

DJANGO

A SUA CÁPSULA DE SOLUÇÕES WEB EM PYTHON

4º Encontro Carioca de Testes – 14 de junho de 2016

SOBRE MIM

Paula Grangeiro

28 anos

Bacharel em Sistemas de Informação

Duque de Caxias

Gateira

Python & Arquitetura de Software

YOROSHIKU!

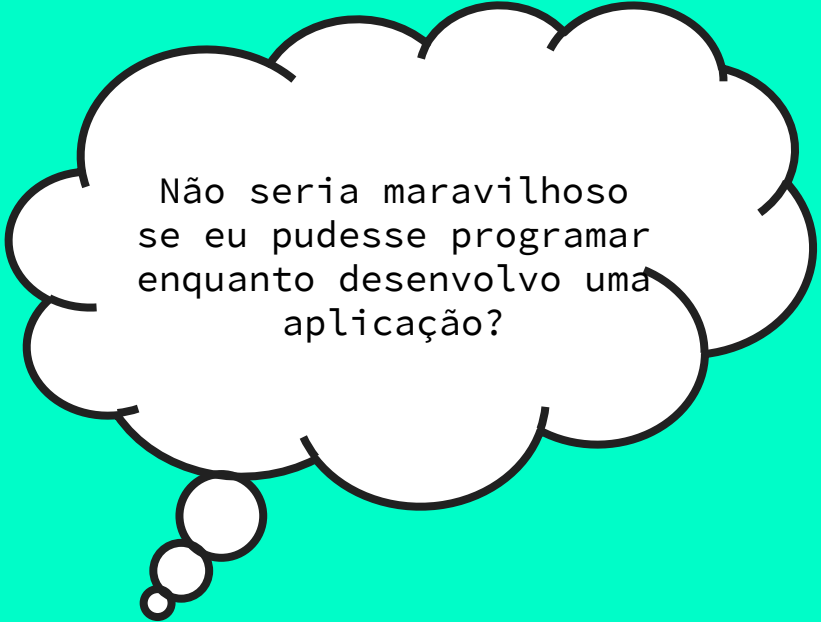




elogroup▶

ONDE ME ENCONTRAR...

POR QUE PYTHON?



Não seria maravilhoso
se eu pudesse programar
enquanto desenvolvo uma
aplicação?

PORQUE...

- É Software Livre
- É multiplataforma
- É uma linguagem de alto nível
- É fortemente tipada
- É de fácil aprendizado
- Possui uma extensa documentação
- Tem a comunidade mais lynda do mundo <3

```
print('É só amor <3')
```

4ª POSIÇÃO

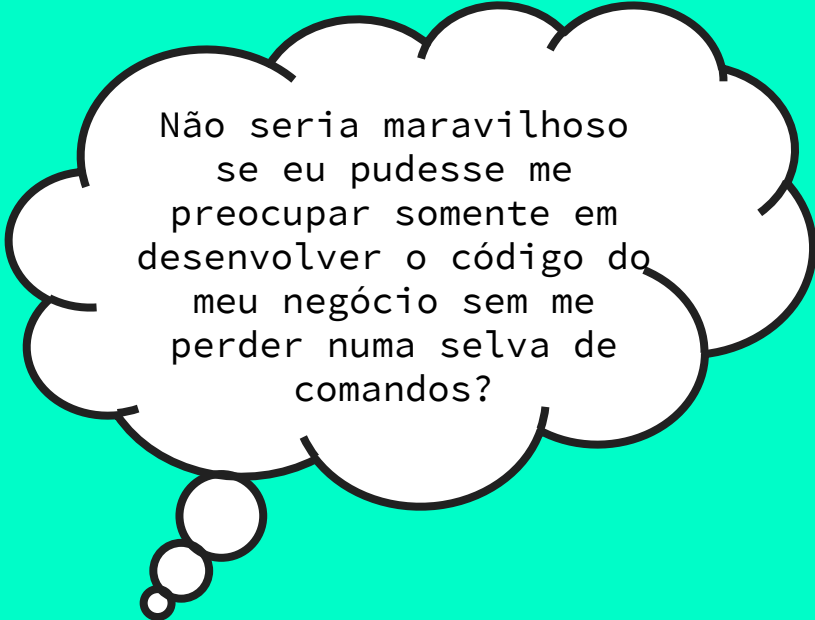
Entre as linguagens mais utilizadas no mundo.

fonte: TIOBE, Junho/2016 (http://www.tiobe.com/tiobe_index)

SIR, EXCUSE ME SIR!

**DO YOU HAVE A MOMENT TO
TALK ABOUT OUR LORD AND
SAVIOR, CTHULHU?**

DJANGO



Não seria maravilhoso
se eu pudesse me
preocupar somente em
desenvolver o código do
meu negócio sem me
perder numa selva de
comandos?

VANTAGENS

START AND GO!

