

# Manual do proprietário de APIs

As sete práticas recomendadas para equipes de API bem-sucedidas

## **Manfred Bortenschlager**

Diretor de desenvolvimento de negócios de soluções de integração baseadas em API e gerenciamento de API, Red Hat

## **Andrew Mackenzie**

Diretor de engenharia de software e gerenciamento de API, Red Hat

## **Jaya Baskaran**

Gerente-chefe de marketing técnico, Red Hat

## **Greg Pack**

Gerente sênior de marketing de produtos, serviços de app e soluções de gerenciamento de APIs, Red Hat



# Sumário

<b>Resumo executivo</b> .....	<b>3</b>
<b>Introdução</b> .....	<b>3</b>
<b>Por que queremos implementar APIs?</b> .....	<b>4</b>
Quem esperamos que use essas APIs?.....	4
Que resultados tangíveis queremos alcançar com essas APIs? .....	4
Como planejamos executar o programa de APIs para alcançar esses resultados?.....	5
A equipe de API .....	5
<b>Prática recomendada 1: concentre-se incansavelmente no valor da API</b> .....	<b>6</b>
O que isso significa nos termos da API? .....	6
Considerações sobre o valor do programa da API .....	7
<b>Prática recomendada 2: defina o modelo de negócios logo no início</b> .....	<b>7</b>
Introdução ao modelo de negócios de API.....	7
Considerações sobre seu modelo de negócios de API.....	8
<b>Prática recomendada 3: projete e implemente tendo o usuário em mente</b> .....	<b>9</b>
Simplicidade .....	9
Flexibilidade.....	9
Considerações sobre design de protocolo de API.....	9
Integração com produtos .....	9
Infraestrutura de implantação e implementação de API.....	10
<b>Prática recomendada 4: priorize as operações de API</b> .....	<b>10</b>
Gráfico de operações de API.....	10
Componentes essenciais de gerenciamento de API.....	11
Considerações sobre operações de API .....	11
<b>Prática recomendada 5: sempre pense na experiência do desenvolvedor</b> .....	<b>12</b>
O sucesso da API é mais do que um ótimo design .....	12
Considerações sobre a avaliação da experiência do desenvolvedor .....	13
<b>Prática recomendada 6: vá além do marketing 101</b> .....	<b>13</b>
Considerações sobre APIs de marketing .....	13
<b>Prática recomendada 7: lembre-se da descontinuação das APIs e da gestão de mudanças</b> .....	<b>14</b>
Considerações sobre a longevidade da API .....	14
Como sustentar sua estratégia de API.....	15
Como desenvolver um programa de APIs duradouro .....	15
APIs não existem no vácuo.....	15
<b>Por que escolher a Red Hat para desenvolver e gerenciar suas APIs?</b> .....	<b>16</b>
<b>Conclusão</b> .....	<b>16</b>

## Resumo executivo

Os avanços no desenvolvimento de aplicações modernas, na conectividade e na infraestrutura, como o 5G e a edge computing, continuam redefinindo como e onde os processos de negócios acontecem. Embora esses avanços estejam acelerando os negócios, seus benefícios não seriam possíveis sem as interfaces de programação de aplicações (APIs) modernas.

Atualmente, as APIs são o tecido conjuntivo digital das empresas modernas, acrescentando novos recursos a tudo, desde operações e produtos até estratégias de parceria. Se você já tem uma estratégia de API ou está se preparando para lançar uma, nós criamos este ebook para oferecer a você e à sua equipe sete práticas recomendadas para programas de APIs bem-sucedidos.

## Introdução

**Antes de ler este ebook, considere os principais objetivos do seu programa de APIs. Algumas das perguntas feitas posteriormente podem ajudar a mudar, validar ou fundamentar seus objetivos.**

Um programa eficaz de API deve basear-se na estratégia corporativa abrangente da empresa e contribuir para alcançar seus objetivos. Você saberá que cumpre os requisitos para uma ótima estratégia quando conseguir responder às quatro perguntas a seguir:

1. Por que queremos implementar APIs?
2. Quem esperamos que use essas APIs?
3. Que resultados tangíveis queremos alcançar com essas APIs?
4. Como planejamos executar o programa de APIs para alcançar esses resultados?

## Por que queremos implementar APIs?

É essencial compreender de forma abrangente o valor e a finalidade das APIs ao considerá-las. Evitar equívocos comuns é fundamental para maximizar seu potencial valor comercial. Um equívoco comum é que as APIs são importantes somente quando os usuários pagam por elas, o que é verdadeiro quando a API é o produto. No entanto, muitos projetos de API permanecem internos, gerando diversas métricas, como velocidade de comercialização, reutilização, vendas, reconhecimento de marca ou indicações de afiliados.

Cinco casos de uso típicos de provedores de API incluem:

1. **Escalonamento omnichannel:** amplie a acessibilidade por meio de diversos canais, como aplicativos mobile, Internet das Coisas (IoT) etc.
2. **Aumento do ecossistema:** promova ecossistemas de parceiros ou clientes para expandi-los.
3. **Ampliação do seu alcance:** estabeleça amplas redes de transações ou de distribuição de conteúdo.
4. **Impulsão de novos modelos de negócios:** gere inovação por meio de novos modelos de negócios.
5. **Geração de inovação interna:** gere inovação na empresa.

