

Fundamentos DevOps

Prof. Rodrigo Macedo

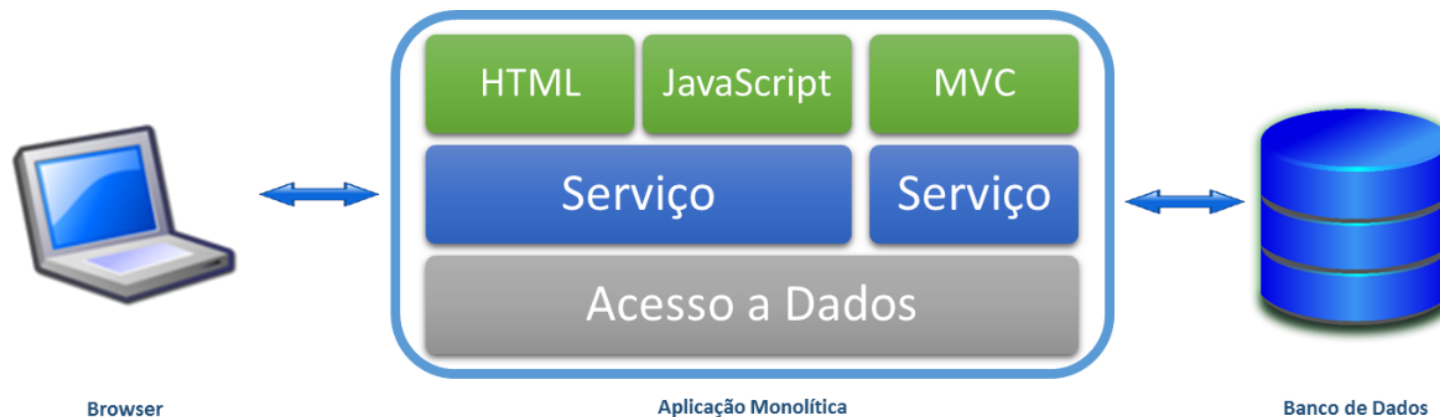
Escopo do Curso

- Microserviços
- DevOps
- Ciclo de Vida no DevOps
- Cultura DevOps
- Práticas DevOps
- Questões de concursos



Arquitetura Monolítica

- Uma Aplicação Monolítica descreve uma única aplicação de software em camadas no qual a **interface de usuário** e código de acesso aos **dados** são **combinados** em um único programa a partir de uma única plataforma.
- Um aplicativo monolítico descreve uma aplicação de software que é projetado sem modularidade.
- Uma aplicação monolítica é autônoma e independente de outras aplicações de computação.



Arquitetura Monolítica - Vantagens

- Pode suportar diferentes tipos de cliente como desktop/mobile.
- Pode exportar APIs para comunicação com terceiros.
- Pode integrar outras aplicações utilizando serviços REST/SOAP ou filas de mensagens.
- Pode tratar requisições HTTP, executar regras de negócio, acessar banco de dados e trocar informações com outros sistemas.
- Podem escalar verticalmente aumentando o poder das máquinas em que a aplicação roda ou horizontalmente com a adição instâncias atrás de um Load Balancer.

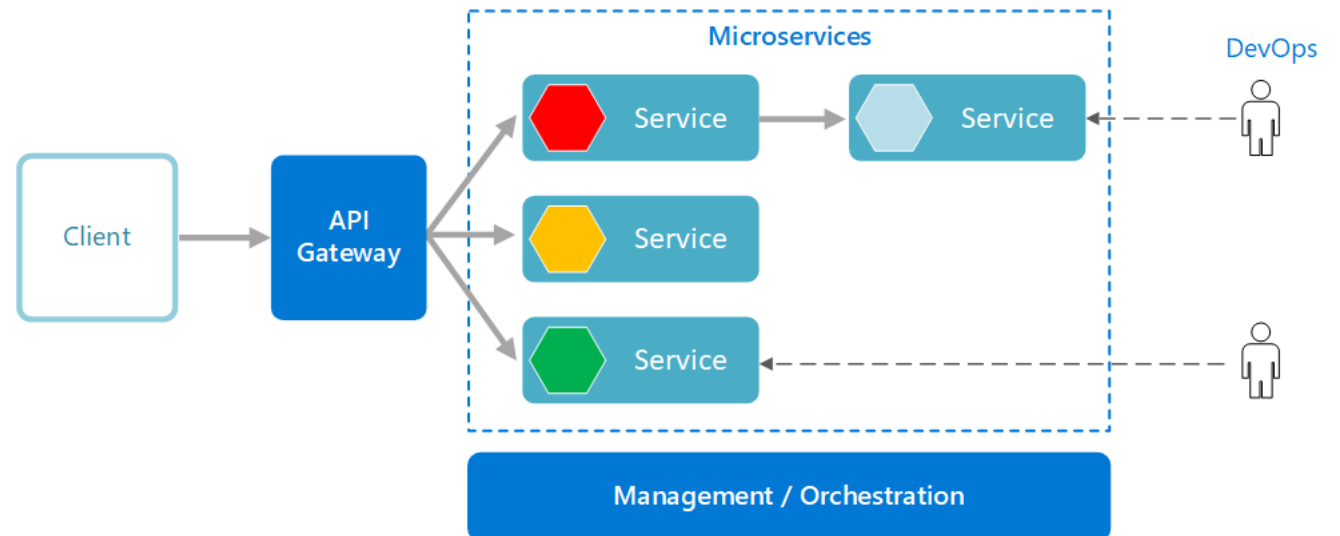
Arquitetura Monolítica - Desafios

- O sistema se torna tão complexo que a manutenção fica cada vez mais cara e lenta.
- Muitas funções são interdependentes e entrelaçadas, de forma que a inclusão ou manutenção de componentes do sistema pode causar inconsistências ou comportamentos inesperados.
- Exige que os desenvolvedores fiquem amarrados à tecnologia originalmente escolhida para o sistema.
- Qualquer mudança, por menor que seja, requer a reinicialização do sistema. Como isso pode causar algum risco operacional, é necessário que as equipes de desenvolvimento, testes e manutenção desses sistema acompanhem essas alterações.

Arquitetura Microserviços

- A arquitetura de micro serviços é utilizada para desenvolver uma aplicação como um **conjunto de pequenos serviços**, que funcionam com seu próprio processo. Cada serviço é desenvolvido em torno de um conjunto de **regras** de negócio **específicas**, e é implementado de forma **independente**.
- A ideia é **separar** os **serviços** para que cada um acesse uma camada do banco de dados ou somente um acesse algum serviço externo.

Arquitetura de microserviços é um modelo arquitetural de desenvolvimento no qual as aplicações são desmembradas em componentes mínimos e independentes



Arquitetura Microserviços - Vantagens

- A manutenção em um serviço específico não interfere diretamente nas outras funcionalidades do sistema.
- Os deploys e replicações de micro serviços são feitos por meio de infraestruturas de servidores, máquinas virtuais e containers que se organizam de forma independente.
- Cada aplicação só utiliza os serviços que necessita. Por isso, você não vai ter gastos extras carregando funcionalidades não utilizadas, afinal os custos estão diretamente associados à funcionalidade e à carga de uso do sistema.
- As mudanças no sistema são executadas por meio de alterações e evoluções feitas nos serviços. Portanto, o sistema como um todo não precisa ser reinicializado para continuar funcionando.

