
Django Web Framework

— O framework web para perfeccionistas com prazos —

Quem sou eu ?

- Lucas Matheus **Testa** (sobrenome mesmo, nada a ver com TDD)
- Tecnólogo em ADS pelo IFSP - São Carlos
- 21 anos.
- Trabalho com python a mais de um ano.
- Cursando Especialização em desenvolvimento mobile.

JAMIE FOKK CHRISTOPH WALTZ LEONARDO DICAPRIO KERRY WASHINGTON and SAMUEL L. JACKSON

DJANGO

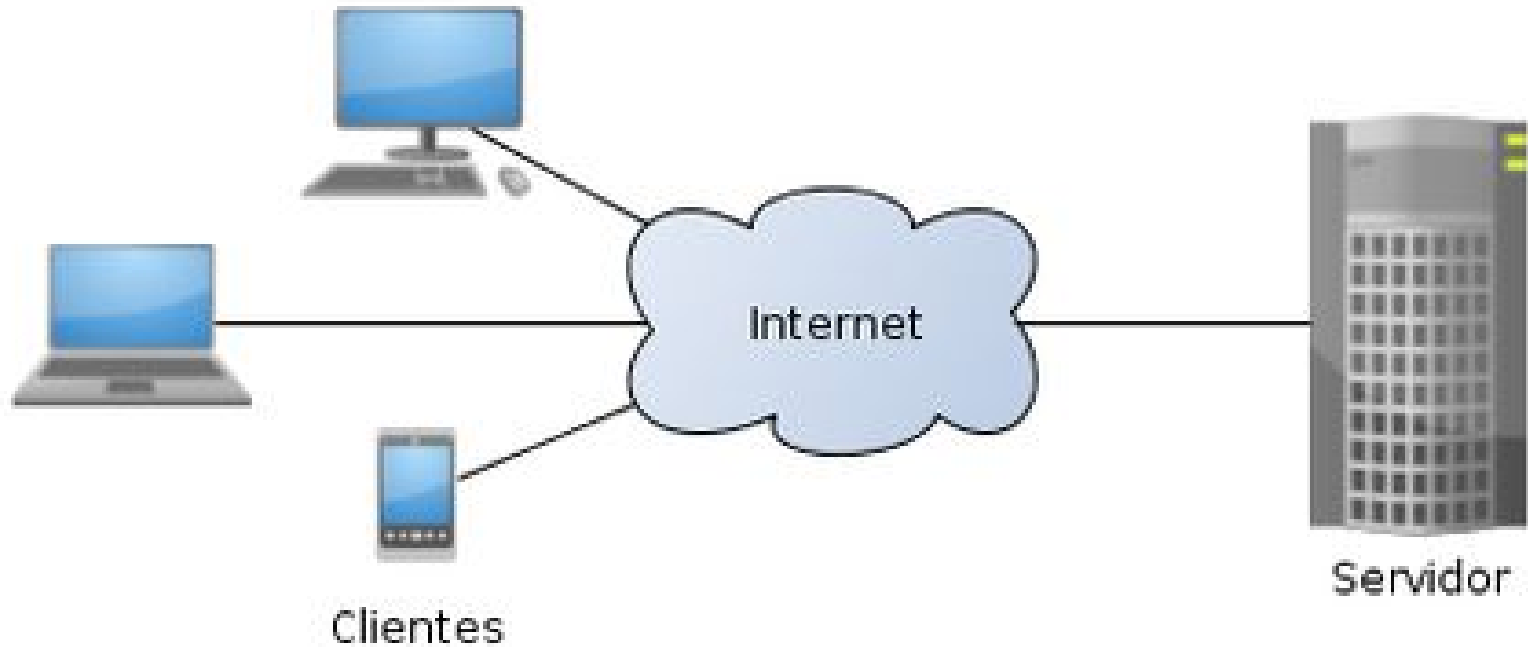
UNCHAINED

DREHBUCH UND REGIE
**QUENTIN
TARANTINO**

django



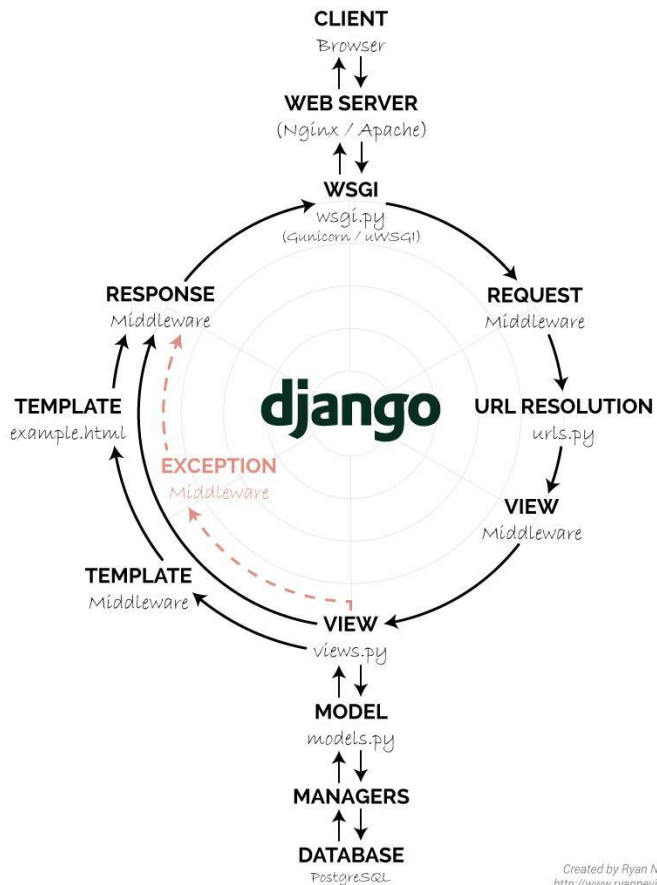
A Web



http://tutorial.djangogirls.org/pt/how_the_internet_works/

O que é Django ?

