

UNIVERSIDADE PAULISTA

BRUNO SANTOS DE OLIVEIRA

GUILHERME DENEZ DOS RAMOS GONÇALVES

GUILHERME MAIELLI MONTORO

PEDRO HENRIQUE DOS SANTOS

VINÍCIUS PAULO SANCHES

**DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA
SIMPLIFICAÇÃO DA CRIAÇÃO DE WORKFLOWS EM YAML**

SÃO PAULO

2025

UNIVERSIDADE PAULISTA

BRUNO SANTOS DE OLIVEIRA

GUILHERME DENEZ DOS RAMOS GONÇALVES

GUILHERME MAIELLI MONTORO

PEDRO HENRIQUE DOS SANTOS

VINÍCIUS PAULO SANCHES

**DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA
SIMPLIFICAÇÃO DA CRIAÇÃO DE WORKFLOWS EM YAML**

Trabalho de conclusão de curso para
obtenção do título de graduação em
Ciência da Computação apresentado à
Universidade Paulista – UNIP.

Orientador: Prof. MSc. Álvaro Prado

SÃO PAULO

2025

BRUNO SANTOS DE OLIVEIRA

GUILHERME DENEZ DOS RAMOS GONÇALVES

GUILHERME MAIELLI MONTORO

PEDRO HENRIQUE DOS SANTOS

VINÍCIUS PAULO SANCHES

**DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA
SIMPLIFICAÇÃO DA CRIAÇÃO DE WORKFLOWS EM YAML**

Trabalho de conclusão de curso para
obtenção do título de graduação em
Ciência da Computação apresentado à
Universidade Paulista – UNIP.

Orientador: Prof. MSc. Álvaro Prado

Aprovado(a) em: ____/____/____

BANCA EXAMINADORA

Prof. ou Profa. Dr(a)/Me(a). xxxxxxxxxxxx
Universidade Paulista - UNIP

Prof. ou Profa. Dr(a)/ Me(a). xxxxxxxxxxxx

RESUMO

No mundo da programação, bibliotecas e ferramentas facilitam o trabalho dos desenvolvedores, ajudando-os a escrever melhor seus projetos. Um problema atual nesse meio é a complexidade dos workflows escritos em YAML, que dificultam sua configuração. O objetivo deste trabalho é desenvolver uma plataforma web que simplifique esse processo de criação, oferecendo uma interface intuitiva que elimine a necessidade de configurar manualmente esses arquivos, aumentando, assim, a produtividade e a eficiência no desenvolvimento. A metodologia adotada para solucionar esse problema será a criação de uma aplicação web com uma interface fácil de entender e amigável. Essa aplicação permitirá que os programadores construam seus workflows personalizados por meio de um gerador de código YAML. A pesquisa bibliográfica abordará conceitos sobre ferramentas de automação de workflows, como o GitHub Actions, investigando as melhores práticas para uma construção sólida. Além disso, analisará as dificuldades enfrentadas pelos desenvolvedores na criação manual de arquivos YAML. A pesquisa experimental envolverá o desenvolvimento e a testagem da plataforma, incluindo a avaliação da interface, usabilidade, eficácia do gerador de workflows e os benefícios da plataforma em comparação com a construção manual. Os resultados esperados são que os desenvolvedores adotem essa plataforma como uma ferramenta indispensável na criação de workflows, devido à sua simplicidade e rapidez.

Palavras-chave: Workflows, YAML, Plataforma Web

ABSTRACT

In the programming world, libraries and tools make developers' work easier by helping them write their projects more efficiently. A current issue in this field is the complexity of workflows written in YAML, which makes their configuration difficult. The goal of this project is to develop a web platform that simplifies the workflow creation process by offering an intuitive interface, eliminating the need for manual configuration of these files, and thus increasing productivity and development efficiency. The methodology adopted to address this problem will be the development of a web application with a user-friendly and easy-to-understand interface. This application will allow developers to build their own customized workflows through a YAML code generator. The literature review will cover concepts related to workflow automation tools, such as GitHub Actions, exploring best practices for solid workflow construction. Additionally, it will analyze the challenges developers face when manually creating YAML files. The experimental research will involve the development and testing of the platform, including the evaluation of its interface, usability, the effectiveness of the workflow generator, and the benefits of the platform compared to manual creation. The expected results are that developers will adopt this platform as an essential tool for creating workflows, due to its simplicity and speed.

