



**INSTITUTO
FEDERAL**

Paraná

Campus
Foz do Iguaçu

API REST com Java

Prof. Daniel Di Domenico

daniel.domenico@ifpr.edu.br

Introdução

Objetivo das aulas

- Vamos desenvolver um CRUD com API REST em Java:
 - **CRUD:** Create, Read, Update and Delete
 - Padrão MVC (Model, View, Controller)
 - A API REST é apenas backend, por isso contempla o Model e Controller
- Ferramentas necessárias:
 - Java
 - Spring Tool Suite
 - Postman
 - Banco de dados MySQL
 - phpMyAdmin ou outro cliente com suporte a MySQL
 - Linux (não é obrigatório, mas é recomendado)

MVC

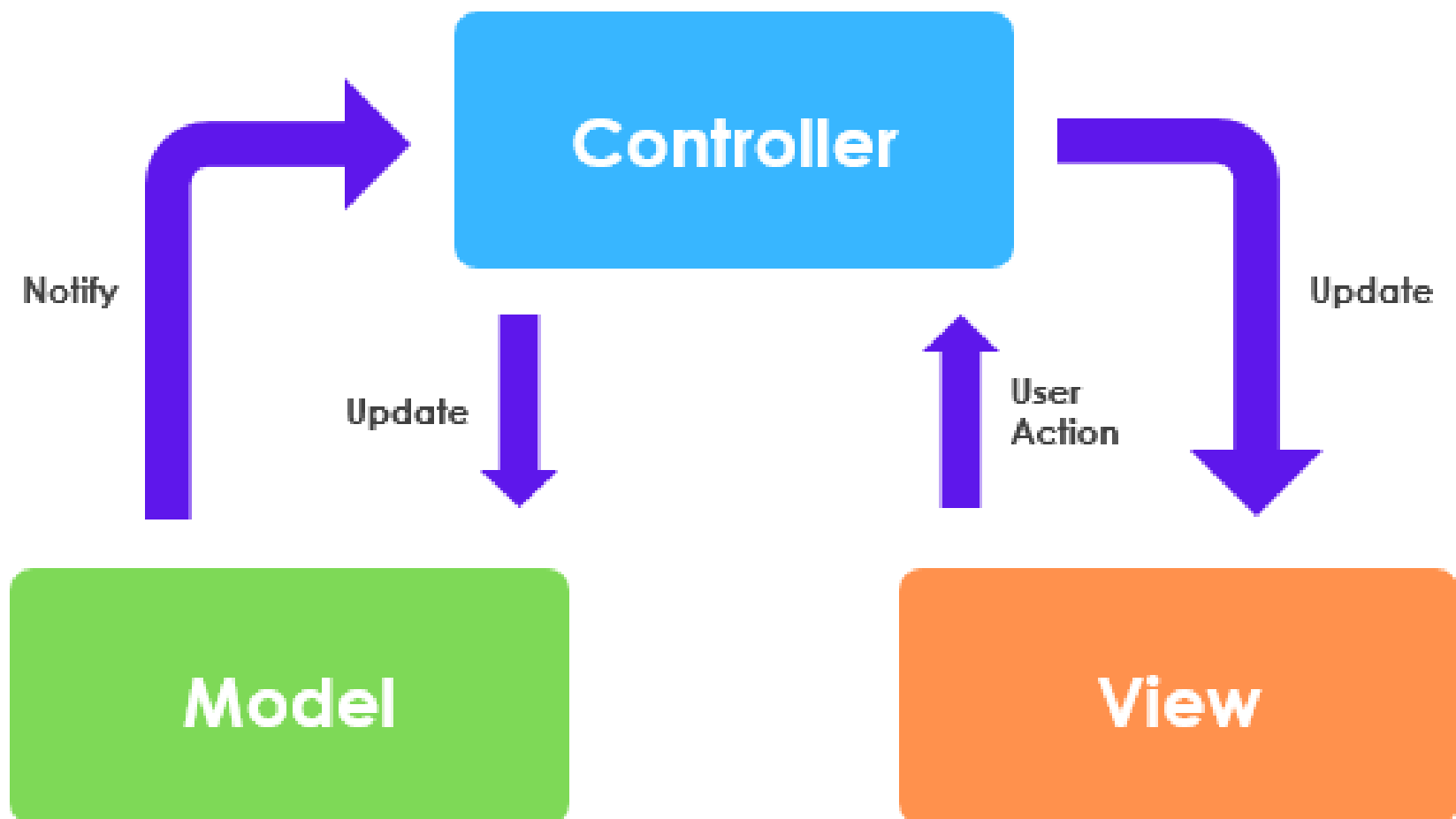
- Forma de estruturar um projeto, dividindo a implementação em (três) camadas com funções específicas
- Vantagens do MVC:
 - Facilita a manutenção e alteração do código (organização)
 - Isolamento das regras de negócio da lógica de apresentação
 - Diminui o acoplamento e aumenta a coesão das classes
 - Possibilita o reaproveitamento de classes e partes da implementação em projetos futuros
 - As camadas são independentes
 - Padrão que pode ser utilizado em diversos tipos de projetos
 - Desktop, Web e Mobile

