

IMPORTÂNCIA DOS TESTES PARA A QUALIDADE DO SOFTWARE

João Marcelo Gomes Correia

Graduando em Engenharia de Software – Uni-FACEF

jomgomes73@gmail.com

João Victor Custódio Ferreira

Graduando em Engenharia de Software – Uni-FACEF

joaovictorjn.fc@gmail.com

Daniel Facciolo Pires

Doutor em Física Médica – Uni-FACEF

daniel@facef.br

Resumo

Este trabalho tem como objetivo apresentar um estudo de teste de usabilidade em um sistema de gerenciamento acadêmico de uma instituição de ensino superior. Assim, procura-se mostrar a importância dos testes de software para a qualidade de aplicativos pelas empresas na atualidade, e tenta responder à seguinte questão: como é possível realizar teste de software? O método utilizado é a realização de pesquisa descritiva e de referencial teórico por meio de levantamento bibliográfico dos principais autores e estudiosos do assunto, bem como a realização de um estudo de caso de teste de software acadêmico. Após a realização dos testes, foram sugeridas melhorias para aumentar a qualidade do software estudado.

Palavras-chave: Estratégia. Vantagem Competitiva. Código-Fonte.

Abstract

This work aims to present a study of usability testing in an academic management system of a higher education institution. Thus, the search shows the importance of software testing for the quality of applications by companies today, and tries to answer the following question: how is it possible to perform software testing? The method used is to carry out descriptive research and theoretical reference through a bibliographic survey of the main authors and scholars on the subject, as well as the realization of a case study of academic software testing. After performing the tests, improvements were suggested to increase the quality of the software studied.

Keywords: Tactics: Software Testing. Source Code. Software Quality.

1 INTRODUÇÃO

Assim como todas as áreas de conhecimento, o conceito de qualidade foi evoluindo ao longo do tempo, de acordo com as contribuições de autores renomados que dedicaram seus estudos a esta área. Joseph Juran, um dos nomes mais reconhecidos quando o assunto é a gestão da qualidade no século XX, propõe que as preocupações com a qualidade foram iniciadas ainda na Era Agrícola, há cerca de oito mil anos. Entretanto, é com os efeitos da Revolução Industrial que o controle da qualidade se torna uma técnica autônoma, ganha reconhecimento, principalmente devido ao início a produção em massa (PINTO, 2011, p. 50).

De acordo com Bartié (2002, p. 4) os primeiros conceitos de qualidade relacionados ao desenvolvimento de softwares começaram a surgir efetivamente no início dos anos 1980. Com essa abordagem, os desenvolvedores e testadores trabalhavam juntos desde o início do processo de desenvolvimento até a sua conclusão, e ao final de cada fase era realizada sua respectiva atividade de conferência, visando garantir que a etapa estava completa e bem compreendida.

Muitas organizações foram formadas nos anos 40, como ASQC (*American Society for Quality Control*), a ABNT (Associação Brasileira de Normas Técnicas) e a ISO (*International Standardization Organization*), cuja função seria regulamentar e controlar a aplicação de normas criadas para determinar padrões de qualidade a serem seguidos.

Entre os anos 1980 e 1990 as ferramentas de teste começaram a ser produzidas, com a promessa de que trariam alta produtividade e qualidade ao processo de teste. Atualmente, compreendemos que a aquisição destas ferramentas é essencial para garantir o sucesso e viabilizar um projeto, principalmente de grande porte, como a implantação de um processode garantia da qualidade de software em uma empresa.

Este trabalho tem como objetivo realizar um estudo de teste de usabilidade em um sistema de gerenciamento acadêmico de uma instituição de ensino superior. Assim, procura-se mostrar a importância dos testes de software para a qualidade de aplicativos pelas empresas na atualidade, e tenta responder à seguinte questão: como é possível realizar teste de software?

Apesar de o conceito de qualidade estar bem difundido, os aspectos que podem influenciar diretamente no resultado da qualidade de um software ainda não são percebidos com clareza. Por esse motivo, algumas empresas ainda se encontram presas a antigos paradigmas, o que impede (ou no mínimo atrasa) a adoção de abordagens mais inovadoras que surgiram com o avanço da tecnologia na área de desenvolvimento de softwares.

Projetando um cenário no qual são envolvidas diversas variáveis durante o processo de desenvolvimento do sistema, percebe-se que a complexidade deste tende a aumentar proporcionalmente. Isso faz com que aumente também a quantidade e gravidade dos riscos de mau funcionamento, tornando a produção do software com o nível de qualidade desejado mais complicada.

Visando aumentar a qualidade dos softwares, a área de Engenharia de Software tem produzido e utilizado algumas técnicas cujo objetivo principal é auxiliar no desenvolvimento do sistema. Uma destas técnicas é a aplicação de testes durante estágios de produção de um software, que tende a identificar os defeitos com antecedência, facilitando a correção dos mesmos e garantindo assim, mais qualidade ao produto.

Com isso, a realização deste trabalho utiliza o material e o conhecimento obtidos pro meio do referencial teórico para justificar a importância da realização de testes durante o desenvolvimento de um software de forma que a qualidade seja mantida no maior nível possível. Ainda, é realizado um estudo de caso de teste de software acadêmico.

A técnica utilizada nos testes é denominada como “Caixa Preta”, que consiste em uma forma de avaliar as funções e operabilidade do sistema, que neste caso será em um software livre. Este nome “Caixa Preta” é definido pelo simples fato de ser um teste onde não há necessidade de ter contato com o código fonte, é feito apenas sobre critério de avaliação conforme as respostas que o software dá às ações.

2 QUALIDADE DE SOFTWARE

2.1 História da qualidade

Embora a qualidade seja um assunto que passou a ser notado com mais frequência a partir do início da década de 1980, não se trata de uma invenção moderna. Alguns autores de livros e artigos acadêmicos concordam que o conceito ou a filosofia da qualidade existe desde muito tempo na antiguidade, discordando apenas em relação ao seu início (ANTONIO et al., 2007, p. 21).

Existem relatos de que, há mais de quatro mil anos, os egípcios estabeleceram um padrão de medida de comprimento que correspondia ao comprimento do braço do faraó reinante, que foi denominado como cúbito. Entretanto, seguindo esse conceito, a medida deveria ser atualizada sempre que ocorria a troca do faraó.

Mesmo assim, todas as construções egípcias deveriam ser realizadas utilizando o cúbito como unidade de medida. Para isso, bastões eram cortados de acordo com o comprimento correto e o responsável por uma construção deveria comparar, periodicamente, o padrão que estava sendo utilizado com o padrão real (JURAN e GRZYNA, 1988).

Já nessa época, a população começava a ter noções de padrão. Caso as construções não estivessem de acordo com o padrão real, o responsável poderia ser condenado até à morte. Assim, cria-se uma esfera na qual é necessário se atentar para o padrão definido buscando a perfeição, a obtenção da qualidade da obra que está sendo realizada. O conceito de qualidade começa a surgir, ainda que de forma implícita.

A história da qualidade seguiu sendo concretizada com inúmeros exemplos de resultados extraordinários, como os grandes templos construídos na Grécia e Roma antigas, as catedrais medievais, e os feitos realizados para a navegação em meados do século XVI. Em todas essas realizações, não se podia contar com instrumentos de precisão, muito menos com técnicas sofisticadas ou métodos especializados (VINCENT, 2004 apud KOSCIANSKI e SOARES, 2007, p. 18).

Com certeza, um dos grandes marcos na história da qualidade foi a Revolução Industrial, no século XVIII. Esse período está relacionado a profundas mudanças sociais e econômicas, como o início dos processos de automação e o surgimento do consumo em massa.

