



APRENDIZAGEM EM FOCO

PRÁTICAS DA CULTURA DEVOPS NO DESENVOLVIMENTO DE SISTEMAS



APRESENTAÇÃO DA DISCIPLINA

Autoria: Stella Marys Dornelas Lamounier

Leitura crítica: Henrique Salustiano Silva

Esta disciplina tem como objetivo apresentar conceitos práticos e teóricos sobre a cultura DevOps no mercado e para maior comunicação entre membros de uma equipe, que pregam qualidade e rapidez no desenvolvimento de seus projetos. Serão especificados problemas antes enfrentados, com o isolamento de membros e falta de comunicação, que, a cada dia, está sendo sanado com o uso de DevOps aliados à utilização das metodologias ágeis, sendo um dos métodos mais utilizados em empresas de desenvolvimento modernas.

No primeiro momento, são apresentados conceitos teóricos sobre a cultura DevOps, como está sendo utilizada no mercado de software e como as empresas tem se beneficiado ao longo do tempo juntamente com a aplicação de métodos ágeis com este método, a fim de otimizar tarefas e aumentar a comunicação de todos os envolvidos no projeto e, por fim, conceituado o processo de implantação de DevOps em um empresa, utilizando ambientes corporativos e dinâmicos, aplicando na prática a integração/ entrega/ implantação contínua, para que haja uma forte integração entre equipes de Tecnologia da Informação e Desenvolvimento, trazendo velocidade, entrega rápida, confiabilidades e segurança no controle de gerenciamento e configuração.

Em seguida, a contextualização e funcionamento de ferramentas de versionamento, que podem ser utilizados por qualquer membro da equipe, a fim de haver forte gerenciamento de versões dos projetos e seu compartilhamento, tanto em modo off-line quanto remoto, sempre de forma colaborativa e ágil. É demonstrado como é realizado

este controle em ferramentas diferentes, também utilizando sistemas operacionais distintos, cada um com sua particularidade única. Neste contexto são apresentados os principais modelos de versionamento utilizados: o centralizado, que fornece versões para equipes menores, e o modelo distribuído, amplamente empregado em equipes grandes, que trabalham em lugares diferentes.

Por fim, de forma prática, são apresentadas as principais ferramentas que o mercado de desenvolvimento utiliza, como o modelo Git e o Docker, dando ênfase ao Git, por se tratar de uma ferramenta mais popular, são abordados seus principais comandos e operações de versionamento, como é feito o armazenamento no repositório de armazenamento e a comunicação entre os entes envolvidos no projeto.

A cultura DevOps já é uma realidade cada vez mais presente nas empresas, principalmente aquelas que são adeptas a metodologia ágil, portanto, compreender seu uso e seus principais comandos dará maior conhecimento que poderá ser aplicado ao longo de sua carreira profissional.

INTRODUÇÃO

Olá, aluno (a)! A *Aprendizagem em Foco* visa destacar, de maneira direta e assertiva, os principais conceitos inerentes à temática abordada na disciplina. Além disso, também pretende provocar reflexões que estimulem a aplicação da teoria na prática profissional. Vem conosco!



TEMA 1

Conceitos Essenciais sobre a Cultura DevOps aplicada ao Desenvolvimento de Sistemas

Autoria: Anderson Pereira de lima Jerônimo

Leitura crítica: Stella Marys Dornelas Lamounier



DIRETO AO PONTO

Com inspiração nos métodos ágeis, o movimento DevOps surgiu com o mesmo alinhamento. Permitido criar uma cultura colaborativa entre as equipes de desenvolvimento, conhecidas como Dev, e a de operações, conhecidas como Ops, ajudando as equipes a terem um trabalho mais coeso.

O DevOps, foca em ambientes dinâmicos, orientados na prática e na automatização dos processos e atividades, e, por meio desse pensamento, a equipe fica livre em apenas se preocupar com pontos mais importantes do software, minimizando esforços desnecessários durante a fase de desenvolvimento.

Figura 1 – Cultura DevOps



Fonte: sabelskaya/ iStock.com.

Esse tipo de cultura favorece a verdadeira transformação digital no ambiente de negócio, tornando a organização mais competitiva, trazendo consigo estímulos positivos para haja uma entrega de software cada vez mais rápida e contínua

Para garantir uma entrega de código com qualidade, é necessário seguir alguns pontos, como mostra a figura abaixo.

Figura 2 – Pontos importantes de impacto da cultura DevOps



Figura: elaborada pelo autor.

DevOps é uma mudança cultural. Não é uma questão de simplesmente adotar um planejamento ágil ou teste automatizado ou entrega contínua, embora essas práticas sejam certamente importantes. A cultura DevOps trata de um entendimento compartilhado entre desenvolvedores e operações e da

responsabilidade compartilhada pelo software que constroem. Isso significa aumentar a transparência, comunicação e colaboração em desenvolvimento, TI / operações e o negócio.

PARA SABER MAIS

DevOps é uma metodologia de ciclo de vida de desenvolvimento de software (SDLC) projetada para:

- Acelerar os ciclos de lançamento de produtos.
- Reduzir processos de desperdício.
- Melhorar a experiência do usuário final com cada iteração de lançamento.

Além das ferramentas e processos que permitem o DevOps, a metodologia SDLC popular é amplamente focada na cultura, nas pessoas e em sua mentalidade. Então, o que é essa cultura DevOps sobre a qual todos falam?

