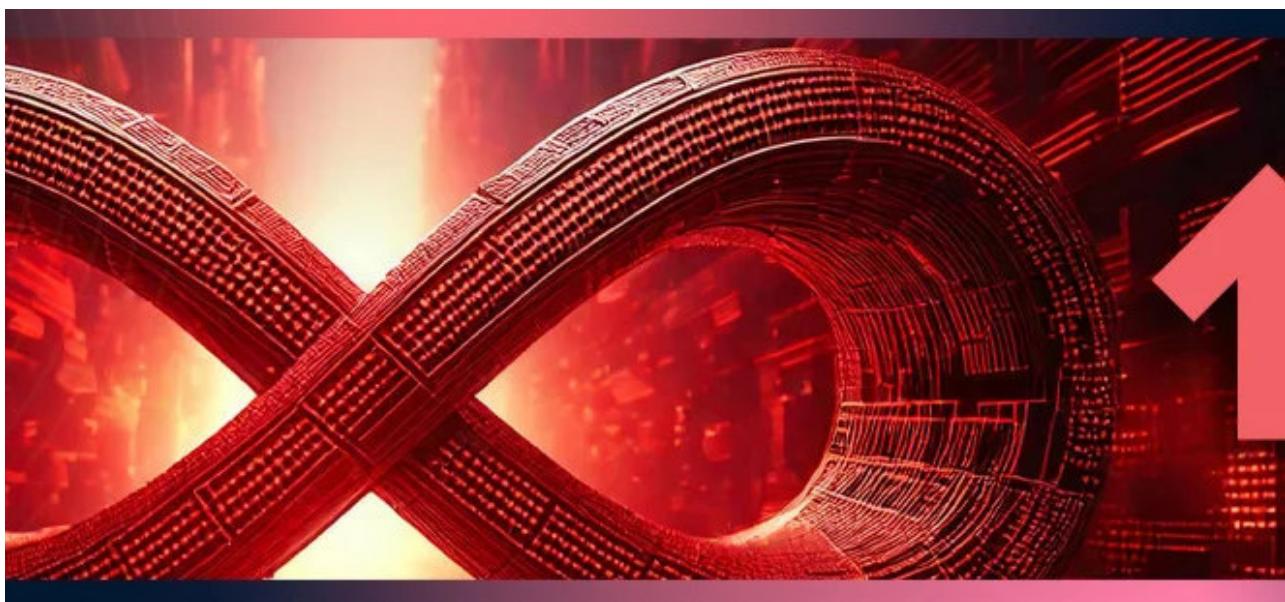


[VER PLANOS](#)[PROGRAMAÇÃO](#)[FRONT-END](#)[DATA SCIENCE](#)[INTELIGÊNCIA ARTIFICIAL](#)[DEVOPS](#)[UX & DESIGN](#)[MOBILE](#)[INOVAÇÃO & GESTÃO](#)[Artigos > DevOps](#)

Guia completo sobre a cultura DevOps para iniciantes

**Victor Costa Santos**

07/11/2024

[COMPARTILHE](#)

Você já ouviu falar sobre DevOps?



VER PLANOS

acelerar a entrega ao cliente.

Confira neste artigo:

- [Conclusão](#)

Imagine uma grande empresa de e-commerce que precisa lançar novas funcionalidades constantemente, sem comprometer a qualidade do serviço.

Últimos dias para participar da **Imersão Cloud DevOps**

Aprenda DevOps em três aulas gratuitas e pratique em projetos reais no Google Cloud.

GARANTA A SUA VAGA



Com DevOps, essa empresa pode implementar mudanças com mais frequência, corrigir erros rapidamente e reduzir o tempo de inatividade, melhorando a experiência do cliente.

Neste artigo, **vamos explorar como o DevOps transforma a maneira como empresas desenvolvem, entregam e mantêm suas aplicações, abordando suas principais práticas e os benefícios que oferecem ao ambiente corporativo.**

O que é DevOps?

Como já mencionamos, DevOps é uma cultura. Quando falamos de cultura, nos referimos a histórias, pensamentos e motivações geradas por uma série de eventos que levaram a uma evolução/revolução, resultando no que hoje conhecemos como DevOps.

Como surge o DevOps?



VER PLANOS

"embalado" pelo time de operações, que cuidava de toda a parte de envio, hospedagem e afins.

O problema era que o time de operações focava em estabilidade, enquanto os desenvolvedores precisavam lançar novas funcionalidades rapidamente. Isso gerava um "atrito" entre as equipes.



Com o tempo, empresas começaram a perceber que essa separação causava complicações, em um mercado que exige que novos recursos e atualizações sejam lançados com frequência.

Foi aí que o DevOps começou a ganhar força, sendo mencionado pela primeira vez em 2009 em uma palestra chamada "["10+ Deploys Per Day"](#)", apresentada por **John Allspaw** e **Paul Hammond**, que trabalhavam no Flickr.

Esse movimento ganhou ainda mais força quando um desses participantes e entusiasta, Patrick Debois, organizou a primeira conferência DevOpsDays no mesmo ano, a partir daí o assunto começou a ser difundido pelo mundo, ajudando a moldar essa cultura.

Benefícios do DevOps

[VER PLANOS](#)

produtividade e transformando a maneira de trabalhar como um todo.

Um ponto de destaque é a **redução de riscos**. Antes do DevOps, era comum que cada alteração fosse quase como um salto no escuro.

Com a adoção dessa abordagem, as equipes passam a trabalhar juntas, permitindo a identificação antecipada de problemas e a resolução de questões antes que afetem o produto final. Isso reflete diretamente na **qualidade das entregas**.

A **mudança passa a ser parte do processo**, permitindo que as empresas se mantenham competitivas, o que é crucial em um mercado dinâmico.

Isso garante que novos desafios sejam enfrentados com mais agilidade, sem comprometer a qualidade, que é tão importante.

Dessa forma, é estabelecido uma cultura de aprendizado contínuo com feedbacks constantes.

Assim, as equipes aprendem a cada nova implementação, ajustando o processo de forma mais iterativa. Isso aumenta a produtividade e fortalece a resiliência de empresas e times.

