Curso Presencial Programação Fullstack

Aula 08

Prof. MSc. Kelson | Senior Software Engineer

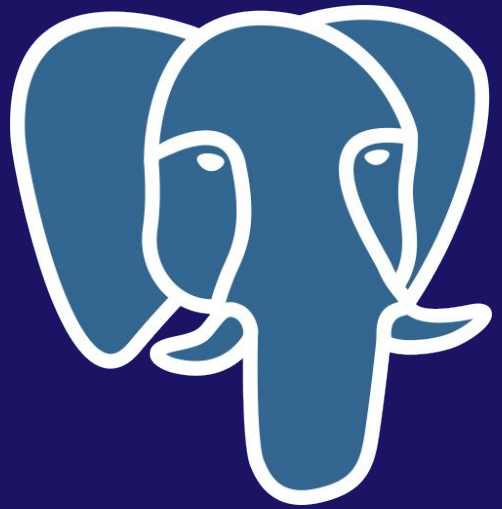
PROFKELSON.DEV

BORA CODAR?

PROFKELSON.DEV
BORA CODAR?

Introdução ao PostgreSQL

- PostgreSQL é um sistema de gerenciamento de banco de dados relacional (SGBD) avançado, open-source e com forte ênfase em conformidade com padrões e escalabilidade.
- Suporte a transações ACID *(tem que acontecer tudo ou nada)*, integridade referencial, tipos de dados customizáveis, e extensões programáveis.
- Usado para armazenar e recuperar dados em aplicações web, móveis, e empresariais.



PROFKELSON.DEV
BORA CODAR?

Introdução ao Docker

- Docker é uma plataforma open-source que automatiza o deploy de aplicações dentro de containers, permitindo que você crie, implante e rode aplicações em qualquer ambiente.
- Lançado em 2013 pela Docker, Inc.
- Rapidamente se tornou a tecnologia padrão para containers.

PROFKELSON.DEV
BORA CODAR?

O que é um Container?

- Um container é uma unidade padronizada de software que empacota código e todas as suas dependências para que a aplicação rode de maneira rápida e confiável de um ambiente computacional para outro.
- Características dos Containers:
  - Isolamento:
  - Cada container opera de forma independente e isolada, garantindo que as configurações de uma aplicação não interfiram em outras.
  - Leveza:
  - Compartilham o kernel do sistema operacional, o que os torna muito mais leves e rápidos para iniciar do que máquinas virtuais.
  - Portabilidade:
  - Contêineres podem ser executados em qualquer lugar, seja no laptop de um desenvolvedor, em um servidor local ou na nuvem, mantendo o ambiente de execução consistente.

PROFKELSON.DEV
BORA CODAR?

Instalação e Configuração do PostgreSQL

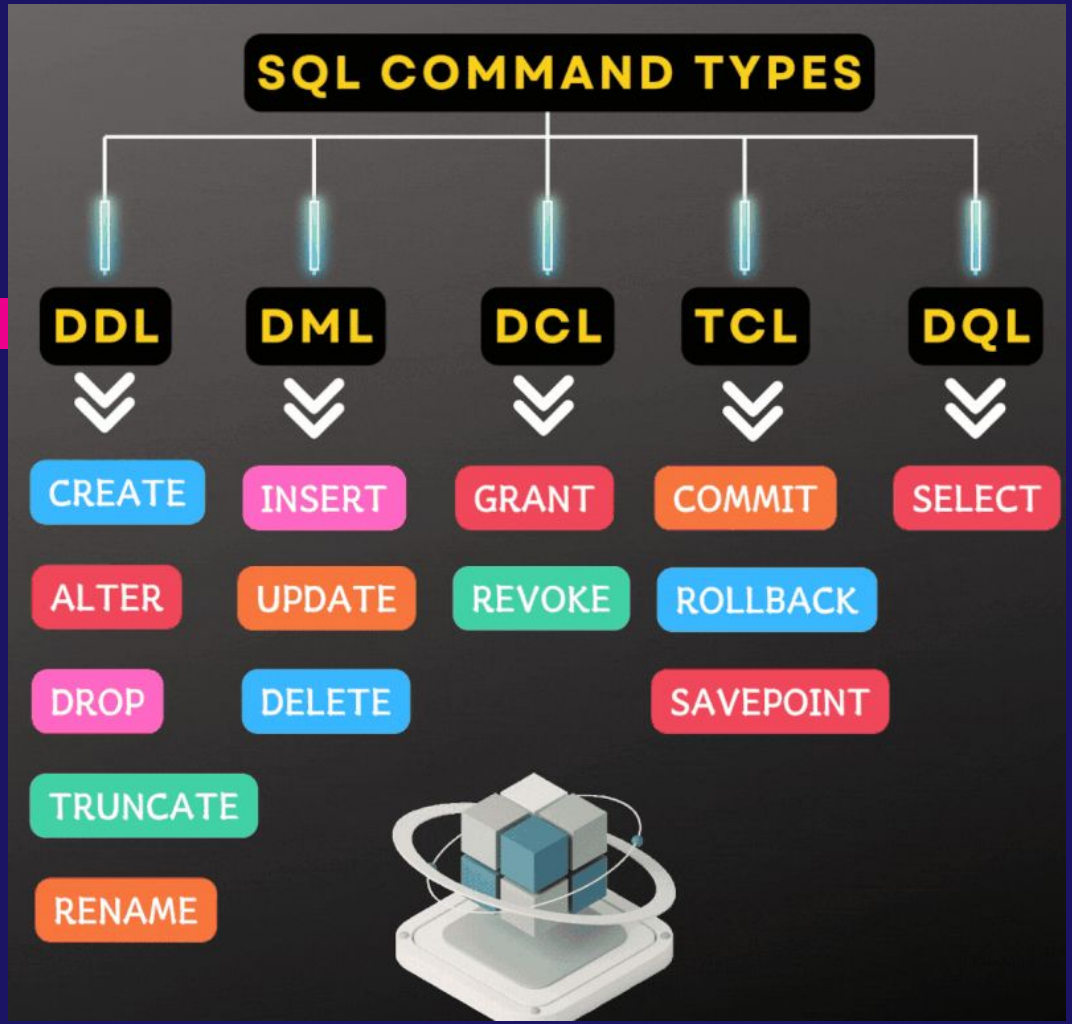
- Siga os passos do prof, para a instalação do PostgreSQL **(SERVER)**
  - Via Docker (vamos criar um container só para o postgres)
  - **docker run --name postgres-curso-fullstack -e POSTGRES_PASSWORD=123 -p 5439:5432 -d postgres**

- Também instale o DBeaver **(CLIENTE)**

PROFKELSON.DEV
BORA CODAR?

