

# CONTRATO DE REQUISIÇÃO

## Gestão de Contribuintes

Os seguintes serviços serão ofertados para a gestão de contribuintes:

- Cadastro de contribuintes
- Alteração de dados dos contribuintes
- Listagem de contribuintes
- Desativação lógica de contribuintes
- Geração de árvores genealógicas

## Dados Mantidos

Os dados dos contribuintes que serão mantidos:

- Nome (civil e social)
- CPF
- Filiação (multiparental)
- Endereço
- Telefones
- E-mails
- Salário
- Categoria

## Funcionamento

- **Cadastro de Contribuintes:** Permite ao Cliente cadastrar novos contribuintes, fornecendo os dados necessários através de requisições **POST**.

Ex:

**POST** /api/contribuintes

```
{
  "nome": "João Silva",
  "cpf": "123.456.789-00",
  "data_nascimento": "1990-05-15",
  "endereco": "Rua das Flores, 123",
  "cidade": "Exemploville",
  "telefone": "(99) 99999-9999"
}
```

Neste exemplo, uma requisição **POST** é feita para o endpoint /api/contribuintes com os dados do novo contribuinte no corpo da requisição.

- **Atualização de Dados:** Possibilita a alteração dos dados dos contribuintes já cadastrados através de requisições **PUT**.

Ex:

**PUT** /api/contribuintes/{id}

```
{
  "endereco": "Rua das Palmeiras, 456",
  "cidade": "Novaville",
  "telefone": "(88) 88888-8888"
}
```

Esta requisição **PUT** atualiza os dados do contribuinte identificado pelo {id} fornecido na URL. No exemplo, os dados de endereço, cidade e telefone estão sendo atualizados.

- **Consulta de Contribuintes:** Permite ao Cliente listar os contribuintes cadastrados, oferecendo recursos de filtragem e ordenação através de requisições **GET**.

Ex:

**GET**/api/contribuintes?cidade=Exemploville&ordenar\_por=data\_nascimento&direcao=asc

Neste exemplo, uma requisição **GET** é feita para o endpoint /api/contribuintes, filtrando os contribuintes pela cidade “Manaus”, ordenando-os por data de nascimento em ordem ascendente.

- **Desativação Lógica:** Permite desativar logicamente contribuintes sem remover seus dados do sistema, garantindo a integridade da informação e permitindo referência futura. Isso é realizado através de requisições **PATCH**.

Ex:

**PATCH** /api/contribuintes/{id}

```
{
  "ativo": false
}
```

Esta requisição **PATCH** desativa logicamente o contribuinte identificado pelo {id} fornecido na URL, alterando o estado ativo para false. Isso mantém os dados do contribuinte no sistema, mas marca-o como inativo.

- **Geração de Árvores Genealógicas:** Oferece a funcionalidade de gerar árvores genealógicas com base nos dados de filiação dos contribuintes cadastrados, utilizando requisições específicas.

Ex: **GET** /api/contribuintes/{id}/arvore-genealogica

Esta requisição busca a árvore genealógica do contribuinte identificado pelo {id}. A resposta será uma representação da árvore genealógica, possivelmente em um formato como JSON.