# Gráficos: Django + Highcharts

Régis da Silva Santos
http://rg3915.github.io

github.com/grupy-sp/encontros

04 de Junho de 2016

# Django

Se você não conhece Django sugiro que leia este tutorial.

http://pythonclub.com.br/tutorial-django-17.html

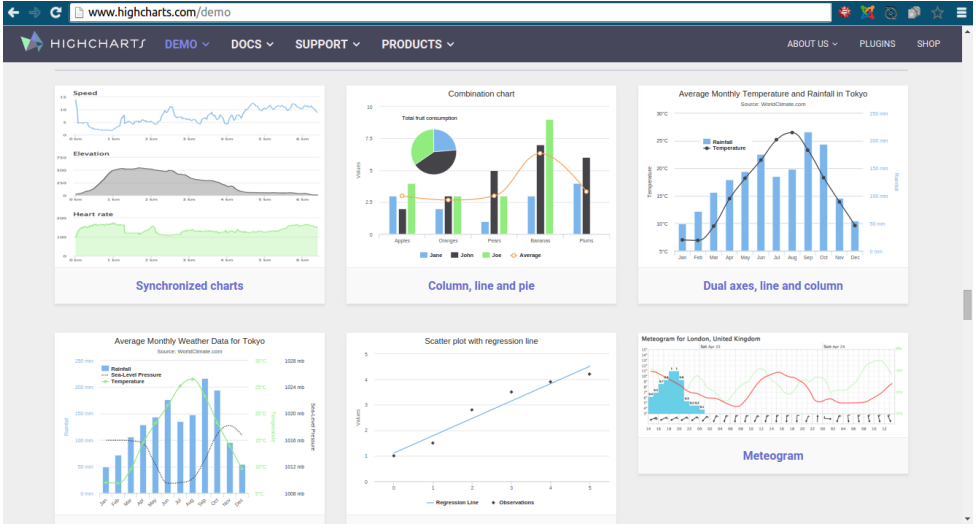
Além da documentação oficial.

https://docs.djangoproject.com/en/1.9/intro/tutorial01/

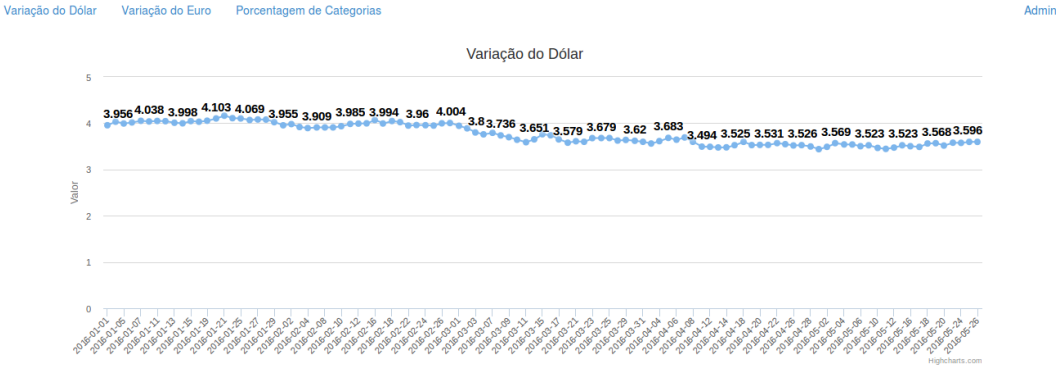
Django==1.9.6

# Highcharts

http://www.highcharts.com/



# Objetivo


Variação do Dólar
Variação do Euro
Porcentagem de Categorias
Admin
Variação do Dólar
Valor
5
4
3
2
1
0
3.956 4.038 3.998 4.103 4.069 3.955 3.909 3.985 3.994 3.96 4.004 3.8 3.736 3.651 3.579 3.679 3.62 3.683 3.494 3.525 3.531 3.526 3.569 3.523 3.523 3.568 3.596
2016-01-01
2016-01-05
2016-01-07
2016-01-11
2016-01-13
2016-01-15
2016-01-19
2016-01-21
2016-01-25
2016-01-27
2016-01-29
2016-02-02
2016-02-04
2016-02-08
2016-02-10
2016-02-12
2016-02-16
2016-02-18
2016-02-22
2016-02-24
2016-02-26
2016-03-01
2016-03-03
2016-03-07
2016-03-09
2016-03-11
2016-03-15
2016-03-17
2016-03-21
2016-03-23
2016-03-25
2016-03-29
2016-03-31
2016-04-04
2016-04-06
2016-04-08
2016-04-12
2016-04-14
2016-04-18
2016-04-20
2016-04-22
2016-04-26
2016-04-28
2016-05-02
2016-05-04
2016-05-06
2016-05-10
2016-05-12
2016-05-16
2016-05-18
2016-05-20
2016-05-24
2016-05-26
Highcharts.com

