

Sistema de Agendamento de Serviços Acadêmicos

Antonio Cauê Oliveira Morais
UFERSA

Pau dos Ferros, Brasil
antonio.morais36234@alunos.ufersa.edu.br

Cristiana de Paulo
UFERSA

Pau dos Ferros, Brasil
cristiana.paulo@alunos.ufersa.edu.br

Eriky Abreu Veloso
UFERSA

Pau dos Ferros, Brasil
eriky.veloso@alunos.ufersa.edu.br

Francisco Renan Leite da Costa
UFERSA

Pau dos Ferros, Brasil
francisco.costa57942@alunos.ufersa.edu.br

Geísa Morais Gabriel
UFERSA

Pau dos Ferros, Brasil
geisa.gabriel@ufersa.edu.br

Lavinia Dantas de Mesquita
UFERSA

Pau dos Ferros, Brasil
lavinia.mesquita@alunos.ufersa.edu.br

Lívia Beatriz Maia de Lima
UFERSA

Pau dos Ferros, Brasil
livia.lima30332@alunos.ufersa.edu.br

Maria Lanuza dos Santos Silva
UFERSA

Pau dos Ferros, Brasil
maria.silva44240@alunos.ufersa.edu.br

Tiago Amaro Nunes
UFERSA

Pau dos Ferros, Brasil
tiago.nunes@alunos.ufersa.edu.br

Resumo—O resumo deve responder os seguintes pontos:

- Qual o contexto?
- Qual o problema?
- Qual a relevância?
- Qual a sua contribuição?
- Quais as conclusões (os achados)?

Index Terms—key1, key, ..., keyn

I. INTRODUÇÃO

Do processo empregado para melhorar a qualidade dos produtos e a produtividade no desenvolvimento do software, define-se Verificação e Validação (V&V) [4]. Por meio da verificação, é possível avaliar um sistema a fim de analisar se os artefatos produzidos em determinada fase do desenvolvimento correspondem às condições impostas anteriormente. Já no que diz respeito ao processo de avaliar se um determinado sistema, no final do seu desenvolvimento, cumpre com os requisitos estabelecidos, conceitua-se validação [5].

Este artigo aborda as etapas necessárias durante o desenvolvimento de sistemas, assim como a definição e processo das atividades de V&V ao longo da construção de softwares. Nesse sentido, a partir do estudo de viabilidade definido para a implantação web do Sistema de Agendamento de Serviços Acadêmicos, faz-se necessário aplicar estratégias de V&V que contribuam para o aperfeiçoamento e qualidade da tecnologia criada.

Assim, durante este estudo, serão definidas técnicas de controle e garantia de qualidade, bem como ferramentas de suporte ao processo de desenvolvimento de software, considerando não apenas aspectos funcionais do sistema, mas também quesitos de avaliação para um adequado gerenciamento e qualidade de um software.

II. FUNDAMENTAÇÃO TEÓRICA

Esta seção, possui como finalidade expor os conceitos necessários para o melhor entendimento do assunto abordado.

Dessa forma, é apresentada a definição de sistemas web; logo após, são relatados aspectos associados à qualidade de software; depois disso é definido o conceito e ramificações a respeito do teste de software. Além disso, são apresentadas a documentação realizada na elaboração de sistemas, bem como as tecnologias e ferramentas utilizadas.

A. Sistemas Web

Um sistema Web é, normalmente, constituído por um conjunto de páginas Web e por uma base de dados [3]. Esses sistemas utilizam a arquitetura cliente-servidor. Neles, a comunicação do cliente no navegador ocorre através do endereço IP (*Internet Protocol*) do servidor, responsável por hospedar a estrutura de dados [7].

Nesse sentido, espera-se desenvolver um sistema Web de agendamento de serviços acadêmicos para a Ufersa, Campus Pau dos Ferros, que facilite a disponibilidade dos serviços ofertados por meio da integração e automatização do processo de agendamento. Além disso, deve-se aplicar técnicas de Verificação e Validação de Software a fim de incluir atividades de garantia da qualidade do software.

