

# DEVOPS BÁSICO

---

Fabrício Veronez e Marco Aurélio Souza Mangan



Qualquer bom profissional de desenvolvimento de software deve conhecer paradigmas.



*Alessandro Valerio*

# Conheça o livro da disciplina

## CONHEÇA SEUS PROFESSORES 3

*Conheça os professores da disciplina.*

## EMENTA DA DISCIPLINA 4

*Veja a descrição da ementa da disciplina.*

## BIBLIOGRAFIA DA DISCIPLINA 5

*Veja as referências principais de leitura da disciplina.*

## O QUE COMPÕE O MAPA DA AULA? 6

*Confira como funciona o mapa da aula.*

## MAPA DA AULA 7

*Veja as principais ideias e ensinamentos trabalhados ao longo da aula.*

## RESUMO DA DISCIPLINA 32

*Relembre os principais conceitos da disciplina.*

## AVALIAÇÃO 33

*Veja as informações sobre o teste da disciplina.*

# Conheça seus professores



## FABRÍCIO VERONEZ

Professor Convidado

Fundador da Formação KubeDev, é um grande apaixonado por tecnologias. Com mais de 13 anos de experiência no mercado de tecnologia, atuando como desenvolvedor e arquiteto em projetos de pequeno e grande porte, atualmente dedica-se a transformar a carreira de profissionais de TI, compartilhando conhecimento sobre o universo de containers e a cultura DevOps. Em 2018, passou a compartilhar a sua experiência através da criação de conteúdos. Foi assim que surgiu o seu site veronez.dev e, em seguida, o seu canal no YouTube.

## MARCO AURÉLIO SOUZA MANGAN

Professor PUCRS

Possui graduação em Ciências da Computação pelo Instituto de Informática/UFRGS, mestrado em Ciências da Computação pelo Instituto de Informática/UFRGS e doutorado em Engenharia de Sistemas e Computação pelo PESC/UFRJ. Atualmente, é professor da Faculdade de Informática/PUCRS e da Faculdade Senacrs Porto Alegre. Tem experiência na área de Ciência da Computação, com ênfase em linguagens de programação, atuando principalmente nos seguintes temas: engenharia de software, CSCW, ambientes de desenvolvimento de software, reutilização de software e mecanismos de cooperação.



# *Ementa da Disciplina*

Introdução aos fundamentos de gerência de configuração. Estudo sobre Integração contínua (CI). Utilização de contêineres, ferramentas e ambientes direcionados ao desenvolvimento de software como Git, GitHub, Maven, Gradle, Npm, Yarn, GitHub Actions, Jenkins, Travis e Docker.

# Bibliografia da disciplina

As publicações destacadas têm acesso gratuito.

## Bibliografia básica

IEEE. IEEE 2675-2021 - IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment. IEEE Xplore, abr. 2021.

MONTEIRO, Eduarda Rodrigues et al. **DevOps**. Porto Alegre: Sagah, 2021.

SHARMA, Sanjeev. **DevOps adoption playbook**: a guide to adopting DevOps in a multi-speed IT enterprise. Indianapólis: Wiley, 2017.

## Bibliografia complementar

FREEMAN, Emily. **DevOps para leigos**. Rio de Janeiro: Alta Books, 2021.

GITLAB. **Documentation**. [s.l.]: GitLab, 2021.

GOOGLE CLOUD. **DevOps**. [s.l.]: Google, 2021.

MUNIZ, Antonio et al. **Jornada DevOps**: unindo cultura ágil, Lean e tecnologia para entrega de software de qualidade. Rio de Janeiro: Editora Brasport, 2020.

RAVICHANDRAN, Aruna; TAYLOR, Kieran; WATERHOUSE, Peter. **DevOps for digital leaders**. Berkeley: APress, 2016.

# O que compõe o Mapa da Aula?

## MAPA DA AULA

São os capítulos da aula, demarcam momentos importantes da disciplina, servindo como o norte para o seu aprendizado.



## EXERCÍCIOS DE FIXAÇÃO

Questões objetivas que buscam reforçar pontos centrais da disciplina, aproximando você do conteúdo de forma prática e exercitando a reflexão sobre os temas discutidos. Na versão online, você pode clicar nas alternativas.



## PALAVRAS-CHAVE

Conceituação de termos técnicos, expressões, siglas e palavras específicas do campo da disciplina citados durante a videoaula.



## VÍDEOS

Assista novamente aos conteúdos expostos pelos professores em vídeo. Aqui você também poderá encontrar vídeos mencionados em sala de aula.



## PERSONALIDADES

Apresentação de figuras públicas e profissionais de referência mencionados pelo(a) professor(a).



## LEITURAS INDICADAS

A jornada de aprendizagem não termina ao fim de uma disciplina. Ela segue até onde a sua curiosidade alcança. Aqui você encontra uma lista de indicações de leitura. São artigos e livros sobre temas abordados em aula.



## FUNDAMENTOS

Conteúdos essenciais sem os quais você pode ter dificuldade em compreender a matéria. Especialmente importante para alunos de outras áreas, ou que precisam relembrar assuntos e conceitos. Se você estiver por dentro dos conceitos básicos dessa disciplina, pode tranquilamente pular os fundamentos.

## CURIOSIDADES

Fatos e informações que dizem respeito a conteúdos da disciplina.

## DESTAQUES

Frases dos professores que resumem sua visão sobre um assunto ou situação.

## ENTRETENIMENTO

Inserções de conteúdos para tornar a sua experiência mais agradável e significar o conhecimento da aula.

## CASE

Neste item, você relembra o case analisado em aula pelo professor.

## MOMENTO DINÂMICA

Aqui você encontra a descrição detalhada da dinâmica realizada pelo professor.

# Mapa da Aula

Os tempos marcam os principais momentos das videoaulas.

