreva, caso se aplique, como são realizados testes de segurança:	Não realizam esses testes, mas a maioria dos clientes realizam testes de vulnerabilidade (pentest - teste de penetração)	Não realizam procedimentos específicos	Rotinas de acesso são feitas pelas API do sistema	Não realizavam esses tipos de testes	Ainda não existem rotinas específicas para esse tipo de teste. O CTO realiza alguns procedimentos gerais de segurança	Vulnerabilidade/segurança são avaliadas via Sonarcube e Fortify
76	Descreva, caso se aplique, como são realizados testes de desempenho:	Não são realizados	O CTO realiza o monitoramento do servidor. Não são realizados testes de desempenho e estresse	Aparentemente não são realizados esses tipos de testes	No ambiente de homologação às vezes eram realizados testes de desempenho, com vários usuários fazendo uso do sistema ao mesmo tempo	É feito o monitoramento de uso do banco de dados e servidor web. O CTO monitora esses aspectos	Existe uma equipe em separado para estressar o sistema, verificando seu desempenho
77	Comente outros aspectos/dinâmicas sobre testes/qualidade utilizados e relevantes e que não foram abordados na entrevista:	Não foi necessário	Não foi necessário	Não foi necessário	Não foi necessário	Não foi necessário	São feitos testes de usabilidade, atendendo aos componentes de interface do banco ****** e à acessibilidade para deficientes visuais e auditivos. O QA faz testes de regressão, testando toda versão nova do sistema

5 DISCUSSÕES

Nesta seção é feita uma análise dos dados coletados e tabulados no Quadro 1. Será dado evidência aos principais aspectos considerados relevantes, além de se buscar responder aos principais objetivos desejados com essa pesquisa.

A partir dos dados, houve o entendimento de como funciona os processos de desenvolvimento de software nas empresas nas quais os profissionais atuam. Pode-se observar que o processo de desenvolvimento e de testagem de software no qual está envolvido cada profissional é único. Ou seja, cada empresa adaptou seu processo de acordo com a sua realidade/necessidade. Esta é uma característica que já era esperada, haja visto que empresas desenvolvedoras podem atuar em nichos de mercado muito distintos e com demandas específicas, além de poderem optar por dezenas de métodos, técnicas e padrões para compor os seus processos de desenvolvimento e testagem de software.

No Quadro 1 pode-se perceber essa diversidade: de processos simples e que exigem poucas etapas de testagem a processos complexos/burocráticos e que exigem várias camadas de testagem de software.

Com relação às seis empresas em que atuam os profissionais, percebe-se que cada uma atua em uma área/setor específico (conforme questão nº 6 do Quadro 1). Todas atuam a nível nacional, sendo que três delas possuem atuação também no exterior. Cinco das seis empresas são empresas de pequeno/médio porte (a empresa na qual o profissional Diogo Filho atua é uma multinacional de grande porte).

A questão nº 10 do Quadro 1 deixa claro que todas as esquipes de desenvolvimento nas quais os profissionais estavam inseridos eram pequenas (o profissional Diogo Filho atua na maior equipe, contando com 11 profissionais. Vale ressaltar, entretanto, que essa equipe é apenas um segmento de uma equipe de desenvolvimento bem maior). As questões nº 12 e 14 evidenciam que quatro das equipes de desenvolvimento utilizam a linguagem C# como principal linguagem de programação e que cinco equipes têm como foco principal o desenvolvimento Web.

Ao se observar os processos de desenvolvimento utilizados pelas empresas, percebe-se que todas elas equipes utilizam métodos ágeis para desenvolvimento de software em incrementos, sendo que cinco delas utilizam o método de desenvolvimento/gestão Scrum, organizando as suas atividades em *sprints, tasks* e/ou *cards* (conforme questões nº 15 e 17 do Quadro 1). Sabe-se que os métodos

ágeis priorizam a entrega rápida de incrementos de software e dos benefícios do seu emprego em equipes desenvolvedoras de pequeno/médio porte. A questão nº 18 do Quadro 1 mostra outro aspecto importante: todas as equipes possuem pelo menos um profissional gestor dedicado, seja ele um Gerente de Projetos ou CTO. A questão nº 21 mostra que quatro equipes possuem um profissional dedicado de Gestão de Qualidade. Sabe-se da importância desse profissional na avaliação da qualidade dos softwares que são entregues aos clientes.