Keywords: Workflows, YAML, Web Platform

SUMÁRIO

1. INTRODUÇÃO.....	7
1.1 CONTEXTUALIZAÇÃO	8
1.2 PROBLEMA	9
1.3 MOTIVAÇÃO.....	10
1.4 OBJETIVOS	11
1.4.1 Objetivo geral.....	11
1.4.2 Objetivos específicos	11
1.5 ORGANIZAÇÃO DO TRABALHO	11
2. REFERENCIAIS TEÓRICOS.....	13
3. DESENVOLVIMENTO	13
4. EXPERIMENTO E ANÁLISE DE RESULTADOS	13
5. CONCLUSÃO.....	13
REFERÊNCIA BIBLIOGRÁFICA	13

1. INTRODUÇÃO

Diante da crescente pressão por parte do mercado, empresas de software enfrentam o desafio de entregar produtos com alta qualidade em prazos cada vez menores (Nunes, 2001). Para atender a essa demanda, a adoção de práticas como Integração Contínua (CI) e Entrega Contínua (CD) tem se mostrado essencial. Essas abordagens permitem automatizar tarefas repetitivas e agilizar o ciclo de desenvolvimento, promovendo entregas mais frequentes, confiáveis e com menor esforço manual — o que contribui diretamente para a redução de falhas humanas e aceleração dos processos de validação e liberação de software (Fowler, 2024).

Nesse contexto, a plataforma GitHub lançou o **GitHub Actions**, uma plataforma de automação de fluxos de trabalho que facilita a implementação de pipelines para compilação, testes e deploy. Essa automação é descrita por meio de arquivos YAML — que definem os passos executados conforme eventos no repositório, podendo ser acionados manualmente ou automaticamente.

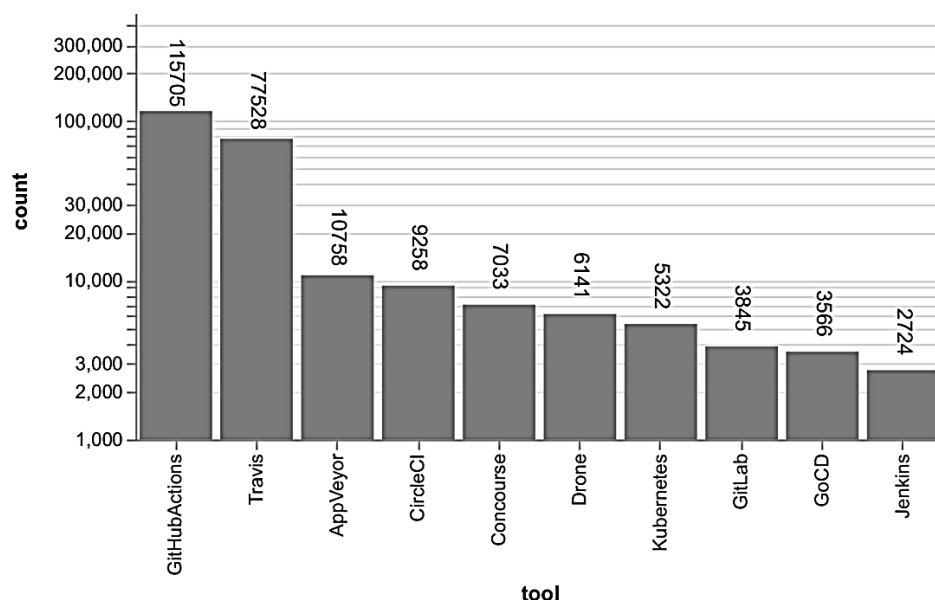
A forma como as equipes colaboram também mudou. O modelo **pull-based** tem se tornado rapidamente um dos principais paradigmas no desenvolvimento de software distribuído. Nele, os contribuidores realizam alterações em branches separados e submetem *pull requests* para revisão e integração no branch principal. Conforme a análise de Gousios *et. al.* (2015), metade dos repositórios compartilhados seguem este modelo, sendo que apenas o GitHub hospeda aproximadamente um milhão desses repositórios. Apesar de eficiente, o processo exige cuidado: um merge malfeito pode comprometer a estabilidade do sistema, introduzindo bugs ou degradando a qualidade do software.

