

E B O O K

DevOps

**Por que você precisa
implantar ou não
na sua empresa?**

Guia completo
Ferramentas
Etapas
Ações e Fluxos
Conceito
Mitos

objective



SUMÁRIO

Introdução	3
O que é DevOps e sua importância	4
Adotando DevOps na sua empresa	7
Sugestões de Ferramentas	11
Ferramentas adotadas e aprovadas pela Objective	20
Test Tools	27
Cloud e DevOps	29
7 mitos sobre DevOps	33
Conclusão	37
Sobre a Objective	38

Apesar de sua importância, algumas empresas ainda não implantaram DevOps em suas equipes de TI.

Diante deste cenário bem comum, podemos detectar alguns problemas semelhantes em ambientes que não possuem as práticas DevOps, como dificuldade na tomada de decisões, baixa produtividade do time, estimativas erradas de entrega dos projetos que refletem em atrasos, demoras e desmotivação do time, probabilidade elevada de erros e bugs, falta de integração e desempenho, além do esforço elevado das pessoas sem um resultado eficaz e eficiente.

Este e-book é para pessoas que trabalham e gerenciam equipes de Tecnologia da Informação e querem aprimorar os seus processos para as entregas contínuas de software e melhorar a governança de TI com o envolvimento dos times de desenvolvimento, operações, qualidade, testes, entre outros.

Aqui você vai encontrar ferramentas, processos e responsabilidades para sua equipe adotar e exercer as práticas DevOps, contribuindo cada vez mais para que as entregas sejam feitas mais rapidamente e com qualidade.

O QUE É DEVOPS, COMO SURGIU E SUA IMPORTÂNCIA

Para começar de forma mais simples possível, você precisa entender que DevOps é a adequação de processos de desenvolvimento (Dev) e operação (Ops) para um formato único e contínuo, facilitando a integração entre os recursos envolvidos, entregando aos clientes, produtos ou serviços de forma rápida, prezando a eficácia e eficiência.

Falando um pouco de sua história, lá atrás, no Manifesto Ágil de 2001, após a disseminação de concepções como entrega contínua, desenvolvimento ágil e deploy contínuo, é que se incorporou a ideia do DevOps. Há uma sinergia entre metodologias ágeis e DevOps, mas não há nenhum vínculo de ordem ou causa e efeito.

Apesar de serem levantados esses pontos críticos, o DevOps, de fato, surgiu apenas em 2008, quando Patrick Debois, administrador de sistemas, gerente de projetos e especialista em desenvolvimento, iniciou uma discussão sobre como solucionar os eternos conflitos entre as áreas de desenvolvimento e operações de TI. O tema foi abordado em uma palestra feita na Conferência de Agile, em Toronto (caso queira assistir à apresentação, clique aqui).



A partir desse momento as empresas passaram a ter um olhar mais crítico e voltado para a aplicação de mudanças rápidas, de forma que estabelecesse uma relação de parceria entre os departamentos, que são capazes de garantir a gestão eficaz com alto desempenho, prazo, escopo, redução de custo e alta qualidade.

Toda a estimativa errada da entrega do projeto, bugs e gaps, falta de procedimentos básicos e a baixa produtividade do time são fatores que podem ser endereçados com o DevOps.

Em março de 2011, o Gartner publicou um relatório afirmando que, até o final de 2015, o DevOps se tornaria a principal estratégia em 20% das organizações mundiais.

De lá para cá o DevOps teve seu “boom” e foi adotado por diversas grandes empresas.

A média de falhas geradas pelas empresas que adotaram DevOps é cerca de 50 vezes menor, se comparada com as que não usam esta prática.

**HOJE, SEGUNDO PESQUISA PUBLICADA PELA PUPPET LABS,
63% DOS PESQUISADOS QUE ADOTARAM DEVOPS EM
SUAS EMPRESAS CONSEGUEM ENTREGAR A DEMANDA
30 VEZES MAIS RÁPIDO.**

MAS PORQUÊ DEVOPS VAI TE AJUDAR TANTO ASSIM?

Se ainda não está convencido com os pontos levantados acima, podemos te garantir que é possível medir em ganhos reais os benefícios, uma vez que um deles é trocar horas de esforço humano por custo de máquina.

Confira alguns benefícios e valores mensuráveis:

BENEFÍCIOS

- Gerencie o desenvolvimento de forma mais assertiva
- Diminua gaps e bugs
- Automatize os processos e melhore o relacionamento das áreas
- Habilite a integração e entrega contínua
- Diminua consideravelmente o Lead Time
- Maior qualidade na entrega
- Menor complexidade de gerência
- Times mais produtivos

VALORES

- Automação
- Confiança
- Cultura
- Colaboração

Vale destacar que a contratação de um profissional ou uma empresa especializada para implementar as práticas DevOps é fundamental, considerando o nível de complexidade do trabalho. Eles serão os responsáveis por adequar a estratégia de negócio através de fluxos de trabalho eficazes, além de garantir a melhor utilização dos softwares adotados.

ADOTANDO DEVOPS NA SUA EMPRESA

Agora que você já teve uma visão geral do que é DevOps, o surgimento dele e a importância da adoção, vamos entender os processos para que sua empresa possa começar o quanto antes a implantar essa prática.

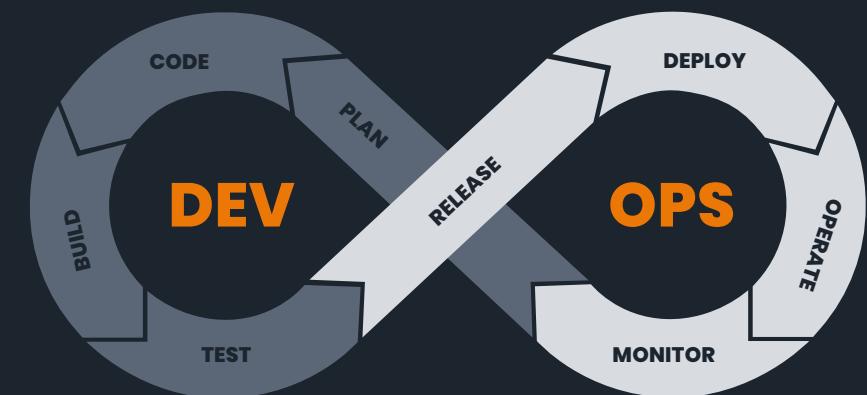
É importante ressaltar que cada empresa atua com um ciclo de DevOps levando em consideração a estrutura e seus aspectos culturais. Vamos compartilhar aqui o modelo aplicado na Objective Solutions, que está em evolução desde 1999.

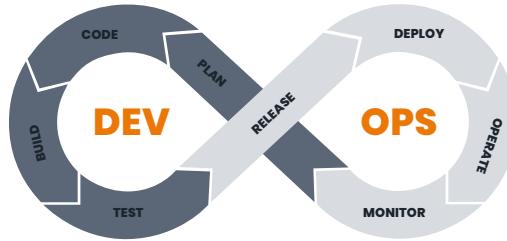
Para entender melhor como funciona na prática, veja pela imagem ao lado as etapas do ciclo do DevOps, podem ser identificadas oito partes que descrevemos para você.

E é seguindo essas etapas definidas na imagem ao lado que a sua empresa será DevOps eficiente!

São elas: Plan, Code, Build, Test, Release, Deploy, Operate, Monitor.