- Django é um framework web de alto nível escrito em Python que estimula o desenvolvimento rápido e limpo
- Criado em 2005 por Lawrence Journal-World
- Framework web python mais utilizado.
- Site oficial: www.djangoproject.com



O que vamos fazer ?

- Criar ambiente virtual
- Instalar o Django com pip
- Criar um projeto novo
- Layout de projeto
- Models
- Querysets
- Definir URLs
- Views
- Templates
- Django Admin
- Próximos passos

Passo 1 - virtualenvwrapper

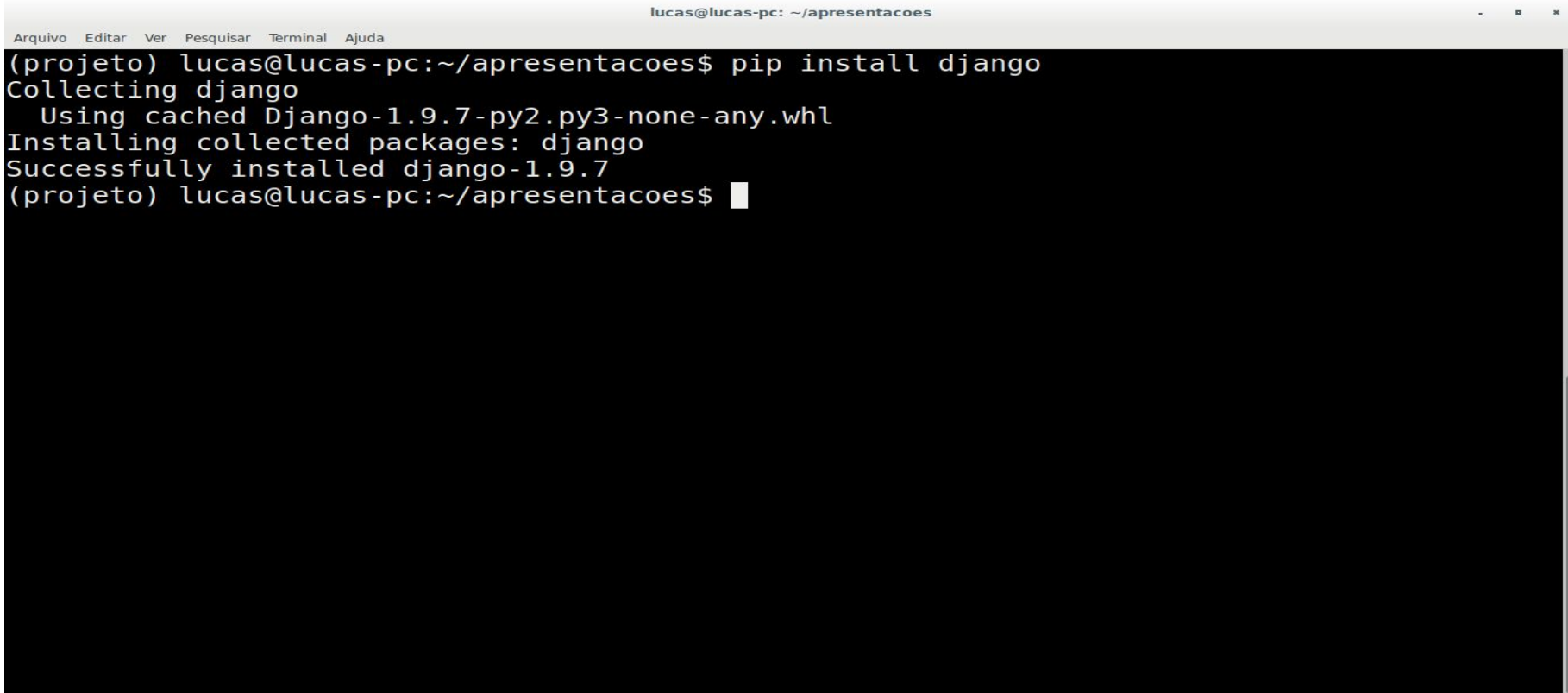
- Instalação

<https://virtualenvwrapper.readthedocs.io/en/latest/>

- Para que serve ?
- Projetos em paralelo
- Separação de dependências

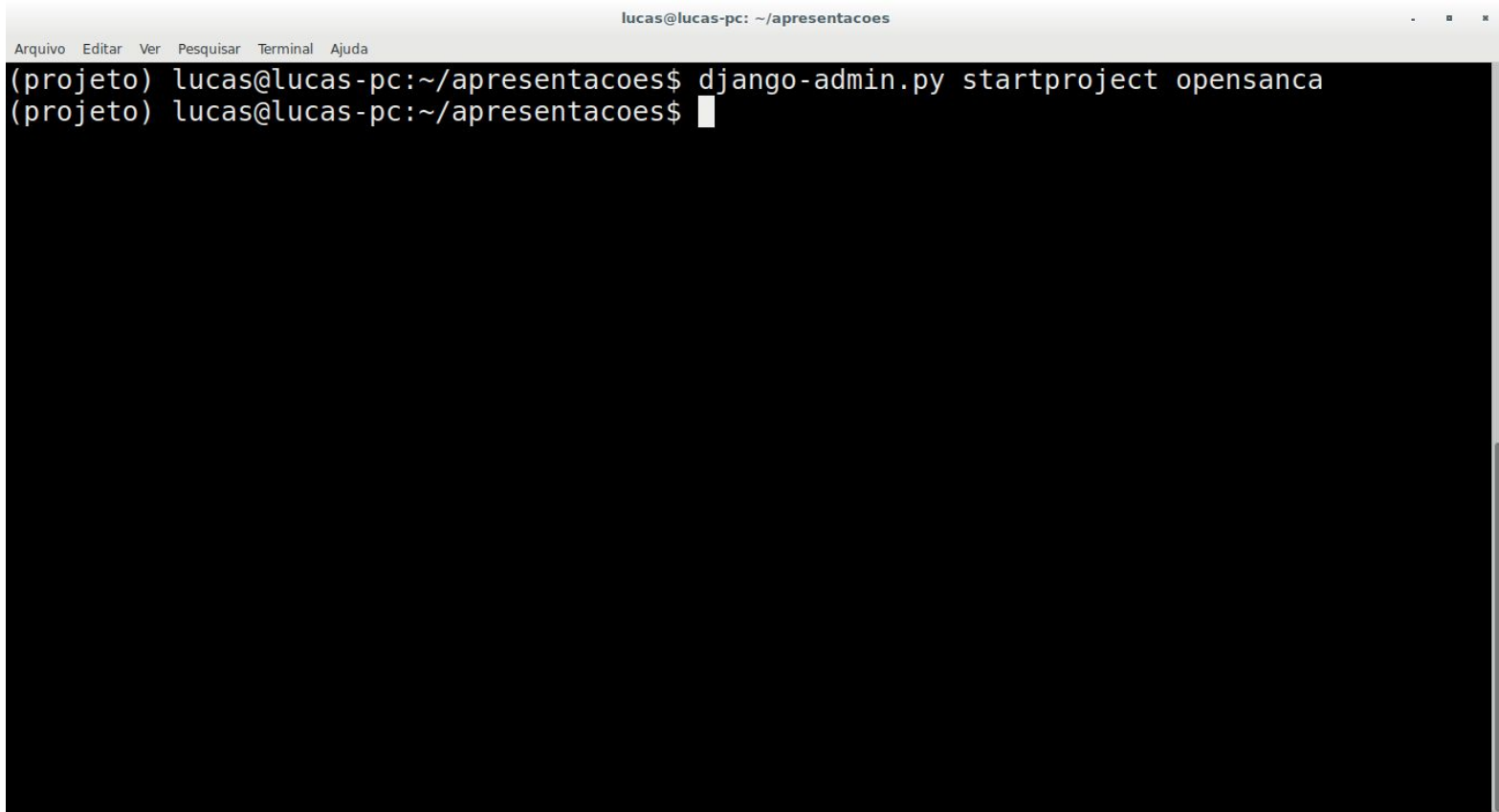
```
lucas@lucas-pc:~/apresentacoes$ mkvirtualenv projeto -p python3
Running virtualenv with interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/lucas/.virtualenvs/projeto/bin/python3
Also creating executable in /home/lucas/.virtualenvs/projeto/bin/python
Installing setuptools, pip, wheel...done.
virtualenvwrapper.user_scripts creating /home/lucas/.virtualenvs/projeto/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/lucas/.virtualenvs/projeto/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/lucas/.virtualenvs/projeto/bin/preactivate