Em qualquer organização DevOps, seja uma única equipe ou toda a organização, equipes pequenas, multidisciplinares e autônomas trabalham de forma colaborativa entre os departamentos de desenvolvedor, QA e Ops com responsabilidades compartilhadas. As equipes devem se concentrar na qualidade do produto e na velocidade de entrega por meio de:

- Esforços colaborativos.
- Automação.
- Resposta ao feedback de todas as partes interessadas.

A própria organização também deve:

- Elimine silos entre departamentos.
- Permitir que as equipes operem de forma autônoma.
- Adotar medidas de governança e políticas que facilitem a autonomia e os processos SDLC orientados para automação.

DevOps promete um sistema de melhoria contínua, por meio de esforços colaborativos de várias equipes e indivíduos contribuintes em todo o pipeline de SDLC.

Com intuito de explorar mais sobre o contexto DevOps, seguem as principais vantagens:

- Se não tivermos uma cultura DevOps, uma cultura para apoiar a adoção do DevOps, todos os esforços de automação serão infrutíferos.
- Uma cultura DevOps é aquela em que as partes interessadas no desenvolvimento de software e no processo de entrega, incluindo o negócio, estão alinhadas em torno de objetivos compartilhados.
- Segurança psicológica é se sentir confortável com a falha, e estar em um ambiente onde é seguro repetir, falhar e melhorar continuamente. Entretanto, construir confiança pode ser um desafio em diversas equipes de DevOps.
- Para desenvolver uma cultura DevOps sustentável, trate sua cultura como um produto.
- As estratégias para acelerar e sustentar uma cultura DevOps são: priorizar essa iniciativa como uma meta organizacional

estratégica; desenvolver um roteiro; e identificar objetivos e resultados-chave.

TEORIA EM PRÁTICA

A empresa em que você trabalha possui diversos profissionais, cada um com sua responsabilidade, e sofre com problemas de atualizações de dados e documentos. Por exemplo, na criação de um documento de requisitos, muitas vezes, não há controle de versões do plano de requisitos, uma vez que podem sofrer alterações sempre que necessário e estas alterações pode chegar de forma desordenada aos desenvolvedores, assim criando sempre projetos que também precisam ao longo do tempo sofrer alterações. Existe ferramentas que podem auxiliar os profissionais dessa empresa a tratar e controlar tantas versões de documentos e códigos fontes, elas tornam o projeto como um todo mais organizado e de fácil acesso a qualquer momento e lugar.

Para conhecer a resolução comentada proposta pelo professor, acesse a videoaula deste *Teoria em Prática* no ambiente de aprendizagem.

LEITURA FUNDAMENTAL

Indicações de leitura

Prezado aluno, as indicações a seguir podem estar disponíveis em algum dos parceiros da nossa Biblioteca Virtual (faça o log in por meio do seu AVA), e outras podem estar disponíveis em sites acadêmicos (como o SciELO), repositórios de instituições

públicas, órgãos públicos, anais de eventos científicos ou periódicos científicos, todos acessíveis pela internet.

Isso não significa que o protagonismo da sua jornada de autodesenvolvimento deva mudar de foco. Reconhecemos que você é a autoridade máxima da sua própria vida e deve, portanto, assumir uma postura autônoma nos estudos e na construção da sua carreira profissional.

Por isso, nós o convidamos a explorar todas as possibilidades da nossa Biblioteca Virtual e além! Sucesso!

Indicação 1

Este livro trabalha fundamentos de gestão e gerenciamento de projetos, discutindo planejamento estratégico e estrutura organizacional, bem como padrões e metodologias de mercado nesse contexto. Com foco na metodologia ágil, apresenta uma análise de seu manifesto, mapeamento e melhoria de processos, e a importância da padronização. A obra, ainda, aborda o trabalho com *framework* Scrum e a necessidade da análise de métricas.

FERREIRA, B. M. **Métodos ágeis e melhoria de processos.** p. 25. Juiz de Fora: Universidade Federal de Juiz de Fora, 2020.

Indicação 2

A divisão do livro em quatro partes foi significativamente reformulada para acomodar novos capítulos sobre engenharia de resiliência, engenharia de sistemas e sistemas de sistemas. Todas essas mudanças foram realizadas visando à abordagem de questões essenciais para a engenharia de software moderna, como gerenciamento da

complexidade, integração da agilidade a outros métodos e garantia de segurança e resiliência dos sistemas.

SOMMERVILLE, I. **Engenharia de software**. 10. ed, p. 29 – 67. São Paulo: Pearson, 2019.

QUIZ

Prezado aluno, as questões do Quiz têm como propósito a verificação de leitura dos itens *Direto ao Ponto, Para Saber Mais, Teoria em Prática e Leitura Fundamental*, presentes neste Aprendizagem em Foco.

Para as avaliações virtuais e presenciais, as questões serão elaboradas a partir de todos os itens do Aprendizagem em Foco e dos slides usados para a gravação das videoaulas, além de questões de interpretação com embasamento no cabeçalho da questão.

1. O termo DevOps é a contração de duas palavras inglesas. Qual das opções abaixo representa a sigla, traduzindo para a Língua Portuguesa?
 - a. Devoluções e Operações.
 - b. Desenvolvimento e Opções.
 - c. Desenvolvimento e Operações.
 - d. Devaneios e Opções.
 - e. Devoluções e Operações.
2. Sobre a concepção do processo de desenvolvimento de software, seus modelos e, considerando toda evolução até os tempos atuais e com o fomento do DevOps, considere as afirmações abaixo:

I - O modelo tradicional para entrega de software favorece a percepção do cliente no que consiste a qualidade.

II - Processos manuais estão amplamente sujeitos a erros, ocasionam atrasos e não respondem coerentemente e de modo rápido às questões de negócio.