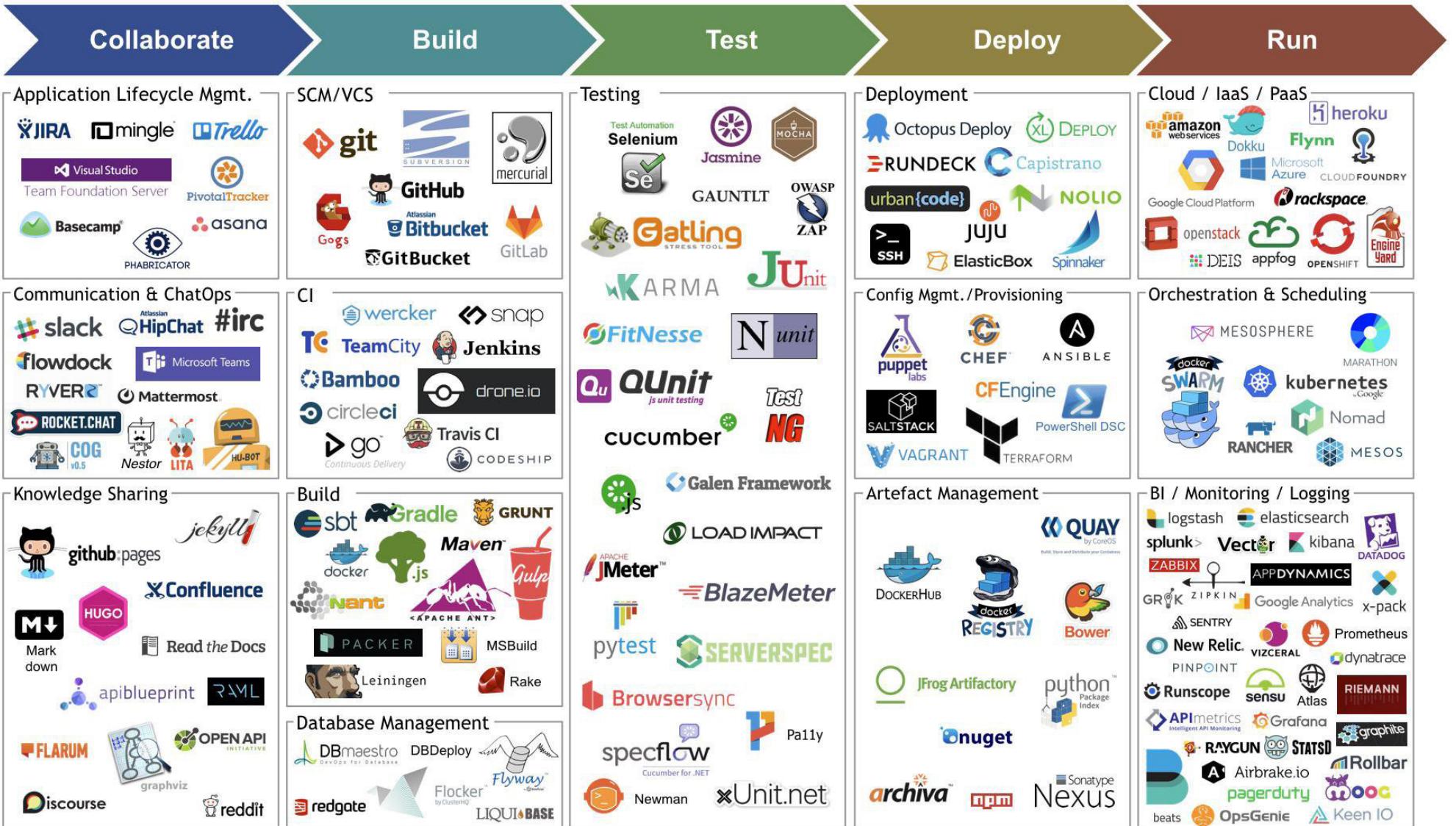
Vem entender um pouco mais sobre cada etapa.





PLAN O planejamento é a etapa que define o escopo da atividade que será executada.	CODE A codificação é a etapa onde será realizado o desenvolvimento de código.	BUILD A construção é a etapa onde os artefatos serão gerados.	TEST Nessa etapa os artefatos são testados.
RELEASE A etapa de entrega é onde os artefatos são enviados para um repositório de artefatos.	DEPLOY Na etapa de implantação os artefatos são configurados nos servidores.	OPERATE Na operação são realizadas as manutenções de eventuais problemas que o ambiente possa apresentar.	MONITOR Na etapa de monitoria são configurados os monitores do ambiente para acompanhamento de eventuais problemas.

Para cada etapa temos uma grande variedade de ferramentas que auxiliam na prática, como mostramos na figura abaixo e abordaremos algumas nos próximos capítulos.



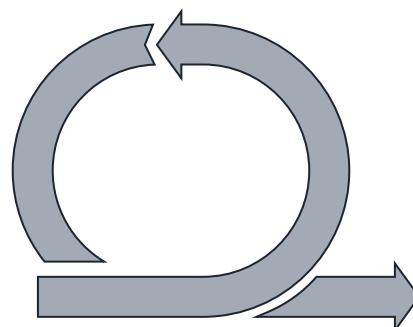


A ESCOLHA DO MÉTODO DE GESTÃO

Ressaltamos que DevOps não é apenas para atividades com começo, meio e fim. DevOps é uma estratégia com ciclo contínuo que deve ser implantado na cultura das equipes de TI.

Por ser uma mudança cultural e impactar no trabalho dos profissionais é sempre algo complexo de ser feito, uma vez que somos acostumados a adotar uma prática porque sempre todos trabalharam assim e somos naturalmente resistentes às mudanças.

Por isso, algumas companhias optam por adotar parceria com empresas que são focadas e especialistas em implantação de Metodologias Ágeis e DevOps, que tem como premissa formar equipes multidisciplinares com a correta adoção de ferramentas, acelerar a implantação de melhorias de software, automatizar os processos de build e reduzir os erros de forma integrada, levando a uma forte vantagem competitiva nos negócios.



