
COVID19-ES-Py

Release 3.1.0

jun. 05, 2021

1	Índices	1
2	Recursos	3
3	Instalação	5
3.1	Pelo repositório PyPI	5
3.2	Pelo código fonte	5
4	Como usar	7
4.1	Exemplos	7
4.2	Considerações finais	8

- [Índice geral](#)
- [Índice do módulo](#)

[API em Python para consulta de casos de COVID-19 no estado do Espírito Santo.](#)

[PyPI](#) [pyversions](#) [PyPI GitHub](#) [code size in bytes](#) [Travis \(.org\)](#) [Read the Docs](#) [Codecov](#) [Codacy](#) [grade](#)

[PyPI - Downloads](#) [License: GPL v3](#)

- Extraí dados sobre:
 - Casos confirmados no estado
 - Critérios de avaliação de casos
 - Status e evolução de casos
 - Faixa etária, sexo, etnia e grau de escolaridade de pacientes
 - Sintomas e comorbidades de pacientes
 - Viagens de pacientes
- Filtro de casos por data
- Estruturação e busca de casos por município

Atualmente, o COVID19-ES-Py possui suporte para Python 3.6+. Uma mesma release é feita ao PyPI e ao GitHub ao mesmo tempo, portanto você pode usar a fonte que achar mais conveniente.

3.1 Pelo repositório PyPI

Utilize seu gerenciador de pacotes preferido para instalar o pacote. Com `pip`:

```
pip install COVID19-ES-Py
```

3.2 Pelo código fonte

3.2.1 Através do `pip`

É possível usar o `pip`, dentre outros gerenciadores de pacotes, para instalar o pacote através do repositório. Usando a release mais recente:

```
pip install https://github.com/AtilioA/COVID19-ES-Py/releases/latest
```

3.2.2 Buildando com `setup.py`

1. Baixe a última release e extraia a pasta;
2. Entre no diretório raiz do pacote pelo terminal e rode o comando `python setup.py install`

A API possui duas classes principais: `LeitorRelatorio` e `Relatorio`. A primeira é capaz de filtrar relatórios por data, etc, usando objetos `Relatorio` para fazer a interface para o programador. A segunda pode ser utilizada para extrair informações de um relatório específico. Confira os exemplos a seguir:

4.1 Exemplos

4.1.1 Inicializando o leitor e obtendo dados do último relatório:

```
import COVID19_ES_Py

# Inicializando o leitor
leitor = COVID19_ES_Py.LeitorRelatorio()

# Carregando objeto Relatorio com dados do último arquivo csv emitido
relatorio = leitor.carrega_ultimo_relatorio() # Relatório do dia 17/04/2020
relatorio.casos
>>> {'AFONSO CLAUDIO': {'casosConfirmados': 2, 'obitos': 0},
...
     'VITORIA': {'casosConfirmados': 209, 'obitos': 8}}
```

Total de casos do relatório:

```
relatorio.totalGeral
>>> {'casosConfirmados': 952, 'obitos': 28}
```

4.1.2 Pesquisando casos por município:

Retorna objeto `Municipio` com casos do município no relatório:

```
relatorio.busca_casos_municipio("Vitória")
>>> "Município VITORIA:
209 casos confirmados.
8 óbitos."

# A busca ignora espaços extras e capitalização
relatorio.busca_casos_municipio("  santa teresa ")
>>> "Município SANTA TERESA:
4 casos confirmados.
0 óbitos."

# Também ignora caracteres especiais
relatorio.busca_casos_municipio("AFONSO CLAUDIO")
>>> "Município AFONSO CLAUDIO:
2 casos confirmados.
0 óbitos."

relatorio.busca_casos_municipio("arapiraca")
>>> exceptions.RelatorioError: O município "arapiraca" não foi encontrado no_
↳ relatório. Pode ter ocorrido um erro de digitação ou o município não é do Espírito_
↳ Santo.
```

Atributos da classe Município:

- nome (str): O nome do município.
- casos (list): (Caso): Uma lista de objetos do tipo Caso.
- casosConfirmados (int): O número de casos confirmados no município.
- obitos (int): O número de óbitos confirmados em decorrência de COVID-19 no município.

4.1.3 Pesquisando relatório por data:

Retorna objetos Relatorio:

Filtrando até um dado dia:

```
relatorio16_04 = leitor.filtro_casos_ate_dia("16/04/2020")
>>> "Relatório do arquivo https://bi.static.es.gov.br/covid19/MICRODADOS.csv:
Total geral: {'casosConfirmados': 935, 'obitos': 28}
35 municípios infectados."
```

Filtrando apenas um dado dia:

```
relatorio16_04 = leitor.filtro_casos_no_dia("16/04/2020")
>>> "Relatório do arquivo https://bi.static.es.gov.br/covid19/MICRODADOS.csv:
Total geral: {'casosConfirmados': 11, 'obitos': 0}
5 municípios infectados."
```

A data de entrada pode ser formatada de várias formas.

4.2 Considerações finais

Se você está no Windows, a versão atual da biblioteca rows (0.4.1) possui um bug que impossibilita seu uso. Neste caso, instale a versão mais recente com:

```
pip install https://github.com/turicas/rows/archive/develop.zip#egg=rows
```

Os dados são disponibilizados pelo Governo do Estado do Espírito Santo com a Superintendência Estadual de Comunicação Social do Espírito Santo (SECOM) e [podem ser encontrados aqui](#).

Devido à natureza dos dados, há uma chance considerável de o scraping parar de funcionar a qualquer momento. Para minimizar essa possibilidade, muitos testes automatizados foram escritos; em caso de discrepância, tentarei atualizar o pacote o mais rápido possível.

Encontrou algum erro? Tem alguma sugestão para melhorar o COVID19-ES-Py? [Crie uma issue!](#) Contribuições são muito bem-vindas.