Durante essa época transformadora, surgiram milhares de novas empresas. A criação de diversas indústrias levou rapidamente à concorrência entre elas, desencadeando um processo de melhoria contínua que continua durando até os dias de hoje. O aumento da eficiência nesse período tornou-se uma condição imprescindível para garantir a sobrevivência de uma organização (KOSCIANSKI e SOARES, 2007, p. 18).

De acordo com Koscianski e Soares (2007), na década de 1920 surge o controle estatístico de produção. Com isso, tornou-se impossível garantir a qualidade individual de cada peça nas fábricas que produziam seus produtos em grande quantidade, ao contrário do que se fazia (e ainda se faz) na produção de peças por meio do trabalho artesanal. Até então, era possível avaliar e conferir cada peça produzida pelos artesãos, que validavam pessoalmente a qualidade da produção.

Já por volta dos anos 1940, começam a surgir vários organismos dedicados à qualidade; por exemplo, a ASQC (*American Society for Quality Control*), a ABNT (Associação Brasileira de Normas Técnicas) e a ISO (*International Standardization Organization*).

No fim desse período, principalmente durante a Segunda Guerra Mundial (1939 a 1945), houve uma contribuição histórica ao processo de conscientização da qualidade, quando as técnicas de manufatura passaram a ser aprimoradas para fabricação de material bélico mais eficaz e eficiente, com mais precisão e assertividade (KOSCIANSKI e SOARES, 2007, p. 19).

Também na década de 1940, o Japão se destacou como um país de referência no assunto e contribuiu com diversas novas técnicas cujo objetivo era garantir e manter a qualidade dos processos. Entre estas, destacam-se o método de Taguchi para projeto experimental, a metodologia dos Seis Sigma ou os diagramas de causa e efeito de Ishikawa, também conhecidos popularmente como diagramas espinha de peixe devido a sua representação gráfica (KOSCIANSKI e SOARES, 2007, p. 19).

O método de Taguchi pode ser entendido como uma abordagem voltada para o projeto do produto e do processo, no qual é executada apenas uma fração do número total de combinações das variáveis de entrada. Taguchi também propôs a utilização de matrizes compostas por colunas ortogonais para a realização deste método (FIGUEIREDO e FERNANDES, 1996).

A metodologia Seis Sigma é considerada revolucionária em relação à melhoria de processos dentro dos mais diversos setores de uma organização (desde a área de manufatura até a área comercial), que tem como objetivo atingir melhorias na qualidade dos processos e obter ganhos de produtividade expressivos, reduzindo os custos consequentemente. Esta metodologia foi desenvolvida originalmente pela Motorola e acabou sendo aperfeiçoada por diversas outras empresas, como a General Electric, que a utilizavam dentro da empresa. (RAMOS, 2009).

Já o diagrama de Ishikawa, ilustrado na Figura 1, representa a técnica que é utilizada para identificar as causas de um problema. Preferencialmente, deve ser utilizada durante uma reunião em que todos os envolvidos (*stakeholders*) estejam presentes e possam discutir livremente sobre as “categorias” (elementos) do diagrama (KOSCIANSKI e SOARES, 2007, p. 20).

Para entender o diagrama, é preciso perceber que o problema específico que afeta a empresa está representado no eixo central do diagrama, que no caso da Figura é a produção de um produto com defeito. As linhas diagonais são direcionadas ao problema contendo os elementos principais que fazem parte do cenário – como trabalhadores e máquinas. Assim, para cada categoria procura-se identificar fatores (causas) que possam contribuir para aumentar ou reduzir o problema (efeitos), e estes são descritos na sua respectiva diagonal, uma vez que são agrupados de acordo com sua natureza.

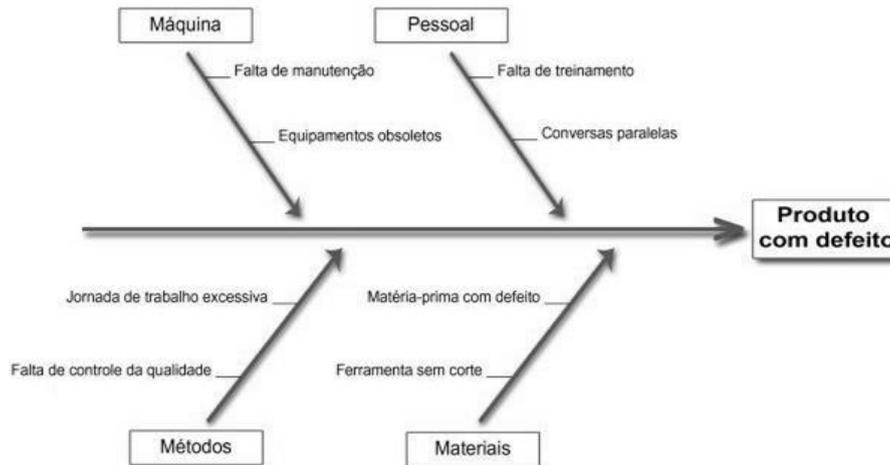


Figura 1 – Diagrama de Ishikawa.

Fonte: Adaptado de Ishikawa, 1943.

2.2 Definição do conceito

Há uma variedade de definições e conceitos na literatura quanto ao termo “qualidade”. Pela definição de Campos (1992), pode-se entender a qualidade como um produto ou serviço que atende às necessidades do cliente perfeitamente, de forma confiável, acessível e segura, no momento correto.

Para Antonio et al. (2007, p. 22), noção de qualidade é “algo de inerente ao ser humano, projetando-se em diversos atos, mesmo nos mais corriqueiros e automáticos da sua vida diária”. Entretanto, os autores ressaltam que a noção de qualidade deve estar associada diretamente às necessidades e/ou desejos que o produto procura satisfazer, devendo ainda considerar a diferença entre a satisfação dos desejos e a satisfação das necessidades.

De acordo com Deming (1998), um dos precursores do conceito da qualidade, qualidade é “atender as necessidades do cliente, resultando em satisfação do cliente” de modo que a produção esteja “livre de deficiências que resultem em insatisfação do cliente”.

Já Lobos (1993, p. 38) define a qualidade como “a condição de perfeição ou se preferir, do exato atendimento das expectativas do cliente, podendo ser atribuída a qualquer coisa ou ação, integrante ou resultante de um processo”.

Outro autor reconhecido ao tratar dos conceitos de qualidade é Crosby (1992), cuja definição de qualidade se dá pela frase “a qualidade é conformidade aos requisitos”. Ou seja, se existir conformidade com os requisitos, existe qualidade, assim como se não houver conformidade, a qualidade se perde. Em 1961, Crosby propôs uma abordagem conhecida como “defeito zero”, na qual todos os colaboradores da organização deveriam ser capazes de desenvolver o trabalho corretamente na primeira vez que fosse realizado.

Para Koscianski e Soares (2007, p. 30), a qualidade de um software depende da definição do termo que será utilizada como base para realizar o desenvolvimento. É preciso considerar diversos fatores que afetam a construção do produto e adotar a perspectiva técnica mais adequada que influencie no julgamento dos usuários. Alguns destes fatores estão descritos a seguir:

- Tamanho e complexidade do software sendo construído;
- Número de pessoas envolvidas no projeto;
- Ferramentas utilizadas;
- Custos associados à existência de erros;
- Custos associados à detecção e à remoção de erros.

2.3 SWEBOK

Na área Engenharia de Software foi realizado um estudo por uma comissão internacional de especialistas, cujo objetivo era identificar e definir as fronteiras que a delimitam. Este estudo foi conduzido no âmbito da IEEE (*Institute of Electrical and Electronics*)¹ e foi denominado como SWEBOK (*Software Engineering Body Of Knowledge*, ou Corpo de Conhecimento de Engenharia de Software, em tradução livre).

