

Sistema para Gestão de Estágios Supervisionados

Allyson Bruno de Freitas Fernandes
UFERSA

Pau dos Ferros, Brasil
allyson.fernandes@alunos.ufersa.edu.br

Emson da Silva França
UFERSA

Pau dos Ferros, Brasil
emson.franca@alunos.ufersa.edu.br

Eriky Abreu Veloso
UFERSA

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

Iverton Emiquison Ribeiro de Bessa
UFERSA

Pau dos Ferros, Brasil
iverton.bessa@alunos.ufersa.edu.br

Matheus Marques Nunes
UFERSA

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

Resumo—O acompanhamento de estágios supervisionados obrigatórios é um processo essencial na formação de estudantes de graduação. Contudo, em muitas instituições de ensino superior, esse processo ainda é marcado por burocracias, registros manuais e ausência de ferramentas digitais eficazes. Diante desse cenário, o projeto Registra surge como uma solução para modernizar e automatizar a gestão de estágios no ensino superior, especialmente no contexto da Universidade Federal Rural do Semi-Árido (UFERSA). Este artigo, elaborado no âmbito da disciplina de Gerência de Configuração e Mudanças, apresenta a documentação técnica e a estrutura de desenvolvimento do sistema Registra, evidenciando práticas fundamentais como versionamento, rastreabilidade de requisitos, controle de alterações e registro de artefatos. O estudo também destaca o uso de testes e ferramentas de apoio à qualidade do software, adotados para garantir a consistência do produto ao longo de sua evolução. Embora o sistema ainda esteja em construção, a organização arquitetônica e a base documental demonstram seu potencial para contribuir de forma significativa com a transformação digital da gestão de estágios. Assim, o artigo justifica-se por integrar teoria e prática, oferecendo um exemplo concreto de aplicação dos princípios da engenharia de software em uma solução real e relevante.

Index Terms—Estágio supervisionado, Gestão acadêmica, Sistema Web.

I. INTRODUÇÃO

A gestão de estágios supervisionados obrigatórios é uma exigência curricular comum no ensino superior brasileiro, sendo parte fundamental da formação prática dos discentes. No entanto, em muitas instituições, esse processo ainda é conduzido manualmente, o que resulta em burocracia excessiva, descentralização das informações e maior propensão a erros [6]. Diante disso, torna-se relevante propor soluções informatizadas que automatizem e centralizem o controle de frequência, validação e acompanhamento dessas atividades.

A construção de sistemas confiáveis, por sua vez, exige mais do que a implementação de funcionalidades: requer organização, versionamento e controle sistemático das alterações realizadas ao longo do ciclo de vida do software. Nesse contexto, a disciplina de *Gerência de Configuração e Mudanças* (GCM), prevista na matriz do curso de Engenharia de Software da UFERSA, fundamentou a elaboração do presente projeto. Segundo Assis et al. [1], a gerência

de configuração é essencial para garantir rastreabilidade e consistência durante o desenvolvimento, reduzindo o risco de retrabalho e facilitando a manutenção evolutiva do sistema.

Este trabalho apresenta o desenvolvimento do Registra, um sistema web destinado ao controle de frequência e à gestão de estágios supervisionados no âmbito da UFERSA – Campus Pau dos Ferros. O sistema foi concebido a partir de práticas da Engenharia de Software, com foco na documentação técnica, no controle de versões e na validação das funcionalidades. Durante sua construção, foram elaborados artefatos como protótipos, modelagem de dados, diagramas de classe e casos de uso, além da aplicação de testes unitários, de integração e funcionais.

A proposta visa facilitar a comunicação entre alunos, supervisores e coordenadores, garantir o registro confiável das atividades e promover maior transparência no processo de validação dos estágios. Além disso, o Registra reforça a importância da adoção de práticas de controle de mudanças em projetos reais, servindo como exemplo aplicado dos conteúdos trabalhados na disciplina de GCP.

II. FUNDAMENTAÇÃO TEÓRICA

A. Sistemas Web

Um sistema web é uma aplicação de software que opera através da internet, acessível via navegador, sua arquitetura é cliente-servidor, no qual o navegador atua como cliente e um servidor hospeda a lógica e os dados da aplicação [8]. Dessa forma, propõe-se o desenvolvimento de um sistema web voltado ao controle de estágios supervisionados no ensino superior, visando integrar e automatizar os processos envolvidos, reduzindo a burocracia existente e promovendo maior transparência em todas as etapas.