## AULA 1 • PARTE 1

“ Tudo que um software precisa ter é feito na etapa de planejamento. ”

“ O profissional de operação é o cara que vai cuidar do software que está em implantação, que está sendo executado. ”

### Desenvolvimento e operação

O profissional de desenvolvimento conhece profundamente os conceitos de programação, mas possui pouco ou nenhum conhecimento sobre infraestrutura. Seu objetivo é entregar novos recursos, não estabelecendo qualquer contato com o processo de execução do software em um ambiente de produção.

O profissional de operação, por sua vez, conhece profundamente a infraestrutura, mas possui pouco ou nenhum conhecimento sobre programação. Seu objetivo é manter a solução estável, não estabelecendo qualquer contato com o desenvolvimento e a compilação do software.

Normalmente, esses dois profissionais trabalham sob gestões diferentes - o diretor de desenvolvimento e o diretor de infraestrutura. Essa separação pode gerar conflitos de interesse e conflitos políticos.

01:46



### Etapas

02:54



04:35



05:39



Na visão do usuário, o desenvolvimento de uma aplicação, de um software, depende de um grupo de programadores que, dominando os códigos necessários, pode criar soluções e colocá-las à disposição do mercado. Para os desenvolvedores, no entanto, esse processo é visto de forma muito mais complexa.

Iniciado no planejamento, quando são levantados os requisitos do software que será desenvolvido, o programa é de fato desenvolvido na fase de desenvolvimento, quando os módulos e as funcionalidades são construídos. Em seguida, a aplicação é submetida a testes; e, por fim, o software é disponibilizado aos usuários na fase de implantação.

Em todas essas etapas, duas figuras estão ativamente envolvidas com o processo: o profissional de desenvolvimento e o profissional de operação. Separados por uma verdadeira barreira, os dois profissionais não se comunicam entre si ao longo das etapas, além de possuírem características bastante distintas.

06:37



“ O profissional de desenvolvimento quer sempre mudar, quer sempre colocar novas features, atualizar o software, mas o profissional de operação não quer. ”

09:29

**DevOps**

11:23

**“** Deu um problema? Os dois [o desenvolvedor e o profissional de operação] vão se debruçar sobre ele para tentar entender o que está acontecendo, resolver e documentar. **”**

12:57

**“** A coisa básica que você precisa fazer pra começar a trabalhar com DevOps é pensar na análise e na otimização, dos processos. **”**

16:04

**PALAVRA-CHAVE**

**Observabilidade:** Na teoria de controle, a observabilidade determina quanto bem podem os estados de um sistema ser inferidos a partir do conhecimento de suas saídas externas. Assim, podemos saber tudo o que se passa com uma aplicação. A observabilidade depende de três pilares: coleta de métricas, logs e tracing.

Junção do desenvolvedor com o profissional de operação, o DevOps atua com os interesses que existem em comum entre as duas funções, com foco no produto e na entrega. Assim, os dois profissionais participam ativamente do processo, independentemente da etapa. A comunicação é ágil e simplificada, rompendo com a barreira outrora existente. O objetivo é sempre resolver o problema e aprender com ele. Além disso, o DevOps é focado em três maneiras: fluxo, feedback, aprendizado contínuo e experimentação.

No caso do fluxo, destaca-se a análise e a otimização dos processos, a inclusão de testes, a integração e o deploy contínuos, e as entregas de baixo risco. No caso do feedback, as métricas são implementadas e coletadas, há observabilidade, testes A/B e feedbacks dos resultados para replanejamento. Por fim, no caso do aprendizado contínuo, aprendemos com os erros, promovemos a experimentação controlada e disseminamos o conhecimento, padronizando aquilo que funciona de forma adequada.

16:50



21:11

**PALAVRA-CHAVE**

**Teste A/B:** Método de teste de design através do qual são comparados elementos aleatórios com duas variantes, A e B, em que estes são o controle e o tratamento de uma experiência controlada, com o objetivo de melhorar a percentagem de aprovação.

**“** Se você tá pensando que DevOps é só esse conjunto de ferramentas, você tá enganado. DevOps tá ligado a cultura, processo, colaboração, experimentação... **”**

## EXERCÍCIO DE FIXAÇÃO

Assinale a alternativa incorreta a respeito dos profissionais de desenvolvimento e de operação.



## AULA 1 • PARTE 2

00:24



### Twelve Factor Apps

Trata-se de um conjunto de doze fatores e princípios que, quando seguidos, tornam possível garantir a criação de aplicações como serviço com grande eficiência. São usados formatos declarativos para a configuração de ambientes e para facilitar a automação, garantindo maior portabilidade entre os ambientes. Com o Twelve Factor Apps, é possível haver compatibilidade com plataformas de nuvem sem depender de servidores ou da administração de sistemas.

## CURIOSIDADE

### Heroku



Criada em 2007 por James Lindenbaum, Orion Henry e Adam Wiggins, é uma plataforma como serviço em nuvem com suporte para várias linguagens de programação. A Heroku foi a primeira empresa a entrar de vez no território da nuvem pública. Assim surgiu o Twelve-Factor App, uma metodologia usada na construção de aplicações SaaS.

#### Fatores

- 1) **Base de código:** cada aplicação possui uma base de código única e centralizada, de forma que todos os deploys são reflexo da base de código;
- 2) **Dependências:** declarar as dependências da aplicação de forma explícita com gerenciadores de pacote;
- 3) **Configurações:** armazenamento de configurações no ambiente. Esse recurso garante portabilidade, além de que não haverá problemas no transporte dos arquivos de configuração;
- 4) **Backing services:** tratar os serviços como recursos;
- 5) **Construa, lance e execute:** separar os estágios de build e release dos projetos. Assim, a construção do processo é muito mais prática e fácil;
- 6) **Processos:** executar a aplicação como um ou mais processos que não armazenam estados. Há processos “stateless”, que não armazenam nenhum dado, e processos “stateful”, em que são armazenados alguns dados. Os processos, portanto, consultam o serviço de apoio ao invés de criar



00:44

Um dos seus objetivos é minimizar o impacto da migração entre ambientes de desenvolvimento e de produção, além de que a aplicação é escalável sem que seja preciso se preocupar com alterações significativas para garantir a escalabilidade.



