

# DevOps

**Disciplina:** Análise e Projetos em Sistemas



## Conteúdos:

DevOps.

## Habilidade(s):

- Conhecer e aplicar os princípios fundamentais do DevOps;
- Gerenciar a aplicação *web*;
- Realizar o monitoramento contínuo das aplicações e da infraestrutura do *website*.

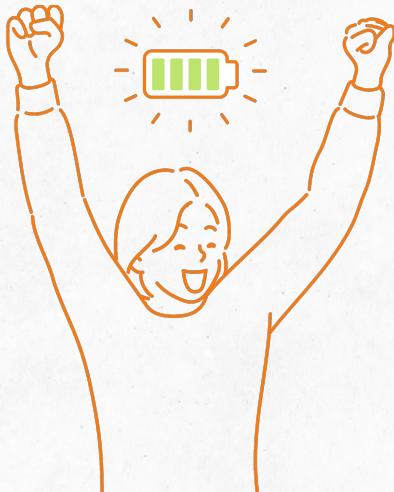
# Bloco 1

---

# Abertura

Qual é o seu *mood* de hoje?

Estou com a bateria totalmente carregada!



A energia está em falta...



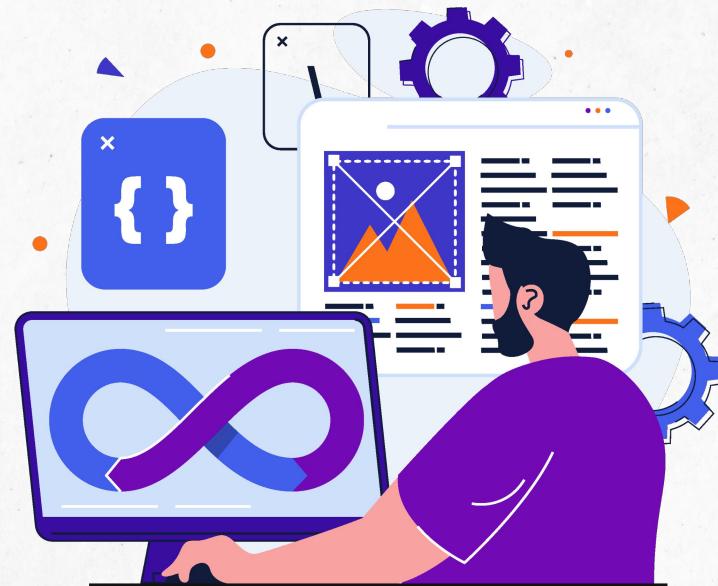
# O que é DevOps?

DevOps vai além de uma prática de desenvolvimento.

Na verdade, DevOps é parte de uma **cultura e filosofia**.

É a junção de “**desenvolvimento**” e “**operação**”. Isso quer dizer que ele reúne práticas que combinam a aprimoração do desenvolvimento e agilização da operação, envolvendo:

- colaboração e visibilidade;
- alinhamentos constantes;
- inovação e estabilidade;
- ciclos curtos;
- aprendizado contínuo.



# Práticas de um DevOps

## CI e CD

Entrega contínua (CD) e integração contínua (CI) são práticas base do DevOps. Juntas, elas buscam agilizar os processos e aumentar a qualidade do projeto.

## Automação de testes

Essa prática consiste em testes unitários, testes de integração e testes de desempenho, garantindo a qualidade do *software* em todo o ciclo de desenvolvimento.

## Gerenciamento de configuração

Preparação de ambientes de teste, produção e homologação, garantindo a consistência e a rastreabilidade dos ambientes de desenvolvimento, teste e produção.

# Benefícios de um DevOps

- 1 Entrega contínua e ágil.
- 2 Diminuição dos erros e das falhas.
- 3 Aumento das colaborações.
- 4 Melhoria na qualidade da comunicação.
- 5 Maior flexibilidade e escalabilidade.
- 6 Possibilita uma resposta mais rápida dos usuários.
- 7 Redução do tempo de resposta aos problemas.
- 8 Cultura de melhoria contínua.