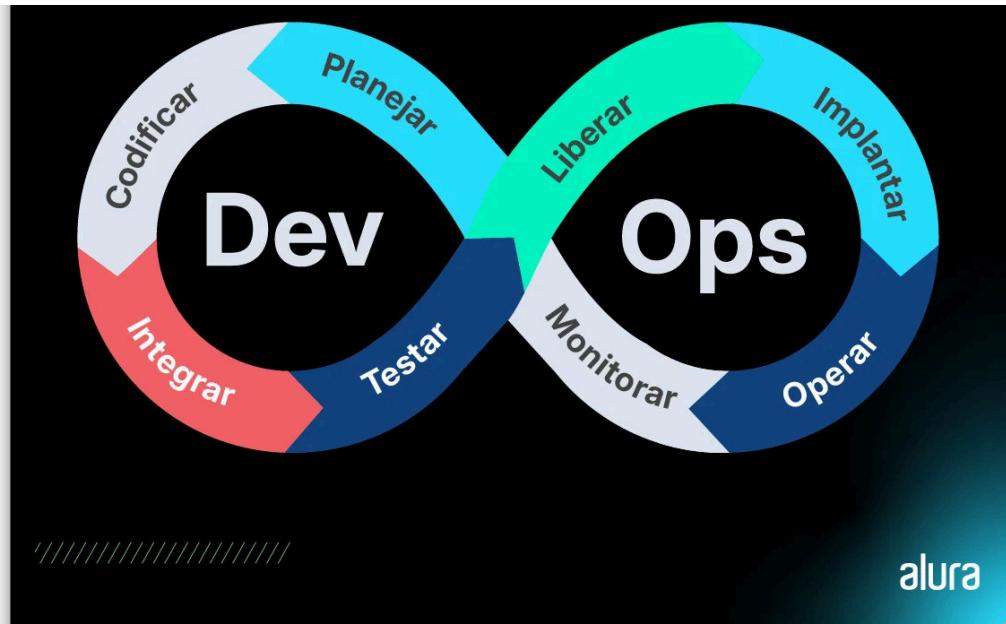
Ciclo de vida do aplicativo e DevOps

Quando falamos sobre o ciclo de vida de um aplicativo no contexto do DevOps, estamos lidando com um processo que envolve várias fases, que são interconectadas.

Diferente dos modelos antigos, em que cada fase tinha início e fim bem delimitados, o roadmap do DevOps pode ser representado por uma imagem que ilustra um movimento de um ciclo. Confira o ciclo do DevOps:



VER PLANOS

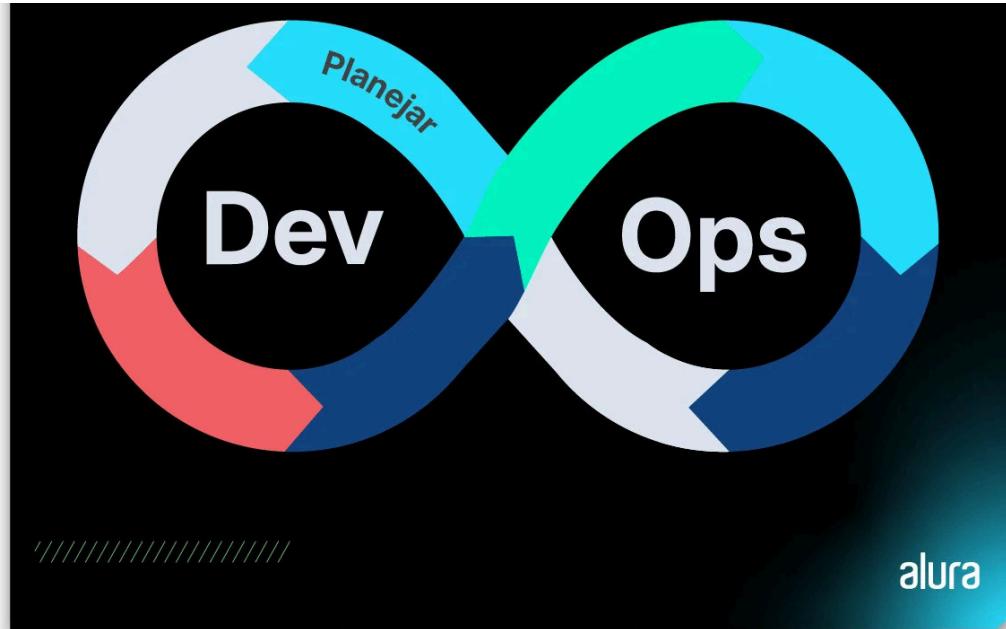


Para entender melhor cada fase desse ciclo, vamos refletir sobre um problema prático, no contexto de desenvolvimento de uma aplicação a seguir.

Planejamento (planejar)



VER PLANOS



Vamos então pensar em uma feature para um e-commerce, como um novo sistema de pagamento.

Nesse primeiro momento é quando vamos discutir os requisitos e objetivos que queremos atingir.

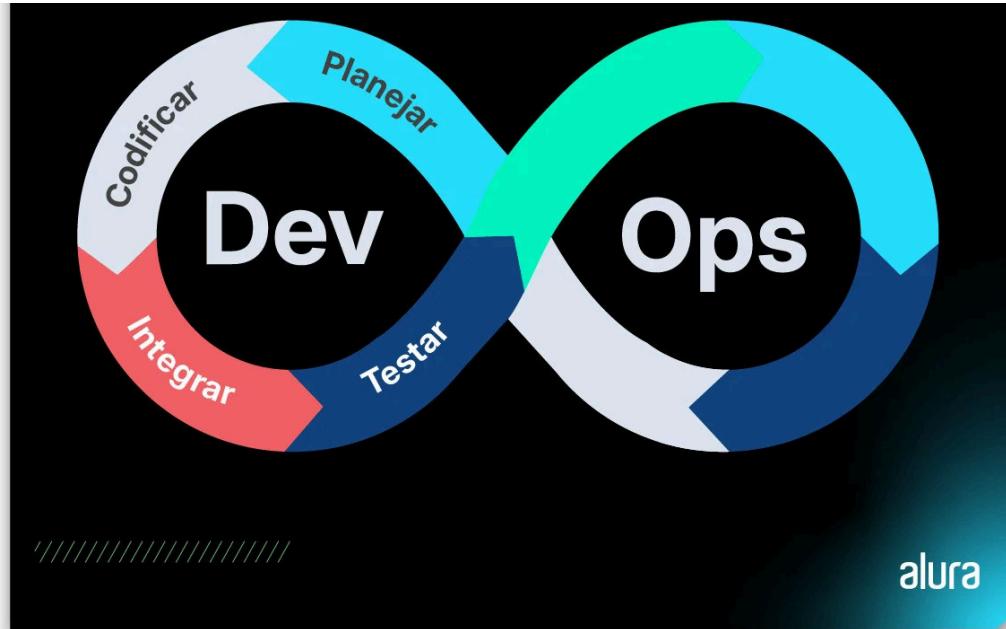
Aqui, a colaboração já é essencial, pois o feedback das operações é tão importante quanto as decisões de desenvolvimento.