### THE TWELVE-FACTOR APP

02:37



03:35



03:51



11:21



13:39



“ O foco do Twelve Factor Apps é pensar em software como serviço. Então, quanto menos você se preocupa com elementos de infraestrutura e operação, melhor fica. ”

“ É muito comum que as aplicações sejam desenvolvidas em vários repositórios, pastas e subdiretórios. Isso é ruim porque a sua aplicação tem que representar apenas uma base de código. ”

“ Quando você começa a pensar em serviços, não em devices ou em tecnologias, você consegue ter abstrações. Então, quando você pensar em serviços de apoio, trate como serviços realmente. ”

“ Quando você for construir uma aplicação, você tem que pensar nela como algo stateless; você não tem que armazenar nada dentro dela. ”

réplicas, duplicando o armazenamento do arquivo;

7) **Vínculo de portas:** utilizar o chamado “port binding” para expor serviços;

8) **Concorrência:** escalar através do processo do modelo. Há duas formas de escalabilidade: a vertical, quando aumentamos a quantidade de hardware em uso pela aplicação, e a horizontal, quando criamos réplicas da aplicação a fim de garantir o processamento das requisições;

9) **Descartabilidade:** processos com rápida inicialização e encerramento gracioso;

10) **Paridade entre desenvolvimento e produção:** manter o desenvolvimento, a homologação e a produção o mais similares possível. Assim, temos a confiança de que o ambiente em que rodamos no desenvolvimento é similar à produção;

11) **Logs:** devem ser tratados como eventos. O Log é o registro dos eventos de uma aplicação. Colocado em um ambiente de centralização para consulta, maior é a sua eficiência;

12) **Processos administrativos:** execução de gestão e administração do software em processos pontuais.

17:29



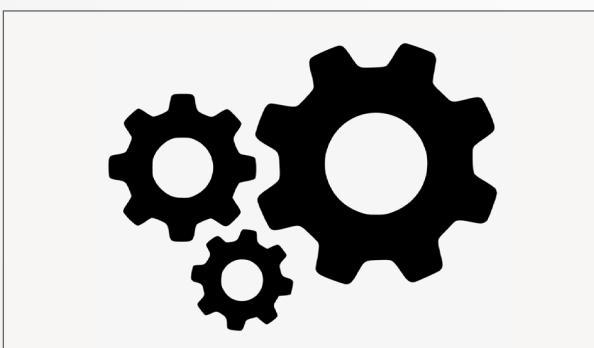
*O melhor modelo para você escalar em um ambiente de nuvem, um ambiente dinâmico, é a escalabilidade horizontal.*



19:21



*Quanto mais rápido eu consigo colocar uma aplicação no ar, mais escalável ela é.*



## AULA 1 • PARTE 3

### GitHub

Criado em 2008, o GitHub é um repositório Git amplamente utilizado no mercado. Trata-se de uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. É uma ótima opção para armazenar remotamente os seus códigos.

Logo no começo da sua exposição, o professor cria uma chave SSH no GitHub para que possa trabalhar com o seu repositório Git no notebook utilizado em aula. Em seguida, ao clonar o repositório, o professor cria um novo diretório no Visual Studio Code e adiciona o seu conteúdo ao Git. Assim, caso esteja trabalhando colaborativamente, todos os membros de sua equipe podem acessar os arquivos adicionados ao servidor. Também é possível criar outras branches, bem como mover os arquivos entre as branches criadas e a branch principal (“main”).

Ao final de sua exposição prática, o professor demonstra uma aplicação de conversão de temperatura (de Celsius para Fahrenheit e vice-versa) baseada em Node.js.



00:24



01:26

### PALAVRA-CHAVE

**SSH:** Trata-se de uma forma eficiente de credenciamento ou autenticação entre um determinado ambiente e o usuário que o utiliza. O SSH fornece um canal seguro sobre uma rede insegura em uma arquitetura cliente-servidor, e é muito usado no login remoto de utilizadores a sistemas de computadores.



02:15

**“** Se você tem a possibilidade de utilizar chaves para autenticação, faça uso, porque é muito importante e garante segurança para o seu ambiente. **”**



11:57

**“** Cada ambiente que está representado na minha aplicação será uma branch. Quando você estiver desenvolvendo uma nova funcionalidade, é uma boa prática criar uma nova branch para cada funcionalidade. **”**



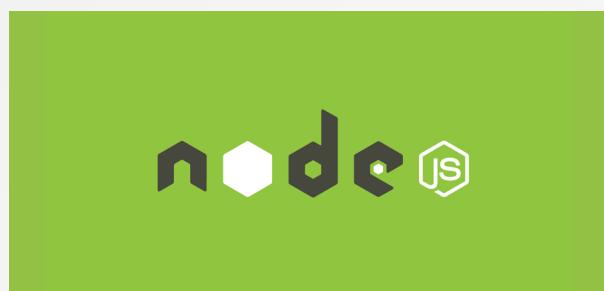
01:26

### PALAVRA-CHAVE

**Fork:** Uma bifurcação é uma cópia de um repositório. Bifurcar um repositório permite que você faça experiências à vontade sem comprometer o projeto original. Mais comumente, as bifurcações são usadas para propor mudanças no projeto de outra pessoa ao qual você não tem acesso de gravação, ou para usar o projeto de outra pessoa como ponto de partida para sua própria ideia.

## CURIOSIDADE

### Node.js



Criado por Ryan Dahl e lançado no ano de 2009, é um software multiplataforma de código aberto que permite a execução de códigos JavaScript fora de um navegador web. O principal diferencial do Node.js é a execução das requisições em single-thread, quando uma única thread é responsável por executar o código JavaScript.



19:21



### EXERCÍCIO DE FIXAÇÃO

Assinale a alternativa incorreta sobre o Twelve Factor Apps.