## Dê *play* no conhecimento!



Conversa entre Marco Bruno e Jeferson, do LINUXtips,  
sobre a atuação de um DevOps.

# Bloco 2

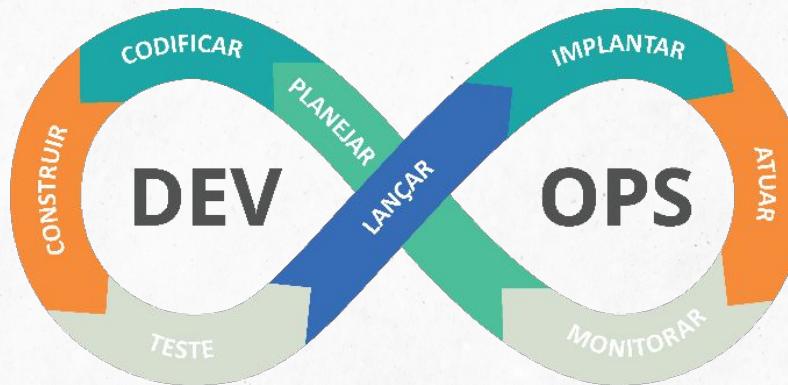
---



**Imagine que um colega de classe faltou à  
aula e quer saber o que é um DevOps.**

**Como você resumiria o conceito para ele?**

# Ciclo de DevOps



# Cultura DevOps

A cultura DevOps é um dos pilares fundamentais para a implementação bem-sucedida da abordagem DevOps. Ela envolve a **mudança de mentalidade** e uma **transformação cultural** nas equipes de desenvolvimento e operações, tendo como foco a **colaboração**, a **comunicação**, a **automação** e a **melhoria contínua**.

Ser DevOps não se trata apenas da **implementação de ferramentas e processos**, mas também de buscar **quebrar as barreiras entre as áreas**, promovendo uma **abordagem integrada e colaborativa** para o desenvolvimento.

# Metodologias de desenvolvimento

1 Modelo em Cascata;

2 Scrum;

3 Prototipagem;

4 Modelagem de processos de negócio;

5 Análise orientada a objetos (OOA);

6 Análise estruturada;

7 *Lean software development;*

8 Kanban.

??

## Tente responder

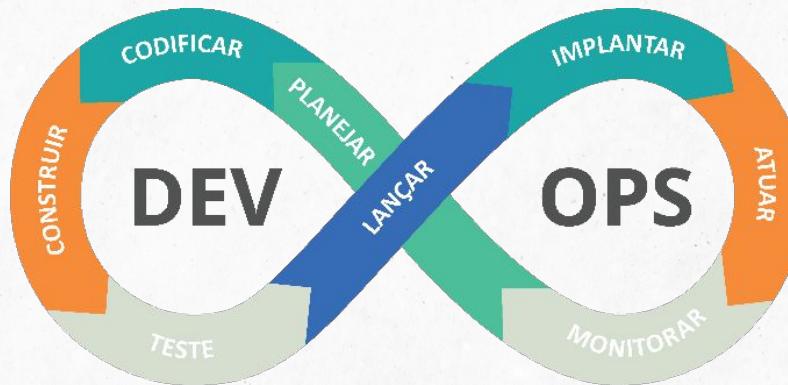
Qual das metodologias apresentadas anteriormente se assemelha com as práticas do DevOps?



# Bloco 3

---

# Ciclo de DevOps





# Integração contínua (CI)

É a prática de **integrar regularmente o código desenvolvido por diferentes membros da equipe em um repositório compartilhado**.

Utilizar um repositório compartilhado permite a detecção precoce de conflitos e erros de integração, garantindo que o código esteja sempre em um estado de funcionamento e forme uma base sólida para a entrega contínua. Por isso, essa prática é fundamental para uma maior cultura ágil, uma vez que une a análise eficiente com o aprimoramento contínuo.