Há muitos exemplos de estratégias de API bem-sucedidas em empresas modernas de tecnologia, como a Amazon, Salesforce, Twilio, entre outras. Mas isso não significa que apenas essas empresas líderes no setor de tecnologia possam usar APIs para monetização, inovação, crescimento de parcerias ou agilidade. Muitas empresas consolidadas no mercado utilizam vários sistemas tradicionais que não desaparecerão de um dia para o outro (provavelmente, nunca desaparecerão). As APIs são a conexão perfeita para fazer integração com esses sistemas tradicionais, realizar sua exposição interna ou externamente e garantir sua combinação com outros serviços para criar um novo valor. Qualquer empresa que esteja em processo de transformação digital pode aproveitar as vantagens da economia da API ao implementar uma estratégia de API bem estruturada.

Os motivos precisam ser sólidos o suficiente para que o investimento em APIs seja uma escolha óbvia para a empresa.

### Quem esperamos que use essas APIs?

Saber quem são os usuários finais da API ajudará a definir as métricas de sucesso do seu programa de APIs.

- **Usuários finais internos:** as APIs podem ser usadas por equipes internas para acessar informações da empresa para quaisquer objetivos de negócio.
- **Usuários finais externos:** as mesmas APIs ou um subconjunto delas podem ser usados por desenvolvedores externos de terceiros em suas aplicações.

Os motivos precisam ser sólidos o suficiente para que o investimento em APIs seja uma escolha óbvia para a empresa.

### Que resultados tangíveis queremos alcançar com essas APIs?

Para identificar resultados concretos gerados pelas APIs, considere as visões internas e externas da empresa:

- **Visão interna:** use seus ativos exclusivos e importantes. Empresas que coletam dados distintos, como os dados de "curtidas" da Meta, podem oferecer APIs muito úteis.
- **Visão externa:** considere a dinâmica do mercado, as tendências, os concorrentes e os comportamentos dos clientes. Forças externas moldam as estratégias empresariais e influenciam as funcionalidades da API.

A dinâmica do mercado afeta profundamente as APIs de mapeamento. As primeiras empresas de mapeamento ignoraram as demandas em tempo real, ajudando startups como o Waze a crescer. O Google adquiriu o Waze e integrou sua tecnologia em uma API bem-sucedida. O Twitter, o Reddit e outros gigantes também adotaram APIs e garantiram inovação e crescimento do ecossistema.

A estratégia certa de API geralmente combina ativos internos com insights do mercado externo.

## Como planejamos executar o programa de APIs para alcançar esses resultados?

A última pergunta, "Como desenvolvemos o programa de APIs para alcançar o que queremos?", refere-se à implementação e execução, que exige um planejamento detalhado de:

1. **Opções tecnológicas:** determine o stack de tecnologia para desenvolver APIs.
2. **Design e manutenção:** elabore estratégias de design e manutenção de API.
3. **Promoção e marketing:** promova as APIs dentro da empresa e comercialize-as externamente.
4. **Alocação de recursos:** aloque os recursos necessários para o desenvolvimento e a manutenção de APIs.
5. **Composição da equipe:** monte uma equipe capacitada para desenvolver e gerenciar APIs.
6. **Comunidade de desenvolvedores:** crie e mantenha um plano de comunicação dedicado para desenvolvedores internos e externos.
7. **Métricas de sucesso:** estabeleça métodos para acompanhar o sucesso em relação aos objetivos de negócios.

Ao descobrir "por que", "quem", "o que" e "como" do uso de APIs, as empresas podem impulsionar a inovação e o crescimento da economia de API, que está em constante evolução. As respostas a essas perguntas serão diferentes para cada empresa e serão influenciadas por suas metas, pela estratégia implantada e também, o que é de extrema importância, pela equipe de API.

## Equipe de API

Normalmente, uma equipe de API é estruturada como qualquer outra equipe de soluções da empresa. Sejam seus clientes internos ou externos, a equipe de API é responsável por criar, implantar, operar e otimizar a infraestrutura da qual os outros dependem.

Assim como as equipes de produtos, as equipes de API também podem ser diversificadas. Elas devem incluir um responsável pelo produto que atue como guardião da estratégia e das metas, profissionais centrados no design de APIs que garanta a aplicação de práticas recomendadas, engenheiros que codifiquem a tecnologia da API e funcionários operacionais que, por fim, executem a API. Ao longo do tempo, também será possível incluir outras pessoas, como membros da equipe da comunidade e de suporte, especialistas em APIs e representantes de segurança. Sua equipe de API estendida também pode incluir membros da sua comunidade de desenvolvedores.

Embora essa equipe possa ter muitos integrantes, alguns deles podem desempenhar várias funções, principalmente em empresas de pequeno porte. É importante garantir que as opiniões de todos os stakeholders sejam representadas, mesmo que seja preciso pedir a um membro da equipe que verifique as preocupações.

Em muitos casos, as equipes de API são temporárias e podem pertencer a diferentes unidades organizacionais com diferentes superiores hierárquicos. Essa estrutura pode dificultar a definição de uma visão compartilhada para a API. Em caso de grandes programas de APIs, pode ser necessário colaborar com diferentes equipes de API.

Seja qual for o porte da sua empresa, as sete práticas recomendadas descritas neste ebook ajudarão a formar uma equipe de API bem-sucedida, que pode acabar incluindo mais pessoas do que você imagina.

# Prática recomendada 1: concentre-se incansavelmente no valor da API

Os programas de APIs devem priorizar sua principal meta de agregar valor evitando complexidade. A proposição de valor determina a utilidade do usuário, um fator crucial para o sucesso da API. Uma proposição de valor convincente é essencial para fazer um marketing eficiente. Alinhá-la com os objetivos corporativos garante sustentabilidade, permitindo que empresas estabelecidas aprimorem suas ofertas por meio de APIs.

O modelo de valor de API de Alex Osterwalder alinha os benefícios do usuário com as funcionalidades da API, criando uma "adequação" fundamental entre as necessidades do usuário e o valor da API.<sup>1</sup> Atender às necessidades, aliviar os pontos problemáticos e gerar valor são fatores cruciais.