Arquitetura Microserviços - Desafios

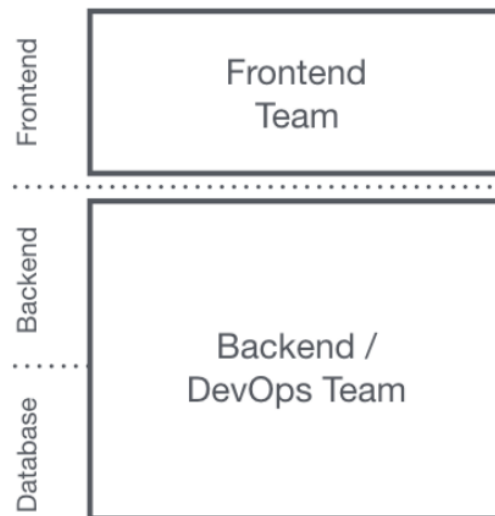
- Quando a arquitetura do sistema é feita, a divisão dos serviços tem que ser feita com muita atenção e cuidado: isso pode fazer com que se leve um pouco mais de tempo para chegar na divisão perfeita, de forma que no futuro a aplicação não sejam vários sistemas monolíticos separados e com funções que até se repetem;
- Complexidade no gerenciamento da aplicação: é um ponto a se tomar muito cuidado para que a organização sempre exista mesmo que novas features sejam implementadas no futuro.
- Há replicação de código de resposta ou de infraestrutura, por exemplo: o que existe de padrão em um serviço provavelmente existirá nos outros serviços também;

Monolítico x Microserviços

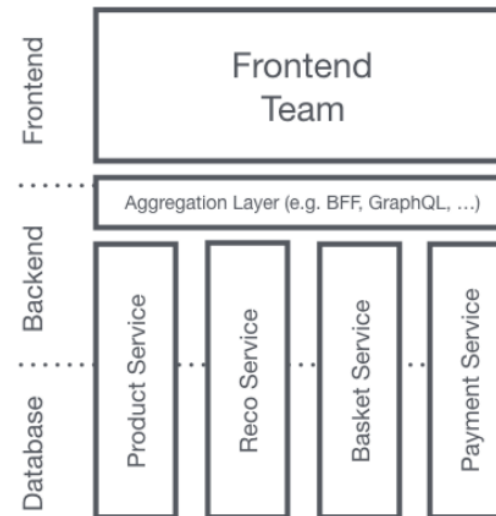
The Monolith



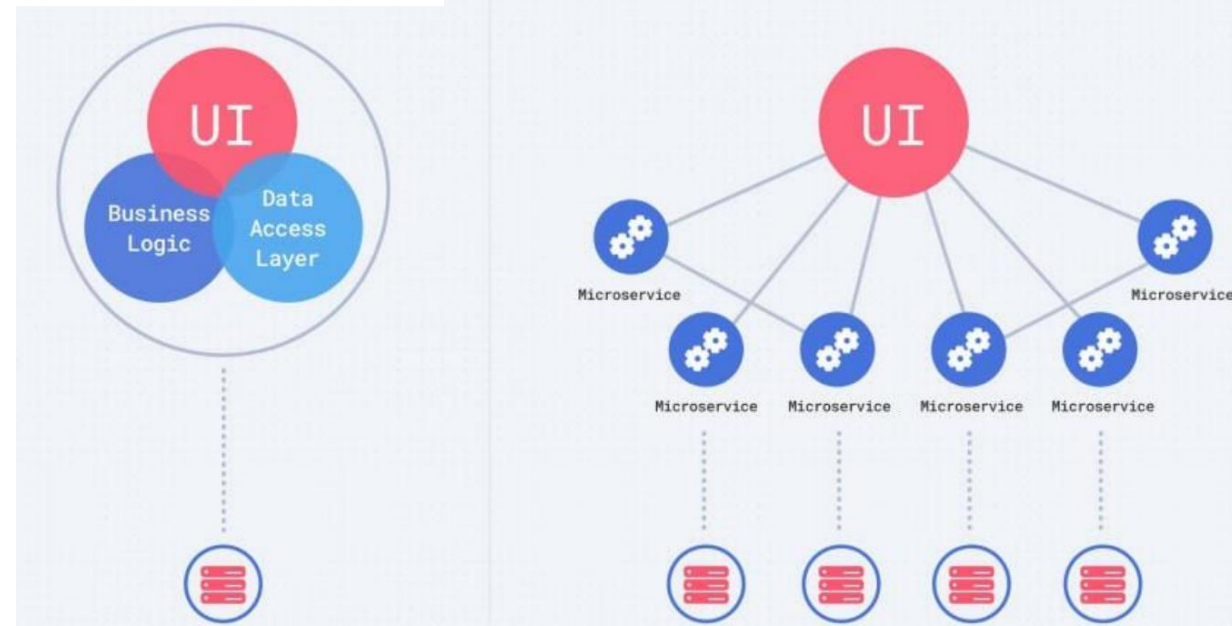
Front & Back



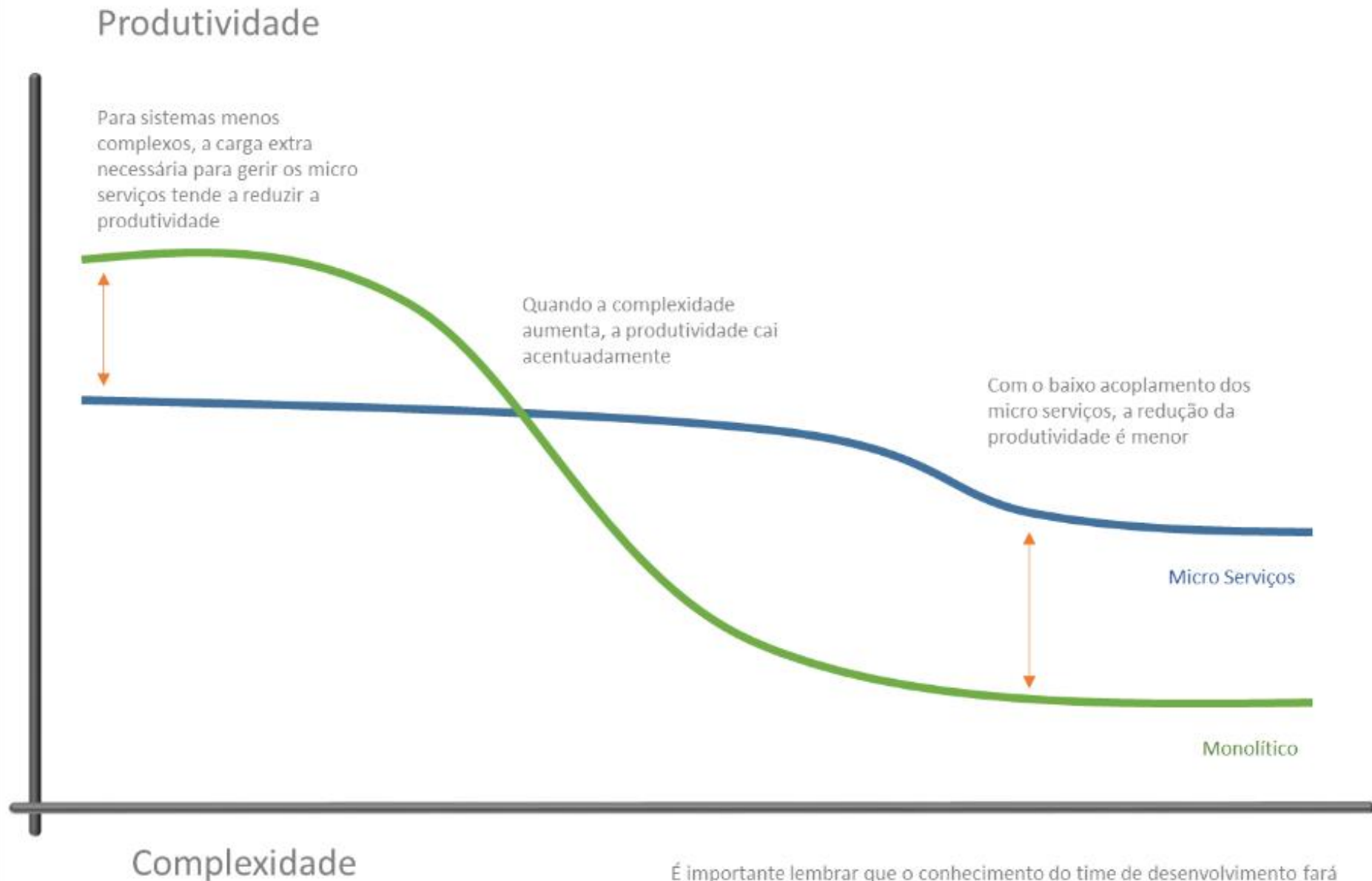
Microservices



Microservices Architecture



Monolítico x Microserviços



É importante lembrar que o conhecimento do time de desenvolvimento fará diferença em qualquer uma das opções escolhidas

