

# Ofereça à sua SOA um descanso com APIs:

A base de uma fundação moderna de negócios digitais

# 01

## INTRODUÇÃO

No início dos anos 2000, o paradigma da SOA (Arquitetura Orientada a Serviços) era anunciado por sua potencial agilidade. Entretanto, sua conexão imprecisa e uso de padrões não se converteram precisamente no tipo de transformação de negócios que os adotantes imaginaram. A tecnologia provou-se mais complexa e desafiadora de gerenciar do que os especialistas previram. Em contraste, a API RESTful, uma configuração evolutiva da SOAP, provou-se competente na entrega de integração de aplicativos flexíveis em um nível de habilidade aplicável para difundir a adoção. Ela possibilita a verdadeira transformação digital das experiências do cliente e dos negócios em si.

**Agora, as organizações que adotaram a SOA podem fazer a transição para uma arquitetura com base em API e se beneficiar de suas melhorias na conectividade e na capacidade de gerenciamento.**

Embora a distância entre a SOA (e os aplicativos legados) e as APIs possa parecer enorme, com a plataforma e as práticas corretas, o caminho da SOA para a API é possível. Este artigo analisa como tomar a ação e converter WebServices SOAP em uma parte produtiva de um ecossistema de API mais amplo.



## 02

**SOA: Uma ideia revolucionária que ficou num impasse**

## 03

**Transformação digital = transformação da experiência do cliente**

## 04

**As diferenças entre SOA e APIs**

**Transição para experiências do cliente impulsionadas por API**

## 05

**Visão geral do processo**

## 06

**Cinco práticas recomendadas de API**

## 07

**A função da plataforma de gerenciamento de API**

## 08

**Conclusão**



## SOA: Uma ideia revolucionária que ficou num impasse

À medida que a popularidade da Internet aumentou na década de 1990, demonstrando o poder de mercado sem precedentes de padrões abertos como HTML e HTTP, as pessoas começaram a se perguntar se o mesmo efeito poderia ser obtido com o próprio software. Os aplicativos poderiam alavancar padrões abertos para quebrar barreiras de integração e interoperação do mesmo modo que os padrões Web que impulsionaram a Internet para fora dos laboratórios e na economia?

Os arquitetos de software se perguntavam se seria muito difícil estabelecer softwares que invocassem funções de outros aplicativos de modo tão simples que os consumidores poderiam avançar de um site a outro. Por exemplo, como um site financeiro do consumidor poderia aproveitar vários feeds de um registrador de cotações sem a codificação personalizada, muito custosa e que consome muito tempo, exigida no momento? Até esse ponto, a conexão de aplicativos tinha envolvido o uso de middleware proprietário e links codificados personalizados que exigiam configurações de firewall específicas para funcionar.

No início do novo milênio, as maiores empresas de tecnologia surpreenderam o setor ao concordar em trabalhar juntas em um cenário compartilhado de padrões abertos que permitiria uma integração de aplicativos gratuita e com base na Internet. Esses eram os padrões de “Arquitetura Orientada a Serviços” ou SOA. Com base na linguagem XML, eles abrangiam SOAP (Protocolo Simples de Acesso a Objetos), WSDL (Linguagem de Descrição de Serviço Web) e vários outros. Esses padrões formaram a base para novas plataformas significativas, como o Microsoft .NET, BEA (posteriormente, Oracle) WebLogic, a nova geração de servidores IBM WebSphere, SAP NetWeaver e Oracle Fusion Middleware.

O conceito SOAP era viável. Com todas as partes importantes de acordo, muitas empresas embarcaram em programas ambiciosos para criar arquiteturas orientadas a serviços. Esse esforço envolveu funcionalidade de

software disponível como serviços Web com base em SOAP que poderiam se conectar facilmente, via HTTP, com WebServices SOAP dentro e fora da empresa. Novos conceitos como ESB (Barramento de Serviços Corporativo) emergiram, possibilitando que as organizações colocassem seus serviços Web em estruturas de software fáceis de integrar em novos aplicativos.

