

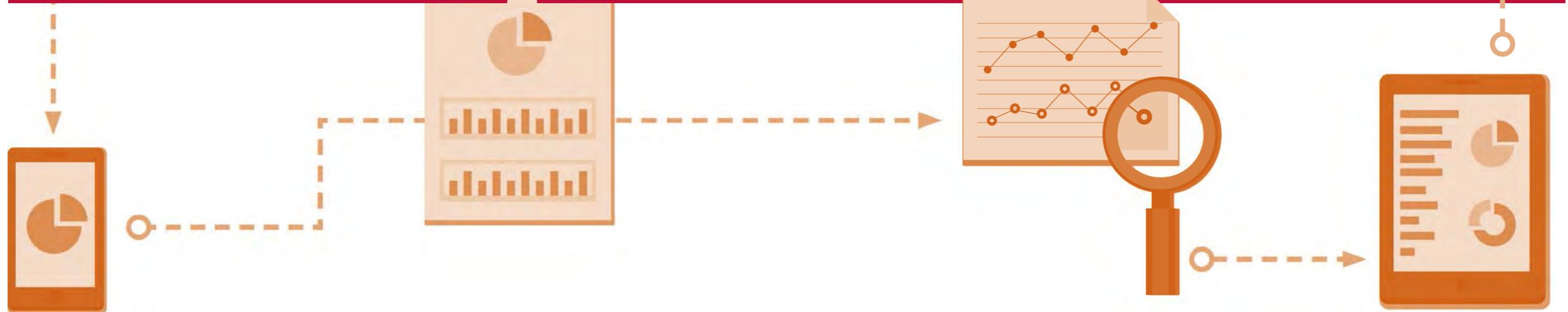
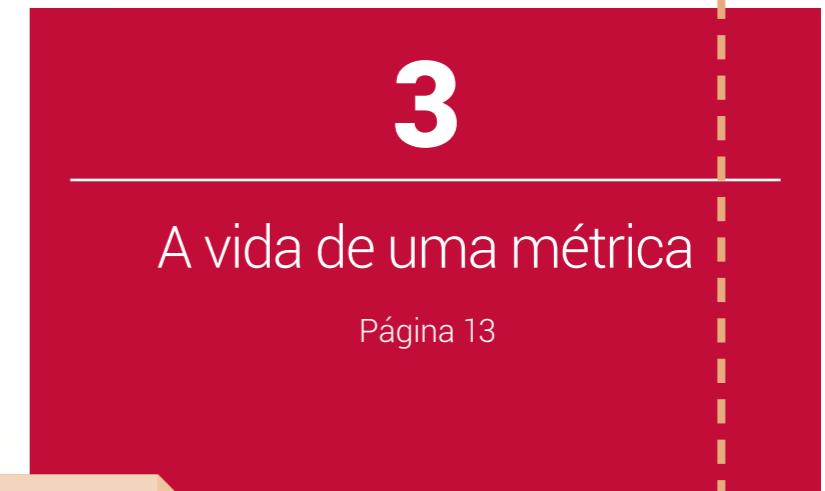
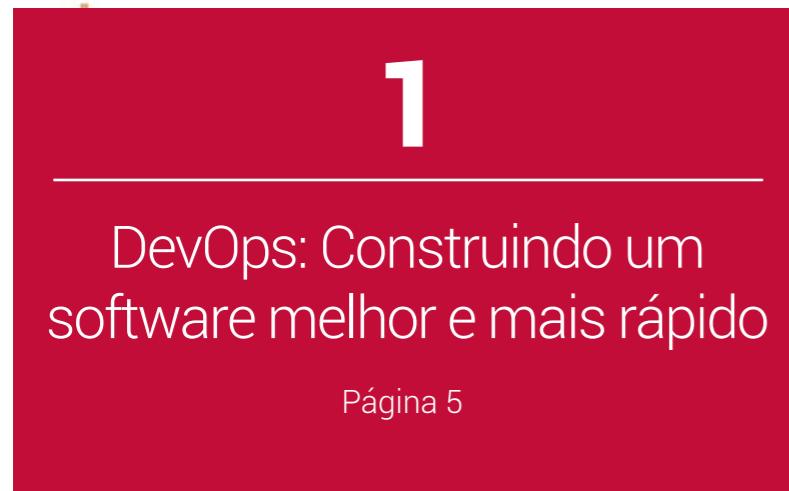


# 7 métricas de performance para lançar um software melhor e mais rápido

---

Com exemplos reais de falhas de aplicações  
e como evitá-las

# Índice



# Introdução

Os consumidores móveis hiperconectados de hoje estão levando a uma transformação através das indústrias.

