

**CEFET/RJ - CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA**

Do Caos à Ordem: Reestruturando um Projeto Go Horse com Boas Práticas de Desenvolvimento

Gabriel Duarte Rodrigues Bastos

Prof. Orientador:

Diego Cardoso Borda Castro

**Rio de Janeiro,
Novembro de 2024**

**CEFET/RJ - CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA**

Do Caos à Ordem: Reestruturando um Projeto Go Horse com Boas Práticas de Desenvolvimento

Gabriel Duarte Rodrigues Bastos

Projeto final apresentado em cumprimento às
normas do Departamento de Educação
Superior do Centro Federal de Educação
Tecnológica Celso Suckow da Fonseca,
CEFET/RJ, como parte dos requisitos para
obtenção do título de Bacharel em Sistemas de
Informação.

Prof. Orientador:
Diego Cardoso Borda Castro

**Rio de Janeiro,
Novembro de 2024**

DEDICATÓRIA

AGRADECIMENTOS

Texto

“Só seria fácil se não fosse difícil”

(Nome do autor)

RESUMO

Texto

Palavras-chaves: Palavra; Palavra; Palavra; Palavra; Palavra

ABSTRACT

Texto

Keywords: Palavra; Palavra; Palavra; Palavra; Palavra

Sumário

1	Introdução	2
1.1	Contextualização	2
1.2	Problemas investigados	3
1.3	Objetivos	4
1.4	Metodologia	4
1.5	Organização do texto	5
2	Fundamentação Teórica	7
3	Revisão da Literatura / Trabalhos Relacionados	8
4	Proposta	9
5	Conclusão	10
	Referências	11

Lista de Figuras

FIGURA 1:	Fluxo metodológico utilizado e adaptado [Pimentel et al., 2020] . . .	5
-----------	---	---

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contextualização

O desenvolvimento de software configura-se como uma atividade essencial no cenário atual, no qual empresas procuram se desenvolver digitalmente para manter sua competitividade no mercado [Cardoso, 2019]. Nesse contexto, boas práticas tornam-se fundamentais para manter um projeto de qualidade, o que facilita sua escalabilidade e manutenção.

As boas práticas de desenvolvimento de software consistem em técnicas utilizadas antes e durante a fase de desenvolvimento, visando garantir a qualidade, escalabilidade, manutenibilidade e segurança [Braga, 2007]. Alguns padrões utilizados na indústria de sistemas da informação é o versionamento de código com git, testes automatizados, criação de códigos genéricos para reutilizar, documentação clara, padrões de design, revisão de código, criação de diagramas da UML e planejamento de arquitetura bem detalhada. Apesar da existência dessas boas práticas, nem todo software acaba utilizando, como é o caso de projetos *Go Horse* [Sturm and da Silva, 2014].

O termo *Go Horse* surgiu de maneira informal e humorística no meio do desenvolvimento de software. Essa expressão, principalmente no Brasil, descreve projetos que foram conduzidos de forma improvisada, sem um planejamento prévio adequado, testes e documentações. Essa prática prioriza entregas rápidas, entretanto com pouca qualidade, resultando assim em falhas no sistema, difícil manutenção e elevados riscos para falhas de segurança, indo contra os padrões de metodologias ágeis [Perry, 2016].

Este trabalho tem como objetivo de estudo o software livre Web Gerenciador para Instituições Assistenciais (WeGIA), que possui características típicas de um projeto *Go Horse* e servirá de base de estudo e aplicação de boas práticas de desenvolvimento.

O WeGIA foi desenvolvido pelos alunos do CEFET-RJ - campus Nova Friburgo. Sua criação surgiu pela falta de softwares *Open Sources*, cujo seu objetivo central é melhorar a gestão, controle e transparência das entidades públicas. Sua versão atual possui seis módulos, dentre eles temos contribuição e sócios, material e patrimônio, memorando, pessoas, pet e saúde.

[Lazarin et al., 2024]

1.2 Problemas investigados

Projetos conduzidos sob a abordagem *Go Horse* tendem a garantir sistemas que envelhecem rapidamente, tornando-se mais difícil de manter sua atualização e manutenção gradual. Esse fenômeno acaba gerando diversas soluções improvisadas durante o desenvolvimento. Aliada à falta de documentação, esses sistemas acabam se tornando rígidos, gerando um grau de complexidade elevado para a entrada de novos desenvolvedores na equipe e aumentando o risco de falhas a cada modificação realizada.