# Etapas de CI

Para uma boa integração contínua, é importante respeitar as diversas etapas, como as apresentadas a seguir:

- o envio do código-fonte para o banco compartilhado;
- a compilação automática, garantindo que esteja livre de erros e sintaxe;
- execução de testes automatizados;
- momentos de *feedback* rápido;
- implantação em ambientes de desenvolvimento e testes.

## Entrega contínua (CD)

Essa é outra prática muito importante na cultura DevOps. Ela busca **automatizar o processo de liberação de software** para produção. Aqui, as equipes de desenvolvimento buscam **entregar mudanças no código de forma mais rápida, segura e confiável**, ao **adicionar novas funcionalidades e correções de maneira contínua e incremental**.

A CD se une à CI, porém, ela vai além, pois não se limita apenas à etapa de compilação e de testes automatizados.

# Características de CD

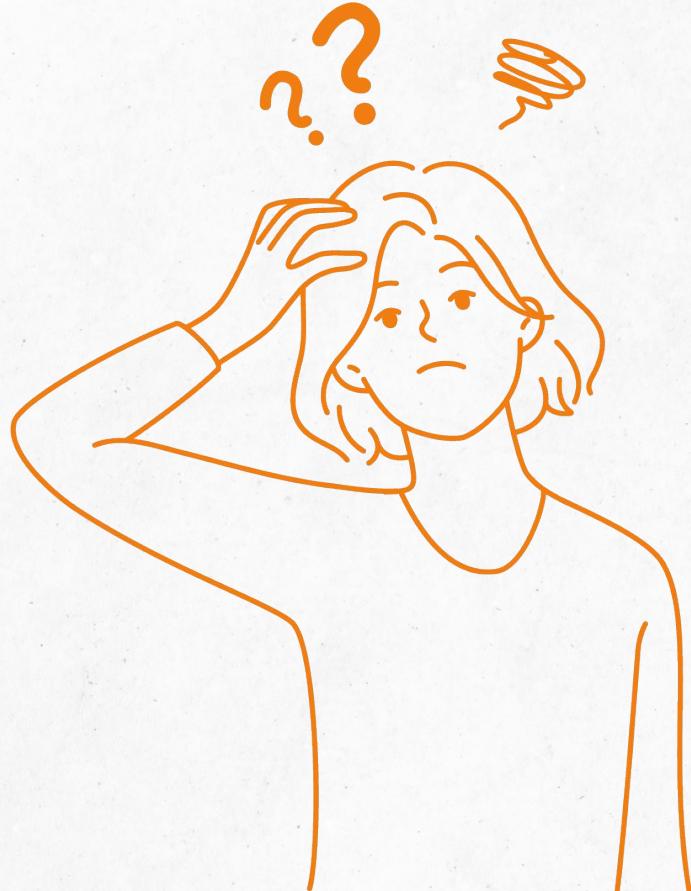
Então, o que destaca a prática de entrega contínua na cultura DevOps?

- A automação do fluxo de trabalho;
- A promoção de ambientes de teste e produção idênticos, validando o comportamento do *software*;
- Implantação contínua para o cliente;
- Monitoramento contínuo;
- Redução de riscos.



E aí?

Será que seria difícil aplicar esse sistema de entrega e integração em um projeto?



# Bloco 4

---

## Hora do quiz

Ana, uma programadora talentosa, afirma que a CD acelera o ciclo de desenvolvimento e aumenta a confiabilidade do produto. Uma das características importantes da CD, nesse sentido, é a **automação do fluxo do trabalho**.



Verdadeiro.



Falso.



## Hora do quiz

Ana, uma programadora talentosa, afirma que a CD acelera o ciclo de desenvolvimento e aumenta a confiabilidade do produto. Uma das características importantes da CD, nesse sentido, é a **automação do fluxo do trabalho**.



Verdadeiro.



Falso.