SQL Básico com PostgreSQL

- Introdução aos comandos SQL básicos (SELECT, INSERT, UPDATE, DELETE).
- **SQL (Structured Query Language)** é **a linguagem padrão para gerenciamento de dados em sistemas de banco de dados relacional como o PostgreSQL.**
- Usada para **inserir**, **consultar**, **atualizar** e **deletar** dados armazenados.


SQL COMMAND TYPES
DDL
DML
DCL
TCL
DQL
CREATE
ALTER
DROP
TRUNCATE
RENAME
INSERT
UPDATE
DELETE
GRANT
REVOKE
COMMIT
ROLLBACK
SAVEPOINT
SELECT

PROFKELSON.DEV
BORA CODAR?

SQL Básico: Comandos básicos

- **CREATE TABLE**
  - **Usado para criar uma nova tabela no banco de dados.**

```sql
CREATE TABLE clientes (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(50),
    email VARCHAR(50),
    telefone VARCHAR(15)
);
```

PROFKELSON.DEV
BORA CODAR?

SQL Básico: Comandos básicos

- **INSERT INTO**
  - **Insere dados em uma tabela.**

```sql
INSERT INTO clientes (nome, email, telefone)
VALUES ('João Silva', 'joao.silva@example.com', '99988-7766');
```

PROFKELSON.DEV
BORA CODAR?

SQL Básico: Comandos básicos

- **SELECT**
  - **Usado para consultar e recuperar dados de uma tabela.**

```sql
SELECT * FROM clientes;
SELECT nome, email FROM clientes WHERE id = 1;
```

PROFKELSON.DEV
BORA CODAR?

SQL Básico: Comandos básicos

- **UPDATE**
  - **Atualiza dados existentes em uma tabela.**

```sql
UPDATE clientes
SET email = 'joao.novoemail@example.com'
WHERE id = 1;
```

PROFKELSON.DEV
BORA CODAR?

SQL Básico: Comandos básicos

- **DELETE**
  - **Remove dados de uma tabela.**

```sql
DELETE FROM clientes WHERE id = 1;
```

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (1)

- Crie uma tabela produtos com as colunas id, nome, preço e estoque.
  - id -> SERIAL
  - nome -> VARCHAR(100)
  - Preco -> DECIMAL(10, 2)
  - estoque -> INT

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (1_1)

- Crie uma tabela consulta com as colunas id, data, horario, paciente e status.
  - id -> SERIAL
  - data_hora -> TIMESTAMP,
  - paciente -> VARCHAR(100),
  - status -> VARCHAR(50)

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (2)

- Insira três produtos na tabela produtos.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (2_2)

- Insira três consultas na tabela consulta.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (3)

- Selecione todos os campos dos produtos na tabela produtos.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (3_1)

- Selecione todos os campos das consultas na tabela consulta.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (4)

- Atualize o preço de um produto específico para 12.50 e o estoque para 95.
  - Filtre pelo id.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (4_1)

- Atualize o status de uma consulta específica para 'Concluída' e altere a data/hora para 2024-08-16 14:00. Filtre pelo id.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (5)

- Delete um dos produtos.
  - Filtre pelo id.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (5_1)

- Delete uma das consultas. Filtre pelo id.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (6)

- Crie uma tabela chamada fornecedores com as colunas id (chave primária, inteiro), nome (texto) e telefone (texto).

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (6_1)

- Crie uma tabela chamada medico com as colunas id (chave primária, inteiro), nome (texto) e especialidade (texto).

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (7)

- Insira dois fornecedores na tabela fornecedores.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (7_1)

- Insira dois médicos na tabela medico.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (7_1)

- Insira dois médicos na tabela medico.

Chave Primária

Chave Estrangeira

CONSULTA c

id	data_consulta	medico_id	paciente_id	descricao
1	2024-11-11	1	1	…
2	2024-11-12	1	2	…

PACIENTE p

id	nome	data_nascimento	plano_saude
1	José	1990-10-10	Unimed
2	Maria	1987-10-01	HapVida

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (8)

- Modifique a tabela produtos para incluir uma coluna fornecedor_id que cria uma chave estrangeira referenciando a coluna id da tabela fornecedores.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (8_1)

- Modifique a tabela consulta para incluir uma coluna medico_id que cria uma chave estrangeira referenciando a coluna id da tabela medico.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (9)

- Atualize a tabela produtos para definir o fornecedor_id para os produtos existentes.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (9_1)

- Atualize a tabela consulta para definir o medico_id para as consultas existentes.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (10)

- Selecione todos os campos dos produtos e o nome do fornecedor correspondente.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? (10_1)

- Selecione todos os campos das consultas e o nome do médico correspondente.

PROFKELSON.DEV
BORA CODAR?

Java

- Linguagem de programação orientada a objetos, usada para desenvolver aplicações.

- Características: Multithread, gerenciamento automático de memória (coleta de lixo), portabilidade (escreva uma vez, execute em qualquer lugar).

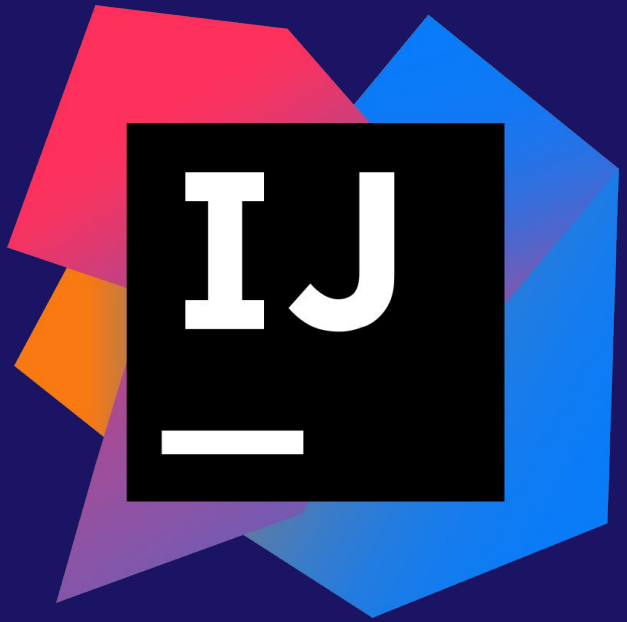
Java

PROFKELSON.DEV
BORA CODAR?

