
Django GIS Brasil Documentation

Versão 0.2

CodeFi.sh Team (Christian S. Perone, Leandro Nunes, Gabriel Wa

18/06/2013

Sumário

Release v0.2. (*Instalação*)

django-gis-brasil é uma iniciativa open-source da CodeFi.sh (*Christian S. Perone, Leandro Nunes e Gabriel Wainer*) para a criação de uma aplicação Django com informações geográficas do território brasileiro. O projeto contém atualmente informações de todos municípios brasileiros fornecidos pelo IBGE bem como polígonos geográficos prontos para serem importados no seu banco geográfico através do [GeoDjango](#).

A aplicação django-gis-brasil realiza a importação de forma automática das informações geográficas para o seu banco geográfico. A aplicação também conta com a parte administrativa já modelada para todos os dados disponíveis (ver [Screenshots](#)) abaixo.

Screenshots

1.1 Listagem de Municípios do Brasil

Todos modelos de dados incluem administração do Django implementada.

Administração do Django Bem-vindo(a), admin. Alterar senha / Encerrar sessão

nício > Gisbrasil > Municípios

Selecione Municipio para modificar

Adicionar Municipio

Q Pesquisar 496 resultados (5566 total)

Ação: [-----] Ir 0 de 100 selecionados

<input type="checkbox"/> Nome do Municipio	Sigla	Região	Nome da Mesorregião	Nome da Microrregião
<input type="checkbox"/> Aceguá	RS	Sul	Sudoeste Rio-grandense	Campanha Meridional
<input type="checkbox"/> Água Santa	RS	Sul	Noroeste Rio-grandense	Passo Fundo
<input type="checkbox"/> Agudo	RS	Sul	Centro Ocidental Rio-grandense	Restinga Seca
<input type="checkbox"/> Ajuricaba	RS	Sul	Noroeste Rio-grandense	Ijuí
<input type="checkbox"/> Alecrim	RS	Sul	Noroeste Rio-grandense	Santa Rosa
<input type="checkbox"/> Alegrete	RS	Sul	Sudoeste Rio-grandense	Campanha Ocidental
<input type="checkbox"/> Alegria	RS	Sul	Noroeste Rio-grandense	Ijuí
<input type="checkbox"/> Almirante Tamandaré do Sul	RS	Sul	Noroeste Rio-grandense	Carazinho
<input type="checkbox"/> Alpestre	RS	Sul	Noroeste Rio-grandense	Frederico Westphalen
<input type="checkbox"/> Alto Alegre	RS	Sul	Noroeste Rio-grandense	Cruz Alta
<input type="checkbox"/> Alto Feliz	RS	Sul	Metropolitana de Porto Alegre	Montenegro
<input type="checkbox"/> Alvorada	RS	Sul	Metropolitana de Porto Alegre	Porto Alegre
<input type="checkbox"/> Amaral Ferrador	RS	Sul	Sudeste Rio-grandense	Serras de Sudeste
<input type="checkbox"/> Ametista do Sul	RS	Sul	Noroeste Rio-grandense	Frederico Westphalen
<input type="checkbox"/> André da Rocha	RS	Sul	Nordeste Rio-grandense	Guaporé
<input type="checkbox"/> Anta Gorda	RS	Sul	Nordeste Rio-grandense	Guaporé
<input type="checkbox"/> Antônio Prado	RS	Sul	Nordeste Rio-grandense	Caxias do Sul
<input type="checkbox"/> Arambaré	RS	Sul	Metropolitana de Porto Alegre	Camaquã
<input type="checkbox"/> Araricá	RS	Sul	Metropolitana de Porto Alegre	Porto Alegre
<input type="checkbox"/> Aratiba	RS	Sul	Noroeste Rio-grandense	Erechim
<input type="checkbox"/> Arrolo do Melo	RS	Sul	Centro Oriental Rio-grandense	Lajeado-Estrela

Filtro
Por sigla
Todos
AC
AL
AM
AP
BA
CE
DF
ES
GO
MA
MG
MS
MT
PA
PB
PE
PI
PR
RJ
RN
RO
RR
RS
SC
SE
SP
TO
Por Região
Todos
Centro-Oeste
Nordeste
Norte
Sudeste
Sul

1.2 Pesquisa de Municípios

Todos modelos de dados incluem pesquisa nos campos.