virtualenvwrapper.user_scripts creating /home/lucas/.virtualenvs/projeto/bin/postactivate
virtualenvwrapper.user_scripts creating /home/lucas/.virtualenvs/projeto/bin/get_env_details
(projeto) lucas@lucas-pc:~/apresentacoes$
```

Passo 2 - Instalando o Django

A screenshot of a terminal window with a light gray title bar. The title bar contains the text 'lucas@lucas-pc: ~/apresentacoes' on the left and standard window control icons (minimize, maximize, close) on the right. Below the title bar is a menu bar with the items 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The main area of the terminal is black with white text. The text shows a user prompt '(projeto) lucas@lucas-pc:~/apresentacoes\$' followed by the command 'pip install django'. The output shows 'Collecting django', 'Using cached Django-1.9.7-py2.py3-none-any.whl', 'Installing collected packages: django', and 'Successfully installed django-1.9.7'. The prompt is shown again at the end of the output.

```
lucas@lucas-pc: ~/apresentacoes
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(projeto) lucas@lucas-pc:~/apresentacoes$ pip install django
Collecting django
  Using cached Django-1.9.7-py2.py3-none-any.whl
Installing collected packages: django
Successfully installed django-1.9.7
(projeto) lucas@lucas-pc:~/apresentacoes$
```