Java

- Vamos utilizar o **IntelliJ Community** para codar em Java :)
  - Segue com o prof!

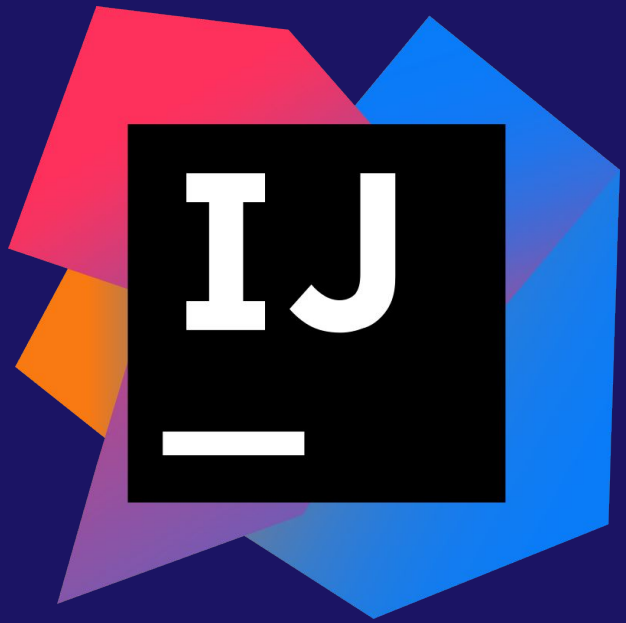
- Crie um novo projeto: **Aula08**



PROFKELSON.DEV
BORA CODAR?

Java

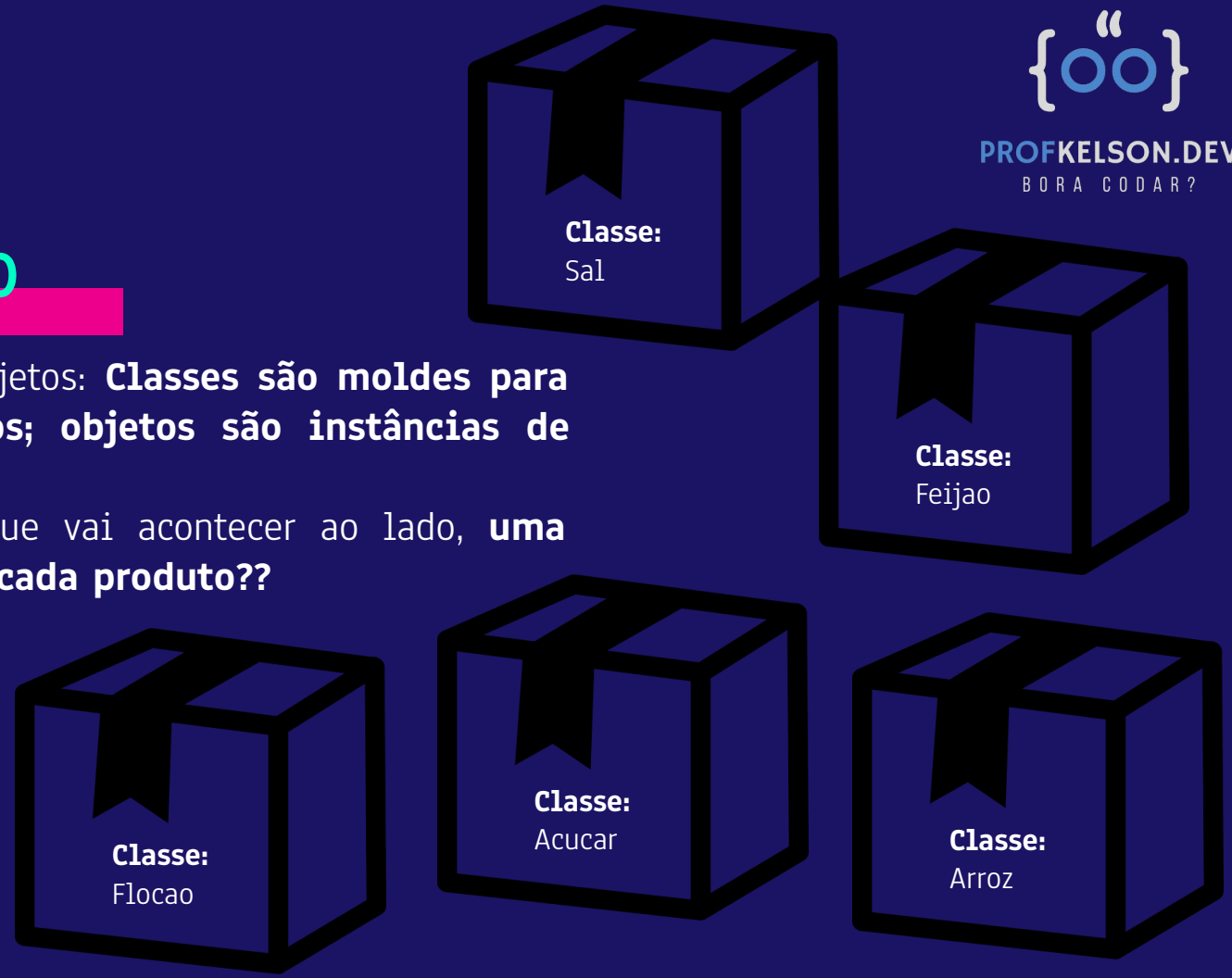
- Vamos de hello world?



PROFKELSON.DEV
BORA CODAR?

Java e OO

- Classes e Objetos: **Classes são moldes para criar objetos; objetos são instâncias de classes.**
- Olha só o que vai acontecer ao lado, **uma classe para cada produto??**



COM OO, NÃO!!!

VAMOS CRIAR UM "MOLDE / **CLASSE**" PARA CADA PRODUTO?

PROFKELSON.DEV
BORA CODAR?

Java e OO

- Molde (**classe**) produto:
  - 1 classe só, chamada “Produto”

java Copy code

```java
public class Produto {
    int id;
    String nome;
    double preço;

    public void exibirDetalhes() {
        System.out.println("Produto ID: " + id + ", Nome: " + nome + ", Preço: " +
    }
}
```

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? [COM PROF]

- Crie uma classe Produto com os atributos id, nome, preço e quantidadeEmEstoque. Inclua métodos para obter e definir o valor de cada atributo (getters e setters). Adicione um método exibirInformacoes() para mostrar todos os detalhes do produto.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? [COM PROF]

- Crie uma classe Fornecedor com os atributos id, nome e telefone. Em seguida, crie uma classe FornecedorPremium que herda de Fornecedor e adiciona o atributo taxaDesconto.

