

Curso Técnico em Informática para Internet



Modulo Específico II

Desenvolvimento de APIs



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

Hilton Elias

APIs: O que são, para que servem e tipos de aplicações

UMA VISÃO GERAL SOBRE APIS E SUA IMPORTÂNCIA
NO DESENVOLVIMENTO DE SOFTWARE



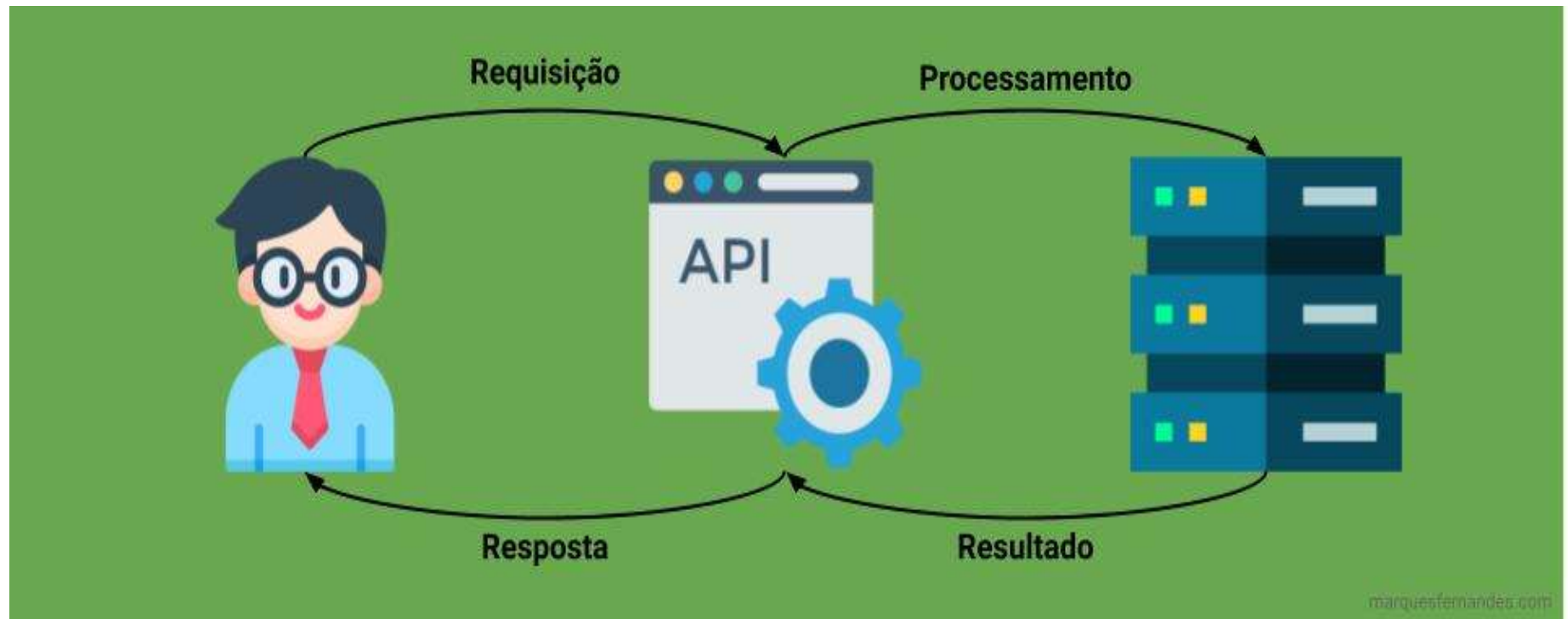
1. Introdução às APIs

Definição de API:

Uma API (Application Programming Interface) é um conjunto de normas e protocolos que permitem a interação e o compartilhamento de informações entre diversas aplicações de software.

Elas atuam como pontes que facilitam o fluxo contínuo de dados e a funcionalidade entre sistemas.