A SOA funcionou até um ponto. Muitos êxitos ofereceram aos usuários um nível de conveniência raramente proporcionado antes. Grandes aplicações para ERP e logística lançaram novas edições com centenas de funções disponíveis como WebServices. Os desenvolvedores descobriram uma nova agilidade à medida que vincularam aplicativos e criaram novas experiências do usuário. Novos “aplicativos compostos” criados usando WebServices possibilitaram mais personalização para os requisitos do usuário.

No entanto, o experimento SOA começou a fracassar após alguns anos. Embora revolucionário por sua acessibilidade e nível não usual de cooperação entre provedores de tecnologia concorrentes em seu uso, a SOA ainda era uma tecnologia difícil. Os ESBs eram custosos e difíceis de manter. A suposição era que o valor de negócios seria visto quando o projeto de SOA estivesse finalizado, mas parecia que o trabalho nunca acabava. O valor de negócios desejado das arquiteturas SOA bem organizadas, com diretórios intuitivos de WebServices que poderiam ser reutilizados em vários aplicativos, era muito difícil de ser obtido.

Em vez disso, o que progrediu durante a primeira década do século XXI foi um ambiente decididamente misto. Ele incluía alguns programas de SOA organizados, uma boa quantidade de arquiteturas orientadas a ESB, muitas empresas com WebServices desorganizados e dispersos e várias que ainda estavam muito atrasadas na curva de adoção SOA. Era necessária uma abordagem nova e mais simples.





## Transformação digital = transformação da experiência do cliente

A SOA foi, de certo modo, vítima do próprio sucesso. Mesmo imperfeita, ela demonstrou conclusivamente que um prêmio econômico massivo seria recebido por quem pudesse transformar seus negócios usando a tecnologia. A SOA abriu os olhos dos líderes de negócios para o potencial das aplicações compostas que poderiam incluir serviços relevantes à função do usuário. Eles viram a promessa de aplicações pouco vinculadas para criar novas e melhores experiências para o cliente. Havia a necessidade de desenvolver a tecnologia que possibilitaria essa transição.

O engajamento do cliente Omnichannel oferece um exemplo de transformação de negócios digitais em ação. Por exemplo, em um banco, seria possível conectar o cliente continuamente a informações da conta, transações, terceiros (como contas de investimento), serviços de empréstimo e representantes de atendimento ao cliente por telefone, computador, smartphone, quiosques locais e etc. Para isso, uma coleção abrangente de aplicações teria que encontrar conexões ágeis e simples. A SOA era muito deselegante para essa visão.

A solução apareceu na forma de APIs (Interfaces de Programa de Aplicações) com base em padrões. A API, uma entidade de software que conectava aplicativos e hardware, não era nova. Entretanto, com a adição de padrões abertos e simples, ela se tornou um meio para uma verdadeira revolução na TI. Atualmente, quando artigos de TI discutem APIs, eles fazem referência invariavelmente às APIs criadas para o protocolo de mensagem REST (Transferência de Estado Representacional) e para a linguagem de programação JSON. Essas APIs RESTful, como são conhecidas, oferecem aos desenvolvedores e aos arquitetos o tipo de conectividade muito simples idealizada, mas nunca concretizada, dos padrões SOA.

---

**As APIs RESTful, usando novos protocolos como HTTP, podem vincular praticamente quaisquer duas partes de um software com muita facilidade. Os padrões acordados para as APIs RESTful contêm um conjunto de métodos simples, como GET e PUT (dados).**

---

No exemplo do banco, enquanto os desenvolvedores tinham endereços de APIs (URLs) e parâmetros de dados, eles conseguiam criar um aplicativo bancário do consumidor que deu ao cliente acesso a contas, sistemas de negociação, etc., com relativa facilidade. Além disso, era muito simples alterar o aplicativo e suas conexões subjacentes de API. Desse modo, as APIs se tornaram as ferramentas da transformação digital.





### As diferenças entre SOA e APIs

As APIs buscam obter benefícios similares, mas, diferentemente da SOA, sua característica de design principal é a simplicidade. A simplicidade era crítica para possibilitar que os dispositivos móveis interagissem com outros sistemas. A SOA nunca foi projetada com esse tipo de simplicidade, então seu uso era limitado fora da empresa, com parceiros e etc. Por exemplo, em relação aos recursos, a API é acessível por uma URL usando a tecnologia Web padrão de HTTP.