PROFKELSON.DEV
BORA CODAR?

VAMOS CODAR? [COM PROF]

- Defina uma interface Negociavel com um método negociar(). Implemente esta interface na classe Produto, adicionando lógica ao método negociar() para diminuir o quantidadeEmEstoque e imprimir uma mensagem.

PROFKELSON.DEV
BORA CODAR?

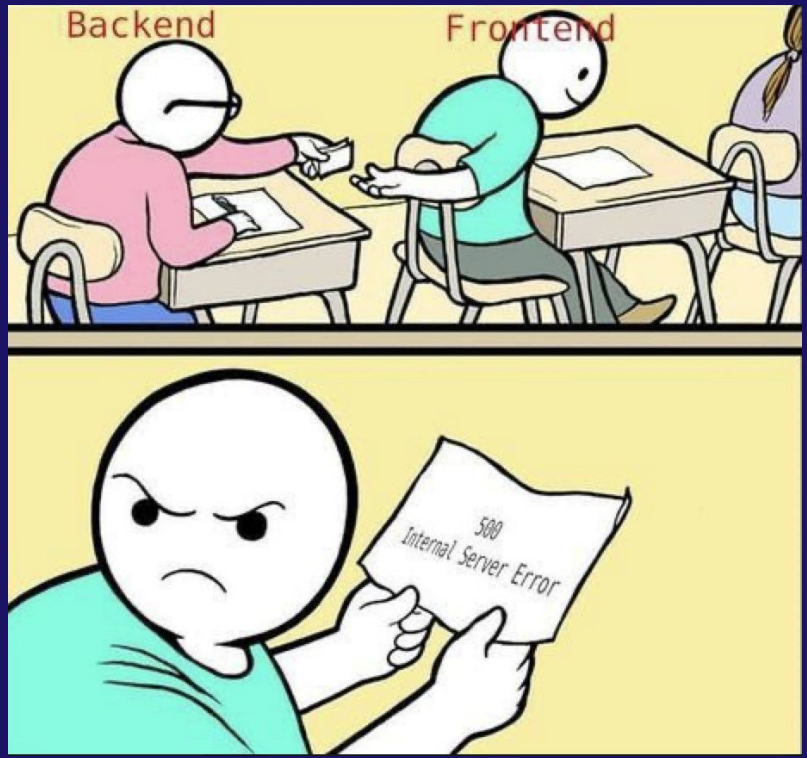
Back-end

- Estrutura que possibilita a operação do sistema
- Como as funções do meu sistema vão se comportar?
- Tudo que dá estrutura e apoio às ações do usuário
- Por trás daquela tela do site / app bonitinha... Quem é o responsável pela lógica interna?
  - Nosso amigo back-end :)

PROFKELSON.DEV
BORA CODAR?

Front-end

- Interface em que o usuário pode interagir com o sistema
- HTML, CSS, JavaScript são as principais tecnologias envoltas neste universo
- Facilitar a usabilidade do sistema
  - Já imaginou o usuário ter que entender as requisições HTTPs para usar um sistema?



Exemplo de Comunicação

John acessa
https://www.profkelson.dev

GET Request

O Servidor Web do site do prof recebe [GET] proveniente do *browser* do Bob.

GET Request

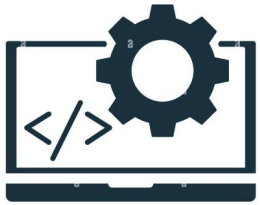
React

Front end

O frontend ficará responsável por fornecer a página web (*interface*) com informações sobre cursos.

GET Request

Java spring



BACK END

Mas quem busca o conteúdo cadastrado no banco de dados é o nosso amigo *back-end*

Atenção: Esta é apenas uma exemplificação, não necessariamente condiz com a realidade das tecnologias utilizadas pelo referido site.

Exemplo de Comunicação

Body Response com o JSON de cursos do prof kelson

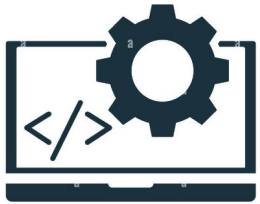
John acessa
https://www.profkelson.dev

Response 200 OK

Response 200 OK

Front end

Response 200 OK



BACK END


```
json                                                    Copy code

{
  "id": 1,
  "tituloDoSite": "Curso Presencial Programação Fullstack - <Turma_02 />",
  "logo": "profkelson-dev.png"
}
```

PROFKELSON.DEV
BORA CODAR?

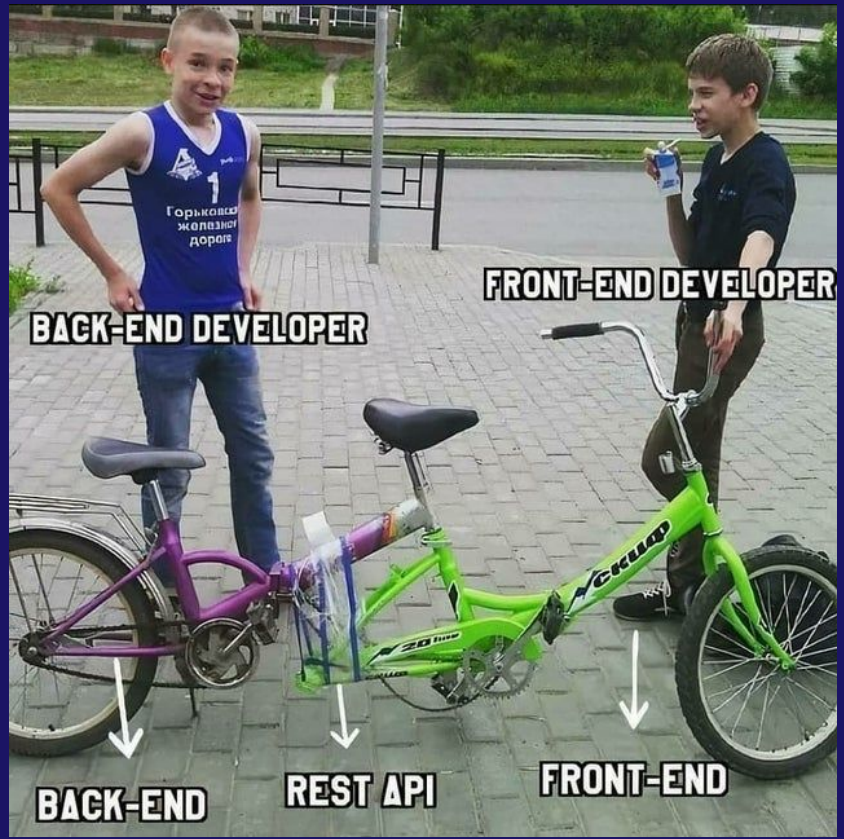
O que é API, prof???