Comunicação entre sistemas usando uma API



2. Funcionamento das APIs

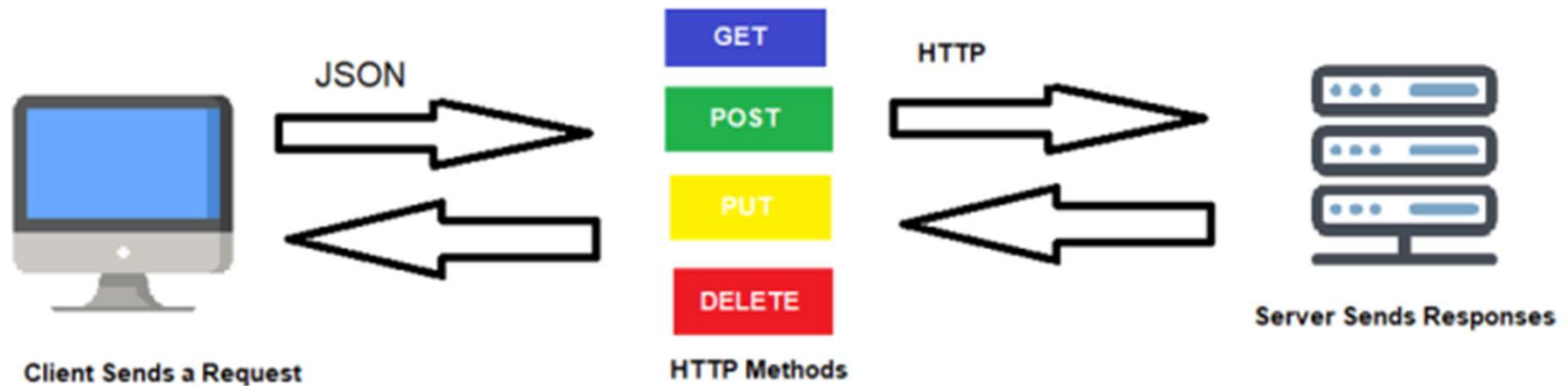
Como funcionam as APIs:

As APIs funcionam como ligações entre sistemas diferentes, como aplicativos e outras formas tecnológicas.

Essas conexões se dividem em dois tipos:

1. O que envia a solicitação é chamado de cliente.
2. O que responde é chamado de servidor.

Processo de uma Requisição API




3. Tipos de APIs

APIs Públicas:

APIs públicas ou abertas são aquelas que podem ser utilizadas por qualquer pessoa, geralmente para serviços mais gerais.

APIs Privadas:

APIs privadas são restritas somente às pessoas que têm autorização para uso, sendo muito utilizadas para sistemas internos devido ao maior grau de segurança.




3. Tipos de APIs

APIs de Parceiros:

APIs de parceiros permitem uma comunicação facilitada entre parceiros de empreendimentos e podem exigir acessos especiais.

APIs Compostas:

APIs compostas combinam várias informações e sistemas, sendo recomendadas quando se deseja facilitar a comunicação entre vários dispositivos.



4. Importância das APIs

Vantagens das APIs:

Sistemas Integrados: APIs permitem a integração de sistemas, facilitando a comunicação e a troca de dados entre eles.

Boa Comunicação com os Desenvolvedores: APIs incentivam uma comunicação mais constante com os desenvolvedores.

4. Importância das APIs

Vantagens das APIs:

Mais Qualidade ao Sistema: O uso de APIs aumenta a amplitude de funcionalidades de um sistema, melhorando a automação de atividades.

Melhor Segurança: APIs utilizam criptografia para garantir a segurança dos dados.

5. Exemplos Práticos

APIs em Aplicações do Dia a Dia:

Aplicativos de Previsão do Tempo: Utilizam APIs de serviços meteorológicos para fornecer informações atualizadas sobre o clima.

Gateways de Pagamento: E-commerces utilizam APIs de gateways de pagamento como PayPal ou Stripe para processar transações de forma segura.

5. Exemplos Práticos

APIs em Aplicações do Dia a Dia:

Redes Sociais: APIs permitem compartilhar notícias de sites no Facebook ou Twitter, gerando previews e links.

Serviços de Mapa: Aplicativos como Google Maps utilizam APIs para mostrar mapas, oferecer percursos e localizar estabelecimentos.

6. Estratégias e Inovações com APIs

APIs e Open Innovation (um modelo onde as empresas estão dispostas a cooperar com terceiros para encontrar soluções inovadoras):

Conectividade: APIs são a espinha dorsal da conectividade na era digital, permitindo a interação entre diferentes sistemas e aplicações.

Inovação: APIs alavancam a inovação ao permitir que desenvolvedores construam soluções a partir de software já existente.

