

# Introdução

Essa API fornece acesso a estatísticas e informações sobre usuários e estatísticas. Você pode obter estatísticas gerais, filtrar por cidade, força de trabalho empregada e nível de educação, entre outras. Também é possível buscar estatísticas de regiões específicas. A API é acessível por meio de chamadas HTTP.

## Autenticação

Essa API é protegida por autorização e requer um login e senha para acessar. O processo de autenticação é baseado em um token JWT (JSON Web Token). O token JWT é gerado após o login bem-sucedido e deve ser incluído em todas as solicitações subsequentes como um cabeçalho de autenticação. Isso garante que apenas usuários autenticados tenham acesso aos recursos protegidos pela API.

## Endpoints UserController

### 1. POST /user

Cadastra um novo usuário no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
name	string	Nome do usuário.
cpf	long	Cpf do usuário
email	string	E-mail do usuário
login	string	Login do usuário
password	string	Senha do usuário

#### Respostas

Código de status	Descrição
201	Usuário cadastrado com sucesso.
400	Parâmetros inválidos.

#### Exemplo de solicitação

bash

```
POST "/" HTTP/1.1
```

Content-Type: application/json

```
{
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

## 2. GET /user

Recupera as informações de um usuário cadastrado no sistema, utilizando paginação.

### Parâmetros

Parâmetro	Tipo	Descrição
-----------	------	-----------

### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

### Exemplo de solicitação

bash

```
GET "/" HTTP/1.1
```

### Exemplo de resposta

HTTP/1.1 200 OK

Content-Type: application/json

```
[
  {
    "name" : "João Silva",
    "cpf" : "12345678910",
    "email" : "joao.silva@gmail.com",
    "login" : "JSilva",
    "password" : "123456"
  }
]
```

```
        "role": "USER"
    }
]
```

## 2. GET /user/id

Recupera as informações de um usuário cadastrado no sistema, utilizando paginação.

### Parâmetros

Parâmetro	Tipo	Descrição
id	number	ID do usuário.

### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

### Exemplo de solicitação

bash

```
GET "/{id}" HTTP/1.1
```

### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

## 4. PUT /user/:id

Atualiza as informações de um usuário cadastrado no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
id	number	ID do usuário.
name	string	Nome do usuário.
cpf	long	Cpf do usuário
email	string	E-mail do usuário
login	string	Login do usuário
password	string	Senha do usuário

#### Respostas

Código de status	Descrição
200	Informações do usuário atualizadas com sucesso.
400	Parâmetros inválidos.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
PUT "/user/{id}" HTTP/1.1
```

```
Content-Type: application/json
```

```
{
  "id" : 12354,
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

#### 5. DELETE /user/:id

Remove um usuário cadastrado no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
id	number	ID do usuário.

#### Respostas

Código de status	Descrição
204	Usuário removido com sucesso.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
DELETE "/user/{id}" HTTP/1.1
```

## 6. GET /user/:name

Recupera as informações de um usuário cadastrado no sistema através de seu nome.

#### Parâmetros

Parâmetro	Tipo	Descrição
name	string	nome do usuário.

#### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
GET "/name/{name}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
```

```
    "id" : 12354,  
    "name" : "João Silva",  
    "cpf" : "12345678910",  
    "email" : "joao.silva@gmail.com",  
    "login" : "JSilva",  
    "password" : "123456",  
    "role": "USER"  
}
```

## 7. GET /user/:cpf

Recupera as informações de um usuário cadastrado no sistema através de seu cpf.

### Parâmetros

Parâmetro	Tipo	Descrição
cpf	long	Cpf do usuário.

### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

### Exemplo de solicitação

bash

```
GET "/cpf/{cpf}" HTTP/1.1
```

### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{  
    "id" : 12354,  
    "name" : "João Silva",  
    "cpf" : "12345678910",  
    "email" : "joao.silva@gmail.com",  
    "login" : "JSilva",  
    "password" : "123456",  
    "role": "USER"  
}
```

```
}
```

## 8. GET /user/:login

Recupera as informações de um usuário cadastrado no sistema através de seu login.

### Parâmetros

Parâmetro	Tipo	Descrição
login	string	login do usuário.

### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

### Exemplo de solicitação

bash

```
GET "/login/{login}" HTTP/1.1
```

### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id" : 12354,
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

## 9. GET /user/:email

Recupera as informações de um usuário cadastrado no sistema através de seu email.

### Parâmetros

Parâmetro	Tipo	Descrição
email	string	email do usuário.

#### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
GET "/email/{email}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id" : 12354,
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

### 10. GET /user/:email

Recupera as informações de um usuário cadastrado no sistema através de seu email, utilizando uma forma alternativa.

#### Parâmetros

Parâmetro	Tipo	Descrição
email	string	email do usuário.

#### Respostas



Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
GET "/email2/{email}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id" : 12354,
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

## Endpoints StatisticsController

### 1. POST /statistics

Cadastra uma nova estatística no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
city	string	Nome da cidade
pib	float	Pib da cidade
workingForce	float	Força de trabalho empregada
educationLevel	float	Nível de educação
poverty	float	Nível de pobreza

salary	float	Faixa salarial
povertyPercentage	float	Porcentagem de pobreza
reristrationDate	LocalDate	Data de registro das estatísticas

#### Respostas

Código de status	Descrição
201	Estatísticas cadastradas com sucesso.
400	Parâmetros inválidos.

#### Exemplo de solicitação

bash

POST "/" HTTP/1.1

Content-Type: application/json

```
{
  "city" : "Tanabi",
  "pib" : 402315.00
  "workingForce" : 80.00,
  "educationLevel" : 95.00,
  "poverty" : 2365.00,
  "salary": 2000.00,
  "povertyPercentage" : 15.5
}
```

## 2. GET /statistics

Recupera as estatísticas das cidades cadastrada no sistema, utilizando paginação.

#### Parâmetros

Parâmetro	Tipo	Descrição
-----------	------	-----------

#### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.

404	Estatísticas não encontradas.
-----	-------------------------------

#### Exemplo de solicitação

bash

```
GET "/" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
[
  {
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

## 2. GET /statistics/id

Recupera as estatísticas das cidades cadastrada no sistema pelo id.

#### Parâmetros

Parâmetro	Tipo	Descrição
id	number	ID do usuário.

#### Respostas

Código de status	Descrição
200	Informações do usuário retornadas com sucesso.
404	Usuário não encontrado.

#### Exemplo de solicitação

bash

```
GET("/{id}" HTTP/1.1
```

#### Exemplo de resposta

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "city" : "Tanabi",
  "pib" : 402315.00,
  "workingForce" : 80.00,
  "educationLevel" : 95.00,
  "poverty" : 2365.00,
  "salary": 2000.00,
  "povertyPercentage" : 15.5
}
```

#### 4. PUT /statistics/:id

Atualiza as informações das estatísticas de uma cidade cadastrada no sistema.

##### Parâmetros

Parâmetro	Tipo	Descrição
city	string	Nome da cidade
pib	float	Pib da cidade
workingForce	float	Força de trabalho empregada
educationLevel	float	Nível de educação
poverty	float	Nível de pobreza
salary	float	Faixa salarial
povertyPercentage	float	Porcentagem de pobreza
reristrationDate	LocalDate	Data de registro das estatísticas

##### Respostas

Código de status	Descrição
200	Estatísticas atualizadas com sucesso.
400	Parâmetros inválidos.

404	Estatísticas não encontradas.
-----	-------------------------------

#### Exemplo de solicitação

bash

```
PUT "/statistics/{id}" HTTP/1.1
```

```
Content-Type: application/json
```

```
{
  "id" : 123,
  "city" : "Tanabi",
  "pib" : 402315.00
  "workingForce" : 80.00,
  "educationLevel" : 95.00,
  "poverty" : 2365.00,
  "salary": 2000.00,
  "povertyPercentage" : 15.5
}
```

#### 5. DELETE /statistic/:id

Remove as estatísticas de uma cidade cadastrada no sistema.

##### Parâmetros

Parâmetro	Tipo	Descrição
id	number	ID da estatística.

##### Respostas

Código de status	Descrição
204	Estatística removida com sucesso.
404	Estatística não encontrada.

#### Exemplo de solicitação

bash

```
DELETE "/statistic/{id}" HTTP/1.1
```

#### 6. GET /statistic/:city

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através de seu nome.

#### Parâmetros

Parâmetro	Tipo	Descrição
city	string	nome da cidade.

#### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

#### Exemplo de solicitação

bash

```
GET "/city/{city}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id" : 123,
  "city" : "Tanabi",
  "pib" : 402315.00
  "workingForce" : 80.00,
  "educationLevel" : 95.00,
  "poverty" : 2365.00,
  "salary": 2000.00,
  "povertyPercentage" : 15.5
}
```

### 7. GET /statistic/:workingforce

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através de sua força de trabalho empregada, utilizando paginação.