Arquitetura Multi-Tenancy

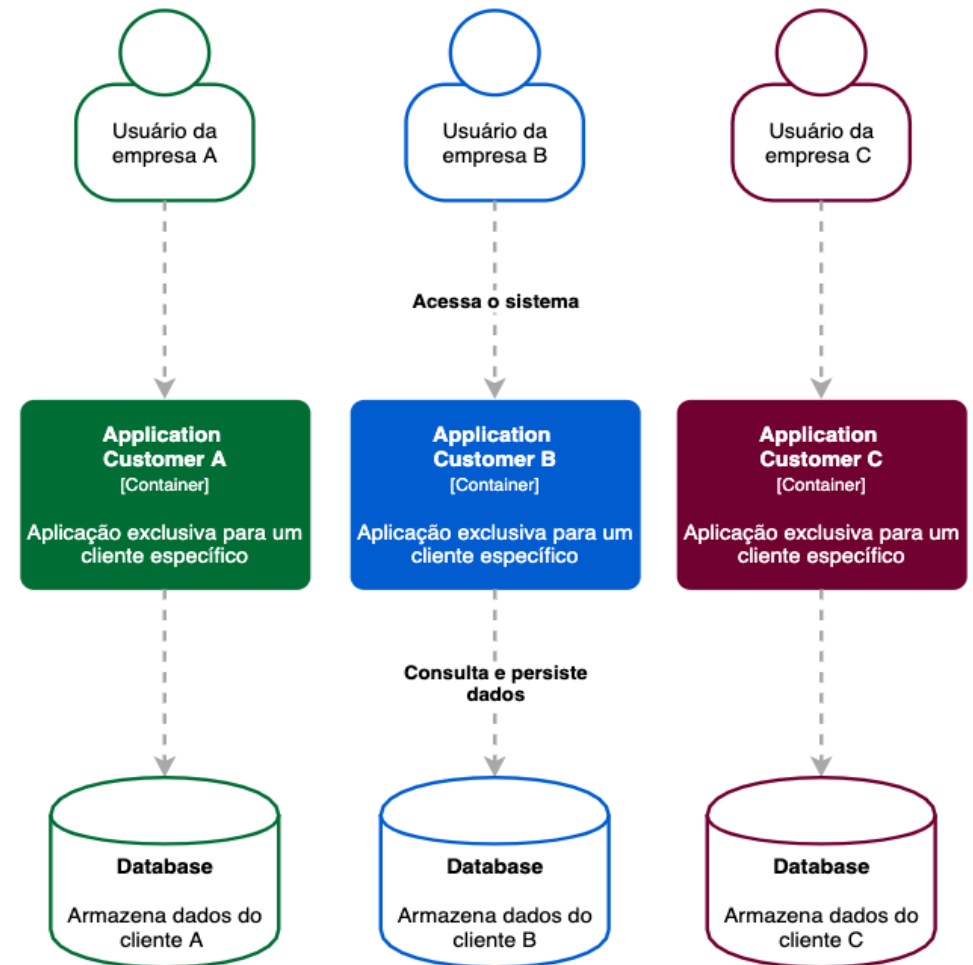
- Multi-Tenancy é um estilo de arquitetura onde você tem uma aplicação centralizada que atende a vários clientes.
- É um termo utilizado em plataformas SAAS — plataformas que oferecem Software Como Serviço, onde, na maioria das vezes os tenants são clientes corporativos. Ou seja, uma empresa utilizando serviços de outra empresa, numa relação B2B.



Arquitetura Single-Tenancy

- Esta estratégia preconiza por total isolamento dos dados e aplicações. Dessa forma, mitiga significativamente os riscos de vazamento de dados entre clientes diferentes na medida que bugs na aplicação não produzem esse tipo de vulnerabilidade.

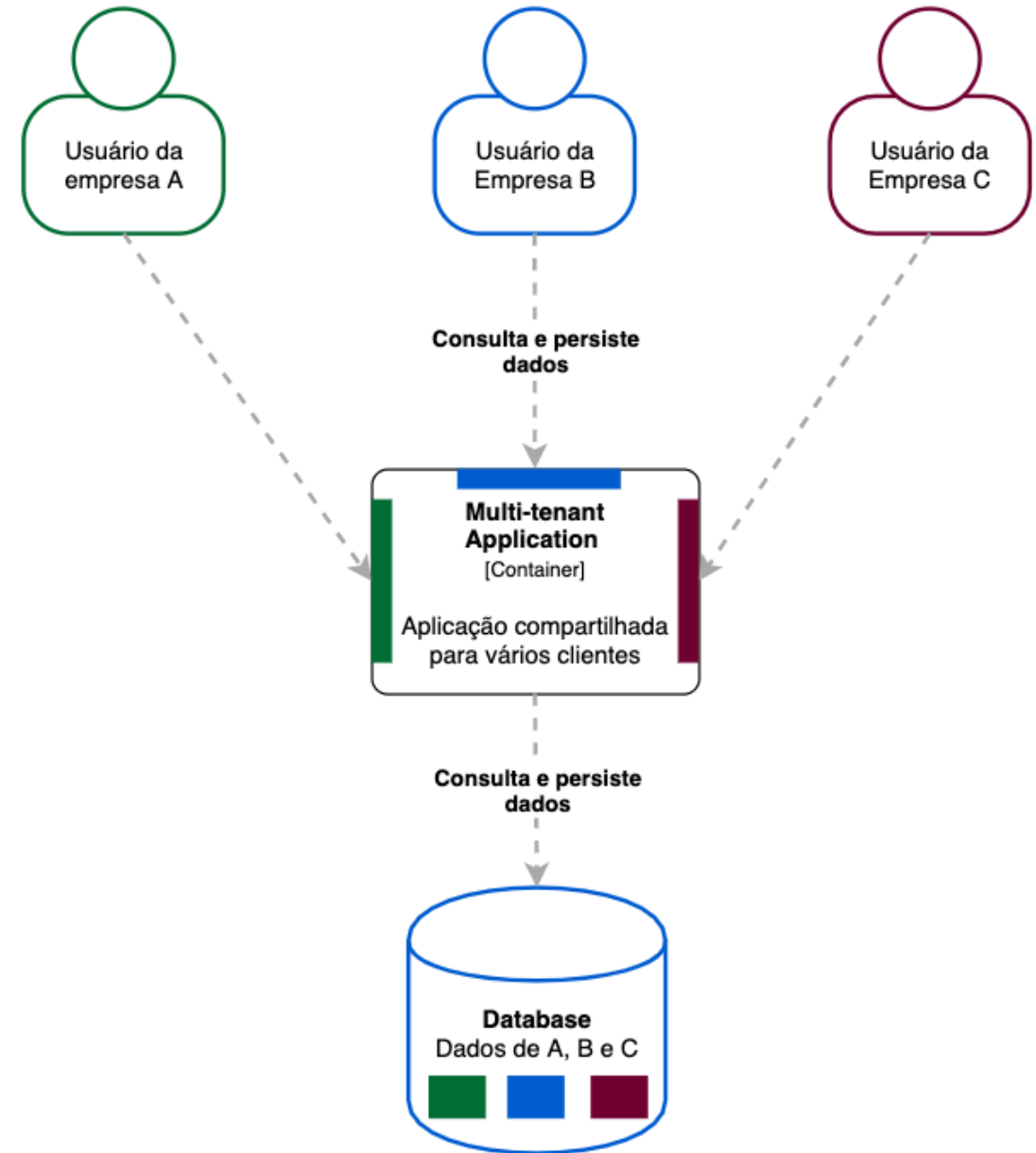
O principal problema dessa estratégia, porém, é que cada novo cliente “adquirido” exige que levantemos instâncias novas do servidor de aplicação e de banco de dados. Todo ganho que tivemos no desenvolvimento se converte em dificuldades para a operação, sobretudo para atividades de deploy.