## AULA 1 • PARTE 4

### Containers



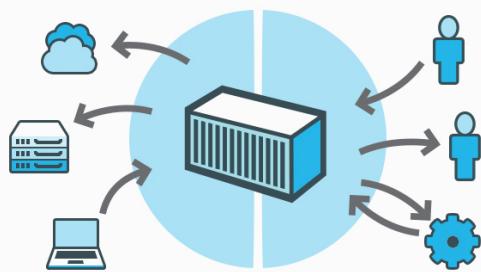
00:47

O desenvolvimento de softwares sem o uso de um container acarreta problemas de gestão de ambiente entre desenvolvedores e profissionais de infraestrutura. Com os containers, que possibilitam o isolamento dos processos (e, portanto, um container para cada um deles) e maior controle sobre esses processos, ganha-se portabilidade. Assim,

garanta-se que a aplicação desenvolvida será executada da mesma forma em qualquer ambiente - no seu notebook, em servidores ou na nuvem. É o que chamamos de "idempotência".

Vale destacar que os containers são diferentes das máquinas virtuais. No caso das virtual machines, há uma camada de infraestrutura (o hardware), uma camada de virtualização e, acima das duas, as máquinas virtuais (infraestruturas virtuais). Dentro de cada máquina virtual, está o sistema operacional (especialmente Windows e Linux).

Por outro lado, quando trabalhamos com containers, também estão presentes as camadas de infraestrutura e de sistema operacional (basicamente Linux). Acima das duas, porém, há o chamado "container runtime".



**“** Todos os processos da minha etapa de criação de ambiente e execução de aplicações serão feitos em containers. **”**

**“** Um container só se comunica com o outro caso eu queira. **”**

01:27



## CURIOSIDADE

### MongoDB



Desenvolvido pela MongoDB Inc. e lançado em 2009, é um software multiplataforma de banco de dados de código aberto. O MongoDB permite que as aplicações modelem informações de forma bastante natural, pois os dados podem ser aninhados em hierarquias complexas e seguem indexáveis e fáceis de buscar.

01:47



## CURIOSIDADE

### RabbitMQ



05:46



Lançado pela Rabbit Technologies em 2007, é um software de mensagens de código aberto. O RabbitMQ implementou os protocolos AMQP ("Advanced Message Queuing Protocol") e MQTT ("Message Queuing Telemetry Transport"), entre outros. Escrito em Erlang, seu código fonte é publicado sob a licença pública Mozilla.

07:46



## Docker

Embora não tenha criado a containerização, e embora existam várias opções no mercado, o Docker é a primeira coisa que vem à nossa cabeça quando pensamos nos containers. Há um conjunto de tecnologias já existentes no kernel do Linux, como: sea groups, relativo à capacidade que tenho de limitar recursos por processo; name space, ligado à capacidade que tenho de executar processos dentro de processos; e overlay file system, referente à tecnologia de file system.

O Docker pode ser usado em máquinas com Mac, Windows ou Linux, e pode ser baixado no [site da plataforma](#). No caso do Windows, ainda é preciso instalar o WSL 2.

Para criar um container no Docker, usa-se o comando docker container. Assim, podemos executar containers, explorar containers em execução, adicionar novos parâmetros, entre outros recursos. Nesse sentido, os containers podem servir de ambientes de estudo, uma vez que os testes executados pelo usuário não comprometerão a integridade do sistema.

“ Se você quer montar um ambiente Linux, quer fazer um teste de Shell, quer começar a estudar sem comprometer o seu ambiente, você pode começar com um container rodando em modo iterativo. ”

14:30

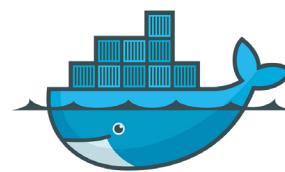


17:33



## CURIOSIDADE

### Docker



Lançado em março de 2013 e desenvolvido pela Docker, Inc., é um conjunto de produtos de plataforma como serviço (PaaS) que usam a virtualização na entrega de softwares em containers. O software que hospeda os containers é denominado Docker Engine.

19:18



20:35



32:52



43:45



58:10



“ Você tem o uso de containers antes do Docker e você tem o uso de containers depois do Docker. ”

“ A imagem é o elemento base para você criar um container. Todo container tem como origem uma imagem. A imagem é o projeto de execução do container. ”

“ Todo container é baseado em uma imagem. Então, eu preciso especificar a imagem que vou utilizar. ”

## PALAVRA-CHAVE

**Robo 3T:** Desde que foi lançado, o Robo 3T (anteriormente Robomongo) fornece uma ferramenta MongoDB de código aberto com funcionalidades inovadoras para atender às necessidades dos entusiastas e da comunidade em geral. O link para o site do Robo 3T está disponível [aqui](#).

## AULA 2 • PARTE 1

### Dockerfile I

Com o Dockerfile, podemos automatizar o processo de criação da imagem, garantindo que essa imagem será sempre criada da mesma forma, e podemos versionar a construção da imagem. O uso do Dockerfile, portanto, pode ser considerado uma boa prática.

Ao executar o build no container, o processo de instalação e de execução do curl foi muito mais veloz, tendo em vista que a construção da imagem é feita a partir da sobreposição de camadas, de modo que elas sejam adicionadas uma a uma à imagem. Nesse sentido, vale destacar que o Docker só trabalha com o cash de camadas até onde não houve alterações.

Uma vez criada a imagem, ela deve ser compartilhada em um repositório, em um Docker Registry, por meio do comando docker push. No caso do Docker Hub, você pode criar imagens públicas gratuitamente, ou repositórios de caráter privado.

“ O nome de uma imagem, a composição que descreve uma imagem, que nomeia uma imagem, é a junção do namespace com o repositório e a tag. ”



01:08



04:06



Com o Dockerfile, você consegue criar a sua imagem em qualquer lugar que crie imagens Docker, e crie imagens de container baseadas em Dockerfile.



06:29



Assim como o container, que é derivado de uma imagem, que é criado a partir de uma imagem, uma imagem sua vai ser criada a partir de outra imagem como base.



20:28



O apt update é utilizado para atualizar a lista de repositórios do Ubuntu.



22:49



Tem certos comandos para a criação de imagem que eu não posso deixar o cache ser resolvido, eu preciso sempre executar naquele momento.