Passo 3 - Criando um novo projeto

A screenshot of a terminal window with a light gray title bar. The title bar contains the text 'lucas@lucas-pc: ~/apresentacoes' on the left and standard window control icons (minimize, maximize, close) on the right. Below the title bar is a menu bar with the items 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The main area of the terminal is black with white text. It shows two lines of text: the first line is '(projeto) lucas@lucas-pc:~/apresentacoes\$ django-admin.py startproject opensanca' and the second line is '(projeto) lucas@lucas-pc:~/apresentacoes\$' followed by a white cursor block.

```
lucas@lucas-pc: ~/apresentacoes
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(projeto) lucas@lucas-pc:~/apresentacoes$ django-admin.py startproject opensanca
(projeto) lucas@lucas-pc:~/apresentacoes$
```

Passo 4 - Layout do projeto

Arquivo que gerencia o projeto django.

```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
lucas@lucas-pc: ~/apresentacoes/opensanca$ tree  
(projeto) lucas@lucas-pc:~/apresentacoes/opensanca$ tree
```

```
├── manage.py  
└── opensanca  
    ├── __init__.py  
    ├── settings.py  
    ├── urls.py  
    └── wsgi.py
```

1 directory, 5 files

```
(projeto) lucas@lucas-pc:~/apresentacoes/opensanca$
```

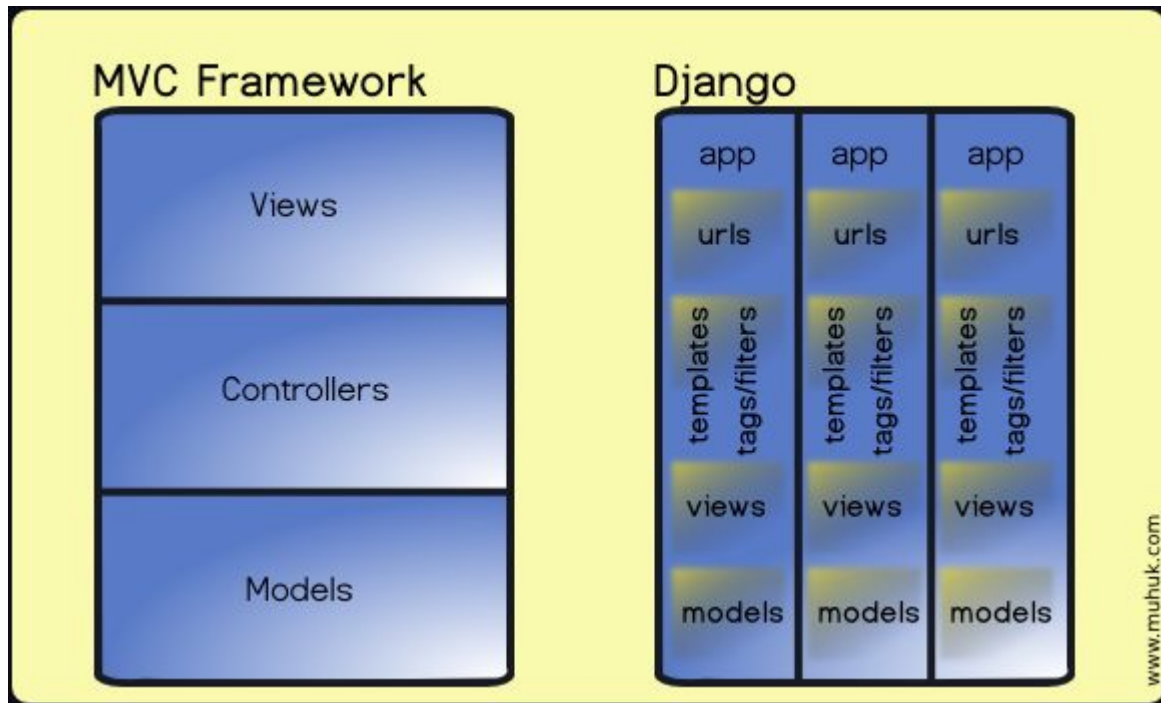
Indica que é um módulo python.

Web Server
Gateway
Interface

Urls, usam
expressões
regulares.