## O que isso significa em termos de API?

Nesse processo iterativo, a primeira etapa descreve as tarefas que os usuários estão tentando realizar, como o envio automático de comunicações urgentes em uma emergência, o backup de arquivos críticos e a amostragem de dados para detectar determinados eventos.

A segunda etapa requer a identificação de pontos problemáticos específicos que afetam os usuários antes, durante ou depois de tentarem realizar uma tarefa. Isso inclui a garantia do envio confiável de mensagens com muitas tentativas, a detecção de falhas, o gerenciamento de várias mensagens, a integração do sistema de mensagens com base em localização, a proteção da entrega de arquivos com uso mínimo de largura de banda e a correlação em tempo real de grandes quantidades de dados.

A terceira etapa da criação de um perfil de usuário envolve a definição de possíveis benefícios, como outros tipos de notificações, que criam oportunidades em vez de alerta contra ameaças, eliminando outros equipamentos de armazenamento se a confiabilidade for boa o suficiente e ativando automaticamente ações com base em eventos.

Passando para o mapa de valores, a funcionalidade, as características e os serviços da API devem ser mapeados, concentrando-se na resolução de problemas e geração de ganhos. O processo transforma-se em exemplos concretos, como uma API de sistema de mensageria que garante a entrega de mensagens, uma API de sincronização de armazenamento para atualizações eficientes de versões e uma API de agregação de dados que fornece fluxos de dados configuráveis.

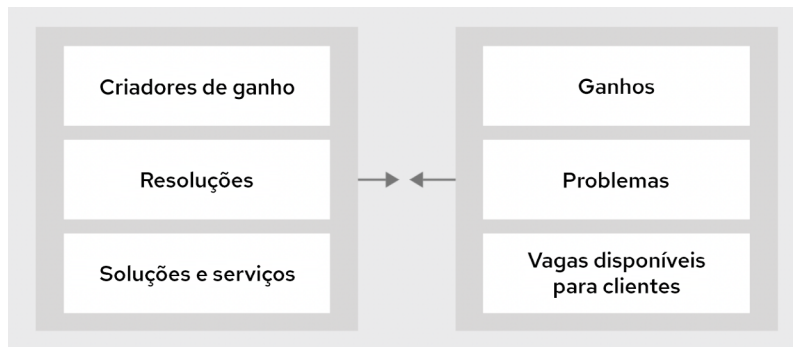


Figura 1. Painel da proposição de valor

Por fim, a equipe de API deve concluir um exercício de esclarecimento para compor diversas declarações que demonstrem a adequação entre a API e o perfil do usuário. Quando você conseguir consolidar e resumir suas declarações em uma única mensagem, terá definido a proposição de valor de suas APIs. Essas declarações de "adequação" solidificam ainda mais o alinhamento entre API e usuário, extraindo a proposição de valor. No caso do exemplo da API de sistema de mensageria, essa declaração seria mais ou menos assim:

*"A declaração do cliente vai aqui. Preste atenção à expansão do idioma ao adicionar conteúdo à barra lateral. Alguns idiomas, como o alemão, ocupam mais espaço do que o inglês. Para acomodar a tradução, deixe pelo menos 1/4 da barra lateral vazia."*

Você pode estar pensando: "Isso é um exagero, pois nossa API é apenas interna". Essa reação é natural, mas ter o valor como foco é essencial até mesmo nos casos de uso interno. Uma proposição de valor mal elaborada fará com que se gaste muito tempo mostrando e apresentando a API para outras equipes. Uma proposição de valor bem definida fará com que o programa de APIs seja de grande contribuição para a empresa.

<sup>1</sup> Strategyzer, "Value proposition canvas." Acessado em setembro de 2023.

## Considerações sobre o valor do programa de APIs

Para definir o valor do seu programa de APIs, considere estas cinco principais áreas:

1. **Identificação do usuário:** identifique os usuários com base em seus relacionamentos (clientes, parceiros e desenvolvedores), funções e preferências.
2. **Aborde pontos problemáticos e ganhos:** a referência ao painel da proposição de valor na Figura 1 determinará os pontos problemáticos, os ganhos e as necessidades essenciais do usuário. Avalie as melhorias das métricas (velocidade, receita e custo) e o potencial para novas oportunidades.
3. **Casos de uso compatíveis:** use o painel de proposição de valor para identificar resoluções de problemas e geradores de ganhos. Projete sua API de forma a atender a esses casos de uso específicos.
4. **Expansão do valor futuro:** planeje sua proposição de valor pensando no futuro. Preveja os próximos marcos, tendências ou inovações tecnológicas para o crescimento contínuo do valor.
5. **Valor organizacional interno:** avalie os benefícios internos da API e o valor potencial para outras equipes.

Ao responder a essas perguntas, seu programa de APIs pode estabelecer uma proposição de valor clara e garantir o alinhamento com as necessidades do usuário e os objetivos organizacionais.

## Prática recomendada 2: defina o modelo de negócios logo no início

A criação de uma API bem-sucedida vai além de uma proposição de valor apenas. É necessário fazer o emparelhamento da API com um modelo de negócios bem definido. Embora seja importante reconhecer e transmitir o valor da API, seus custos também devem ser equilibrados em relação aos seus benefícios financeiros ou tangíveis. Em seu livro "Business Model Generation", Alex Osterwalder define o modelo de negócios de uma empresa como a forma como a empresa propõe, cria, entrega e captura valor.