- Application Programming Interface.
- Interface de Programação de Aplicação
- Uma interface que permite a comunicação entre sistemas diferentes.
- Conjunto de normas que possibilita a comunicação entre plataformas
- Através de protocolos e padrões.

PROFKELSON.DEV
BORA CODAR?

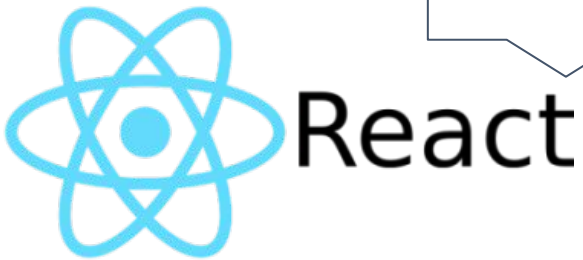
Rest API? RESTful?

- Também chamada de API RESTful.
- Interface de programação de aplicações.
- Em conformidade com a arquitetura REST.
- REST: Representational State Transfer. É um conjunto de princípios que devem ser seguidos ao projetar um sistema web.
- RESTful: é a forma de implementar os princípios ditados na arquitetura REST.



PROFKELSON.DEV
BORA CODAR?

Faaala, Javinha!! Meu considerado!



Tenho uma telinha bonitinha aqui que faz o cadastro de Alunos no Sistema, vou te mandar os dados desse Aluno, armazena esse aluno no banco de dados, por favor!

PROFKELSON.DEV
BORA CODAR?

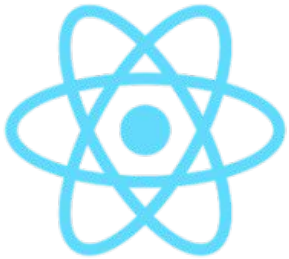
React

JS

Java

spring

PROFKELSON.DEV
BORA CODAR?



React

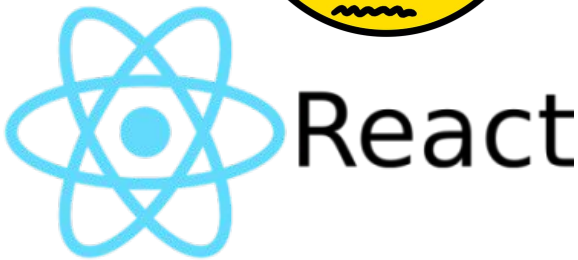
Sai pra lá! Seu frontzinho! Só sabe fazer telinha mesmo! Não fale de qualquer jeito comigo!

Java

JS

spring

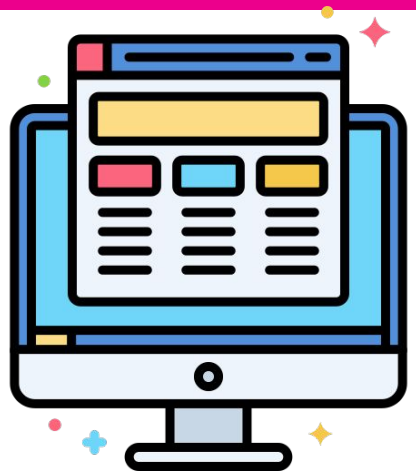
PROFKELSON.DEV
BORA CODAR?



No mínimo, fale comigo via arquitetura REST! OK? Não entendo seu *JSzinho*.

PROFKELSON.DEV
BORA CODAR?

Fluxo de Comunicação usando API REST



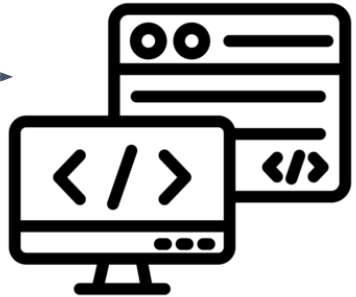
FRONT-END

React

POST, GET, PUT

http rest

200 OK
201 created



BACK-END
(REST API)

Java

spring

SQL

SQL

BANCO DE DADOS

PROFKELSON.DEV
BORA CODAR?

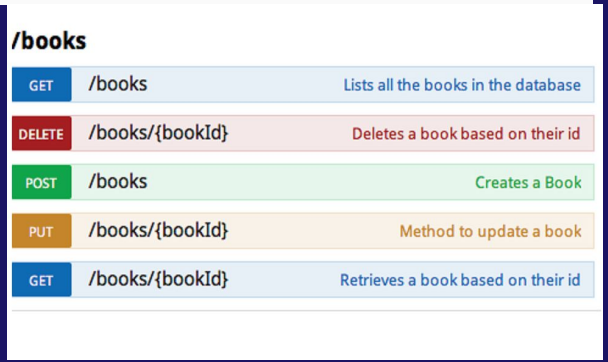
O que é HTTP?

- Hypertext Transfer Protocol
- Camada de Aplicação do modelo OSI.
- REGRAS da comunicação entre o cliente (ex: Navegador) e um servidor na internet.
- Requisição (request): Todo pedido que é enviado ao servidor.
- Resposta (response): Resposta do servidor, seguindo o fluxo do request.

PROFKELSON.DEV
BORA CODAR?