Configurações
do seu
projeto

Layout de projetos



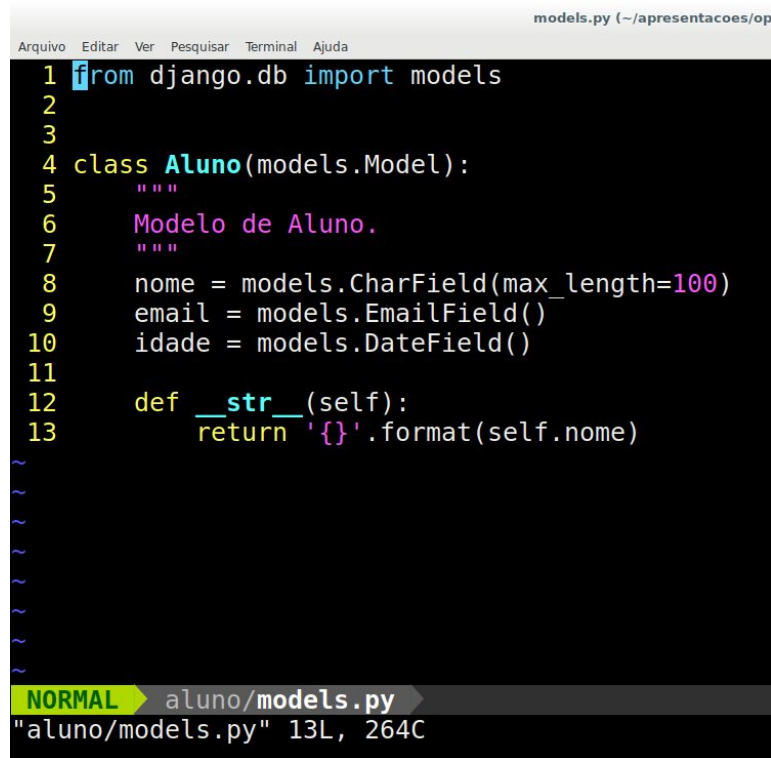
- Apps se comunicam entre si quando necessário
- Cada app deve ser criado com um único objetivo (exceções)

Django não é MVC e sim MVT

- **M**: Model, lidam com as informações
- **V**: Views, lidam com as requisições e normalmente responde http.
- **T**: Templates, exibição de informação (HTML).

Iniciando um novo app

- `python manage.py startapp aluno`
- Incluir novo app no `settings.py`
- Exemplo de Model.



The screenshot shows a code editor window titled 'models.py (~/apresentacoes/op)'. The editor has a menu bar with 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The code is as follows:

```
1 from django.db import models
2
3
4 class Aluno(models.Model):
5     """
6     Modelo de Aluno.
7     """
8     nome = models.CharField(max_length=100)
9     email = models.EmailField()
10    idade = models.DateField()
11
12    def __str__(self):
13        return '{}'.format(self.nome)
```

At the bottom of the editor, there is a status bar showing 'NORMAL' in a green box, followed by 'aluno/models.py' and a right-pointing arrow. Below that, it shows the file path and line/column information: '"aluno/models.py" 13L, 264C'.

O que é um Queryset?

Um QuerySet (conjunto de pesquisa), no fundo, é uma lista de objetos de um dado modelo. Um QuerySet permite que você leia os dados do banco, filtre e ordene o mesmo.

- Criar um aluno

```
Aluno.objects.create(nome='lucas', idade=date(1994, 7, 13),  
                      email='lucasmatheus3492@gmail.com')
```

- Listar todos os alunos
 - *Aluno.objects.all()*
- Todos os alunos que tenham e-mail do gmail
 - *objects.filter(email__endswith='@gmail.com')*
 - *Aluno.Contar os registros*
 - *Aluno.objects.count()*

- Pegar um unico objeto
 - `Aluno.objects.get(pk=1)`
 - `Aluno.objects.get(nome="lucas")`
- Ordenar por idade
 - `Aluno.objects.order_by('idade')`
 - `Aluno.objects.order_by('-idade')`
- Filtrar data maior ou menor que
 - `Aluno.objects.filter(idade__gte=date(2015, 7, 1))`
 - `Aluno.objects.filter(idade__gt=date(2015, 7, 1))`
 - `Aluno.objects.filter(idade__year=2015)`
- Contains
 - `Aluno.objects.filter(nome__contains="lucas")`

Queryset slicing

- Pegar os 5 primeiros
 - `Aluno.objects.all()[0:5]`
- Filtrar
 - `Aluno.objects.filter(email__endswith="@gmail.com")[0:5]`
- Filtrar e Ordenar
 - `Aluno.objects.filter(email__endswith="@gmail.com").order_by('idade')[0:5]`

