

# Desenvolvimento de API - Farmácia Já

## Checkpoint 03

### Definição do escopo:

A ideia é que o usuário possa pesquisar o nome de um medicamento e descobrir quais farmácias o possuem, além de ter a opção de filtrar os resultados por bairro para facilitar encontrar o local mais próximo.

- Criar uma API para consulta de farmácias e medicamentos.
- Permitir pesquisa por nome do medicamento e filtro por bairro.
- Cadastrar, editar e remover farmácias e medicamentos.
- Estabelecer um relacionamento **muitos-para-muitos** entre farmácias e medicamentos.
- Retornar as informações em formato **JSON**.
- Armazenar os dados em um **banco PostgreSQL**.
- Criar uma **documentação dos endpoints** da API.

### Definição dos endpoints:

#### Usuários

##### GET /usuarios

- **Função:** Retorna a lista de todos os usuários cadastrados. Aceita filtros via query parameters.
- **Busca:** queries opcionais.
  - ?nome=... (busca parcial por nome)
  - ?cpf=... (busca exata por CPF)
  - ?email=... (busca exata por email)
  - ?bairro=... (filtra por bairro, graças à tabela de endereços!)
- **Exemplos:**
  - GET /usuarios
  - GET /usuarios?nome=Flávio
  - GET /usuarios?bairro=Centro
  - GET /usuarios?cpf=123.456.789-00

## GET /usuarios/:id

- **Função:** Retorna os dados de um usuário específico.
- **Busca:** params → id

GET /usuarios/5

Host: api.farmacija.com

Accept: application/json

### Resposta:

```
{
  "id": 5,
  "nome_completo": "Flávio Santos",
  "telefone": "(11) 98888-2222",
  "email": "flavio@email.com",
  "cpf": "123.456.789-00",
  "tipo_usuario": "cliente",
  "criado_em": "2025-10-25T17:30:00Z",
  "atualizado_em": "2025-10-25T17:30:00Z",
  "endereco": {
    "id": 1,
    "logradouro": "Rua das Flores",
    "numero": "50",
    "complemento": null,
    "bairro": "Centro",
    "cidade": "Muriaé",
    "estado": "MG",
    "cep": "36880-000"
  }
}
```

## POST /usuarios

- **Função:** Cadastra um novo usuário.
- **Busca:** body

Body (Corrigido com Endereço Normalizado):

JSON

```
{
  "nome_completo": "Flávio Santos",
  "telefone": "(11) 98888-2222",
  "email": "flavio@email.com",
  "senha": "SenhaSuperF0rte!",
  "cpf": "123.456.789-00",
  "endereco": {
    "logradouro": "Rua das Flores",
```

```
    "numero": "50",
    "bairro": "Centro",
    "cidade": "Muriaé",
    "estado": "MG",
    "cep": "36880-000"
  }
}
```

## PUT /usuarios/:id

- **Função:** Atualiza os dados de um usuário existente.
- **Busca:** params → id; body → novos dados.

Body (Exemplo de Atualização):

JSON

```
{
  "nome_completo": "Flávio Santos da Silva",
  "telefone": "(11) 97777-3333",
  "email": "flavio.silva@email.com",
  "cpf": "123.456.789-00",
  "endereco": {
    "logradouro": "Av. Paulista",
    "numero": "1000",
    "bairro": "Bela Vista",
    "cidade": "São Paulo",
    "estado": "SP",
    "cep": "01310-100"
  }
}
```

## PATCH /usuarios/:id

- **Função:** Atualiza apenas campos específicos de um usuário.
- **Busca:** params → id; body → apenas os campos a modificar.

Body (Exemplo):

JSON

```
{
  "telefone": "(11) 98888-9999",
  "endereco": {
    "logradouro": "Rua Nova Esperança",
    "numero": "200"
  }
}
```

## DELETE /usuarios/:id

- **Função:** Remove um usuário.
- **Busca:** params → id.

DELETE /usuarios/5

## Farmácias

### GET /farmacias

- Retorna todas as farmácias.
- **Busca:** queries opcionais.

?nome=... (busca parcial por nome)

?bairro=... (Implementa RF05!)

?entregas=true (filtra farmácias que fazem entrega)

?ativo=true (filtra farmácias ativas)

Exemplos:

GET /farmacias

GET /farmacias?bairro=Barra

GET /farmacias?nome=Central&entregas=true

### GET /farmacias/:id

- Retorna detalhes de uma farmácia específica.
- **Busca:** params → id

### POST /farmacias

- Cadastra uma nova farmácia.
- **Busca:** body

Body (Corrigido com Endereço Normalizado e campos do RF01):

JSON

```
{
  "nome": "Farmácia São José",
  "telefone": "(11) 98877-6655",
  "entregas": true,
  "cnpj": "12.345.678/0001-99",
  "horario_funcionamento": "Seg-Sex 08h-20h",
```

```
"endereço": {
  "logradouro": "Rua das Flores",
  "numero": "120",
  "bairro": "Centro",
  "cidade": "Muriaé",
  "estado": "MG",
  "cep": "36880-001"
}
```

## PUT /farmacias/:id

- Atualiza informações de uma farmácia.
- **Busca:** params → id; body → novos dados.

