

Faculdade de Tecnologia de Sorocaba

Tecnologia em Análise e Desenvolvimento de Sistemas

**PROGRAMAÇÃO PARA WEB:** **APIs e integrações**

ATIVIDADE 2

Prof.º Denilce de Almeida Oliveira Veloso

Disciplina: Programação para Web

Vinicius de Freitas Vieira 0030482323018

Sorocaba

Agosto/2025

# 1. INTRODUÇÃO

Atualmente, a evolução tecnológica tem proporcionado diversas ferramentas e arquiteturas para o desenvolvimento de sistemas, o que resultou em uma grande variedade de padrões. É comum encontrarmos sistemas com estruturas em camadas bem definidas, assim como outros organizados em uma arquitetura monolítica. Um dos principais avanços que acompanham essas mudanças estruturais são as APIs, que permitem a comunicação entre diferentes sistemas por meio de uma interface padronizada.

As APIs (Application Programming Interface), traduzidas literalmente como “Interface de Programação de Aplicações”, funcionam como pontes que conectam sistemas distintos. Essa comunicação se dá, geralmente, por meio de requisições (requests) e respostas (responses), seguindo um conjunto de regras e protocolos como HTTP, HTTPS ou TCP, que garantem uma troca de dados segura e organizada.

Além disso, existem diferentes tipos de APIs, cada uma adequada a contextos específicos. A escolha do tipo de API pode variar conforme a necessidade do sistema, como, por exemplo, em uma aplicação web, que pode demandar uma API diferente da utilizada por um serviço de e-mail.

Portanto, decisões como o tipo de API, o protocolo de comunicação e a estrutura do projeto dependem diretamente da complexidade do problema a ser resolvido. Em projetos mais robustos, é comum o uso da arquitetura em camadas, onde cada camada possui uma responsabilidade bem definida. Em muitos casos, uma dessas camadas é dedicada ao controle das APIs, sendo frequentemente chamada de “controller”.

# 2. APIs e integrações

## 2.1 O que são as APIs?

As APIs (Application Programming Interface) são interfaces que permitem a comunicação entre diferentes sistemas. Ela funciona como um conjunto de regras que define como essa troca de informações deve acontecer.

Por meio de protocolos como HTTP, a API recebe requisições e responde com dados em formatos padronizados, como JSON. Assim, um sistema pode acessar informações ou funcionalidades de outro sem precisar saber como ele foi construído.

Entre os principais benefícios estão: integração facilitada, redução de complexidade e acesso sob demanda a dados, sem necessidade de armazenamento local.

## 2.2 Protocolos de APIs

Os protocolos de API definem as regras e convenções que orientam como deve ocorrer a comunicação entre sistemas. Eles determinam como as requisições devem ser feitas, quais métodos podem ser utilizados e como as respostas devem ser formatadas.

Esses protocolos funcionam como uma linguagem comum entre quem consome a API e quem a fornece. Além de padronizar a troca de dados, eles especificam o que pode ser requisitado, de que forma e em qual estrutura os dados serão retornados.

Entender os protocolos é essencial para garantir que a integração entre sistemas aconteça de forma segura, eficiente e compatível.

## 2.3 Arquiteturas e estilos de protocolo

Entre os estilos mais utilizados na criação de APIs, o REST (Representational State Transfer) é um dos mais comuns. Ele segue princípios que tornam as APIs mais simples, leves e baseadas em requisições HTTP. Uma API construída seguindo esses princípios é chamada de API RESTful.

Essas APIs operam com métodos como GET, POST, PUT e DELETE, que representam ações de leitura, criação, atualização e remoção de dados, respectivamente. O uso desses métodos ajuda a manter uma estrutura padronizada, facilitando a integração com diferentes sistemas.

Além do REST, existem outros estilos, como SOAP, que é mais complexo e baseado em XML, geralmente usado em sistemas corporativos, e GraphQL, que permite consultas mais flexíveis, onde o cliente pode definir exatamente quais dados deseja receber.