# Ferramentas para o DevOps

Não dá para buscar entender uma nova metodologia ou prática sem, antes disso, buscar as **ferramentas que facilitam a sua aplicação**.

Afinal, do que adianta se adianta se inserir na cultura DevOps, buscar a agilização dos processos, mas não ter as ferramentas que permitam realizar essas atividades?

Como atuar sem **plataformas flexíveis**?



## Jenkins

Ferramenta de código aberto amplamente utilizada para a automação de integração contínua e para a implantação contínua (CI/CD) em desenvolvimento de *software*. Ela oferece um ambiente de automação que permite aos desenvolvedores, às equipes de operações e aos outros profissionais de TI automatizar várias partes do processo de desenvolvimento, teste e implantação de aplicativos.

## Docker

Docker é uma plataforma de código aberto que permite criar, implantar e gerenciar aplicativos em contêineres, que são ambientes isolados e leves que incluem tudo o que é necessário para que um aplicativo seja executado, como códigos, bibliotecas e dependências.

## Grafana

Grafana é uma plataforma de código aberto para a visualização e a análise de dados. Ela foi projetada para ajudar as organizações a criarem painéis interativos e gráficos detalhados a partir de uma variedade de fontes de dados, como bancos de dados, sistemas de monitoramento e métricas.

## ELK Stack

ELK Stack é um conjunto de ferramentas de código aberto amplamente usado para a análise de *logs* e dados em tempo real. A sigla contém as primeiras letras dos termos a seguir.

- **Elasticsearch**, um mecanismo de pesquisa e análise distribuído que armazena e recupera dados de forma eficiente;
- **Logstash**, uma ferramenta de ingestão de dados que coleta, transforma e enriquece *logs* e outros dados;
- **Kibana**, uma interface de usuário que permite a visualização e a exploração de dados por meio de gráficos, *dashboards* e pesquisas avançadas.

## Ansible

Uma plataforma altamente escalável e amplamente adotada em ambientes de DevOps e gerenciamento de infraestrutura. Auxilia na automação de TI de código aberto e simplifica a automação de tarefas repetitivas, como a implantação de *software*, a configuração de servidores e o gerenciamento de infraestruturas. Ela utiliza linguagem YAML para definir *playbooks* que descrevem os passos de automação, oferecendo uma abordagem simples e eficaz para orquestrar e controlar ambientes de TI de maneira consistente.



## Git

Git é um sistema de controle de versões distribuído e amplamente utilizado. Registra as alterações em arquivos ao longo do tempo, facilitando o trabalho colaborativo e o gerenciamento eficiente de projetos de *software*, permitindo o rastreamento de modificações, criação de *branches* e fusão de código de forma organizada.



## Mão na massa!

Dividam-se em trios, acessem o Git e tentem  
automatizar um processo!

# Bloco 5

---

Fale mais!

Como foi a experiência de usar o Git?  
Você já tinha utilizado o sistema antes?



## Automação de testes

Pensar na agilidade é fundamental na cultura DevOps. Hoje em dia, o desenvolvimento de processos automatizados criou novos parâmetros para a análise de sistemas. Aqui não seria diferente, por isso é muito importante tentar **ganhar tempo em processos que podem ser automatizados!**

A prática de automação de testes é um das características básicas da prática de CD. Ela se relaciona diretamente com os objetivos do DevOps.

# Testes automatizados

Utilizar testes automatizados pode ser muito útil na hora de desenvolver sistemas!

## Vantagens

Eficiência

Integração com *pipelines*

Reprodutibilidade

Economia de tempo

Redução de erros

Colaboração mais fácil



# Gerenciamento de configurações

Para estabelecer a **consistência dos atributos de um produto**, é importante que ele seja gerenciado e acompanhado. É justamente essa a preocupação do gerenciamento de configurações. Essas são algumas questões que acompanham esse tipo de gerenciamento são:

- os metadados do sistema de *software* atuam de maneira correta?
- como lidar com as mudanças desses elementos?