Tendo isso em vista, Wessel (2023) expõe que desenvolvedores frequentemente recorrem a ferramentas de automação, como Integração Contínua e Entrega Contínua (CI/CD), para executar tarefas repetitivas, como verificar a compilação do código, se os testes são aprovados e se a contribuição está em conformidade com os parâmetros definidos.

1.1 CONTEXTUALIZAÇÃO

A partir de um levantamento realizado por Gião *et. al.* (2024) entre 612.557 repositórios criados entre 2012 e 2023 (da consolidação do DevOps até os dias atuais), 200.023 destes (32,7%) contém pelo menos uma tecnologia de CI/CD, o que significa que um terço dos repositórios nos últimos doze anos tem ou teve algumas dessas tecnologias. A Figura 1 apresenta uma distribuição das tecnologias de CI/CD com maior presença nos repositórios, focando em tecnologias cuja utilização supera 1% do total de repositórios que adotam esse tipo de ferramenta.

Figura 1 — Distribuição das tecnologias pelos repositórios de CI/CD



Fonte: Gião *et. al.* (2024)

Conforme ilustrado na Figura 1, percebe-se a partir da frequência de uso de diferentes ferramentas de Integração Contínua (CI) que há uma ampla popularidade do GitHub Actions em relação a outras tecnologias, totalizando 115.705 ocorrências. Outras ferramentas aparecem em seguida, como Travis (77.528), AppVeyor (10.758), CircleCI (9.258), Concourse (7.033), Drone (6.141), Kubernetes (5.322), GitLab (3.845), GoCD (3.566) e Jenkins (2.724).

Ainda que cada ferramenta apresente uma abordagem distinta, um aspecto permanece constante entre cada uma delas: o uso da linguagem YAML para definir

pipelines e automações. YAML, acrônimo recursivo para “*YAML Ain’t Markup Language*”, é uma linguagem para serialização de dados projetada para ser legível por humanos. Em outras palavras, é utilizado para armazenar e transportar dados de forma estruturada, oferecendo uma sintaxe mais simples e direta que a XML (*Extensible Markup Language*), por exemplo (Evans, Bem-Kiki e Döt Net, 2021). Contudo, embora a sua estrutura possua uma semântica mais enxuta, novos desafios emergem da sua sintaxe sensível e da ausência de recursos como sugestões automáticas.

1.2 PROBLEMA

Com isso, a **complexidade do processo de desenvolvimento** se intensifica. Automatizar fluxos de trabalho é essencial — mas escrever e manter esses fluxos, especialmente usando arquivos YAML, se mostra um novo desafio.

O estudo realizado por Saroar e Nayebe (2023) demonstrou que 60,87% dos usuários do Actions consideravam a composição do YAML desafiadora e propensa a erros. Foram identificados como principais desafios da ferramenta Actions do GitHub a dificuldade na criação, depuração e reutilização dos arquivos YAML, que podem se tornar complexos e de difícil compreensão (Figuras 1 e 2).











