

Desenvolvimento Web

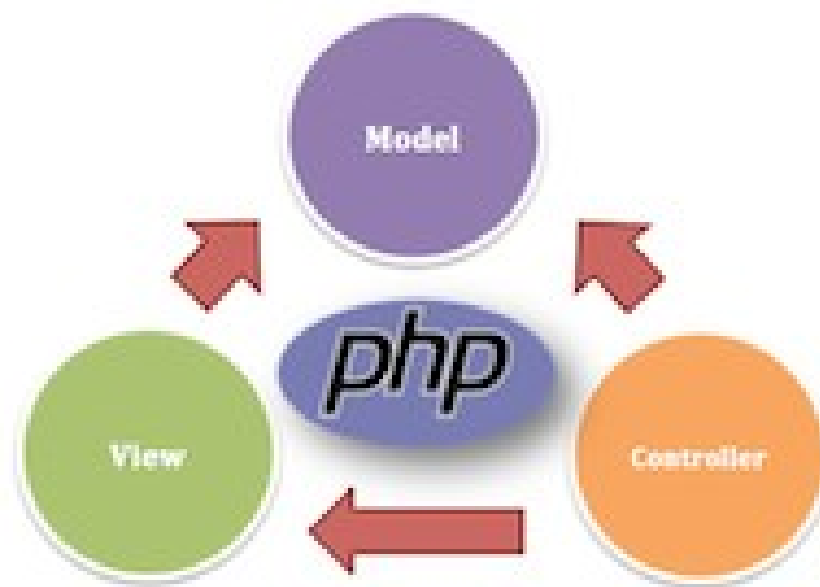
Prof. Daniel Di Domenico

API REST



O que já sabemos

- Envio de informações do cliente para o servidor
- Verbos/métodos HTTP:
 - GET
 - POST
- Formulários
- Persistência de dados
- Validações
- Orientação à objetos
- Arquitetura MVC (camadas)