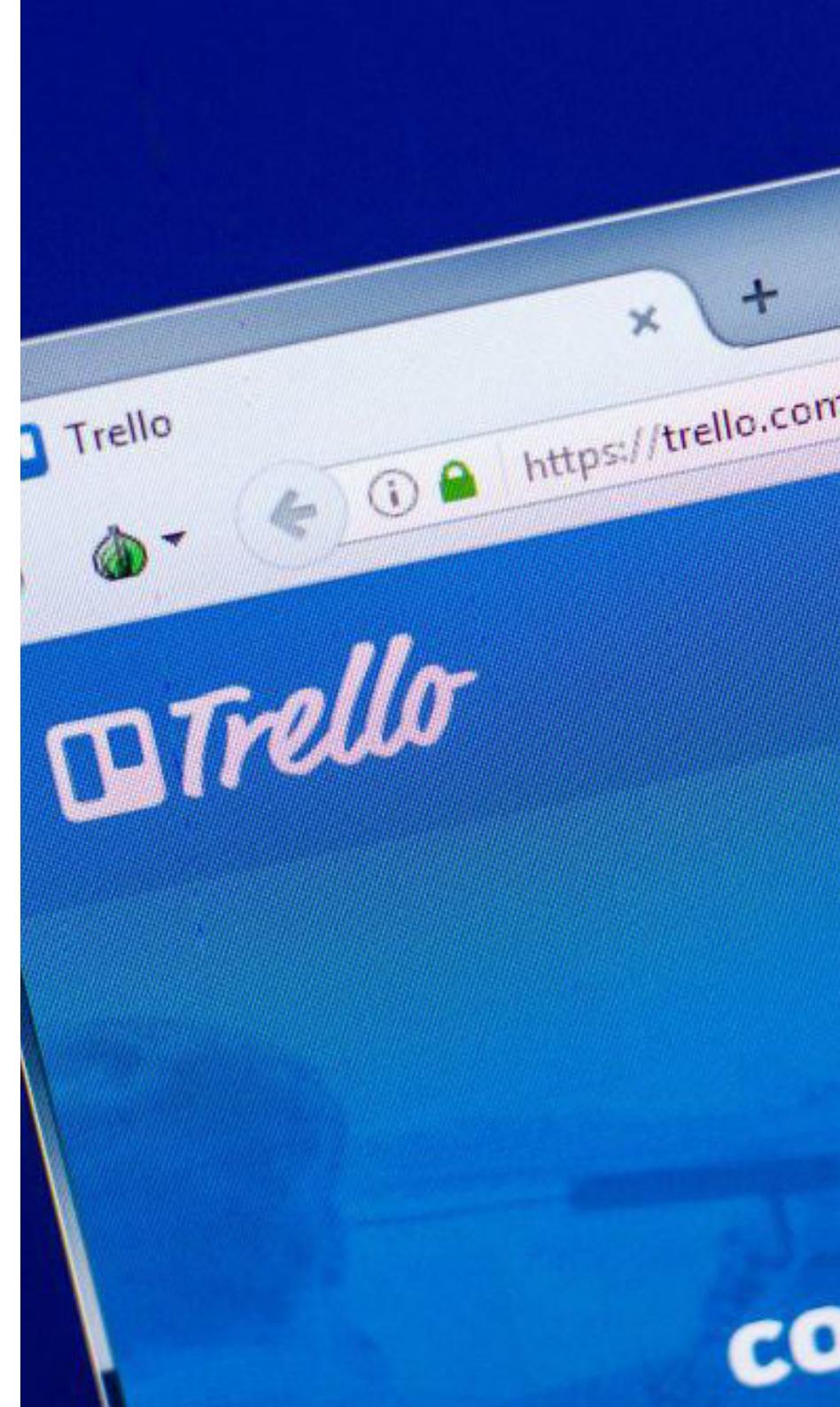
SUGESTÕES DE FERRAMENTAS

Como falamos anteriormente, para cada etapa do DevOps temos uma grande variedade de ferramentas que auxiliam na prática.

A seleção de algumas dessas ferramentas pode ser exclusiva em relação ao projeto, como por exemplo, o Java utiliza ferramentas como: Maven para automatização de build e JUnit como ferramenta de testes automatizados.

Já o One Click é uma ferramenta que realiza a configuração para novos projetos no ambiente, facilitando a inicialização das atividades. A configuração é realizada em sistemas como, Jira, Taskboard, Integração contínua, Versionamento e Análise de código.

Descreveremos à seguir as ferramentas que consideramos importantes para atuar em cada fase do DevOps.



PLANEJAMENTO (PLAN)

TRELLO

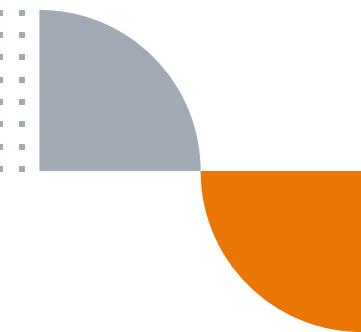
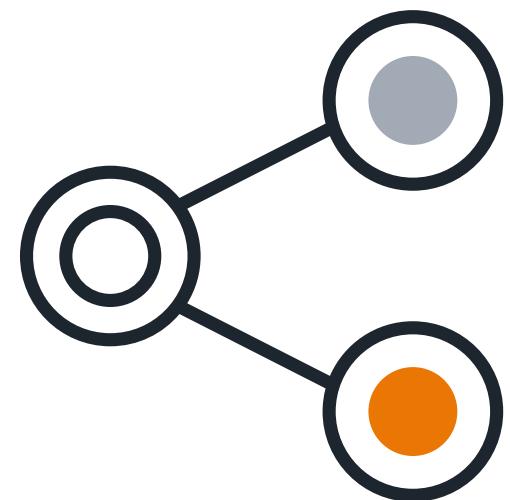
O Trello é uma ferramenta prática, versátil e ajustável conforme necessidade para o gerenciamento de projetos. Nele são criados cartões com as tarefas para o uso da equipe.

JIRA

Essa ferramenta permite que sejam criadas tarefas de projetos, garantindo um gerenciamento unificado. É uma ferramenta bem abrangente, que com a utilização de plugins atende diversas funcionalidades de gestão de projetos.

ASANA

Com intuito de permitir e aumentar a colaboração em time, o Asana é uma ferramenta que ajuda as equipes terem visibilidade da trajetória de suas atividades, focada em gestão de projetos.



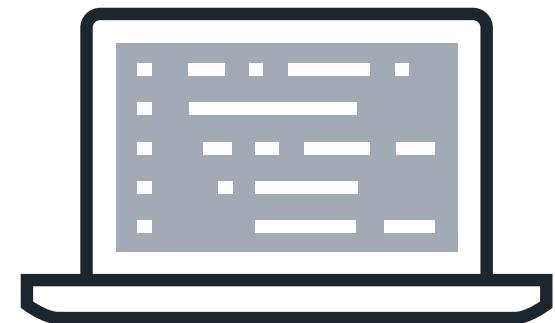
SCM SOURCE CODE MANAGEMENT CONTROLE DE VERSÃO DE CÓDIGO (CODE)

SUBVERSION

O Subversion, mais conhecido com SVN, realiza o controle de versionamento e revisão do código fonte ou conteúdo de softwares, documentos e páginas da web.

GIT

O Git é um sistema de controle de versionamento e gerência de código fonte, que facilita a colaboração entre várias pessoas para criação e manutenção no código fonte de um software, pois não segue o modelo cliente-servidor.



CONSTRUÇÃO (BUILD)

MAVEN

O Maven realiza compilação de código de forma automatizada e pode ser utilizado com várias linguagens diferentes por ser baseado em plugins.

DBMAESTRO

Com o DBMaestro podem ser realizadas automatizações nas entradas de versões de bancos de dados com maior confiança, uma vez que todas as atualizações serão validadas e auditadas pela ferramenta.



TESTES (TEST)

JUNIT

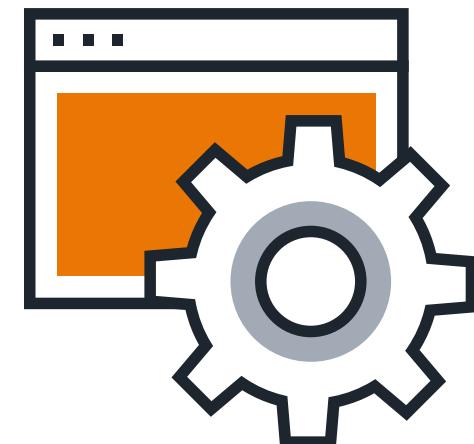
O JUnit é uma ferramenta que facilita a construção de testes com a apresentação de resultados, possibilitando o teste de componentes do sistema de maneira isolada para linguagem de programação Java.

NUNIT

O NUnit, assim como o JUnit, é uma ferramenta de construção de testes automatizados, porém sua plataforma de atuação é a Microsoft .NET.

SELENIUM

O Selenium realiza testes automatizados em websites em duas perspectivas. Uma delas é a do programador através de códigos e a outra é a do usuário através das telas do site.



ENTREGA (RELEASE)

DOCKERHUB

O DockerHub é um gerenciador de versionamento de artefatos do Docker. Com ele é possível ter várias versões de sistemas operacionais e softwares.

NEXUS

O Nexus provê um repositório de versionamento centralizado para qualquer artefato construído ou utilizado no projeto.

PYTHON PACKAGE INDEX

Mais conhecido como PyPI, o Python Package Index, é o repositório centralizado de artefatos da linguagem Python.



IMPLEMENTAÇÃO (DEPLOY)

PUPPET

O Puppet realiza o gerenciamento de configurações, assegurando que todas elas sejam feitas exatamente iguais em todas máquinas do ambiente de forma rápida e segura.

CHEF

O Chef instala e configura as ferramentas necessárias para o uso do ambiente pelo sistema de forma rápida.





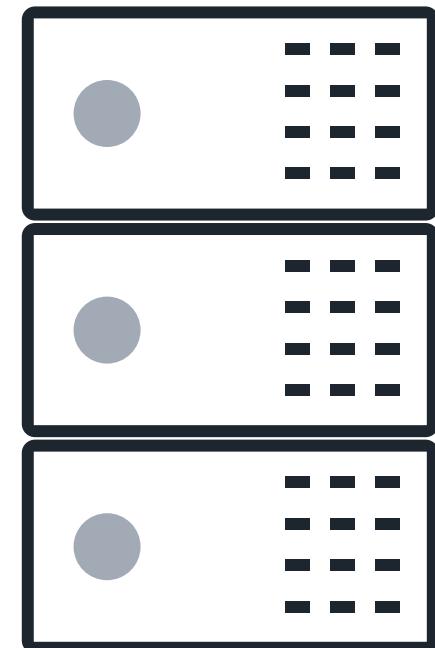
OPERAÇÃO (OPERATE)

AMAZON AWS

A Amazon possui um serviço chamado Amazon Web Service (AWS), que disponibiliza servidores pré configurados para utilização de serviços e ferramentas.