Cada arquitetura possui vantagens e se encaixa melhor em determinadas necessidades, sendo importante conhecê-las para escolher a mais adequada ao projeto.

## 2.4 Tipos de APIs

As APIs podem ser classificadas de acordo com o nível de acesso que oferecem e os públicos a que se destinam. Veja abaixo os principais tipos:

**Privada:** Usada apenas dentro da empresa. Garante mais controle e segurança interna.

**De Parceiros:** Compartilhada com parceiros específicos. Permite integrações comerciais sem abrir tudo ao público.

**Pública:** Disponível para qualquer desenvolvedor. Estimula a criação de novas soluções e amplia o alcance da empresa.

## 2.5 Ferramentas relacionadas a APIs

Há diversas ferramentas relacionadas a APIs, voltadas para diferentes funções enquanto trabalhando com elas. Essas opções de ferramentas abrangem testes de API, documentações, facilitadores de conexão e criadores de API.

**Postman:** usada para testar e simular requisições em APIs.

**Swagger (OpenAPI):** ajuda a documentar e visualizar como a API funciona.

**RapidAPI:** plataforma para buscar e conectar APIs prontas.

**Insomnia:** alternativa ao Postman, focada em testes e organização de requisições.

**Apigee (Google):** usada para criar, gerenciar e proteger APIs em ambientes empresariais.

**Spring Boot:** framework Java que facilita a criação rápida e organizada de APIs RESTful, automatizando configurações e acelerando o desenvolvimento.

## 2.6 Exemplos práticos de APIs

As APIs são amplamente usadas em diversos sistemas que temos contato no nosso dia a dia. No entanto, muitas vezes não percebemos o quão conetados estamos a elas, como quando usamos o google Maps ou vemos uma previsão do tempo. Alguns exemplos de API abaixo:

**Google Maps API:** permite integrar mapas em apps de entrega, mobilidade ou turismo.

**API do WhatsApp Business:** usada por empresas para automatizar mensagens e atendimentos.

**APIs de pagamento (ex: PayPal, Stripe, Mercado Pago):** permitem integrar sistemas de checkout a lojas virtuais.

**API de clima (ex: OpenWeather):** usada em apps que mostram previsão do tempo em tempo real.

# 3. CONCLUSÃO

Atualmente, o uso de APIs tem se tornado essencial para o desenvolvimento de sistemas modernos, permitindo que aplicações diferentes se conectem e troquem informações de forma padronizada. Como visto no texto, as APIs funcionam como conjuntos de regras que garantem essa comunicação, utilizando protocolos como HTTP, REST e SOAP para organizar as requisições e respostas.

Também foi mencionado que as APIs podem ser classificadas em tipos variados (privada, de parceiros e pública), cada uma adequada a diferentes necessidades. Além disso, o uso de ferramentas como Postman, Swagger e Spring Boot facilita a criação, o teste e a documentação dessas interfaces, acelerando o desenvolvimento e melhorando a qualidade das soluções.

Portanto, como discutido, a escolha adequada do tipo de API, do protocolo e das ferramentas está diretamente ligada à complexidade do projeto e aos objetivos do sistema, garantindo integrações mais eficientes e soluções mais flexíveis e escaláveis.

# 4. REFERÊNCIAS

BrowserStack*. Top 15 API Testing tools (free/open source API testing tools included).* Disponível em:https://www.browserstack.com/guide/open-source-api-testing-tools*.* Acesso em: 13 ago. 2025.

Red Hat. *API (Interface de Programação de Aplicações)*. Disponível em: https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces. Acesso em: 19 jan. 2023

ServiceNow. *O que é uma integração de APIs?* Disponível em: https://www.servicenow.com/br/workflows/creator-workflows/what-is-api-integration.html. Acesso em: 13 ago. 2025.