III - DevOps faz com que as equipes possam construir, testar, liberar e manter as aplicações com mais frequência e mais eficiência.

São corretas:

- a. Somente a afirmação I está correta.
- b. Somente as afirmações I e III estão corretas.
- c. Somente as afirmações I e II estão corretas.
- d. Somente as afirmações II e III estão corretas.
- e. Todas as afirmações estão corretas.



GABARITO

Questão 1 - Resposta C

Resolução: O significado do termo DevOps representa a união de forma abreviada das palavras *development* (Dev) e *operations* (Ops), que representam as equipes de Tecnologia da Informação.

Questão 2 - Resposta D

Resolução: O modelo tradicional, conhecido como modelo em cascata, não consegue acompanhar com tempo hábil as mudanças no decorrer do projeto, dificultando o fato de trabalhar com interações paralelas.



TEMA 2

Selecionando Conhecendo Ferramentas do DevOps no Desenvolvimento Web

Autoria: Anderson Pereira de lima Jerônimo

Leitura crítica: Stella Marys Dornelas Lamounier



DIRETO AO PONTO

Na metodologia DevOps, é necessário conhecer ferramentas, bibliotecas e *frameworks* que auxiliam no ambiente colaborativo entre as equipes de desenvolvimento e de operações. Vimos também que código limpo e atualizado, torna um estímulo a equipe para entregas contínuas. Uma ferramenta que podemos citar, que garante esse tipo de entrega, chama-se sistema de controle de versões. Essa ferramenta atua como repositório centralizado de código, que mantém histórico das mudanças feitas no projeto, oferecendo a possibilidade de um código mais atualizado e limpo, conforme a figura1.

Figura 1 - Código fonte único

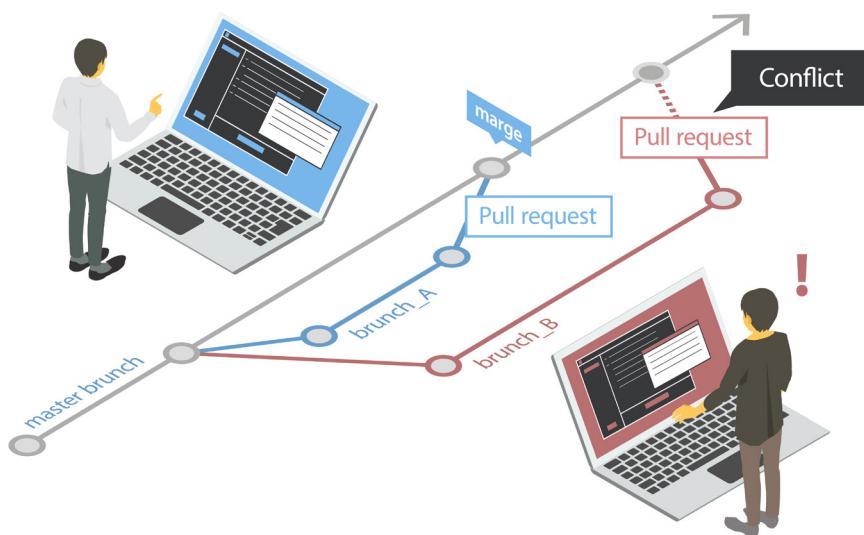


Fonte: Mohammed Haneefa Nizamudeen/ iStock.com.

Outra característica, caso tenham duas pessoas trabalhando no mesmo arquivo, é que o próprio sistema de controle de versão já consegue resolver os conflitos de forma automática, seja com mesclagem de código ou até manualmente, por meio de análise de algum membro da equipe de desenvolvimento. Uma ferramenta muito popular no mercado é o GIT, que trabalha um pouco diferente

de alguns sistemas de controle de versão, pois trabalha de forma distribuída, não havendo necessidade de rodar um servidor central para receber todos commits dos projetos. Como trabalha em um ambiente distribuído, cada membro de sua equipe utiliza um servidor local em seu próprio host. Sendo a ferramenta do GIT, dando a possibilidade subir para outros servidores, isso auxilia no compartilhamento do código.

Figura 2 – Tecnologia GIT



Fonte: interem/ iStock.com.

Ainda nesse contexto de ferramentas, também podemos falar sobre a tecnologia de containers, que trabalha com processo de virtualização de forma diferente. Essa tecnologia auxilia as equipes de desenvolvimento e de operações a terem sempre um ambiente de desenvolvimento mais idêntico possível quando há necessidade de compartilhar em diferentes sistemas operacionais entre suas equipes.

Semelhante ao virtualbox para as máquinas virtuais, no caso dos containers, temos o Docker, que, diferente das máquinas pelo fato de não utilizar Hypervisor, pode acessar os recursos da máquina local hospedeira diretamente. Esse compartilhamento de recursos ajuda

as imagens dos containers a terem apenas o essencial para ambiente de desenvolvimento, agilizando o processamento da máquina e possibilitando as equipes DevOps a terem um ambiente mais leve de forma rápida e compartilhada.

PARA SABER MAIS

Com intuito de aprofundar sobre a ferramenta que auxilia o desenvolvimento para equipes DevOps, destacaremos a ferramenta **Azure DevOps**, que se sobressai no mercado, oferecendo um conjunto de recursos com metodologias ágeis agregadas para ajudar no monitoramento e construção do projeto. Trata-se de um servidor que já obtém várias ferramentas, que auxilia em todo ciclo de vida do projeto, com controle de versão, gerenciamento *builds* e testes.

