**APIs**

**Introdução e Conceitos Fundamentais**

**O que é uma API?**

Uma API (Application Programming Interface) é um conjunto de protocolos, rotinas e ferramentas que permitem a comunicação entre diferentes sistemas de software. Sua estrutura permite que exista comunicação independentemente da liguagem de programação utilizada.

“De modo prático, os recursos da API são compartilhados através de Web Servicese utiliza XMLou JSONcomo formato de comunicação, ou seja, quando um cliente faz uma requisição, é retornado uma resposta no formato XML ou JSON, quando a API precisa de alguma informação do cliente, o formato enviado também precisa ser o mesmo” (GALINDO JUNIOR *et al.*, 2021)

Ou seja, o formato de comunicação entre os sistemas deve ser consistente, ainda que em linguagens de programação diferentes.

**Tipos de APIs:**

1. APIs Web (HTTP/REST, SOAP, GraphQL)

2. APIs de Sistema Operacional

3. APIs de Bibliotecas

4. APIs de Hardware

**Vantagens das APIs:**

- Reutilização de código

- Escalabilidade

- Interoperabilidade

**APIs Web mais Comuns:**

1. REST (Representational State Transfer)

- Baseada em HTTP

- Stateless

- Usa JSON/XML

2. SOAP (Simple Object Access Protocol)

3. GraphQL

**Princípios do REST:**

- Recursos identificados por URIs

- Métodos HTTP padronizados

- HATEOAS

**Comunicação:**

“Conforme a documentação do protocolo HTTP os verbos são:

GET: Solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

POST: É utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

PUT: Substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

DELETE: Remove um recurso específico.” (GALINDO JUNIOR *et al.*, 2021)

**Cache – Uma boa prática**

“Tem como objetivo melhorar o desempenho da rede, para isso, toda resposta do servidor deverá ser rotulada como armazenável com cache ou não armazenáveis em cache. Se uma resposta poder ser armazenada em cache, ela fica salva em cache, quando o cliente fizer a mesma requisição ou equivalente, a informação da resposta será utilizada novamente pelo cache, a menos que a informação no servidor tenha sofrido alterações. “ (GALINDO JUNIOR *et al.*, 2021)

**Demais boas Práticas:**

✔ Versionamento

✔ Documentação Clara

✔ Autenticação Segura

✔ Tratamento de Erros

**Desenvolvimento de APIs com Spring Boot – Framework Java**

**Por que Spring Boot?**

- Produtividade

- Ecossistema robusto

- Embedded Server

**Estrutura Básica:**

1. Controlador (Controller)

2. Serviço (Service)

3. Repositório (Repository)

**Ferramentas para Testar APIs:**

Fazem as chamadas as URI’s que retornam, a depender do verbo, as requisições solicitadas.

- Postman

- Swagger/OpenAPI

- Insomnia

**Exemplo de Chamada no Postman:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**(GALINDO JUNIOR *et al.*, 2021)**

**Tendências:**

- APIs Assíncronas

- Serverless APIs

- APIs em Microserviços

**Referência:**GALINDO JUNIOR, Edemilton Alcides *et al*. DESENVOLVIMENTO DE API REST COM SPRING BOOT. **Revista Científica do Unirios**, Rio de Janeiro, p. 499-525, jan. 2021. Disponível em: https://www.publicacoes.unirios.edu.br/index.php/revistarios/article/view/102/102. Acesso em: 13 ago. 2025.