- É Software Livre
- É multiplataforma
- É de ~~nem-tão~~ fácil aprendizado
- Possui uma extensa documentação
- Tem a comunidade mais lynda do mundo <3
- Baixo acoplamento e alta coesão
- Retrocompatibilidade
- É de fácil instalação e configuração

```
PRINT('CODE TIME')
```

BATTERIES INCLUDED

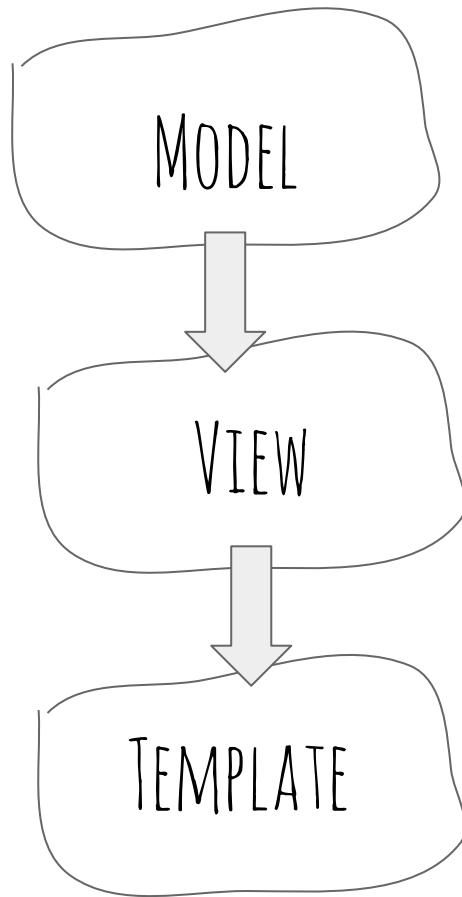
- ORM
- Multi DB
- Migrations
- Authorization
- Authentication
- Admin
- Sites
- Forms
- CSRF

- Dev Server
- WSGI
- TestCases
- Caching
- Templates
- Estáticos
- Views
- URL

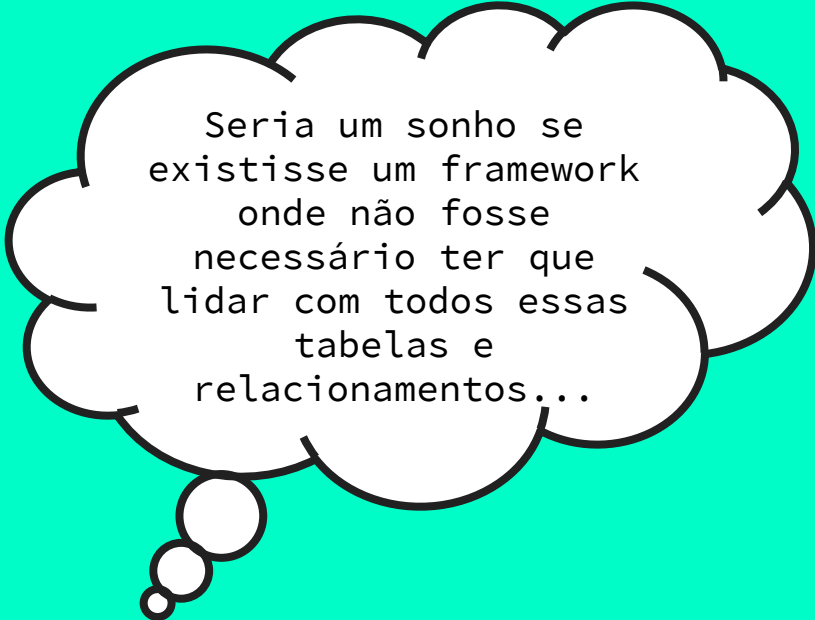
E POR AÍ VAI...

MVT

A arquitetura MVT do Django é similar ao conceito de MVC, onde os Models com a ajuda da ORM intermediam o acesso ao banco constituindo a camada de Model, os middlewares administram as requisições HTTP e enviam para um sistema de renderização de templates constituindo a camada de View e as Views com a ajuda do sistema que resolve as URLs constituem a camada equivalente a de Controller.



MODEL



Seria um sonho se
existisse um framework
onde não fosse
necessário ter que
lidar com todas essas
tabelas e
relacionamentos...