Embora não seja um conceito formal na literatura acadêmica, *Go Horse* se relaciona com problemas amplamente discutidos nos dias atuais, como **dívida técnica**, **antipadrões de software**, **código legado** e **ad hoc**. Compreender o impacto no desenvolvimento de software evidencia a necessidade de aplicar boas práticas de desenvolvimento, garantindo que um projeto improvisado se transforme em um sistema sustentável e de fácil manutenção.

O Wegia apresenta diversas características de um projeto com dívidas técnicas adquiridas no decorrer do tempo, tais como a falta de testes automatizados, tabelas sem normalização, código sem padrões estabelecidos, baixa segurança, documentação incompleta e uma arquitetura rígida. Essas características acabam impactando em retrabalho, demotivação da equipe e insatisfação de pessoas envolvidas.[Borges, 2022]

Por estar no mercado de código aberto, considera-se que novos usuários irão se utilizar e dar melhorias no projeto. Com isso, torna-se primordial a existência de uma documentação clara e padrões de códigos pré estabelecidos, permitindo que o conhecimento sobre o software seja passada de forma fácil e prática entre membros da comunidade que se interessem pelo projeto.[Souza et al., 2010]

Diante esse cenário e os problemas investigados, surge a questão de pesquisa desse trabalho: **Quais técnicas e boas práticas de desenvolvimento de software podem ser empregadas na refatoração de sistemas legados para garantir segurança, escalabilidade e facilidade de manutenção?**

1.3 Objetivos

Diante dos problemas associados ao desenvolvimento *Go Horse*, a reestruturação de um software com tecnologias legadas e práticas inadequadas surge como uma solução estratégica. O objetivo geral é modernizar o sistema, aplicando boas práticas de programação, prevenindo o seu envelhecimento. Esse objetivo principal pode ser decomposto em objetivos específicos, como:

- Analisar o projeto atual do Wegia, identificando suas falhas, dívidas técnicas, código legado e vulnerabilidades de segurança;
- Reestruturar a aplicação, realizando a migração de uma aplicação em PHP puro para os frameworks Laravel no *backend* e Nuxt no *Frontend*, visando segurança e maior organização da estrutura do projeto;
- Implementar padrões e boas praticas de desenvolvimento, garantindo um projeto escalável e estável;
- Propor uma arquitetura de software que promova modularidade, segurança e evolução contínua;
- Criar um script automático para facilitar a subida em servidores para fácil implementação;
- Elaborar um *README* detalhado como forma de documentação técnica do projeto, tornando mais fácil e acessível para outros programadores darem início ao desenvolvimento;
- Utilizar ferramentas como o docker para facilitar o uso em qualquer sistema operacional;
- Realizar normalização das tabelas, quando necessário, migrando os dados do banco antigo para o novo;
- Validar o funcionamento da aplicação após sua migração tecnológica.

1.4 Metodologia

Este trabalho se inspirou no modelo *Design Science Research* [Pimentel et al., 2020], sendo adaptado conforme as necessidades específicas do projeto conforme ilustrado na Figura 1.:

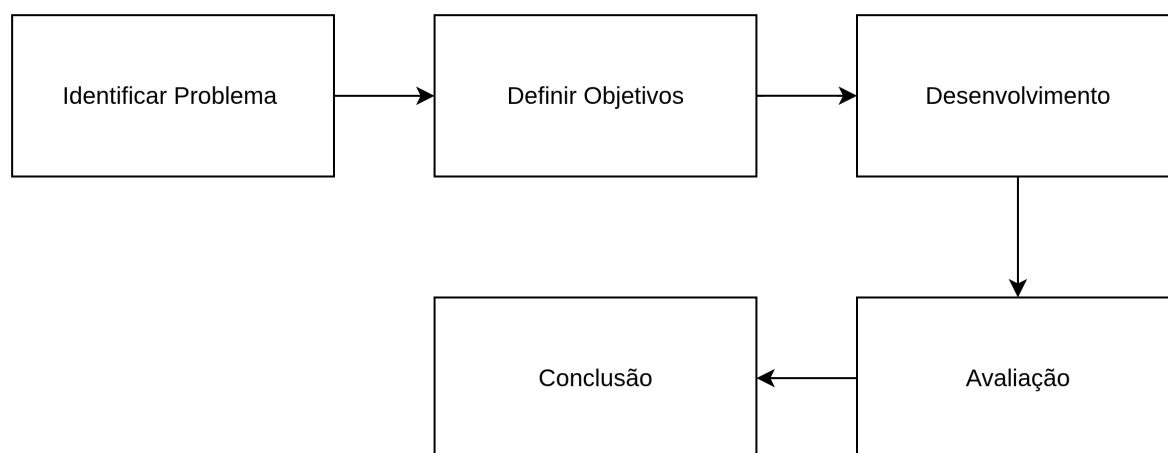


Figura 1: Fluxo metodológico utilizado e adaptado [Pimentel et al., 2020] .

