

---

# PySIGEP Documentation

*Versão 0.1.0*

**Michell Stuttgart**

**30 mar, 2019**



---

## Conteúdo

---

<b>1</b>	<b>Features</b>	<b>3</b>
1.1	Instalação . . . . .	3
1.2	Utilização . . . . .	4
1.3	Contribuindo . . . . .	7
1.4	Créditos . . . . .	9
1.5	Historico . . . . .	9



Interface python para uso dos serviços fornecidos pelo SIGEPWeb dos Correios.

O SIGEP WEB é um sistema com o propósito de preparar e gerenciar as postagens de Clientes dos Correios. Seus principais atributos técnicos são: facilidade e rapidez na preparação das postagens e gestão das informações sobre os objetos postados.



- Verificar *status* de um Cartão de Postagem
- Obter dados do endereço a partir de seu respectivo CEP.
- Verificar disponibilidade de um dado serviço.
- Gerar etiquetas para postagem de mercadoria.

## 1.1 Instalação

Atualmente, a PySIGEP possui suporte para Python 3.5+.

### 1.1.1 Versão estável

Versão atual **ainda esta em fase de desenvolvimento**, sendo que os recursos disponíveis podem ser removidos sem aviso prévio. Portanto, não é recomendável seu uso em ambiente de produção.

Para instalar PySIGEP, execute este comando no seu terminal:

```
$ pip install pysigep
```

Este é o método recomendado para instalar a PySIGEP. Desse modo sempre será instalado a versão mais recente.

### 1.1.2 A partir do código fonte

O código fonte do PySIGEP pode ser baixado [aqui](#).

Você pode realizar o *clone* do repositório público:

```
$ git clone https://github.com/mstuttgart/pysigep.git
```

Ou baixar o [tarball](#):

```
$ curl -OL https://github.com/mstuttgart/pysigep/tarball/master
```

Uma vez que você tenha uma cópia do código fonte, você pode instalá-lo com:

```
$ python setup.py install
```

## 1.2 Utilização

A versão atual **ainda esta em fase de desenvolvimento**, sendo que os recursos disponíveis podem ser removidos sem aviso prévio. Portanto, não é recomendável seu uso em ambiente de produção.

PySigep no momento possui suporte para os seguintes serviços providos pelo SigepWeb:

- consultaCEP
- verificaDisponibilidadeServico
- getStatusCartaoPostagem
- solicitaEtiquetas
- geraDigitoVerificadorEtiquetas

Novos serviços serão implementados futuramente. A seguir temos alguns exemplos de utilização da *pysigep*. Para mais detalhes sobre os serviços fornecidos, por favor, consulte o Manual do SigepWeb.

### 1.2.1 SOAPClient

A grande maioria dos serviços do SigepWeb, exigem um cadastro de *usuário* e *senha*. De modo a tornar mais prático a consulta, foi desenvolvido uma classe para armazenar estes dados durante o uso da biblioteca. Sendo assim, antes de qualquer consulta devemos criar um objeto *SOAPClient*.

```
from pysigep.utils import HOMOG_USUARIO, HOMOG_SENHA, HOMOLOGACAO

# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

# Realizamos a consulta di servico
endereco = cliente.nomeservico(..)
```

As constantes *HOMOG\_USUARIO*, *HOMOG\_SENHA*, *HOMOLOGACAO* são constantes fornecidas para auxiliar o desenvolvedor durante o processo de desenvolvimento.

No momento, a *pysigep* possui as seguintes constantes:

- *HOMOG\_USUARIO*: usuario para consultas em ambiente de homologação,
- *HOMOG\_SENHA*: senha para o usuário de homologação,
- *HOMOLOGACAO*: constante a ser fornecida na criação do *SOAPClient*, permitindo utilizar o ambiente de homologação,
- *PRODUCAO*: constante a ser fornecida na criação do *SOAPClient*, permitindo utilizar o ambiente de homologação,
- *HOMOG\_CODIGO\_ADMIN*: código administrativo de demonstração, fornecido pelos correios,

- HOMOG\_CARTAO: código do cartão de postagem de demonstração, fornecido pelos correios,
- HOMOG\_CNPJ: CNPJ de demonstração, fornecido pelos correios,
- URLS: *dict* contendo as urls dos ambientes de homologação e produção.