O SWEBOK aborda o conhecimento de engenharia de software necessário, mas não completamente suficiente, para a formação de um bom engenheiro de software, pois não foca em assuntos específicos como: linguagens de programação, bancos de dados relacionais e redes, por exemplo. O SWEBOK destaca o conhecimento essencial que possa servir como suporte de seleção da tecnologia mais adequada, no momento e na circunstância apropriados (SWEBOK, 2004).

O guia do conhecimento de Engenharia de Software (SWEBOK, 2004) foi elaborado com base em cinco objetivos principais:

- Promover uma visão consistente da engenharia de software mundialmente;
- Esclarecer o lugar – e definir uma fronteira – da engenharia de software em relação a outras disciplinas.
- Caracterizar os conteúdos da disciplina de engenharia de software;
- Proporcionar acesso topificado do conjunto de conhecimento na área de Engenharia de Software.
- Prover uma base para desenvolvimento de um currículo, para certificação de profissionais e licenciamento de materiais.

Como resultado deste estudo, a Engenharia de Software foi dividida no SWEBOK em um total de onze áreas de conhecimento: requisitos, gerência de engenharia, projeto, métodos e ferramentas de engenharia, construção, processo de engenharia, testes, qualidade, manutenção, disciplinas relacionadas e gerência de configuração (KOSCIANSKI e SOARES, 2007, p. 38).

Cada área de conhecimento do SWEBOK pode ser subdividida em até dois níveis, catalogando os assuntos abordados em uma estrutura hierárquica. Na Figura 2 está representando a organização hierárquica referente a área de qualidade.

¹O IEEE é a maior associação profissional dedicada ao avanço da inovação tecnológica e excelência em benefício da humanidade. O IEEE e seus membros inspiram uma comunidade global por meio de publicações relevantes, conferências, padrões e atividades profissionais e educacionais.

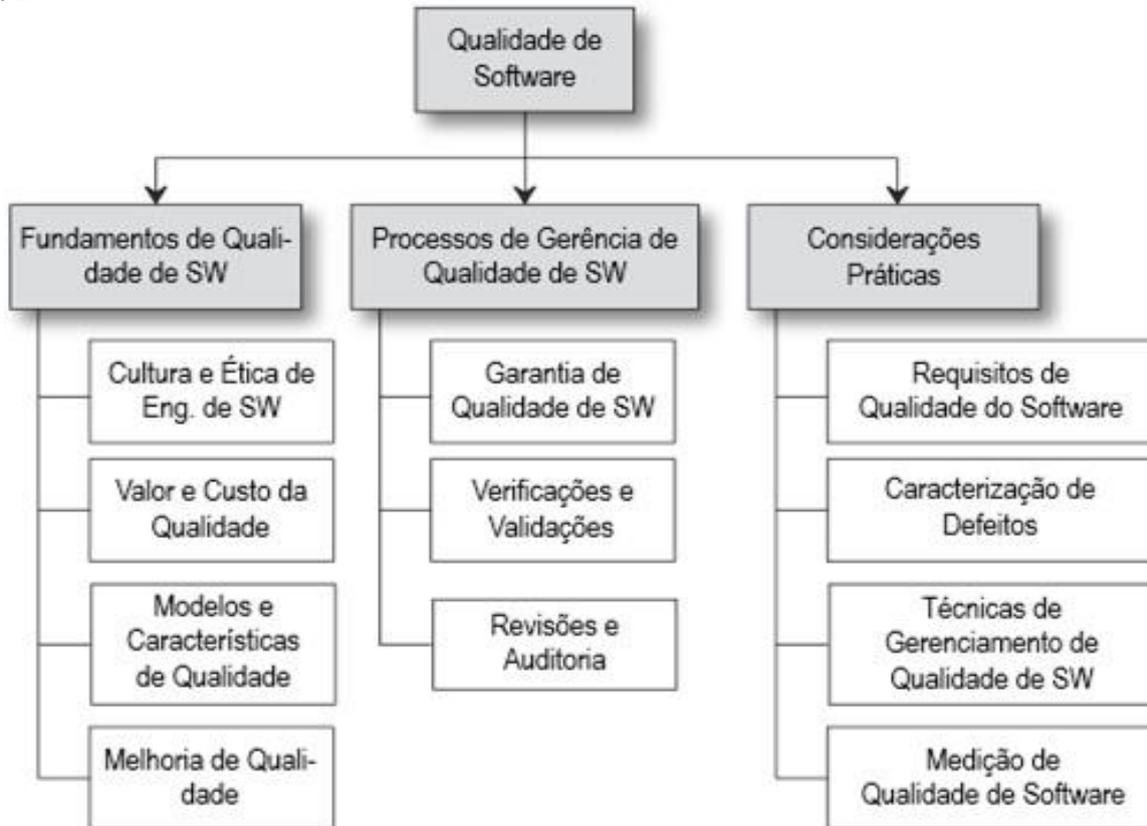


Figura 2 – Hierarquia da qualidade de software.

Fonte: SWEBOK, 2004.

3 TESTES DE SOFTWARE

3.1 Definição do conceito

Atualmente, existem diversas definições para esse conceito. O SWEBOK (2004) define o teste de software como “verificação dinâmica do comportamento de um programa em um conjunto finito de casos de teste contra o comportamento esperado”.

De acordo com Neto (2007, p. 54)

o conceito de teste de software pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. [...] De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos

Entretanto, para que seja iniciada uma discussão concreta sobre teste de software, é preciso esclarecer alguns conceitos que estão diretamente relacionados a essa atividade. Inicialmente, serão apresentados a seguir as definições de defeitos, erros e falhas, de acordo com o IEEE (1990), para que se possa perceber a diferença entre eles.

- **Defeito** é definido como um ato inconsistente cometido por um indivíduo ao tentar entender uma informação específica, para resolver um problema ou utilizar um método ou uma ferramenta;
- **Erro** é definido como uma manifestação concreta de um defeito em um artefato de software. É a diferença entre o valor obtido e o valor esperado. Ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro;
- **Falha** é definida como um componente operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros, e alguns erros podem nunca causar uma falha.

A partir da definição destes termos é possível inferir que os defeitos fazem parte do universo físico (a aplicação propriamente dita) e que podem ser causados por pessoas, pelo uso incorreto de uma tecnologia, por exemplo. E a ocorrência de defeitos pode propiciar a manifestação de erros em um produto, fazendo com que a construção de um software ocorra de forma diferente do que foi especificado (universo de informação).

Por fim, entende-se que os erros geram as falhas, que são comportamentos inesperados em um software e afetam diretamente o usuário final da aplicação (universo do usuário), podendo assim inviabilizar a utilização de um software.

Na Figura 3 ilustração ilustrando os conceitos inseridos em seus respectivos universos:



Figura 3 – Representação gráfica de defeito, erro e falha.

Fonte: Autoria própria.

A realização de testes de software dentro de um processo de desenvolvimento de software é composta por alguns elementos essenciais, cujo objetivo é auxiliar na formalização desta atividade. Estes elementos são descritos por alguns autores como:

- **Caso de Teste:** elemento que descreve uma condição particular a ser testada. É composto por valores de entrada e restrições para a sua execução, bem como um resultado ou comportamento esperado (CRAIG e JASKIEL, 2002).
- **Procedimento de Teste:** é uma descrição do passo-a-passo que deve ser realizado para executar um caso de teste (CRAIG e JASKIEL, 2002).

- **Critério de Teste:** elemento que tem como função selecionar e avaliar casos de teste de modo que as possibilidades de provocar falhas seja aumentada ou, se isso não for possível, deve estabelecer um nível de confiança elevado na correção do produto (ROCHA et al., 2001).