AZURE

O Microsoft Azure é similar ao Amazon AWS e também disponibiliza um ambiente para utilização de serviços e ferramentas, porém ele possui algumas soluções específicas Microsoft.



MONITORIA (MONITOR)

SPLUNK

O Splunk é uma ferramenta analítica de logs. Nele podem ser criados alertas com frases específicas que venham aparecer nos logs.

ZABBIX

O Zabbix realiza as monitorias de Sistemas Operacionais, Switches e Storages de forma rápida e prática. Ele possui monitores já configurados, mas podem ser criados monitores customizados.

APPDYNAMICS

As monitorias de máquinas virtuais do Java (JVMs) podem ser realizadas utilizando o AppDynamics.

Agora, aprofundamos mais, detalhando algumas ferramentas das quais a Objective já tem vasta experiência e utilizamos em nosso dia a dia.



FERRAMENTAS ADOTADAS E APROVADAS PELA OBJECTIVE

GIT

As ferramentas de controle de versionamento de código, como o Git, são fundamentais para a aplicação do DevOps, uma vez que todo o processo de automatização, na maioria das vezes, é iniciado a partir do código contido nessas ferramentas.

Essas ferramentas trazem benefícios como realização de cópias da trilha principal para o desenvolvimento paralelo sem alteração direta na trilha principal, múltiplos times atuando simultaneamente no mesmo código com organização e controle das alterações do mesmo.

O Git é uma ferramenta gratuita de código aberto. Isso significa que seu código pode ser alterado em caso de necessidade conforme demanda.

MAVEN

O Maven é uma ferramenta que automatiza a construção de artefatos escritos em linguagens como C#, RUBY, Scala e Java. Ele possui uma arquitetura baseada em plugins, o que permite a implementação deles fazer interface com as ferramentas de construção para muitas linguagens.

Essa ferramenta traz benefícios como automatização da construção do código, automatização de testes do código de diferentes linguagens, padronização na entrega, escalabilidade com pequenos ajustes.

E o bom é que o Maven é uma ferramenta gratuita!

JUNIT

O JUnit, como ferramenta de teste de código Java, facilita o processo de criação de código para automatização dos testes. Com ele podem ser realizados testes unitários, em pequenos pontos do código, realizando testes chamados de “caixa branca”, onde o código é exercitado retornando o resultado esperado, o que facilita a correção de métodos e objetos.

Essa ferramenta traz benefícios como testes simultâneos ao desenvolvimento, menos tempo na depuração de códigos, criação rápida de testes aumentando a qualidade do sistema.

O JUnit também é uma ferramenta gratuita de código aberto.rado em caso de necessidade conforme demanda.

NEXUS

Após a construção dos artefatos é interessante o armazenamento centralizado dos mesmos. Partindo desse princípio, o Nexus realiza essa centralização facilitando o acesso às versões do sistema e o fluxo de automatização do ambiente.

Essa ferramenta traz benefícios como centralização das versões do sistema e historização de versionamento. O Nexus possui duas modalidades de licenciamento, uma delas é gratuita, porém com menos recursos e a outra é licenciamento por usuário.

JENKINS

O Jenkins é uma ferramenta muito versátil, pois muitas das tarefas são configuradas através de plugins que podem ser diversos. Ele pode ser utilizado em alguns ciclos do processo. Um deles é o Release, onde nele podem ser criadas tarefas de compilação, construção e testes do código. A outra é o Deploy, onde as tarefas são configuradas para realizar a atualização do sistema.

Essa ferramenta traz benefícios como automatização das tarefas de teste, deploy e construção.

Como as mesmas ferramentas citadas acima, o Jenkins é gratuito e de código aberto.

ZABBIX

Para realizar a monitoria do ambiente pode ser utilizado o Zabbix. Ele realiza a coleta e armazenamento de algumas métricas do uso dos equipamentos para melhor dimensionamento, escalabilidade e monitoria do ambiente.

Essa ferramenta traz benefícios como monitoria ativa e automatização na solução de problemas.

Como os outros, o Zabbix é gratuito e de código aberto.

DOCKER

O Docker é uma ferramenta que realiza a criação de pacotes de software padronizados, facilitando a criação, o teste e a implantação de aplicações.

Essa ferramenta traz benefícios como escalabilidade, praticidade, agilidade e padronização de ambientes.

O Docker possui duas modalidades de licenciamento, uma delas é gratuita, porém com menos recursos, e a outra varia dependendo dos recursos e suporte agregados.

JIRA

Para gerenciar as tarefas pode ser utilizado o Jira, que contém recursos personalizáveis para o controle das atividades independentemente do tamanho do projeto.

Essa ferramenta traz benefícios como controle de atividades, controle de log de horas, controle de horas gastas em um projeto.

O Jira é uma ferramenta paga, e sua cobrança está relacionada diretamente com o número de usuários que usam a ferramenta. Quanto maior o número de usuários, menor seu valor por usuário.

PANDOISE

O Pandoise foi desenvolvido utilizando-se de uma interface gráfica e intuitiva, com respostas e classificação dos principais erros nas atividades de criação de ambiente de desenvolvimento, como: nova branch, checkout, down-merge, reintegração e replicação.

Essa ferramenta traz benefícios como padronização das principais ações de criação do ambiente de desenvolvimento, fácil usabilidade para novos recursos alocados no projetos, monitoramento do tempo de resposta e os erros das principais ações, visando a otimização e classificação dos erros comuns, com provas reais.

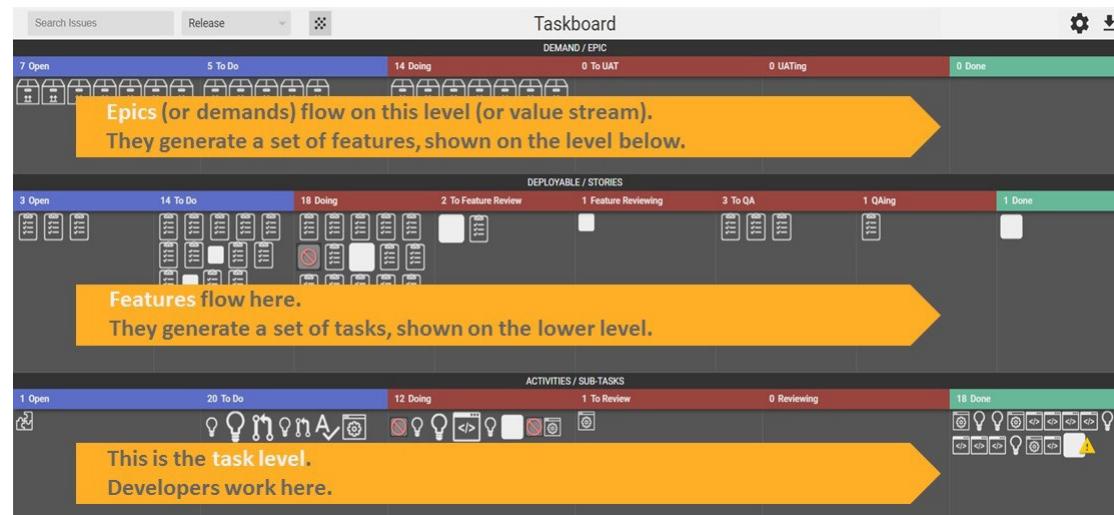
O Pandoise é uma ferramenta proprietária da Objective.