Arquitetura Multi-Tenancy

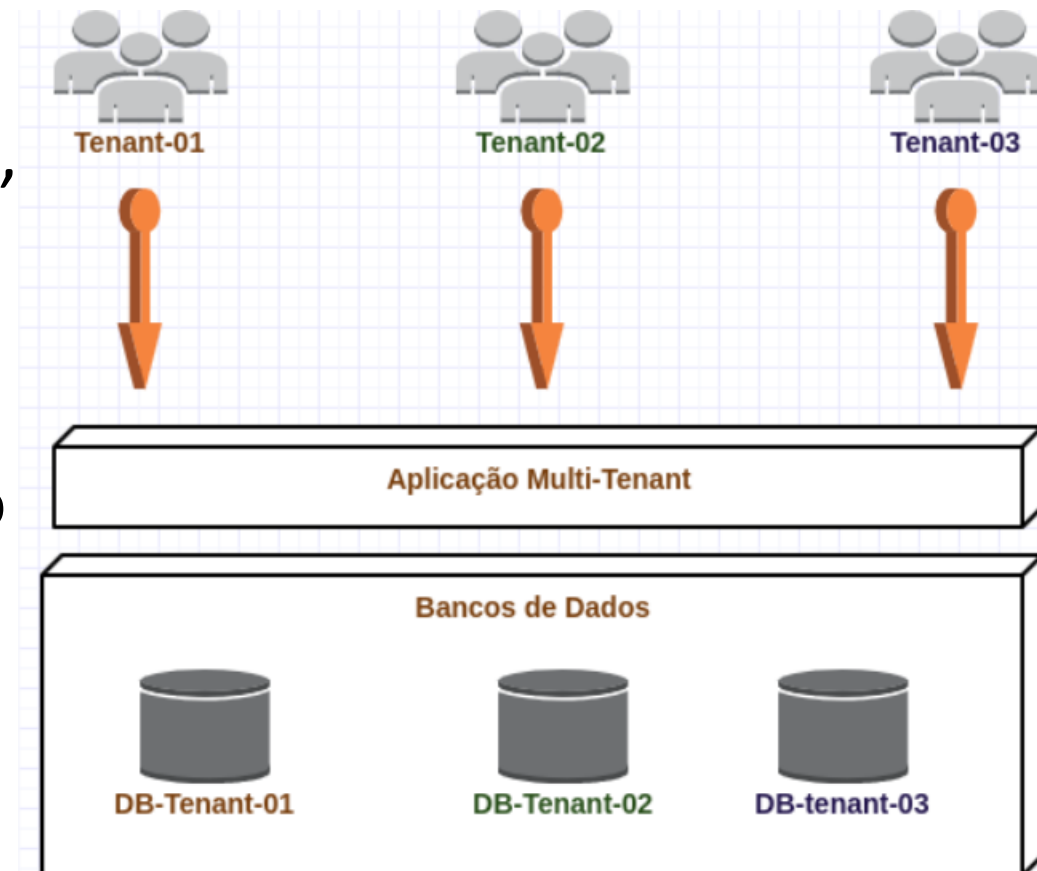
- Em uma estratégia multi-tenant todos clientes acessam uma mesma aplicação e ela é capaz de definir quais dados o usuário possui acesso. A ênfase, nessa estratégia, é otimizar a utilização de recursos computacionais.

Nesta estratégia a segregação é estritamente lógica. Ou seja, tanto o banco de dados quanto a aplicação são compartilhados. Por isso, é necessário cuidado redobrado é necessário extremo cuidado no código que trata as permissões dos usuários para vazar os dados de um cliente para outro.



Arquitetura Híbrida

- De forma resumida multi-tenancy é quando você tem um sistema que aceita varios usuário, onde cada usuário utiliza o sistema de forma isolada.
- Um exemplo é o wordpress.com onde cada usuário tem o seu blog independente e isolado dos outros blogs.



DevOps

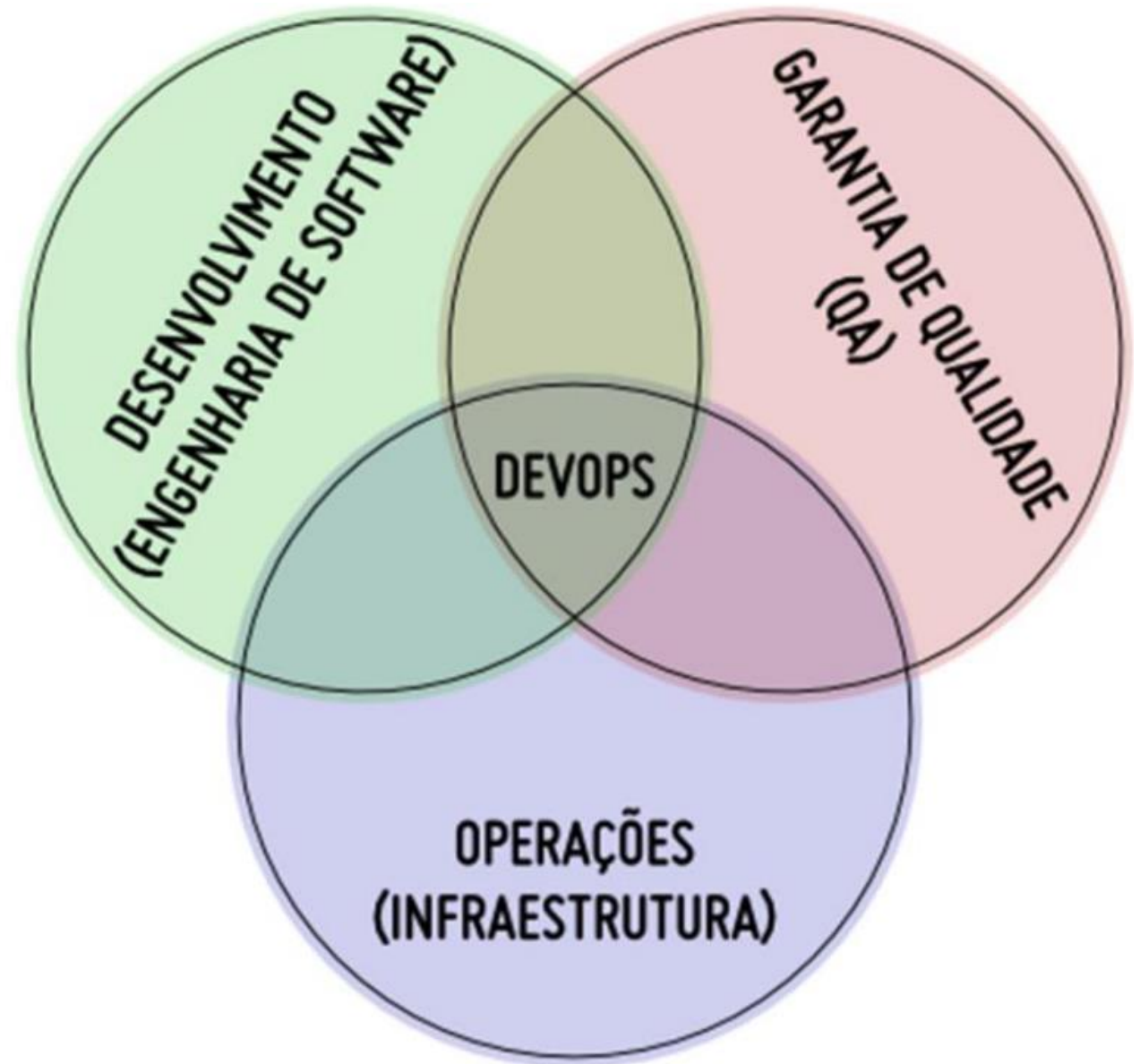
- DevOps é uma cultura na engenharia de software que **aproxima** os **desenvolvedores** de software (Dev) e os **operadores** do software com característica principal de melhorar a comunicação dos dois papéis dentro de um projeto e defender a automação e monitoramento em todas as fases da construção de um software.
- Auxilia empresas no gerenciamento de lançamento de novas versões, padronizando ambientes em ciclos de desenvolvimento menores, frequência de implantação aumentada, liberações mais seguras, em alinhamento próximo com os objetivos de negócio.