A abordagem da API oferece a capacidade de começar aos poucos. Com a funcionalidade inicial limitada, é fácil obter desenvolvimento rápido em casos de uso mais sofisticados.

### Transição para foco na experiência do cliente impulsionadas por API

O lema da "lean startup" de "fail fast" é muito apropriado para o processo de envolvimento em um programa de API. De certo modo, você seria como uma startup no meio dos negócios. Você estaria desenvolvendo tecnologias que uniriam sistemas e processos de negócios anteriormente separados de novos modos, como uma startup. Você também pode acabar desestabilizando um pouco o processo, com os pontos positivos e negativos fornecidos.

*Comece aos poucos. Falhe rapidamente.*

**Você não precisa começar uma iniciativa de API corporativa muito grande. Na verdade, isso pode acabar se tornando um problema. É melhor começar com as metas de negócios e expandi-las para fornecer valor rapidamente.**

---

Comece com o cliente. As interações de negócios digitais atuais incluem um conjunto crescente de touchpoints. Pense sobre como a experiência do cliente pode ser beneficiada ao passar por esses touchpoints.

Por exemplo, uma interação com o cliente pode ser movida do desktop para o tablet ou telefone e depois voltar para o desktop sem nenhum inconveniente? Se não puder, esse é um excelente ponto para começar a pensar sobre as APIs e como elas podem facilitar a experiência do cliente com os touchpoints. A tecnologia de API é a base para conectar sistemas e touchpoints. Ela também conecta as pessoas ao centro da experiência do cliente.