De forma geral, os profissionais informaram que a qualidade dos softwares entregues é boa e que seus usuários/clientes avaliam de forma muito positiva o grau de satisfação com relação ao uso dos softwares (conforme questões nº 22 e 23 do Quadro 1). Estes profissionais também deixam claro que enxergam muitos pontos positivos nos processos de desenvolvimento que utilizam (como equipe com boa comunicação, bem gerida com e funções bem definidas) e que são satisfeitos com eles (conforme questões nº 25 e 27 do Quadro 1). Entretanto, o profissional Diogo Filho relatou que considera o processo de desenvolvimento no qual participa complexo e burocrático (mas que entende que isso é uma necessidade do negócio).

Dando início à análise das questões envolvendo diretamente os testes de software, todos os entrevistados informaram que testar o software que está sendo desenvolvido é uma atividade muito importante e obrigatória, e que eles gastam entorno de 15% a 30% do tempo do desenvolvimento para testar software (conforme questões nº 30 e 31 do Quadro 1). Vale ressaltar também que quatro das equipes de desenvolvimento realizam a inspeção/revisão do código criado e que essa atividade também é considerada muito importante (conforme questões nº 32 e 33 do Quadro 1).

Sobre os testes unitários, três profissionais informaram que eles não são usados, um informou que os utilizam apenas de forma manual e apenas duas equipes informaram que sempre programam estes tipos de testes e que os executam de forma automatizada (conforme questão nº 34 do Quadro 1). Esta informação surpreendeu os entrevistadores, pois sabe-se da importância que uma base de testes unitários automatizados tem para encontrar eventuais erros e colaborar para a garantia da qualidade do software desenvolvido.

Sobre os testes de componentes, apenas duas equipes informaram que estes tipos de testes fazem parte de sua rotina de desenvolvimento (conforme questão nº 40 do Quadro 1). Diante do tamanho reduzido das equipes e do emprego por elas de métodos ágeis, este era um resultado de certa forma esperado pelos entrevistadores.

Sobre os testes de sistema, todos os profissionais informaram que sempre fazem uso deste tipo de teste e que ele é uma etapa essencial do desenvolvimento, com eles realizando testes nas interfaces dos sistemas na medida em que o desenvolvimento da *task*/*card* avançava e efetuando as correções que fossem necessárias (conforme questões nº 46, 47, 51 e 52 do Quadro 1).

Com relação aos testes de release, cinco das seis equipes realizam este tipo de teste, com elas contando com um profissional de QA ou equivalente para planejá-lo ou executá-lo (conforme questões nº 53 e 54 do Quadro 1). Esta informação surpreendeu os entrevistadores pois, apesar das equipes serem de tamanho pequeno, elas têm o cuidado de terem um profissional dedicado às atividades de qualidade dos softwares desenvolvidos. Vale ressaltar que esses profissionais não participavam ativamente da programação dos sistemas e que, em geral, o release gerado a cada *sprint* era testado pelo profissional de QA em um ambiente de homologação (conforme questões nº 55 e 56 do Quadro 1). A aprovação da release pelo profissional de QA é uma etapa essencial na maioria das equipes, sendo essa aprovação necessária para implantação da release no ambiente de produção ou para que ela avançasse para a próxima etapa de testes (conforme questão nº 62 do Quadro 1).

Sobre os testes de usuário, eles são empregados por cinco das seis equipes, sendo por elas considerada uma etapa fundamental da testagem (conforme questões nº 63 e 72 do Quadro 1). Entretanto, pelas respostas obtidas na questão nº 64, percebe-se que os usuários não possuem participação durante a etapa de programação das *tasks*/*cards*. Basicamente, os usuários testam o sistema que já foi homologado. Sabe-se que os próprios métodos ágeis preconizam a participação ativa dos usuários durante todas as fases de desenvolvimento do software, desde a definição dos requisitos até a etapa de criação do código e entrega do sistema. Os testes do tipo alfa e beta não são realizados pelas equipes (conforme questões nº 66 a 69 do Quadro 1). Esta informação já era esperada, haja visto que os softwares desenvolvidos pelas equipes eram customizados para os seus usuários e que testes do tipo alfa e beta geralmente são empregados para softwares do tipo de prateleira.