DevOps

"O principal objetivo é aumentar a colaboração entre os envolvidos no processo de entrega de software, de um modo que possamos entregar valor mais rápido e de modo mais confiável"

Patric Debois



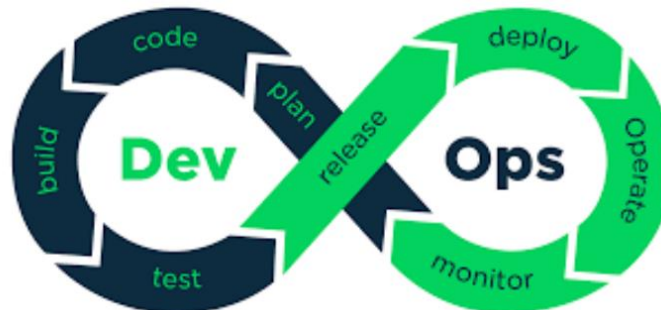
DevOps

- É um termo criado para descrever um conjunto de práticas para integração entre as equipes de desenvolvimento de softwares, operações (infraestrutura ou sysadmin) e de apoio envolvidas (como controle de qualidade) e a adoção de processos automatizados para produção rápida e segura de aplicações e serviços.
- O conceito propõe novos pensamentos sobre o trabalho para a valorização da diversidade de atividades e profissionais envolvidos e atitudes colaborativas. É um processo que torna possível o desenvolvimento ágil de aplicações em um modelo de gestão de infraestrutura definido sob regras rígidas e burocráticas.



DevOps

- A cultura DevOps auxilia empresas no gerenciamento de lançamento de novas versões, estimular a comunicação entre os dois papéis. Eventos podem ser acompanhados com maior facilidade, assim como o controle de processos documentados e emissão de relatórios granulares. Empresas com problemas no processo de liberação/implantação de novas versões, normalmente possuem automação, mas querem maior flexibilidade para gerenciar e conduzir esse processo - sem precisar editar tudo na linha de comando.
- Idealmente, essa automação deve ser disparada por recursos não operacionais, em ambientes específicos que não estejam "em produção". O desenvolvedor ganha maior controle sobre o ambiente, e o administrador do sistema maior entendimento sobre os aplicativos.



DevOps - História

- O movimento DevOps não começou em apenas um local, existem muitos lugares que se relacionam às origens do termo. Por volta de 2008 começaram a utilizar o termo **infraestrutura ágil** em algumas listas de discussão com foco em desenvolvimento ágil.
- O evento Agile 2008 abriu caminho para o DevOps, desviando as direções das metodologias de desenvolvimento de software modelo cascata e indo em direção a um ciclo contínuo de desenvolvimento.
- O termo DevOps foi criado durante a conferência Velocity da O'Reilly em 2009, onde John Allspaw (Etsy.com) e Paul Hammond (Typekit) o apresentaram, com o objetivo de unir desenvolvedores (Dev) e administradores da infra de TI (Ops), promovendo a integração contínua até a entrega.



DevOps - História

- Um dos participantes foi grande entusiasta do assunto. Ele é Patrick Debois, que após ter visto a palestra ficou muito animado, tendo a grande ideia de criar um encontro chamado DevOpsDay.
- Essa ideia saiu do papel e teve seu primeiro encontro em Ghent – Bélgica, no final de 2009 e durou 2 dias. Foi a partir daí que o assunto começou a ser conhecido pelo mundo.
- Desde então, ocorreram conferências "DevOps Days" na Índia, EUA, Brasil, Austrália, Alemanha e Suécia, etc.



DevOps - Benefícios

- **Integração entre os times de desenvolvimento e infraestrutura.** Esta integração também promove uma quebra de barreiras que pode existir com o negócio e os gestores do processo. Além disso, você conseguirá gerenciar seus ambientes de implantação, teste e produção de um modo mais eficiente.
- **Facilidade para gerência de ambientes de desenvolvimento.** É uma prática onde a infraestrutura é gerenciada utilizando técnicas de desenvolvimento de software (como, por exemplo, controle de versão e integração contínua). Assim, os desenvolvedores podem interagir com a infraestrutura utilizando ferramentas baseadas em código. Dessa forma, a infraestrutura e os servidores são implantados rapidamente.

DevOps - Benefícios

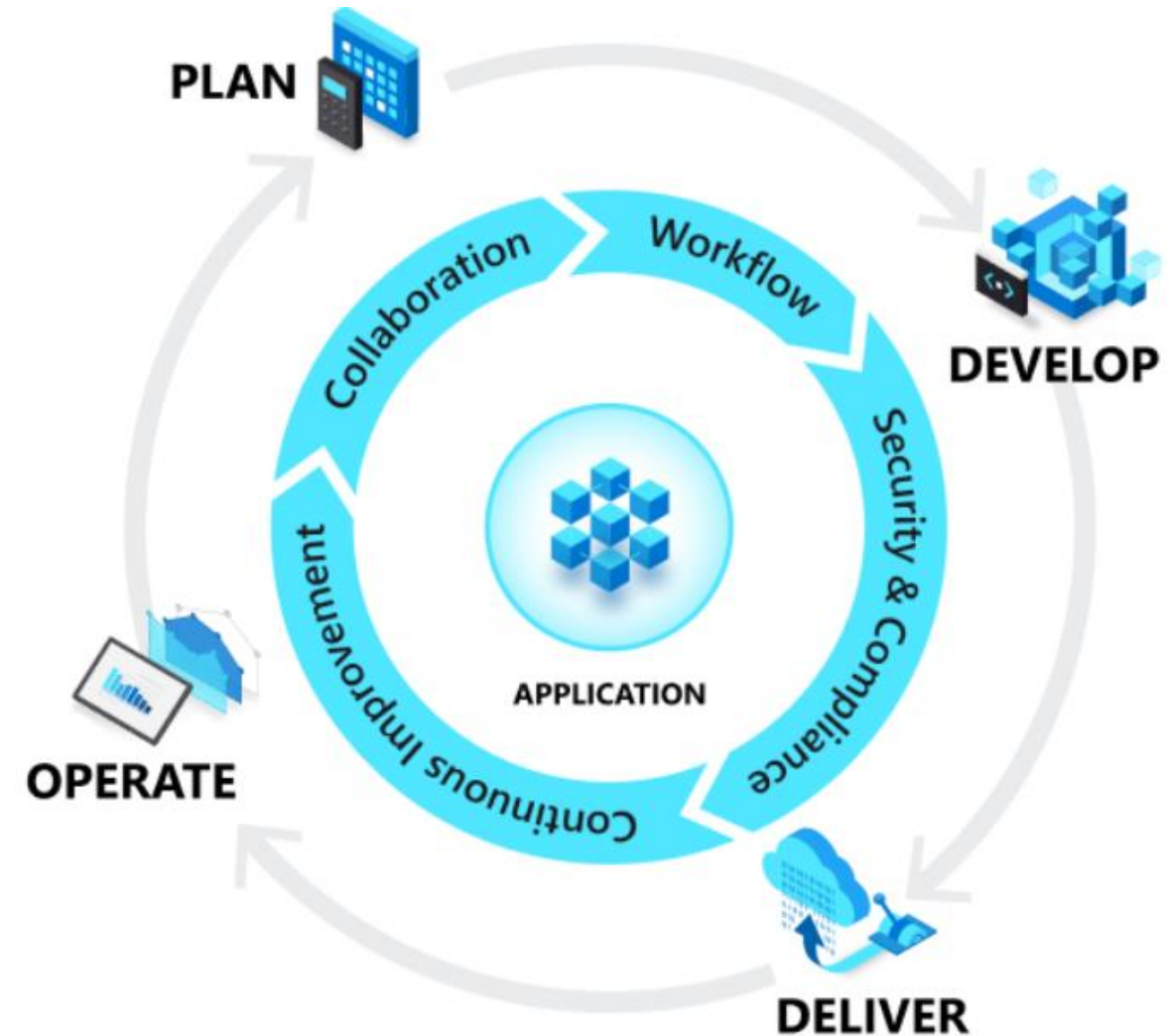
- **Maior aderência ao ambiente de execução das aplicações desde o ambiente de desenvolvimento.** Algumas ferramentas altamente voltadas às técnicas de DevOps, como o Docker, mitigam estes possíveis problemas com a padronização de ambientes de execução. Geralmente, as ferramentas e a cultura DevOps conseguem fazer com que os diferentes ambientes de execução de uma aplicação sejam idênticos, o que evita problemas de execução da aplicação em locais diferentes do local onde ocorre o desenvolvimento do software, por exemplo.
- **Pipeline de deploy mais rápido e automatizado.** Com o DevOps, você tem a oportunidade de otimizar e aperfeiçoar os processos de desenvolvimento e implantação. Com isso, ganhamos a possibilidade de ter um ritmo mais rápido no desenvolvimento de software e gerenciamento de infraestrutura. Com essas práticas, o processo será automatizado, utilizando ferramentas que ajudam a desenvolver de modo rápido e confiável