Camadas do MVC



**INSTITUTO
FEDERAL**
Paraná

Campus
Foz do Iguaçu



Camadas do MVC

- **Model:** gerencia como os dados do sistema se comportam por meio de funções e das regras de negócio previamente definidas
- **View:** camada que apresenta as informações do sistema de forma visual ao usuário
 - Interage com o usuário por meio de botões, campos e mensagens, permitindo entradas (questionamentos) e saídas (respostas)
- **Controller:** liga as requisições enviadas pelo View com as respostas do Model
 - Permite a comunicação entre o Model e o View
 - É a única camada que conhece quem é o responsável por executar a operação que retornará uma resposta ao usuário

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 denominadas clientes
 - Um cliente, ao utilizar uma API, está **consumindo** seus recursos

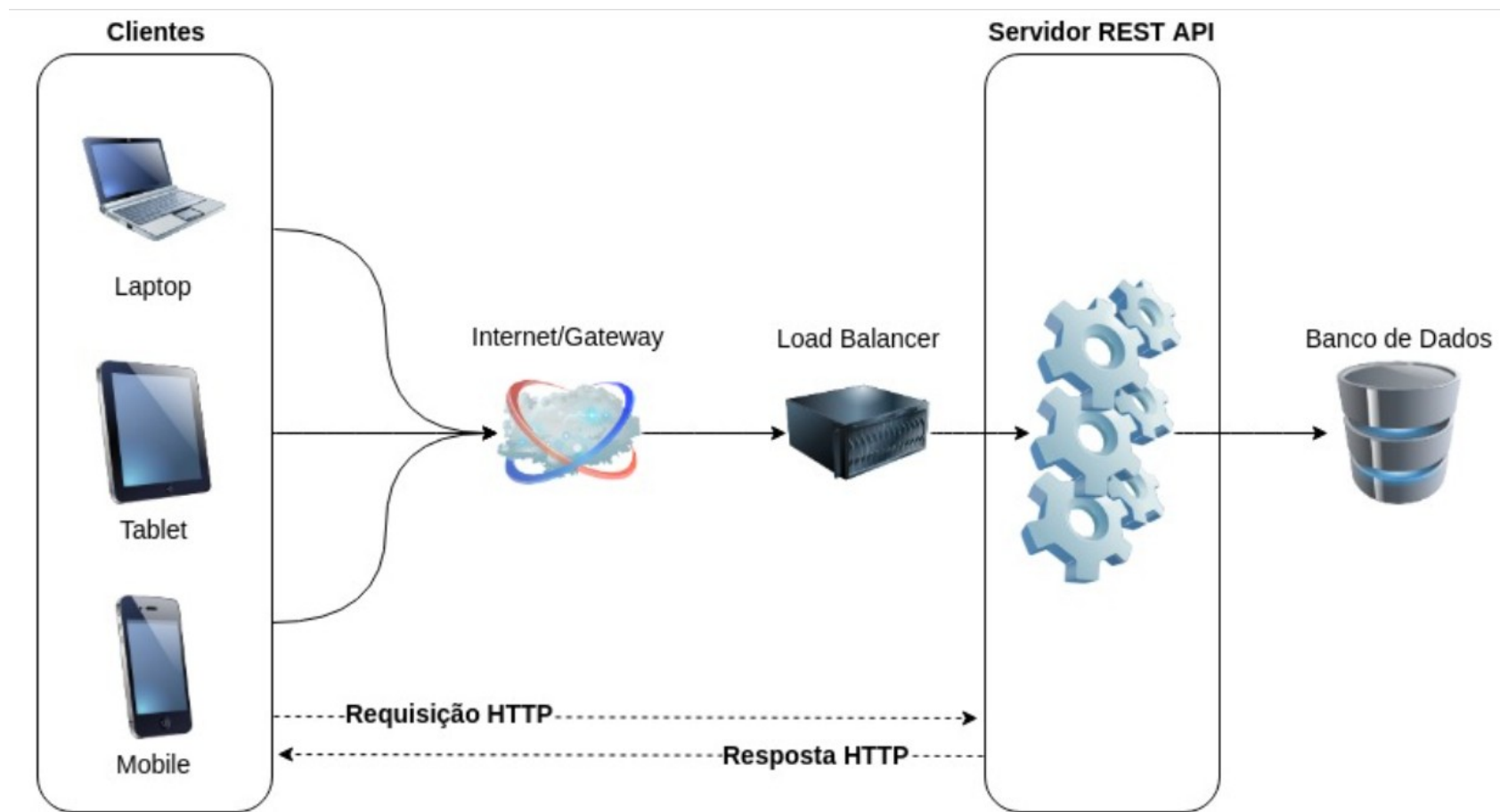


API REST

- 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
 - Normalmente utiliza-se o MVC