Eles têm grandes expectativas e para competir você deve fornecer uma experiência digital responsiva e superior. As práticas de DevOps recomendadas de Entrega Contínua tornam isso possível, mas para fazer bem isso, você vai precisar de uma rede de segurança. É sobre isso que as métricas orientadas para gestão de performance de aplicação são. Elas permitem que você implante mais rápido sem falhar – com métricas importantes de arquitetura, escalabilidade e desempenho, dando a você controle total sobre a conduta de entrega de software.

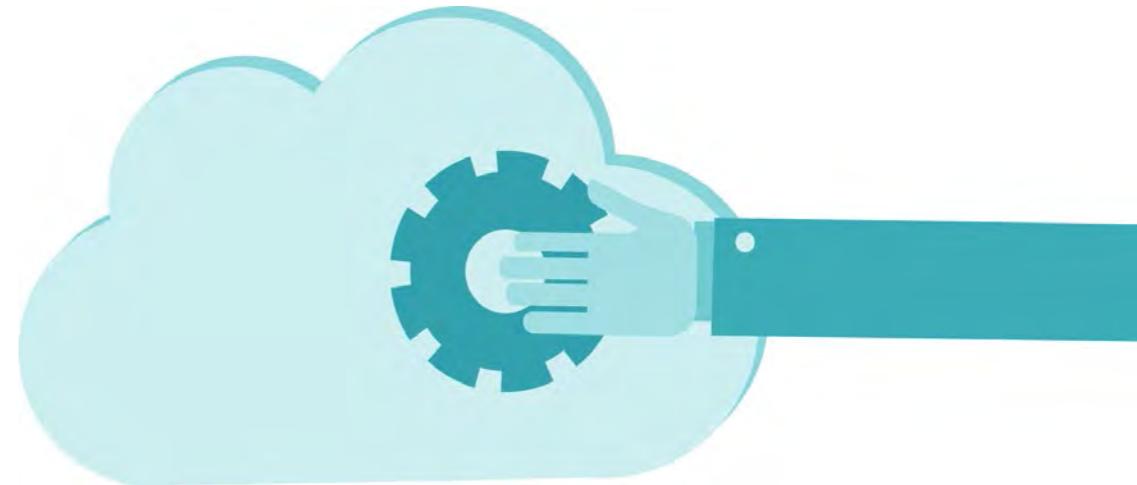
Neste e-book, vamos dizer como começar com monitoramento de métricas de performance. Vamos compartilhar exemplos reais de falhas de aplicação e como evitá-las usando as 7 principais métricas para encontrar problemas desde o início. Então vamos mostrar a você como realmente funciona em "a vida de uma métrica" e como integrar métricas de desempenho em ferramentas de automação em toda sua cadeia de entrega da aplicação.

Feliz monitoramento!



# Nossos autores:

**Dois caras que comem, bebem e respiram DevOps**



**Andreas Grabner**

Defensor de Performance,  
Centro de Excelência,  
Dynatrace

[Blog](#): [blog.dynatrace.com](http://blog.dynatrace.com)  
[Twitter](#): [@grabnerandi](https://twitter.com/grabnerandi)

Andreas Grabner tem mais de 15 anos de experiência como desenvolvedor e arquiteto no mercado .NET e JAVA e é um defensor para aplicações de alta performance. É colaborador regular de grandes comunidades de performance e palestrante frequente em conferências de tecnologia.



**Brett Hofer**

Arquiteto Sênior,  
Dynatrace

[Blog](#): [blog.dynatrace.com](http://blog.dynatrace.com)  
[Twitter](#): [@brett\\_solarach](https://twitter.com/brett_solarach)

Brett Hofer é apaixonado por DevOps e especializado no fornecimento de software complexo de missão crítica. Com mais de vinte anos de experiência - desde designer de produto e arquiteto de soluções até gerenciamento sênior - ele tem uma perspectiva de 360º única de TI.