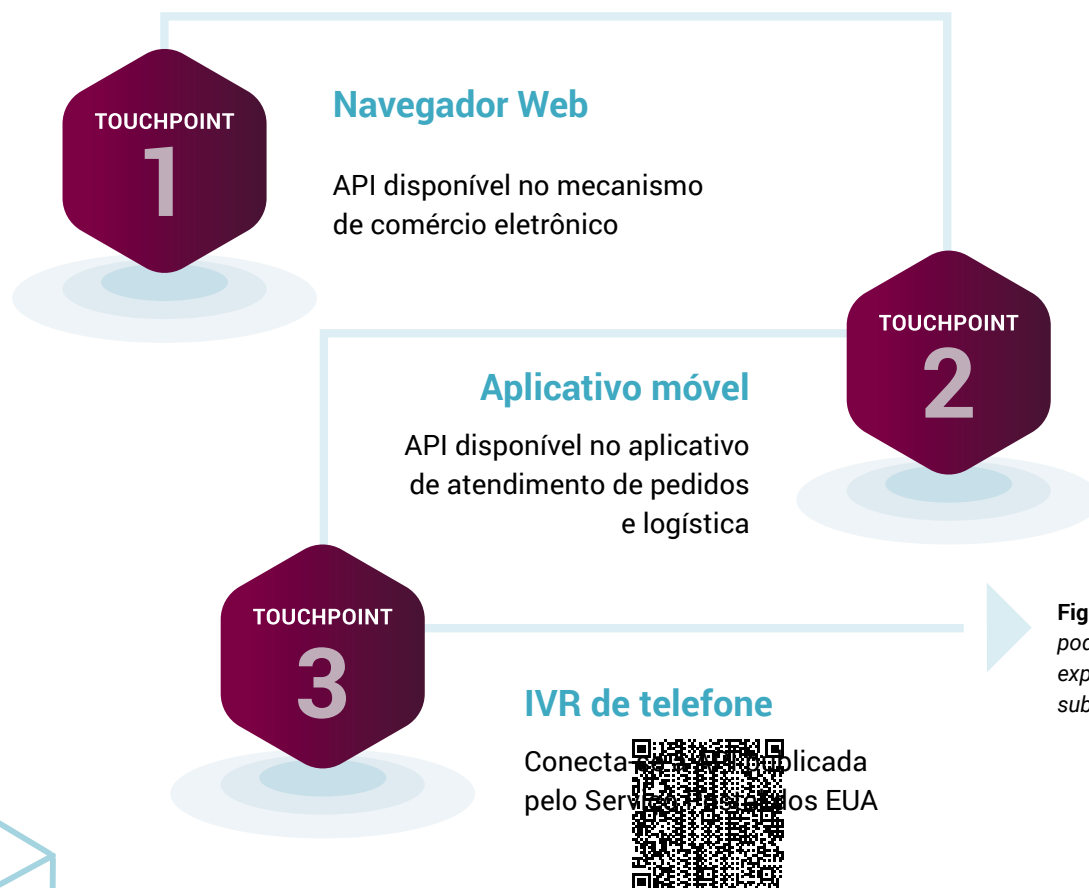
## Visão geral do processo

Após ter identificado uma experiência do cliente que pode ser aprimorada com as APIs, é hora de começar. A primeira etapa é modelar o fluxo do processo para entender o que acontecerá em cada etapa nas interações do cliente com vários sistemas subjacentes. Esse fluxo pode ser um pouco complexo devido ao modo como os negócios digitais operam.

Por exemplo, o primeiro touchpoint do cliente pode ser por um navegador que alcance um mecanismo de comércio eletrônico. O próximo touchpoint poderia ser por um aplicativo móvel que obtém o status da transação de um aplicativo de atendimento de pedidos e logística. Um terceiro touchpoint poderia ser por telefone e um sistema de IVR (Resposta Interativa de Voz), que obtém os dados do status do pedido de um aplicativo de acompanhamento de frete de um terceiro, como o Serviço Postal dos Estados Unidos.

Como mostra a **Figura 1**, cada touchpoint (sua aplicação associada) precisa ser disponibilizado como uma API. Algumas das funções dessa aplicação já podem estar disponíveis como WebServices SOAP. Nesse caso, será necessário fazer a conversão de SOAP para API. Muitas ferramentas de gerenciamento de API fornecem esse processo nativamente. Ele renderiza o WebService SOAP em uma API RESTful.

Cada API precisa de um “consumidor de API”, que é, normalmente, uma parte do código no lado do cliente. Em nosso exemplo, o aplicativo de rastreamento de pedidos para dispositivos móveis precisa “consumir” a API exposta no aplicativo de atendimento de pedidos.



**Figura 1:** Muitos touchpoints podem necessitar de APIs expostas em vários sistemas subjacentes.





## Cinco práticas recomendadas de API

Migrar dos WebServices SOAP e de aplicações legadas para experiências de cliente impulsionadas por API funciona melhor quando sua organização segue práticas recomendadas comprovadas. Você precisa de planejamento detalhado, adesão executiva, orçamentos, responsabilidade, tarefas escolhidas, calendários, etc. Isso é certo. Se não tiver esses tipos de práticas básicas de gerenciamento de projeto, você terá problemas com as APIs.

Recentemente, surgiram muitas práticas recomendadas que fazem com que o processo de transição de SOA para API funcione o melhor possível. Os destaques incluem:

### 01

#### **Mapeamento do comportamento do cliente**

Para que as APIs sejam relevantes, elas precisam se alinhar e ter a capacidade de adaptação de acordo com os requisitos em constante mudança dos clientes. Por exemplo, isso pode significar antecipar informações adicionais importantes para o cliente. Entender os padrões de navegação é fundamental.

### 02

#### **Compreender a diferença entre o uso de serviços WebServices**

A implementação de SOA muitas vezes resultou em centenas, se não milhares, de WebServices de acordo com a granularidade do design. As APIs, por design, devem executar a tarefa do desenvolvedor, que as usa com mais facilidade, e isso normalmente resulta na combinação de WebServices em uma única chamada de API. Por exemplo, um desenvolvedor que escreve um aplicativo não se importa com as muitas etapas que podem ser necessárias para inserir uma venda ou um novo contato. Um método de API que usa a quantidade de serviços da web necessária para realizar a tarefa ajuda a simplificar a tarefa do desenvolvedor.

### 03

#### **Planejar como as APIs migrarão de equipes isoladas para toda a organização**

Se usar as APIs de modo certo, você migrará naturalmente da fase de “startup”. Haverá um momento no qual você desejará fazer com que várias pessoas da organização possam usar as APIs. Afinal, as APIs são uma ferramenta para a empresa tanto quanto (ou mais) que fazem parte da TI. Uma boa prática é planejar a dispersão delas pela organização.