### Introdução ao modelo de negócios de API

O painel de modelo de negócios analisa com detalhes os principais componentes de um modelo de negócios, que incluem:<sup>2</sup>

1. **Proposição de valor:** define o valor exclusivo da API.
2. **Fluxos de receita:** descreve como a API gera receita.
3. **Estrutura de custos:** custos associados à manutenção da API.
4. **Segmentos de clientes:** grupos de usuários-alvo.
5. **Relacionamento com o cliente:** como a empresa se relaciona com os usuários.
6. **Canais:** métodos de distribuição para alcançar os usuários.
7. **Principais parceiros:** colaborações fundamentais para o sucesso da API.
8. **Principais atividades:** tarefas essenciais necessárias para a operação da API.
9. **Principais recursos:** ativos vitais necessários para a implementação da API.

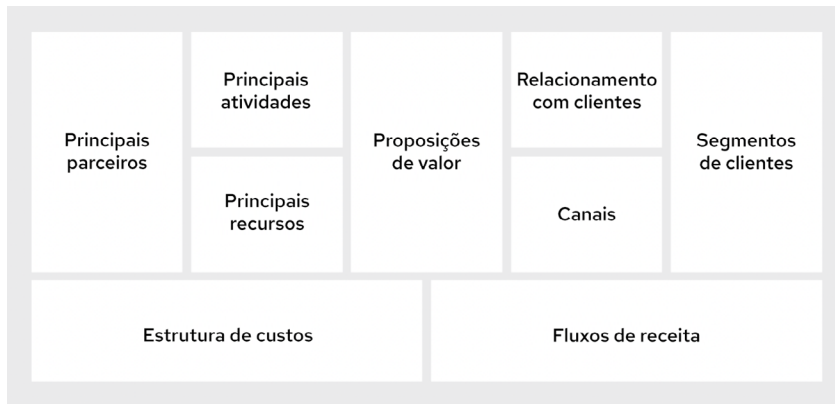


Figura 2. O painel do modelo de negócios

As APIs podem criar novas oportunidades, usando ativos existentes. No entanto, embora o reconhecimento do valor da API seja fundamental, seus custos devem ser gerenciados. Um modelo inadequado pode escalonar os custos, apesar do valor da API.

## Considerações sobre seu modelo de negócios de API

Para alinhar seu modelo de negócios com o uso da API, considere cinco áreas vitais, incluindo:

- 1. O valor da API para a empresa:** avalie diversos valores, não apenas monetários, descobrindo como a API ajuda a empresa a ampliar seu alcance ou inovação.
- 2. Captura de valor:** determine a melhor maneira de capturar o valor identificado, minimizando as barreiras para a realização do valor máximo.
- 3. Cobertura de custos:** reconheça os custos relacionados à API, geralmente além da equipe de API, como engenharia ou marketing, e planeje a cobertura deles.
- 4. Compromisso de longo prazo:** saiba que as APIs exigem um compromisso contínuo de operação e manutenção além do investimento inicial.
- 5. Parcerias estratégicas:** identifique parcerias essenciais e use ofertas complementares de parceiros e fornecedores durante o desenvolvimento de APIs e a entrada no mercado.



# Prática recomendada 3: projete e implemente API tendo o usuário em mente

O design de API inclui princípios fundamentais de design para uma experiência de usuário consistente. A meta é atingir um estado de design "pronto para o uso", em que os desenvolvedores experientes entendam intuitivamente as APIs. O design de API deve se concentrar em duas áreas principais: simplicidade e flexibilidade.

## Simplicidade

A simplicidade em design de API depende do contexto, com designs que variam em complexidade de acordo com os casos de uso. É essencial encontrar o equilíbrio certo na granularidade do método da API. A simplicidade pode ser considerada em diferentes níveis:

1. **Formato de dados:** escolha entre XML, JSON, formatos proprietários ou combinações.
2. **Estrutura do método:** os métodos variam de genéricos a altamente específicos, muitas vezes sequenciados para atingir casos de uso específicos.
3. **Modelo de dados:** o modelo de dados da API exposto pode ser diferente do modelo subjacente, o que afeta a usabilidade e a capacidade de manutenção.
4. **Autenticação:** diferentes mecanismos de autenticação têm pontos fortes e fracos e a escolha depende do contexto.
5. **Políticas de uso:** os direitos e as cotas do desenvolvedor devem ser compreensíveis e gerenciáveis.

## Flexibilidade

O equilíbrio entre simplicidade e flexibilidade é fundamental. Uma API simplista demais pode atender apenas a casos de uso específicos, limitando sua adaptabilidade. Para estabelecer a flexibilidade, descreva o espaço potencial de operações, incluindo sistemas subjacentes e modelos de dados. Defina um subconjunto viável e importante dessas operações. Para encontrar o equilíbrio certo entre simplicidade e flexibilidade, considere o seguinte:

1. **Exponha operações atômicas:** combine operações atômicas para cobrir todo o espaço operacional.
2. **Identifique casos de uso comuns e relevantes:** projete uma segunda camada de metaoperações que combine operações atômicas para atender a esses casos de uso.

## Considerações sobre design de protocolo de API

Embora o estilo arquitetônico de transferência representacional de estado (REST) continue sendo o padrão dominante para o desenvolvimento de APIs, os protocolos de APIs estão se tornando muito mais diversificados. Conceitos mais recentes incluem APIs de streaming ou WebSockets. As clássicas APIs de serviço web também não desaparecerão. A escolha do protocolo deve seguir o seguinte princípio: escolha o protocolo mais relevante para o usuário e, em seguida, encontre o equilíbrio certo entre simplicidade e flexibilidade. A infraestrutura para fornecer APIs REST continua evoluindo.