29:18



36:50



A latest é sempre a última imagem. Então, toda vez que eu quiser executar a última versão de uma imagem, eu utilizo a latest.





## EXERCÍCIO DE FIXAÇÃO

Assinale a alternativa incorreta sobre o Docker e o Dockerfile.

## AULA 2 • PARTE 2

### Dockerfile II

O primeiro passo da criação da imagem no Dockerfile é determinar a imagem de origem, a imagem base. Se estivermos trabalhando com o Node, essa imagem deve logicamente conter o Node instalado. Em seguida, deve-se definir o diretório de trabalho, dentro do qual será construída a solução. Logo, copiamos o pacote JSON e executamos o npm install. Por fim, novamente copiamos toda o resto da solução para o diretório e determinamos a execução da aplicação.



00:24

08:17



**“**Se eu alterar o código [...] e executar, o npm install não vai ser executado, ele vai utilizar o cache. Por quê? Porque eu não estou alterando o pacote. **”**

09:21



**“**O uso de cache das imagens pode trazer benefícios se aplicado corretamente. **”**

**“** É uma boa prática você também colocar a tag aqui no seu Dockerfile, porque aí você garante a indempotência. **”**

10:30  
16:44

**“** Sempre que você quer criar uma imagem de container você vai utilizar o Dockerfile. **”**

## Opções de uso no Dockerfile



- 1) **FROM**: inicializa o build de uma imagem a partir de uma imagem base;
- 2) **RUN**: executa um comando;
- 3) **LABEL**: adiciona metadados à imagem;
- 4) **CMD**: define o comando e/ou os parâmetros-padrão;
- 5) **EXPOSE**: define que o container precisa expor a porta em questão;
- 6) **ARG**: define um argumento para ser usado no processo de construção;
- 7) **ENV**: define variáveis de ambiente;
- 8) **ADD**: copia arquivos, diretórios ou arquivos remotos e os adiciona ao sistema de arquivos da imagem;
- 9) **COPY**: copia arquivos ou diretórios e os adiciona ao sistema de arquivos da imagem;
- 10) **ENTRYPOINT**: auxilia na configuração de um container que pode ser executado como um executável;
- 11) **VOLUME**: define volumes que devem ser definidos;
- 12) **WORKDIR**: define o seu diretório corrente.

## AULA 2 • PARTE 3

### PALAVRA-CHAVE

**Commit:** Refere-se ao processo de tornar permanente um conjunto de alterações - ou seja, de efetivar as alterações. Um commit é o ato de enviar dados ou códigos para armazenamento em um banco de dados ou em um sistema de controle de versão.

**Toda vez que eu quero gerar uma nova release, uma nova versão da minha aplicação, eu preciso executar a integração contínua.**

### PALAVRA-CHAVE

**Deploy:** A implantação é a fase do ciclo de vida de um software que corresponde à passagem do software para a produção. A liberação, a instalação, a ativação e a desativação são as principais atividades ou etapas do deploy.

**Quando a gente fala em DevOps, a gente fala em agilidade. Quando a gente fala em agilidade, pensamos sempre em começar pequeno e ir escalando.**

00:45



### Pipeline

Uma pipeline nada mais é do que um conjunto de etapas realizadas sequencialmente (embora possam ser executadas paralelamente), de forma que um novo passo só ocorre quando o anterior for validado com sucesso.

A pipeline de integração contínua tem como objetivo integrar aquilo que está sendo desenvolvido em um produto já existente. Na integração contínua, há, em sequência, um processo de integração, o commit, o build (compilação da solução), o teste e a geração do pacote (executável).

A pipeline de deploy contínuo, por sua vez, parte da criação de uma release, a qual é logo submetida a um teste; depois do aceite, caso ele venha a ocorrer, o deploy em produção é iniciado.

03:20



03:47



07:54

O conjunto formado pela pipeline de integração contínua e pela pipeline de deploy contínuo corresponde à pipeline de CI e de CD. Automatizado, esse conjunto pode garantir escalabilidade ao processo de desenvolvimento, confiabilidade entre as equipes e mitigação de erros.

10:07

12:59



**A integração contínua, uma vez ligada ao processo de commit e à parte de software, está muito mais próxima do desenvolvedor.**



## CURIOSIDADE

### GitHub Actions



GitHub Actions

O GitHub Actions facilita a automatização de todos os seus fluxos de trabalho de software, agora com CI/CD de alto nível. Crie, teste e implante seu código diretamente do GitHub. Faça revisões de código, gerenciamento de filiais e triagem de problemas funcionarem da maneira que você deseja.

“ É importante que a pipeline seja construída exatamente assim: passo a passo. ”

## EXERCÍCIO DE FIXAÇÃO

Assinale a alternativa incorreta sobre pipeline.



15:31



17:54

### Criando uma pipeline I

Utilizando o GitHub Actions, no qual podemos automatizar, personalizar e executar nossos fluxos de trabalho de desenvolvimento de software diretamente no repositório, o professor demonstra o processo de compilação e de execução de uma solução. Iniciando com o setup do Node.js, Veronez ainda realiza a instalação da ferramenta de testes Mocha. É crucial que cada passo seja submetido a testes a fim de garantir o seu funcionamento.



25:56



## AULA 2 • PARTE 4

00:45



### Criando uma pipeline II

Na última parte de sua segunda aula, o professor demonstra a atribuição de uma segunda variável no GitHub para que a tag seja alterada quando o programa for novamente executado. Nesse caso, basta que o programador realize o commit para que a nova versão da solução já esteja disponível no GitHub, com os novos elementos ou reparos.