6. Estratégias e Inovações com APIs

Eficiência: APIs levam fluidez aos processos e melhoram a eficiência.

User Experience: APIs melhoram a experiência do usuário ao permitir a integração entre diversos sistemas.

Crescimento do Negócio: Empresas que expõem suas APIs de forma segura podem aumentar seu alcance.

7. Desafios na Implementação de APIs

Segurança e Gerenciamento:

As APIs precisam utilizar criptografia e mecanismos de autenticação para garantir a segurança dos dados trafegados.

É essencial realizar testes de penetração regularmente para assegurar a consistência da segurança.

8. Protocolos e Padrões de Comunicação de APIs

Protocolos de Comunicação:

As APIs geralmente utilizam protocolos como HTTP e HTTPS para a comunicação entre sistemas. Esses protocolos garantem a segurança e a integridade dos dados transmitidos.

Padrões de Dados:

APIs utilizam padrões como JSON e XML para formatar os dados transmitidos entre sistemas. Esses formatos são escolhidos por sua simplicidade e facilidade de uso.



XML e JSON (O que é, semelhanças, diferenças, utilização)

O que é XML?

XML é um formato de dados que foi consolidado pelo W3C, sendo iniciados estudos em meados das décadas de 1990.

O objetivo era criar um tipo de formato que poderia ser lido por software e que tivesse flexibilidade e simplicidade, visando, entre outras coisas:

- Possibilidade de criação de tags (você é quem cria as tags).
- Concentração na estrutura da informação e não em sua aparência.

XML e JSON (O que é, semelhanças, diferenças, utilização)

O que é JSON?

JSON, um acrônimo (palavra formada pela inicial) para “JavaScript Object Notation”, é um formato de padrão aberto que utiliza texto legível a humanos para transmitir objetos de dados consistindo de pares atributo-valor.

XML x JSON

Os dois modelos representam informações no formato texto. Ambos possuem natureza auto-descritiva (ou seja, basta “bater o olho” em um arquivo JSON ou em um arquivo XML para entender o seu significado).

Ambos são capazes de representar informação complexa, difícil de representar no formato tabular. Alguns exemplos: objetos compostos (objetos dentro de objetos), relações de hierarquia, atributos multivalorados, arrays, dados ausentes, etc.

Ambos podem ser utilizados para transportar informações em aplicações AJAX.

XML x JSON

Ambos podem ser considerados padrões para representação de dados. XML é um padrão W3C, enquanto JSON foi formalizado na RFC 4627.

Ambos são independentes de linguagem. Dados representados em XML e JSON podem ser acessados por qualquer linguagem de programação, através de API's específicas (API javascript por exemplo).

Exemplos de dados formatados em JSON e XML

XML



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<linkedin>
  <artigo>
    <nome>Entendendo o Json</nome>
    <data_publicacao>20-05-2022</data_publicacao>
  </artigo>
  <artigo>
    <nome>Entendendo o Json detalhadamente</nome>
    <data_publicacao>30-05-2022</data_publicacao>
  </artigo>
</linkedin>
```

JSON



