

# MS Doações

A aplicação foi projetada para permitir doações através de uma API RESTful utilizando Spring Boot e .Net. para armazenar foi utilizado sql no dot net e no spring o banco local . Aqui estão alguns benefícios principais dessa aplicação:

- **Facilidade de Doação:** A aplicação oferece uma forma simples e direta para os usuários realizarem doações através de uma API RESTful. Isso facilita o processo de contribuição para causas específicas sem necessidade de interações complexas.
- **Integração com Diferentes Tecnologias:** Utilizando Spring Boot para a parte Java e .NET para a parte de login e armazenamento, a aplicação demonstra a capacidade de integrar diferentes tecnologias em um ambiente coeso. Isso permite aproveitar as vantagens específicas de cada plataforma para otimizar o desenvolvimento e o desempenho da aplicação.
- **Segurança e Confiabilidade:** A utilização de HTTPS para comunicação segura entre a aplicação e a API de doações garante que os dados dos usuários sejam protegidos durante as transações. Além disso, a implementação de autenticação por CPF e senha reforça a segurança no acesso e nas operações realizadas.
- **Escalabilidade:** Com o uso de frameworks como Spring Boot e .NET, a aplicação é construída com escalabilidade em mente. Isso significa que ela pode lidar com um aumento no número de usuários e volume de transações sem comprometer o desempenho.
- **Armazenamento Eficiente de Dados:** O uso de bancos de dados SQL tanto no .NET quanto no Spring assegura que todas as transações de doação sejam registradas e armazenadas de forma eficiente. Isso permite que os administradores da plataforma possam acessar e gerenciar facilmente o histórico completo de doações.
- **Documentação e Manutenção:** A documentação detalhada da API e a estrutura modular da aplicação facilitam a manutenção contínua e futuras expansões. Isso garante que a aplicação possa ser atualizada e melhorada conforme necessário para atender às necessidades dos usuários e das organizações beneficentes.

Esses benefícios combinados visam oferecer uma plataforma robusta e confiável para facilitar o ato de doar, promovendo maior transparência e eficiência nas contribuições para causas importantes.

## Documentação da API de Cliente

### Descrição

Esta API permite gerenciar informações de clientes.

```
1 https://sua-api.com/api/clientes
```

### Endpoints

## Criar Cliente

**Descrição:** Cria um novo cliente com os dados fornecidos.

- **Método HTTP:** POST
- **URL:** /clientes
- **Parâmetros de Requisição:**

- `cpf` (string, obrigatório): CPF do cliente.
- `nome` (string, obrigatório): Nome do cliente.
- `email` (string, obrigatório): Email do cliente.
- `senha` (string, obrigatório): Senha do cliente.
- `datainclusao` (string, opcional): Data de inclusão do cliente.
- `dataalteracao` (string, opcional): Data de alteração do cliente.
- 

- **Exemplo de Corpo da Requisição:**

```
1 {
2   "cpf": "12345678900",
3   "nome": "Fulano da silva",
4   "email": "fulano@example.com",
5   "senha": "senha123",
6   "datainclusao": "2024-06-30T00:00:00Z",
7   "dataalteracao": "2024-06-30T00:00:00Z"
8 }
```

- **Resposta de Sucesso:** Código HTTP 201 Created

```
1 {
2   "cpf": "123.456.789-00",
3   "nome": "Fulano de Tal",
4   "senha": "senha123",
5   "email": "fulano@example.com",
6   "datainclusao": "2024-06-30T00:00:00Z",
7   "dataalteracao": "2024-06-30T00:00:00Z"
8 }
```

- **Resposta de Erro:** Código HTTP 400 Bad Request

```
1 {
2   "timestamp": "2024-06-30T12:00:00Z",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Erro ao processar requisição",
6   "path": "/api/clientes"
7 }
```

## Login de Cliente [↗](#)

**Descrição:** Realiza a autenticação do cliente utilizando CPF e senha.

- **Método HTTP:** POST
- **URL:** `/login`
- **Parâmetros de Requisição:**
  - `cpf` (string, obrigatório): CPF do cliente.
  - `senha` (string, obrigatório): Senha do cliente.

- **Exemplo de Corpo da Requisição:**

```
1 {
```

```
2  "cpf": "12345678900",
3  "senha": "senha123"
4  }
```

**Resposta de Sucesso:** Código HTTP 200 OK

```
1  {
2    "cpf": "123.456.789-00",
3    "nome": "Fulano de Tal",
4    "email": "fulano@example.com",
5    "datainclusao": "2024-06-30T00:00:00Z",
6    "dataalteracao": "2024-06-30T12:00:00Z"
7  }
```

**Resposta de Erro:** Código HTTP 401 Unauthorized

```
1  {
2    "timestamp": "2024-06-30T12:00:00Z",
3    "status": 401,
4    "error": "Unauthorized",
5    "message": "Credenciais inválidas",
6    "path": "/api/clientes/login"
7  }
```

## Excluir Cliente por CPF [↗](#)

**Descrição:** Remove um cliente existente pelo CPF.

- **Método HTTP:** DELETE
- **URL:** `/clientes/{cpf}`
- **Parâmetros de Path:**
  - `{cpf}` (string, obrigatório): CPF do cliente.
- **Resposta de Sucesso:** Código HTTP 204 No Content
- **Resposta de Erro:** Código HTTP 404 Not Found

```
1  {
2    "timestamp": "2024-06-30T12:00:00Z",
3    "status": 404,
4    "error": "Not Found",
5    "message": "Cliente não encontrado para o CPF informado",
6    "path": "/api/clientes/123.456.789-00"
7  }
```

## Realizar Doação [↗](#)

- **URL:**
  - `/api/doacoes/realizar`
- **Método:**
  - POST
- **Descrição:**
  - Realiza uma doação com os dados fornecidos.
- **Parâmetros de Requisição:**

- `cpf` (String): CPF do doador.
- `valor` (double): Valor da doação.
- `app_name` (String): Nome da aplicação cliente.

- **Exemplo de Requisição:**

```
1 curl -X POST "http://localhost:8080/api/doacoes/realizar" -H "Content-Type: application/json" -d '  
2 {"cpf": "12345678900",  
3 "app_name": "MinhaApp",  
4 "valor": 100.0}'
```

### Códigos de Resposta:

- `200 OK` : Doação realizada com sucesso.
- `400 Bad Request` : Erro nos parâmetros da requisição.
- `500 Internal Server Error` : Erro interno no servidor.

### Exemplo de Resposta (sucesso):

```
1 {  
2   "id": 1,  
3   "cpf": "12345678900",  
4   "valor": 100.0,  
5   "dataHora": "2024-06-29T20:00:00Z"  
6 }
```

### Exemplo de Resposta (erro):

```
1 {  
2   "timestamp": "2024-06-29T20:05:00Z",  
3   "status": 400,  
4   "error": "Bad Request",  
5   "message": "Campo 'cpf' é obrigatório",  
6   "path": "/api/doacoes/realizar"  
7 }
```