Estas constantes podem ser acessadas através do pacote *utils*:

```
from pysigep.utils import (URLS,
                           HOMOLOGACAO,
                           PRODUCAO,
                           HOMOG_USUARIO,
                           HOMOG_SENHA,
                           HOMOG_CODIGO_ADMIN,
                           HOMOG_CARTAO,
                           HOMOG_CNPJ)
```

## 1.2.2 consultaCEP

Este método retorna o endereço correspondente ao número de CEP informado.

```
# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

# Realizamos a consulta de CEP
endereco = cliente.consulta_cep('37.503-130')

print(endereco.bairro)
print(endereco.cep)
print(endereco.cidade)
print(endereco.complemento)
print(endereco.complemento2)
print(endereco.end)
print(endereco.id)
print(endereco.uf)
print(endereco.unidadesPostagem)
```

## 1.2.3 verificaDisponibilidadeServico

Verifica se um serviço que não possui abrangência nacional está disponível entre um CEP de Origem e de Destino.

```
params = {
    'cod_administrativo': HOMOG_CODIGO_ADMIN,
    'numero_servico': '04162',
    'cep_origem': '70002900',
    'cep_destino': '70.002-900',
}

# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

# Realizamos a verificacao de disponibilidade
```

```
disponibilidade = cliente.verifica_disponibilidade_servico(**params)

print(disponibilidade)
# Saída: True ou False
```

## 1.2.4 getStatusCartaoPostagem

Este método retorna o situação do cartão de postagem, ou seja, se o mesmo está “Normal” ou “Cancelado”. É recomendada a pesquisa periódica para evitar tentativa de postagens com cartão suspenso, ocasionando a não aceitação dos objetos nos Correios.

```
# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

params = {
    'numero_cartao_postagem': HOMOG_CARTAO,
}

# Realizamos a consulta do status do cartao de postagem
status = cliente.get_status_cartao_postagem(**params)

print(status)
# Saída: 'Normal' ou 'Cancelado'
```

## 1.2.5 solicitaEtiquetas

Retorna uma dada quantidade de etiquetas sem o dígito verificador.

```
params = {
    'tipo_destinatario': 'C',
    'cnpj': HOMOG_CNPJ,
    'id_servico': 124849,
    'qtd_etiquetas': 2,
}

# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

params = {
    'numero_cartao_postagem': HOMOG_CARTAO,
}

# Realizamos a solicitacao de etiquetas
lista_etiquetas = cliente.solicita_etiquetas(**params)

print(lista_etiquetas)
# Saída: lista_etiquetas = [
#         'DL76023727 BR',
#         'DL76023728 BR',
#     ]
```

## 1.2.6 geraDigitoVerificadorEtiquetas

Este método retorna o DV - Dígito Verificador de um lista de etiquetas.

```
params = {
    'etiquetas': ['DL76023727 BR', 'DL76023728 BR'],
}

# Criamos o cliente SOAP
cliente = SOAPClient(ambiente=HOMOLOGACAO,
                    senha=HOMOG_SENHA,
                    usuario=HOMOG_USUARIO)

# Realizamos a consulta do status do cartao de postagem
lista_digitos = cliente.gera_digito_verificador_etiquetas(**params)

print(lista_digitos)
# Saída: lista_digitos = [2, 6]
```

## 1.3 Contribuindo

Contribuições são bem-vindas e muito apreciadas!

Você pode contribuir de várias maneiras:

### 1.3.1 Tipos de Contribuições

#### Reportando erros

Informe erros em <https://github.com/mstuttgart/pysigep/issues>.