Para pensar no design de sua API, considere as seguintes áreas:

1. **Alinhamento com os casos de uso:** projete a API para dar suporte aos casos de uso identificados, mantendo a flexibilidade para cenários inovadores e menos frequentes.
2. **APIs de REST:** embora elas sejam de última geração, avalie se elas realmente atendem às suas necessidades, pois diferentes estilos de arquitetura podem ser mais adequados para casos de uso específicos.
3. **Modelo de dados abstrato:** implemente uma API com uma camada de abstração entre ela e o modelo de dados, evitando o acesso direto ao banco de dados, exceto quando for necessário.
4. **Considerações geográficas:** leve em conta os aspectos não funcionais, como latência e disponibilidade, posicionando estrategicamente os data centers próximos às regiões de seus principais usuários.
5. **Integração com soluções:** garanta que o design da API esteja em harmonia com outras soluções por meio de coordenação ou desacoplamento, mantendo uma clara comunicação interna e externa da estrutura.

## Infraestrutura de implantação e implementação de API

À medida que as empresas migram para a infraestrutura de dados e aplicações na nuvem e na nuvem híbrida, as infraestruturas de implantação e implementação de API também se tornam mais diversificadas. As APIs podem ser criadas com base em microsserviços e combinadas com APIs criadas como um monólito e tudo mais. Elas podem ficar em containers, máquinas virtuais (VMs), bare metal ou em algum lugar em um ambiente de nuvem pública.

Como em qualquer outra aplicação, ter um pipeline de dados automatizado de integração contínua e entrega contínua (CI/CD) é fundamental para o gerenciamento do ciclo de vida da API. [Adoção de GitOps](#) e do Argo CD pode fornecer gerenciamento centralizado das configurações da API, CD de implantações automatizadas, além de capacitar os esforços de colaboração entre a equipe de API, levando a um desenvolvimento rápido e a uma melhor qualidade do produto da API.

## Prática recomendada 4: priorize as operações de API

A equipe da plataforma de APIs gerencia as APIs quando elas são ativadas para garantir que sejam acessíveis e entregues de acordo com as expectativas dos desenvolvedores. Embora os fornecedores ofereçam soluções, selecionar a estratégia certa é essencial para o sucesso. O gerenciamento de APIs envolve duas funções principais:

1. Simplificar os processos internos para ser eficiente e reduzir custos.
2. Tornar as operações eficazes para atender às expectativas dos desenvolvedores externos do programa.

O artigo [Criando APIs excelentes, Parte 1: O padrão ouro](#) e o gráfico de operações de API na figura 3 podem ajudar a alcançar esses objetivos.

### Gráfico de operações de API

O gráfico de operações de API pode ser usado para definir táticas de operações para alcançar a estratégia de API de uma empresa. O círculo interno do gráfico representa as atividades e os efeitos internos de uma empresa e tudo o que está fora do círculo mostra os efeitos externos.

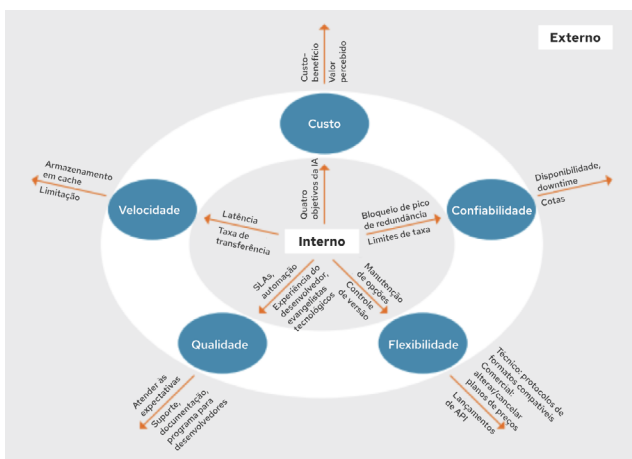


Figura 3. Gráfico de operações de API

1. **Confiabilidade:** garanta a disponibilidade de API por meio de redundância ou cotas e limites de taxa. Eles se alinham aos modelos de negócios e evitam downtime.
2. **Flexibilidade:** ofereça opções de adoção técnica e comercial, como alterações entre planos de preços ou cancelamentos. Lembre-se de que uma maior flexibilidade pode gerar maiores esforços e custos internos.
3. **Qualidade:** mantenha uma adesão consistente às expectativas do desenvolvedor usando acordos de nível de serviço (SLAs) e processos simplificados.
4. **Velocidade:** obtenha baixa latência e alta taxa de transferência com técnicas como limitação e armazenamento em cache, potencialmente alinhadas aos modelos de negócios.
5. **Custo:** otimize o valor do desenvolvedor e minimize os custos internos sem comprometer a qualidade.

Deve-se observar que vários fornecedores oferecem infraestrutura técnica para muitos desses desafios operacionais, inclusive a Red Hat. Usar um fornecedor geralmente é uma maneira econômica de resolver esses problemas, mas a estratégia deve ser completa.

## Componentes essenciais de gerenciamento de APIs

A operação de um ecossistema de APIs requer um conjunto exclusivo de componentes para que ele seja gerenciado com eficiência. Embora esses componentes variem de acordo com a estratégia de API, os três componentes principais serão os mesmos:

1. **Controle de acesso:** implemente sistemas de autenticação e autorização para permitir o acesso e identificar o tráfego de entrada;
2. **Limites de taxa e políticas de uso:** aplique cotas e restrições no tráfego para prever a carga;
3. **Análises:** capture e analise padrões de tráfego para monitorar o uso da API.

## Considerações sobre operações de API

Garantir que sua estratégia de operações de API esteja em harmonia com as estratégias gerais da sua empresa ajudará você a alocar recursos de acordo com a importância da API.