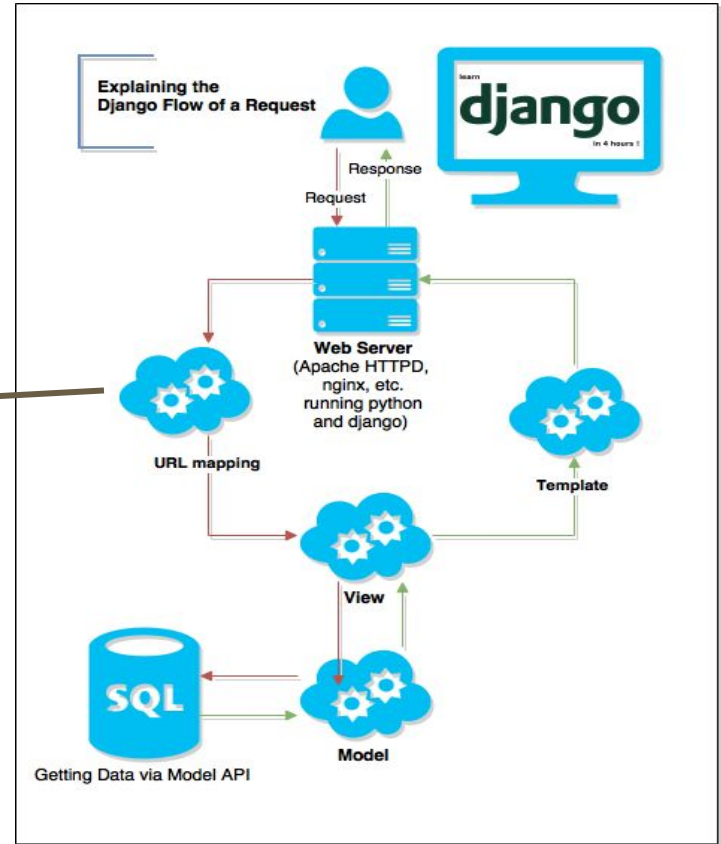
urls.py

- Include
- regex
- views

```
urls.py (~/apresentacoes/open/open) - V
```

```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
1 from django.conf.urls import include, url
2 from django.contrib import admin
3
4 urlpatterns = [
5     # Examples:
6     # url(r'^$', 'open.views.home', name='home'),
7     # url(r'^blog/', include('blog.urls')),
8
9     url(r'^admin/', include(admin.site.urls)),
10 ]
```

```
NORMAL open/urls.py unix
"open/urls.py" 10L, 252C
```



Nova URL

urls.py (~/apresentacoes/open/open) - VIM

Arquivo Editar Ver Pesquisar Terminal Ajuda

```
1 from django.conf.urls import include, url
2 from django.contrib import admin
3
4 urlpatterns = [
5     # Examples:
6     # url(r'^$', 'open.views.home', name='home'),
7     # url(r'^blog/', include('blog.urls')),
8
9     url(r'^admin/', include(admin.site.urls)),
10    url(r'url_teste/$', 'aluno.views.teste_view'),
11 ]
```

Views

- Uma view pode ser uma função ou uma classe, para fins didáticos começaremos com uma função.
- Exemplo:

```
def myview(request):  
    pass
```

Ela recebe como parâmetro uma requisição.

Primeira View

O pacote `django.http` tem uma classe chamada `HttpResponse`, que responde a requisição.

```
views.py (~/apresentacoes/open

Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

1 from django.http import HttpResponse
2
3
4 def teste_view(request):
5     return HttpResponse('nossa nova view !')
```

URLS com parâmetros GET

Existem urls que recebem parâmetros na URL, usando regex no arquivo de urls.py é possível definir quais parâmetros são esperados.

Exemplo:

```
url(r'minha_url/(?P<idade>\d+)/$', ....)
```

Essa URL é esperado receber um parâmetro “idade” do tipo inteiro.

O que é regex e Raw Strings ?

Regex são padrões utilizados para encontrar determinadas combinações em string. Ela faz parte da vida de todos os programadores, afinal é graças a ela que validadores de campos, por exemplo, podem ser facilmente implementados.

Views que recebem parâmetros

```
def minha_view(request, idade):
```

```
    return HttpResponse(idade)
```

```
def minha_view(request, idade):
```

```
    raise Exception(idade, request.GET)
```

Levantamos uma exceção só para verificar esse valor.

View com Template

O django possui uma função no pacote `django.shortcuts` chamada `render`

`render(requisição, endereço do template, contexto)`

Parâmetro 1: recebe a própria requisição


Parâmetro 2: caminho do template.

Parâmetro 3: um contexto.

Um contexto nada mais é que um dicionário, esse dicionário será passado para o seu HTML.

Boa prática

- templates/namespace
- Cada app tem seus templates
- Django cookiecutter



```
aluno
├── admin.py
├── __init__.py
├── migrations
│   ├── 0001_initial.py
│   ├── __init__.py
│   └── __pycache__
│       └── 0001_initial.cpython-35.pyc
│           └── __init__.cpython-35.pyc
├── models.py
├── __pycache__
│   ├── admin.cpython-35.pyc
│   ├── __init__.cpython-35.pyc
│   ├── models.cpython-35.pyc
│   └── views.cpython-35.pyc
├── templates
│   └── alunos
│       └── index.html
├── tests.py
└── views.py
```

Onde estamos ?

```
1 <html>
2   <head>
3     <title>Nossa querida página !</title>
4   </head>
5   <body>
6     <h1>Olá, meu nome é {{ nome }}, e eu tenho {{ idade }} ano{{ idade|pluralize }}</h1>
7   </body>
8 </html>
```

```
12
13 def view_com_template(request, idade, nome):
14     context = {
15         'nome': nome,
16         'idade': idade
17     }
18     return render(request, 'alunos/index.html', context)
```

```
url(r'minha_url_com_template/(?P<idade>\d+)/(?P<nome>[\w\.-]+)/$', 'aluno.views.view_com_template'),
```

Mas e se eu quiser exibir mais no template ?

```
lista_alunos.html (~/open/alunos/
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
1 <html>
2     <head>
3         <title>Listando Alunos</title>
4     </head>
5     <body>
6         <ul>
7             {% for aluno in alunos %}
8                 <li>{{ aluno }}</li>
9             {% endfor %}
10        </ul>
11    </body>
12 </html>
```

views.py (~/open/alunos) - VIM

Arquivo Editar Ver Pesquisar Terminal Ajuda

```
1 from django.shortcuts import render
2 from .models import Aluno
3
4
5 def lista_alunos(request):
6     context = {
7         'alunos': Aluno.objects.all()
8     }
9     return render(request, 'alunos/lista.html', context)
```