DevOps - Benefícios

- **Faster mean time to recovery (MTTR):** Com a utilização de DevOps o **tempo médio de reparo** é encurtado. Isso porque a metodologia agiliza sua própria manutenção, com os constantes monitoramentos e testes nas aplicações que estão sendo desenvolvidas. MTTR muito longo significam ineficiência, e consequentemente em custos. Para quem ainda pertence ao mundo da “velha TI”, dos grandes data centers, grande zeladora da estabilidade, o DevOps parece algo utópico. Mas, no mundo empresarial como vivemos, quem não se atualiza morre na praia. A nova TI preza por velocidade e utilização consciente de recursos.
- **Segurança:** Nos últimos anos, houve um grande aumento no número de ataques virtuais, sendo que os cibercriminosos estão cada vez mais refinados em suas técnicas, utilizando de toda e qualquer brecha para invadir aplicações e roubar dados. Ao aplicar uma cultura DevOps, pode-se rastrear e garantir a conformidade de código, mantendo um padrão de qualidade esperado dentro da segurança da informação.

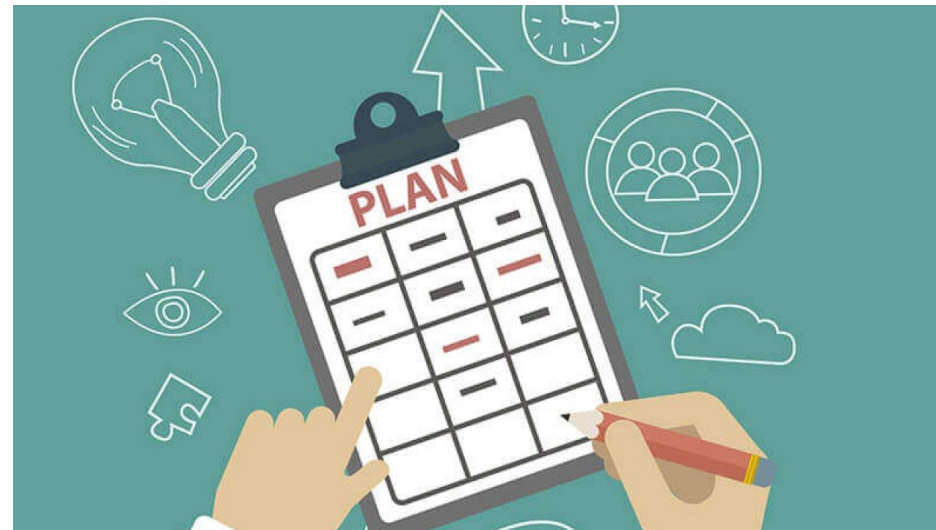
DevOps - Ciclo de Vida

- O DevOps influencia o ciclo de vida do aplicativo em todas as fases do planejamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de DevOps, cada função está envolvida de alguma forma em cada fase.



DevOps - Ciclo de Vida

- 1. Plano:** Na fase de planejamento, as equipes de DevOps idealizam, definem e descrevem os recursos e as funcionalidades dos aplicativos e sistemas que estão construindo. Elas acompanham o progresso em níveis altos e baixos de granularidade – desde tarefas de produto único até tarefas que abrangem portfólios de vários produtos. Criar lista de pendências, acompanhar bugs, gerenciar o desenvolvimento de software Agile com o Scrum, usar quadros Kanban e visualizar o progresso com dashboards são algumas das maneiras pelas quais as equipes de DevOps planejam com agilidade e visibilidade.



DevOps - Ciclo de Vida

2. Desenvolver: A fase de desenvolvimento inclui todos os aspectos da codificação – gravação, teste, revisão e integração do código pelos membros da equipe – bem como a compilação do código em artefatos de compilação, que podem ser implementados em vários ambientes. As equipes de DevOps buscam inovar rapidamente sem sacrificar a qualidade, a estabilidade e a produtividade. Para fazer isso, elas usam ferramentas altamente produtivas, automatizam etapas elementares e manuais e iteram em pequenos incrementos por meio de testes automatizados e integração contínua.



DevOps - Ciclo de Vida

3. Entrega: é o processo de implantação de aplicativos nos ambientes de produção de maneira consistente e confiável. A fase de entrega também inclui a implantação e a configuração da infraestrutura fundamental totalmente governada que compõe esses ambientes. Na fase de entrega, as equipes definem um processo de gerenciamento de versão com estágios claros de aprovação manual. Elas também definem portões automatizados que movem os aplicativos entre os estágios, até que sejam disponibilizados aos clientes. A automação desses processos os torna escalonáveis, repetíveis e controlados. Dessa forma, as equipes que praticam o DevOps com frequência podem atuar e entregar com facilidade, confiança e tranquilidade.



DevOps - Ciclo de Vida

4. Operação: A fase de operação envolve manter, monitorar e solucionar problemas de aplicativos em ambientes de produção. Ao adotar as práticas de DevOps, as equipes trabalham para garantir a confiabilidade do sistema, a alta disponibilidade e o objetivo de tempo de inatividade igual a zero, reforçando a segurança e a governança. As equipes de DevOps buscam identificar os problemas antes que eles afetem a experiência do cliente e mitigar os problemas rapidamente quando ocorrem. Manter esse nível de vigilância requer telemetria avançada, alertas acionáveis e visibilidade total sobre os aplicativos e o sistema subjacente.



DevOps - Cultura

- Embora a adoção de práticas de DevOps automatize e otimize processos por meio da tecnologia, tudo começa com a cultura dentro da organização – e com as pessoas que fazem parte dela. O desafio de cultivar uma cultura de DevOps exige mudanças profundas na maneira como as pessoas trabalham e colaboram. Mas quando se comprometem com a cultura de DevOps, as organizações podem criar o ambiente ideal para o desenvolvimento de equipes de alto desempenho.



DevOps - Cultura

1. **Colaboração, visibilidade e alinhamento:** Uma característica marcante de uma cultura íntegra de DevOps é a colaboração entre as equipes, que começa com a visibilidade. Equipes diferentes, como as de desenvolvimento e de operações de TI, precisam compartilhar os processos, as prioridades e as preocupações de DevOps. Essas equipes também devem planejar o trabalho em conjunto, bem como alinhar as metas e as medidas do sucesso conforme se relacionam aos negócios.



DevOps - Cultura

2. Mudanças de escopo e de responsabilidade: À medida que as equipes se alinham, elas assumem a propriedade e se envolvem em outras fases do ciclo de vida – não apenas naquelas que são essenciais para suas funções. Por exemplo, os desenvolvedores tornam-se responsáveis não apenas pela inovação e a qualidade estabelecidas na fase de desenvolvimento, mas também pelo desempenho e estabilidade que suas alterações trazem a software na fase de operação. Ao mesmo tempo, os operadores de TI certamente incluirão governança, segurança e conformidade na fase de planejamento e desenvolvimento.