Para pensar em seus planos de operações de API, considere as seguintes áreas:

1. **Controle de acesso:** defina quem acessa, executa ações e impõe limites, garantindo o uso da API com foco na segurança;
2. **Métricas e alertas:** obtenha visibilidade com análises, medindo métricas personalizadas e configurando alertas para o desempenho da API;
3. **Gerenciamento de picos:** planeje a infraestrutura usando políticas e controle de acesso e estabeleça mecanismos de fallback, como limitação ou bloqueio de pico;
4. **Responsabilidade pelo uptime da API:** a propriedade clara do uptime da API é crucial, pois está diretamente ligada à geração e à captura de valor;
5. **Tratamento do uso indesejado:** distinguir o uso indesejado esperado e inesperado, gerenciando por meio de operações proativas e termos e condições;
6. **Comunicação:** tenha um plano de comunicação claro para desenvolvedores internos e externos para informá-los sobre o downtime planejado, a manutenção e as mudanças na API.

## Prática recomendada 5: sempre pense na experiência do desenvolvedor

Embora a experiência do desenvolvedor possa parecer uma questão de design de API, ela vai muito além disso. Pense na experiência do desenvolvedor como o empacotamento e a entrega da API, e não na API em si. Você pode ter uma API REST ou em containers perfeitamente projetada, mas se for difícil se inscrever para acessá-la, ler documentos e testá-la, então você criou uma experiência do desenvolvedor ruim que pode afetar significativamente o sucesso da sua API.

### O sucesso da API é mais do que um ótimo design

Uma API projetada com simplicidade e flexibilidade será desperdiçada se os desenvolvedores não se relacionarem com a API e não a adotarem. Ao mesmo tempo, o design de API bem concebido afeta consideravelmente a experiência do desenvolvedor e a adoção da API. A adoção é uma parte essencial da experiência do desenvolvedor.

Um dos pioneiros no campo de APIs e do gerenciamento de APIs, John Musser descreveu várias formas de melhorar o engajamento, que ainda são válidas:

1. Defina claramente o objetivo da API;
2. Ofereça inscrição fácil e acesso gratuito;
3. Informe os preços de forma transparente;
4. Forneça documentação completa.

Uma métrica importante para aprimorar o design da API e garantir sua fácil adoção é o "Time to First Hello World" (TTFHW), que mede a interação inicial com a API. Um contexto mais amplo é capturado pela métrica "Time to First Profitable App" (TTFPA), enfatizando os programas para desenvolvedores. Essa métrica é mais complexa, porque "lucrativo" ("profitable", no nome da métrica) é apenas uma questão de definição e depende da sua API e estratégia de negócios. Considerar a TTFPA é útil porque leva você a pensar nas operações da API como parte do seu programa.

A experiência do desenvolvedor envolve dois princípios: design eficiente da solução ou serviço e fácil acessibilidade. Um sólido programa de engajamento do desenvolvedor, incluindo portal, comunidade, evangelistas tecnológicos, eventos e medição, aumenta o engajamento. Alguns exemplos de métricas típicas para o portal do desenvolvedor são visitas à página, inscrições, tráfego de API ou solicitações de suporte. Os eventos podem ser medidos pelo número de participantes, adoção de API em hackathons ou leads. É útil fazer correlações, como "uma palestra em um evento gerou mais inscrições na API?"

Um programa para desenvolvedores deve incluir os seguintes elementos:

- **Criação de comunidades:** eventos como hackathons estimulam interações, promovendo a adoção;
- **Evangelistas tecnológicos que são desenvolvedores:** essenciais para o sucesso, pois envolvem e promovem os benefícios da API;
- **Parceiros-piloto e estudos de caso:** os primeiros usuários fornecem feedback e estudos de caso;
- **Parceiros do ecossistema:** as parcerias ampliam a adoção de forma sinérgica;
- **Medição:** as métricas se alinham aos objetivos da API, auxiliando no gerenciamento eficiente;
- **Comunicação:** portais, newsletters, servidores privados do Slack ou do Discord.

Criar um programa para desenvolvedores adaptado ao público reforça o engajamento e a experiência.

## Considerações sobre a avaliação da experiência do desenvolvedor

Proporcionar uma experiência bem planejada pode inspirar os desenvolvedores a maximizar seu potencial. Veja abaixo seis considerações essenciais para avaliar a experiência do desenvolvedor para ajudar a criar APIs que não sejam apenas linhas de código, mas gateways para a criatividade e a produtividade.

1. **Explicação do valor:** elabore uma apresentação rápida para transmitir o valor da API aos desenvolvedores;
2. **TTFHW e TTFPA:** avalie e minimize o TTFHW e o TTFPA, considerando todos os elementos da experiência do desenvolvedor (como portais);
3. **Processo de integração:** alinhe a integração com casos de uso de API, mantendo a simplicidade para o sucesso inicial do desenvolvedor;
4. **Fornecimento de valor:** garanta que a API entregue valor suficiente para atrair e reter desenvolvedores;
5. **Suporte ao desenvolvedor:** priorize o suporte self-service por meio de documentação, perguntas frequentes e fóruns, com mecanismos adicionais para resolver problemas mais complexos;
6. **Uso não convencional:** forneça suporte e documentação para desenvolvedores que exploram casos de uso não-padrão.

## Prática recomendada 6: vá além do marketing básico

O marketing de APIs para desenvolvedores pode ser desafiador, principalmente se o valor apresentado não corresponder às necessidades comerciais ou técnicas do desenvolvedor. As APIs devem ser comercializadas como qualquer outro produto, sendo alinhadas com a segmentação, o direcionamento e o posicionamento (STP). O marketing eficaz combina a API certa com os desenvolvedores certos, movido pelo valor da API e pelos princípios de STP.