Desenvolvimento (codificar, integrar, testar)

Logo depois do planejamento, começamos a “colocar a mão na massa”.



VER PLANOS



Na fase de desenvolvimento, as funcionalidades são criadas(codificar) e o código é constantemente integrado (integrar) e testado (testar). Em nosso roadmap, estamos na “etapa” de testar, a um passo de “liberar” o projeto.

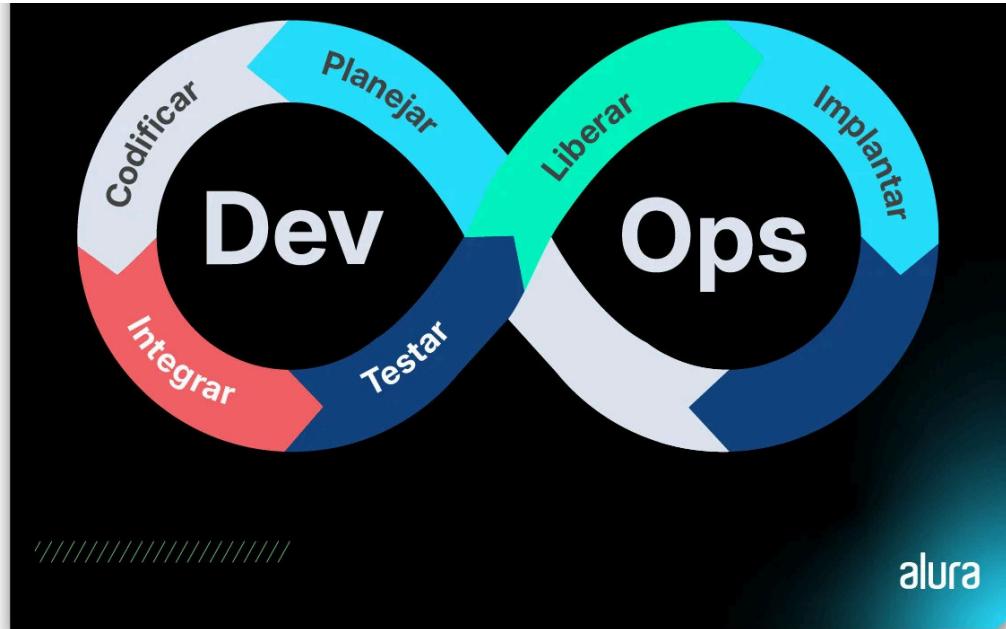
É importante ressaltar que precisamos realizar entregas rápidas e frequentes, mantendo o código sempre em um estado funcional.

Entrega (liberar, implantar)

Após passar pelas fases de planejamento e desenvolvimento, chegamos à fase de **entrega**.



VER PLANOS



Aqui, todo o esforço de codificação e testes resulta em um produto pronto para ser disponibilizado. No roadmap do DevOps, essa é a etapa que conecta o desenvolvimento ao ambiente de produção, do testar para o implantar.

A fase de entrega garante que as funcionalidades estão preparadas para o público final, sempre com foco em qualidade e eficiência.

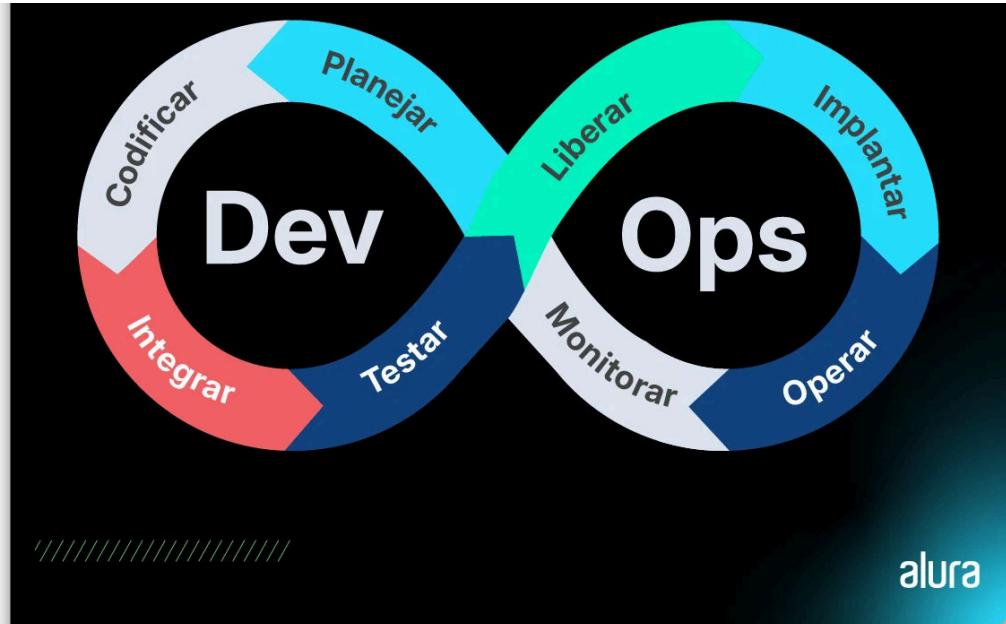
É nesse momento que o código é liberado para o ambiente de produção de forma ágil, sem causar interrupções significativas.

