

---

# **br\_registrations**

*Release 0.1.0*

**Siatecinos**

**23 jul. 2022**



|          |                                 |          |
|----------|---------------------------------|----------|
| <b>1</b> | <b>Conteúdo</b>                 | <b>3</b> |
| 1.1      | Utilização . . . . .            | 3        |
| 1.2      | Métodos disponíveis . . . . .   | 3        |
| 1.3      | API . . . . .                   | 5        |
|          | <b>Índice de Módulos Python</b> | <b>7</b> |
|          | <b>Índice</b>                   | <b>9</b> |



**BR registrations** (/br-registrations/) é uma biblioteca Python para validação de números de CNPJ/CPF utilizados no Brasil. Seu código-fonte está disponível no [GitHub - br\\_registrations](#) e pode ser instalada facilmente através do repositório *PyPi.org*.

Visite a seção dessa documentação *Instalação* para as informações de instalação, e sobre a forma de utilização em *Métodos disponíveis*.

---

**Nota:** Este projeto ainda está sob desenvolvimento.

---



## 1.1 Utilização

### 1.1.1 Instalação

Para usar, primeiro efetue a instalação com o pip:

```
(.venv) $ pip install br-registrations
```

### 1.1.2 Obter a versão atual

Você pode usar `br_registrations.__version__()` para obter a versão atual:

```
>>> import br_registrations as brr
```

```
>>> brr.__version__  
'0.0.1.0'
```

## 1.2 Métodos disponíveis

### 1.2.1 Obter os dígitos da inscrição

Você pode usar os métodos `validateCNPJ.CNPJ.get_cnpj_dv` ou `validateCPF.CPF.get_cpf_dv`, quando precisar obter os últimos dois dígitos da inscrição pesquisada. Estes métodos possuem o parâmetro opcional *Parâmetro regex*.

Por exemplo, para o CNPJ o método será:

`br_registrations.validateCNPJ.CNPJ.get_cnpj_dv(inscricao: str, regex: bool = False) → list`

Retorna uma lista dos dois últimos números de uma inscrição CNPJ.

Exemplos:

```
:return cnpj_dv: ('112223330001') -> [8, 1]
```

```
:return cnpj_dv: ('46.293.332/0001') -> [0, 2]
```

E para o CPF será:

`br_registrations.validateCPF.CPF.get_cpf_dv(inscricao: str, regex: bool = False) → list`

Retorna uma lista dos dois últimos números de uma inscrição CPF.

Exemplos:

```
:return cpf_dv: ('111444777') -> [3, 5]
```

```
:return cpf_dv: ('912.441.670') -> [3, 7]
```

### 1.2.2 Validar a inscrição

Você pode realizar a validação de um número de inscrição com os métodos `validateCNPJ.CNPJ.valid_cnpj` ou `validateCPF.CPF.valid_cpf` disponíveis. Estes métodos possuem o parâmetro opcional *Parâmetro regex*.

Por exemplo, para o CNPJ o método será:

`br_registrations.validateCNPJ.CNPJ.valid_cnpj(cnpj: str, regex: bool = False) → bool`

Retorna True (válido) ou False (inválido) para um documento CNPJ.

Exemplos:

```
:return valid_cnpj: ('11222333000181') -> True
```

```
:return valid_cnpj: ('46.293.332/0001-02') -> True
```

E para o CPF será:

`br_registrations.validateCPF.CPF.valid_cpf(cpf: str, regex: bool = False) → bool`

Retorna True (válido) ou False (inválido) para um documento CPF.

Exemplos:

```
:return valid_cpf: ('11144477735') -> True
```

```
:return valid_cpf: ('912.441.670-37') -> True
```

### 1.2.3 Parâmetro regex

Em todos os métodos há a opção de parâmetro `regex`, com valor *default*: **False**, que permite uma pesquisa que contemple pontuações no meio da inscrição. Caso a pesquisa seja realizada com esse argumento diferente de `True` será retornada uma *Exception* pelo interpretador.

Exemplo:

```
>>> from br_registrations import validateCPF
>>> response_2=cpf.valid_cpf('912.441.670-37', regex=True)
>>> print('CPF "912.441.670-37" is valid is', response_2, '.')
'CPF "912.441.670-37" is valid is True .'
```

## 1.3 API

---

*br\_registrations*

---

..

---

### 1.3.1 br\_registrations

” BR registrations - Biblioteca Python usada para validação de números de CNPJ/CPF usados no Brasil.



**b**

`br_registrations`, 5



## B

br\_registrations  
    módulo, 5

## G

get\_cnpj\_dv()                   (no            módulo  
    br\_registrations.validateCNPJ.CNPJ), 3  
get\_cpf\_dv()                   (no            módulo  
    br\_registrations.validateCPF.CPF), 4

## M

módulo  
    br\_registrations, 5

## V

valid\_cnpj()                   (no            módulo  
    br\_registrations.validateCNPJ.CNPJ), 4  
valid\_cpf()                   (no            módulo  
    br\_registrations.validateCPF.CPF), 4