1. **Segmentação:** categorize os clientes como usuários internos, parceiros, usuários finais ou desenvolvedores externos. Para um engajamento eficaz, segmente o vasto mercado de desenvolvedores usando o método jobs to be done (JTBD).
2. **Direcionamento:** avalie a atratividade do segmento com base na acessibilidade, substancialidade e diferenciação. Selecione segmentos promissores e adapte as táticas de marketing adequadamente.
3. **Posicionamento:** faça com que sua API se destaque ao atender a necessidades específicas, resolver pontos problemáticos e proporcionar ganhos para segmentos de desenvolvedores escolhidos.

Priorizar a experiência do desenvolvedor no marketing é fundamental para o sucesso. Estratégias como evangelistas tecnológicos, desenvolvedores, portais robustos para desenvolvedores, hackathons e outros eventos podem criar relacionamentos sólidos com desenvolvedores.

## Considerações sobre APIs de marketing

1. **Público-alvo:** priorize os principais grupos de usuários, desenvolvendo sua API em etapas com base em usuários internos, parceiros, clientes ou público.
2. **Seleção de especialistas:** escolha especialistas alinhados com a proposição de valor da sua API para uma promoção eficaz, considerando áreas como engenharia, suporte, vendas e gestão de produtos.
3. **Estratégia de eventos:** selecione os tipos de evento (horizontal ou vertical, global ou local, conferência ou hackathon) com base nas metas da API.
4. **Adequação do hackathon:** avalie a relevância do hackathon (inscrições, downloads de SDK, aplicações, recrutamento, branding) e planeje adequadamente.
5. **Marketing interno:** garanta suporte de todas as unidades, esclareça com o departamento de marketing e comunique os benefícios às equipes de produtos e aos clientes.

# Prática recomendada 7: lembre-se da descontinuação das APIs e da gestão de mudanças

A consultoria de API tende a focar muito no projeto, na criação e na operação de APIs. No entanto, um dos segmentos mais importantes, mas ignorado, da trajetória da API é o que acontece muitos meses após o lançamento e a operação: o gerenciamento das atualizações da API, incluindo sua depreciação.

Interrupções causadas por mudanças abruptas podem afetar a confiança e acarretar custos significativos, principalmente com desenvolvedores desconhecidos, aprovações de aplicativos mobile ou dispositivos sem recursos de atualização.

Mudanças na API normalmente são classificadas como disruptivas e não disruptivas. As mudanças não disruptivas incluem novos métodos e aprimoramentos, enquanto as mudanças disruptivas envolvem remoção, modificações ou depreciação total. Os números da versão principal e os planos de migração tratam de mudanças disruptivas e as versões secundárias, de mudanças não disruptivas. Pode ser difícil certificar-se de que as alterações não corromperão algumas aplicações, portanto, ao implantar quaisquer alterações na API, mesmo que sejam categorizadas como não disruptivas, recomendamos fortemente:

- Fornecer um endpoint de teste com a nova versão antes do lançamento;
- Enviar um email ou outro tipo de comunicado aos desenvolvedores informando sobre a alteração e indicando datas e detalhes.

Comunicação eficaz e contratos transparentes são essenciais. Compartilhe os detalhes do problema, cumpra os compromissos e defina a duração do suporte à versão. Por exemplo, a descontinuidade de uma API exige uma abordagem estruturada, incluindo aviso prévio, cobertura da mídia, planos de migração e ferramentas de exportação de dados, se necessário.

Um plano de migração facilita as atualizações da API:

1. Introduza uma nova versão para testes;
2. Notifique os usuários sobre a descontinuidade da versão antiga;
3. Ajude os usuários durante a transição;
4. Aposente a versão antiga.

O gerenciamento integral de APIs inclui previsão de atualizações e descontinuidades, comunicação eficaz e manutenção da confiança por meio de ações transparentes.

## Considerações sobre a longevidade da API

1. **Compromisso com a garantia do cliente:** este é, sem dúvida, o aspecto mais importante para seu programa de APIs. É preciso se perguntar: com que nível de estabilidade de serviço a empresa está disposta a se comprometer com seus usuários? Defina o compromisso de estabilidade do serviço para os usuários. Esse comprometimento influencia a adoção.
2. **Processo de mudanças e mudanças disruptivas:** determine procedimentos de lançamento para cumprir a garantia. Identifique as partes envolvidas e as etapas da aprovação.
3. **Comunicação de mudanças:** detecte, documente e comunique mudanças usando formatos de definição de API por meio de compatibilidade e comunicação clara.
4. **Gerenciamento de versão:** monitore o uso de versões mais antigas usando IDs de desenvolvedor e de usuário. Estabeleça um processo de descontinuidade de APIs para evitar problemas.
5. **Alinhamento de soluções:** coordene a evolução da API com as mudanças de solução relacionadas, considerando os compromissos do cliente e fazendo as adaptações necessárias.

## Como sustentar sua estratégia de API

Essas práticas recomendadas destinam-se a ajudar a definir, implementar e aprimorar sua estratégia de API. O objetivo é ajudar você a adaptar sua estratégia e descobrir novas oportunidades de API. É provável que muitas das perguntas das seções anteriores precisem ser respondidas em detalhes e novas oportunidades ou riscos podem surgir com o tempo. Alguns exemplos:

- Existem formas de expandir o valor da API para o cliente?
- Temos provisionado o suficiente para o futuro?
- A proposição de valor é atraente o suficiente para que os desenvolvedores se empenhem verdadeiramente?