O Azure DevOps é uma plataforma de software, como serviço (SaaS) da Microsoft, que fornece uma cadeia de ferramentas DevOps de ponta a ponta para desenvolver e implantar software. Também se integra com a maioria das ferramentas líderes do mercado e é uma ótima opção para orquestrar uma cadeia de ferramentas DevOps. No DevOpsGroup, temos muitos clientes que descobriram que o Azure DevOps atende às suas necessidades, independentemente de seu idioma, plataforma ou nuvem.

O Azure DevOps comprehende uma variedade de serviços que abrange todo o ciclo de vida de desenvolvimento. Até o momento, são:

- Quadros Azure: planejamento ágil, rastreamento de itens de trabalho, visualização e ferramenta de relatórios.

- Azure Pipelines: uma plataforma de CI/ CD agnóstica de linguagem, plataforma e nuvem com suporte para contêineres ou Kubernetes.
- Azure Repos: fornece repositórios git privados hospedados em nuvem.
- Artefatos do Azure: fornece gerenciamento de pacote integrado com suporte para feeds de pacote Maven, npm, Python e NuGet de fontes públicas ou privadas.
- Planos de teste do Azure: fornece uma solução integrada de teste planejado e exploratório.

O Azure DevOps também pode ser usado para orquestrar ferramentas de terceiros.

TEORIA EM PRÁTICA

Você é desenvolvedor (a) em uma empresa de criação de sistemas computacionais, está trabalhando em um grande projeto para a criação de um software para uma universidade.

Você precisa corrigir alguns erros apresentados e também adicionar recursos solicitados ao longo do desenvolvimento pelo cliente.

Como você faria essas alterações citadas, utilizando o Git, mas sem comprometer a parte estável do código, isto é, sem que haja alteração nas linhas principais?

Para conhecer a resolução comentada proposta pelo professor, acesse a videoaula deste *Teoria em Prática* no ambiente de aprendizagem.



LEITURA FUNDAMENTAL

Indicações de leitura

Prezado aluno, as indicações a seguir podem estar disponíveis em algum dos parceiros da nossa Biblioteca Virtual (faça o log in por meio do seu AVA), e outras podem estar disponíveis em sites acadêmicos (como o SciELO), repositórios de instituições públicas, órgãos públicos, anais de eventos científicos ou periódicos científicos, todos acessíveis pela internet.

Isso não significa que o protagonismo da sua jornada de autodesenvolvimento deva mudar de foco. Reconhecemos que você é a autoridade máxima da sua própria vida e deve, portanto, assumir uma postura autônoma nos estudos e na construção da sua carreira profissional.

Por isso, nós o convidamos a explorar todas as possibilidades da nossa Biblioteca Virtual e além! Sucesso!

Indicação 1

Este livro trabalha nos fundamentos que auxilia as equipes de tecnologia da informação e suas organizações a adotarem a filosofia do DevOps em seus projetos, centrado no cliente, sendo no foco na produtividade.

FREEMAN, E. **DevOps Para Leigos.** São Paulo: Alta Books, 2021.

Indicação 2

O livro disponível na Biblioteca Virtual, traz de forma bem sistemática sobre o contexto das principais ferramentas DevOps, suas especificidades.

MONTEIRO, E. R. et al. DevOps. **Grupo A**. São Paulo: FC Nuvem, 2021.

QUIZ

Prezado aluno, as questões do Quiz têm como propósito a verificação de leitura dos itens *Direto ao Ponto, Para Saber Mais, Teoria em Prática e Leitura Fundamental*, presentes neste Aprendizagem em Foco.

Para as avaliações virtuais e presenciais, as questões serão elaboradas a partir de todos os itens do Aprendizagem em Foco e dos slides usados para a gravação das videoaulas, além de questões de interpretação com embasamento no cabeçalho da questão.

1. As ferramentas de controle de versão de software representam o instrumento que facilita no acompanhamento do projeto, em todo seu ciclo de vida até em seu *deploy*. Entre as tecnologias de controle de versão, aquela que trabalha de forma distribuída é:
 - a. TortoiseSVN.
 - b. Subversion.
 - c. Perforce.
 - d. GIT.
 - e. Todas as tecnologias.

2. Quando começamos a utilizar controle de versionamento, há alguns comando de operações que trabalham de forma semelhante, tanto para ambientes centralizados como para ambientes distribuídos. Nesse contexto, qual comando SVN update traz mudanças no repositório para máquina local e é equivalente no sistema do GIT?
- a. git pull.
 - b. git commit -m.
 - c. git diff.
 - d. git clone.
 - e. git logi.



GABARITO

Questão 1 - Resposta D

Resolução: O GIT se destaca das demais tecnologias, pelo fato de ser distribuído e não haver a necessidade de ter um ambiente centralizado, facilitando no compartilhamento e controle de versões, trazendo consigo um gerenciamento mais rápido.

Questão 2 - Resposta A

Resolução: O comando git pull baixa uma atualização de forma imediata do repositório remoto para repositório, local para que os conteúdos fiquem alinhados com mesmo código.



TEMA 3

Tecnologia de versionamento GIT e de integração continua

Autoria: Anderson Pereira de lima Jerônimo

Leitura crítica: Stella Marys Dornelas Lamounier



DIRETO AO PONTO

Trabalhar em equipe traz consigo alguns desafios, como a metodologia DevOps, que exige a integração das equipes que possam trabalhar de forma coesa e com recursos compartilhados. Esses conceitos geram desafios a serem enfrentados por todos os membros das equipes.

Figura 1 - Equipe DevOps



Fonte: ribkhan/ iStock.com.

Para equipe de desenvolvimento DevOps, o principal objetivo é trabalhar com estrutura de código fonte único do projeto, gerenciar diferentes versões geradas pelos membros da equipe.