B. Qualidade de software

Historicamente, empresas perdiam bilhões de dólares em software devido a más implementações de funcionalidades. Sob esse aspecto, a qualidade de software refere-se à avaliação de um produto que atenda aos requisitos solicitados pelo cliente. O conceito de qualidade estende-se, portanto, em dois segmentos: qualidade do processo e qualidade do produto [14].

Nesse sentido, a garantia da qualidade de software engloba atividades de apoio aos processos, a fim de construir produtos adequados para o uso pretendido. Logo, a área de garantia da qualidade empenha-se em verificar os processos e suas definições, bem como validar o produto final a partir de um ciclo de revisões, melhoria e adaptação contínua [14].

C. Teste de software

Em sistemas de software cada vez mais complexos, a prática de testes para a descoberta de inconsistências torna-se cada vez mais necessária. Assim, a aplicação de testes, além de identificar a presença de erros, é fundamental para a garantia de qualidade e validação dos requisitos de um sistema de software [13].

Na realização de testes, cria-se um conjunto de casos de teste, ou seja, uma gama de condições que descrevem os comportamentos a serem testados no software a partir da análise das especificações do sistema [10]. Para tal, são estabelecidas condições necessárias a fim de determinar se o teste passou ou falhou.

Com base nisso, os testes podem ser definidos em dois aspectos gerais - a estrutura interna do programa e o seu aspecto funcional. Assim, os testes podem ser direcionados para o exercício da lógica dos componentes internos (teste de caixa branca) ou ainda podem ser voltados para descobrir erros no funcionamento, comportamento e desempenho do programa (teste de caixa preta) [10].

A organização dos testes é ainda efetuada em etapas, adotadas através das técnicas de teste de unidade, de integração e de sistema. Nesse viés, o teste unitário foca na garantia do funcionamento correto do componente do sistema. Do agrupamento entre os componentes origina-se o teste de integração. Após a integração, é realizado o teste de sistema, responsável por verificar o desempenho adequado da combinação dos elementos [11].

D. Documentação de sistemas

A documentação de software é essencial por dois motivos principais, uma das razões é facilitar a comunicação no processo de desenvolvimento do projeto e outra para esclarecer o conhecimento do programa nas atividades de manutenção (Ambler apud de [2]). Para serem elaborados os mais variados casos de testes, é necessário o entendimento sobre aquilo que está sendo testado. A documentação atua, portanto, como uma facilitadora, esclarecendo precisamente ao testador o comportamento esperado do programa, bem como os possíveis desvios de fluxo.

E. Tecnologias e ferramentas

Devido à crescente complexidade dos softwares, irrompe a necessidade de sistematizar tarefas, a fim de torná-la menos suscetível ao erro humano e menos custosa [1]. Para tal, apresentam-se algumas tecnologias e ferramentas de apoio ao processo de Verificação e Validação.

1) *SonarLint*: SonarLint¹ é um *plugin* para IDE gratuito e de código aberto responsável por encontrar e corrigir problemas de codificação. A disposição para o uso da ferramenta está em garantir a qualidade do código e aumentar a produtividade na resolução de problemas.

¹Para mais informações: <https://docs.sonarsource.com/sonarlint/eclipse/>

A tecnologia possui suporte para mais de 20 linguagens e usa mais de 5000 regras de *Clean Code* específicas de linguagem que buscam identificar erros comuns de codificação, *bugs* e vulnerabilidades.

2) *NUnit*: O NUnit² é um *framework* de teste unitário para todas as linguagens .Net. No processo de elaboração e identificação dos testes, o NUnit usa atributos personalizados e fornece um conjunto de asserções como métodos estáticos da classe Assert.