DevOps - Cultura

3. Ciclos de Lançamentos mais Curtos: As equipes de DevOps se mantêm ágeis lançando versões de software em ciclos curtos. Ciclos de lançamento mais curtos facilitam o planejamento e o gerenciamento de riscos, pois o progresso é incremental, reduzindo também o impacto sobre a estabilidade do sistema. A redução do ciclo de lançamento também permite que as organizações se adaptem melhor e reajam de maneira mais ágil à evolução das necessidades dos clientes e à pressão da concorrência



DevOps - Cultura

4. Aprendizado Contínuo: As equipes de alto desempenho de DevOps estabelecem uma mentalidade de crescimento. Elas falham rapidamente e incorporam aprendizados aos seus processos, melhorando continuamente, aumentando a satisfação do cliente e acelerando a inovação e a capacidade de adaptação ao mercado. O DevOps é uma jornada, portanto, há sempre espaço para crescer.



DevOps - Práticas

- Além de estabelecer uma cultura de DevOps, as equipes dão vida ao DevOps implementando certas práticas ao longo do ciclo de vida do aplicativo. Algumas dessas práticas ajudam a acelerar, automatizar e melhorar uma fase específica. Outras abrangem várias fases, ajudando as equipes a criar processos contínuos que ajudam a melhorar a produtividade.

1. **Integração contínua:** A integração contínua é uma prática de desenvolvimento de software em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central. Depois disso, criações e testes são executados. Os principais objetivos da integração contínua são encontrar e investigar erros mais rapidamente, melhorar a qualidade do software e reduzir o tempo necessário para validar e lançar novas atualizações de software.



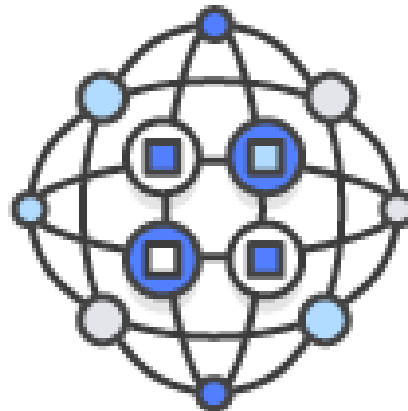
DevOps - Práticas

2. Entrega contínua: A entrega contínua é uma prática de desenvolvimento de software em que alterações de código são criadas, testadas e preparadas automaticamente para liberação para produção. Ela expande com base na integração contínua, pela implantação de todas as alterações de código em um ambiente de teste e/ou ambiente de produção, após o estágio de criação. Quando a integração contínua for implementada adequadamente, os desenvolvedores sempre terão um artefato de criação pronto para ser implantado, e que passou por um processo de teste padronizado.



DevOps - Práticas

3. Microserviços: A arquitetura de microserviços é uma abordagem de projeto para a criação de um aplicativo único como um conjunto de pequenos serviços. Cada serviço é executado em seu próprio processo e se comunica com outros serviços por meio de uma interface bem definida usando um mecanismo leve, geralmente uma interface de programação de aplicativo (API) baseada em HTTP. Os microserviços são criados em torno dos recursos empresariais, e cada serviço tem uma finalidade única. Você pode usar estruturas ou linguagens de programação diferentes para gravar microserviços e implantá-los independentemente como um único serviço ou um grupo de serviços.



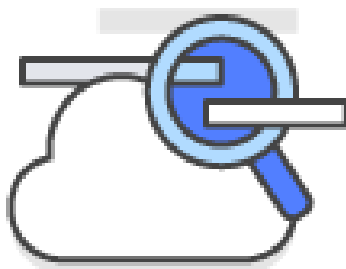
DevOps - Práticas

4. Infraestrutura como Código: A infraestrutura como código define os recursos e as topologias do sistema de uma maneira descritiva que permite às equipes gerenciar esses recursos da maneira como codificariam. Essas definições também podem ser armazenadas e versionadas em sistemas de controle de versão, onde podem ser revisadas e revertidas – novamente como código. A prática de infraestrutura como código ajuda as equipes a implantar os recursos do sistema de maneira confiável, repetível e controlada. A infraestrutura como código também ajuda a automatizar a implantação e reduz o risco de erro humano, especialmente para ambientes grandes e complexos. Essa solução confiável e repetível para a implantação de ambientes permite que as equipes mantenham ambientes de desenvolvimento e teste idênticos ao ambiente de produção. Da mesma maneira, a duplicação de ambientes para diferentes data centers e plataformas em nuvem também se torna mais simples e eficiente.



DevOps - Práticas

5. Monitoramento e Registro em Log: As empresas monitoram métricas e logs para ver como a performance do aplicativo e da infraestrutura afeta a experiência do usuário final do seu produto. Ao capturar, categorizar e analisar dados e logs gerados pelos aplicativos e pela infraestrutura, as empresas compreendem como as alterações ou atualizações afetam os usuários, o que proporciona um esclarecimento sobre as causas raiz dos problemas ou das alterações inesperadas. Como os serviços devem estar disponíveis 24/7 e a frequência de atualização do aplicativo e da infraestrutura aumenta, o monitoramento ativo torna-se cada vez mais importante. A criação de alertas ou a execução de análise em tempo real desses dados também ajuda as empresas a monitorar de modo mais proativo seus serviços.



Q1) [CESGRANRIO Caixa 2021] No âmbito de DevOps, o termo “shift left testing” significa que os testes devem

- a) ser feitos junto com a entrada em operação do produto.
- b) ser feitos apenas no ambiente de desenvolvimento.
- c) ser feitos apenas por meio de ferramentas de automação de testes.
- d) ser feitos desde as fases iniciais do ciclo de vida do produto.
- e) dirigir o desenvolvimento do produto.

Q2) [CESPE Ministerio da Economia 2020] Julgue o próximo item, a respeito de arquitetura em nuvem e virtualização.

Considere que um provedor de serviço de infraestrutura disponibilize recursos para seus usuários, que podem fazer uso em formato multi-tenant e têm flexibilidade para aumentar e diminuir rapidamente o uso de recursos. Nessa situação, a infraestrutura tem características de uma nuvem computacional.

Q1) [CESGRANRIO Caixa 2021] No âmbito de DevOps, o termo “shift left testing” significa que os testes devem

- a) ser feitos junto com a entrada em operação do produto.
- b) ser feitos apenas no ambiente de desenvolvimento.
- c) ser feitos apenas por meio de ferramentas de automação de testes.
- d) ser feitos desde as fases iniciais do ciclo de vida do produto.
- e) dirigir o desenvolvimento do produto.

Q2) [CESPE Ministerio da Economia 2020] Julgue o próximo item, a respeito de arquitetura em nuvem e virtualização.

Considere que um provedor de serviço de infraestrutura disponibilize recursos para seus usuários, que podem fazer uso em formato multi-tenant e têm flexibilidade para aumentar e diminuir rapidamente o uso de recursos. Nessa situação, a infraestrutura tem características de uma nuvem computacional. CERTO.

Q3) [CESPE SEFAZ-CE 2021] Julgue o item a seguir, referentes ao conjunto de práticas de desenvolvimento de software, operação e de apoio envolvidas (DevOps).

Para tornar a integração contínua mais efetiva no DevOps, é recomendável centralizar todos os commits em uma máquina de integração.

Q4) [CESPE SERPRO 2021] Acerca dos fundamentos e dos princípios da qualidade de software e da gestão da configuração, julgue o item que se segue.

Na gestão da configuração em que se observa os valores do DevOps, é importante implantar imediatamente as versões desenvolvidas no ambiente de produção, cabendo, nesse contexto, à equipe de desenvolvimento a criação de scripts automatizados para a garantia da integração contínua.

Q5) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

DevOps é uma tecnologia que pode ser empregada somente em softwares da categoria open source.

Q3) [CESPE SEFAZ-CE 2021] Julgue o item a seguir, referentes ao conjunto de práticas de desenvolvimento de software, operação e de apoio envolvidas (DevOps).

Para tornar a integração contínua mais efetiva no DevOps, é recomendável centralizar todos os commits em uma máquina de integração. CERTO.

Q4) [CESPE SERPRO 2021] Acerca dos fundamentos e dos princípios da qualidade de software e da gestão da configuração, julgue o item que se segue.

Na gestão da configuração em que se observa os valores do DevOps, é importante implantar imediatamente as versões desenvolvidas no ambiente de produção, cabendo, nesse contexto, à equipe de desenvolvimento a criação de scripts automatizados para a garantia da integração contínua. ERRADO.