### 04

#### **Gerenciamento do controle das APIs por toda organização**

Dispersar o uso das APIs não significa perder o controle. Controles avançados sobre as APIs são fundamentais para a adoção efetiva em vários projetos em diferentes unidades de negócios. Mantenha aspectos unificados de suas APIs para maior eficiência. A melhor abordagem para obter controle é adotar uma plataforma de API em toda a empresa.

### 05

#### **Tornar o desenvolvimento das APIs mais acessível para um pool maior de desenvolvimento**

Programas de API de êxito incluem um portal do desenvolvedor de API. O portal fornece a capacidade de impulsionar sua equipe inicial de especialistas em API para capacitar outras pessoas na organização, tanto os consumidores quanto os criadores de API. Parte disso envolve fornecer um catálogo de APIs existentes. Com essa capacidade, os desenvolvedores podem testar APIs existentes para aprender como usá-las.

Os fóruns da comunidade fornecem um local para aprendizagem “peer-to-peer” e etc. É surpreendente ver como as pessoas querem usar as APIs assim que entendem seu potencial. Não é raro ver inovação não planejada à medida que equipes técnicas de fora do setor de TI usam APIs existentes para criar ferramentas que resolvam seus desafios de negócios.



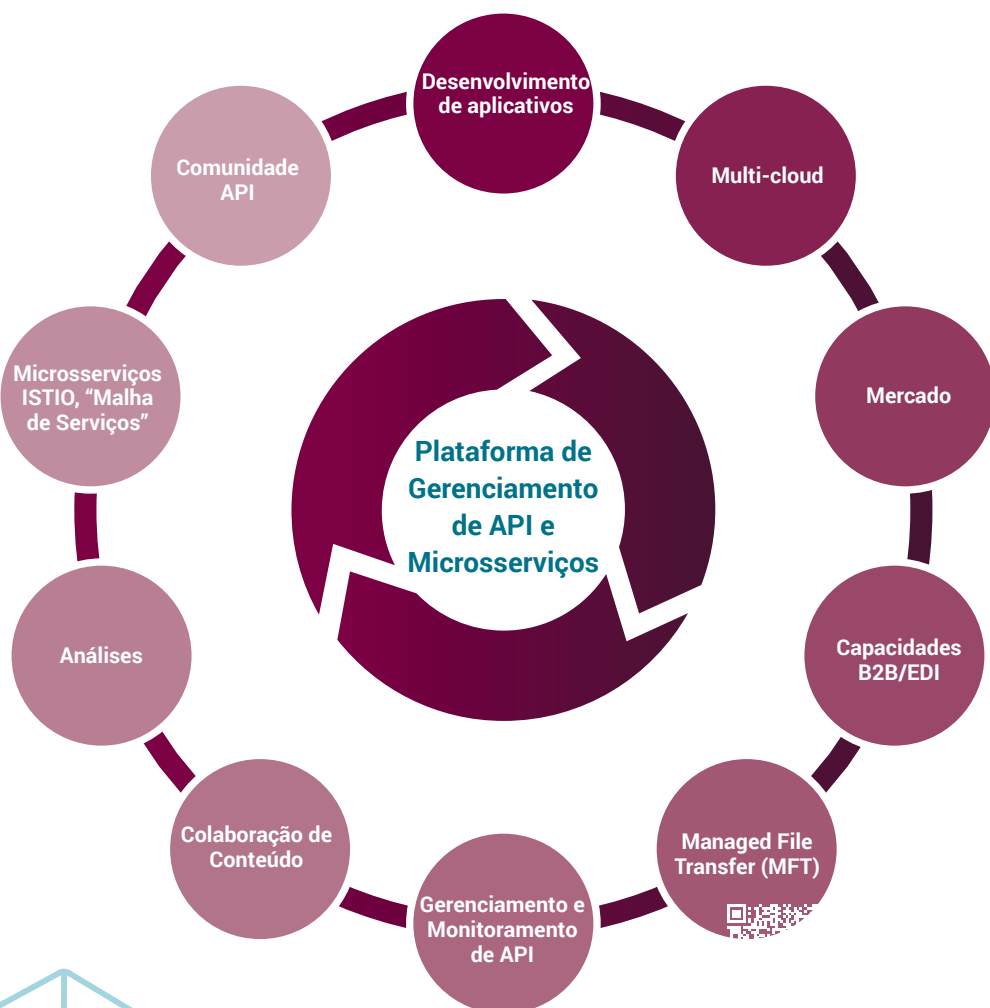
## A função da plataforma de gerenciamento de API