Operação (operar, monitorar)

Nesse momento o produto já está nas mãos dos usuários, chegamos à fase de **operação**.



VER PLANOS



No DevOps as equipes de operações trabalham lado a lado com os desenvolvedores, para operar e monitorar o aplicativo em produção.

Aqui, o desempenho é constantemente avaliado e eventuais problemas com essa nova **feature**, são rapidamente identificados e corrigidos.

Mas não acaba por aí, o ciclo continua com melhorias contínuas, em que o feedback da operação volta para o desenvolvimento, dando movimento contínuo ao ciclo.

Cultura de DevOps

A cultura do DevOps vai além das ferramentas e processos técnicos. É todo um movimento focado em **colaboração, alinhamento e melhoria contínua**.

Vamos nos aprofundar em alguns dos pilares desta cultura que faz do DevOps um modelo tão eficiente.



VER PLANOS



Colaboração

Ao longo desse artigo, já deu para notar que essa palavra foi muito mencionada e não é à toa.

Colaboração é alma do DevOps, já que a ideia central é não ter separações, e sim que todos jogam pelo mesmo time, com o propósito de fazer tudo funcionar corretamente.

Alinhamento

Esse pilar é crucial para o time, afinal não é possível, saber a direção de todo time, se todos não estão alinhados.

[VER PLANOS](#)

desenvolvimento.

E, falando em visibilidade...

Visibilidade

Além, do time está alinhado sobre a direção do barco, todos os tripulantes têm que enxergar a mesma ilha, quando o grito de "terra a vista" for dado.

Ou seja, o time tem que estar sintonizado, tanto sobre os problemas que estão ocorrendo quanto com o que está funcionando corretamente.

No mundo de DevOps, algumas ferramentas vão servir como "lunetas" para visualizar novos horizontes.

Mudanças de escopo

A capacidade de lidar com mudanças de escopo é crucial, afinal é comum no mundo de desenvolvimento, os requisitos mudam rapidamente.

Mas com o DevOps, ambas as equipes estão melhor preparadas para essas mudanças. Assim, caso a rota mude, não será necessário diminuir a velocidade.

Ciclos de versão mais curtos

Um benefício de destaque é a diminuição de tempo graças a ciclos de versões mais curtos, afinal tempo é dinheiro.

Com a adoção do DevOps as equipes podem entregar novas funcionalidades ou corrigir erros em menos tempo.

Melhorias e aprendizado contínuo

E pra fechar com chave de ouro, o que faz esse motor girar, é que com o DevOps há prática de melhorias e aprendizado contínuo.

O feedback constante e a análise de dados em tempo real permitem que as equipes aprendam com cada nova implementação, ajustando processos e



VER PLANOS

pilares:

<https://www.youtube.com/watch?v=feMN-iWMIk8>

Práticas do DevOps

Agora que já entendemos a importância da cultura DevOps, é hora de abordar as principais práticas que fazem essa engrenagem funcionar.

Essas práticas permitem que as equipes integrem, entreguem e monitorem seus softwares de forma mais eficiente e ágil.

Integração contínua e entrega contínua (CI/CD)

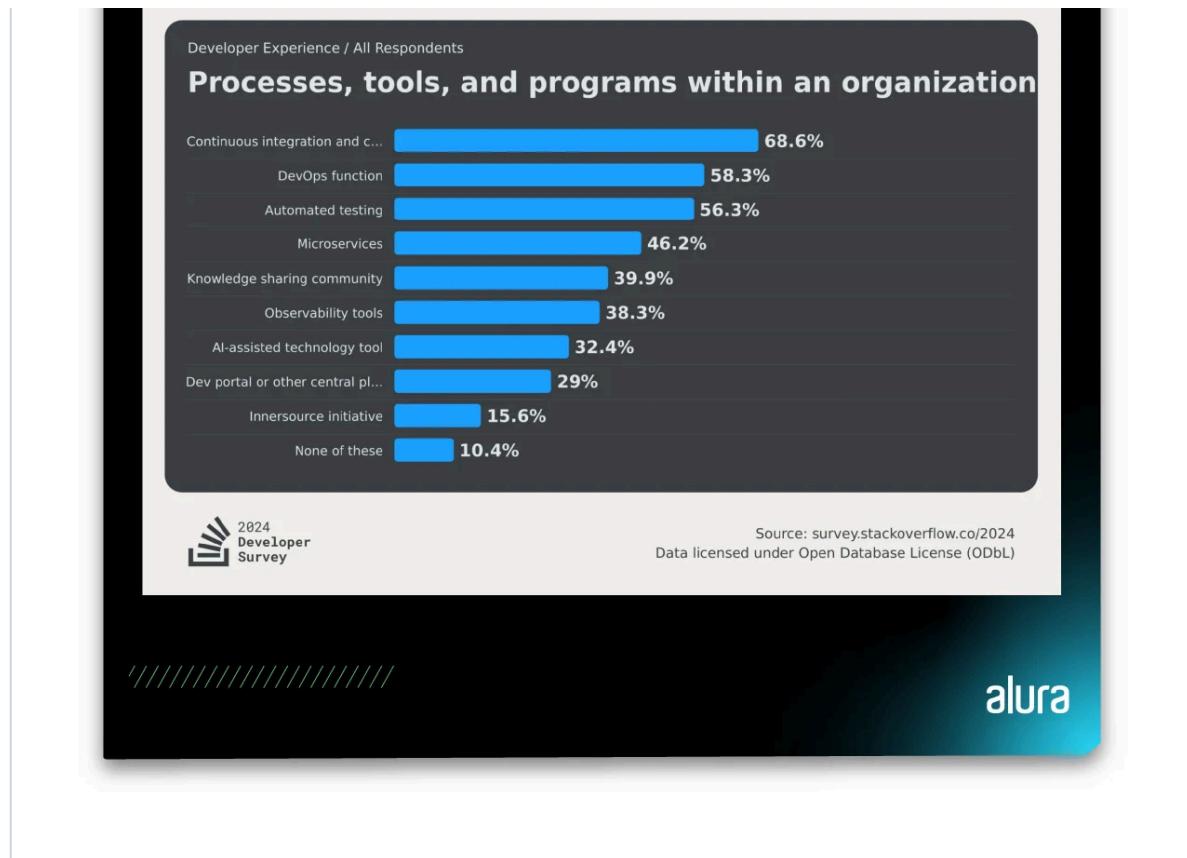
A Integração Contínua (CI) e a Entrega Contínua (CD) são termos populares nesse universo, pois são práticas essenciais no DevOps.