## Seção 1

---

# DevOps: Construindo um software melhor e mais rápido

# DevOps: Construindo um software melhor e mais rápido

## Uma criança de 4 anos com um iPad, o Uber chinês e um unicórnio

O que uma criança de 4 anos com um iPad, um Uber chinês (a.k.a. Didi Kuaidi) e um unicórnio têm em comum? Eles estão nos forçando a construir um software melhor e mais rápido.

Há uma revolução digital acontecendo em todos os setores, impulsionado pelos consumidores móveis hiperconectados de hoje. Eles possuem grandes expectativas - eles querem o que eles querem, quando eles querem e dentro de seu contexto (com base em localização, tarefas, preferências, etc.).

Seus dispositivos móveis são responsáveis por mais tráfego e receitas do que nunca:

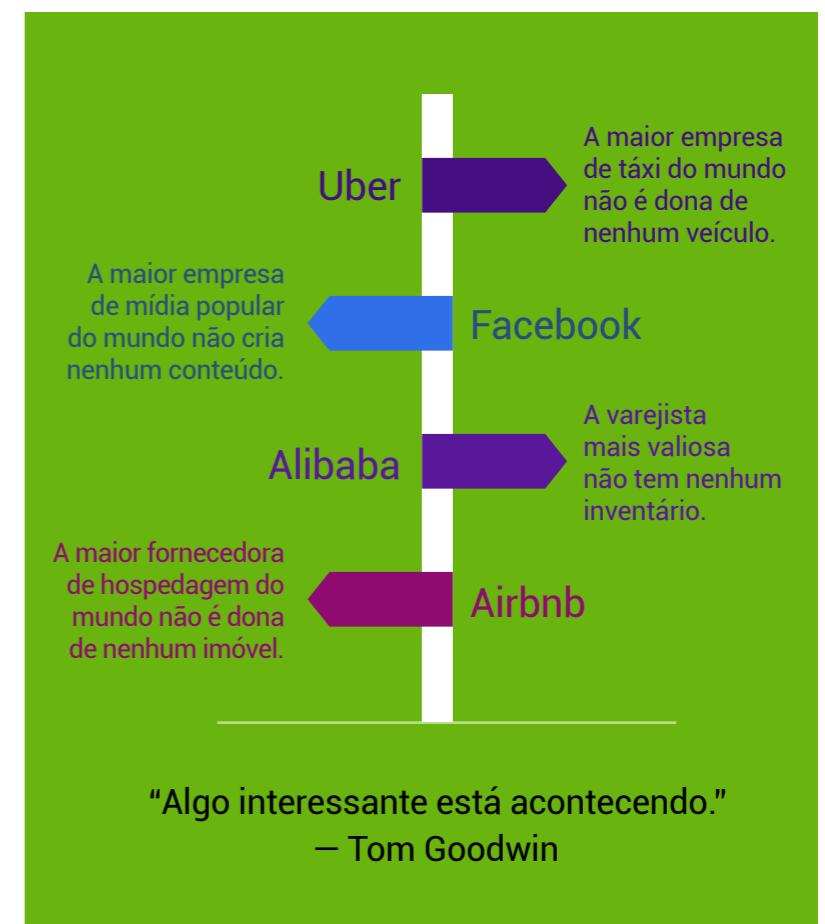
- > **No Walmart na Cyber Monday de 2014, 70% do tráfego foi através de dispositivos móveis.**
- > **A Visa prevê que metade de todos os pagamentos serão por dispositivos móveis em 2020.**
- > **Em 2014, as sessões de compras em aplicativos (iOS e Android) cresceu 174% YOY.<sup>1</sup>**

E o unicórnio em tudo isso? Os gigantes da tecnologia como Facebook, Google, Etsy, e Pinterest. Eles estão liderando o caminho ao fornecer experiências digitais ágeis e de qualidade superior.

Estes unicórnios também estão respondendo ao consumidor com modelos completamente novos de negócios. O Uber, a empresa mundial de taxi, não possui veículo próprio e o Facebook, proprietário da mídia mais popular do mundo, não cria nenhum conteúdo.