Uma plataforma de gerenciamento de API fornece à organização maneiras de simplificar o ciclo de vida da API, desde a criação até a desativação. É possível fazer uma mediação, como a conversão de SOAP para API, que simplifica a criação de APIs de seus serviços existentes. A plataforma ajuda a acelerar o uso da API com confiança, segurança e escalabilidade. Desde o planejamento até o desenvolvimento, a implantação e a desativação, ela torna possível colocar as APIs em uso consistente para melhorar a experiência do cliente.

Um conceito originalmente simples, o gerenciamento de API continua a evoluir para acompanhar a complexidade e a escala das crescentes estratégias sofisticadas de API. De fato, a nova geração de plataformas normalmente faz mais do que apenas gerenciar o ciclo de vida da API. Como a **Figura 2** sugere, as plataformas podem ser multifacetadas. Por exemplo, agora elas lidam rotineiramente com microsserviços além das APIs RESTful.

Elas auxiliam no uso de APIs e microsserviços no desenvolvimento de aplicativos, casos de uso B2B e colaboração de conteúdo. Por exemplo, elas podem facilitar o processo de fazer as APIs trabalharem entre parceiros. Elas também possibilitam análises avançadas do desempenho e dependências da API e de microsserviço.

Devido à capacidade das APIs de simplificar a integração, o gerenciamento de API normalmente é um componente central de uma plataforma de integração híbrida.



**Figura 2:** A nova geração de plataformas de gerenciamento de API e microsserviços abrange uma ampla gama de recursos, incluindo desenvolvimento de aplicativo, colaboração de conteúdo e análises.



## CONCLUSÃO

**As organizações que adotaram os WebServices SOA e a Arquitetura Orientada para Serviços podem migrar e aproveitar os benefícios das APIs RESTful. Fazer isso as coloca no caminho para melhorias transformadoras na experiência do cliente e na transformação digital.**

O uso de APIs capacita as empresas digitais dinâmicas em uma escala que foi primeiramente idealizada com o advento da Arquitetura Orientada para Serviços. A primeira etapa é começar a transição para uma arquitetura com base em API. As práticas recomendadas e os planos de gerenciamento modernos de API ajudam no processo. As principais considerações incluem pensar como uma startup e “falhar rapidamente” ao começar em uma escala pequena.

De lá, o desafio é expandir o acesso a APIs na organização de modo gerenciável. Aqui, a plataforma de gerenciamento de API pode ajudar bastante, possibilitando a criação de portais do desenvolvedor e a aplicação consistente de políticas de ciclo de vida da API. A combinação de práticas recomendadas e uma plataforma de gerenciamento de API para fornecer consistência e controle estabelecerá as condições para uma transformação bem-sucedida na experiência do cliente orientada à API.

As APIs devem ser integradas, automatizadas e disseminadas. A plataforma de gerenciamento deve ser compatível com a malha de serviços ISTIO, bem como com implementação da multi-cloud, DevOps e criação de CI/CD. Com essas capacidades, APIs e microsserviços podem servir como a base para experiências novas e empolgantes para clientes e parceiros.

### O que é uma estratégia focada em API para negócios digitais?

Descubra como você pode ajudar sua empresa a se conectar ao mundo digital.

[ASSISTA AO VÍDEO](#)

### Transforme a experiência do cliente – e sua empresa – com o gerenciamento de API

Como o gerenciamento de API pode alavancar sua empresa?

[SAIBA MAIS](#)

**Try AMPLIFY  
API Management  
online for free**

**INSCREVA-SE HOJE MESMO**

[axway.com/en/products/api-management](https://axway.com/en/products/api-management)

Copyright © Axway 2019. Todos os direitos reservados.  
axway\_WP\_give\_your\_SOA\_a\_REST\_with\_APIs:  
The\_cornerstone\_of\_a\_modern  
digital\_business\_foundation