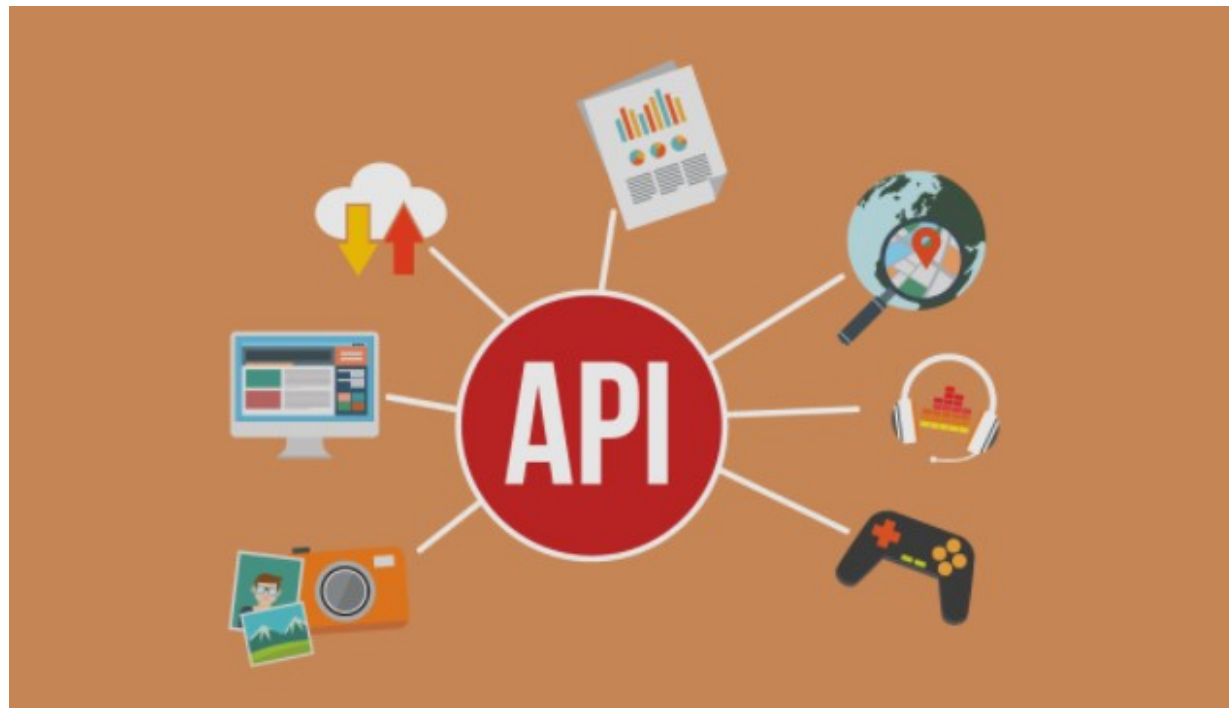


API REST

- API: Application Programming Interface (*Interface de Programação de Aplicações*)
 - Abstração de arquitetura de software que visa fornecer respostas a requisições utilizando dados padronizados
 - Mediador entre o usuário/cliente e os serviços WEB
- REST: Representational State Transfer (*Transferência de Estado Representacional*):
 - Transfere uma informação para um solicitante ou *endpoint*
 - A transferência ocorre utilizando o protocolo HTTP
 - **Formatos da informação:** JSON, XML, texto e outros

API REST

- Uma API pode fornecer dados para diversos solicitantes:
 - Tais solicitantes são denominados **clientes**
 - Um cliente, ao utilizar uma API, está **consumindo** seus recursos

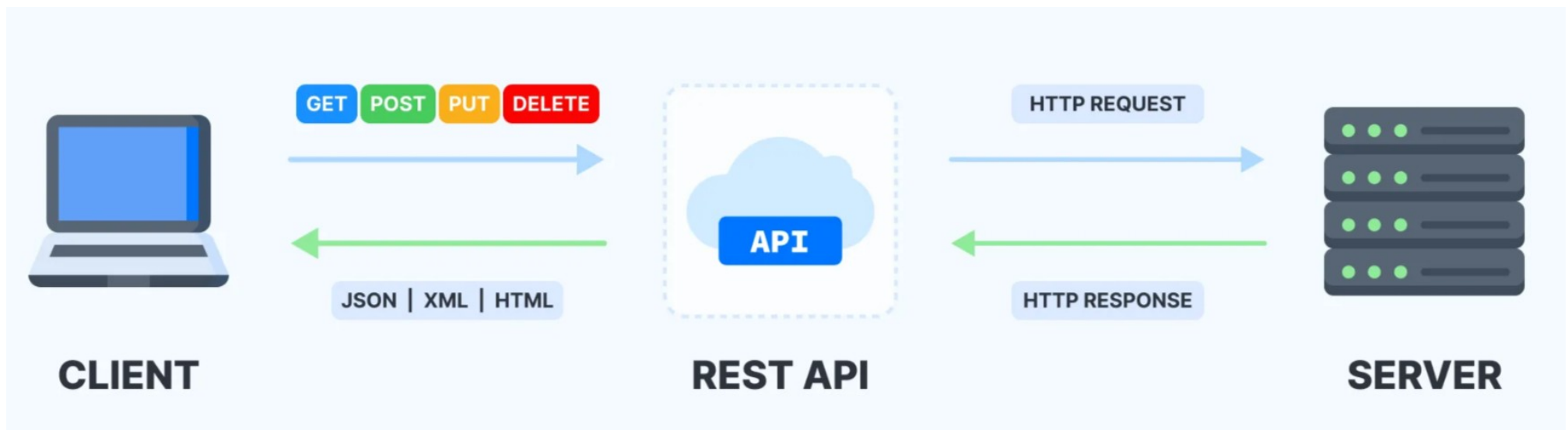


API REST

- Características de uma API Rest:
 - Disponibiliza um serviço Web na forma de back-end
 - Funciona no modo cliente-servidor
 - Permite a **comunicação entre aplicações** de forma rápida e segura
 - Sem estado (*stateless*)
 - **O servidor não armazena dados de estado**, sendo que todas as informações necessárias para atender a requisição devem ser providas pelo cliente
 - Permite dividir uma aplicação em camadas
 - Facilita a organização e manutenção