B. Documentação de sistemas

A documentação de sistemas é um conjunto organizado de artefatos textuais e visuais que descrevem o funcionamento, a arquitetura, as interações e os processos de um software ao longo de seu ciclo de vida. A produção de uma documentação estruturada torna-se essencial para entender requisitos, processos e decisões técnicas [4].

Chomal e Saini [2] evidenciam que uma documentação incorreta, incompleta ou desatualizada é um dos principais fatores que impactam negativamente a qualidade e a confiabilidade do software, podendo levar a falhas, aumento de esforço para manutenção e dificuldades na integração de novos colaboradores. Dessa forma, a documentação do Registra torna-se uma atividade fundamental, permitindo que todos os envolvidos (desenvolvedores, usuários e gestores), compreendam o propósito, o funcionamento e as restrições do software.

C. Gerência de configuração

A gerência de configuração é um campo de estudo da Engenharia de Software cujo propósito é controlar sistematicamente todas as modificações nos artefatos do projeto, buscando garantir rastreabilidade e consistência ao longo do ciclo de vida do sistema. Buscando mitigar inconsistências e retrabalhos decorrentes de alterações desorganizadas, as atividades relacionadas a gerência de configurações de um software consiste em identificação dos itens de configuração, controle de versões, controle de mudanças, liberação de versões, entre outras tarefas [1].

Figueiredo, Santos e Rocha [3], reforçam que a adoção de práticas estruturadas de gerência é crítica para alcançar maturidade organizacional conforme normas como ISO/IEC 12207 e modelos como CMMI, assegurando qualidade e previsibilidade nas entregas. Dessa forma, compreende-se a importância de manter os artefatos do Registra atualizados, assegurando que as modificações e a evolução do sistema seja do conhecimento de todos os stakeholders. Além disso, é válido ressaltar a relevância do versionamento realizado na ferramenta GitHub, tornando-se essencial para o controle da evolução da aplicação.

D. Tecnologias e ferramentas

Segundo Pressman e Maxim [6], a adoção de frameworks e ambientes de apoio fortalece a estrutura de desenvolvimento e promove a criação de sistemas mais robustos e confiáveis. Para tal, apresentam-se algumas tecnologias e ferramentas de apoio ao processo de desenvolvimento do Registra, sendo estas de extrema importância por tratar-se de um sistema que manuseia dados pessoais.

1) *Next.js*: Next.js¹ é um *framework* baseado em React, projetado para a construção de aplicações web *full-stack*. O *framework* permite que desenvolvedores utilizem Componentes React para a criação de interfaces de usuário, fornecendo, adicionalmente, uma série de recursos e otimizações. Next.js também configura automaticamente ferramentas de baixo nível, como *bundlers* e *compilers*, permitindo que os desenvolvedores concentrem-se na construção do produto e na entrega rápida.

2) *Dotnet*: O .NET² constitui uma plataforma de desenvolvimento livre, multiplataforma e de código aberto, criada pela Microsoft para a construção de aplicações em diversas

escalas e dispositivos. Permite a construção de APIs robustas e seguras, além de possuir integração com múltiplos bancos de dados, como o PostgreSQL.

3) *Docker*: Docker³ é uma plataforma na qual fornece um método padronizado para empacotar aplicações juntamente com suas dependências completas, garantindo a portabilidade e a execução consistente em diversos ambientes computacionais, desde ambientes de desenvolvimento locais até infraestruturas de produção híbridas ou em nuvem.

4) *PostgreSQL*: PostgreSQL⁴ constitui um sistema de gerenciamento de banco de dados objeto-relacional de código aberto. Reconhecido por sua conformidade com uma ampla parte do padrão SQL e por oferecer diversos recursos, o PostgreSQL se destaca por sua arquitetura extensível, permitindo aos usuários a inclusão de novos tipos de dados, funções, operadores e métodos de acesso.

5) *xUnit*: xUnit⁵ constitui um framework de teste unitário projetada para a plataforma .NET. Inspirada no JUnit, pioneiro no conceito de frameworks de teste unitário para linguagens orientadas a objeto, o xUnit.net oferece uma abordagem moderna e extensível para escrita e execução de testes unitários em aplicações .NET. Ele permite a criação de testes modulares com foco em boas práticas de verificação de funcionalidades, buscando garantir a estabilidade do sistema.