Entretanto, é preciso ressaltar que os processos de planejamento e execução dos testes devem ocorrer em diferentes níveis, paralelamente ao desenvolvimento do software, conforme o Modelo V proposto por Craig e Jaskiel (2002), ilustrado na Figura 4.

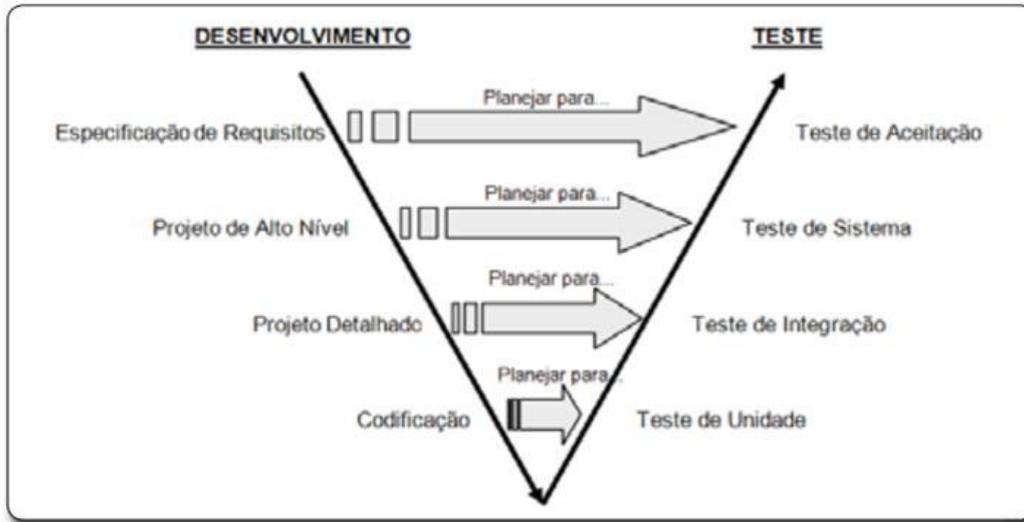


Figura 4 – Modelo V representando o paralelismo entre atividades de desenvolvimento e testes de software.

Fonte: Craig e Jaskiel, 2002.

No Modelo V apresentado, o planejamento e a execução dos testes devem ocorrer de cima para baixo, tendo início com o teste de aceitação, que é planejado a partir do documento de requisitos. A seguir, o teste de sistema é planejado a partir do projeto de alto nível do software, conforme ilustrado na figura. Em seguida, ocorre o planejamento dos testes de integração a partir o projeto detalhado, e por fim, acontece o planejamento dos testes de unidade de acordo com a codificação.

O teste de usabilidade é um conjunto de técnicas que são aplicadas com o intuito de identificar funcionalidades que não estão de acordo com os requisitos especificados durante o planejamento do projeto. A principal vantagem de se aplicar o teste de usabilidade é que não se faz necessário ter acesso ao código do projeto, sendo necessário entender os requisitos e analisar os resultados produzidos (BARTIE, 2002).

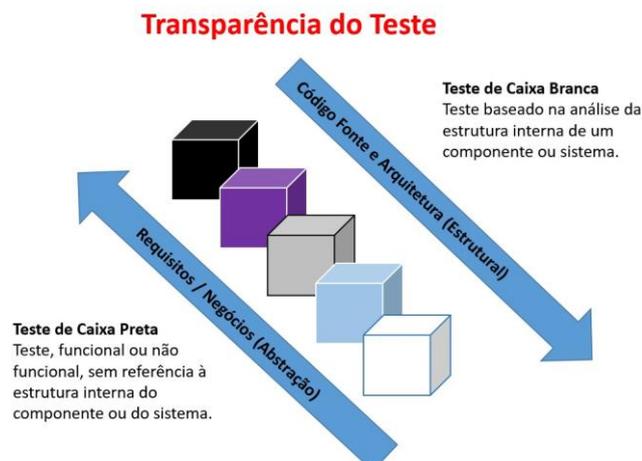


Figura 5 –Modelo de transparência do teste.

Fonte: Adaptado de Ribeiro, 2010.

4. Teste de usabilidade em um sistema acadêmico

Para este projeto, foi aplicada a técnica de teste usabilidade no sistema SGA (Sistema de Gerenciamento Acadêmico), se utilizando da métrica de particionamento de equivalência, técnica que consiste em analisar os possíveis retornos que o sistema pode gerar de acordo com as entradas do usuário. As partições se dividem em partições de equivalência para valores válidos e inválidos.

4.1. Tela de Login

Nos primeiros testes, verificou-se o comportamento da aplicação ao tentar realizar login com as credenciais incorretas, foram considerados:

- a) matrícula inexistente,
- b) credenciais em branco,
- c) e-mail incorreto, e
- d) senha não preenchida.

Observou-se as mensagens de erro, conforme ilustradas nas Figuras 6, 7, 8 e 9, respectivamente.

Na Figura 6 buscamos simular o acesso ao sistema com uma matrícula inexistente, porém, com uma senha válida. O resultado foi o esperado, não foi possível fazer o acesso à aplicação. A mensagem retornada foi um pop-up com um ícone azul, localizado no canto superior direito, indicando que as credenciais passadas não são válidas.

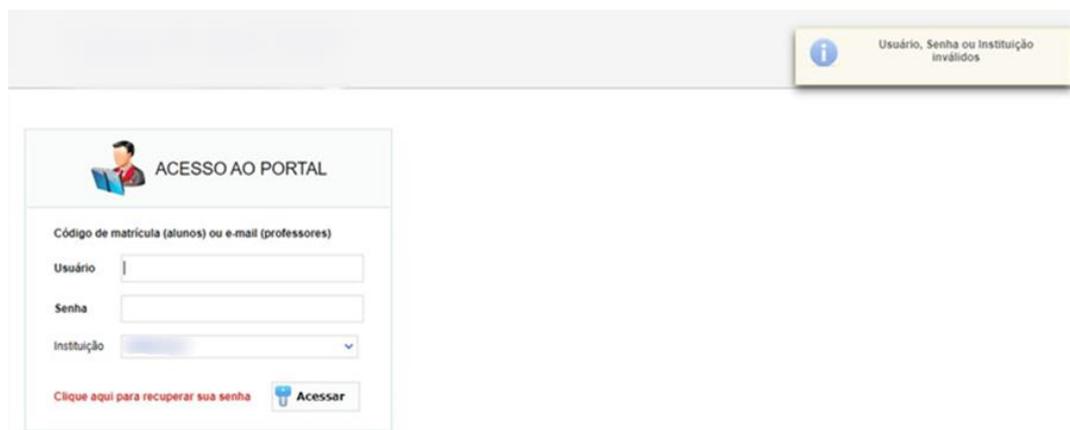
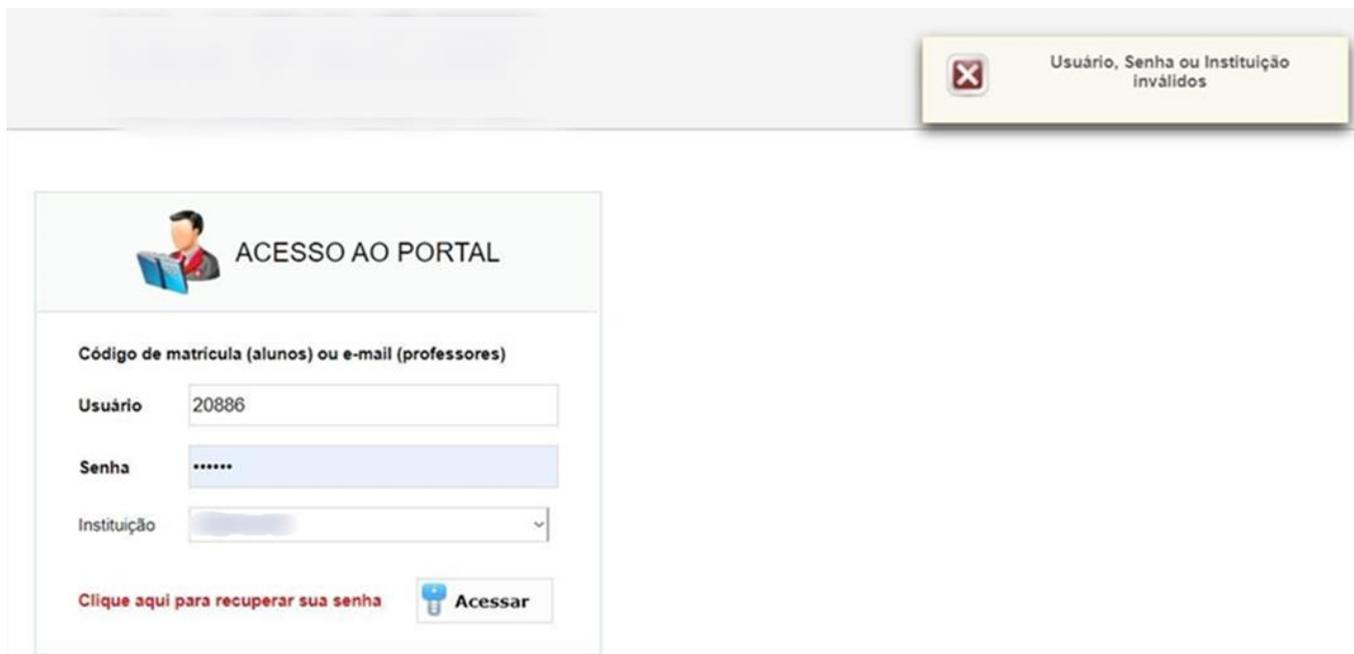