Administração do DjangoBem-vindo(a), a

Início > Gisbrasil > Municípios

Selecione Município para modificar

1 resultado (5566 total)

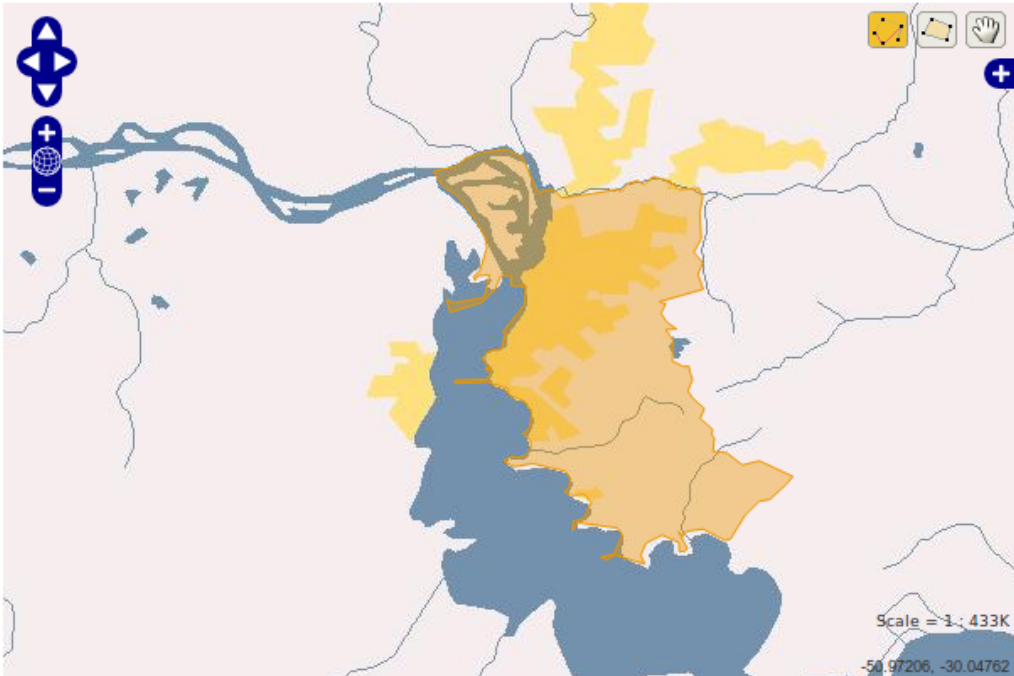
Ação: 0 de 1 selecionados

<input type="checkbox"/>	Nome do Município	Sigla	Região	Nome da Mesorregião	Nome da Microrregião
<input type="checkbox"/>	Porto Alegre	RS	Sul	Metropolitana de Porto Alegre	Porto Alegre

1 Município

1.3 Dados Geográficos

Todos modelos de dados incluem polígonos geográficos.

Sigla:	<input type="text" value="RS"/>
Nome do Município:	<input type="text" value="Porto Alegre"/>
Região:	<input type="text" value="Sul"/>
Mesorregião:	<input type="text" value="4305"/>
Nome da Mesorregião:	<input type="text" value="Metropolitana de Porto Alegre"/>
Microrregião:	<input type="text" value="43026"/>
Nome da Microrregião:	<input type="text" value="Porto Alegre"/>
Multipoly:	<div></div>

1.4 Informação Bairros

Informações geográficas (incluindo polígonos geográficos de bairros).