Verbos HTTP

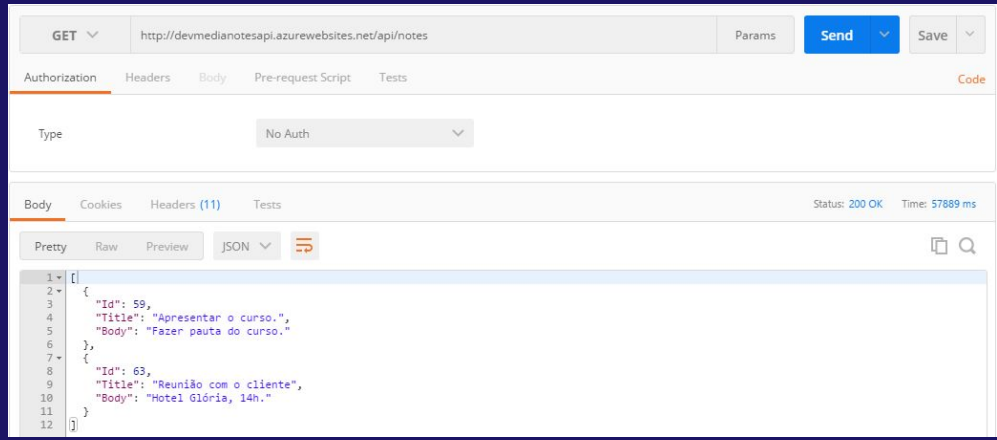
- Utilizados no desenvolvimento e/ou no consumo de serviços RESTful.
- Objetivo: o serviço vai prover uma URL base e os verbos HTTP tem a responsabilidade de indicar a ação que é requisitada pelo consumidor do serviço em questão.



PROFKELSON.DEV
BORA CODAR?

Obter alguma coisa

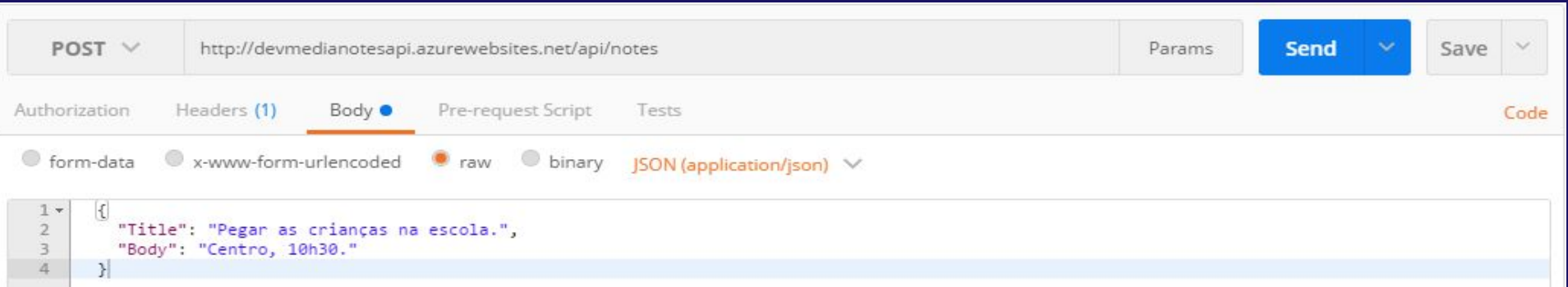
- **GET**
- Solicita a representação de um recurso específico.
- Requisições com GET retornam apenas dados.
- Ler dados, jamais alterar!



PROFKELSON.DEV
BORA CODAR?

Criar alguma coisa

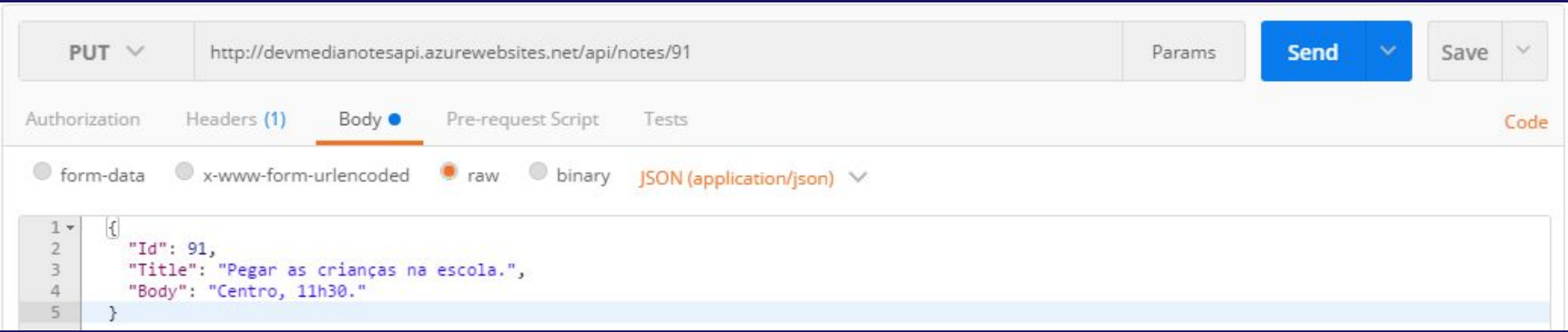
- **POST**
- Criação de recursos
- Adiciona informações a um recurso
- Geralmente é utilizado para enviar dados de formulários



PROFKELSON.DEV
BORA CODAR?

Atualiza, tudo, de algo

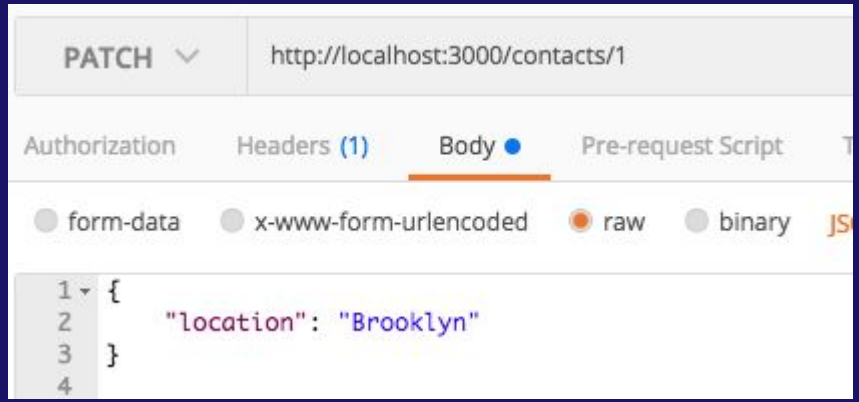
- **PUT**
- Atualiza informações de um recurso (update)



PROFKELSON.DEV
BORA CODAR?