Na Ásia, esta revolução digital é amplificada. O Uber gera um grande número de corridas de táxi – 1 milhão. Mas e o equivalente na China? Gera 5 milhões.<sup>2</sup>

Como você está respondendo a estas rápidas alterações de mercados, exigências e expectativas do usuário? Você tem os processos e métricas certos no local para dar suporte a entrega contínua e construir uma experiência digital melhor e mais rápida – ao mesmo tempo que mantém todos os seus clientes felizes?



# DevOps: Construindo um software melhor e mais rápido

## Como implantar mais rápido e não falhar tão rápido assim

DevOps é sobre como responder às mudanças na dinâmica do mercado para obter ideias aos clientes finais o mais rápido possível, minimizando o ciclo do tempo da feature. Os unicórnios de DevOps não são só o Facebook e o Google.

Eles são organizações de serviços financeiros e varejistas também:<sup>3</sup>

- > A CapitalOne faz 1 implantação por mês para cada uma das suas 200 aplicações.
- > A IG diz, "Todos podem fazer entrega contínua."
- > A Target, com 30 APIs e mais de 500M de visitantes mensais, tem 80 implantações por semana e menos de 10 incidentes por mês.

Você pode estar se perguntando sobre resultados além dessas empresas individuais. DevOps realmente compensa? Bem, usuários de DevOps de alta performance são:

- > Mais ágeis, com implantações 30x mais frequentes e 8.000x mais rápidos nos prazos de entrega do que seus pares.
- > Mais confiáveis, com 2x de taxa de sucesso na mudança e um tempo de recuperação 12x mais rápido.



## Seção 2

---

# Métricas: Eu ❤️ aprender com os outros

# Métricas: Eu ❤️ aprender com os outros

Não force sem um plano

Se as suas aplicações não estão preparadas e são lentas para carregar (ou falham), a sua campanha de marketing mais inteligente é realmente a sua pior campanha.

## Metrics

- ① Número de recursos
- ② Tamanho dos recursos
- ③ Tamanho da página



## Estudo de caso 1: 400 selfies

Aqui está um exemplo de uma grande campanha de marketing com implementação escassa. Um dos anunciantes do Superbowl pediu aos consumidores para tirar *selfies* e em seguida, enviá-los para o site móvel. O plano era um anúncio a ser executado durante o Superbowl, direcionando o tráfego para o site para ver os últimos 400 *selfies* carregados.

Infelizmente, ele foi implementado como uma grade de imagem 20x20 com arquivos de imagem individuais servidos no mesmo domínio de imagem. Quando nós testamos com um iPhone, tivemos que baixar um conteúdo colossal de **20 MB e 434 recursos – ignorando completamente as práticas recomendadas de performance da web**.

## Estudo de caso 2: O favicon que foi 150x maior

Durante outro grande evento esportivo, o maior elemento na página do website móvel do organizador foi um favicon com tamanho de 370 kilobytes. Este é o ícone que geralmente está no canto superior esquerdo da janela do navegador e normalmente tem 16x16 ou 32x32 pixels.

Eles pegaram o logo de grande resolução e converteram direto para um ícone e colocaram no site

**Deveria ser um par de bytes  
e não 370 kilobytes. Alguém  
não estava pensando.**

Estas duas instâncias necessitavam de otimização básica de performance da web e testes da máquina do desenvolvedor em toda a conduta de desenvolvimento - antes de ir ao vivo para o grande evento.

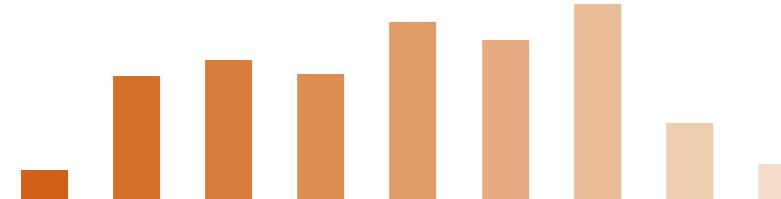
# Métricas: Eu ❤️ aprender com os outros

Não presuma que você conhece o ambiente

Suas aplicações executam na borda da Internet, através de uma variedade de dispositivos (smartphones, tablets, navegadores) através da rede principal de provedores de serviços e terceiros. Existem muitos pontos de falha potencial, então não presuma que você conhece o ambiente sem a percepção real do usuário.

