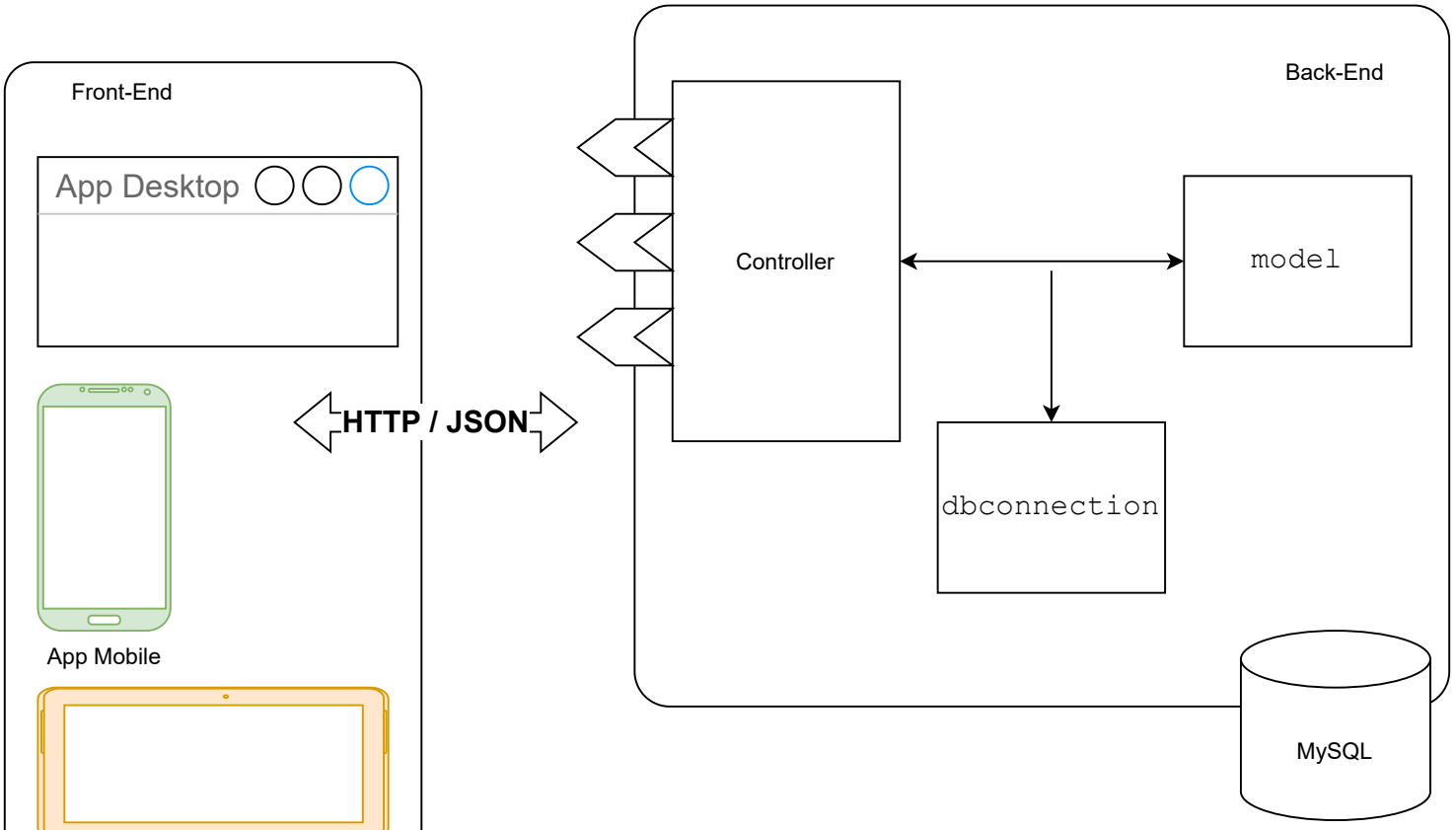


url (HTTP)

especificar um recurso no servidor e receber seu conteúdo

Componentes básicos

http://meusite.com.br:3000/produto

1. protocolo
http

2. localização do servidor (máquina)
meusite.com.br (ou um endereço IP)