Atualiza, parcialmente, algo

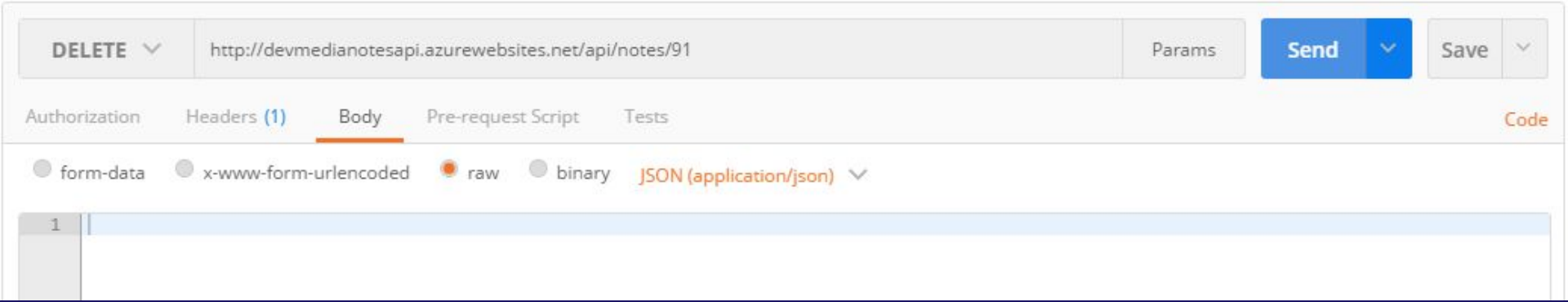
- **PATCH**
- Envia apenas o que precisa ser alterado, sem a necessidade de precisar enviar todos os dados.

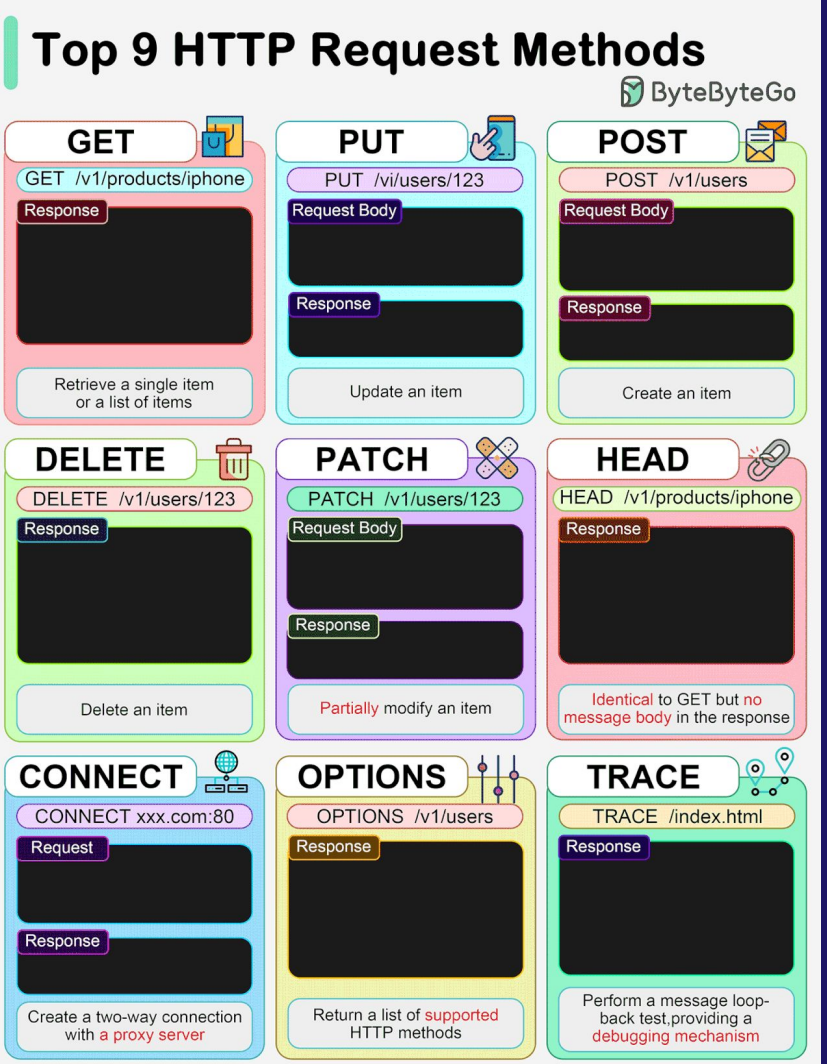


PROFKELSON.DEV
BORA CODAR?

Deleta, alguma coisa

- DELETE
- Remove um dado passado na URI





PROFKELSON.DEV
BORA CODAR?

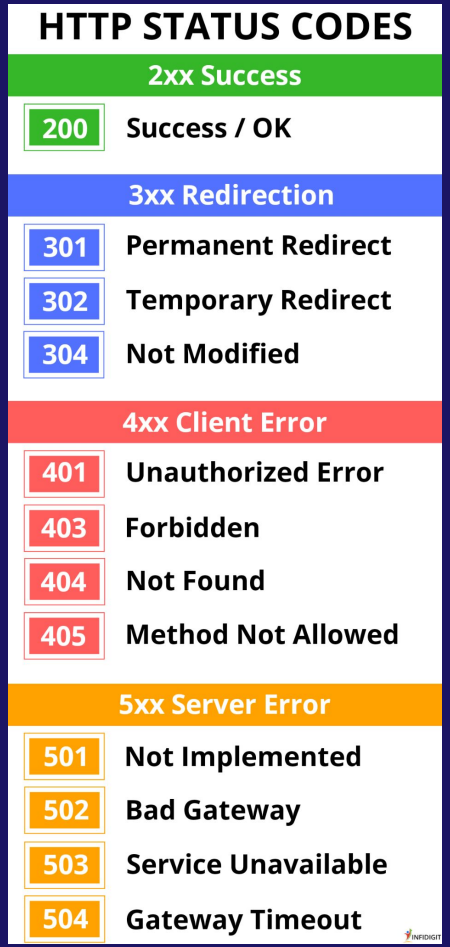
PROFKELSON.DEV
BORA CODAR?

Códigos de Status HTTP

- A cada response, o protocolo HTTP nos retorna um código de “status” referente a cada tipo de retorno.
- Mas, prof. Esses códigos são aleatórios?
- Não... São numerações pré-definidas e separadas por classes.