Quando temos um projeto pequeno, o gerenciamento sem auxílio de algum software torna a tarefa bastante simples, mas na medida em que projeto vai crescendo, a equipe acaba enfrentando enormes

problemas para organizar e gerenciar a versão mais atualizada do código fonte do projeto.

Nesse cenário, entra o sistema de versionamento, em especial a tecnologia GIT, por ser um sistema com maior aceitação pela comunidade DevOps. Isso devido a sua simplicidade em seu gerenciamento e por trabalhar no modelo de versionamento distribuído.

No modelo distribuído que a tecnologia GIT faz uso, permite ter uma resposta rápida no gerenciamento, bem como no compartilhamento do código fonte para outros servidores. Embora o GIT utilize, inicialmente, repositório locais, por meio dele conseguimos gerenciar todos os arquivos, monitorando cada modificação feita no código fonte. Isso ajuda as equipes terem um código limpo e atualizando.

Para entendermos o funcionamento do GIT, o quadro abaixo apresenta os principais conceitos e finalidades da tecnologia GIT.

Quadro 1 - Termos da tecnologia GIT

Termos	Descritivo
Repositório.	É um diretório responsável que permite guardar os arquivos do projeto, podendo ser local e remoto.
Commit.	O commit indica um ponto no histórico do projeto, que indica se aconteceram modificações em um ou mais arquivos do projeto.
Diretório de trabalho.	Quando temos os arquivos do projeto de determinada versão do repositório, que ficam disponíveis para manipulação para os demais membros das equipes DevOps.

Fonte: elaborado pelo autor.

Git é uma ferramenta DevOps usada para gerenciamento de código-fonte. É um sistema de controle de versão gratuito e de código aberto, usado para lidar com projetos pequenos a muito grandes com eficiência. Git é usado para rastrear alterações no código-fonte, permitindo que vários desenvolvedores trabalhem juntos no desenvolvimento não linear. Linus Torvalds criou o Git em 2005, para o desenvolvimento do kernel Linux.

Antes de ver mais sobre o assunto, veremos um cenário antes do Git :

- Os desenvolvedores costumavam enviar seus códigos ao servidor central sem ter cópias de seus próprios.
- Quaisquer alterações feitas no código-fonte eram desconhecidas para os outros desenvolvedores.
- Não houve comunicação entre nenhum dos desenvolvedores.

Agora, examinaremos o cenário após o Git:

- Cada desenvolvedor tem uma cópia inteira do código em seus sistemas locais.
- Quaisquer alterações feitas no código-fonte podem ser rastreadas por outros.
- Há comunicação regular entre os desenvolvedores.

PARA SABER MAIS

Com a finalidade de aprofundar sobre a tecnologia de versionamento GIT, podemos citar a documentação do GIT, que demonstra de forma ampla os comandos a serem manipulados para gerenciar e monitorar os arquivos.

Git é um sistema de controle de versão usado para rastrear alterações em arquivos de computador. Geralmente, é usado para gerenciamento de código-fonte no desenvolvimento de software.

- Git é usado para rastrear alterações no código-fonte.
- A ferramenta de controle de versão distribuída é usada para gerenciamento de código-fonte.
- Permite que vários desenvolvedores trabalhem juntos.
- Suporta o desenvolvimento não linear, por meio de seus milhares de ramificações paralelas.

Recursos do Git:

- Histórico de trilhas.
- Livre e de código aberto.
- Suporta desenvolvimento não linear.
- Cria backups.
- Escalável.
- Suporta colaboração.
- Ramificar é mais fácil.
- Desenvolvimento distribuído.

O fluxo de trabalho Git é dividido em três estados:

- Diretório de trabalho: modifica os arquivos em seu diretório de trabalho.

- Área de teste (índice): testa os arquivos e adicione instantâneos deles à sua área de teste.
- Diretório Git (repositório): executa um commit que armazena os instantâneos permanentemente em seu diretório Git. Faz check-out de qualquer versão existente, faça mudanças, testa e submete.

Branch no Git é usado para manter suas mudanças até que estejam prontas. Você pode fazer seu trabalho em um branch, enquanto o branch principal (*master*) permanece estável. Depois de concluir seu trabalho, você pode mesclá-lo com o escritório principal.

O diagrama acima mostra que existe um *branch master*. Existem duas ramificações separadas, chamadas recurso pequeno e recurso grande. Assim que terminar de trabalhar com as duas ramificações separadas, você pode mesclá-las e criar uma ramificação mestre.

TEORIA EM PRÁTICA

Você trabalha em uma empresa de desenvolvimento, que possui filiais espalhadas em vários estados do Brasil e estão à frente de um projeto para uma empresa de alimentos congelados, que deseja realizar seu controle de faturamento. Você acabou de criar uma funcionalidade nova neste sistema, mas deseja que mais três membros da equipe editem o código e acrescentem outros aspectos relevantes para deixar ainda com mais qualidade. O problema é que cada um se encontra em uma localidade diferente. Como é possível sanar este problema, para que o arquivo seja compartilhado remotamente entre todos os envolvidos, a fim de que possam alterar em tempo real e observar

onde foram feitas as alterações para que haja maior controle de versões?

Para conhecer a resolução comentada proposta pelo professor, acesse a videoaula deste *Teoria em Prática* no ambiente de aprendizagem.

LEITURA FUNDAMENTAL

Indicações de leitura

Prezado aluno, as indicações a seguir podem estar disponíveis em algum dos parceiros da nossa Biblioteca Virtual (faça o log in por meio do seu AVA), e outras podem estar disponíveis em sites acadêmicos (como o SciELO), repositórios de instituições públicas, órgãos públicos, anais de eventos científicos ou periódicos científicos, todos acessíveis pela internet.