```
{
  "linkedin": {
    "artigo": [{
      "nome": "Entendendo o Json",
      "data_publicacao": "20-05-2022"
    },
    {
      "nome": "Entendendo o Json detalhadamente",
      "data_publicacao": "30-05-2022"
    }
  ]
}
```

9. Autenticação e Autorização em APIs

Mecanismos de Autenticação:

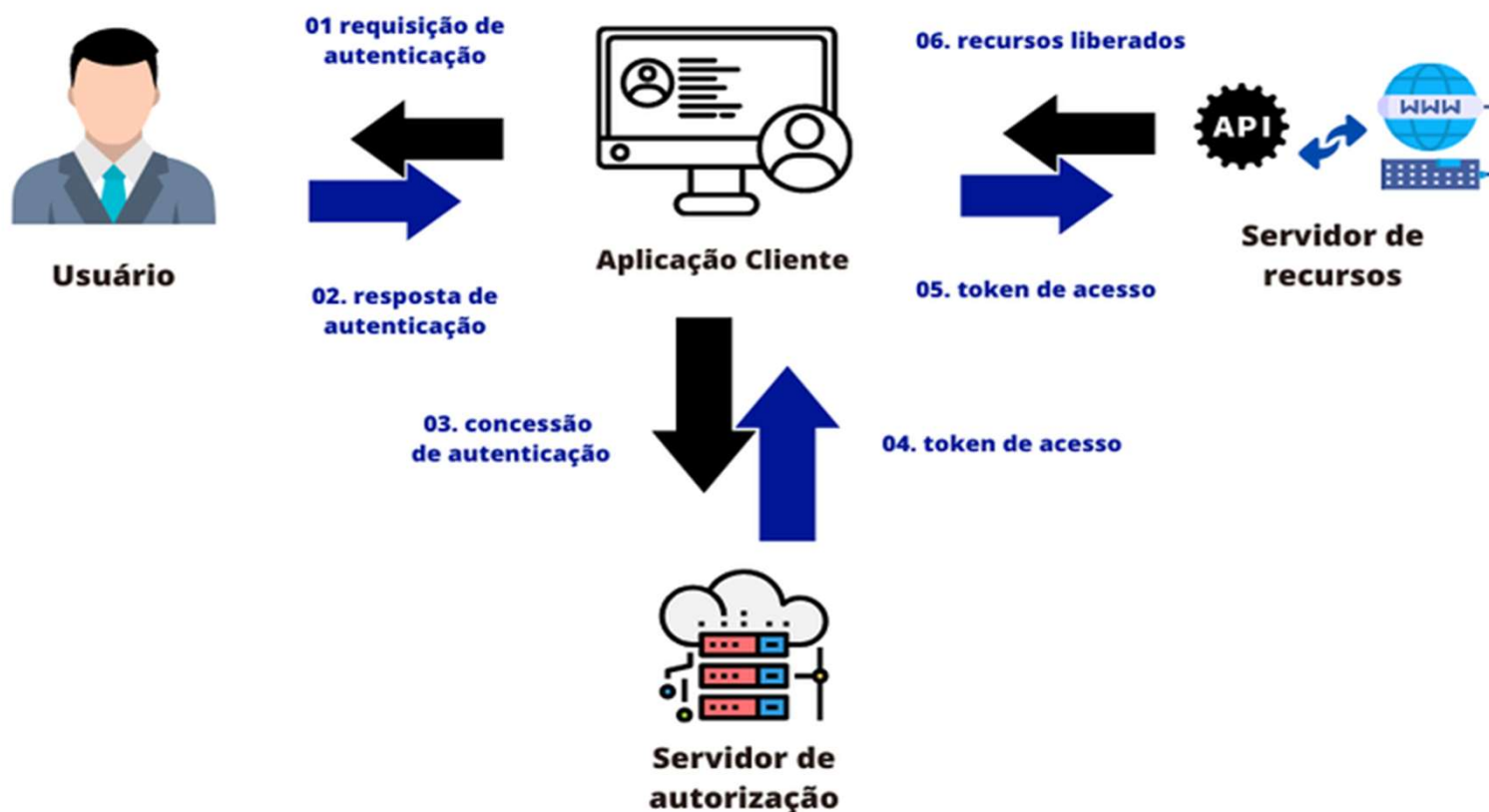
APIs utilizam diversos mecanismos de autenticação, como tokens, OAuth, e API keys, para garantir que apenas usuários autorizados possam acessar os serviços.

Autorização:

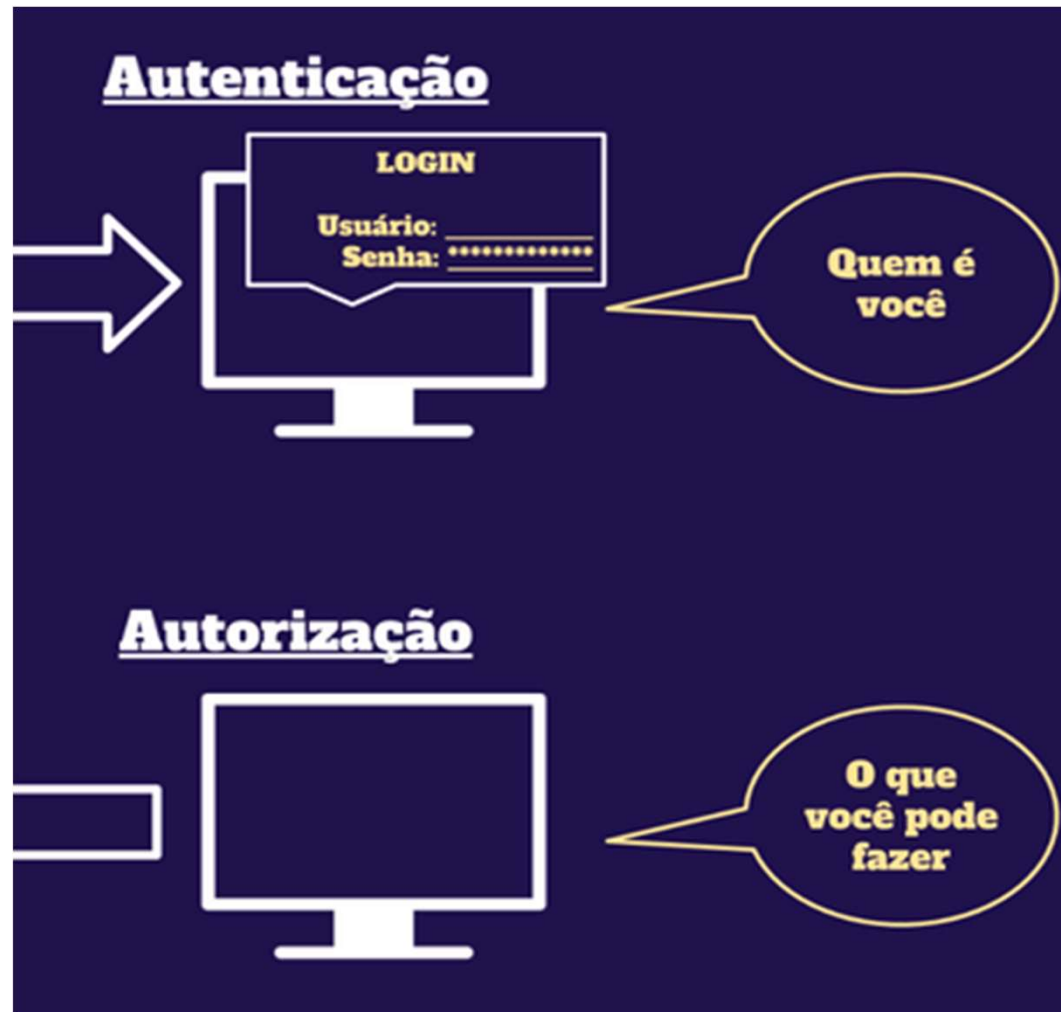
- Após a autenticação, a autorização determina o que o usuário autenticado pode fazer. APIs utilizam controles de acesso para limitar as operações que podem ser realizadas.

Fluxo de Autenticação usando OAuth

Fluxo do Protocolo OAuth 2.0



Diferença entre Autenticação e Autorização




10. Desempenho e Escalabilidade de APIs

Desempenho:

Medir e otimizar o desempenho das APIs é crucial para garantir uma boa experiência do usuário. Isso inclui monitorar tempos de resposta e "latência", o termo também é sinônimo de atraso, ou seja, quanto menor a latência de um sistema, melhor será a experiência do requisitante.

Escalabilidade:

Projetar APIs escaláveis (que podem ser aplicadas em diferentes proporções) é importante para lidar com aumentos de carga e tráfego. Isso pode incluir técnicas como balanceamento de carga e caching.



11. Boas Práticas no Desenvolvimento de APIs

Design de API:

Boas práticas de design incluem a utilização de endpoints intuitivos (uma das extremidades de uma conexão), documentação clara, e a adoção de padrões RESTful (uma interface que dois sistemas de computador usam para trocar informações de forma segura pela internet).

Manutenção e Versionamento:

Manter e versionar APIs é essencial para garantir compatibilidade e continuidade dos serviços à medida que novas funcionalidades são adicionadas.

Conclusão

Resumo da importância das APIs:

- APIs são fundamentais para o desenvolvimento moderno de software, permitindo a integração de sistemas, alavancando a inovação e melhorando a eficiência operacional.

Impacto das APIs no Futuro do Desenvolvimento de Software:

- APIs continuarão a desempenhar um papel crucial na evolução do desenvolvimento de software, facilitando a criação de soluções inovadoras e integradas.