# Começando...

```
$ git clone https://github.com/rg3915/highcharts.git
$ cd highcharts
$ source setup.sh
```

# Modelo

```
1  # models.py
2  from django.db import models
3
4
5  class Dollar(models.Model):
6      date = models.DateField('data')
7      value = models.DecimalField('valor', max_digits=4,
8                                  decimal_places=3)
9
10
11 class Euro(models.Model):
12     date = models.DateField('data')
13     value = models.DecimalField('valor', max_digits=4,
14                                 decimal_places=3)
```

```
class Category(models.Model):
    category = models.CharField('categoria', max_length=50,
                                unique=True)


class Product(models.Model):
    category = models.ForeignKey('Category',
                                 verbose_name='categoria')
    product = models.CharField('Produto', max_length=60,
                               unique=True)
    price = models.DecimalField('Preço', max_digits=6,
                                decimal_places=2)
```

# Importando os dados de um CSV

Variação do dólar

http://www.dolarhoje.net.br/dolar-comercial.php

```
1  # dollar.csv
2  date,value
3  1/1/2016,3.956
4  4/1/2016,4.033
5  5/1/2016,3.994
6  6/1/2016,4.017
7  7/1/2016,4.052
8  8/1/2016,4.038
9  11/1/2016,4.049
10 ...
```

Variação do euro

http://br.investing.com/currencies/eur-brl-historical-data

```python
# shell_dollar.py
import csv
import datetime as dt
from highcharts.core.models import Dollar

dollar_list = []
with open('highcharts/fix/dollar.csv', 'r') as f:
    r = csv.DictReader(f)
    for dct in r:
        # Convert '%d/%m/%Y' to '%Y-%m-%d'.
        d = dt.datetime.strptime(dct['date'], '%d/%m/%Y')\
                       .strftime('%Y-%m-%d')
        dollar_list.append((d, dct['value']))
    f.close()

obj = [Dollar(date=val[0], value=val[1]) for val in dollar_list]
Dollar.objects.bulk_create(obj)
```

# Importando os dados via shell

```
python manage.py shell < highcharts/shell/shell_dollar.py
```

# graphics.py

```
1  # graphics.py
2  import json
3  from django.db.models import Count
4  from django.core.serializers.json import DjangoJSONEncoder
5  from django.http import HttpResponse
6  from .models import Dollar, Euro, Product
7
8
9  def dollar_json(request):
10     data = Dollar.objects.values('date', 'value')
11     lista = [{'dia': i['date'],
12               'valor': float(i['value'])} for i in data]
13     resp = json.dumps(lista, cls=DjangoJSONEncoder)
14     return HttpResponse(resp)
```

# graphics.py

```
1  def euro_json(request):
2      data = Euro.objects.values('date', 'value')
3      lista = [{'dia': i['date'],
4                'valor': float(i['value'])} for i in data]
5      resp = json.dumps(lista, cls=DjangoJSONEncoder)
6      return HttpResponse(resp)
```

# urls.py

```
1  # urls.py
2  from django.conf.urls import include, url
3  from django.contrib import admin
4
5  urlpatterns = [
6      url(r'', include('highcharts.core.urls', namespace='core')),
7      url(r'^admin/', include(admin.site.urls)),
8  ]
```

# core/urls.py

```
# core/urls.py
from django.conf.urls import url
from highcharts.core.graphics import dollar_json, euro_json, proc
from highcharts.core import views as v

urlpatterns = [
    url(r'^$', v.home, name='home'),
    url(r'^dollar-graphic/$', v.dollar_graphic, name='dollar-grap
    url(r'^euro-graphic/$', v.euro_graphic, name='euro-graphic'),
    url(r'^product-graphic/$', v.product_graphic, name='product-g
    url(r'^dollar_json/$', dollar_json),
    url(r'^euro_json/$', euro_json),
    url(r'^product_json/$', product_json),
]
```