Isso não significa que o protagonismo da sua jornada de autodesenvolvimento deva mudar de foco. Reconhecemos que você é a autoridade máxima da sua própria vida e deve, portanto, assumir uma postura autônoma nos estudos e na construção da sua carreira profissional.

Por isso, nós o convidamos a explorar todas as possibilidades da nossa Biblioteca Virtual e além! Sucesso!

Indicação 1

Este livro trabalha nos fundamentos que auxilia as equipes de tecnologia da informação e suas organizações a adotarem a filosofia do DevOps em seus projetos, centrado no cliente, sendo no foco na produtividade.

FREEMAN, E. **DevOps Para Leigos**. São Paulo: Alta Books, 2021.

Indicação 2

O livro, disponível na Biblioteca Virtual, traz de forma bem sistemática sobre o contexto das principais ferramentas DevOps, suas especificidades.

MONTEIRO, E. R. *et al.* DevOps. **Grupo A**. São Paulo: FC Nuvem, 2021.

QUIZ

Prezado aluno, as questões do Quiz têm como propósito a verificação de leitura dos itens *Direto ao Ponto, Para Saber Mais, Teoria em Prática e Leitura Fundamental*, presentes neste Aprendizagem em Foco.

Para as avaliações virtuais e presenciais, as questões serão elaboradas a partir de todos os itens do Aprendizagem em Foco e dos slides usados para a gravação das videoaulas, além de questões de interpretação com embasamento no cabeçalho da questão.

1. A tecnologia open source também chamada de código aberto, são programas gratuitos e livres construídos e aprimorados por uma comunidade, que podem apresentar inúmeros benefícios para as empresas que adotam. Como exemplo de tecnologia open source para o processo de versionamento (sistemas de controle de versão de arquivos), temos os softwares, livres para o uso, tais como:

1	CVS
2	Git
3	SVN

Considerando as relações acima, podemos considerar corretas:

- a. Da relação apresentada existem apenas o 1 e 2.
 - b. Da relação apresentada existem apenas o 1 e 3.
 - c. Da relação apresentada existem apenas o 2 e 3.
 - d. Da relação apresentada existem apenas o 3.
 - e. Da relação apresentada existem 1, 2 e 3.
2. A tecnologia do GIT pode ser considerada uma das melhores ferramentas de versionamento disponíveis, no que diz respeito ao gerenciamento das informações. Isso porque toda a ferramenta é otimizada para entregar velocidade e agilidade durante o desenvolvimento.

Qual a vantagem de utilizar um sistema de versionamento descentralizado como o Git?

- a. Aumenta o tráfego de rede.
- b. Como os usuários não precisam estar conectados o tempo inteiro, o tráfego de rede diminui, gerando lentidão, conhecido como efeito espelho.
- c. Não há necessidade de se manter conectado o tempo inteiro, resultando em um ganho de performance.
- d. Não há vantagens, só desvantagens.
- e. Os usuários precisam estar conectados ao repositório constantemente.



GABARITO

Questão 1 - Resposta C

Resolução: O GIT e SVN são tecnologias de controle de versionamento utilizadas de forma gratuita pela comunidade.

Questão 2 - Resposta C

Resolução: Como a tecnologia Git trabalha com repositório local, só precisa compartilhar com outros servidores em momentos esporádicos, havendo, nesse sentido, uma otimização do tráfego.



TEMA 4

Adoção das Práticas DevOps em Equipes de Desenvolvimento

Autoria: Anderson Pereira de lima Jerônimo

Leitura crítica: Stella Marys Dornelas Lamounier

DIRETO AO PONTO

As práticas DevOps permitem, às equipes de desenvolvimento e de operações, a possibilidade de entregas mais rápidas e contínuas, o melhor gerenciamento e monitoramento do código fonte do sistema. Uma demanda crescente, nos sistemas atuais, é a exigência cada vez mais por funcionalidade e integrações por fontes externas, e isso acarreta aumento de sua complexidade.