Figura 6 - Matrícula não existente.

Fonte: SGA Institucional

Na Figura 7 a tentativa de login foi feita sem o preenchimento dos campos de acesso, senha e usuário em branco. O retorno foi um pop-up com ícone em vermelho, que sai do padrão da mensagem retornada na Figura 6, porém o texto é o mesmo. O aconselhável seria manter o padrão dos ícones, para que o usuário fixe o padrão de erro de navegação.

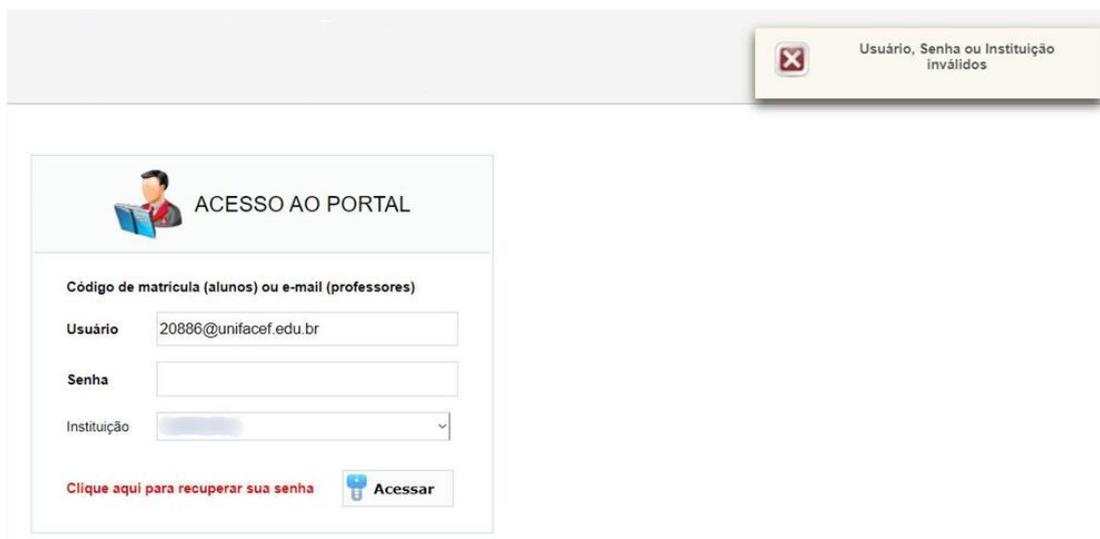


The screenshot shows a web interface for portal access. At the top right, a yellow error box with a red 'X' icon contains the text "Usuário, Senha ou Instituição inválidos". Below this, the main content area is titled "ACESSO AO PORTAL" and features a user icon. The form includes a label "Código de matrícula (alunos) ou e-mail (professores)" above three input fields: "Usuário" with the value "20886", "Senha" with masked characters "*****", and "Instituição" with a dropdown menu. At the bottom of the form, there is a red link "Clique aqui para recuperar sua senha" and a blue "Acessar" button with a lightbulb icon.

Figura 7 - Caixa alta.

Fonte: SGA Institucional.

O teste apresentado na Figura 8 foi uma tentativa de login com o e-mail institucional e senha, ambos válidos. A tentativa não obteve êxito, retornando a mesma mensagem da Figura 8.



This screenshot is similar to Figure 7, showing the same login form and error message. The "Usuário" field now contains the email address "20886@unifacef.edu.br". The "Senha" field is empty. The error message "Usuário, Senha ou Instituição inválidos" remains at the top right.

Figura 8 – Email incorreto.

Fonte: SGA Institucional

Simulação de teste na Figura 9 foi feito tentando realizar o login sem o preenchimento da senha de usuário. Retorno foi o mesmo do teste da Figura 6.

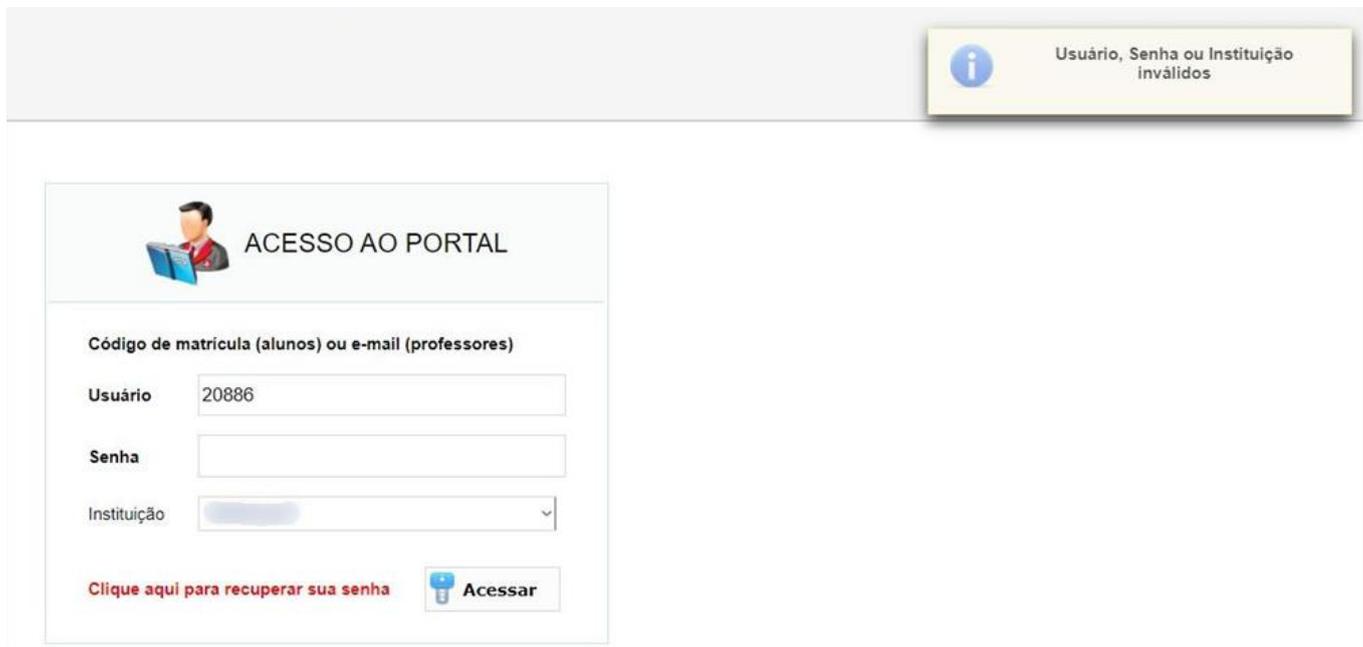


Figura 9 – Senha.