# Views

```python
# views.py
from django.shortcuts import render

def home(request):
    return render(request, 'index.html')

def dollar_graphic(request):
    return render(request, 'dollar_graphic.html')

def euro_graphic(request):
    return render(request, 'euro_graphic.html')

def product_graphic(request):
    return render(request, 'product_graphic.html')
```

# Templates

Dentro da pasta `highcharts/core/` crie a pasta `templates`.

```
mkdir templates
touch templates/base.html
touch templates/dollar_graphic.html
touch templates/euro_graphic.html
touch templates/product_graphic.html
```

# base.html

```
1  # base.html
2  {% load static %}
3  <html>
4  <head>
5    <!-- jQuery -->
6    <script src="{% static "js/jquery.min.js" %}"></script>
7    <!-- HighCharts JS -->
8    <script src="{% static "js/highcharts.js" %}"></script>
9  </head>
10 <body>
11   {% include "nav.html" %}
12   <div class="container">
13     {% block content %}{% endblock content %}
14     {% block js %}{% endblock js %}
15   </div>
16 </body>
17 </html>
```

# dollar_graphic.html

```
1  # dollar_graphic.html
2  {% extends "base.html" %}
3  {% load static %}
4
5  {% block content %}
6    <div id="dollar-chart"></div>
7  {% endblock %}
8
9  {% block js %}
10   <script src="{% static 'js/dollar_graphic.js' %}"></script>
11 {% endblock js %}
```

Crie a pasta `static/js`.

```
mkdir -p static/js
touch static/js/dollar_graphic.js
touch static/js/euro_graphic.js
touch static/js/product_graphic.js
```

# dollar_graphic.js

```
1  # dollar_graphic.js
2  $(function () {
3      var url = "/dollar_json/";
4
5      $.getJSON(url, function(res){
6          /* Transformando o dicionário em lista.
7             Com o comando map eu coloco uma lista dentro da outra,
8             necessário para este tipo de gráfico. */
9          var data = res.map(function (v) {
10             return [v.dia, v.valor]
11         });
```

# dollar_graphic.js

```
1          $('#dollar-chart').highcharts({
2              chart: {
3                  type: 'line'
4              },
5              title: {
6                  text: 'Variação do Dólar'
7              },
8              xAxis: {
9                  type: 'category'
10             },
```

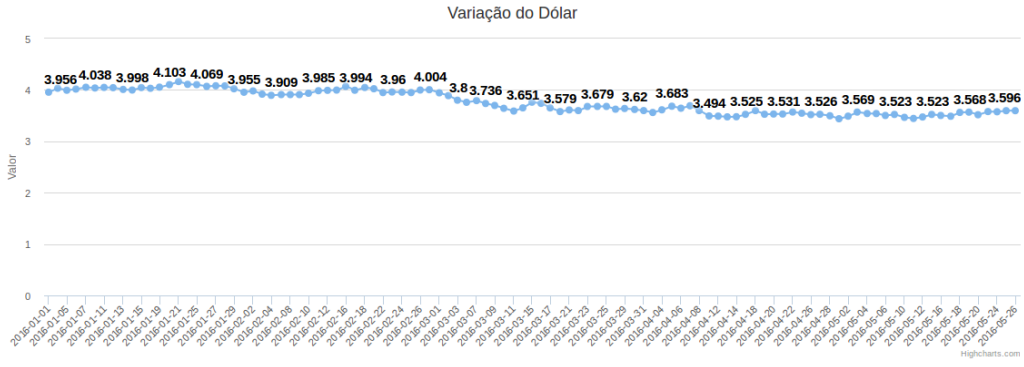
# dollar_graphic.js

```
1               yAxis: {
2                   min: 0,
3                   title: {
4                       text: 'Valor'
5                   },
6                   plotOptions: {
7                       line: {
8                           dataLabels: {
9                               enabled: true
10                          },
11                      }
12                  },
13              },
14              legend: {
15                  enabled: false
16              },
```