- **Identificar Problema:** Nesta Etapa, realizou-se uma análise no atual código fonte do projeto do WeGIA e no seu banco de dados para buscar informações do que poderia ser melhorado. Além disso, teve um estudo conforme o artigo [Lazarin et al., 2024] onde ocorreu uma pesquisa de campo para uma análise mais aprofundada das necessidades dos usuários no sistema.
- **Definir Objetivos:** Esta etapa segue os objetivos citados dentro da introdução na subseção 1.3, os quais orientaram a reconstrução do sistema na etapa de desenvolvimento. Para que fosse possível alcançar esse objetivo, foram utilizadas as tecnologias Laravel no backend e Nuxt.js no frontend, adotando padrões de arquitetura REST.
- **Desenvolvimento:** Nessa etapa, foi realizada a migração tecnológica visando buscar um sistema com uma arquitetura modular, escalável, seguro e moderno.
- **Avaliação:** O sistema após a reconstrução, foi submetido a diversos testes, como de carga e manuais, garantindo que todas as funcionalidades estejam de acordo com o planejado.
- **Conclusão:** Os resultados obtidos na etapa de avaliação foi utilizada para revisar a solução proposta e identificar oportunidades de melhorias em cima dela.

1.5 Organização do texto

Este projeto de conclusão de curso está dividido em cinco capítulos. O atual introduziu e contextualizou sobre o assunto abordado, evidenciando os problemas investigados e a questão de pesquisa

O capítulo 2 aborda a fundamentação teórica sobre o tema, servindo como base para entendimentos dos demais capítulos.

O Capítulo 3 apresenta os trabalhos relacionados, discutindo estudos anteriores sobre o tema de reestruturação de software.

O Capítulo 4 descreve a proposta de solução do trabalho, apresentando características, diagramação, técnicas utilizadas, testes realizados e avaliações descrito em mais detalhes.

O capítulo 5 conclui o trabalho, apresentando uma discussão final sobre o tema, bem como sugestões de trabalhos futuros e considerações finais.

Capítulo 2

Fundamentação Teórica

Capítulo 3

Revisão da Literatura / Trabalhos Relacionados

Capítulo 4

Proposta

Capítulo 5

Conclusão

Referências

- Adriano Rocha Borges. Uma investigação sobre o impacto da dívida técnica em atividades de desenvolvimento de software. Master's thesis, Universidade Salvador (UNIFACS), 2022. URL <http://tede.unifacs.br/tede/handle/tede/897>.
- Alexandre Melo Braga. Visão geral das boas práticas para construção de softwares seguros. *Revista Técnica IPEP, São Paulo, SP*, 7(2):65–78, 2007.
- Telcio Elui Cardoso. and afonso sales d school of technology, pucrs, porto alegre, rs 90619-900, brazil telcio. cardoso@ edu. pucrs. br {alan. santos, rafael. chanin, afonso. sales}@ pucrs. br. In *Software Business: 10th International Conference, ICSOB 2019, Jyväskylä, Finland, November 18–20, 2019, Proceedings*, volume 370, page 175. Springer Nature, 2019.
- Nilson Lazarin, Rafael Elias Escalfoni, and Vinícius Ferreira. Wegia: Web gerenciador para instituições assistenciais. In *Anais do XXI Congresso Latino-Americano de Software Livre e Tecnologias Abertas*, pages 322–330, Porto Alegre, RS, Brasil, 2024. SBC. doi: 10.5753/latinoware.2024.245668. URL <https://sol.sbc.org.br/index.php/latinoware/article/view/31544>.
- Gabriela Trindade Perry. Qual a função do método no projeto?: um ensaio sobre o ensino de metodologia na graduação. *Blucher Design Proceedings. Blucher*, 2016.
- Mariano Pimentel, Denise Filippo, and Thiago Marcondes dos Santos. Design science research: pesquisa científica atrelada ao design de artefatos. *RE@D - Revista de Educação a Distância e Elearning*, 3(1), março–abril 2020.
- Yóris Linhares Souza, Maria Celeste Reis Lobo Vasconcelos, Valéria Maria Martins Judice, and George Leal Jamil. A contribuição do compartilhamento do conhecimento para o gerenciamento de riscos em projetos: um estudo na indústria de software. *JISTEM-Journal of Information Systems and Technology Management*, 7:183–204, 2010.
- Junior Sturm and Madalena Pereira da Silva. Aplicação de padrões de projeto no desenvolvimento de software para a melhoria da qualidade e da manutenibilidade. *Revista RETEC*, 2014.