Na CI, o código é integrado e testado constantemente, enquanto a CD garante que essas mudanças estejam prontas para entrar em produção de forma rápida.

A pesquisa do Stackoverflow, aponta que 68,6% dos desenvolvedores utilizam esse processo em suas organizações.



VER PLANOS



Caso queira se aprofundar mais nesse processo que envolve várias etapas e ferramentas recomendo a formação de [Integração Contínua e Entrega Contínua](#).

Controle de versão

O controle de versão é uma prática essencial para rastrear e gerenciar mudanças no código.

Além de facilitar o trabalho em equipe, o controle garante que você possa voltar a uma versão anterior caso algo dê errado.

Essa prática é utilizada para manter a integridade do código durante todo o processo de desenvolvimento, é como um CTRL + Z.

Quer saber mais sobre esse assunto? Acesse o artigo [Começando – Sobre Controle de Versão](#).

Desenvolvimento de software Agile



VER PLANOS

Enquanto a metodologia tradicional seguia um plano rígido, o Agile permite ajustes e melhorias contínuas ao longo do desenvolvimento.

Isso significa que podemos identificar e corrigir falhas rapidamente, garantindo resultados mais eficientes.

Em vez de seguir uma rota fixa, essa abordagem flexível permite que as equipes se adaptem a mudanças de forma ágil, enfrentando os desafios do mercado com mais eficiência.

Para saber mais sobre a metodologia Agile, acesse o artigo [Metodologia Ágil - O que é?](#)

Infraestrutura como código

A **Infraestrutura como código(IaC)** é uma prática essencial pois ela se resume a ganho de tempo. Mas o que ela faz?

Essa prática permite que a configuração de servidores e ambientes seja de forma **automatizada**, da mesma forma que o código do software.

Isso elimina a necessidade de ajustes manuais, tornando o processo mais ágil e menos suscetível a erros humanos.

Com IaC, tudo é documentado e versionado, o que facilita a replicação e o controle das infraestruturas. Para se aprofundar nessa prática, recomendo a [formação de Infraestrutura como código](#).

Monitoramento contínuo

Outra prática importante é o monitoramento contínuo, com ele, as equipes acompanham o desempenho e a saúde dos sistemas em tempo real, identificando e corrigindo problemas antes que eles afetem os usuários.

Ferramentas de DevOps

Você já passou pelo ciclo de vida do DevOps e viu como cada fase — do planejamento à operação — é essencial para manter o fluxo contínuo. Agora, chegou a hora de explorar as ferramentas que fazem tudo isso acontecer.



VER PLANOS

para ajudar em todo o ciclo de produção do software, incluindo desenvolvimento, gerenciamento e entrega.

A escolha das ferramentas pode variar de acordo com o contexto e as necessidades de cada organização, mas os objetivos permanecem os mesmos:

- **Otimizar processos;**
- **Garantir a eficiência do ciclo de desenvolvimento;**
- **Garantir a eficiência do ciclo de operações.**

As organizações que utilizam a prática DevOps coordenam essas ferramentas e as encaixam em uma ou mais atividades de produção, como planejamento, teste, monitoramento e muito mais.

Vamos explorar algumas ferramentas populares que atuam em determinadas fases, movimentando todas as engrenagens. Vamos lá!

Ferramentas para planejamento

Nessa fase, podemos usar ferramentas como:

- [Jira](#)
- [Confluence](#)
- [Slack](#)

Elas ajudam equipes a organizar e documentar todas as tarefas necessárias, desde a definição de requisitos até a criação de documentação técnica.

Já para as inúmeras reuniões que surgem nesse processo, podemos usar:

- Zoom;
- Google Meet;
- Microsoft Teams.



VER PLANOS

Ferramentas para desenvolvimento

Para desenvolvimento, o **Git** é uma ferramenta indispensável. Pensando nas diversas linguagens de programação e [frameworks](#) que podem ser utilizadas, algumas plataformas se destacam para ajudar a coordenar o código, dessas as mais populares são:

- [GitHub](#): permite que devs versionem códigos e colaborem em projetos de maneira eficaz.
- [GitLab](#): além de controle de versão, o GitLab oferece recursos de CI/CD integrados, tornando o fluxo de trabalho ainda mais eficiente. Ele permite que as equipes automatizem seus processos de entrega e integrem testes diretamente em seu pipeline.
- [Bitbucket](#): essa ferramenta fornece controle de versão e suporte para repositórios Git.

Esses são alguns exemplos de plataformas que ajudam a manter o código em ordem em todo ciclo de desenvolvimento, garantindo a colaboração entre os desenvolvedores e assegurando que o trabalho em equipe seja mais produtivo e organizado.

Ferramentas para empacotamento/build

No processo Build/empacotamento, ou de integração, alguns nomes de peso aparecem.

Essas ferramentas são essenciais para compilar e preparar o código para produção. As mais populares no mercado, são:

- **Docker**: é a cara do DevOps, ele ajuda no processo de Build, empacotamento, execução e afins, essa ferramenta daria um artigo só para ela, sendo assim recomendo a leitura do artigo [O que é o Docker e quais as Vantagens de usá-lo?](#)
- **Maven**: é muito utilizado para build e gerenciamento de dependências em projetos Java. Caso queira aprender mais sobre essa ferramenta recomendo o artigo ["Conhecendo melhor o Maven"](#).



VER PLANOS

Compreender como o CI/CD, tasks e controles básicos.