## Metrics

- ④ Número de erros funcionais
- ⑤ Chamadas de terceiros



7 métricas de performance para lançar um software melhor e mais rápido

© 2016 Dynatrace

## 480 km/hora - um recorde mundial!

O Runtastic é um app móvel social que acompanha seu treino e permite que você coloque-o no Facebook para mostrar aos seus amigos. Nesse caso, alguém pedalou durante uma hora e o app informou que ele pedalou 480 km. Que é 480 km/hora – um recorde mundial!

Quando isso acontece, as pessoas estão menos propensas a usar o seu app e vão no Twitter, Facebook, ou na app store e dizem: "Isto é uma % s #&! O app não funcional! Não compre!" Os engenheiros, neste caso, tinham um plano.

Eles sabiam que não podiam testar todas as combinações do sistema operacional Android em todos os dispositivos, então:

- > Desenvolveram um teste de unidade para verificar o cálculo da distância do GPS cada vez que o aplicativo implantado iniciava.
- > Então, usando o monitoramento do usuário real cada vez que o app executava em milhões de instalações, descobriram que o problema ocorre apenas em uma certa configuração de hardware do Android.



- > Uma pesquisa no Google disse que era um problema bem conhecido com a configuração de hardware/sistema operacional e fora do controle do Runtastic.
- > Eles contaram seus usuários sobre o problema e sugeriram que era hora de atualizar a sua versão do Android, usando canais como Twitter ou Facebook e um feed de notícias embutido no aplicativo.

Os usuários do Runtastic também têm problemas, às vezes, ao fazer o carregamento de seus resultados no Facebook e Twitter. Ao monitorarativamente, o Runtastic aponta a 3º fonte do problema e diz aos usuários para tentar novamente.

# Métricas: Eu ❤️ aprender com os outros

## Não (re) use cegamente seus componentes

Sua equipe de engenharia é engenhosa. Mesmo que não saibam como fazer algo imediatamente, eles vão dar um jeito. Isso nem sempre funciona a seu favor e é por isso que você precisa acompanhar de perto antes que as coisas cheguem à produção.

### Métricas

⑥ Número de execuções SQL

⑦ Número dos mesmos SQL's



Os desenvolvedores estão sempre reutilizando componentes existentes. O próximo exemplo realmente aconteceu conosco na Dynatrace. Precisávamos de um relatório HTML simples em nossos registros de teste gratuitos. Ele não tinha muita experiência na criação de páginas HTML ou relatórios voltados ao cliente, então ele fez o que os desenvolvedores fazem e pesquisou no Google. Nossa desenvolvedor de retaguarda encontrou o código de exemplo no GitHub e inseriu.

O relatório funcionou bem no início, mas depois abrandou ao longo de alguns meses a um rastreamento - levava minutos para processar.

Procuramos nos bastidores e descobrimos que a amostra que ele reutilizava era baseada em Hibernate — uma Estrutura de Mapeamento de Objeto Relacional (OR) muito popular para acessar dados no banco de dados. O código não estava utilizando qualquer um dos recursos avançados de otimização do Hibernate para melhor acesso de banco de dados. Neste caso, Hibernate estava executando 4K de confirmação do banco de dados para relatar três linhas na saída do relatório HTML.



### Mais métricas

Estas 7 métricas são apenas o começo. Uma maneira fácil de começar é controlá-las manualmente - em toda a sua conduta de desenvolvimento de aplicações. Uma vez que você tem um controle sobre o que você precisa saber, é hora de começar a olhar para a forma de automatizar o monitoramento da performance. Este é o objetivo da entrega contínua: automatizar a sua conduta de desenvolvimento com portais de qualidade com base em métricas em cada etapa.

Aqui estão mais algumas métricas a considerar:

- **Tempo gasto em API**
- **Chamadas em API**
- **nº de domínios**
- **Tamanho total**
- **nº de itens por página**
- **nº AJAX por página**

Para mais métricas, veja a nossa série sobre [métricas de qualidade de software](#) no blog.

## Seção 3

---

# A vida de uma métrica

# A vida de uma métrica

Você deve se perguntar: E agora, como isso realmente funciona?

Como e quando eu recebo essas métricas de performance? Para responder a esta pergunta, vamos levá-lo através da vida de uma métrica - neste caso, número de execuções SQL - com algumas etapas do ciclo de vida de entrega contínua.