Built-in template tags mais usadas

- If
- For
- Load
- Include
- Block
- Url
- Date
- Floatformat

<https://docs.djangoproject.com/en/1.9/ref/templates/builtins/>

Bonus time - Class-based view

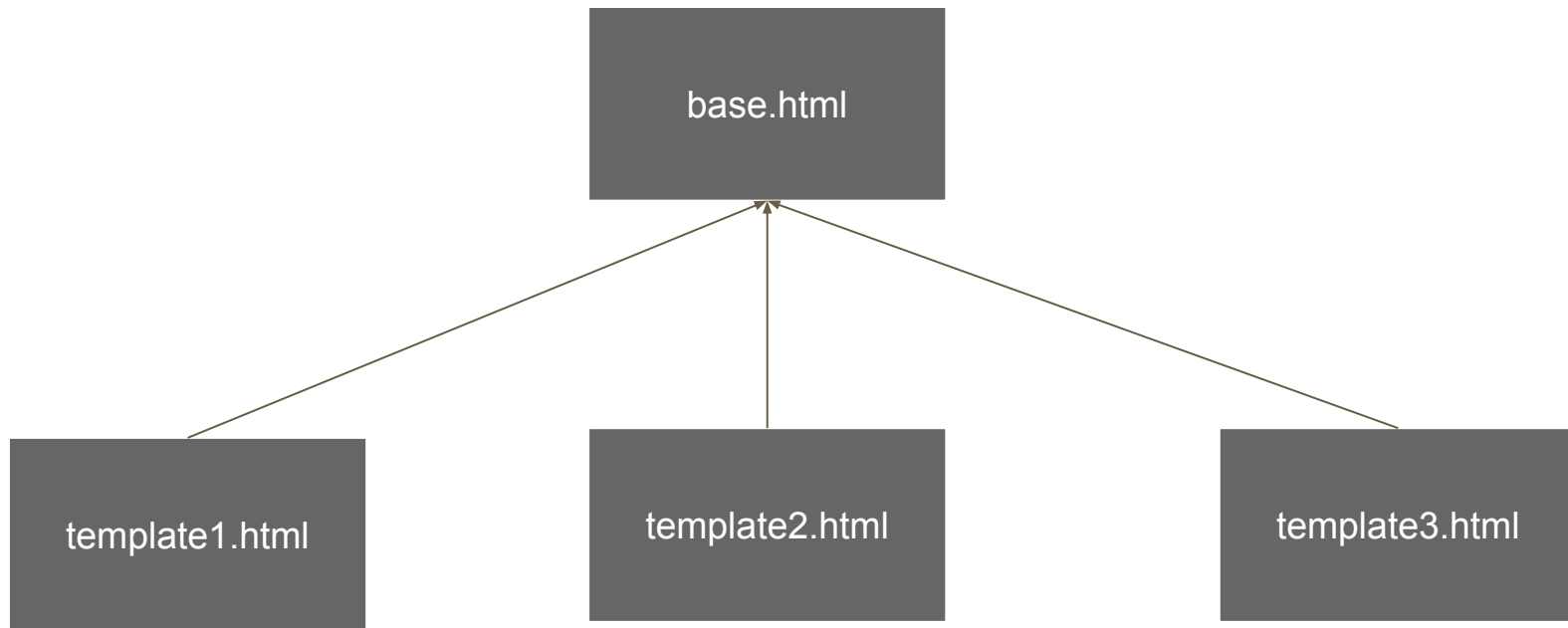
- Fornece meios para implementar views baseadas em classes no lugar de funções
- Vantagens:
 - Código mais organizado
 - Maior flexibilidade
- Built-in class-based generic views

Built-in class-based generics views

- ListView
- DetailView
- TemplateView
- CreateView
- UpdateView
- DeleteView
- View

```
views.py + (~/open/alunos)
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
1 from django.shortcuts import render
2 from .models import Aluno
3 from django.views.generic import ListView
4
5
6 class ListAlunosView(ListView):
7     template_name = 'alunos/lista.html'
8     model = Aluno
```

Templates



Exemplo de Extensão de templates

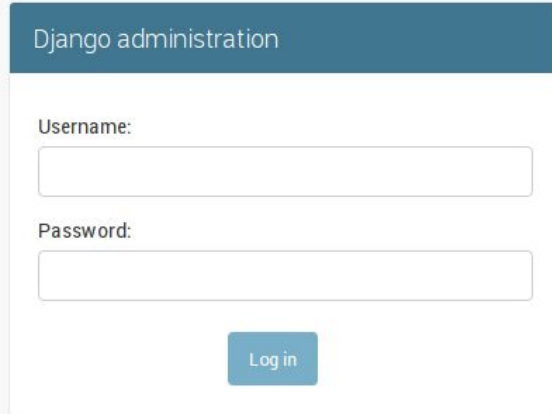
```
base.html (~/.open/alunos/template)
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
1 <html>
2     <head>
3         <title>Exemplo de templates</title>
4     </head>
5     <body>
6         {% block conteudo %}
7
8         {% endblock %}
9     </body>
10 </html>
```

lista_alunos.html + (

Arquivo Editar Ver Pesquisar Terminal A

```
1 {% extends "base.html" %}
2
3 {% block conteudo %}
4     <h1>Lucas</h1>
5 {% endblock %}
6
```

Django Admin



The image shows a screenshot of the Django Admin login interface. It features a dark blue header bar with the text "Django administration". Below this, the form is white with a light gray border. It contains two input fields: "Username:" and "Password:", each followed by a text input box. At the bottom right of the form is a blue "Log in" button.

