Manual Completo de Estudo sobre APIs

---

1. O que é uma API?

API significa *Application Programming Interface* (Interface de Programação de Aplicações).
É uma ponte que permite que diferentes sistemas, softwares, ou aplicações conversem entre si e troquem informações.

Imagine a API como o garçom em um restaurante: você (aplicação cliente) fala com o garçom (API), ele leva seu pedido para a cozinha (servidor/sistema), e depois traz a comida (dados/resposta) para você.

---

2. Tipos de APIs

2.1 APIs Web (mais comuns)

- **RESTful APIs**: Usam o protocolo HTTP para comunicação, com padrões simples e claros.
- **SOAP APIs**: Usam XML para troca de mensagens, mais complexas e estruturadas.
- **GraphQL**: Permite ao cliente especificar exatamente os dados que quer, evitando excesso ou falta de informação.

2.2 Outras APIs

- **APIs de Sistema Operacional** (ex: Windows API).
- **APIs de Biblioteca/Framework** (ex: API do React).
- **APIs de Hardware** (ex: APIs para sensores ou impressoras).

---

3. Como funciona uma API Web?

1. **Cliente envia uma requisição (request):** A aplicação cliente envia uma mensagem para o servidor pedindo alguma coisa (ex: dados de usuário).
2. **Servidor processa a requisição:** O servidor recebe o pedido, executa a lógica, acessa banco de dados, etc.
3. **Servidor envia resposta (response):** O servidor retorna os dados ou o resultado da operação para o cliente.

---

4. Componentes principais de uma API Web

4.1 Endpoints

São URLs que representam recursos ou ações. Exemplo:
`https://api.meusite.com/usuarios` — aqui acessamos os usuários.

4.2 Métodos HTTP

São verbos que indicam o tipo de operação:

Método	Descrição	Uso comum
GET	Buscar dados	Buscar lista de usuários
POST	Criar novo recurso	Criar um novo usuário
PUT	Atualizar recurso inteiro	Atualizar dados completos do usuário
PATCH	Atualizar parte do recurso	Atualizar só o email do usuário
DELETE	Apagar recurso	Deletar um usuário

4.3 Cabeçalhos (Headers)

Informações extras enviadas junto com a requisição ou resposta, como:

- Tipo do conteúdo (`Content-Type: application/json`)
- Autenticação (`Authorization: Bearer token`)
- Controle de cache e muito mais.

4.4 Corpo da Requisição (Request Body)

Dados enviados para o servidor em métodos como POST, PUT, PATCH. Normalmente em JSON ou XML.

4.5 Código de Status HTTP

Indica se a requisição foi bem sucedida ou não. Exemplos comuns:

Código	Significado
200	OK (sucesso)
201	Criado (recurso criado)
400	Requisição inválida
401	Não autorizado
404	Recurso não encontrado
500	Erro interno do servidor

---

5. Documentação de APIs

Uma API só é útil se for bem documentada!

Documentação deve conter:

- Lista de endpoints e seus métodos
- Parâmetros aceitos
- Formato das requisições e respostas
- Exemplos práticos
- Erros possíveis e como lidar

Ferramentas para documentação:

- **Swagger / OpenAPI** (padrão muito usado)
- **Postman** (para testar e documentar APIs)

---

6. Autenticação e Segurança

Como garantir que só pessoas/automações autorizadas usem a API?

- **API Keys:** Chaves simples para identificar o cliente.
- **OAuth 2.0:** Padrão mais seguro para autorizar acessos.
- **JWT (JSON Web Tokens):** Tokens compactos usados para autenticar sessões.
- **HTTPS:** Sempre use conexão segura para proteger dados em trânsito.

---

7. Versionamento de API

Para manter a compatibilidade e permitir melhorias, APIs usam versões. Exemplo:

- `https://api.meusite.com/v1/usuarios`
- `https://api.meusite.com/v2/usuarios`

Assim, quem usa a API pode escolher qual versão usar, evitando quebras inesperadas.

---

8. Testes e Qualidade

Testar APIs é fundamental para garantir que elas funcionem bem e não quebrem com o tempo.

Tipos de testes:

- **Teste funcional:** Verifica se a API responde corretamente.
- **Teste de carga:** Avalia o comportamento sob alto uso.
- **Teste de segurança:** Busca vulnerabilidades.

Ferramentas comuns: **Postman**, **Insomnia**, **JMeter**, **Swagger UI**

---

9. Monitoramento e Logs

Depois de lançar a API, é importante monitorar o uso e o desempenho.

- Monitorar tempos de resposta e erros.
- Armazenar logs para análise de problemas.
- Usar ferramentas como **New Relic**, **Datadog** ou **Elastic Stack**.

---

10. Boas práticas no desenvolvimento de APIs

- Use nomes claros para endpoints e recursos.
- Seja consistente no uso de métodos HTTP.
- Retorne mensagens de erro claras e padronizadas.
- Evite retornar dados desnecessários.
- Documente tudo.
- Garanta segurança e privacidade.
- Mantenha a API simples e intuitiva.

---

11. Exemplos Práticos (Simples em JSON)

Requisição para criar um usuário (POST):

```
POST /usuarios
Content-Type: application/json

{
  "nome": "Maria Silva",
  "email": "maria@example.com"
}
```

Resposta do servidor:

```
201 Created
{
  "id": 123,
  "nome": "Maria Silva",
  "email": "maria@example.com",
  "criado_em": "2025-08-11T19:00:00Z"
}
```

---

12. Tecnologias e Frameworks Populares para APIs

- **Node.js (Express, Fastify)**
- **Python (Django REST Framework, Flask)**
- **Java (Spring Boot)**
- **PHP (Laravel)**
- **Ruby (Rails API Mode)**

Conclusão

APIs são essenciais para a comunicação entre sistemas, conectando aplicações de forma padronizada, segura e eficiente. Entender bem seus conceitos, funcionamento, segurança e boas práticas é fundamental para qualquer profissional de tecnologia.

MANUAL DE ESTUDOS DE APIs

O QUE È UMA API?

Un API (Application Programming interface) e um ← conjäntó-de regras que permitem , comunicaçáo entre aplicaçoes”

TIPOS DE APIs

OPEN
Publicas que podem ser accesadós por quaiquer qes-

INTERNAS
Usadas dentro de uma ore organizáçáo

PARCEIRAS
Disponibilizadas apenas a parterios espeeifiços

MODELAGEM DE APIs

REST

- Usa á inrio nerós (CET. POST. PUT, DELETE) e nosis dados
- GraphQL permirria clientes exceçificár dados pode sàs

AUTENTICAÇÃO E AUTORIZAÇÁO

Autentificação

Verifica neinżda identidade

Autorizaçåo valida permissais

OAauth A/tonles

OAuih2.0

DOCUMENTACÃO

Usa Swagger/Qpen API e Postmán

Decumentar a un protocool jero

BOAS PRATICAS

- Use naneroc conémo tempo
- Propuv'i error essengegens

VERSÃO E DEPRECIAÇÃO

Versionam por compatibilidade

Deprecaçåo entre versóes

Abre ver-escoberaedores