TASKBOARD OBJECTIVE

A Taskboard é um dos melhores meios de comunicação do status do projeto entre todo o time e do time para outros envolvidos como, gerentes, áreas afins e stakeholders. Favorecendo a colaboração e fluxo, com o mantra “Stop starting and start finishing”. A Taskboard da Objective possui três níveis de atividades: Demandas ou Épicos, Estórias ou Funcionalidades, Tarefas.

Demandas ou Épicos são mapeados para a especificação do cliente, ou seja, correspondem a incrementos de negócio. Estórias ou Funcionalidades possuem uma descrição mais detalhada e podem ser testados pelo cliente, pois possuem critérios de aceitação bem definidos.

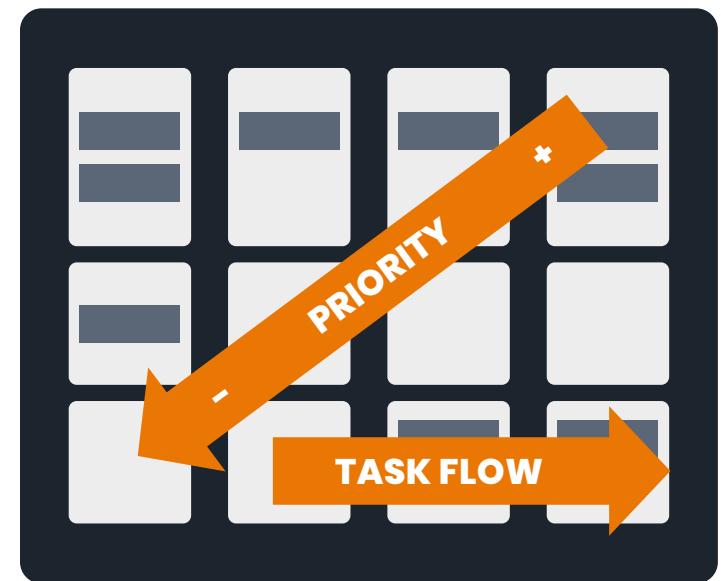
Tarefas são atividades rápidas, executadas pelo time e que usualmente são categorizadas como planejamento, desenvolvimento, teste, etc.





O objetivo de qualquer time é sempre buscar fluidez das atividades em todos os níveis no melhor ritmo sustentável possível. Para isso devemos ler a Taskboard do nível mais alto para o mais baixo e da direita para a esquerda.

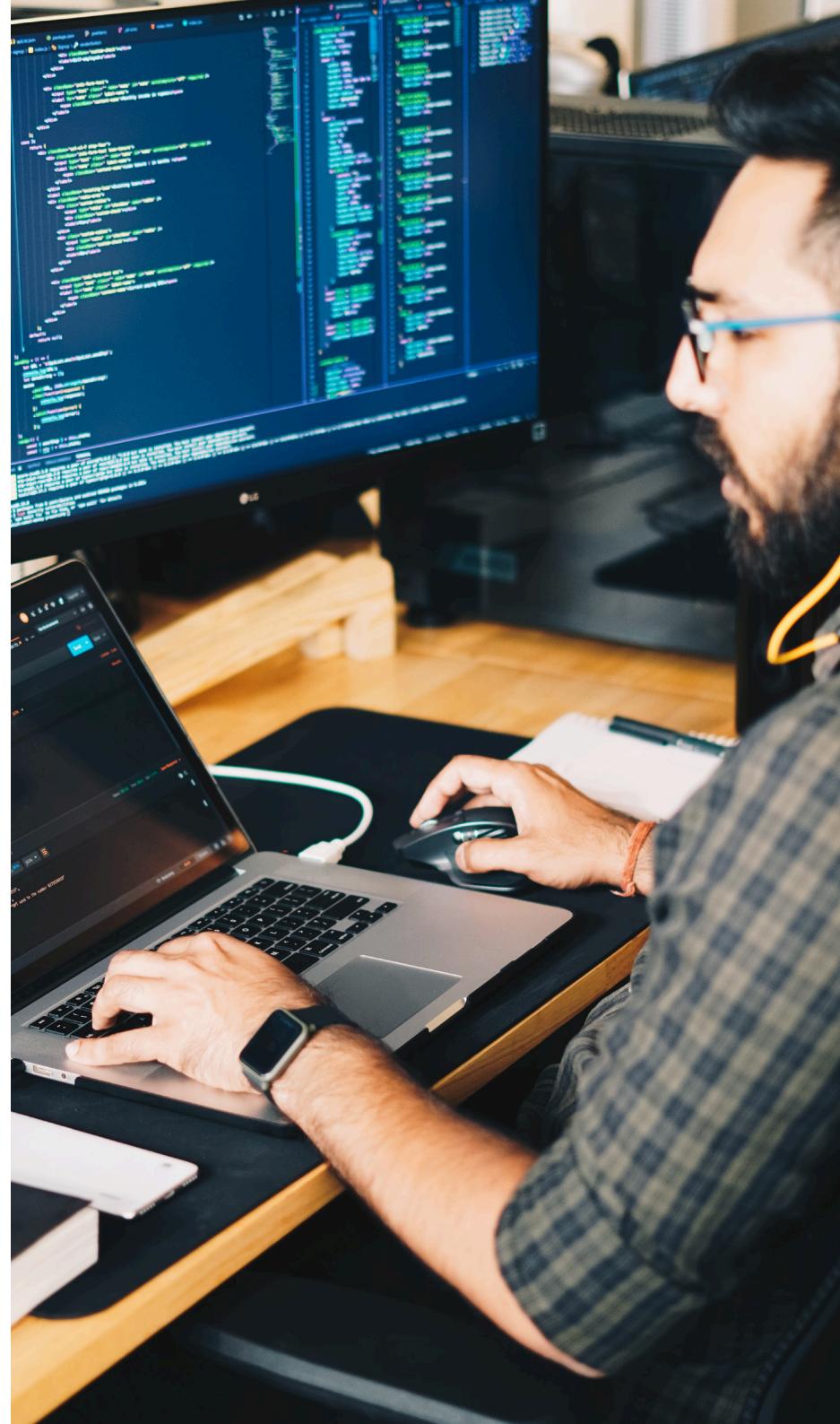
Isso garante prioridade às tarefas que estão mais próximas de serem completadas. O time deve considerar colaborar em atividades em andamento antes de começar novas.



TEST TOOLS

São ferramentas utilizadas efetivamente para garantia de qualidade e manutenibilidade. Esses testes são a base para todas as demais práticas ágeis de desenvolvimento. Temos diversos benefícios, descritos abaixo:

- Garantia de assertividade no “go-live” de projetos complexos.
- Redução de custos expressiva em relação a testes manuais no longo prazo
- Prevenção contra bugs de soluções de terceiros.
- Redução de erros causados por falhas em análises
- Prevenção contra problemas de integrações.
- Redução de problemas em projetos customizados.
- Transparência no apontamento de problemas.
- Redução de tempo de desenvolvimento.
- Possibilita refactoring e rápidas validações de alterações
- Garante a sobrevivência do sistema a longo prazo
- Treinamento do sistema



TEST DRIVEN DEVELOPMENT

Test Driven Development (TDD) ou em português

Desenvolvimento guiado por testes é uma técnica onde o desenvolvedor escreve o caso de teste e produz um código que possa ser validado pelo teste. Posteriormente o código é refatorado para padrões aceitáveis.



