

**FACULDADE DE INFORMÁTICA E  
ADMINISTRAÇÃO PAULISTA**

**Preztech**

**Sistema de teleconsultas**

**Isis Macedo e Andrade  
Ana Clara Nascimento  
Annabelle Rosseto**

## SUMÁRIO

<b>1</b>	<b>Objetivo e escopo do projeto.....</b>	<b>3</b>
<b>2</b>	<b>funcionalidades.....</b>	<b>4</b>
<b>6</b>	<b>Endpoints.....</b>	<b>5 a 7</b>
<b>4</b>	<b>Modelo de Entidade-Relacionamento (MER).....</b>	<b>8</b>
<b>5</b>	<b>Diagrama de Classes.....</b>	<b>9</b>
<b>3</b>	<b>Protótipo do sistema.....</b>	<b>10</b>

## Objetivo e Escopo do Projeto

O principal objetivo deste projeto é desenvolver uma API (Application Programming Interface) RESTful que sirva como uma solução de back-end simples e modular para o gerenciamento de registros de pacientes, focando especificamente nas entidades de Consultas e Receitas Médicas.

A aplicação foi desenvolvida em **Java** utilizando o framework **Quarkus** que garante uma arquitetura leve, otimizada e adequada para microserviços. O design de software segue a separação de responsabilidades em camadas:

- Resource (API): Responsável por receber e responder às requisições HTTP (endpoints REST).
- Business Object (BO): Onde residem as regras de negócio e validações.
- Data Access Object (DAO): Camada de persistência, utilizando JDBC (Java Database Connectivity) para interagir diretamente com o banco de dados relacional (Oracle).

O foco principal deste projeto é validar a integração das tecnologias (Quarkus, REST, JDBC e Persistência Oracle). O escopo do projeto não contempla funcionalidades de:

- Interface gráfica (Front-end).
- Segurança robusta (Autenticação e Autorização de usuários).
- Validação de entrada de dados complexa ou tratamento de múltiplas conexões concorrentes.

## Descrição das Funcionalidades

O sistema implementa o ciclo completo de gerenciamento de dados (CRUD - Create, Read, Update, Delete) para as seguintes entidades:

1. **Gerenciamento de Consultas (/consultas):**
  - Permite a inserção, busca por ID, listagem de agendamentos por CPF e exclusão de consultas.
  - A busca por CPF retorna uma lista de todas as consultas associadas a um paciente específico.
2. **Gerenciamento de Receitas Médicas (/receitas):**
  - Permite a inserção e exclusão de receitas médicas.
  - Cada receita possui uma chave estrangeira, estando obrigatoriamente vinculada a um registro de **Consulta** existente.

**Tabela de Endpoints da API Restful**  
**(Base: <http://localhost:8080>)**

Esta tabela resume os endpoints para os recursos de **Consultas** e **Receitas Médicas**.

### Recurso: Consultas (/consultas)

Ação	Verbo HTTP	URI Completa	Corpo da Requisição (JSON)	Resposta Esperada	Observações
<b>Criar Nova Consulta</b>	POST	/consultas	<pre>{"cpfPaciente": "...", "dataHora": "..."} </pre>	201 Created (Retorna a consulta criada) / 400 Bad Request	Gerenciamento de agendamentos.
<b>Buscar por ID</b>	GET	/consultas/{id}	N/A	200 OK (Retorna a consulta) / 404 Not Found	
<b>Listar por Paciente (CPF)</b>	GET	/consultas/paciente/{cpf}	N/A	200 OK (Retorna array de consultas) / 404 Not Found	
<b>Atualizar Consulta</b>	PUT	/consultas/{id}	<pre>{"cpfPaciente": "...", "dataHora": "..."} </pre>	200 OK (Retorna a consulta atualizada) / 404 Not Found, 400 Bad Request	

<b>Excluir Consulta</b>	DELETE	/consultas/{id}	N/A	204 No Content / 404 Not Found	
-------------------------	--------	-----------------	-----	--------------------------------	--

## Recurso: Receitas Médicas (/receitas)

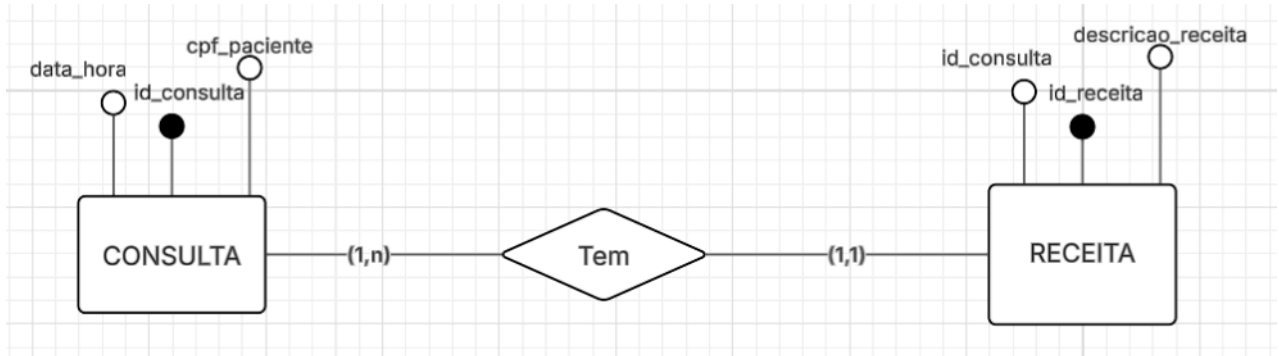
Ação	Verbo HTTP	URI Completa	Corpo da Requisição (JSON)	Resposta Esperada	Status de Implementação
<b>Criar Nova Receita</b>	POST	/receitas	{"idConsulta": ..., "descricaoReceita": "..."} 	201 Created (Retorna a receita criada) / 400 Bad Request	Vinculada a um idConsulta existente.
<b>Buscar por ID</b>	GET	/receitas/{id}	N/A	200 OK (Retorna a receita) / 404 Not Found	
<b>Atualizar Receita</b>	PUT	/receitas/{id}	Não aplicável	Não aplicável	<b>Não Implementado</b>
<b>Excluir Receita</b>	DELETE	/receitas/{id}	N/A	204 No Content / 404 Not Found	