Figura 2 — Como desenvolvedores criam arquivos YAML

Como desenvolvedores criam YAMLs	% de participantes
Modificando um arquivo já existente	69,57
Usando modelos configuráveis do GitHub	50,72
Escrevendo do zero	49,28
Copiando de sites ou fóruns	43,48
Documentação do Actions	1,45

Fonte: Adaptado de Saroar e Nayebe (2023)

A partir dos resultados apresentados pelos autores, é possível perceber que *modificando um arquivo já existente* (69,57%) e *usando modelos configuráveis do GitHub* (50,72%) foram os métodos mais populares na criação deste tipo de arquivo entre os entrevistados. Além disso, muitas vezes usuários demonstraram escrever arquivos YAML do zero (49,28%) ou copiar de sites ou fóruns como *Stack Overflow* (43,48%). Apenas 1,45% relatou utilizar a documentação do Actions.

Figura 3 — Principais desafios ao criar arquivos YAML

Desafios ao criar YAMLs	% de participantes
Sem desafios	39,13 
Nunca configurou arquivos YAML	10,14 
Sintaxe	11,59 
Depurar	8,7 
Generalização	7,25 
Indentação	5,8 
Ausência de IntelliSense	4,35 
Documentação inadequada	4,35 
Curva de aprendizado	2,9 
Outros	5,8 

Fonte: Adaptado de Saroar e Nayeibi (2023)

Complementando os resultados anteriores, observa-se que 39,13% dos usuários não encontram desafios ao configurar arquivos YAML. No entanto, uma parcela significativa relatou dificuldades, destacando-se a falta de familiaridade com a configuração de arquivos YAML (10,14%), problemas com a sintaxe da linguagem (11,59%) e obstáculos no processo de teste e depuração (8,7%). Outros desafios mencionados incluem a dificuldade em tornar os arquivos reutilizáveis ou adaptáveis (7,25%), lidar com regras de indentação (5,8%), ausência de sugestões automáticas (IntelliSense) e documentação inadequada (ambos com 4,35%), além de uma curva de aprendizado considerada complexa (2,9%) e outros fatores não especificados (5,8%).

1.3 MOTIVAÇÃO

Ainda segundo Saroar e Nayeibi (2023, tradução nossa), após o desenvolvimento de sua análise expressa que "um framework gerador automático de fluxos de trabalho pode ajudar a atenuar esse problema e tornar o GitHub Actions mais acessível a um público mais amplo". Com base nisso, o presente trabalho visa abranger não apenas os workflows desenvolvidos para o GitHub, mas também, e sem se limitar a isso, os containers Docker e qualquer outros processos que utilizem arquivos YAML. Apesar da sintaxe desses arquivos ser simples, pequenos erros de indentação podem causar falhas críticas — e identificar esses erros pode consumir tempo e gerar frustração.

1.4 OBJETIVOS

1.4.1 Objetivo geral

O objetivo desse trabalho consiste em desenvolver uma plataforma para mitigar as dificuldades e facilitar a criação de arquivos YAML de forma inteligente, guiada e explicativa. A proposta é auxiliar especialmente os desenvolvedores iniciantes, oferecendo uma experiência mais amigável e reduzindo a curva de aprendizado. Ao integrar boas práticas de automação com usabilidade e acessibilidade, esse projeto visa se posicionar como um importante aliado no processo de desenvolvimento moderno.


1.4.2 Objetivos específicos

- Criar uma aplicação web para criação, edição e compartilhamento de arquivos YAML de forma assistida e inteligente;
- Diminuir a curva de aprendizado de tecnologias que usam YAMLs para automatizar seus fluxos de trabalho, como o Github faz usando o Github Action;
- Aumentar a produtividade das equipes na estruturação de novos projetos;
- Realizar uma pesquisa de campo com o objetivo de identificar o impacto da ferramenta no contexto de desenvolvimento de projetos;
- Exibir os resultados da implementação da ferramenta.