Para realizar testes unitários utilizando o *framework*, o código de teste desenvolvido deve conter asserções capazes de demonstrar o correto funcionamento da funcionalidade testada. Posto isto, os principais tipos de asserções estão entre as de igualdade, comparação, condição, identidade e tipos.

3) *Selenium*: O Selenium³ é caracterizado como uma ferramenta para automação de testes de aplicação web. Sendo portátil e possuindo código aberto, o Selenium oferece suporte para diversos navegadores web, aplicações web e tecnologias.

O Selenium pode ser definido como um conjunto de diferentes ferramentas de software, cada qual com um objetivo específico a fim de auxiliar o processo de automação de testes baseado nas principais necessidades para testes em aplicação web [12].

4) *DevTools e Lighthouse*: O Chrome DevTools⁴ é um conjunto de ferramentas para desenvolvedores da Web integrado diretamente ao navegador Google Chrome. Com ele, é possível diagnosticar problemas em tempo real, o que colabora no tempo de criação e edição de sites.

O Lighthouse⁵ é uma ferramenta automatizada de código aberto incorporada ao DevTools e criada para melhorar a qualidade das páginas da Web. Por meio da execução dos testes usando a ferramenta, é possível gerar um relatório sobre o desempenho da página web.

5) *Code Contracts*: O Code Contracts⁶ é uma ferramenta de plataforma .NET, que permite a incursão de contratos no código-fonte e falcitando a verificação formal, com o objetivo de garantir que o software se comporte de maneira correta e previsível.

Ele introduz três tipos de contratos: Pré-condições, que especificam o que deve ser cumprido antes da execução de um método; Pós-condições: Que estabelecem o estado ideal de certos artefatos após a execução do método; e invariantes: Que garantem que certas condições permaneçam no mesmo estado durante o ciclo de vida de um objeto e validando a integridade do estado interno da classe.

Esses contratos são validados tanto no tempo de execução quanto na compilação, auxiliando na detecção de erros lógicos no sistema, além de servirem de documentação implícita, descrevendo o comportamento esperado sem anotações externas.

²Para mais informações: <https://nunit.org/>

³Para mais informações: <https://www.selenium.dev/documentation/>

⁴Para mais informações: <https://developer.chrome.com/docs/devtools?hl=pt-br>

⁵Para mais informações: <https://developer.chrome.com/docs/lighthouse/overview?hl=pt-br>

⁶Para mais informações: <https://learn.microsoft.com/en-us/dotnet/framework/debug-trace-profile/code-contracts>

III. TRABALHOS RELACIONADOS

O trabalho redigido pelos autores [8], é referente a elaboração de um software com intuito da informatização de processos e gestão do setor de Assistência Estudantil. No decorrer do artigo é destacado o processo de criação e desenvolvimento do software com propósito de otimizar os serviços relacionado a moradia estudantil.

Análogo, o trabalho de Helen [6], aborda uma análise de impacto em utilizar softwares para a automatização dos processos de gestão acadêmica. Nessa perspectiva, a autora ainda destaca, que um dos problemas da evasão dos alunos na adesão dos serviços é a demora pela conclusão do processo.

Ambos trabalhos demonstram a importância da automatização dos processos vinculados a gestão acadêmica, assim percebem-se alguns pontos em comum deste trabalho aos demais. Primeiramente por se tratar de temas de gestão acadêmica, sendo neste trabalho um nicho mais aprofundado em serviços relacionado a psicologia, nutrição e afins. Segundo, por se tratar da automatização dos processos e efetividade dos serviços realizados na universidade.

IV. ABORDAGEM

Este estudo adota uma abordagem de cunho qualitativo com a finalidade da construção e análise do Sistema de Agendamento de Serviços Acadêmicos (ASA).

Para isso, foi utilizado o método de pesquisa exploratória com intuito de explorar um problema para propor soluções e conhecimento acerca da problemática [9]. Dessa forma, o trabalho parte inicialmente da criação da identidade visual e padronização de informações do sistema, com o Manual de Identidade Visual e Prototipagem do Software.