## Códigos de Status Comuns

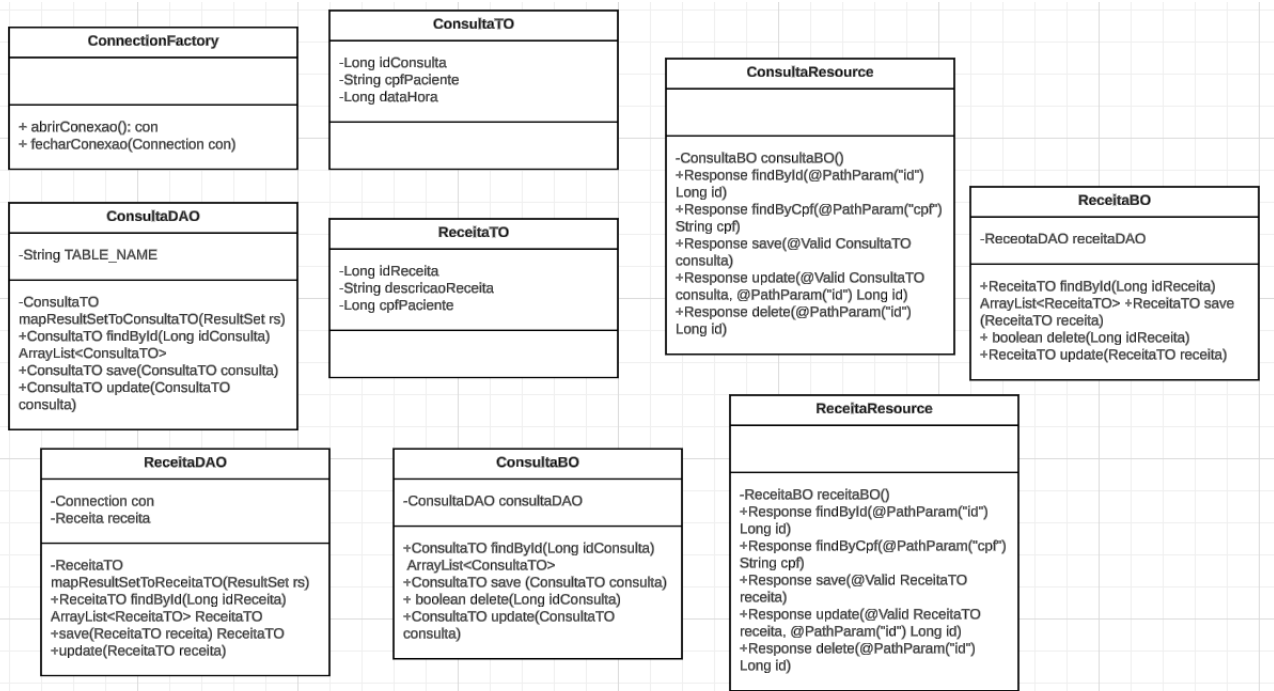
Categoria	Código	Descrição	Uso Típico
<b>Sucesso (2xx)</b>	200 OK	Ação bem-sucedida.	GET, PUT
	201 Created	Novo recurso criado com sucesso.	POST
	204 No Content	Requisição bem-sucedida, sem conteúdo para retornar.	DELETE
<b>Erro do Cliente (4xx)</b>	400 Bad Request	Requisição inválida, dados malformados/ausentes.	POST, PUT (erros de validação)
	404 Not Found	Recurso ou URI não encontrado.	Todos os verbos (ID inexistente)

### Modelo de Entidade-Relacionamento (MER)



## Diagrama de Classes





## Protótipo do Sistema

POST save x GET Mostrar x +

POST http://localhost:8080/receitas Send 201 Created 131 ms 144 B 2 Days Ago

Params Body Auth Headers 4 Scripts Docs Preview Headers 2 Cookies Tests 0/0 → Mock

URL PREVIEW  
http://localhost:8080/receitas

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

Raw

```
{
  "idReceita": 2,
  "idConsulta": 2,
  "descricaoReceita": "Cetorolaco 10mg, tomar um comprimido a cada 6 horas por 2 dias. Usar em caso de dor intensa."
}
```

POST save x GET Mostrar x +

GET http://localhost:8080/consultas/2 Send 200 OK 570 ms 77 B 2 Days Ago

Params Body Auth Headers 3 Scripts Docs Preview Headers 2 Cookies Tests 0/0 → Mock

URL PREVIEW  
http://localhost:8080/consultas/2

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

Preview

```
1 {
2   "idConsulta": 2,
3   "cpfPaciente": "99988877766",
4   "dataHora": "2026-06-01T10:00:00"
5 }
```

< > ↺ 🔒 preztech-561497.onrender.com/consultas/paciente/11122233344

📁 Home

Estilos de formatação

```
[
  {
    "idConsulta": 3,
    "cpfPaciente": "11122233344",
    "dataHora": "2026-03-15T10:00:00"
  }
]
```