Fonte: SGA Institucional

Conforme foi observado, em nenhuma das simulações de erro foi possível fazer o login. A aplicação basicamente só emite uma mensagem de erro, não deixando claro para o usuário qual a credencial de acesso está incorreta. Isto pode significar um fator de segurança, pois informar o usuário qual está errado, pode ajudá-lo a quebrar uma senha, por exemplo, uma tentativa de *sql injection*. Também foi observado que ao tentar fazer login sem senha, o design da mensagem de erro é diferente das outras tentativas, tendo o seu ícone de informação na cor azul e no restante das tentativas esse ícone é diferente e em vermelho.

4.1 TELA DE LOGIN		
Teste Simulado	Problema encontrado	Possível Solução
Matrícula Inexistente	Não foi possível acessar, após a tentativa um aviso com ícone azul foi apresentado dizendo: "Usuário, Senha ou Instituição está invalido."	Uma alteração no critério de tentativa de acesso com as credenciais em branco, pois como já citado é impossível que o usuário espere um resultado sem utilizar nenhum critério de login. Nos demais manter o mesmo aviso, porém padronizar o ícone de erro.
Credenciais em branco	Não foi possível acessar, após a tentativa um aviso com ícone vermelho foi apresentado dizendo: "Usuário, Senha ou Instituição está invalido."	
Email Incorreto	Não foi possível acessar, após a tentativa um aviso com ícone vermelho foi apresentado dizendo: "Usuário, Senha ou Instituição está invalido."	

Senha não preenchida

Não foi possível acessar, após a tentativa um aviso com ícone azul foi apresentado dizendo: "Usuário, Senha ou Instituição está inválido."

4.2. Módulo Financeiro

O próximo teste foi feito com base no fluxo de navegação do usuário pela aba de extrato financeiro. Nessa tela é possível o usuário visualizar e emitir as parcelas em aberto, ver as parcelas pagas, parcelas canceladas, parcelas em dívida ativa e parcelas renegociadas, os testes realizados nessa seção não apresentaram erros.

Além da visualização das parcelas, existe um botão de emissão do extrato financeiro, porém, essa opção não funciona, nenhum relatório é emitido pelo sistema, e nenhuma mensagem de erro ou de resposta é exibida pela solução, sendo essa uma função extremamente necessária, já que pode haver casos de alunos que necessitem desse documento para apresentar em uma declaração de imposto de renda ou caso um aluno bolsista necessite da amostragem desse documento para a instituição que custeie a sua matrícula.

Top right: jomgomes73@gmail.com, Ano Letivo: 2021

User: - JOÃO MARCELO GOMES CORREIA
ENGENHARIA DE SOFTWARE NOTURNO - 8º SEMESTRE

Visualizar todas mensuralidades

#	Vencido	Valor	Líquido	Dt.Pagto	Valor Pago	Boleto	#	#
12								
11								
10								
9								
8								
7								
6								
5								
4								
3								
2								
1								

LEGENDA:

- Parcelas Geradas
- Parcelas Canceladas
- Renegociação
- Dívida Ativa
- Parcelas em atraso
- Parcelas pagas

A impressão da 2ª via do boleto, na versão 9.0 do Internet Explorer, depende de uma atualização do plugin do Java.
Caso tenha dificuldades com a emissão da 2ª via do boleto, entre em contato com o suporte técnico do Banco do Brasil, pelo telefone 0800 729 0200.
Caso tenha problemas ao imprimir o boleto instale a última versão do Adobe Reader [aqui](#).

Figura 10 – Extrato Financeiro.

Fonte: SGA Institucional.

4.2 MÓDULO FINANCEIRO

Teste Simulado	Problema encontrado	Possível Solução
<p>Visualizar e emitir parcelas em aberto, parcelas pagas, parcelas canceladas, parcelas em dívida ativa e parcelas renegociadas.</p>	<p>A funcionalidade do botão de extrato financeiro não funciona, nenhum relatório é emitido, não é apresentado nenhuma mensagem de erro, onde é considerado uma função essencial para diversos fins financeiros.</p>	<p>Uma correção em todo cenário com melhoria nos problemas encontrados, fazendo com que todas as parcelas existentes sejam possíveis para visualização e emissão de um relatório em formato PDF por exemplo.</p>

4.3. Módulo de Biblioteca

A sequência de testes foi feita no módulo de Biblioteca da aplicação. Nesse módulo, os alunos podem pesquisar a disponibilidade da obra desejada e em qual plataforma a instituição tem para emprestá-lo ou disponibilizá-lo. O módulo é composto por 3 campos de texto e 1 campo de seleção, sendo eles: “Autor”, “Título”, “Assunto” e a guia de seleção “Pesquisar em”. Todos esses campos são utilizados para refinar a pesquisa pelo conteúdo, conforme Figura 11 a seguir.



Figura 11 – Biblioteca.

Fonte: SGA Institucional.

Mesmo que o usuário tente pesquisar sem preencher os campos de texto, o sistema retorna uma mensagem de erro não muito intuitiva como ilustrado na Figura 12. Somente a opção Livro em complemento com o preenchimento de algum dos campos de texto é capaz de retornar algo e a tabela será preenchida com o título do livro, quantidade de livros da instituição, quantidade de livros disponíveis para reserva e a opção de visualizar alguns dados da obra literária, ilustrado pela Figura 13. Caso a busca seja feita procurando por alguma obra que não possui registro o retorno é uma mensagem informando que nenhum livro foi localizado. Quando a busca retorna um livro, o sistema fornece um ícone de uma lupa que trás maiores informações e abordagens sobre o livro (Figura 14).