Figura 1 - Gerenciamento do projeto

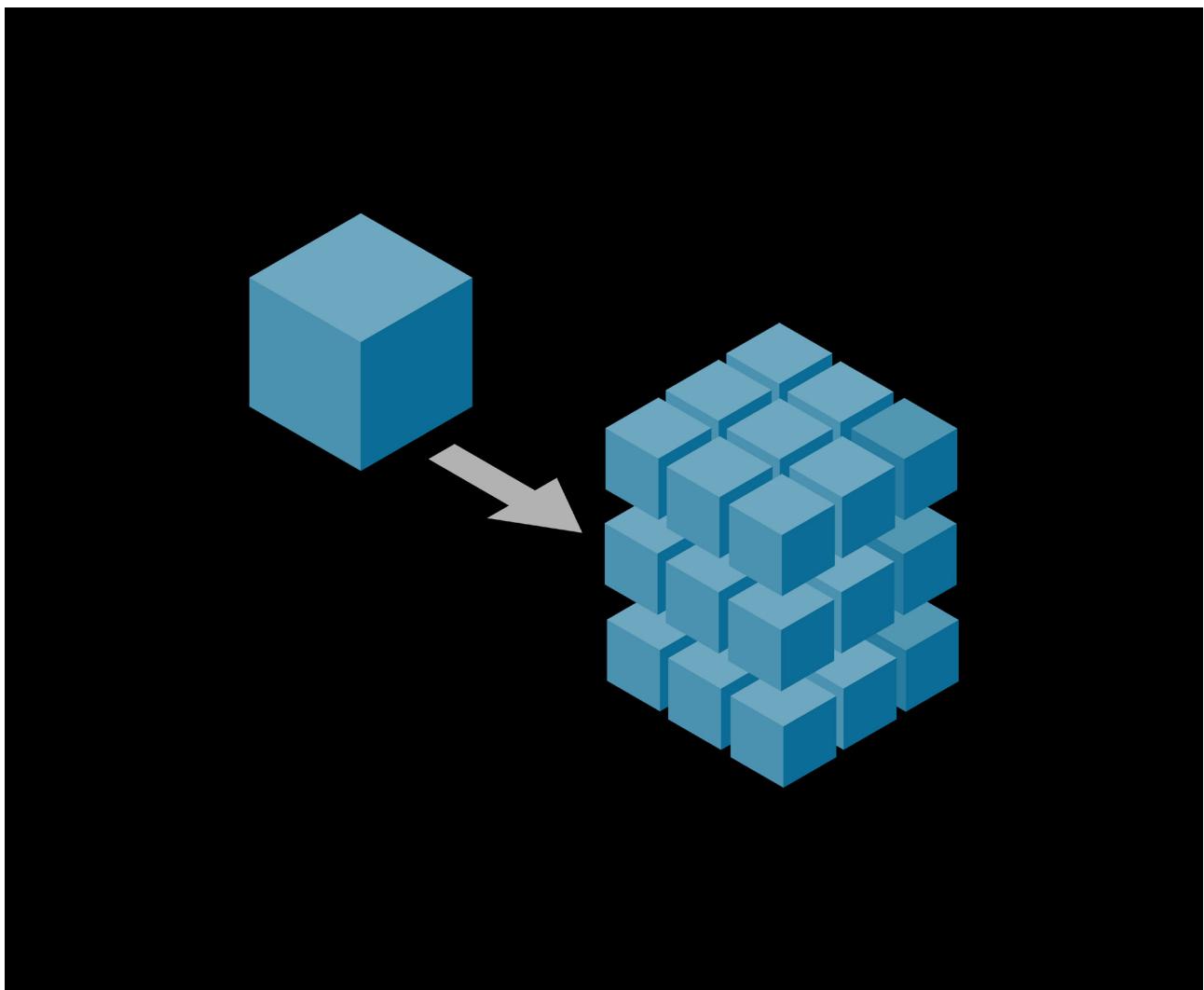


Fonte: VLADGRIN/ iStock.com.

Para que a produtividade de desenvolvimento não seja comprometida pela complexidade do sistema, as equipes estão

adotando, em seus projetos, o conceito de micros-serviços. Esse tipo de arquitetura de software possui como caracterizar, criar aplicações menores que detenham algum tipo de funcionalidade e/ou serviço que possa ser integrada por diferentes sistemas, o que chamamos de serviços fracamente acoplados.

Figura 2 – Monolítico e micros-serviços



Fonte: Piscine/ iStock.com.

Em sistema com tipo de arquitetura muito conhecida, chamado de monolítico, podemos notar que nem sempre os micros-serviços será uma boa adoção, há situações em sistemas não complexos, que o processo de gerenciamento dos micros-serviços podem comprometer a produtividade do projeto (MONTEIRO, et al. 2021). Outro fator de queda de produtividade do projeto é mais acentuada

quando o sistema aumenta seu nível de complexidade. Nesse caso, os micros-serviços se tornam uma boa opção a ser adotada, pois permitirão os sistemas terem uma complexidade maior, sem ter, necessariamente, uma perda na produtividade mais acentuada.

Uma tecnologia, usualmente utilizada entre as equipes DevOps, é a ferramenta de containers que auxiliam no ambiente baseado em micros-serviços. O conceito de containers bem usado na plataforma open source Docker, que garante um novo método de virtualização sem exigir um grande esforço da máquina hospedeira, por serem ambientes mais enxutos, trazendo apenas mínimas dependências necessárias para rodar a aplicação.

A plataforma Docker deixou de ser um programa e, agora, passa a ser um ecossistema, como podemos analisar no quadro a seguir.

Quadro 1 - Ecossistema da plataforma Docker

Docker Engine.	Sistema base de toda solução. É tanto o daemon responsável pelos containers, como o cliente usado para enviar comandos para o daemon.
Docker Compose.	Responsável pela definição e execução de múltiplos contêineres, com base em arquivo de definição.
Docker Machine.	Possibilita criar e manter ambientes docker em máquinas virtuais, ambientes de nuvem e até mesmo em máquina física.

Fonte: elaborado pelo autor.



PARA SABER MAIS

Com a finalidade de aprofundar sobre a adoção das práticas DevOps em equipes de desenvolvimento, podemos citar o dockerhub, onde encontramos images oficiais para criação de containers. Caso precise criar vários containers, existe um orquestrador de containers chamado docker compose. Essa ferramenta foi desenvolvida para ajudar a definir e compartilhar aplicativos de vários contêineres. Com ela, podemos criar um arquivo YAML para definir os serviços e, com um único comando, é possível girar tudo ou destruir tudo.

Docker Container pode ser um dos blocos de construção mais essenciais no campo de DevOps, atualmente. Atua como uma plataforma, que usa virtualização no nível do sistema operacional para entregar pacotes de software, conhecidos como contêineres. É uma arquitetura cliente-servidor. É uma ferramenta que tem ajudado a reduzir a distância entre a equipe de desenvolvimento e operações de um projeto. Essa ferramenta ajuda os desenvolvedores na criação, implementação e execução de aplicativos em contêineres, em várias plataformas. Neste material, exploraremos o Docker e o DockerHub e, junto com isso, também exploraremos seus benefícios e aplicativos.