Body (Exemplo):

JSON

```
{
  "nome": "Farmácia São José - Matriz",
  "telefone": "(11) 97777-1111",
  "entregas": false,
  "cnpj": "12.345.678/0001-99",
  "horario_funcionamento": "Seg-Sab 08h-22h",
  "endereço": {
    "logradouro": "Rua das Flores",
    "numero": "120",
    "bairro": "Centro",
    "cidade": "Muriaé",
    "estado": "MG",
    "cep": "36880-001"
  }
}
```

## PATCH /farmacias/:id

- Função: Atualiza parcialmente os dados da farmácia.
- Busca: params → id; body → campos específicos.

Body (Exemplo):

JSON

```
{
  "telefone": "(11) 95555-3333",
  "entregas": true
}
```

## DELETE /farmacias/:id

- Remove uma farmácia.
- **Busca:** params → id.

## Medicamentos

### GET /medicamentos

- Retorna todos os medicamentos cadastrados. Aceita filtros via query parameters.
- **Busca:** queries opcionais

?nome=... (busca parcial por nome comercial)

?principio\_ativo=... (busca parcial por princípio ativo)

Exemplos:

GET /medicamentos

GET /medicamentos?nome=Dipirona

GET /medicamentos?principio\_ativo=Losartana

### GET /medicamentos/:id

- Retorna informações detalhadas de um medicamento.
- **Busca:** params → id.

### POST /medicamentos

- Cadastra um novo medicamento.
- **Busca:** body

Body (Completo, baseado no nosso SQL):

```
JSON
{
  "nome": "Dipirona Sódica 500mg",
  "principio_ativo": "Dipirona Monoidratada",
  "dosagem": "500mg"
}
```

## PUT /medicamentos/:id

- Atualiza informações de um medicamento.
- **Busca:** params → id; body → novos dados.

## PATCH /medicamentos/:id

- **Função:** Atualiza apenas campos específicos de um medicamento.
- **Busca:** params → id; body → apenas os campos a modificar.
- **Uso comum:** alterar a dosagem.

PATCH /medicamentos/5

```
{  
  "dosagem": "Dipirona Sódica 500mg"  
}
```

## DELETE /medicamentos/:id

- Remove um medicamento.
- **Busca:** params → id

## Relação Farmácias/Medicamentos

### GET /farmacias/medicamentos

- **Função:** Busca farmácias que possuem medicamentos.
- **Busca:** query → nome, bairro

Exemplo:

GET /farmacias/medicamentos?nome=dipirona&bairro=Centro

### POST /farmacias/:id/medicamentos

- **Função:** Associa um ou mais medicamentos a uma farmácia.

- **Busca:** params → id da farmácia; body → array de ids de medicamentos.

Body:  
JSON

```
{  
  "medicamentos_ids": [1, 3, 7]  
}
```

## Padronização de Respostas

### Sucesso

```
{  
  "mensagem": "Cadastro realizado com sucesso",  
  "status": 201  
}
```

### Erro

```
{  
  "erro": "Farmácia não encontrada",  
  "status": 404  
}
```

**OBS:** Muitos endpoints poderão contar com a implementação de filtros para auxiliar o usuário a encontrar o medicamento procurado.

### Funcionais:

- **RF01:** Cadastrar farmácias com informações como nome, endereço, bairro, telefone e se realiza entregas.
- **RF02:** Cadastrar medicamentos disponíveis no programa Farmácia Popular.
- **RF03:** Associar medicamentos às farmácias que os possuem.
- **RF04:** Permitir a busca de farmácias que tenham determinado medicamento.
- **RF05:** Permitir filtragem por bairro para facilitar a busca.
- **RF06:** Atualizar e remover farmácias e medicamentos quando necessário.

### Não funcionais:

- **RNF01:** O sistema deve ser desenvolvido em **Node.js** com **TypeScript**.



- **RNF02:** O banco de dados deve ser **PostgreSQL**.
- **RNF03:** A API deve seguir o padrão **RESTful**, retornando os dados em **JSON**.
- **RNF04:** O código deve ser organizado em **estrutura MVC** (controllers, services, routes, models).
- **RNF05:** Deve haver **tratamento de erros** e mensagens claras de retorno.
- **RNF06:** Informações sensíveis devem ficar em um arquivo **.env**.
- **RNF07:** A API deve possuir **documentação dos endpoints** (Postman).