#### Parâmetros

Parâmetro	Tipo	Descrição
workingForce	float	força de trabalho empregada

### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

### Exemplo de solicitação

bash

```
GET "/workingforce/{workingforce}" HTTP/1.1
```

### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

## 8. GET /statistic/:poverty

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através de seu nível de pobreza, utilizando paginação.

### Parâmetros

Parâmetro	Tipo	Descrição
poverty	float	nível de pobreza

### Respostas

Código de status	Descrição
------------------	-----------

200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

#### Exemplo de solicitação

bash

```
GET "/poverty/{poverty}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

### 9. GET /statistic/:salary

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através de sua média salarial, utilizando paginação.

#### Parâmetros

Parâmetro	Tipo	Descrição
salary	float	média salarial

#### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

#### Exemplo de solicitação



bash

```
GET "/poverty/{poverty}" HTTP/1.1
```

#### Exemplo de resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

#### 10. GET /statistic/:povertyPercentage

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através de sua porcentagem de pessoas na pobreza, utilizando paginação.

##### Parâmetros

Parâmetro	Tipo	Descrição
povertyPercentage	float	porcentagem de pessoas na pobreza

##### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

#### Exemplo de solicitação

bash

```
GET "/povertypercentage/{povertypercentage}" HTTP/1.1
```

### Exemplo de resposta

HTTP/1.1 200 OK

Content-Type: application/json

```
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

## 11. GET /statistic/:biggerthan

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através dos dados de tabela e valor, trazendo as estatísticas que tenham os dados maiores que os valores passados, utilizando paginação.

### Parâmetros

Parâmetro	Tipo	Descrição
tableEntity	string	tabela do banco de dados
value	float	valor a ser pesquisado
page(opcional)	int	página inicial
size(opcional)	int	quantidade de dados na página

### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

### Exemplo de solicitação

bash

```
GET "{biggerthan/{tableEntity}/{value}" HTTP/1.1
```

### Exemplo de resposta

```
HTTP/1.1 200 OK
Content-Type: application/json
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

## 12. GET /statistic/:lesserthan

Recupera as informações de estatísticas de uma cidade cadastrada no sistema através dos dados de tabela e valor, trazendo as estatísticas que tenham os dados menores que os valores passados, utilizando paginação.

### Parâmetros

Parâmetro	Tipo	Descrição
tableEntity	string	tabela do banco de dados
value	float	valor a ser pesquisado
page(opcional)	int	página inicial
size(opcional)	int	quantidade de dados na página

### Respostas

Código de status	Descrição
200	Estatísticas retornadas com sucesso.
404	Estatísticas não encontradas.

### Exemplo de solicitação

bash

GET “/{lesserthan/{tableEntity}/{value}}” HTTP/1.1

#### Exemplo de resposta

HTTP/1.1 200 OK

Content-Type: application/json

```
[
  {
    "id" : 123,
    "city" : "Tanabi",
    "pib" : 402315.00
    "workingForce" : 80.00,
    "educationLevel" : 95.00,
    "poverty" : 2365.00,
    "salary": 2000.00,
    "povertyPercentage" : 15.5
  }
]
```

## Endpoints LoginController

### 1. POST /login:register

Cadastra um novo usuário no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
name	string	Nome do usuário.
cpf	long	Cpf do usuário
email	string	E-mail do usuário
login	string	Login do usuário
password	string	Senha do usuário
role	string	Papel do usuário

#### Respostas

Código de status	Descrição
201	Usuário cadastrado com

	sucesso.
400	Parâmetros inválidos.

#### Exemplo de solicitação

bash

POST "/register" HTTP/1.1

Content-Type: application/json

```
{
  "name" : "João Silva",
  "cpf" : "12345678910",
  "email" : "joao.silva@gmail.com",
  "login" : "JSilva",
  "password" : "123456",
  "role": "USER"
}
```

### 1. POST /login:authenticate

Valida um usuário no sistema.

#### Parâmetros

Parâmetro	Tipo	Descrição
email	string	E-mail do usuário
password	string	Senha do usuário

#### Respostas

Código de status	Descrição
200	OK
400	Parâmetros inválidos.
401	Não autorizado

#### Exemplo de solicitação

bash

POST "/authenticate" HTTP/1.1

Content-Type: application/json

```
{  
  "email" : "joao.silva@gmail.com",  
  "password" : "123456"  
}
```

#### **Exemplo de resposta**

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "token" :  
    eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c  
}
```