III. TRABALHOS RELACIONADOS

O trabalho redigido pelos autores [7], apresenta o desenvolvimento do SIGESTAGIOS, um sistema web criado para gerenciar o estágio supervisionado obrigatório da Universidade Estadual do Piauí (UESPI), Campus Parnaíba. A motivação do projeto originou-se a partir da ausência de um sistema informatizado para organizar os processos relacionados ao estágio, no qual era anteriormente realizado manualmente, gerando grande volume de documentos físicos. O trabalho destaca a relevância da informatização na gestão de estágios, contribuindo para maior eficiência administrativa e redução de erros.

Analogamente ao trabalho supracitado, Leoncio [5] desenvolveu um sistema que possibilita o gerenciamento completo dos dados envolvidos no processo de estágio, incluindo cadastros de alunos, empresas, professores e defesas, bem como a geração automática de documentos oficiais exigidos ao final do estágio, como ata de defesa, banca de estágio, orientação de estágio e parecer do orientador. Nessa perspectiva, o autor destaca que o sistema demonstrou-se funcional e eficaz nos testes realizados, permitindo maior segurança e organização no controle de estágios.

Ambos trabalhos propõem soluções web para substituir processos manuais no gerenciamento de estágios, priorizando a centralização de dados e a segurança da informação. Realizando uma breve análise, é possível observar objetivos semelhantes ao software descrito no presente trabalho, sendo

¹Para mais informações: <https://nextjs.org/docs>

²Para mais informações: <https://learn.microsoft.com/pt-br/dotnet/>

³Para mais informações: <https://docs.docker.com/>

⁴Para mais informações: <https://www.postgresql.org/docs/>

⁵Para mais informações: <https://xunit.net/?tabs=cs>

o Registra, projetado com foco em aderência a normas institucionais e usabilidade, buscando atingir destaque perante outras plataformas para atender um maior número de universidades.

IV. ABORDAGEM

O sistema Registra foi desenvolvido com o propósito de informatizar, padronizar e automatizar o processo de controle de frequência e acompanhamento de estágios supervisionados obrigatórios no ensino superior. A proposta surgiu diante da constatação de que muitas instituições ainda utilizam práticas manuais, como planilhas, documentos impressos e comunicações descentralizadas, o que dificulta a rastreabilidade, a segurança das informações e a agilidade do processo. Assim, buscou-se propor uma solução web acessível, centralizada e segura, capaz de contemplar as necessidades de alunos, supervisores e coordenadores. Para alcançar esse objetivo, adotou-se uma abordagem iterativa e incremental, com entregas contínuas organizadas em sprints semanais. Essa estratégia envolveu revisões periódicas dos requisitos e a definição clara de papéis e responsabilidades entre os membros da equipe de desenvolvimento, conforme recomendam Pressman e Maxim [6].

Durante a fase inicial do projeto, foi realizada a elicitação dos requisitos funcionais e não funcionais por meio de reuniões e análise documental. Em seguida, foram definidas as regras de negócio com base na realidade institucional da UFERSA e no regulamento de estágios vigente. Essa etapa permitiu estabelecer uma base sólida para a tomada de decisões técnicas e para a definição da arquitetura geral do sistema.

A prototipação das interfaces foi conduzida utilizando a ferramenta Figma, o que permitiu representar visualmente os fluxos e validar, junto aos envolvidos, a experiência do usuário antes da implementação definitiva. Esse processo foi essencial para garantir maior alinhamento entre a proposta do sistema e as expectativas dos usuários finais. Ademais, foi utilizada a ferramenta Stark, visando realizar testes de acessibilidade e contraste de cores. Essa prática buscou alinhar o projeto desde suas fases iniciais às Diretrizes de Acessibilidade para Conteúdo Web (WCAG), promovendo uma interface mais inclusiva e adequada a usuários com diferentes necessidades visuais e cognitivas. A adoção antecipada dessas diretrizes contribui para a construção de um sistema mais acessível, minimizando retrabalho e ampliando seu alcance dentro da comunidade acadêmica.