Se você está reportando um *bug*, por favor inclua:

- O nome e a versão do seu sistema operacional.
- Qualquer detalhe sobre sua configuração local que possa ser útil na solução de problemas.
- Etapas detalhadas para reproduzir o bug.

#### Corrigindo erros

Busque na lista de *issues* por aquelas com a tag *Bug*. Qualquer *issue* marcada com «Bug» está aberta para quem quiser corrigi-la.

#### Adicionando novos recursos

Busque na lista de *issues* por aquelas com a tag *Improvement* ou *New feature*. Qualquer *issue* marcada com essas *tags* está aberta para quem quiser implementá-la.

## Melhorando a documentação

A documentação da PySIGEP sempre pode ser melhorada, seja como parte da documentação oficial do PySIGEP, em docstrings, ou mesmo na web em postagens de blog, artigos e etc. Então caso você tenha escrito alguma postagem sobre a PySIGEP, por favor, me avise para que a mesma seja incluída aqui como referência.

## Enviar Comentários

A melhor maneira de enviar comentários é abrir uma *issue* em <https://github.com/mstuttgart/pysigep/issues>.

Se você está propondo um novo recurso para PySIGEP, por favor siga os seguintes passos:

- Explique em detalhes como isso funcionaria.
- Mantenha o escopo o mais simples possível, para facilitar a implementação.
- Lembre-se que este é um projeto voluntário, e que contribuições são bem-vindos :)

## 1.3.2 Começando a contribuir!

Pronto para contribuir? Veja como configurar *pysigep* para desenvolvimento local.

1. Fork o repositório *pysigep* no GitHub.
2. Clone sua *branch* localmente:

```
$ git clone git@github.com:your_name_here/pysigep.git
```

3. Instale sua cópia local em um *virtualenv*. Supondo que você tenha instalado o *virtualenv*, é assim que você configura a sua *fork* para o desenvolvimento local:

```
$ cd pysigep
$ virtualenv -p python3 env
$ pip3 install -r requirements.txt
```

4. Crie uma *branch* para desenvolvimento:

```
$ git checkout -b nome-da-sua-branch
```

Agora você pode fazer suas mudanças localmente.

5. Quando terminar de fazer alterações, verifique se suas alterações passam no *flake8* e nos testes:

```
$ flake8 pysigep tests
$ python setup.py test
```

6. Confirme as suas alterações e as envie para o GitHub:

```
$ git add .
$ git commit -m "Descricao detalhada das sua alteracoes."
$ git push origin nome-da-sua-branch
```

7. Envie um *Pull Request* para o repositório oficial da PySIGEP no GitHub.

### 1.3.3 Instruções de Solicitação de Pull Request

Antes de enviar um pedido de *Pull Request*, verifique se ele atende a essas diretrizes:

1. O pedido de *Pull Request* deve incluir testes, quando for uma nova *feature*.
2. Se o *Pull Request* adicionar funcionalidades, a documentação deve ser atualizada adicionando detalhes de uso da nova funcionalidade.
3. O pedido de *Pull Request* deve funcionar para o Python 3.5+. Verificar [https://travis-ci.org/mstuttgart/pysigep/pull\\_requests](https://travis-ci.org/mstuttgart/pysigep/pull_requests) e certifique-se de que os testes passem para todas as versões do Python suportadas.

## 1.4 Créditos

### 1.4.1 Desenvolvedor

PySIGEP foi desenvolvido originalmente por:

- Michell Stuttgart (<https://github.com/mstuttgart>)

### 1.4.2 Contribuidores

Agradecimentos aos seguintes contribuidores pelo esforço de fazer a PySIGEP melhor:

- [Alessandro Martini](<https://github.com/martini97>)
- [Danimar Ribeiro](<https://github.com/danimaribeiro>)
- [Michel Sabchuk](<https://github.com/michelts>)

## 1.5 Historico

### 1.5.1 0.0.1 (2018-05-06)

- Versão inicial de desenvolvimento.

Encontrou algum erro? Tem alguma sugestão para melhorar a PySIGEP? Deixe-me saber. Contribuições são muito bem vindas!