Essas ferramentas facilitam a transição entre as etapas do desenvolvimento, minimizando erros e mantendo a consistência.

Ferramentas para testes

As ferramentas para testes visam mitigar todos os possíveis erros, evitando que essas falhas cheguem ao usuário final.

Algumas opções que se destacam e são utilizadas para diferentes [tipos de testes](#) são:

- [Selenium](#): automatiza os testes de interface, simulando o uso dos usuários e analisando erros potenciais ou erros concretos.
- [Snyk](#): é uma solução bem completa, e muito utilizada, para fins de segurança. Essa ferramenta busca garantir que o software esteja protegido contra ameaças.
- [Jmeter](#): focado em testes de carga e desempenho, o JMeter ajuda a medir a performance de aplicações web e servidores, simulando múltiplos usuários e transações para avaliar como o sistema se comporta sob pressão.

Ferramentas para liberação e implantação

Depois de testados, é chegada a hora da liberação ou **release**. Algumas ferramentas são essenciais para que tudo ocorra bem na hora implantar o código. Algumas opções, são:

- [Jenkins](#): é muito popular, dado sua capacidade de automação, se DevOps são um conjunto de engrenagens, o Jenkins funciona como uma graxa, melhorando a performance e otimizando todo o trabalho e ciclo.
- [GitHub Actions](#): permite automatizar tarefas diretamente no GitHub, otimizando o processo de entrega.



VER PLANOS

O que é o projeto Kubernetes e para que ele serve: .

Ferramentas para gerenciamento de configuração e automação

Além das ferramentas de liberação e implantação, o gerenciamento de configuração e automação é crucial para garantir a eficiência em DevOps. Segue algumas opções:

- [**Terraform**](#): ajuda equipes a criar e gerenciar ambientes em nuvem de forma ágil e segura, promovendo consistência e reduzindo erros.
- [**Ansible**](#): é uma ferramenta de automação que facilita a configuração, gerenciamento e implementação de aplicações.

Ferramentas para operação e monitoramento

Por fim, as ferramentas de operação e monitoramento são essenciais para garantir que tudo funcione bem após a implantação e se algo passar errado e surgir erros, ela rapidamente avisará, "ops, algo de errado não está certo" rs. Dessa ferramentas temos:

- [**New Relic**](#): oferece *insights* em tempo real sobre o desempenho das aplicações, ajudando a identificar falhas rapidamente.
- [**Datadog**](#): monitora tanto a infraestrutura quanto às aplicações, coletando métricas e eventos para análise.
- [**Prometheus**](#) e [**Grafana**](#): essa dupla em conjunto, ajudam a coletar e visualizar métricas, proporcionando uma visão clara do desempenho do sistema.

Importante salientar que essas ferramentas são bem completas, sendo possível usar a mesma ferramenta em diversos estágios do ciclo de vida de um software no DevOps. Seja no desenvolvimento, CI/CD, testes,



VER PLANOS

DevOps e a nuvem



DevOps sem a nuvem é como Harry Potter sem Hermione — você pode até ter muito poder, mas sem a sabedoria e o suporte, as coisas ficam bem mais complicadas.

A adoção de soluções em nuvem revolucionou o desenvolvimento e a operação de software. **DevOps e a nuvem caminham lado a lado**, permitindo entrega contínua, escalabilidade e, acima de tudo, **agilidade**.

Então, vamos conhecer as vantagens de trabalhar com essas ferramentas em conjunto!

Agilidade

A nuvem possibilita uma entrega mais eficiente de recursos.

[VER PLANOS](#)

Escalais aplicações em questão de minutos, em vez de semanas.

Gigantes como **AWS** da Amazon, **Microsoft Azure** e **Google Cloud** facilitam essa transição e garantem essa rapidez.

Kubernetes

Quando falamos de DevOps na nuvem, não podemos ignorar o [Kubernetes](#). Aqui vale uma menção extra, que vai além de mais uma ferramenta.

Tanto no ambiente de nuvem quanto local, o Kubernetes é essencial para garantir que as aplicações rodem de forma estável e escalável, gerenciando automaticamente recursos e balanceando a carga entre os servidores.

Seja na [AWS](#), [Azure](#) ou Google Cloud, o Kubernetes facilita a administração de milhares de containers em múltiplos ambientes, sem que os times precisem se preocupar com a infraestrutura manualmente.

Essa parceria entre o Kubernetes e provedores de nuvem faz com que o desenvolvimento e a operação sejam muito mais ágeis, especialmente em sistemas complexos que exigem alta disponibilidade.

A título de curiosidade, caso esses termos tenham suados estranhos esse **#HipstersPontoTube**, esses termos serão desmistificados:

<https://www.youtube.com/watch?v=wxLvvMxzc1Q>

Computação sem servidor

A [computação serverless](#) (ou sem servidor) é outra peça fundamental nesse quebra-cabeça.

Com ela, as empresas podem executar seus aplicativos sem a necessidade de gerenciar servidores ou infraestrutura. Isso permite que os desenvolvedores foquem mais na criação de funcionalidades e menos em questões de infraestrutura.

Serviços como [AWS Lambda](#), [Azure Functions](#), e [Google Cloud Functions](#) permitem que as aplicações escalem automaticamente, conforme a necessidade, sem que o time de operações precise intervir manualmente.

[VER PLANOS](#)

integrar a segurança no ciclo de desenvolvimento desde o início.

Se o DevOps revolucionou a colaboração entre as equipes de desenvolvimento(DEV) e de operações(OPS), o DevSecOps adiciona uma camada extra de cuidado com a segurança, daí vem o "Sec".

O DevSecOps tem como objetivo garantir que todos os envolvidos no desenvolvimento, tenham responsabilidade pela segurança.

Na prática, isso significa garantir que a segurança esteja integrada no ciclo de desenvolvimento desde o início, seja por meio de revisões de código mais rigorosas, testes automatizados de vulnerabilidades ou políticas de conformidade.

O DevSecOps transforma a **segurança em uma prioridade contínua**, em vez de um processo isolado no final.