Códigos de Status HTTP

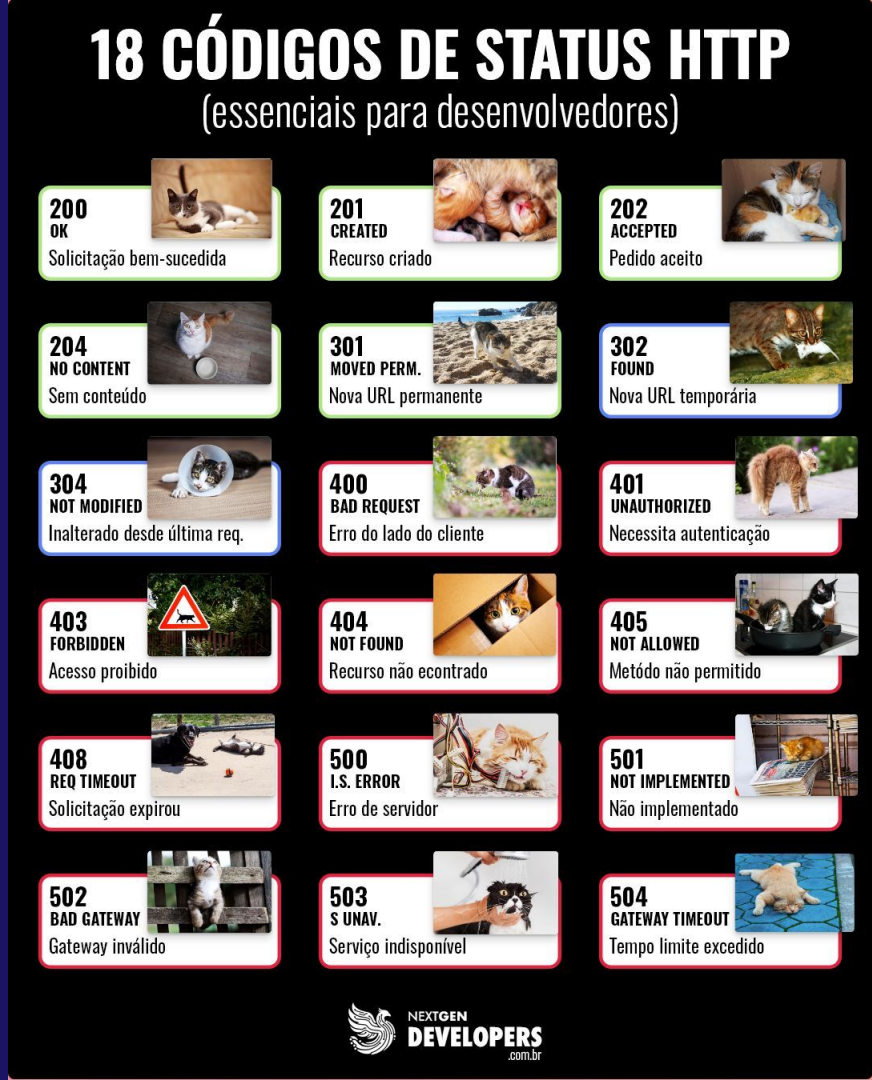
- Estão divididos em 5 classes:
- 100s: A solicitação iniciada pelo cliente HTTP continua...
- 200s: O pedido feito no request foi recebido, compreendido e processado pelo servidor.
- 300s: Redirecionamento. Um novo recurso foi substituído pelo recurso solicitado.
- 400s: Erros. Houve problema com o pedido.
- 500s: A solicitação foi aceita, mas aconteceu algum erro no servidor que impede o processamento completo da solicitação



PROFKELSON.DEV
BORA CODAR?

Códigos de Status HTTP

- Estão divididos em 5 classes:
- 100s: A solicitação iniciada pelo cliente HTTP continua...
- 200s: O pedido feito no request foi recebido, compreendido e processado pelo servidor.
- 300s: Redirecionamento. Um novo recurso foi substituído pelo recurso solicitado.
- 400s: Erros. Houve problema com o pedido.
- 500s: A solicitação foi aceita, mas aconteceu algum erro no servidor que impede o processamento completo da solicitação



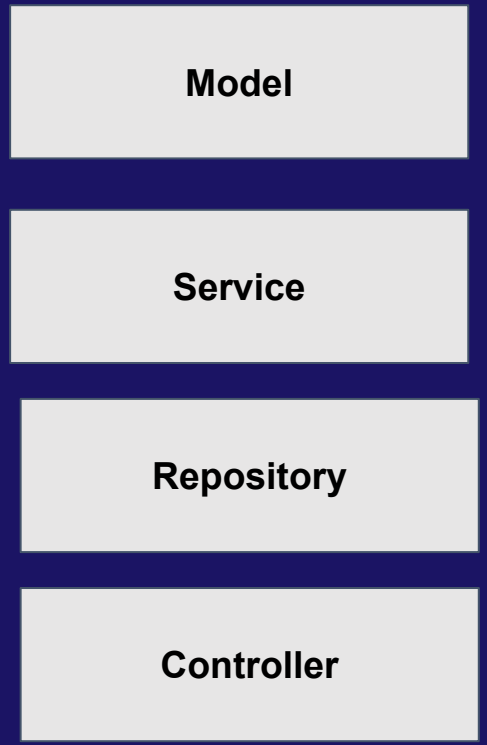
PROFKELSON.DEV
BORA CODAR?

Vamos começas a nossa API Backend?

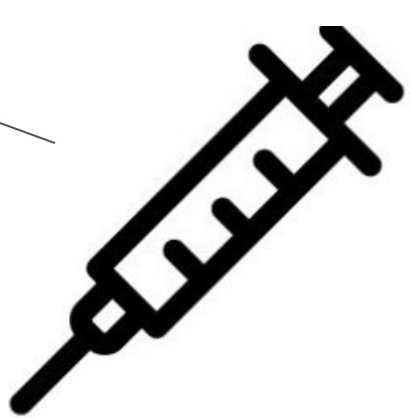
- Acompanha com o prof!
- start.spring.io
- Baixe o Insomnia REST Client
- Let's go!

Pacotes - Estrutura

- **Model:** Responsável pelas classes de entidade, conexão com o BD;
- **Services:** Separar as regras de negócio, regras da aplicação e regras de aplicação para que possam ser testadas e reutilizadas por outras partes;
- **Repository:** Responsável pela comunicação com os dados que o service precisa;
- **Controller:** Orquestrador. Recebe chamadas e retorna dados.



Injetando Dependências do Service no Controller



Injetando Dependências do Repository do Service