

T H E   A C H I E V E R S

A P I   -   2 0 2 3 . 1

---

D E V O P S

I N T E G R A Ç Ã O   &   E N T R E G A  
C O N T Í N U A

---



---

# O QUE É DEV OPS ?

---

DevOps é uma abordagem que busca integrar equipes de desenvolvimento e operações para promover uma cultura de colaboração contínua e acelerar o ciclo de vida do desenvolvimento de software. Uma das práticas-chave no contexto do DevOps é a integração e entrega contínua (CI/CD), que envolve a automação de processos, como a compilação, teste e implantação, para permitir a entrega rápida e confiável de novas funcionalidades. Através da integração e entrega contínua, as equipes podem detectar problemas precocemente, garantir a estabilidade do software e disponibilizar atualizações frequentes para os usuários, resultando em uma maior eficiência, qualidade e satisfação geral.



---

# F E R R A M E N T A S

---

Uma das vantagens do DevOps é a vasta gama de ferramentas e tecnologias que apoiam suas práticas e processos, permitindo que as equipes escolham as ferramentas mais adequadas às suas necessidades e contextos específicos de desenvolvimento e operações.

A seguir, serão apresentadas algumas das muitas alternativas disponíveis, as quais foram aplicadas no sistema referente a API do primeiro semestre de 2023 em parceria com a empresa TECSUS:

- JIRA
- Github Actions
- Cypress
- SonarQube Cloud
- Vercel



---

# J I R A

---

O Jira é uma ferramenta de **gerenciamento de projetos** com o objetivo de auxiliar equipes no planejamento, acompanhamento e rastreamento de tarefas e eventuais problemas ao longo desenvolvimento de software.

Em relação à integração e entrega contínua (CI/CD), desempenha um papel importante ao permitir a conexão com ferramentas de automação e controle de versão, como o Git e o GitHub, permitindo que os fluxos de trabalho CI/CD possam ser integrados diretamente ao Jira, facilitando o acompanhamento do progresso das tarefas e atualizações de código em tempo real, crucial para o DevOps e para a metodologia Scrum.



---

# J I R A

---

Essa conexão entre o Jira e as práticas de CI/CD melhora a visibilidade, colaboração e eficiência das equipes de desenvolvimento e operações, facilitando o planejamento e a entrega contínua de software de qualidade.

Figura 1 - Logo JIRA



Fonte: IconScout



---

# G I T H U B   A C T I O N S

---

O GitHub Actions é uma plataforma de **automação de fluxos** de trabalho fornecida pelo GitHub, o qual tem como objetivo principal permitir que as equipes automatizem e gerenciem tarefas repetitivas ou complexas relacionadas ao desenvolvimento de software.

No contexto do CI/CD, desempenha um papel crucial, permitindo a criação de fluxos de trabalho personalizados, nos quais é possível definir etapas específicas, como compilação, teste, implantação e até mesmo notificações.



---

# G I T H U B   A C T I O N S

---

Ao conectar-se ao GitHub, o Actions pode ser configurado para acionar esses fluxos automaticamente sempre que ocorrerem alterações no repositório, como push de código ou abertura de pull requests, possibilitando a execução contínua de testes, compilações e implantações, o que garante a entrega confiável e rápida do software. A

Figura 2 - Logo Github Actions



Fonte: Github



---

# C Y P R E S S

---

O Cypress é uma ferramenta de **teste de interface** de usuário (UI) que visa simplificar e melhorar a experiência de teste de aplicações web. Seu principal objetivo é permitir que as equipes realizem testes automatizados eficientes e confiáveis na interface de usuário de seus aplicativos.

No que tange ao CI/CD, o Cypress desempenha um papel importante ao fornecer recursos de teste automatizado que podem ser incorporados aos fluxos integração e entrega contínua.





---

# C Y P R E S S

---

Com o Cypress, é possível escrever e executar testes automatizados que verificam a funcionalidade, a usabilidade e a integridade da interface de usuário, que podem ser integrados aos pipelines de CI/CD, permitindo que sejam acionados automaticamente em cada atualização ou nova versão do código.