**1 IDE/Eclipse Desenvolvedor:** A estação de trabalho do desenvolvedor é o lugar mais subutilizado para monitoramento de desempenho. É muito simples. Se os seus desenvolvedores são capazes de fazer a análise antes de verificar no código, eles podem impedir problemas no início do ciclo de vida e reduzir os custos totais.

Neste exemplo, ferramentas de monitoramento de performance serão integrados no Eclipse. Os máximos são definidos para métricas de desempenho, incluindo nossa métrica – nº de execuções SQL. Então, toda vez que o desenvolvedor executa uma chamada de SQL, o monitoramento de desempenho irá rastreá-la.



# A vida de uma métrica

Você deve se perguntar: E agora, como isso realmente funciona?



- 2 **Teste de Unidade/JUnit:** Nessa próxima etapa, os autores desenvolveram o clássico caso de teste JUnit. O monitoramento de performance é configurado para assistir casos de testes e auxiliar cada métrica pré-definida por caso de teste. É onde o desenvolvedor avaliará o número de execuções de SQL por teste.
- 3 **Automação de Compilação/Maven:** O desenvolvedor então irá executar um teste MAVEN com código local e usa o monitoramento de performance para avaliar as pilhas de chamada e o número de execuções SQL. Neste momento, o arquiteto pode fazer uma análise rápida das métricas e validação de arquitetura, em vez de uma longa revisão de código. Então o desenvolvedor decide se o código está pronto para ser promovido para o sistema de gestão de configuração.

- 4 **Servidor de compilação de CI/Jenkins:** Se tudo estiver de acordo com as etapas acima, o desenvolvedor verifica no código. O servidor de compilação Jenkins derruba a mudança e executa os casos de teste. Os alertas de monitoramento de performance podem ser acionados. O Jenkins vai para o servidor de monitoramento de desempenho e coleta os dados sobre o que aconteceu, incluindo o número de execuções SQL. As decisões de implantação são feitas "Ir" ou "Não ir" com emissão de relatórios de exceção.

O resultado? Confiança funcional e arquitetônica antes da auto implantação.

# Seus próximos passos

Sabemos que você está enfrentando algumas sérias mudanças no mercado, nas exigências e expectativas do usuário. Os usuários querem uma experiência digital responsiva e querem agora. A entrega contínua é a resposta — se você construir uma rede de performance segura. Esperamos que estas métricas forneçam um bom ponto de partida. Ir em frente para automatizar o máximo possível e continuar concentrando-se no mais rápido e melhor!

Abaixo estão dos nossos alguns recursos adicionais que irão ajudá-lo em sua jornada.

## Ferramentas de DevOps que adoramos

Aqui está uma coleção de ferramentas que promovem a colaboração entre as equipes de Gestão de Produto, Desenvolvimento, Operações de TI e Suporte Técnico que permitem que você construa mais qualidade aos seus produtos e fornecem suporte na criação de loops de feedback melhores.

- > Controles de Mudança – JIRA Agile
- > Desenvolvimento – Eclipse
- > Controle de Fonte – Git/GitHub
- > Automação de compilação – Ant e Maven
- > Gestão de Configuração – Ansible, Chef e Puppet
- > Automação de teste – LoadRunner e Selenium
- > Tecnologias de VM, Nuvem e Recipientes: – Vagrant, Packer, VeeWee, Docker e Cloud Foundry