(Por hora, disponível apenas em Porto Alegre/RS):

Administração do DjangoBem-vindo(a), **admin**. [Alterar senha](#) / [Encerrar sessão](#)

[Início](#) > [Gisbrasil](#) > [Porto Alegre - Bairro](#) > [Auxiliadora](#)

Modificar Porto Alegre - Bairro


Histórico

Código do Bairro:

Oficial:

Nome do Bairro:

Poly:



Scale = 1 : 108K
-51.28765, -29.98472

Delete all Features

[Anular](#)

[Salvar e adicionar outro\(a\)](#)

[Salvar e continuar editando](#)

[Salvar](#)

Pesquisas Geográficas

Utilizando o GeoDjango você pode fazer pesquisas geográficas, serializar dados automaticamente para o formato GeoJSON e outras facilidades mostradas nos exemplos abaixo.

2.1 Procurando por cidade

Neste exemplo, a cidade de “Porto Alegre” é encontrada e serializamos o polígono geográfico referente ao município em GeoJSON:

```
>>> from gisbrasil.models import *
>>> cidade = Municipio.objects.get(nome_municipio__iexact='Porto Alegre')
>>> cidade.multipoly.json
u'{ "type": "MultiPolygon", "coordinates":
  [ [ [ [ -51.233122078168442, -29.937217337009614 ],
    [ -51.220177248016007, -29.96659503107206 ],
    (... omitido por brevidade ...)
```

2.2 Procurando por Latitude/Longitude

Neste exemplo, temos a coordenada Latitude -51.22 e a Longitude -30.03 e queremos encontrar detalhes da cidade que contém esta coordenada (tudo isto usando apenas uma pesquisa geográfica no banco de dados, sem acesso ao Google Maps ou qualquer outro serviço):

```
>>> from gisbrasil.models import *
>>> from django.contrib.gis.geos import Point
>>> coordenada = Point(-51.22, -30.03)
>>> qs = Municipio.objects.filter(multipoly__contains=coordenada)
>>> qs
[<Municipio: Porto Alegre>]
```

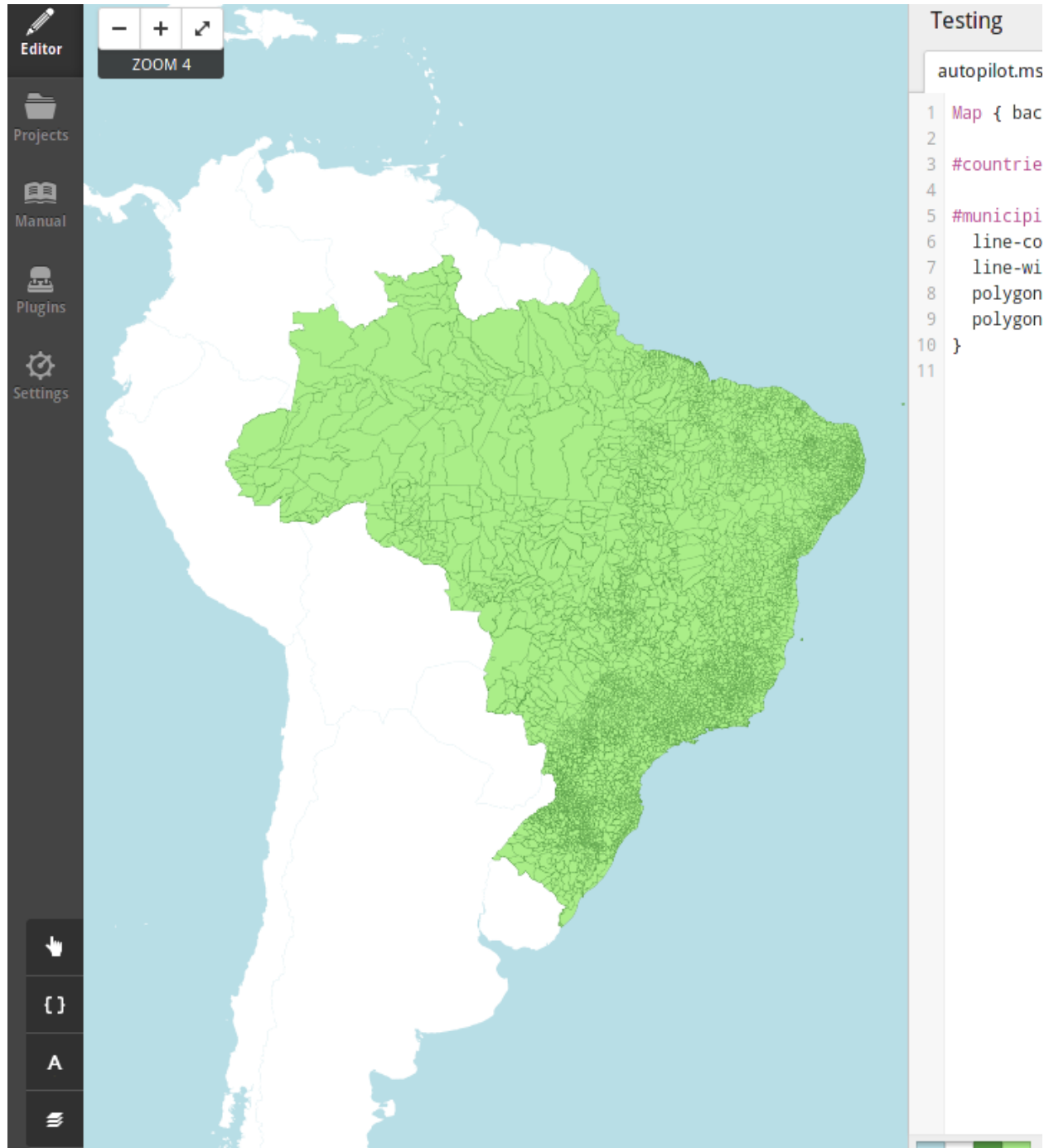
Como podemos notar, a cidade de “Porto Alegre” contém a coordenada especificada.