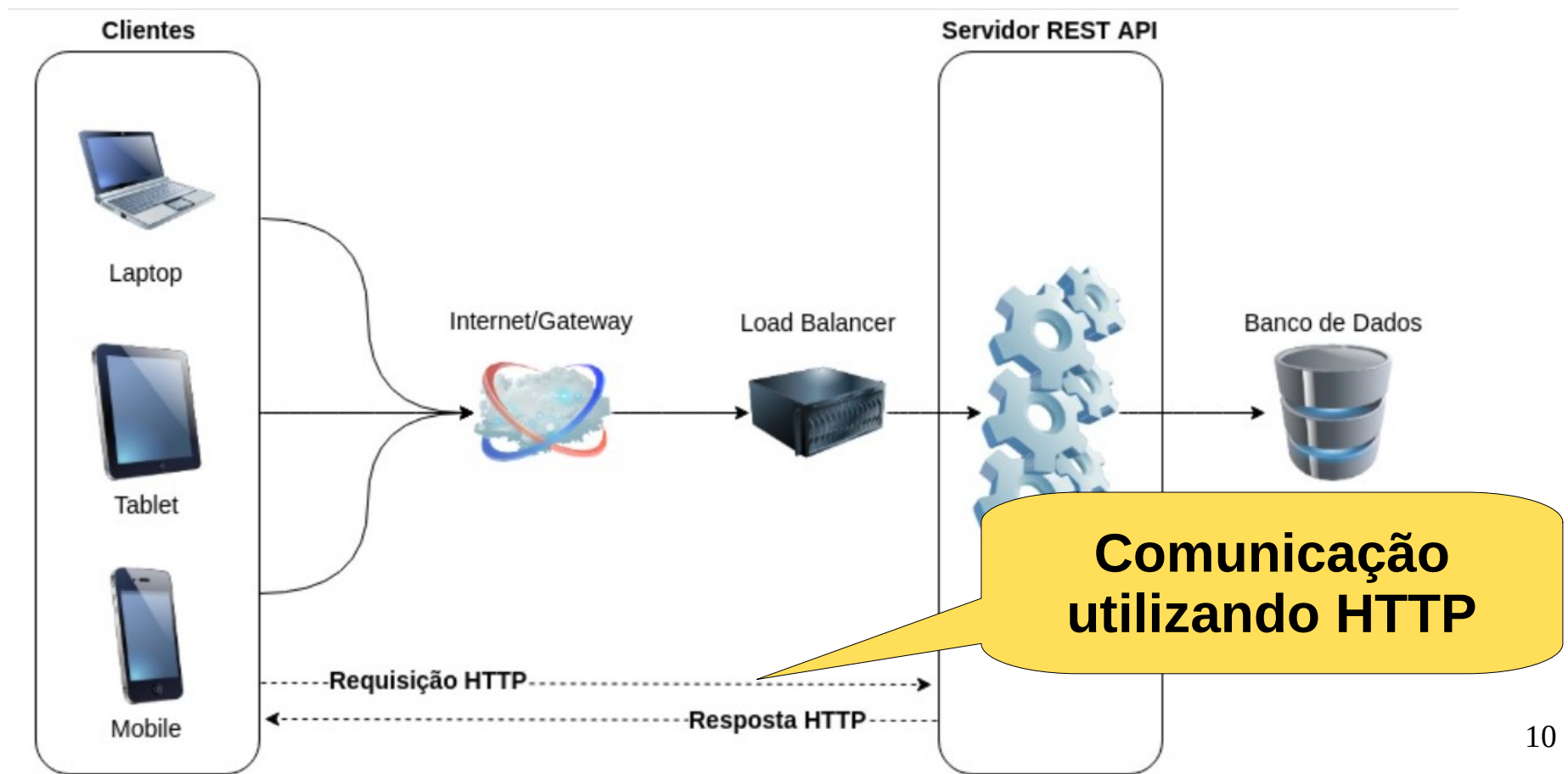
API REST

- Exemplo de camadas de uma API REST:



API REST

- Exemplo de camadas de uma API REST:



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....

API REST

- **JSON:**
 - *JavaScript Object Notation*
 - Formato de troca de dados e informações entre sistemas
 - Descreve **atributos** e **valores**
 - Exemplo:

```
{  
  "id":1,  
  "nome":"Fulano de Tal",  
  "endereco":"R. Qualquer"  
}
```



Java

- É um linguagem de programação que se destaca pela **portabilidade**:
 - Capacidade da mesma aplicação ser executada em diferentes sistemas operacionais
- Por que devo aprender Java? Java é utilizado por empresas?
 - <https://pesquisa.codigofonte.com.br/2022>



API REST com Java

- Nossa API com Java será desenvolvida utilizando o **Spring**:
 - É um ecossistema composto de vários projetos que ajudam o desenvolvedor a criar aplicações Java com simplicidade e flexibilidade
 - <https://spring.io/>
- Por que utilizar Spring?
 - Permite acesso a vários recursos (conexão com base de dados, projetos Web, API Rest, segurança...)
 - Tecnologia madura, popular e com alta empregabilidade, sendo utilizada por grandes empresas
 - Modularidade: requer a adição apenas dos recursos Spring que precisa-se em um projeto
 - Evolução constante e open source



API REST com Spring Boot

- **Spring Boot** é um framework do Spring que permite criar projetos auto configuráveis:
 - O usuário define o que precisa utilizar e o Spring Boot gera o projeto completo com tudo o que é necessário
 - **Inclusive com o servidor embarcado**
 - O usuário não precisa se preocupar com a maioria das configurações
 - Portanto, o Spring Boot facilita a criação de aplicações utilizando o ecossistema Spring



Classe proposta



**INSTITUTO
FEDERAL**

Paraná

Campus
Foz do Iguaçu



Classe proposta

- Implementação do CRUD será baseado na classe proposta:
 - O mesmo poderá ser incrementado futuramente
- Veremos conceitos de orientação a objetos e persistência desses objetos
- Em cada aula:
 - Novas implementações serão realizadas para completarmos o CRUD com todas as operações