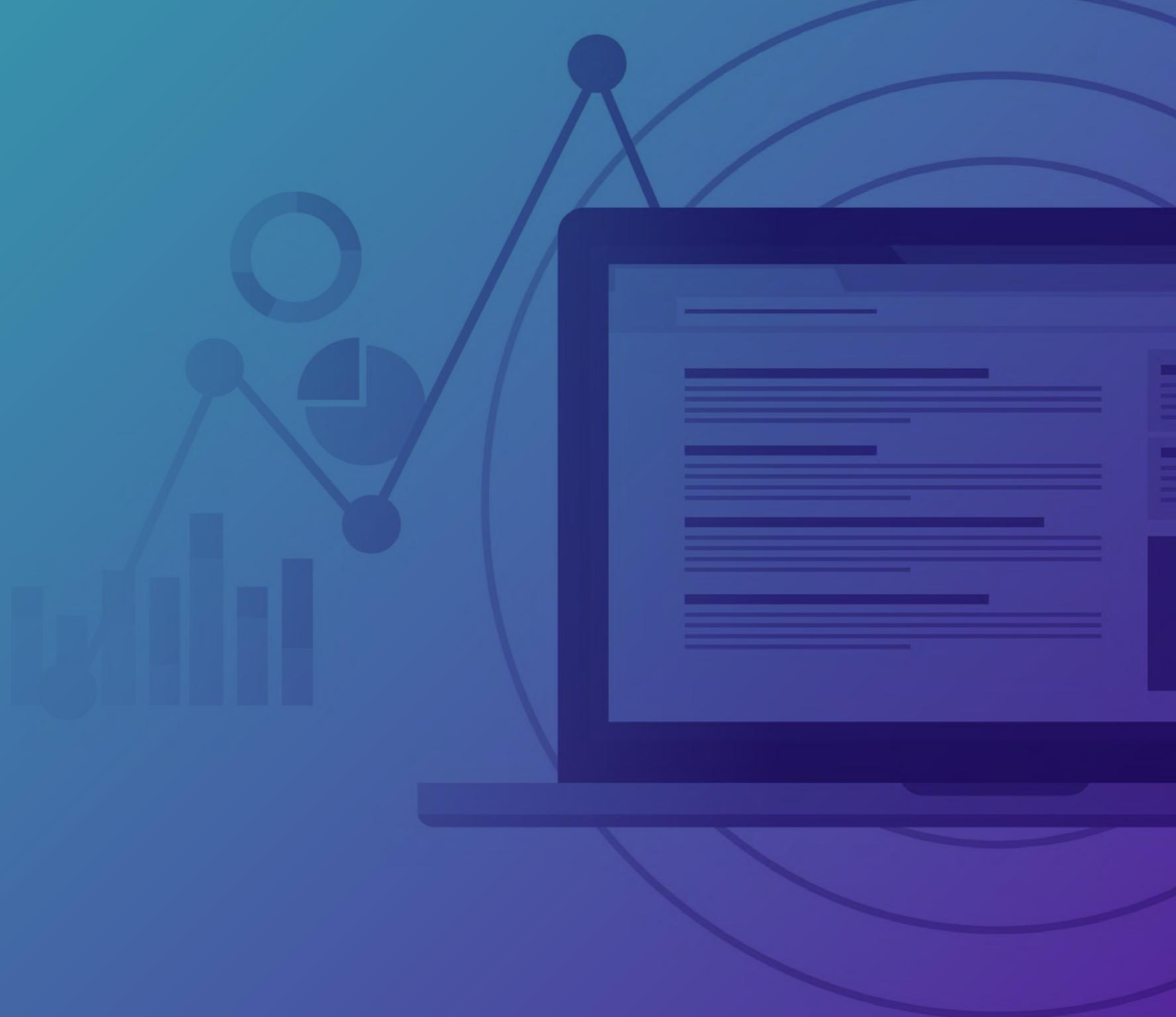
- ▶ **Reapresentação do Webinar**
  - > [5 Métricas Principais para Lançar um Software Melhor e Mais Rápido](#)
  - > [DevOps: Da Adoção ao Desempenho](#)

## Os blogs favoritos

- > [Reações DevOps – Curta um pouco do humor DevOps!](#)
- > [Dynatrace APM em DevOps](#)
- > [O DevOps na Revolução de TI](#)
- > [Métricas de Qualidade de Software para Sua Entrega Contínua](#)

## Leitura recomendada

- > [Entrega Contínua](#) por Jez Humble e David Farley



Saiba mais em [dynatrace.com](https://www.dynatrace.com)

A Dynatrace é líder mundial em soluções de Gerenciamento de Performance Digital. A empresa oferece plataforma inovadora que apoia empresas de todos os setores na transformação de seus negócios e no desempenho digital, tornando visíveis e úteis as informações em tempo real. A Dynatrace ajuda seus clientes de todos os tamanhos a enxergar suas aplicações e canais digitais pelo ponto de vista dos usuários finais. Mais de 8.000 organizações utilizam nossa percepção para dominar a complexidade, obter agilidade operacional e aumentar a receita ao oferecer experiências com excelência.