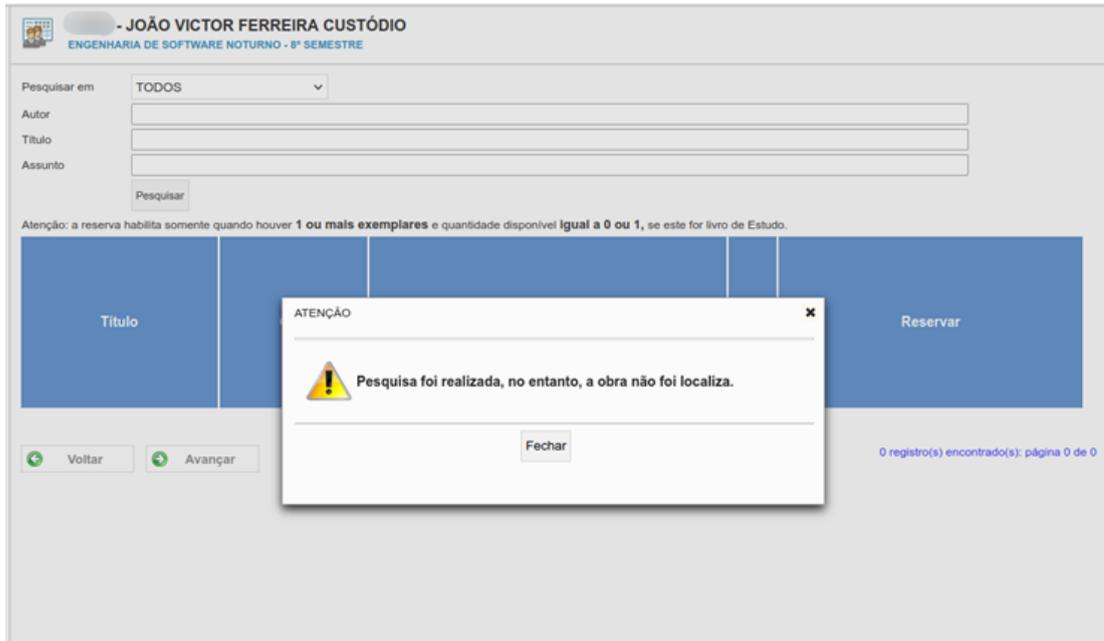


Figura 12 – Pesquisa da Biblioteca.

Fonte: SGA Institucional.



Figura 13 – Pesquisa da Biblioteca.

Fonte: SGA Institucional.

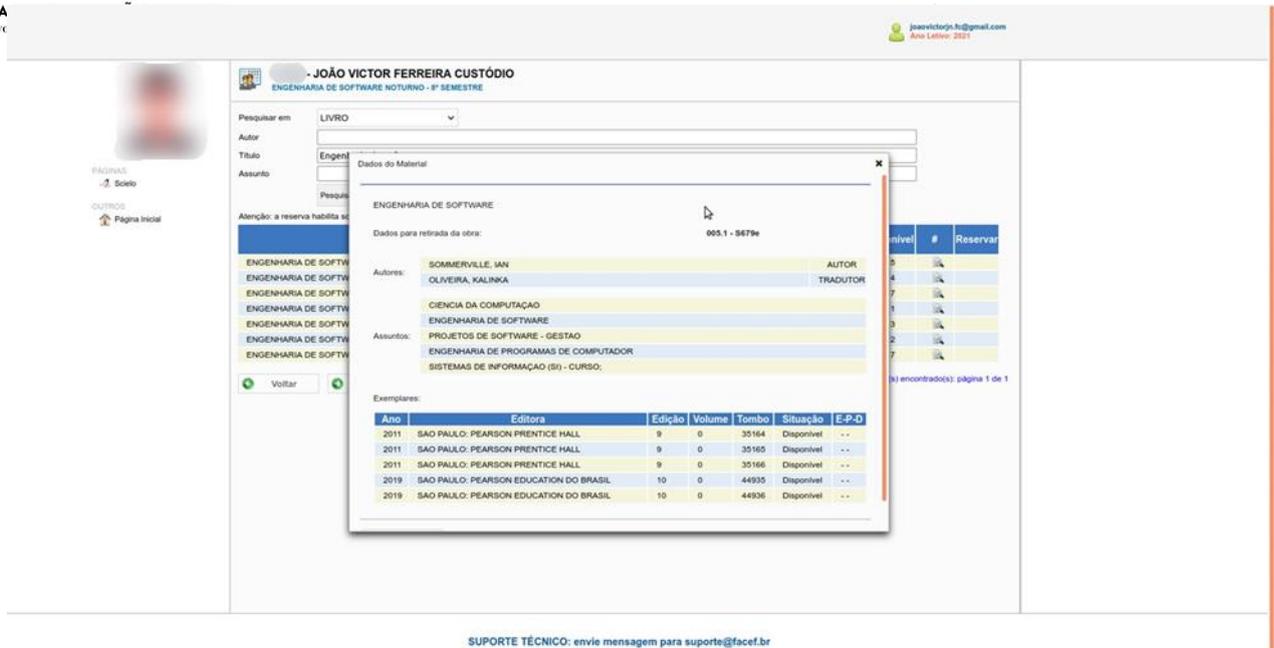


Figura 14 – Dados do material.

Fonte: SGA Institucional.

4.3 MÓDULO BIBLIOTECA		
Teste Simulado	Problema encontrado	Possível Solução
Tentativa de pesquisa sem a utilização de nenhum critério de pesquisa.	Erro esperado, não foi possível realizar a pesquisa e um alerta de erro com erro de ortografia foi emitido.	Reformulação do erro dessa tentativa de pesquisa informando ao usuário que não é possível realizar uma pesquisa sem a utilização de um parâmetro de pesquisa preenchido.

4.4. Módulo de Matrícula

Já no módulo de Matrícula, não foi possível fazer muitos testes já que a maioria dos campos de textos estão desabilitados para edição. Os campos que são possíveis alterar são os “Dados Profissionais” (Figura 15). Todas as alterações realizadas foram concluídas com êxito, sendo elas: Inserção de números e caracteres especiais nos campos, campos com valores nulos (em branco).

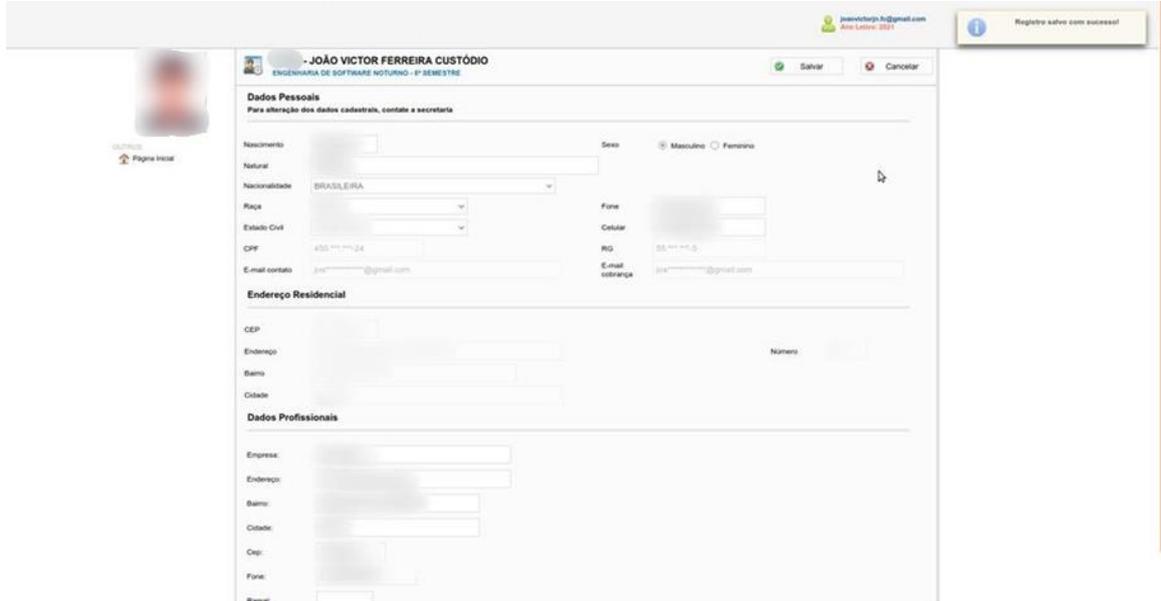


Figura 15 – Dados Profissionais.

Fonte: SGA Institucional.