A arquitetura do Registra foi concebida segundo o modelo cliente-servidor, que promove a separação entre a lógica de apresentação (no cliente) e a lógica de negócio e persistência (no servidor). Esse padrão é amplamente adotado em aplicações web modernas por favorecer a modularização e a escalabilidade da aplicação [8]. No back-end, foi adotado o padrão de arquitetura Domain-Driven Design (DDD), estruturando o sistema em camadas distintas: WebAPI, UseCases, Domain (com entidades, agregados e value objects), e Validators. Essa separação facilita a manutenção do código, a testabilidade e a compreensão dos fluxos de negócio.

A aplicação foi desenvolvida em C# com o framework .NET Core, uma plataforma moderna e de código aberto voltada para o desenvolvimento de APIs performáticas e multiplataforma [6]. Para persistência dos dados, utilizou-se o PostgreSQL, executado via Docker Compose utilizando imagem oficial, o que permitiu maior portabilidade e controle do ambiente de desenvolvimento. A autenticação foi estruturada com uso de tokens JWT (JSON Web Token), permitindo o gerenciamento seguro de sessões e acesso restrito aos recursos. A documentação e o teste dos endpoints REST foram facilitados com a integração do Swagger.

As validações de entrada foram implementadas com a biblioteca FluentValidation, permitindo aplicar regras de forma desacoplada e reutilizável. No front-end, a aplicação foi construída com o framework Next.js, aliando os benefícios do React com renderização híbrida e recursos avançados de roteamento. O uso de TypeScript contribuiu para a segurança do código e clareza nos contratos. A estilização foi feita com TailwindCSS, a comunicação com a API com Axios, e a manipulação dos formulários com React Hook Form, garantindo reatividade e validações dinâmicas.

As funcionalidades centrais do sistema foram planejadas para atender às principais demandas operacionais dos estágios supervisionados. O Registra permite o cadastro de usuários em três perfis distintos: aluno, supervisor e coordenador, com permissões específicas para cada tipo de operação. Além disso, o sistema oferece o cadastro de projetos de estágio, associados a empresas, supervisores e representantes legais. O aluno pode registrar suas frequências diariamente, informando data, horário de entrada e saída, descrição das atividades e anexando comprovantes digitais (fotos ou documentos). Esses registros ficam com status pendente até que o supervisor responsável revise e aprove (ou reprove com justificativa). Após o cumprimento da carga horária mínima, o coordenador realiza a validação final do estágio. Também é possível visualizar o progresso individual, com histórico de frequência e geração de relatórios mensais consolidados, contendo indicadores de presença e carga horária acumulada.

Para assegurar a confiabilidade do sistema, a validação da aplicação foi realizada com base nos requisitos especificados e em testes em múltiplos níveis. Os testes unitários foram responsáveis por verificar o comportamento isolado de funções críticas, como validação de horários, regras de permissão e consistência de dados. Utilizou-se o framework xUnit como base para os testes automatizados, junto com a biblioteca Moq para simulação de dependências, Bogus para geração de dados fictícios e FluentAssertions para tornar as asserções mais claras e legíveis. Já os testes de integração abordaram fluxos completos, como autenticação, registro de frequência e validações hierárquicas. A ferramenta reportgenerator foi utilizada para calcular a cobertura de testes, excluindo do escopo os módulos auxiliares de exceções e comunicação, a fim de obter métricas mais representativas da lógica de negócio.

Adicionalmente, foram aplicadas ferramentas de apoio para garantir qualidade e estabilidade. O SonarLint foi empregado

para análise estática de código, identificando inconsistências, duplicações e possíveis falhas de manutenção. O Lighthouse e o DevTools foram utilizados para testar a responsividade da aplicação e avaliar métricas de desempenho, acessibilidade e boas práticas em diferentes tamanhos de tela e navegadores. O projeto foi versionado por meio do GitHub, com organização em branches individuais e uso de pull requests revisados, seguindo boas práticas de colaboração recomendadas por Assis et al. [1].

A aplicação foi hospedada na plataforma Vercel, que possibilitou deploy contínuo com integração ao repositório GitHub. O pipeline de integração e entrega contínua (CI/CD) foi configurado com o uso do GitHub Actions, contemplando etapas de build da aplicação, execução de testes com geração de relatório, empacotamento da aplicação (.dll) e, por fim, deploy automatizado para o serviço Azure App Service, garantindo agilidade, rastreabilidade e estabilidade nas entregas.