1.5 ORGANIZAÇÃO DO TRABALHO

O trabalho será conduzido no modelo de pesquisa bibliográfica visando conduzir o leitor ao entendimento do problema enfrentado hoje pelas empresas e desenvolvedores, apresentando uma solução de software denominada Codeflow e os benefícios de sua utilização, principalmente para o público iniciante. Este capítulo visa introduzir a problemática da utilização, manutenção e depuração de arquivos YAML dentro de softwares, como Docker, GitHub Actions e outros, a fim de esclarecer quaisquer dúvidas a respeito da necessidade de um software que auxilie na construção assistida de tais arquivos. Na Figura 2 é possível compreender a estrutura da organização do presente trabalho com os principais pontos desenvolvidos em cada capítulo.

Figura 4 — Organização do Trabalho



Capítulo 2	Capítulo 3	Capítulo 4	Capítulo 5
<ul style="list-style-type: none">▪ Referencial Teórico▪ Trabalhos Relacionados	<ul style="list-style-type: none">▪ Documento de Visão▪ Escopo do Projeto▪ Arquitetura de Solução▪ Projeto do Banco de Dados▪ Modelagem do Software	<ul style="list-style-type: none">▪ Experimento e Análise de Resultados	<ul style="list-style-type: none">▪ Conclusão

Fonte: Autoria Própria.

- O **Capítulo 2** irá apresentar e contextualizar alguns conceitos importantes para o entendimento do trabalho, assim como o referencial teórico - necessários para entender o meio no qual o software Codeflow se propõe a auxiliar.
- O **Capítulo 3** irá demonstrar as práticas adotadas, modelos e padrões utilizados durante o desenvolvimento, assim como toda a documentação relacionada ao projeto desenvolvido.
- O **Capítulo 4** irá apresentar os resultados obtidos com a implementação do software produzido e os impactos da sua utilização.
- No **Capítulo 5** serão mostradas as perspectivas futuras e as dificuldades encontradas.

2. REFERENCIAIS TEÓRICOS

3. DESENVOLVIMENTO

4. EXPERIMENTO E ANÁLISE DE RESULTADOS

5. CONCLUSÃO

REFERÊNCIA BIBLIOGRÁFICA

NUNES, W. **Implantação de práticas de Integração Contínua: Um relato de experiência em um laboratório de pesquisa e desenvolvimento de software**. 2017. 64 f. Monografia (Graduação em Sistemas de Informação) — Universidade Federal de Ouro Preto, Minas Gerais, 2001.

FOWLER, M. **Continuous Integration**. Martin Fowler, 2024. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>. Acesso em 05 de abril de 2025.

GITHUB. **Início rápido para GitHub Actions**. 2025. Disponível em: <https://docs.github.com/pt/actions/writing-workflows/quickstart>. Acesso em: 05 de abril de 2025.

GOUSIOS, G. *et. al.* Work Practices and Challenges in Pull-Based Development: The Integrator's Perspective. In: **2015 IEEE/ACM 37th IEEE International Conference on Software Engineering**, 2015, Florence. Disponível em: <http://dx.doi.org/10.1109/ICSE.2015.55>. Acesso em: 05 de abril de 2025.

SAROAR, S; NAYEBI, M. Developers Perception of GitHub Actions: A Survey Analysis. In: **Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering**, 2023, Oulu. Disponível em: <https://doi.org/10.1145/3593434.3593475>. Acesso em: 03 de abril de 2025.

WESSEL, M. *et. al.* GitHub Actions: The impact on the Pull Request Process. **Empirical Software Engineering**, v.28, n. 6, p. 131, 2023. Disponível em: <https://doi.org/10.1007/s10664-023-10369-w>. Acesso em 04 de abril de 2025.

EVANS, C; BEM-KIKI, O; DÖT NET, I. **YAML Ain't Markup Language**. YAML, 2021. Disponível em: <https://yaml.org/spec/1.2.2/>. Acesso em: 19 de abril de 2025.

GIÃO, H. *et. al.* **Chronicles of CI/CD: A Deep Dive into its Usage Over Time**. arXiv preprint arXiv:2402.17588, 2024. Disponível em: <https://arxiv.org/html/2402.17588v1>. Acesso em 20 de abril de 2025.