**Introdução**

Em uma era de alta interconectividade digital, a otimização constante dos processos de comunicação entre diferentes dispositivos, aplicações, tecnologias, usuários, servidores, ... permite que os sistemas atinjam funcionalidades que possibilitam o aproveitamento daquilo que já foi desenvolvido em diversos ambientes, fazendo com que a roda não precise ser reinventada toda vez que algo é requerido. Além disso, a alta integração dentre diferentes tecnologias possibilita a manutenção do estilo de vida contemporâneo, onde tudo é integrado ao simples toque de uma tela – desde pagamentos até o controle de utensílios -.

Uma ferramenta na área do desenvolvimento de Softwares e programas que é essencial nesta integração são as APIs, fundamentais para qualquer jovem aspirante desenvolvedor entrando no mercado de trabalho. Este trabalho apresenta um breve resumo do que são as APIs e como funcionam, mostrando também os tipos de APIs presentes no desenvolvimento e alguns exemplos de aplicações. Além disso, será feita uma breve menção dos protocolos, das linguagens e dos estilos de arquitetura presentes nas APIs. Finalmente, este artigo descreverá a diferença entre APIs e *Web Services* e citará alguns benefícios trazidos pelo uso de APIs.

**O que é uma API?**

Em suma, uma API (do Inglês, “*Application Programming Interface*”) é um conjunto de regras e protocolos que permitem que aplicações de *Software* se comuniquem entre si para trocar informação, funcionalidades e recursos.

As APIs permitem que os desenvolvedores integrem capacidades de outras aplicações, simplificando e acelerando o desenvolvimento de *Software* e aplicações. Além disso, as APIs fornecem aos donos das aplicações uma maneira simples de tornar as informações e funções destas disponíveis para os diferentes departamentos do grupo ou organização, ou então para parceiros comerciais e terceiros.

Toda API apresenta uma documentação no qual fornece instruções a respeito de tal e informações para os desenvolvedores em como trabalhar com a API e seus serviços.

**Como as APIs funcionam?**

O processo de comunicação possibilitado pela API pode ser pensado em termos de “pedidos” e “respostas” entre um “cliente” e um “servidor”. A aplicação enviando a solicitação é o cliente, e o servidor fornece a resposta. A API é a “ponte” estabelecendo a conexão dentre estes.

Uma maneira simples de entender como uma API funciona é olhando para um exemplo comum: serviços de processador de pagamentos de terceiros. Quando um usuário compra um produto em um *e-commerce*, o site talvez solicite ao usuário a “pagar com o PayPal” ou outro sistema terceirizado. Essa função depende de APIs para fazer a conexão.

* Quando o comprador clica no botão de pagar, uma chamada da API é enviada para trazer informação. Essa é a solicitação. Essa solicitação é processada para o servidor *web* através do URI (do Inglês, *Uniform Resource Identifier*) da API.
* Após receber uma solicitação válida da *webpage* do produto, a API chama o programa externo / o *web server*.
* O servidor manda uma resposta para a API com a informação requisitada.
* A API transfere a informação para a aplicação inicial, neste caso, o *e-commerce.*

**Tipos de APIs**

APIs podem ser categorizadas por caso de usuário, e incluem:

* *Data APIs;*
* APIs do Sistema Operacional;
* APIs remotas;
* *Web APIs.*

A maioria das APIs hoje em dia são *Web APIs*, que são um tipo de APIs remotas. Os quatro tipos de *Web APIs* são:

* *Open APIs;*
* *Partner APIs;*
* *Internal APIs;*
* *Composite APIs.*

**Exemplos de APIs**

Como as APIs permitem que organizações abram acesso para seus recursos enquanto mantém segurança e controle, elas se tornaram um aspecto fundamental no comércio moderno e de aplicações pessoais.

Alguns exemplos de APIs que usuários frequentemente encontram são:

* Logins universais;
* Internet das Coisas;
* Comparações presentes em sites de viagem;
* Aplicativos de navegação;
* Redes sociais;
* Aplicações SaaS.