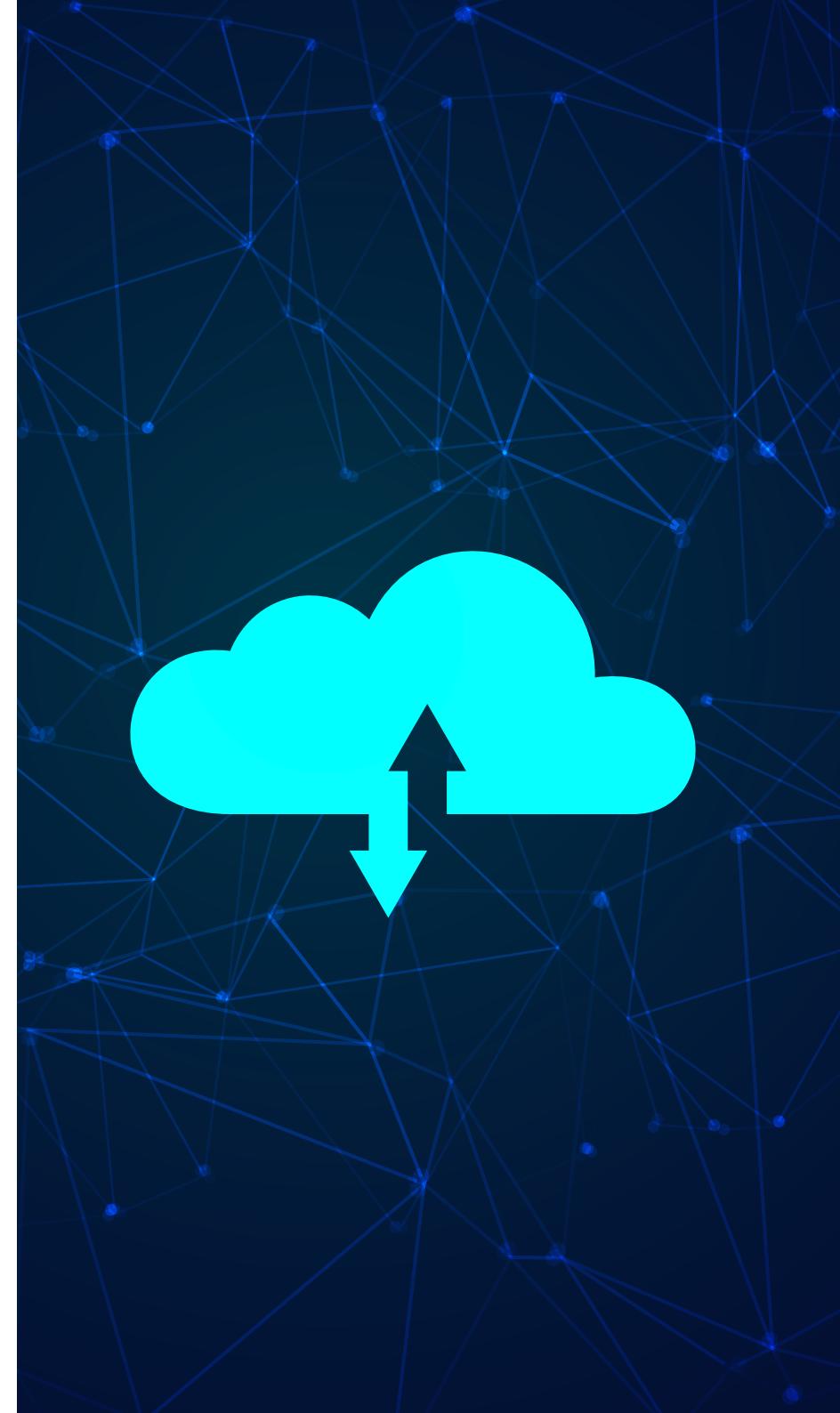
CLOUD E DEVOPS: O QUE MUDOU?

O cloud computing revolucionou muitos quesitos e um deles foi o próprio funcionamento da TI. Muitas empresas optam por aplicar o DevOps na nuvem.

Costumamos indicar a utilização da Cloud para a adoção do DevOps por tornar os processos mais fáceis, rápidos e econômicos, já que teremos uma oferta de recursos e ferramentas para atingir os objetivos de negócios da empresa.

O ambiente virtual oferece diversas possibilidades para as empresas que trabalham com DevOps sem precisar contar com altos investimentos.

Na Cloud, é possível contratar diversos serviços, como infraestrutura, plataformas e softwares de acordo com a demanda da empresa. Além disso, esse cenário é ideal para corporações iniciantes, pois elas poderão contar com o suporte das equipes dos fornecedores, que são experientes e especializadas na área.





Citamos aqui alguns pontos que detectamos como benefícios da implantação do DevOps as a Service em nossos clientes:

- ✓ **Entrega contínua**
- ✓ **Maior qualidade na entrega**
- ✓ **Menor complexidade de gerência**
- ✓ **Times mais produtivos**

Essas vantagens são conquistadas porque a computação em nuvem permite que os processos de submissão de códigos sejam modificados e isso seja feito de forma centralizada.



Cloud Computing é uma evolução tecnológica que traz diversos benefícios, porém, com isso, surge também novos desafios. Ou seja, nem tudo são flores.

As equipes não precisam mais se preocupar apenas com seu legado de aplicações cliente-servidor rodando em seus próprios servidores. Eles também passaram a ter ambientes em nuvens públicas, privadas ou híbridas para gerenciar, além da sua tradicional arquitetura.

Se esse é o seu caso e você não consegue mais demandar tempo das equipes, uma solução indicada é a contratação de empresas parceiras, com o core exclusivo para aplicação de DevOps ou Metodologias Ágeis, onde elas têm como objetivo garantir que esses desafios sejam solucionados com facilidade.

Para os casos de aplicação de DevOps na nuvem, também existem ferramentas que auxiliam nos processos. Citamos dois exemplos a seguir.



AMAZON AWS

O ambiente pode ser hospedado completamente na nuvem utilizando a Amazon AWS, onde todas as máquinas que auxiliam o processo de DevOps serão instaladas e configuradas.



Essa ferramenta traz benefícios como escalabilidade, centralização de gerenciamento de máquinas, redução de custos com máquinas físicas e segurança.

A Amazon AWS é uma ferramenta com uma abordagem de pagamento conforme o uso do cliente com valores iniciais em poucos centavos de dólar por hora/máquina.

MICROSOFT AZURE

O Microsoft Azure é similar ao Amazon AWS e também disponibiliza um ambiente para utilização de serviços e ferramentas, porém ele possui algumas soluções específicas Microsoft.