Para mais informações sobre como usar as pesquisas geográficas, veja a documentação do [GeoDjango](#).

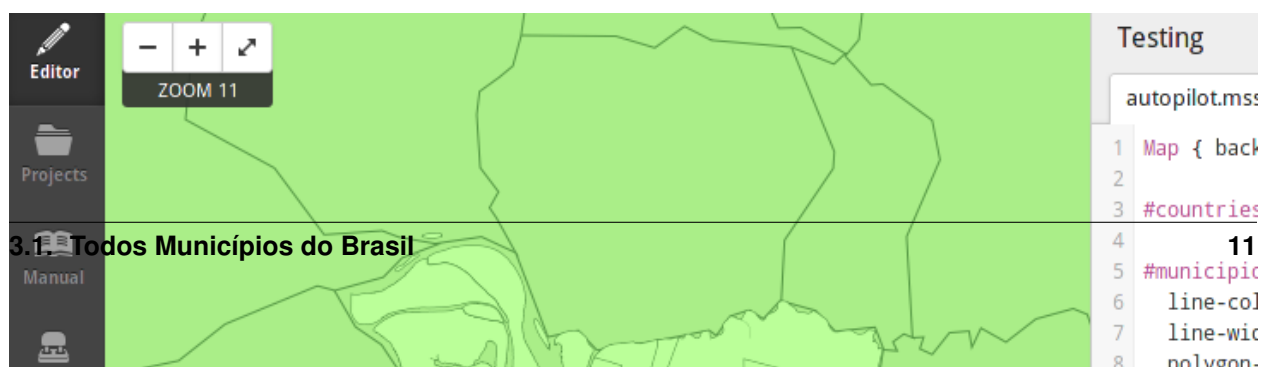
Criação de mapas com TileMill

Você pode utilizar o [TileMill](#) para conectar no seu banco geográfico e gerar mapas com estilos customizados utilizando os dados do Django GIS Brasil. Veja exemplos abaixo.