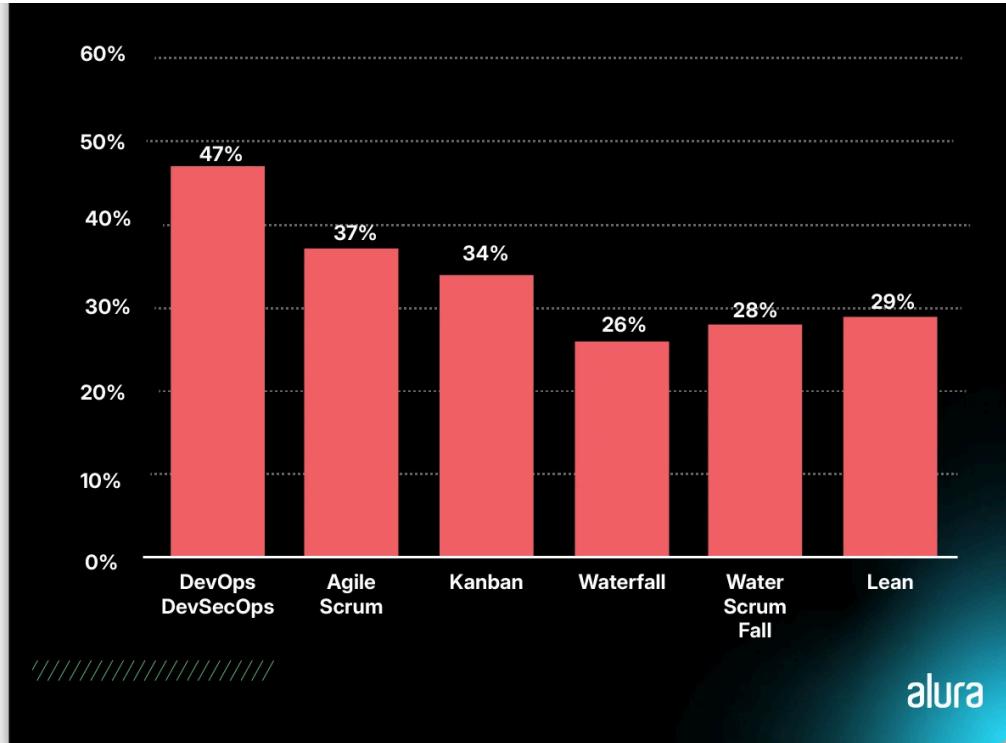
Se você quer entender mais sobre como DevSecOps funciona e por que ele é essencial para a segurança no desenvolvimento moderno, recomendo a leitura do artigo sobre [DevSecOps](#).

Oportunidades de carreira na área de DevOps

A essa altura, podemos concluir que DevOps está em constante crescimento. Atualmente já é a metodologia mais utilizada no mundo segundo uma pesquisa realizada pela [statista](#):



VER PLANOS



A cada dia Devops vem sendo mais adotado e, consequentemente, carreiras nessa área estão a todo vapor, aumentando assim a demanda por profissionais especializados.

O mercado global de tecnologia ultrapassou os US\$ 8 bilhões em 2022, muito impulsionado pelo cloud. E é esperado que até 2032, esse mercado chegue a US\$ 58,88 bilhões segundo [pesquisas](#).

Esse crescimento reflete a adoção massiva do DevOps como metodologia padrão de desenvolvimento de software em empresas de todos os tamanhos.

Vamos explorar alguns dos fatores que impulsionam esse crescimento e o que é necessário para aproveitar essas oportunidades.

Salários competitivos e estabilidade

No Brasil, o salário de um profissional de DevOps gira em torno R\$ 6.000,00 e R\$ 15.000,00 mensais com uma **média de R\$ 8.500,00**, aponta o [glassdoor](#).

Alguns profissionais mais experientes e com habilidades mais escassas no mercado podem chegar a receber salários ainda mais altos.



VER PLANOS

A adoção de DevOps não está restrita a organizações focadas em tecnologia.

Bancos, seguradoras, e-commerce, telecomunicações e até indústrias de manufatura estão buscando implementar práticas DevOps para aumentar sua eficiência e agilidade.

Isso amplia ainda mais o leque de oportunidades, já que profissionais podem gerar soluções personalizadas para cada setor.

O que faz um profissional de DevOps

O profissional de DevOps é, basicamente, o elo entre o desenvolvimento e as operações.

Ele não apenas garante que o software seja entregue de forma mais ágil, mas também que ele funcione com segurança e eficiência quando estiver em produção.

Mas é claro, o papel de um profissional de DevOps vai muito além de automatizar pipelines e configurar servidores como já exploramos ao longo deste artigo.

Quais habilidades são necessárias para trabalhar com DevOps?

Para atuar em DevOps, você precisa ser versátil. Um dia você pode estar codificando scripts de automação e no outro, garantindo que o ambiente de produção esteja rodando de forma estável.

As principais áreas que você deve desenvolver suas habilidades:

- **Automação:** um dos principais pilares de DevOps. Automatizar tarefas manuais é essencial para acelerar entregas e evitar erros humanos.
- **Monitoramento contínuo:** garantir que o software esteja sempre rodando de forma estável, monitorando métricas de desempenho e prevenindo falhas antes que elas se tornem grandes problemas.



VER PLANOS

- **Infraestrutura como código (IaC)**: saber configurar servidores e ambientes de forma programática, garantindo consistência e escalabilidade.

Principais carreiras em DevOps

Agora que você já tem uma ideia do que é preciso para trabalhar como DevOps, vamos falar sobre as carreiras para trilhar dentro desse universo:

- **Engenheiro de DevOps**: um dos cargos mais comuns, esse profissional é o responsável por construir e manter os pipelines de CI/CD e garantir que todo o processo de entrega de software funcione sem interrupções.
- **SRE (Site Reliability Engineer)**: aqui, o foco é na confiabilidade do sistema. Você estará monitorando, identificando falhas e garantindo que a aplicação continue estável, mesmo sob alta demanda.
- **Administrador de sistemas DevOps**: responsável por toda a parte de infraestrutura, seja em nuvem ou local, garantindo que os servidores e redes estejam configurados corretamente e rodando com segurança.
- **Arquiteto de soluções DevOps**: se você gosta de ver o quadro completo, essa é uma carreira interessante. Aqui, você vai desenhar soluções que integram desenvolvimento, operação e infraestrutura de forma harmoniosa.