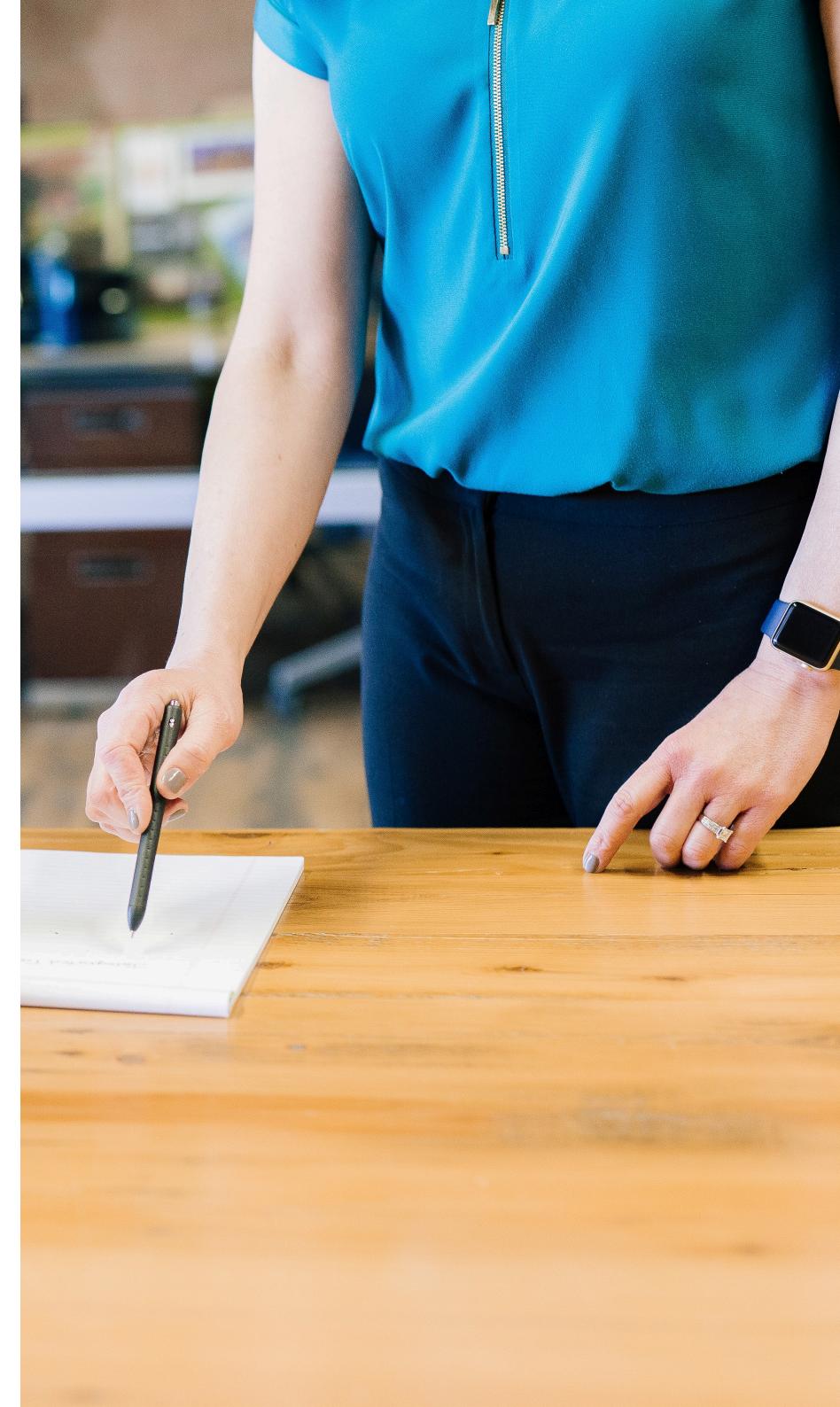


7 MITOS SOBRE DEVOPS

1. DEVOPS É SOMENTE SOBRE COMUNICAÇÃO ENTRE AS ÁREAS

Muitas pessoas erram ao explicar que DevOps é a comunicação e conhecimento alinhado entre desenvolvimento e operações.

Mais do que isso, DevOps é o sucesso da governança de TI. Ele está diretamente ligado a automação dos processos, monitoramento em todas as fases de construção do software e ferramentas, que de forma customizada, são capazes de garantir a gestão eficaz com alto desempenho, prazo, escopo, redução de custo e alta qualidade.





2. DEVOPS NÃO É PARA SISTEMAS GRANDES E COMPLEXOS

Exatamente os sistemas de grande complexidade que precisam ainda mais da disciplina e colaboração do DevOps, que vai facilitar a coordenação do ciclo de entrega e nível de qualidade de qualquer sistema.

Até porque qualquer que seja o sistema desenvolvido, sempre há diversos componentes de software e hardware.

3. DEVOPS SÓ PODE SER APLICADO EM DESENVOLVIMENTO E OPERAÇÕES

Talvez o nome DevOps possa remeter isso, uma vez que surgiu a partir da integração de desenvolvimento (dev) com operações (ops) conforme explicamos neste e-book.

Entretanto, o DevOps pode ser aplicado em todo o time, uma vez que parceiros, comerciais, fornecedores, entre outros, também tem participação na entrega dos softwares.

4. DEVOPS NÃO É PARA DESENVOLVIMENTO TERCEIRIZADO DE SISTEMAS

As execuções das práticas de DevOps são mais fáceis quando os times estão próximos, mas mesmo em times remotos ou terceirizados há ganhos significativos.

5. DEVOPS SÃO PROFISSIONAIS DE OPERAÇÕES APRENDENDO A CODIFICAR

Por conta da integração de equipes, existe sim uma evolução da infraestrutura usando programação.

As equipes de operações precisam gerenciar grandes quantidades de código com práticas de engenharia de software, como código de versão, check-in, check-out, ramificação e fusão.

Então um membro da equipe de operações cria uma nova versão do código que a define. Mas isso não significa, no entanto, que as equipes de operações precisam aprender a codificar em Java ou C#.

A maioria das equipes de infraestrutura usam idiomas como o Ruby, que são fáceis de utilizar, especialmente para pessoas com experiência em scripts.

6. SÓ É POSSÍVEL APLICAR DEVOPS COM CLOUD COMPUTING

É normal que quando pensamos em DevOps as pessoas relacionarem automaticamente com a nuvem.

Uma vez que a Cloud Computing proporciona recursos de infraestrutura que permitem aos desenvolvedores obterem um ambiente de testes automatizados rapidamente, sem esperar dias ou semanas para que uma solicitação manual seja cumprida.

Entretanto, a nuvem não é necessária para a aplicação das práticas de DevOps. Mas vale a ressalva de que a aplicação de DevOps sem a nuvem só será satisfatória caso a organização tenha processos eficientes para obter recursos para implantação de teste, por exemplo.

7. DEVOPS É APENAS PARA EMPRESAS QUE JÁ NASCERAM NO DIGITAL

Hoje grandes empresas estão investindo em DevOps porque as práticas aplicadas, já descritas neste e-book, tem um nível de maturidade que os torna aplicáveis em grandes empresas que, inclusive, utilizam de múltiplas plataformas, tecnologias e diversos times.

CONCLUSÃO

Agora que você já leu todos os pontos que abordados neste e-book pode perceber que aplicar DevOps na equipe de TI da sua empresa não é uma tarefa fácil.

Exige muito estudo, dedicação, aceitação e colaboração das equipes como um todo. Implementar DevOps requer um plano estratégico passo a passo, que depende de como você gerencia e integra os times envolvidos.

Como vimos, embora o DevOps seja principalmente sobre pessoas, as ferramentas também desempenham um papel importante.

Mas como um start nessa mudança cultural, você já pode apresentar os benefícios mensuráveis que o DevOps poderá agregar a sua empresa e começar com implementações simples das práticas no dia a dia.

Claro que as informações que compartilhamos irão variar dependendo da estratégia de negócio e do segmento de cada empresa, mas independente de por onde começar ou com quem irão aplicar, é importante seguir as etapas do ciclo para realmente ter uma diferença visível.

NÃO ESPERE O AMANHÃ. APLIQUE SUAS IDEIAS HOJE MESMO!

SOBRE A OBJECTIVE

A Objective atua desde 1999 com DevOps e pode ajudar a sua empresa na remoção das barreiras departamentais, melhorar o fluxo dos processos e estruturar as ferramentas corretas nesta implantação avaliando aspectos tecnológicos e culturais da sua empresa com mais agilidade e qualidade.

Elaboramos um plano de transformação cultural para reduzir drasticamente o tempo que a empresa leva para lançar novas iniciativas digitais.