Posteriormente, criou-se a documentação do sistema com finalidade de registrar as informações que caracterizam o software em sua totalidade. A partir disso, foi elicitado os requisitos funcionais e não funcionais, realizado a modelagem do sistema por meio de diagramas de caso de uso e de classe e também realizado a modelagem do banco de dados.

Sobre o modelo do sistema, foi designado o uso do padrão de projeto criacional *Singleton* e a linguagem de programação C# utilizando para o back-end o framework ASP.NET Core Web API e para o front-end o framework Blazor WebAssembly, diante de uma arquitetura cliente-servidor. Para a persistência dos dados foi utilizado o banco de dados MySQL e o Entity Framework Core, onde permitiu com essa estrutura criar e manter o sistema.

Para garantir o controle de qualidade, foi elaborado um plano de teste que inclui o escopo, metodologia e objetivos que guiam a avaliação do sistema. Dessa forma, foram escolhidos os testes de caixa branca e de caixa preta para a verificação e validação das funcionalidades do ASA, em diferentes níveis de testes. Sendo assim, serão conduzidos testes de unidade, integração e de sistema com auxílio de ferramentas de apoio como NUnit, Selenium, SonarLint e outros.

Por fim, após a análise diante dos testes será realizado um relatório contendo os casos de teste efetuados, bem como os erros e/ou falhas identificados. Dessa maneira, o estudo tem

como principal característica a implementação e manutenção do sistema proposto.

V. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

REFERÊNCIAS

- [1] BOAS, A. *Gestão de configuração para teste de software*. PhD thesis, Dissertação de Mestrado, FEEC/UNICAMP, Campinas, SP, 2003.
- [2] DE SOUZA, S. C. B., DAS NEVES, W. C. G., ANQUETIL, N., AND DE OLIVEIRA, K. M. Documentação essencial para manutenção de software ii. In *IV Workshop de Manutenção de Software Moderna (WMSWM)*, Porto de Galinhas, PE (2007).
- [3] DELAMARO, M. E., MALDONADO, J. C., AND JINO, M. *Introdução ao Teste de Software*. Elsevier, 2016.
- [4] HIRAMA, K. *Engenharia de Software*. Grupo GEN, 2011.
- [5] IEEE. Ieee standard glossary of software engineering terminology. Standard ANSI/IEEE Std 610.12-1990, IEEE, New York, 1990.
- [6] MARQUES, H. M. F. *Gestão em processos acadêmicos: estudo de caso de uma Instituição de Ensino Superior privada na cidade de São Luis—Maranhão-Brasil*. PhD thesis, Escola Superior de Educação João de Deus, 2019.
- [7] MARTHA, L. F. *Desenvolvimento de uma aplicação web para modelagem colaborativa*. PhD thesis, PUC-Rio, 2022.
- [8] MELO, E. C., DA SILVA, G., AND CRUZ, M. A. Gestão informatizada de um setor de assistência estudantil: Um estudo de caso no ifmg-campus. *International Journal of Management-PDVG 1*, 1 (2021).
- [9] MORESI, E., ET AL. Metodologia da pesquisa. *Brasília: Universidade Católica de Brasília 108*, 24 (2003), 5.
- [10] PRESSMAN, R. S. *Engenharia de software*, 7 ed. AMGH, Porto Alegre, 2011.
- [11] PRESSMAN, R. S., AND MAXIM, B. R. *Engenharia de software*. McGraw Hill Brasil, 2016.
- [12] SANTORI, R. P., ET AL. Avaliação da ferramenta de testes selenium no desenvolvimento guiado por teste de uma aplicação web. *Universidade de Brasília* (2019).
- [13] SOMMERVILLE, I. *Engenharia de software*. Pearson Universidades, 2011.
- [14] ZANIN, A., JÚNIOR, P. A. P., AND ROCHA, B. C. *Qualidade de software*. Grupo A, 2018.