Os contêineres (que atuam como uma instância da imagem do aplicativo) funcionam de maneira muito semelhante a qualquer máquina virtual. No entanto, em vez de criar um sistema operacional convidado virtual inteiro, o container usa o mesmo kernel Linux, no qual está sendo executado, para executar os aplicativos. Isso dá um aumento significativo de desempenho, em comparação com as máquinas virtuais, pois não é necessário fazer ajustes na máquina para executar um aplicativo específico e os próprios contêineres não colocam nenhuma carga no hipervisor, por serem leves. Ao usar a virtualização no nível do sistema operacional, o Docker

pode executar vários aplicativos leves em contêineres na máquina, isolados uns dos outros.

Para criar, implantar e executar contêineres Docker, basta instalar o daemon Docker. Este daemon Docker é um serviço executado no sistema operacional host, o que ajuda na criação, implantação e execução do contêiner Docker. Ajuda na criação e gerenciamento de objetos Docker como Container, Imagens, Redes e Volumes. Podemos interagir com o daemon Docker, usando a interface de linha de comando (CLI), enquanto o daemon se expõe à CLI usando a API REST. CLI, REST API e Docker daemon são conhecidos coletivamente como Docker Engine.

TEORIA EM PRÁTICA

Você trabalha em uma empresa de desenvolvimento e é responsável por utilizar Docker em uma máquina Linux, mas a empresa só trabalha com sistema proprietário. Que alternativa você usaria para sanar este problema, uma vez que você sabe os benefícios de utilizar sistemas Linux em empresas de desenvolvimento?

Para conhecer a resolução comentada proposta pelo professor, acesse a videoaula deste *Teoria em Prática* no ambiente de aprendizagem.

LEITURA FUNDAMENTAL

Indicações de leitura

Prezado aluno, as indicações a seguir podem estar disponíveis em algum dos parceiros da nossa Biblioteca Virtual (faça o log

in por meio do seu AVA), e outras podem estar disponíveis em sites acadêmicos (como o SciELO), repositórios de instituições públicas, órgãos públicos, anais de eventos científicos ou periódicos científicos, todos acessíveis pela internet.

Isso não significa que o protagonismo da sua jornada de autodesenvolvimento deva mudar de foco. Reconhecemos que você é a autoridade máxima da sua própria vida e deve, portanto, assumir uma postura autônoma nos estudos e na construção da sua carreira profissional.

Por isso, nós o convidamos a explorar todas as possibilidades da nossa Biblioteca Virtual e além! Sucesso!

Indicação 1

Este livro trabalha nos fundamentos que auxiliam as equipes de tecnologia da informação e suas organizações a adotarem a filosofia do DevOps em seus projetos, centrado no cliente, sendo no foco na produtividade.

FREEMAN, E. **DevOps Para Leigos**. São Paulo: Alta Books, 2021.

Indicação 2

O livro, disponível na Biblioteca Virtual, traz de forma bem sistemática sobre o contexto das principais ferramentas DevOps, suas especificidades.

MONTEIRO, E. R. *et al.* **DevOps. Grupo A.**

São Paulo: FC Nuvem, 2021.



QUIZ

Prezado aluno, as questões do Quiz têm como propósito a verificação de leitura dos itens *Direto ao Ponto, Para Saber Mais, Teoria em Prática e Leitura Fundamental*, presentes neste Aprendizagem em Foco.

Para as avaliações virtuais e presenciais, as questões serão elaboradas a partir de todos os itens do Aprendizagem em Foco e dos slides usados para a gravação das videoaulas, além de questões de interpretação com embasamento no cabeçalho da questão.

1. Podemos definir container como segregação de processos no mesmo kernel, de forma que o processo esteja isolado o máximo possível de todo o resto do ambiente. Quais são as principais características de um container? Marque verdadeiro ou falso nas opções abaixo:

- () É um conjunto de comandos para gerenciamento do ambiente.
- () Pode conter diversos ambientes ou tipos de SO.
- () São imagens de sistemas operacionais.
- () É a execução de um micro-serviços.
- () Tem como parte os comandos Docker Pull e Docker Run.

A sequência correta das assertivas é:

- a. V - F - V - V - V.
- b. F - V - F - V - F.
- c. F - F - F - V - V.
- d. V - V - F - V - F.
- e. V - F - F - V - F.

2. A plataforma Docker possui como característica um conjunto de produtos, como serviços que usam virtualização de nível de sistema operacional. Analise as assertivas a seguir sobre as principais definições de Docker:
- I. É um conjunto de componentes para gerenciar o seu ambiente de micros-serviços.
 - II. Tem como parte os comandos Docker Pull e Docker Run.
 - III. É um conjunto de comandos para gerenciamento do ambiente.
 - IV. São imagens de sistemas operacionais.
 - V. Pode ser executado apenas em servidores Linux.

São corretas:

- a. Somente a afirmação I está correta.
- b. Somente as afirmações I e II estão corretas.
- c. Somente as afirmações III e V estão corretas.
- d. Somente as afirmações I, II, IV estão corretas.
- e. Todas as afirmações estão corretas



GABARITO

Questão 1 - Resposta B

Resolução: A plataforma Docker possibilita a criação de diferentes imagens, cada uma com uma responsabilidade específica.

Questão 2 - Resposta B

Resolução: O Docker possibilita, por meio de comando simples, a criação e gerenciamento de imagens, baseado em micros-serviços, compartilhamento e gerenciamento de processos do host hospedeiro.



BONS ESTUDOS!