# E qual é a sua importância?

O gerenciamento de configurações auxilia bastante o DevOps, pois os sistemas devem sempre ser **flexíveis**.  
Mas, afinal, em quais pontos esse gerenciamento ajuda de verdade?

- Permite que a equipe de desenvolvimento controle e **acompanhe as diferentes versões de um artefato**;
- Torna possível **rastrear as mudanças feitas em um artefato** ao longo do tempo;
- Caso ocorram problemas ou erros em uma nova versão de um artefato, ele permite **restaurar versões antigas**;
- Ajuda a **garantir a segurança dos artefatos** do projeto;
- Facilita a **realização de auditorias detalhadas** para avaliar as mudanças do projeto.

# Bloco 6

---

Atualmente, qual é a sua maior dificuldade nesse tema?



Compartilhe as suas  
dúvidas!

## Monitoramento e *feedback*

A cultura ágil não deve focar apenas nos processos e nas ferramentas, mas também **nas pessoas**. É por isso que a comunicação é extremamente importante para o DevOps. Ampliar a prática de *feedbacks* permite que os **stakeholders** e os clientes estejam alinhados, o que permite alcançar os objetivos do projeto.

E, claro, não dá para construir bons *feedbacks* sem **monitorar e acompanhar o alcance das metas e dos parâmetros de qualidade** do projeto!

# E é preciso monitorar o quê?

 Coletar e avaliar **métricas relevantes** para o funcionamento do projeto. Isso inclui o uso de recursos, o tempo de resposta, a latência etc.;

 *Logs* são essenciais para rastrear **eventos e atividades do sistema**, permitindo identificar **erros e falhas**;

 Acompanhar o **desempenho do sistema** de maneira contínua é fundamental para a qualidade do produto;

 As **notificações** e os **alertas** são ótimas ferramentas para monitorar sistemas.

# Um bom *feedback*

Um *feedback* precisa ser direto e objetivo. Isso ajuda a garantir o alinhamento das expectativas e a compreensão dos stakeholders;

Feedbacks constantes são fundamentais no decorrer de um projeto. Eles podem ser feitos em reuniões periódicas, como as **retrospectivas**;

Além disso, uma **análise contínua** do projeto ajuda na constante melhoria do produto;

Faça um *feedback* levando em consideração quem o recebe. **Tenha empatia**, mas também considere as **métricas de qualidade** e as **exigências do projeto**;

**O feedback precisa ser completo!** Quem escuta a crítica precisa direcioná-la para melhoria e possível adoção.

# Um exemplo prático!

Você pode substituir a frase

**“O seu trabalho está muito ruim!”**

por:

# Como você se sente?

1

Você acha que o conteúdo se aplica ao seu cotidiano?

2

Do que foi aprendido aqui, o que é importante lembrar durante a sua atuação profissional?