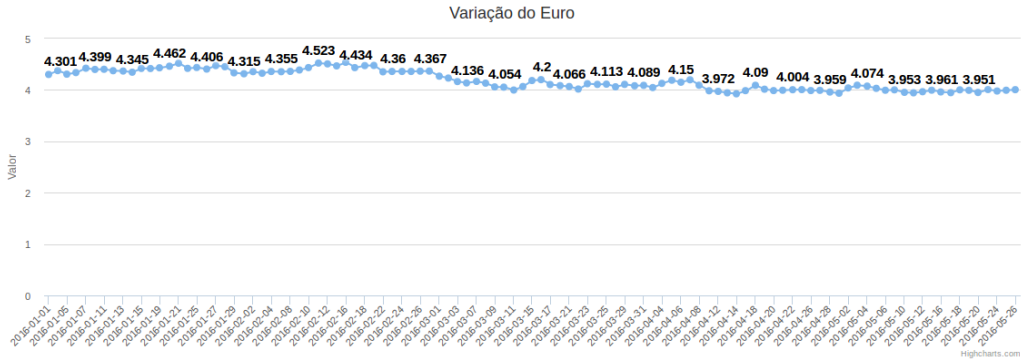
# dollar_graphic.js

```
            series: [{
                data: data,
                dataLabels: {
                    enabled: true,
                    align: 'center',
                    style: {
                        fontSize: '15px'
                    }
                }
            }],
        });
    });
});
```

Variação do Dólar  Variação do Euro  Porcentagem de Categorias



Variação do Dólar Variação do Euro Porcentagem de Categorias



# graphics.py

```
1  def product_json(request):
2      ''' Porcentagem de produtos por categoria '''
3      data = Product.objects.values('category')\
4          .annotate(value=Count('category'))\
5          .order_by('category')\
6          .values('category', 'category__category', 'value')
7      total = Product.objects.all().count()
8      '''Podemos reescrever o dicionário com nossos próprios
9      nomes de campos. '''
10     lista = [{'categoria': item['category__category'],
11         'porcentagem': float((item['value'] / total) * 100)}
12         for item in data]
13     s = json.dumps(lista, cls=DjangoJSONEncoder)
14     return HttpResponse(s)
```

# product_graphic.js

```
1  $(function () {
2      var url = "/product_json/";
3
4      $.getJSON(url, function(res){
5          /* Transformando o dicionário em lista.
6             Com o comando map eu coloco uma lista dentro da outra,
7             necessário para este tipo de gráfico. */
8          var data = res.map(function (v) {
9              return [v.categoria, v.porcentagem]
10         });
```

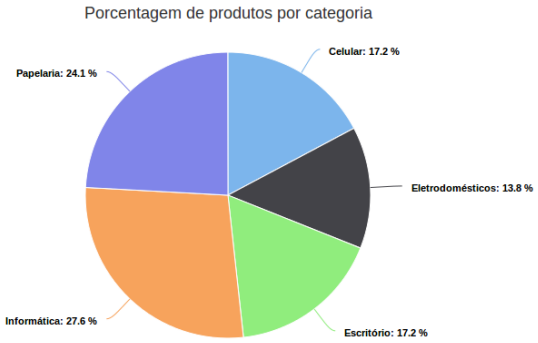
## product_graphic.js

```
1          $('#product-chart').highcharts({
2              chart: {
3                  type: 'pie'
4              },
5              title: {
6                  text: 'Porcentagem de produtos por categoria'
7              },
8              tooltip: {
9                  pointFormat: '<b>{point.percentage:.1f}%</b>'
10             },
11             plotOptions: {
12                 pie: {
13                     allowPointSelect: true,
14                     cursor: 'pointer',
15                     dataLabels: {
16                         enabled: true,
17                         format: '<b>{point.name}</b>: {point.perc
```

# product_graphic.js

```
            series: [{
                name: 'Categoria',
                colorByPoint: true,
                data: data
            }],
        });
    });
});
```

Variação do Dólar Variação do Euro Porcentagem de Categorias



# Obrigado!

Dúvidas?

# Gráficos: Django + Highcharts

Régis da Silva Santos
http://rg3915.github.io

github.com/grupy-sp/encontros

04 de Junho de 2016