Já discorrendo sobre as últimas questões presentes no Quadro 1, pode-se perceber que cada empresa/equipe possui uma dinâmica própria de realização de testes de recuperação, segurança e desempenho (conforme questões nº 74, 75 e 76 do Quadro 1).

Finalizando esta etapa de discussão, a questão 77 do Quadro 1 demonstra que o questionário elaborado foi suficiente para coletar/entender como se dá o processo de desenvolvimento de software nas empresas, envolvendo aspectos de testagem e gestão/qualidade (apenas o profissional Diogo Filho acrescentou que eram realizados também testes de usabilidade por sua equipe).

6 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Em uma tentativa de se obter uma “visão geral” sobre como empresas e profissionais desenvolvedores de software de Uruaçu-GO e região estão testando os softwares por eles criados/mantidos, esse trabalho realizou uma coleta de dados com seis profissionais que atuam em Uruaçu (a quantidade de amostras é pequena, mas condizente com a realidade de profissionais que atuam na região).

Para que a coleta de dados fosse feita de forma estruturada, foi criado um formulário que foi respondido pelos profissionais durante as entrevistas online que foram realizadas. O formulário era composto por 77 questões, que abrangiam aspectos como o processo de desenvolvimento utilizado pelas equipes, gestão/qualidade de software e, obviamente, as técnicas de teste de software empregados (dentre as inúmeras disponíveis). Vale ressaltar que todos os profissionais entrevistados são egressos ou concluintes do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFG-Câmpus Uruaçu.

Com os dados coletados e tabulados, foi realizada uma análise dos principais aspectos encontrados. Conseguiu-se, então, obter o entendimento sobre como os processos de desenvolvimento ocorriam. Observou-se que cada empresa adaptou o seu processo de acordo com a sua realidade/necessidade e que os processos variaram do simples ao complexo/burocrático.

Em linhas gerais, os profissionais estavam envolvidos em equipes pequenas e que faziam uso de métodos ágeis para o desenvolvimento de software em incrementos. Os entrevistados se mostraram satisfeitos com seus processos de desenvolvimento, destacando a boa comunicação entre a equipe, boa gestão e funções bem definidas.

Ainda como aspectos positivos, observou-se que a quase totalidade das equipes possuíam profissionais dedicados à gestão de projetos e de garantia da qualidade, e que vários tipos de teste eram empregados nos processos, como os testes de sistema, de release e de usuário (todos os entrevistados destacaram a importância de se testar o software em seus vários estágios de desenvolvimento). Estas boas práticas contribuíam para a boa qualidade dos softwares entregues pelas equipes.

Dentre as lacunas identificadas, verificou-se que apenas a metade das equipes realizavam a programação de testes unitários automatizados e que os usuários

(interessados pelos softwares) possuíam pouca interação com as equipes durante as etapas de codificação do sistema. Estes dois aspectos são importantes para a mitigação de erros e melhoria da qualidade dos softwares entregues e representam sugestões de melhorias que podem ser integradas aos processos de desenvolvimento analisados.

O planejamento e execução desse trabalho de pesquisa foi desafiador. Muito esforço foi necessário para o entendimento dos tipos de testes existentes e elaboração do formulário de forma a contemplar os elementos que seriam necessários para a coleta de dados. A realização das entrevistas e posterior tabulação de todos os dados coletados exigiu grande esforço, mas, ao final, esse processo foi bastante proveitoso e gratificante, pois conseguiu-se alcançar os objetivos desejados com esse trabalho.

A realização deste trabalho possibilitou o uso de conhecimentos adquiridos no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, além de permitir a aquisição de novos conhecimentos e habilidades.

7 REFERÊNCIAS BIBLIOGRÁFICAS

LAKATOS, E. M.; MARCONI, M. A. **Fundamentos de metodologia científica**. 6. ed. São Paulo: Atlas, 2006.

PRESMANN, R. MAXIM, B. **Engenharia de Software**. 8. ed. Porto Alegre: Bookman, 2016.

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

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. 2. ed. Rio de Janeiro: Elsevier, 2014.