#### Perguntas

##### a) Qual é a importância de um desenvolvedor entender a infraestrutura?

Para que exista uma boa relação entre desenvolvedores e profissionais de infraestrutura, é preciso uma linguagem em comum. Nesse sentido, mesmo que seu trabalho seja restrito à codificação, é de suma importância que você saiba conceitos de infraestrutura.

b) Um desenvolvedor pode ser um profissional DevOps? Pode, desde que esteja ciente que deverá conhecer com mais profundidade a parte de infraestrutura.

c) Onde entra o Kubernetes no tema trabalhado nas aulas? Sendo necessária a escalabilidade, de forma que seja possível criar réplicas de um processo, é preciso de aplicações dedicadas a isso, tendo em vista que o Docker não é capaz de replicar um processo. Nesse sentido, o uso do Kubernetes é recomendável no gerenciamento de containers.

d) O DevOps está ligado diretamente ao uso da nuvem? Não, já que podemos ter DevOps em uma infraestrutura on premise. Na nuvem, porém, a escalabilidade é muito mais dinâmica.

e) O que é infraestrutura como código? A infraestrutura como código ajuda o profissional de operação e a equipe DevOps a trabalhar com a infraestrutura forma parecida com a de

03:37



12:29



15:31

#### CURIOSIDADE

##### Kubernetes



##### kubernetes

Também conhecido como K8s, é um sistema de código aberto para automatizar a implantação, o dimensionamento e o gerenciamento de aplicativos em containers. O Kubernetes agrupa os containers que compõem um aplicativo em unidades lógicas a fim de facilitar o gerenciamento e a descoberta.

um desenvolvedor – isto é, de forma declarativa.

f) **Eu tenho um projeto, eu tenho uma equipe, tenho as tarefas que vimos no curso... Basta uma pessoa da equipe saber fazer e os outros usam ou o ideal é que todo mundo tenha um entendimento melhor do que está acontecendo com DevOps?** Quando falamos em DevOps, falamos em colaboração. Quando falamos em colaboração, falamos em uma equipe multidisciplinar atuando para resolver um problema. Quanto mais centralizado é o trabalho, menor é a amplitude da equipe na sua capacidade de resolução.

g) **Aqui, estamos falando bastante em aplicar a pipeline no código de um servidor, mas eu poderia ter uma imagem que, ao invés de sair do Ubuntu, sai do Android?** Há uma série de ferramentas no mercado que permitem pegar o conceito de pipeline em aplicação web, o mais comum, e aplicar em dispositivos móveis ou dispositivos de IoT.

h) **O que é YML?** O arquivo .yml é uma notação declarativa adotada hoje no mercado, sendo amplamente usada em DevOps.

i) **O que é WSL 2?** Trata-se de um projeto criado pela Microsoft para possibilitar a execução do Linux dentro do Windows.



## AULA 3 • PARTE 1

“ Esclarece bastante o que é a integração contínua se eu a vejo configurada não importando, na primeira vez que a gente vê, qual é a linguagem de programação. ”

### PetClinic

Inicialmente escrita e adaptada para a plataforma Java, a aplicação PetClinic serve como demonstração das capacidades de um framework de desenvolvimento de software. Existem adaptações para as mais diversas arquiteturas e frameworks, servindo de divulgação de boas práticas.

Em uma exposição usando a variante PetClinic, que não utiliza o Java Spring, o professor argumenta que o Java Enterprise Edition possui poucas convenções, deixando que os usuários selecionem em meio a diferentes opções. Em seguida, apresenta outro repositório, o Spring PetClinic, com uma equipe maior e um número de commits mais elevado. A PetClinic Angular, por sua vez, é uma versão para quem deseja um front-end em Angular para o PetClinic REST.

“ Quanto mais cedo eu passar a usar scripts, mais cedo eu favoreço a comunicação de roteiros dentro da equipe e preparamos a equipe para entender os roteiros que são usados mais adiante em integração contínua, entrega contínua e assim por diante. ”

03:11

05:59

07:56

09:12

20:00

25:27

28:19

“ A parte mais difícil do DevOps é que eu posso ter um ambiente em que um único fornecedor ofereceu todas as ferramentas que eu uso, e aí a integração entre elas é muito mais simples, ou eu quero adotar gradualmente ferramentas que eu considero as melhores disponíveis. ”

“ Tentem acrescentar periodicamente novas ferramentas, novas práticas, e usem as convenções, as sugestões, dos próprios componentes da arquitetura que a gente tá usando na aplicação. ”

“ Olhando para um repositório, a gente já consegue perceber se ele está adotando as ferramentas que a gente está falando ou não, ou ferramentas equivalentes. ”

“ Eu quero, com esse esforço de DevOps, saber mais cedo o que está acontecendo, e aqui a gente tem uma automação visual. ”

“ As pessoas levam boas ideias de uma plataforma para outra plataforma, de um ecossistema para outro ecossistema. ”

### DevOps

Algumas das principais funções do DevOps giram em torno de definir, controlar e aprimorar os processos do ciclo de vida do software, além de construir, testar, empacotar e implantar software e sistemas de forma segura e confiável, e ampliar a comunicação e integração de desenvolvedores e demais envolvidos.

As preocupações são **construir**, de forma que o processo possa gerar um sistema executável e testável a partir de código-fonte; **empacotar**, combinando componentes relacionados em um único item implantável; e **implantar**, iniciando a etapa do ciclo de vida do software em que a operação acontece.

33:11



34:12



34:45



### CURIOSIDADE

#### Angular



Desenvolvida pela Google e lançado no ano de 2016, é uma plataforma de aplicações web de código aberto. O Angular utiliza HTML e TypeScript a fim de desenvolver a interface com o usuário - ou seja, o front-end de aplicações web. Em outras palavras, o Angular é um framework que simplifica a construção da interface de usuário.



### EXERCÍCIO DE FIXAÇÃO

Qual das opções abaixo refere-se a uma função do DevOps?

## AULA 3 • PARTE 2

### Desenvolvedores e Operações

Simplificando ao extremo, podemos dizer que os desenvolvedores estão preocupados com a velocidade e com a implementação de mudanças nos processos, ao passo em que o setor de Operações está mais preocupado com a estabilidade e com a segurança desses mesmos processos.

No caso do kernel do Linux, a verificação das linhas de código é realizada manualmente, uma vez que uma simples falha pode inviabilizar inúmeros servidores. Em outros casos, o número de implementações diárias é bastante elevado, sugerindo uma mudança no conceito de release.