Isso ajuda a garantir que as alterações de código não introduzam regressões ou problemas na experiência do usuário.



---

# C Y P R E S S

---

Dessa forma, o Cypress contribui para a qualidade e confiabilidade contínuas do software, ao facilitar a execução automatizada de testes de interface de usuário como parte integrante do processo de CI/CD.

Figura 3 - Logo Cypress



Fonte: Cypress



---

# S O N A R Q U B E   C L O U D

---

O SonarQube Cloud é uma plataforma de análise estática de código que tem como objetivo ajudar as equipes de desenvolvimento a melhorar a **qualidade** do código de seus projetos.

O principal objetivo do SonarQube é identificar problemas, bugs, vulnerabilidades de segurança, violações de boas práticas de programação e outros tipos de problemas relacionados à qualidade do código, realizando uma detalhada análise do source code em busca desses problemas, fornecendo relatórios e métricas que ajudam as equipes a entender e corrigir as áreas problemáticas em seu código.



---

# S O N A R Q U B E   C L O U D

---

No contexto da integração e entrega contínua, desempenha um papel importante ao ser incorporado aos pipelines, uma vez que pode ser configurado para executar análises estáticas automáticas do código a cada nova versão ou atualização, fornecendo feedback imediato sobre a qualidade do código e identificando problemas em estágios iniciais.

Isso ajuda as equipes a detectar e corrigir problemas de qualidade de código antes que eles se tornem problemas maiores no processo de desenvolvimento.



---

# S O N A R Q U B E   C L O U D

---

Com a integração do SonarQube Cloud ao CI/CD, as equipes podem garantir que seu código esteja sempre em conformidade com as melhores práticas e padrões de qualidade estabelecidos, melhorando assim a confiabilidade e a manutenibilidade contínuas do software.

Figura 4 - Logo SonarQube Cloud



Fonte: SonarCloud Docs



---

# V E R C E L

---

O Vercel é uma plataforma de **implantação e hospedagem** de aplicativos voltada para o desenvolvimento web moderno, que visa principalmente a simplificação do processo de implantação e fornecer um ambiente de hospedagem escalável e confiável para aplicativos web.

No contexto do CI/CD, desempenha um papel fundamental ao permitir a integração perfeita com pipelines de integração e entrega contínua.

Com o Vercel, as equipes podem configurar a implantação contínua de seus aplicativos diretamente a partir do repositório de código, como o Git.



---

# V E R C E L

---

Ou seja, sempre que houver uma nova versão do aplicativo ou uma alteração no código, o Vercel pode acionar automaticamente a construção e implantação do aplicativo.

Com o Vercel integrado ao CI/CD, as equipes podem garantir que suas alterações de código sejam rapidamente implantadas e disponibilizadas aos usuários, permitindo um ciclo de entrega contínua eficiente e uma experiência de usuário aprimorada.

Figura 5 - Logo Vercel



Fonte: SonarCloud Docs



---

# I N T E G R A Ç Ã O

---

A integração das ferramentas citadas anteriormente proporciona um fluxo de trabalho ágil e eficiente no desenvolvimento de software, sinergizando com a metodologia Scrum.

Quando integradas, essas ferramentas permitem um ciclo de desenvolvimento contínuo, desde o gerenciamento de tarefas e problemas no Jira, passando pela automação de CI/CD com o GitHub Actions, até os testes automatizados com o Cypress e análise de qualidade de código com o SonarQube.





---

# I N T E G R A Ç Ã O

---

Por fim, o Vercel possibilita a implantação eficiente e escalável dos aplicativos. Essa integração resulta em um fluxo de trabalho mais colaborativo, eficiente e confiável, permitindo que as equipes desenvolvam e entreguem software de alta qualidade de maneira rápida e consistente, mantendo-se sempre alinhado com os princípios de DevOps.



# T H E   A C H I E V E R S

A P I   -   2 0 2 3 . 1



V I S I T E   N O S S O   G I T H U B

---