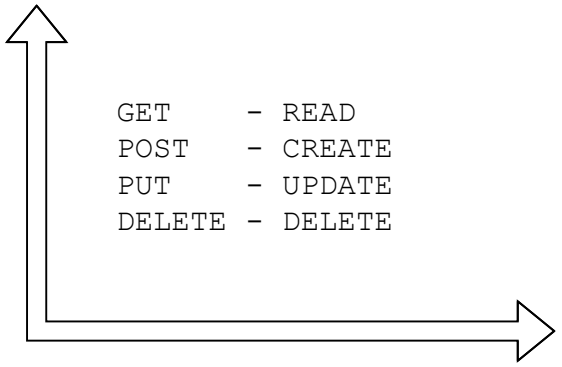
3. localização do serviço na máquina (porta)
3000

4. recurso
/produtos

Rotas/Endpoints

São definições de URL que englobam não apenas o caminho do recurso, como também a operação (método ou verbo) HTTP

```
GET      --> pegar/obter/consultar/acessar
POST     --> enviar/inserir
PUT      --> enviar para atualização
DELETE   --> remoção/apagar
```



Banco de Dados

Operações Fundamentais em Bancos de Dados

```
C - create - INSERT INTO
R - read   - SELECT
U - update - UPDATE
D - delete - DELETE
```

Richardson Maturity Level

Vamos implementar o Level 2

Posso ter a mesma URL, porém associar diferentes Métodos. Ex:

```
/produtos - GET    --> consulta de produtos
/produtos - POST   --> inclusão de novo produto
/produtos - PUT    --> alteração dos dados de um produto
/produtos - DELETE --> remoção de um produto da base
```

Estrutura hierárquica

```
GET
/produtos    --> todos os produtos da base
/produtos/1  --> apenas os dados do produto 1

/clientes    --> dados mínimos de apresentação do cliente (nome,email)
/clientes/1  --> todos os dados do cliente 1
/clientes/1/pedidos --> todos os pedidos do cliente 1
```

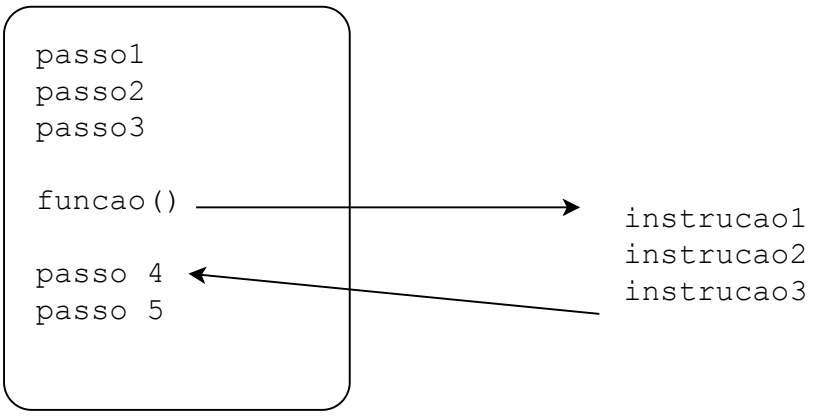
Escopo

1. Endpoint para retornar todos os produtos
/produtos

2. Endpoint para retornar detalhes de um determinado produto
/produtos/:id

3. Endpoint para retornar todas as categorias
/categorias

4. Endpoint para retornar todos os produtos por categoria
/categorias/:id




```
Execucao "normal"
passo1
passo2
passo3
instrucao1
instrucao2
instrucao3
passo4
passo5
```

```
Execucao "promisse"
passo1
passo2
passo3

passo4 / instrucao1
passo5 / instrucao2
instrucao 3
```

Problema da Pomisse

Quando eu sei que ela acabou? Ou mesmo se ela falhou ou não?

Pois é: uma promisse precisa de uma função callback (função que é chamada ao retornar da execução da promisse)

Exemplo

```
fecth("http://teste.com").then(res => trataResultado(res));
```

o que vai dentro do "then" é justamente o callback. Quando o fetch termina de executar, o resultado obtido por ele é chamado (ou executado) pela função definida na função **trataResultado**

Async/Await

"tornar a escrita das instruções Promisses mais próximas de uma forma declarativa". Como assim? Lembra do fetch acima?

```
async fetch(...) {
  ...
}
try{
   var res = await fetch("http://teste.com");
   trataResultado(res);
}
catch(erro){
   console.log("Deu ruim "+err);
}
```