3

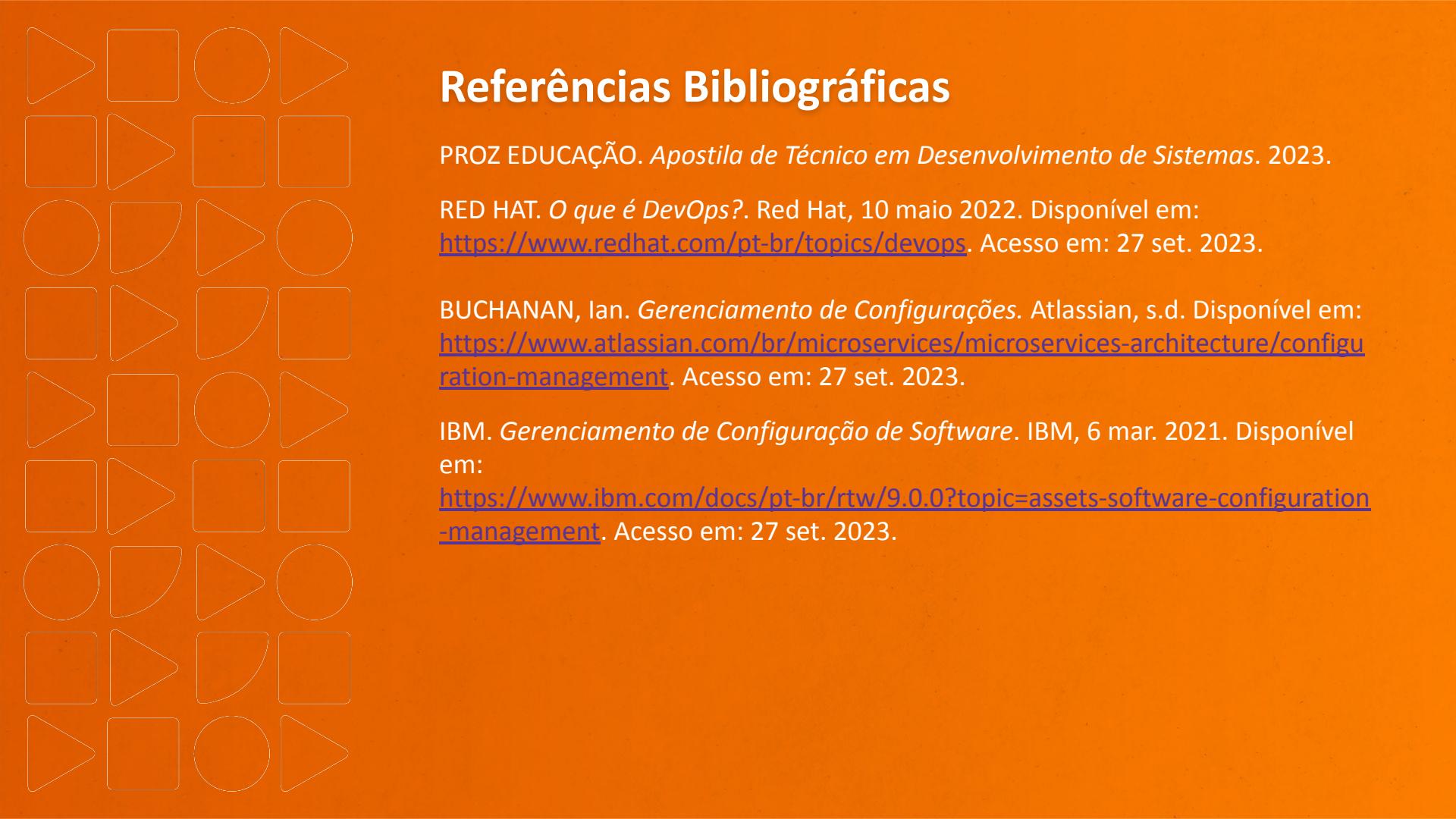
Quais mudanças de pensamento as aulas geraram em você?



## Para saber mais...



Vídeo do YouTube *Como é trabalhar com DevOps? - com Jeferson Fernando da LINUXTips #DEVaneio#Corte.*, do canal Marco Bruno (19 abr. 2021). Disponível em: <https://www.youtube.com/watch?v=zrrdkV1JCDE>. Acesso em: 27 set. 2023.



# Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Técnico em Desenvolvimento de Sistemas*. 2023.

RED HAT. *O que é DevOps?*. Red Hat, 10 maio 2022. Disponível em:  
<https://www.redhat.com/pt-br/topics/devops>. Acesso em: 27 set. 2023.

BUCHANAN, Ian. *Gerenciamento de Configurações*. Atlassian, s.d. Disponível em:  
<https://www.atlassian.com/br/microservices/microservices-architecture/configuration-management>. Acesso em: 27 set. 2023.

IBM. *Gerenciamento de Configuração de Software*. IBM, 6 mar. 2021. Disponível em:  
<https://www.ibm.com/docs/pt-br/rtw/9.0.0?topic=assets-software-configuration-management>. Acesso em: 27 set. 2023.