**“**O desenvolvedor quer colocar uma velocidade maior, ele quer entregar coisas, fazer mudanças [...], e, pra operação, o sistema tem que estar disponível, tem que estar estável. **”**

**“**Quanto mais cedo eu tentar fazer a implantação, mais cedo eu descubro que, assim como está, já não dá para implantar. **”**

00:24

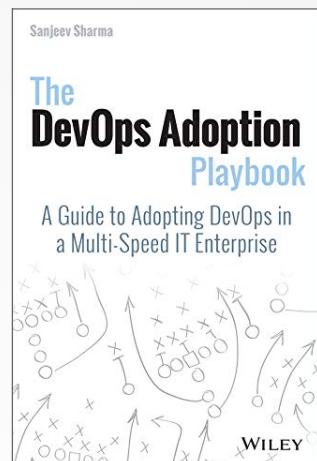


00:44



### LEITURA INDICADA

#### The DevOps Adoption Playbook



Publicado em 2017 por Sanjeev Sharma, o livro fornece orientação prática, açãoável e baseada no mundo real sobre a implementação de DevOps em escala corporativa. O autor mostra como você pode acelerar a produção e inovar até que a sua organização chegue ao topo.

02:03



03:54



06:10



09:09



**“**Leva tanto tempo pra fazer essa implantação que, quando a equipe de teste libera, o dev já uma ou duas novas versões que o teste tem que processar. **”**

### PERSONALIDADE

#### Linus Torvalds



Engenheiro de software finlandês, é o criador do sistema operacional Linux e do Git, sistema de controle de versão amplamente utilizado. Agraciado com diferentes prêmios, foi reconhecido por importantes veículos, como a revista Time, a InfoWorld e a BusinessWeek.

## PALAVRA-CHAVE

**PDCA:** Formado pelo conjunto de passos planejar, fazer, verificar e agir (plan, do, check e act, em inglês), um método iterativo de gestão utilizado no controle e na melhoria contínua de processos e produtos. Baseada na repetição, essa ferramenta é aplicada sucessivamente nos processos buscando melhorias de forma continuada.

### Mudanças com DevOps

Com o chamado “left-shift” ou “shift-left”, prioriza-se a participação dos envolvidos ao aplicar atividades de garantia de qualidade, segurança, privacidade, desempenho, verificação e validação mais cedo no ciclo de vida. Nesse sentido, ao pensar a segurança de forma adiantada, por exemplo, garantimos a sua disponibilidade em momentos mais tardios do processo.

Também há mudanças no sentido da automação, a qual é capaz de produzir registros de auditoria ao longo das etapas do ciclo de vida, o que inclui validação e validação do software. Além disso, mudanças culturais giram em torno de pequenas e simples implantações que cabem no cotidiano da equipe.

**“** O Extreme Programming é um dos métodos que dá mais ênfase na questão de engenharia. Uma equipe ágil com XP provavelmente está muito próxima, se já não obteve, de um modelo de DevOps operacional. **”**



03:20



11:44

## PALAVRA-CHAVE

**Lean:** Em português, “lean” quer dizer “enxuto”. Esta filosofia é baseada em modelos que aumentam o valor agregado reduzindo desperdícios. Sua história remonta ao final do século XIX, mas foi no período pós-guerra que encontrou grande destaque, quando o Japão passou a investir no Sistema Toyota de Produção.



13:37



21:00



*Eu vou ter dois perfis. Tem o usuário de inovação, que quer ver a última mudança, e um usuário mais conservador, que vai ficar pra uma próxima implantação.*



22:34

*Você não pode chegar e pedir assim: ‘eu quero integração contínua’. Você tem que fazer uma série de práticas para que, combinadas, auxiliem a fazer integração contínua.*



22:53



26:01

## PALAVRA-CHAVE

**Programação extrema:** Metodologia ágil, é usada em projetos em constante mudança. A programação extrema adota a estratégia de permanente acompanhamento e realização de ajustes durante o desenvolvimento do software. Algumas de suas características marcantes são: feedback constante, abordagem incremental e a comunicação entre as pessoas envolvidas.

## Competências

Para que o DevOps seja devidamente executado, são necessárias competências técnicas específicas, como o uso de ferramentas de gerência de configuração, a administração de sistemas operacionais, a adoção da integração contínua, a implantação de registros de auditoria, o aprimoramento do monitoramento de execução, a facilitação da configuração de senhas, a automatização de tarefas, a aprimoração da segurança da aplicação, entre outras.



26:18



31:19



*A automação faz com que a atividade seja executada sem os intervalos, ou sem a necessidade da atenção de uma pessoa, a qual pode gerar um erro quando a atenção se desvia.*



## AULA 3 • PARTE 3



00:24

### Configuração



00:59

*“Se chega uma peça nova naquela montagem, talvez eu tenha que ajustar outros parâmetros para compensar a mudança da peça que foi alterada.”*



05:08

*“Dependência interna é aquilo que fica dentro da aplicação, e dependência externa é o que fica executando junto, é um outro processo, e a gente tem que combinar tudo isso para que possamos ter uma configuração viável.”*

*“Existem ferramentas, existem bibliotecas que eu só uso quando estou desenvolvendo. Então, quando eu colocar a aplicação em produção, a biblioteca não vai acompanhar.”*



11:15

A gerência de configuração envolve gerência de versões e de mudanças, além da gerência de dependências internas (bibliotecas) e automação de construção e da gerência de pacotes. Nesse sentido, há uma parte relativa à aplicação e outra referente às bibliotecas necessárias para executar ou testar a aplicação em questão.

Pensando em uma aplicação web, teremos uma máquina de mesa e uma máquina que já executará um serviço, o qual não conta com interface de usuário, em que estarão teste e produção. O diferencial da gerência de configuração é identificar os itens de configuração que existem no ambiente do desenvolvedor que precisam ser movimentados para o ambiente de teste, e depois aqueles que devem ser movidos para o ambiente de produção.



## EXERCÍCIO DE FIXAÇÃO



Assinale a alternativa correta sobre o desenvolvedor e as Operações.