Quer se destacar? Invista em certificações!

Se destacar no mercado é muito importante, e as certificações são um caminho certo para mostrar que você domina as melhores práticas e ferramentas. Algumas das mais valorizadas no mercado são:

- **AWS Certified DevOps Engineer**: para quem quer se especializar nas práticas DevOps dentro da infraestrutura da AWS.
- **Google Professional DevOps Engineer**: essa certificação foca nas práticas DevOps dentro do Google Cloud, ideal para quem deseja



VER PLANOS

Kubernetes é uma das principais tecnologias. Essa certificação garante que você saiba como gerenciar, implantar e escalar aplicações usando essa ferramenta.

Essas certificações não apenas mostram ao mercado que você está preparado, mas também te ajudam a desenvolver habilidades práticas e concretas que são essenciais em qualquer equipe DevOps.

Quer aprender sobre DevOps com projetos reais e de forma gratuita?

Aprenda DevOps na prática na [Imersão DevOps](#), com propósito e direção, em **três aulas gratuitas** dos dias **01 a 03 de julho de 2025**, com especialistas da Alura e do Google Cloud.

Faça sua inscrição gratuita na [página oficial da Imersão Cloud DevOps](#) até dia 30 de junho e prepare-se para mergulhar em DevOps enquanto pratica em projetos reais com o Google Cloud.

Cada aula te leva pra um novo nível de conhecimento, de forma totalmente prática.

- **Conteúdo direto ao ponto:** três aulas práticas para você conhecer cultura, ferramentas e bons hábitos de DevOps.
- **Acesso ao Google Cloud:** acesso exclusivo ao Google Cloud Platform para testar e validar seu projeto.
- **Masterclass introdutória:** não sabe nada sobre o assunto? Você terá acesso antecipado a um aula introdutória sobre DevOps.
- **Certificado da Alura e Google Cloud:** ao assistir todas as aulas, você vai receber um certificado de participação para potencializar seu currículo.

Para quem é a Imersão Cloud DevOps?

A Imersão Cloud DevOps é ideal para você que já tem, pelo menos, conhecimento básico em programação e é:



VER PLANOS

- Profissional de infra que quer mergulhar em DevOps

Conclusão

Quanta coisa até aqui, não é mesmo?

Ao longo deste artigo, vimos como o DevOps transformou a maneira como as empresas desenvolvem, operam e escalam suas aplicações.

Essa abordagem multifuncional une desenvolvimento, operações e, mais recentemente, segurança, com a introdução do DevSecOps.

Como vimos, o impacto dessa cultura vai além de melhorar a produtividade: ela redefine a colaboração, acelera entregas e aumenta a qualidade e segurança dos produtos finais, impactando em toda a cultura da organização.

Se você está buscando construir uma carreira ou consolidar seus conhecimentos nessa área, essa é a hora!

A demanda por profissionais capacitados só aumenta, e as oportunidades se expandem em diversos setores do mercado.

A Alura oferece, não só um curso, mas diversas formações, para te ajudar a desenvolver habilidades nas áreas de automação, segurança e práticas DevOps mais avançadas.

A Alura tem o conteúdo que você precisa para ser um profissional completo e indispensável. O Paulo Silveira, CEO da Alura, tem o recadinho do que veremos por lá:

<https://www.youtube.com/watch?v=fSBLPaj0aWM&t=12s>

Se interessou ainda mais? Não deixe de verificar todas as possibilidades:

- [Escola de DevOps Alura](#)

Por fim, foi um enorme prazer compartilhar com você esse universo incrível, repleto de possibilidades que estão transformando a maneira como desenvolvemos softwares.



VER PLANOS

NÃO FIQUE DE FORA!



As inscrições para a Imersão Cloud DevOps estão quase acabando, mas ainda dá tempo de garantir sua vaga e virar referência no assunto.

INSCREVA-SE GRATUITAMENTE

**Victor Costa Santos**

Olá, sou o Victor, formado em jogos digitais pela FIAP, sou um entusiasta da tecnologia e apaixonado pela suas possibilidades, adoro ler e me informar sobre o que se passa no mundo, Jogar é claro, e ver filmes e séries que trazem reflexão sobre o seu eu, acredito

[VER PLANOS](#)[Artigo Anterior](#)

O que é Gateway: como essa ferramenta facilita a comunicação na internet

[Próximo Artigo](#)

Kubernetes: como usá-lo para gerenciar suas aplicações



Veja outros artigos sobre
DevOps

Quer mergulhar em tecnologia e aprendizagem?

Receba conteúdos, dicas, notícias, inovações e tendências sobre o mercado tech diretamente na sua caixa de entrada.

bins.br@gmail.com

ENVIAR

Nossas redes e apps

[VER PLANOS](#)

Institucional

[Sobre nós](#)[Carreiras Alura](#)[Para Empresas](#)[Para Sua Escola](#)[Política de Privacidade](#)[Compromisso de Integridade](#)[Termos de Uso](#)[Documentos Institucionais](#)[Status](#)[Conteúdos](#)[Alura Cases](#)[Imersões](#)[Artigos](#)[Podcasts](#)[Artigos de educação corporativa](#)[Imersão Cloud Devops](#)

A Alura

[Como Funciona](#)[Formações](#)[Plataforma](#)[Depoimentos](#)[Instrutores\(as\)](#)[Dev em <T>](#)[Luri, a inteligência artificial da Alura](#)[IA Conference 2025](#)[Cursos imersivos](#)[Certificações](#)

Fale Conosco

[Email e telefone](#)[Perguntas frequentes](#)

[VER PLANOS](#)

bins.br@gmail.com

[ENVIAR](#)

CURSOS

Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

Cursos de Inteligência Artificial

IA para Programação | IA para Dados

Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design

Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile

Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas

CURSOS UNIVERSITÁRIOS FIAP

Graduação | Pós-graduação | MBA