Q5) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

DevOps é uma tecnologia que pode ser empregada somente em softwares da categoria open source. ERRADO.

Q6) [COVEST UFPE 2019] Sobre a arquitetura de microsserviços, é correto afirmar que ela:

- a) facilita a localização de serviços graças ao uso padronizado do UDDI.
- b) facilita e acelera o deploy por permitir que serviços sejam atualizados, independentemente do resto do sistema.
- c) diminui a tolerância a falhas no sistema, o que pode fazer que uma falha em um microsserviço se propague para todo o sistema.
- d) melhora a escalabilidade do sistema devido à sua abordagem monolítica.
- e) traz homogeneidade por propor uma só tecnologia para a construção dos microsserviços.

Q6) [COVEST UFPE 2019] Sobre a arquitetura de microsserviços, é correto afirmar que ela:

- a) facilita a localização de serviços graças ao uso padronizado do UDDI.
- b) facilita e acelera o deploy por permitir que serviços sejam atualizados, independentemente do resto do sistema.
- c) diminui a tolerância a falhas no sistema, o que pode fazer que uma falha em um microsserviço se propague para todo o sistema.
- d) melhora a escalabilidade do sistema devido à sua abordagem monolítica.
- e) traz homogeneidade por propor uma só tecnologia para a construção dos microsserviços.

Q7) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

As organizações DevOps não possuem controles eficientes como, por exemplo, a separação de tarefas e revisões de segurança manuais ao final do projeto; por isso, o DevOps é incompatível com segurança da informação e conformidade.

Q8) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

DevOps permite que os desenvolvedores aumentem a sua produtividade.

Q9) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

Em DevOps, ao submeter o código ao sistema de controle de versão, o desenvolvedor utiliza, entre outros, o teste de carga (estresse), que, basicamente, mede e avalia o tempo de resposta, o número de transações e outros requisitos sensíveis ao tempo.

Q7) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

As organizações DevOps não possuem controles eficientes como, por exemplo, a separação de tarefas e revisões de segurança manuais ao final do projeto; por isso, o DevOps é incompatível com segurança da informação e conformidade. ERRADO.

Q8) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

DevOps permite que os desenvolvedores aumentem a sua produtividade. CERTO.

Q9) [CESPE Ministério Economia 2020] Acerca da containerização de aplicações e DevOps, julgue o item subsequente.

Em DevOps, ao submeter o código ao sistema de controle de versão, o desenvolvedor utiliza, entre outros, o teste de carga (estresse), que, basicamente, mede e avalia o tempo de resposta, o número de transações e outros requisitos sensíveis ao tempo. ERRADO.

Q10) [CESPE MPE-PI 2018] Julgue o item a seguir, concernentes a microsserviços e arquiteturas de integração.

Um princípio básico dos microsserviços é que cada serviço gerencia seus próprios dados, sendo responsável pelo armazenamento particular desses dados e também pela execução em seus próprios processos.

Q11) [CESPE SLU-DF 2019] Com relação a DevOps e TOGAF, julgue o seguinte item.

Em DevOps, o princípio monitorar e validar a qualidade operacional antecipa o monitoramento das características funcionais e não funcionais dos sistemas para o início do seu ciclo de vida, quando as métricas de qualidade devem ser capturadas e analisadas.

Q10) [CESPE MPE-PI 2018] Julgue o item a seguir, concernentes a microsserviços e arquiteturas de integração.

Um princípio básico dos microsserviços é que cada serviço gerencia seus próprios dados, sendo responsável pelo armazenamento particular desses dados e também pela execução em seus próprios processos. CERTO.

Q11) [CESPE SLU-DF 2019] Com relação a DevOps e TOGAF, julgue o seguinte item.

Em DevOps, o princípio monitorar e validar a qualidade operacional antecipa o monitoramento das características funcionais e não funcionais dos sistemas para o início do seu ciclo de vida, quando as métricas de qualidade devem ser capturadas e analisadas. CERTO

Q12) [CESPE MPC-PA 2019] Assinale a opção que apresenta uma das principais características que diferencia o DevOps (desenvolvimento e operações) dos demais métodos da engenharia de software.

a) Atribuir as atividades inerentes à equipe de desenvolvimento de software e as atividades da equipe de operações, para que não haja sobreposição de responsabilidades.

b) Apresentar processo sistematizado de tarefas sequenciais adotadas para alinhar o que é feito isoladamente pela equipe de desenvolvimento, de modo a garantir que, ao sair da fábrica, o produto esteja de acordo com os artefatos e requisitos solicitados, no início do projeto, pela equipe de produção.

c) Ter como referência o conceito de entrega contínua ou CD (continuous delivery), realizada por um processo integrado e automatizado de desenvolvimento, teste de software e operações.

d) Trabalhar, primeiramente, a análise e a validação da documentação de software para, depois, planejar a viabilidade de automação de uma das fases do processo de desenvolvimento.

e) Ter sido o primeiro método a cunhar o termo “integração contínua”, que permite que todas as áreas envolvidas no desenvolvimento de software participem, por meio de reuniões rápidas, do planejamento, desenvolvimento e teste de software.

Q12) [CESPE MPC-PA 2019] Assinale a opção que apresenta uma das principais características que diferencia o DevOps (desenvolvimento e operações) dos demais métodos da engenharia de software.

a) Atribuir as atividades inerentes à equipe de desenvolvimento de software e as atividades da equipe de operações, para que não haja sobreposição de responsabilidades.

b) Apresentar processo sistematizado de tarefas sequenciais adotadas para alinhar o que é feito isoladamente pela equipe de desenvolvimento, de modo a garantir que, ao sair da fábrica, o produto esteja de acordo com os artefatos e requisitos solicitados, no início do projeto, pela equipe de produção.

c) Ter como referência o conceito de entrega contínua ou CD (continuous delivery), realizada por um processo integrado e automatizado de desenvolvimento, teste de software e operações.

d) Trabalhar, primeiramente, a análise e a validação da documentação de software para, depois, planejar a viabilidade de automação de uma das fases do processo de desenvolvimento.

e) Ter sido o primeiro método a cunhar o termo “integração contínua”, que permite que todas as áreas envolvidas no desenvolvimento de software participem, por meio de reuniões rápidas, do planejamento, desenvolvimento e teste de software.

Q13) [CESPE ABIN 2018] Acerca de microsserviços e práticas ágeis, julgue o item a seguir.

Em uma arquitetura de microsserviço, caso sejam criados vários sistemas, a falha de algum deles não afetará os demais sistemas.

Q14) [FCC SEFAZ-SC 2018] No âmbito da infraestrutura, uma das vantagens DevOps é

- a) realizar deploy sob controle manual de especialistas.
- b) estabelecer uma divisão bem delimitada entre infraestrutura e desenvolvimento.
- c) possuir um ambiente aberto (não padrão) sem muitos controles.
- d) ter a infraestrutura como código.
- e) fazer com que cada etapa do processo seja aprovada formalmente pelos gerentes.

Q15) [CESPE Ministério Economia 2020] A respeito de arquiteturas orientadas a serviços, web services e DevOps, julgue o próximo item.

Soluções orientadas a serviços devem ser compostas de serviços construídos ou como web services ou como componentes.

Q13) [CESPE ABIN 2018] Acerca de microsserviços e práticas ágeis, julgue o item a seguir.

Em uma arquitetura de microsserviço, caso sejam criados vários sistemas, a falha de algum deles não afetará os demais sistemas. CERTO.

Q14) [FCC SEFAZ-SC 2018] No âmbito da infraestrutura, uma das vantagens DevOps é

- a) realizar deploy sob controle manual de especialistas.
- b) estabelecer uma divisão bem delimitada entre infraestrutura e desenvolvimento.
- c) possuir um ambiente aberto (não padrão) sem muitos controles.
- d) ter a infraestrutura como código.
- e) fazer com que cada etapa do processo seja aprovada formalmente pelos gerentes.

Q15) [CESPE Ministério Economia 2020] A respeito de arquiteturas orientadas a serviços, web services e DevOps, julgue o próximo item.

Soluções orientadas a serviços devem ser compostas de serviços construídos ou como web services ou como componentes. ERRADO.