**Protocolos, designs de arquitetura e linguagens em APIs**

Tradicionalmente, APIs se referem a interfaces conectados com uma aplicação criada com linguagens de programação de baixo nível, como JavaScript. Porém, APIs modernas variam em relação à suas arquiteturas e uso de formatos de informação. Elas são geralmente construídas para HTTP, resultando em uma interface amigável para desenvolvedores que é facilmente entendível por aplicativos escritos em Java, Ruby, Python e muitas outras linguagens.

A medida que o uso de APIs tem aumentado, o desenvolvimento e uso de protocolos, padrões e linguagens específicas tem ocorrido. Essas estruturas fornecem aos usuários com um conjunto de regras definidas, ou “especificações API”, que criam tipos de data, comandos e sintaxe aceitas. Em suma, estes protocolos API facilitaram a troca de informação padronizada.

Alguns exemplos desses protocolos são:

* SOAP (*Simple Object Access Protocol*);
* RPC (*Remote Procedure Call*);
* XML-RPC *(XML-Remote Procedure Call);*
* JSON-RPC;
* gRPC;
* WebSocket;
* REST (Representational State Transfer);
* GraphQL.

**APIs e serviços de web**

Um serviço web é um componente de Software da Internet que facilita a transferência de informação em uma rede. Como um serviço web expõem a informação e funcionalidade de uma aplicação para outras, de certa maneira, todo serviço web é uma API. Porém, nem toda API é um serviço WEB.

**Benefícios das APIs**

Em suma, as APIs simplificam o design e o desenvolvimento de novas aplicações e serviços, e a integração e manutenção de serviços existentes. Além disso, as APIs fornecem benefícios significantes para desenvolvedores e grandes organizações:

* Colaboração melhorada: empresas utilizam centenas e milhares de aplicações e plataformas *Cloud*. APIs permitem a integração destas para a comunicação eficiente. Através dessa integração, empresas podem automatizar *workflows* e melhorar a colaboração no ambiente de trabalho;
* Inovação acelerada: APIs fornecem flexibilidade, permitindo a empresas fazerem conexões com novos parceiros de negócios e fornecerem serviços para o mercado. Essa flexibilidade também possibilita as empresas de adentrarem novos mercados, incentivando a transformação digital e o acesso à novos mercados.
* Monetização da informação: muitas empresas escolhem oferecer suas APIs de graça, pelo menos inicialmente, para que possam construir uma audiência de desenvolvedores ao redor de sua marca e forjar relações com potenciais parceiros. Se a API garante acesso a ativos digitais de valor, o negócio pode monetizar a API ao vender acesso. Essa prática é conhecida como Economia da API.
* Segurança de sistema: As APIs separam a aplicação requerente da infraestrutura do serviço que responde e oferecem camadas de segurança entre os dois à medida que comunicam. Por exemplo, as chamadas de API requerem normalmente credenciais de autenticação. Os cabeçalhos HTTP, os cookies ou as cadeias de consulta podem proporcionar segurança adicional durante a troca de dados. Um gateway de API pode controlar o acesso para minimizar ainda mais as ameaças à segurança.
* Segurança e Privacidade do usuário: as APIs fornecem proteção adicional numa rede. Podem também fornecer outra camada de proteção para os utilizadores pessoais. Quando um sítio Web pede a localização de um utilizador (uma API de localização fornece esta informação), o utilizador pode decidir se permite ou recusa este pedido.

**Conclusão**

Entender como as APIs funcionam e qual API se adequa melhor para a aplicação desenvolvida possibilita ao desenvolvedor simplificar seu processo de criação de código, enquanto ao mesmo tempo potencializa seu projeto com ferramentas poderosas presentes no mercado. O impacto das APIs vai além dos arquivos de código e de suas funcionalidades digitais: esses protocolos permitem empresas construir pontes e fomenta a inovação tecnológica coletiva.

**Referências**

# GOODWIN, Michael. “*What is an API (application programming interface)?”.* Disponível em: <https://www.ibm.com/think/topics/api>. Acesso em: 18 ago. 2025.