4.5. Módulo de Notas

O módulo de Notas consiste em uma tela na qual o usuário pode acessar sua linha do tempo de notas e consultar seu boletim escolar, de todos os semestres já cursados e os semestres e as matérias pendentes ou em andamento, podendo verificar todas as matérias já cursadas e os status delas: Aprovado, Reprovado e Dependência. Os testes nesse módulo são bem básicos, o usuário pode mudar a listagem para o semestre desejado por meio de uma caixa de seleção que lista todos os semestres já finalizados e em andamento, como ilustrado na Figura 16, não sendo possível consultar não cursado. Há também um botão que emite esse boletim escolar do aluno, resultando em um relatório com as informações do semestre escolhido como na Figura 17. Caso o usuário necessite desse boletim impresso o resultado será fornecido no modelo da Figura 18.

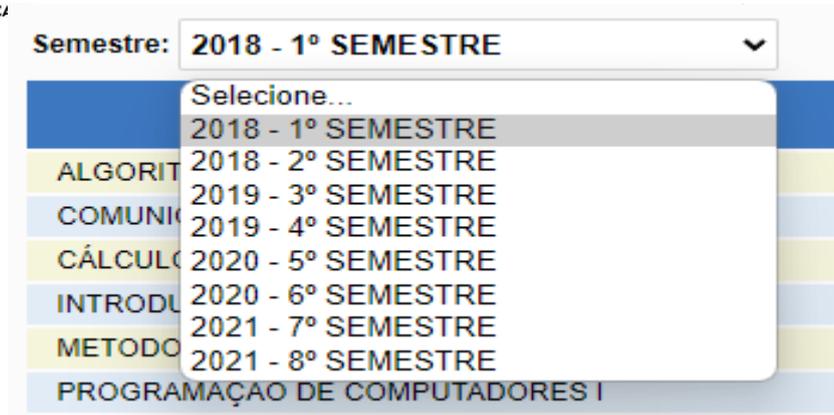


Figura 16 – Caixa de seleção dos anos letivos.

Fonte: SGA Institucional.

Disciplina	1ºN	2ºN	SUB	MED	Falta 1	Falta 2	%	Cursar DP/AD/CH	SITUAÇÃO
ALGORITMO E LÓGICA DE PROGRAMAÇÃO I									APROVADO
COMUNICAÇÃO E EXPRESSÃO I									APROVADO
CÁLCULO DIFERENCIAL E INTEGRAL									APROVADO
INTRODUÇÃO À COMPUTAÇÃO									APROVADO
METODOLOGIA DE PESQUISA CIENTÍFICA I									APROVADO
PROGRAMAÇÃO DE COMPUTADORES I									APROVADO

Figura 17 – Boletim do Aluno.

Fonte: SGA Institucional

BOLETIM DE NOTAS									
ANO LETIVO: 2018									
Curso: ENGENHARIA DE SOFTWARE - NOTURNO					Série: 1º SEMESTRE				
Nome: JOÃO VICTOR FERREIRA CUSTÓDIO					Matrícula: [REDACTED]				
DISCIPLINAS	1ª NOTA	2ª NOTA	SUB	FALTA 01	FALTA 02	MÉDIA	FREQ	Cursar DP/AD/CH	SITUAÇÃO
ALGORITMO E LÓGICA DE PROGRAMAÇÃO I									APROVADO
COMUNICAÇÃO E EXPRESSÃO I									APROVADO
CÁLCULO DIFERENCIAL E INTEGRAL									APROVADO
INTRODUÇÃO À COMPUTAÇÃO									APROVADO
METODOLOGIA DE PESQUISA CIENTÍFICA I									APROVADO
PROGRAMAÇÃO DE COMPUTADORES I									APROVADO
Observações									20/10/2021 18:35:34

Figura 18 – Boletim do Aluno Impresso.

Fonte: SGA Institucional.

Este estudo permitiu ter uma melhor noção da importância que os testes de software têm em determinados casos, como o teste que foi realizado na plataforma do SGA de uma instituição de ensino superior, já que os métodos de melhoria de processos de software no mercado buscam a qualidade e a organização dos processos da empresa, o que leva ao aprimoramento da plataforma.

A conciliação entre teoria e prática foi uma das partes mais importantes no desenvolvimento deste trabalho, já que todo o referencial teórico influencia diretamente no desenvolvimento dos testes aplicados no estudo de caso. Este critério em si foi o motivo da escolha do tema, já que uma vez analisada a plataforma SGA, foi possível identificar que se encaixaria no cenário de teste de usabilidade no decorrer do projeto.

Para melhorar os resultados encontrados no estudo de caso, sugere-se que a instituição possa analisar os problemas aqui apresentados e corrigi-los.

É importante concluir o impacto positivo que a experiência de realização dos testes trouxe para os atores deste trabalho, que puderam aprender como a técnica de caixa preta pode ajudar a melhorar a qualidade de um software.

6 REFERÊNCIAS

- ANTONIO, N. S.; TEIXEIRA, A; ROSA, A. **Gestão da qualidade: de Deming ao modelo de excelência da EFQM**. 2 ed. Lisboa: Edições Sílabo, 2007.
- BARTIÉ, A. **Garantia da qualidade de software**. Gulf Professional Publishing, 2002.
- CAMPOS, V. F. **TQC: Controle da qualidade no estilo japonês**. Belo Horizonte: Fundação Christiano Ottoni, Escola de Engenharia da UFMG, 1992.
- CRAIG, R. D.; JASKIEL, S. P. **Systematic Software Testing**. Artech House Publishers, Boston, 2002.
- DEMING, W. E. **Os 14 pontos da Qualidade**. Rio de Janeiro. Publicações Diversas, 1998.
- FIGUEIREDO, M. A. V.; FERNANDES, A. A. **O Método de Taguchi no Planejamento e Análise de Operações de Soldadura**. Ingenium – Revista Ordem dos Engenheiros, 11:79-85, 1996.
- IEEE Standard 610-1990: **IEEE Standard Glossary of Software Engineering Terminology**, IEEE Pres, 1990.
- IEEE. **Guide to the Software Engineering Body of Knowledge (SWEBOK)**. 2004 Version. Disponível em: < <http://www.swebok.org> >. Acesso em: ago. 2021.
- INTHURN, Cândida. **Qualidade e teste de software**. Florianópolis: Visual Books, 2001.
- JURAN, J. M.; GRZYNA, F. M. **Juran – Controle da Qualidade Handbook**. Pirelli Makro, 1988.
- LOBOS, J. **Encantando o cliente Externo e Interno**. 7 ed. São Paulo. 1993.
- NETO, A. C. D. **Introdução a teste de software**. Engenharia de Software Magazine, v. 1, p. 22, 2007. Disponível em: <https://www.researchgate.net/profile/Arilo_Neto/publication/266356473_Introducao_a_Teste_de_Software/links/5554ee6408ae6fd2d821ba3a/Introducao-a-Teste-de-Software.pdf>. Acesso em: ago. 2021.
- PINTO, J. C. **A importância da certificação de sistemas de gestão da qualidade em Portugal**. Revista de Gestão dos Países de Língua Portuguesa, v. 10, n. 1-2, p. 48-61, 2011. Disponível em: <<http://www.scielo.mec.pt/pdf/rpbg/v10n1-2/v10n1-2a06.pdf>>. Acesso em: ago. 2021.
- PRESSMAN, Roger. **Engenharia de software**. São Paulo: Makron Books, 1995.
- RAMOS, A. W. **Metodologia Seis Sigma**. Escola Técnica de São Paulo. São Paulo, 2009. Disponível em: <http://www.vanzolini.org.br/download/Metodologia_Seis_Sigma.pdf>. Acesso em ago. 2021.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. **Qualidade de software – Teoria e Prática**. Prentice Hall, São Paulo, 200.