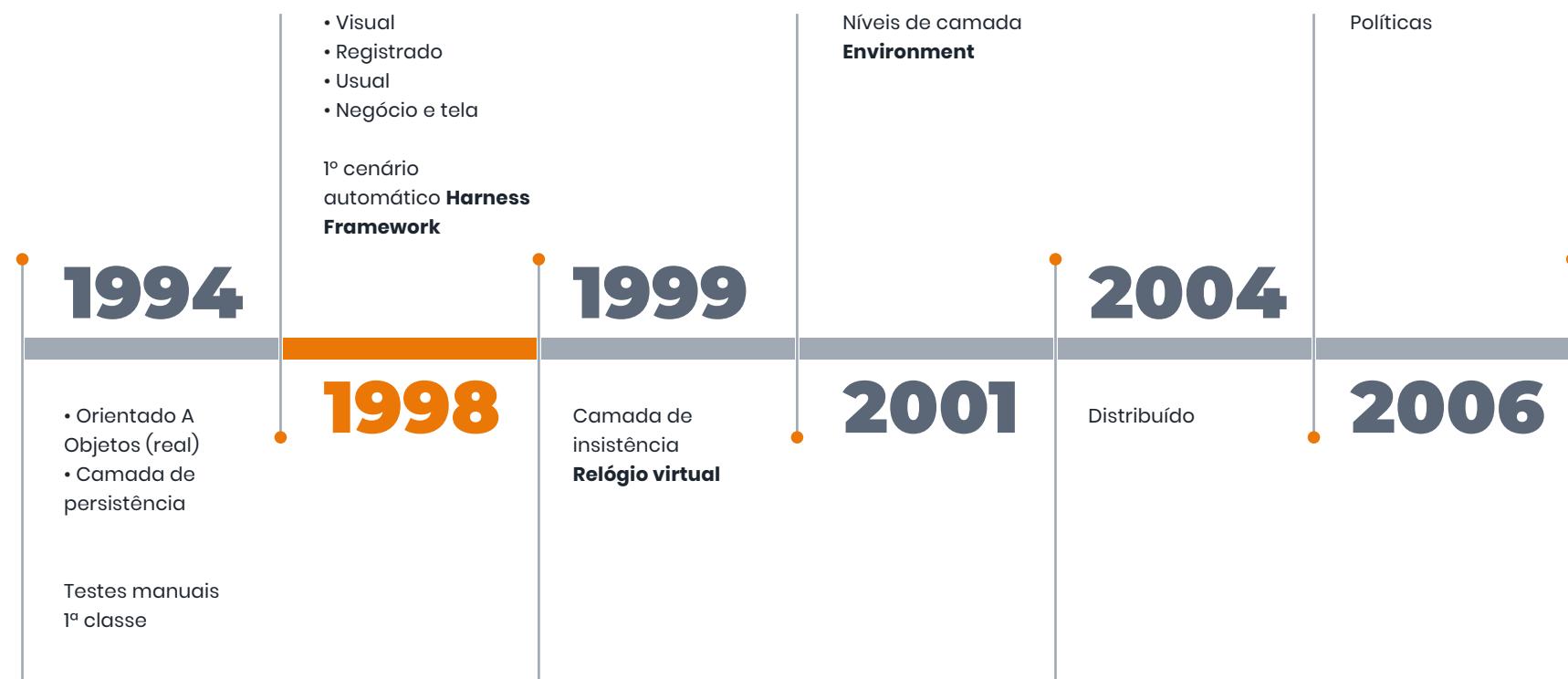
Nossa análise é realizada por meio de nossa expertise e referência nacional em metodologias ágeis, proporcionando um planejamento estratégico e acompanhando a execução das ações sugeridas para acelerar o tempo do ciclo com base em maiores áreas de oportunidade, impacto nos negócios e velocidade de implantação.

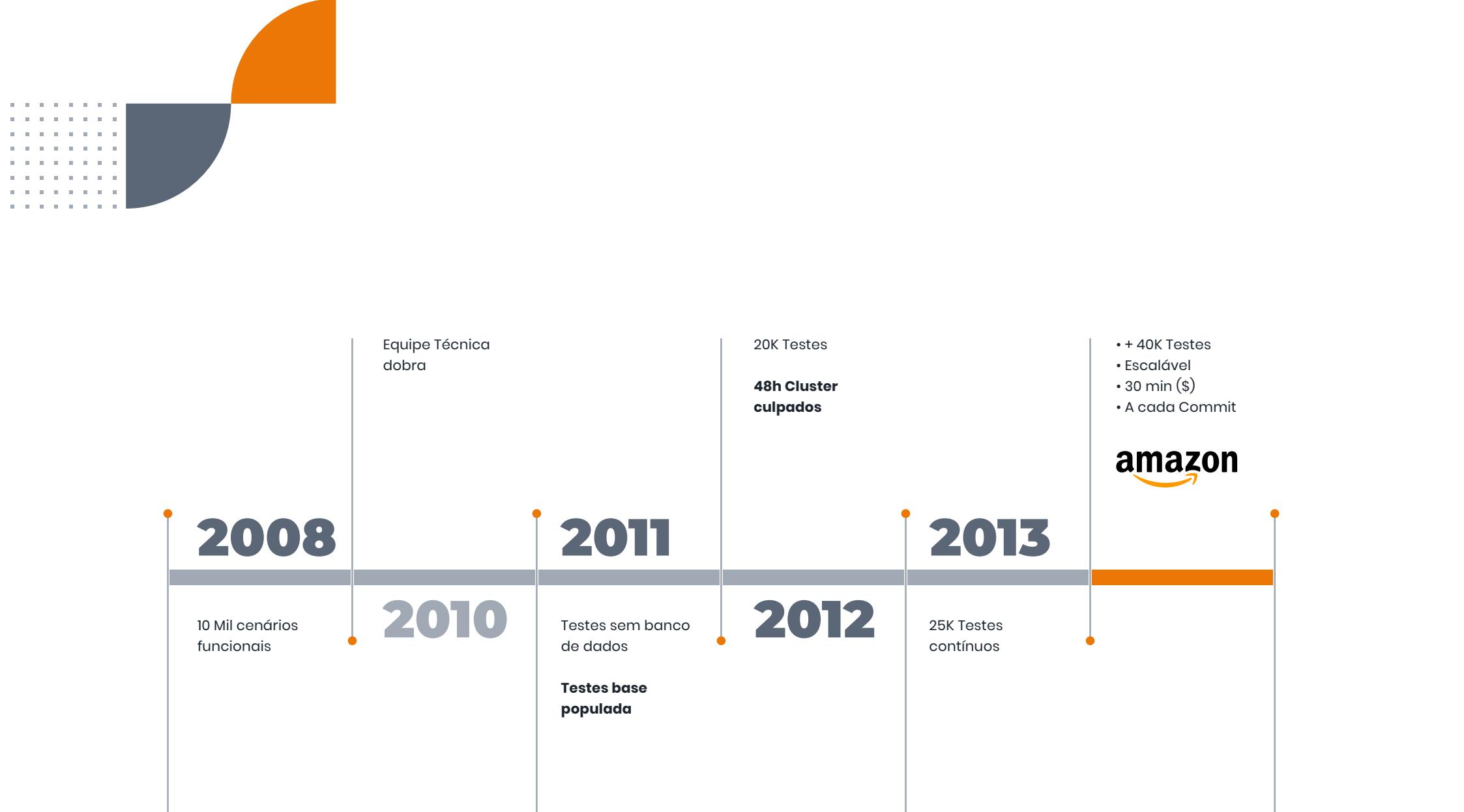
Nossa cultura de desenvolvimento ágil é estruturada através do conceito DevOps, que reúne profissionais com conhecimentos em Desenvolvimento, Operações e Infraestrutura, para facilitar a interação entre os colaboradores do Centro de Desenvolvimento e aumentar a entrega de valor do projeto.

Entendemos que cada projeto é único e as ferramentas que se aplicaram ao último podem não se aplicar ao próximo. Para cada projeto repensamos nosso modo de trabalho. Entretanto, mesmo com exigência de maior prescritividade, restrições de escopo, tempo e custo, utilizamos práticas ágeis de desenvolvimento.

Com a observação de comportamentos, análise de indicadores de performance e qualidade, identificamos problemas e pontos de otimização para então atuarmos de maneira responsável, precisa e rápida. Desta forma, viabilizamos um bom desempenho para nossos projetos em todo o ciclo de desenvolvimento.

Para você entender e conhecer mais sobre a expertise da Objective em DevOps, apresentamos uma linha do tempo com nossa trajetória e evolução. Foram anos de experiência e madureimento para hoje sermos referência no mercado





**SÃO PAULO | SP**

Rua Peixoto Gomide, 996 - 6º andar
Cerqueira César - CEP 01409-000
Tel.: (55 11) 3139-1900
comercial@objective.com.br

CURITIBA | PR

Av. João Gualberto, 1740 - 9º andar
Juvevê - CEP 80030-001
Tel.: (55 41) 3122-9100

MARINGÁ | PR

Av. Horácio Raccanelo Filho, 5355
SI 1 - Zona 7 - CEP 87020-035
Tel.: (55 44) 3032-9150

CHICAGO | USA

222 Merchandise Mart Plaza - Suite 1225
Chicago - Illinois - 60654
Tel.: (1 312) 646-5077
sales@objectivegroup.com