Por fim, como parte da documentação técnica do projeto, foram elaborados diagramas de classe, casos de uso e modelo lógico de dados. Esses artefatos oferecem uma visão estruturada da aplicação e servem como suporte para futuras manutenções, atualizações e expansão da solução para outros contextos institucionais.

A combinação entre organização em camadas, uso de frameworks modernos, aplicação de testes em diferentes níveis e controle rigoroso de versão assegurou a consistência e a qualidade do sistema. O apoio de diagramas e documentação técnica contribuiu para a rastreabilidade dos requisitos e viabilizou a entrega de um produto alinhado às demandas da disciplina de Gerência de Configuração e Mudanças.

V. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O desenvolvimento do sistema Registra representou um avanço significativo na informatização do controle de frequência em estágios supervisionados obrigatórios, promovendo a substituição de métodos manuais por uma solução centralizada, segura e acessível. O sistema atendeu aos principais requisitos funcionais e não funcionais definidos durante a fase de levantamento e demonstrou estabilidade, organização e coerência nos fluxos aplicados aos perfis de alunos, supervisores e coordenadores.

Este projeto foi desenvolvido no contexto da disciplina de *Gerência de Configuração e Mudanças* (GCM), componente curricular do curso de Engenharia de Software da UFERSA. Ao longo da sua construção, foram aplicadas práticas de versionamento, controle de mudanças, documentação técnica, prototipação e testes, consolidando os conceitos abordados na disciplina em um sistema real. A adoção dessas práticas é recomendada como forma de garantir rastreabilidade e consistência no ciclo de vida do software [1].

Além da implementação das funcionalidades essenciais, o projeto foi sustentado por uma base documental estruturada, que inclui diagramas, protótipos, registros de requisitos e um plano de testes detalhado. Esses elementos reforçam a qualidade técnica do sistema e demonstram sua viabilidade para ser adotado ou adaptado por outras instituições.

Como trabalhos futuros, propõe-se expandir as funcionalidades relacionadas à visualização e exportação de relatórios, a fim de oferecer maior suporte à tomada de decisão por parte da coordenação. Também se prevê a realização de testes com usuários reais, com o objetivo de coletar feedbacks sobre usabilidade e possíveis ajustes de interface. Além disso, será dada atenção especial à correção de erros e inconsistências identificadas durante os testes estruturais e funcionais, com foco na estabilidade e confiabilidade do sistema.

A evolução contínua do Registra contribuirá para sua consolidação como uma ferramenta robusta e replicável, com potencial de atender às demandas de diferentes cursos e instituições de ensino superior.

REFERÊNCIAS

- [1] ASSIS, A., BRANDÃO, A., FRAZÃO, D., PALHANO, D., AYRES, F., AND SALLES, G. Gerência de configuração, 2008.
- [2] CHOMAL, V. S., AND SAINI, J. R. Significance of software documentation in software development process. *International Journal of Engineering Innovation Research* 3, 4 (2014), 410–416.
- [3] FIGUEIREDO, S., SANTOS, G., AND ROCHA, A. R. Gerência de configuração em ambientes de desenvolvimento de software orientados a organização. *Simpósio Brasileiro de Qualidade de Software* 3 (2004).
- [4] FRITOLA, R. G. Documentação de requisitos de sistemas legados: Uma proposta baseada na engenharia de requisitos orientada a objetivos. Monografia, Universidade Estadual do Oeste do Paraná, Cascavel, 2023. Ciência da Computação.
- [5] LEONCIO, D. Sistema para controle de estágio supervisionado do departamento acadêmico de informática. Monografia, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2018. Tecnologia em Análise e Desenvolvimento de Sistemas.
- [6] PRESSMAN, R. S., AND MAXIM, B. R. *Engenharia de software*. AMGH, 2021.
- [7] SOUZA, I. B. S. D., ROCHA, F. D. C., AND BALUZ, R. A. R. S. Sigestagios: Sistema web para controle de estágios supervisionados no ensino superior. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica* 10, 1 (2019), 172–192.
- [8] ZANETI JUNIOR, L. A., AND VIDAL, A. G. D. R. Construção de sistemas de informação baseados na tecnologia web. *Revista de Administração - RAUSP* 41, 3 (2006), 232–244.