ORM

```
patient = Patient.objects.get(id=1)
```

```
patient.name = 'Sara Shiva'
```

```
patient.save()
```

- API única de acesso independente do banco de dados do projeto
- Abordagem orientada à objetos
- Transações ACID

MIGRATIONS

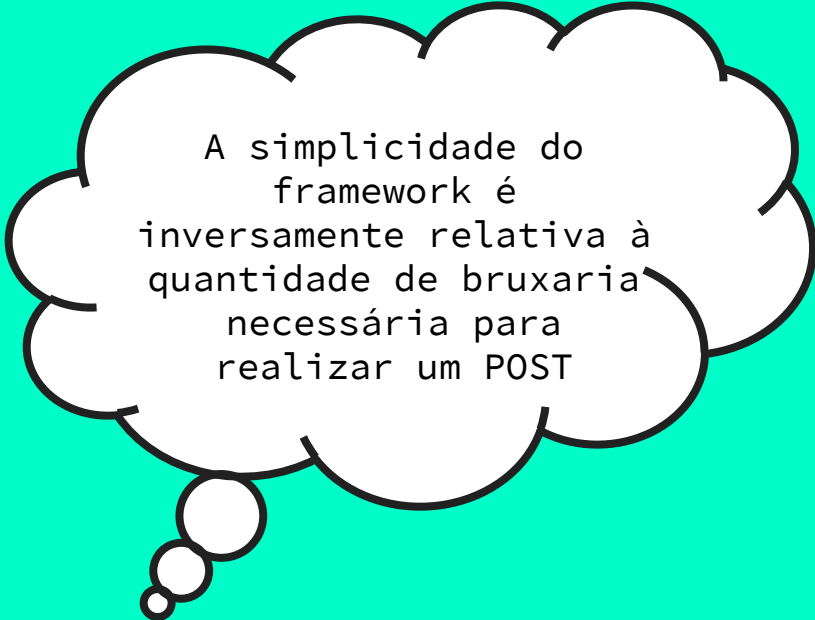
```
python manage.py makemigrations
```

```
python manage.py migrate
```

- Mantém a consistência entre os modelos e tabelas do banco de dados

```
PRINT('CODE TIME')
```


VIEW



A simplicidade do
framework é
inversamente relativa à
quantidade de bruxaria
necessária para
realizar um POST

VIEWS

Responsável por receber uma requisição(request) Web e retornar uma resposta (response)

```
from django.http import HttpResponse
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>It is now %s.</body></html>" %
now
    return HttpResponse(html)
```

URL DISPATCHER

Necessário para ligar a requisição ao responsável pela resposta.

```
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^today/$', views.current_datetime),
]
```

```
PRINT('CODE TIME')
```

CLASS BASED VIEWS

Views especializadas em certos tipos de objects request e response

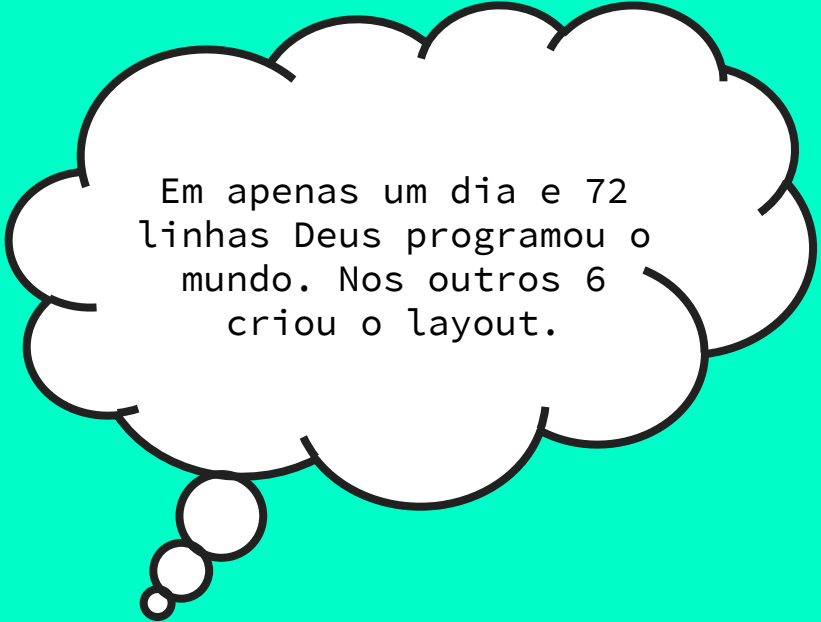
```
from django.http import HttpResponseRedirect
from django.views.generic import View

import datetime

class MyView(View):
    def get(self, request):
        now = datetime.datetime.now()
        html = "<html><body>It is now %s</body></html>" %
now
        return HttpResponseRedirect(html)
```

```
PRINT('CODE TIME')
```

TEMPLATE



Em apenas um dia e 72
linhas Deus programou o
mundo. Nos outros 6
criou o layout.

TEMPLATES

- HTML friendly
- São extensíveis
- Tags
- Filters

```
{% extends "admin/base.html" %}
{% block content %}
<ul>
{% for athlete in athlete_list %}
    <li>{{ athlete.name }}</li>
{% endfor %}
</ul>
{% endblock %}
```



```
PRINT('CODE TIME')
```

OBRIGADA!

- Email: contato@paulagrangoiro.com.br
- Site: www.paulagrangoiro.com.br
- GitHub: [@pgrangoiro](https://github.com/pgrangoiro)
- Twitter: www.twitter.com/paulagrangoiro
- Facebook: www.facebook.com/paula.grangoiro