À medida que o cenário de TI evolui, sua API e suas ofertas também devem evoluir. Os quatro principais fatores de mudança são os seguintes:

1. **Forças do setor:** novos concorrentes ou serviços podem substituir sua API.
2. **Forças do mercado:** mudanças nas demandas dos usuários ou nas condições do segmento de mercado.
3. **Forças macroeconômicas:** as mudanças no mercado global afetam os orçamentos dos usuários.
4. **Tendências:** legislações regulatórias ou tecnologia em evolução.

## Como desenvolver um programa de APIs duradouro

Um programa de APIs que combina casos de uso sólidos com potenciais clientes tem o potencial de ser lançado, crescer e evoluir com sucesso. Os programas de APIs bem-sucedidos criam espaço para inovar por meio da flexibilidade planejada da API e da resolução proativa de possíveis cenários negativos de uso por meio de design e operações. Também é fundamental garantir que os termos e condições da API ofereçam as ferramentas necessárias para reagir diante de comportamentos inesperados.

Talvez seja preciso reavaliar sua estratégia de API se:

- A dependência de inovações inesperadas ofuscar o valor. Uma infraestrutura robusta é fundamental para o sucesso dessa abordagem;
- A falta de casos de uso convincentes para sua empresa e seus usuários indicar a necessidade de uma abordagem diferente;
- Dúvidas internas sobre comportamento negativo persistirem, sinalizando medidas ou comunicação insuficientes;
- O frequente comprometimento ou uso indevido da API revelar um desalinhamento entre o valor pretendido e o valor real.

## APIs não existem no vácuo

O cenário de tecnologia em constante evolução introduziu uma série de avançadas ferramentas de desenvolvimento de aplicações nativas em nuvem, incluindo Kubernetes, Red Hat® OpenShift® e Red Hat OpenShift Service Mesh, que criam sinergias para melhorar a conectividade e agilizar o desenvolvimento de aplicações excepcionais. No entanto, em meio a esses avanços, os princípios básicos de uma estratégia robusta de API e a importância de um sistema eficiente de gerenciamento de APIs permanecem inabaláveis. Ao iniciar esse processo, busque uma solução de gerenciamento de APIs que não apenas se alinhe à sua estratégia global, mas também demonstre um entendimento sólido ou uma integração consistente com essas plataformas para facilitar a execução de sua visão estratégica.

# Por que escolher a Red Hat para desenvolver e gerenciar suas APIs?

A Red Hat defende a priorização de uma abordagem estratégica de design de API para garantir o sucesso do seu programa de APIs. Essa abordagem inclui todo o ciclo de vida da API, desde o planejamento e o design, com mapeamento de recursos e a modelagem de cenários de negócios, até o gerenciamento das APIs realizado com controle de acesso, acesso às APIs e a análise do uso.

Para ajudar as equipes de desenvolvimento a aplicar as práticas recomendadas da abordagem de priorização do design de API apresentada no nosso Manual do Proprietário de API, a Red Hat oferece o [Red Hat 3scale API Management](#), que é a solução de gerenciamento de APIs criada especificamente para ambientes de nuvem híbrida. Nossa solução distribuída, nativa em nuvem e baseada em container lightweight possibilita processar grandes cargas de trabalho e promove colaboração e abordagem focadas na segurança. O compartilhamento e o controle do acesso a serviços, recursos, aplicações e sistemas empresariais em ambientes de nuvem pública e privada capacitam sua empresa em ecossistemas complexos.

Mas, como mencionado acima, as APIs não existem no vácuo e, para aplicar algumas dessas práticas recomendadas, será preciso usar uma plataforma de aplicações completa para o desenvolvimento de aplicações nativas em nuvem. Assim sendo, o Red Hat 3scale API Management está incluído no portfólio do [Red Hat Application Foundations](#), uma coleção de ferramentas de middleware que incluem recursos, como integração, design de API, governança e registro de API, streaming e sistema de mensageria, que foram projetados para trabalhar com o [Red Hat OpenShift Container Platform](#). Ao usar o Red Hat OpenShift e o Red Hat Application Foundations juntos, as empresas têm uma plataforma completa de aplicações nativas em nuvem para fornecer novos software aos usuários de forma mais eficiente e segura, além de desbloquear recursos e benefícios estratégicos para suas empresas.

Aprimore sua abordagem de desenvolvimento de aplicações nativas em nuvem com as soluções Red Hat 3scale API Management, Red Hat Application Foundations e Red Hat OpenShift e experimente o futuro do desenvolvimento de aplicações simplificadas, eficientes e colaborativas.

## Conclusão

Esperamos que essas práticas recomendadas ajudem a orientar algumas de suas escolhas ou, pelo menos, respondam às suas perguntas. O processo de criação de sua estratégia de API começa com uma compreensão clara, consistente e de toda a empresa do valor da sua API. Porém, o valor e os objetivos de negócios da sua API não são estáticos. Mudanças acontecem e sua estratégia de API precisa acompanhar essas mudanças.

Com base em nossa experiência com os clientes e nossas observações da economia de API, podemos resumir algumas características de um programa de API eficiente da seguinte forma:

- Os usuários adotam uma API quando ela é importante para eles; A API faz um trabalho essencial ao resolver pontos problemáticos e gerar ganhos;
- Uma API fornece valor para os usuários, oferecendo uma proposição de valor e uma estratégia que muda de acordo com o ambiente;
- Uma API tem grande valor interno para a empresa; Ela é importante para a empresa, pois a ajuda a alcançar objetivos significativos;
- O maior número possível de stakeholders (idealmente, todos) ficam satisfeitos.

Conheça o [Red Hat 3scale for API management](#) e experimente os [recursos de gerenciamento de APIs](#) da Red Hat.