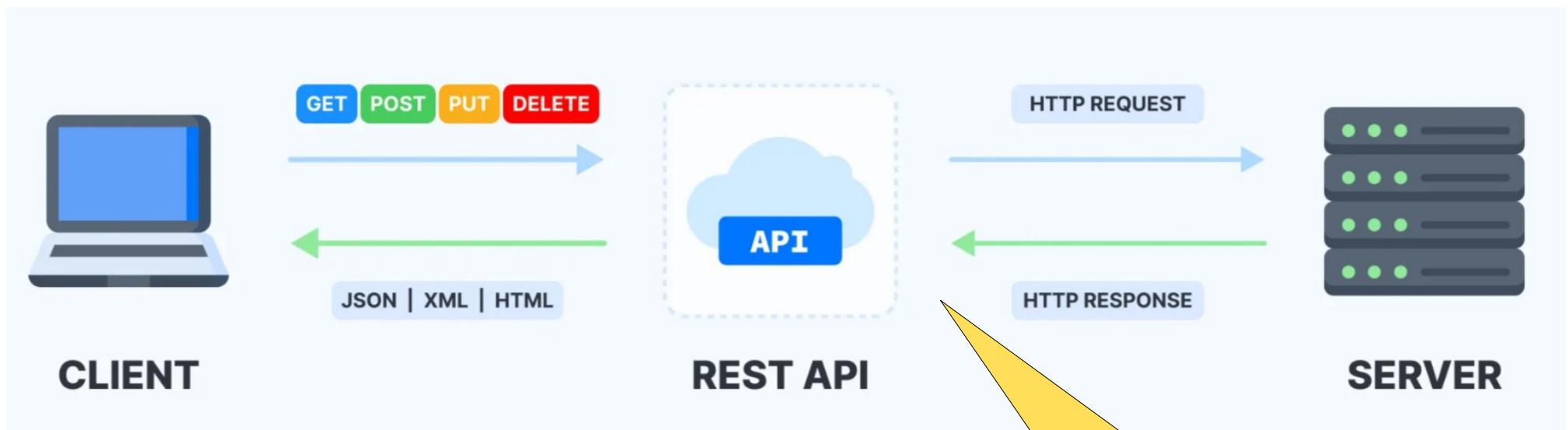
API REST

- Exemplo de fluxo de uma API REST:



API REST

- Exemplo de fluxo de uma API REST:



**Comunicação
utilizando HTTP**

API REST

- A API REST disponibiliza seus recursos através de **métodos/verbos** do protocolo HTTP
- Exemplos:
 - **GET**: utilizado para ler informações
 - **POST**: utilizado para inserir uma informação através da API
 - **PUT**: utilizado para atualizar uma informação através da API
 - **DELETE**: utilizado para deletar uma informação através da API
 - Outros:
<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

API REST

- O retorno após a requisição HTTP para a API REST utiliza códigos, sendo:
 - Sucesso (prefixo 2):
 - **200**: OK
 - **201**: Criado
 - **204**: Sem conteúdo
 - Erro no cliente (prefixo 4):
 - **400**: Erro na requisição
 - **404**: Recurso não encontrado
 - Erro no servidor (prefixo 5):
 - **500**: Erro interno no servidor
 - Outros: <https://www.httpstatus.com.br/>

API REST

- Vantagens:
 - Separação entre cliente e servidor
 - Pode-se alterar a interface (cliente) de uma aplicação sem mudar nada na API (servidor)
 - Escalabilidade
 - Pode-se adicionar mais clientes sem alterar a API
 - Para atender mais clientes, basta ter mais recursos de hardware
 - Multiplataforma, com retornos em JSON, XML....