## AULA 3 • PARTE 4

00:58



### Guias II

O professor apresenta um conjunto de sites relacionados a dependências e a construção e a pacotes. Sua primeira demonstração é o Maven, que funciona através de um banco de dados, no qual pode ser buscada uma biblioteca. Vale notar que o Maven é capaz de empacotar a aplicação com bastante facilidade.

Há quem opte por outro gerenciador, o Gradle, em razão da menor quantidade de textos no que tange aos seus arquivos. Ainda em relação ao Maven, o professor apresenta o POM, cujo formato de arquivo é o núcleo da configuração de um projeto no Maven.

02:30



07:37



“ O gerenciador de pacotes, se ele instala a biblioteca uma vez, ele vai guardar também, numa pasta dele, escondida, igual ao que o Git faz, aquele pacote. ”

“ Se você vai começar um programa em Java Spring, é natural que você tenha uma certa configuração, que você tenha alguns arquivos... ”

**“** A adoção do Maven também tem que ser gradual. **”**

## SDKMAN

Instalador de ferramentas de desenvolvimento, o Software Development Kit Manager realiza o levantamento da configuração de software e hardware, filtra ferramentas compatíveis a partir de índice geral e transfere, instala e configura o JDK. Além disso, o instalador é capaz de adequar o produto ao equipamento que está disponível.

## Spring Boot

Além de contar com uma biblioteca para auxiliar os desenvolvedores no trabalho de programação, o Spring criou um conjunto de utilitários. Nesse sentido, ao executar o Maven, há fases específicas para quem trabalha com Spring. O professor ainda ressalta que, para colocar um aplicativo em execução, é preciso instalar uma série de coisas. Sendo assim, a equipe do Spring criou, dentro do framework de programação, uma maneira de fazer a máquina executar um servidor.

## PERSONALIDADE

### Kent Beck



Engenheiro de software americano, é o criador da metodologia de programação extrema (extreme programming) e do desenvolvimento guiado por testes (TDD). Beck sistematizou algumas das principais práticas em integração contínua, como a manutenção de um repositório unificado, o incentivo à contribuição diária, entre outras.

**“** 17:57

21:37

22:19

22:41

24:10

28:05

32:52

37:26

**“** Quando que vai aparecer o apt? Provavelmente quando você tiver que criar uma imagem, acrescentar alguma coisa ou então, na máquina de vocês, trazer um produto novo. **”**

## PALAVRA-CHAVE

**JDK:** O Java Development Kit é um conjunto de utilitários que permitem criar sistemas de software para a plataforma Java. Considera-se JDK qualquer implementação da Plataforma Java Standard Edition, Plataforma Java Enterprise Edition ou Plataforma Java Micro Edition destinada a desenvolvedores Java.

**“** A gente não precisa usar a ferramenta no nível mais baixo de abstração, a gente pode tentar usar num nível um pouco mais alto com um pouco mais de automação. **”**

## Integração contínua

Trata-se do processo recorrente de tentar combinar os diversos elementos da aplicação. As primeiras tarefas correspondem a mover a aplicação para fora do computador do desenvolvedor e combinar todas as contribuições. Isso ocorre porque, ao longo de seu trabalho, o desenvolvedor pode esquecer de realizar o add em algum arquivo, ou pode ser que ele utilize uma biblioteca que exista apenas na sua máquina.

Enquanto a integração contínua tem o objetivo de colocar o sistema em produção, a entrega contínua tem por objetivo verificar que o sistema está pronto para implantação.

## Sonar

O Sonar é um dos muitos recursos que podemos usar para inspecionar o código do programa, ao invés de executá-lo. Esse recurso oferece uma avaliação estática de um conjunto de regras de verificação e automatiza a revisão de código que seria realizada por um programador experiente.



**“** Qual é a vantagem de usar o action que alguém fez? É que, quando eu usar o action que alguém fez, quando essa pessoa evoluir o script, o meu script evolui também. **”**

## Contêineres

Contêineres como o Docker promovem o isolamento de serviços e o empacotamento da aplicação. Também há outros contêineres, como os servidores de aplicação ligados ao Java. Algumas linguagens têm os seus próprios contêineres. Nesse sentido, por vezes as aplicações Java necessitam utilizar outros serviços, o que pode nos levar a manusear o Docker de outras maneiras.

# Resumo da disciplina

Nesta página, veja um resumo dos principais conceitos trabalhados ao longo da disciplina.

## AULA 1

Junção do desenvolvedor com o profissional de operação, o DevOps atua com os interesses que existem em comum entre as duas funções, com foco no produto e na entrega.



Embora não tenha criado a containerização, e embora existam várias opções no mercado, o Docker é a primeira coisa que vem à nossa cabeça quando pensamos nos containers.



Com os containers, que possibilitam o isolamento dos processos e maior controle sobre esses processos, ganha-se portabilidade.

## AULA 2

Uma pipeline nada mais é do que um conjunto de etapas realizadas sequencialmente, de forma que um novo passo só ocorre quando o anterior for validado com sucesso.



O uso do Dockerfile pode ser considerado uma boa prática.



O conjunto formado pela pipeline de integração contínua e pela pipeline de deploy contínuo corresponde à pipeline de CI e de CD.

## AULA 3

Os desenvolvedores estão preocupados com a velocidade, ao passo em que o setor de Operações está mais preocupado com a estabilidade.



Para que o DevOps seja devidamente executado, são necessárias competências técnicas específicas.

A integração contínua é o processo recorrente de tentar combinar os diversos elementos da aplicação.



# Avaliação

Veja as instruções para realizar a avaliação da disciplina.

Já está disponível o teste online da disciplina. O prazo para realização é de **dois meses a partir da data de lançamento das aulas**.

Lembre-se que cada disciplina possui uma avaliação online.  
A nota mínima para aprovação é 6.

Fique tranquilo! Caso você perca o prazo do teste online, ficará aberto o teste de recuperação, que pode ser realizado até o final do seu curso. A única diferença é que a nota máxima atribuída na recuperação é 8.