3.1 Todos Municípios do Brasil



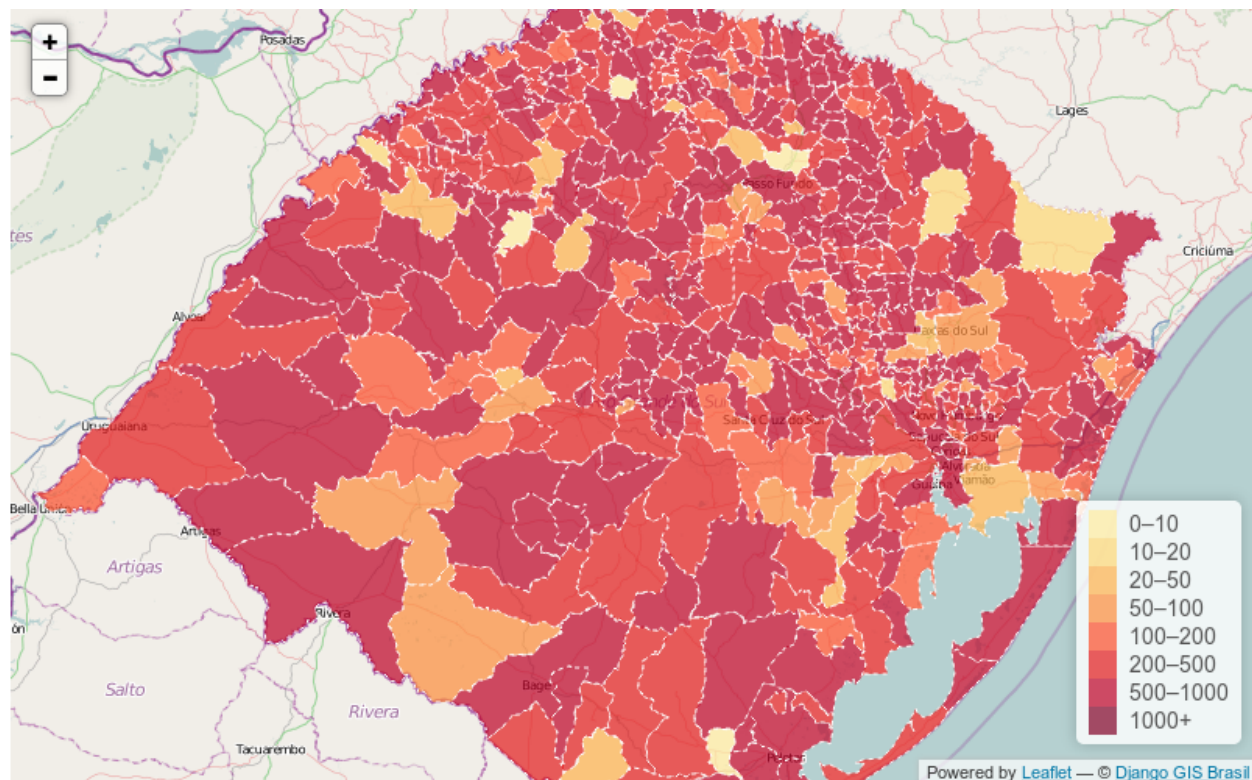
3.2 Bairros de Porto Alegre



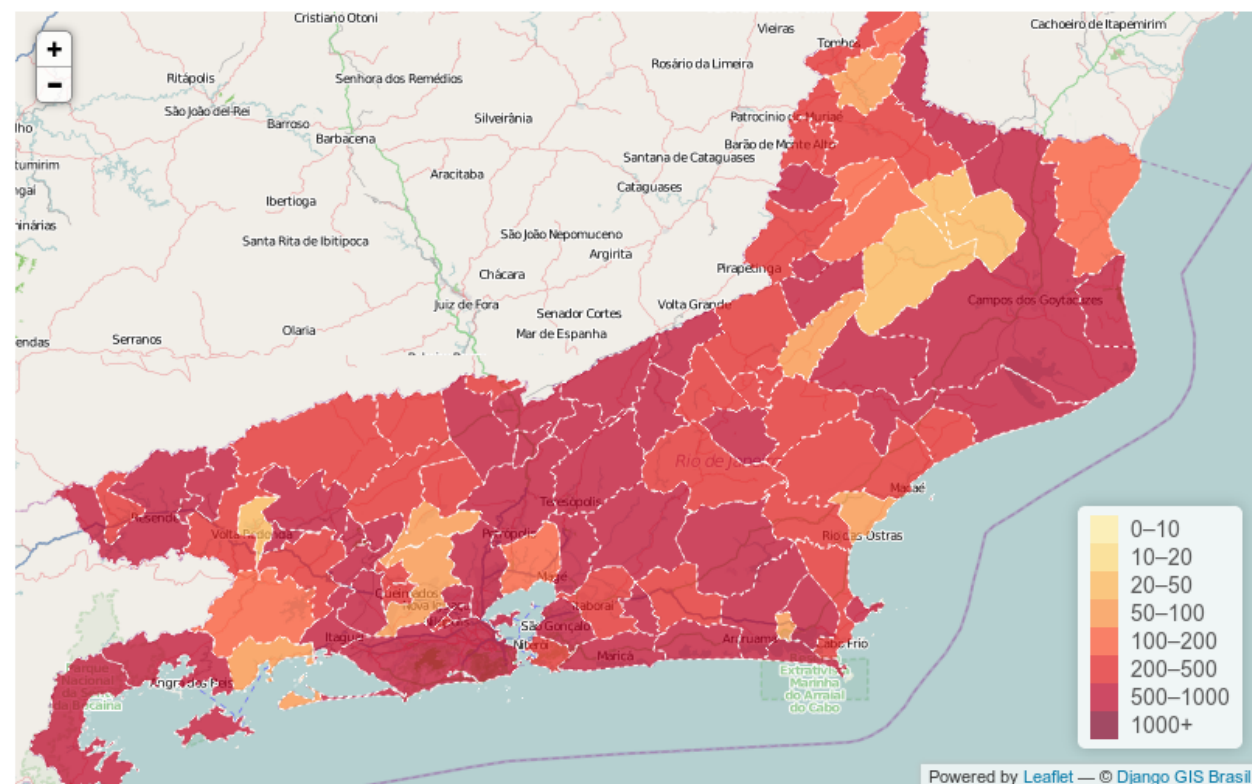
Mapas coropléticos

Você também pode utilizar o GeoJSON gerado utilizando os dados do Django GIS Brasil para criar **mapas coropléticos** interativos utilizando o [Leaflet](#) como nos exemplos abaixo.

4.1 Mapa coroplético do Rio Grande do Sul



4.2 Mapa coroplético do Rio de Janeiro



Instalação

Siga as instruções abaixo para instalar esta aplicação no seu projeto Django.

1. Instale o `django-gis-brasil` através do *pip* ou *easy_install*:

```
pip install django-gis-brasil
```

2. Adicione a aplicação “gisbrasil” na propriedade `INSTALLED_APPS` do seu projeto Django:

```
INSTALLED_APPS = (  
    ...  
    'gisbrasil',  
)
```

3. Execute:

```
python manage.py syncdb
```

para criar os modelos no seu banco de dados.

4. Execute:

```
python manage.py gisloader
```

para carregar os dados para o seu banco de dados.

Changelog

6.1 Release v.0.2 (*bug fix*)

- Corrigidos alguns typos na documentação;
- Corrigido empacotamento da aplicação;

6.2 Release v.0.1

- Lançamento do projeto open-source;

Licença

Mais informações:

Copyright (c) 2013, CodeFi.sh
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by CodeFi.sh.
4. Neither the name of the CodeFi.sh nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* This software contains data public available from other sources, see the README files on each data directory for more information.

THIS SOFTWARE IS PROVIDED BY CODEFI.SH ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CODEFI.SH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7.1 Colaboradores

Christian S. Perone (CodeFi.sh) [[twitter](#)] [[blog](#)] [[github](#)].

Gabriel Wainer (CodeFi.sh) [[twitter](#)] [[github](#)].

Leandro Nunes (CodeFi.sh) [[twitter](#)] [[blog](#)] [[github](#)].

Página de pesquisa

- *search*