Django administration

Username:

Password:

Log in

Site administration

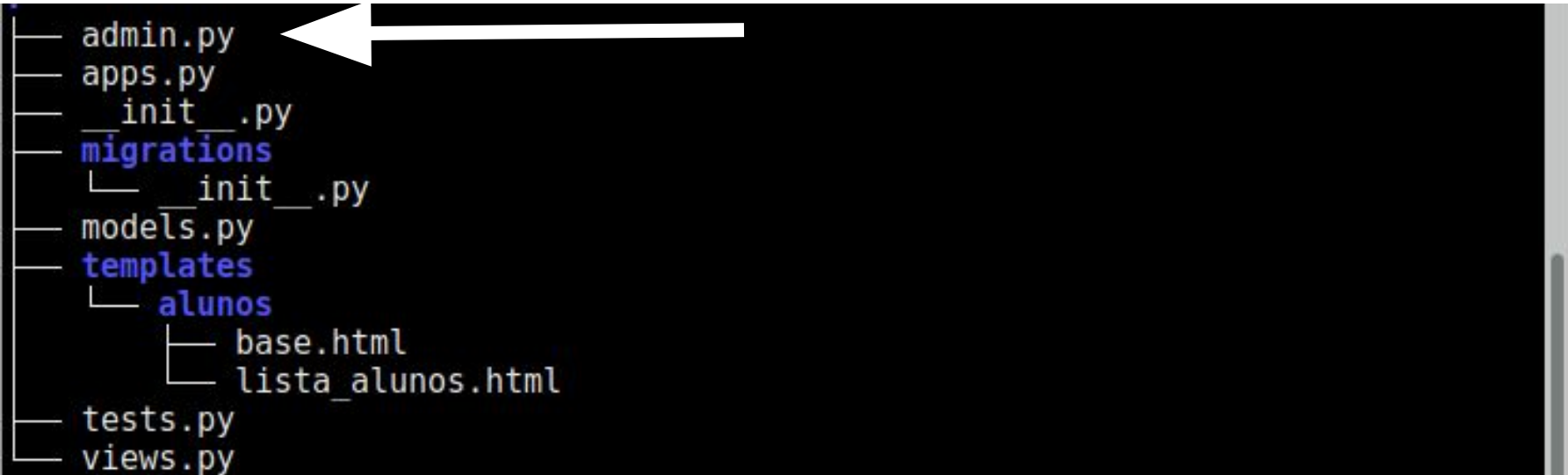
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

Recent Actions

My Actions

None available

Mas...para o que serve esse arquivo admin.py ?



```
— admin.py
— apps.py
— __init__.py
— migrations
  |__ __init__.py
— models.py
— templates
  |__ alunos
    |__ base.html
    |__ lista_alunos.html
— tests.py
— views.py
```



```
1 from django.contrib import admin
2 from .models import Aluno
3
4 admin.site.register(Aluno)
```

Django administration

Site administration

ALUNOS	
Alunos	+ Add ✎ Change
AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add ✎ Change
Users	+ Add ✎ Change

Recent Actions

My Actions

None available

Add aluno

Nome:

Email:

Idade:

Today |



Note: You are 3 hours behind server time.

Professor:

 ▼

JUNE 2016						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		
Yesterday Today Tomorrow						
Cancel						

Save and add another

Save and continue editing

SAVE

Select aluno to change

0 alunos

ADD ALUNO +

FILTER

By nome

All

By idade

Any date

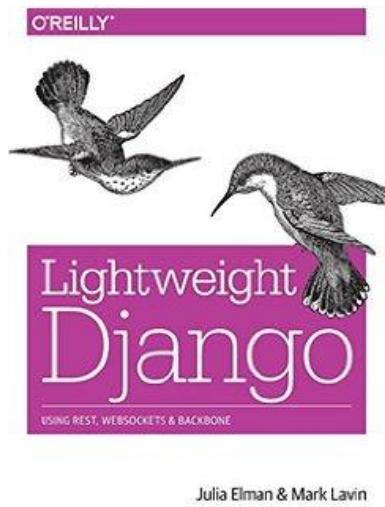
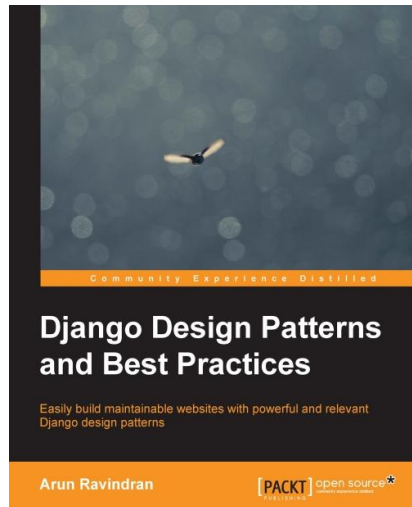
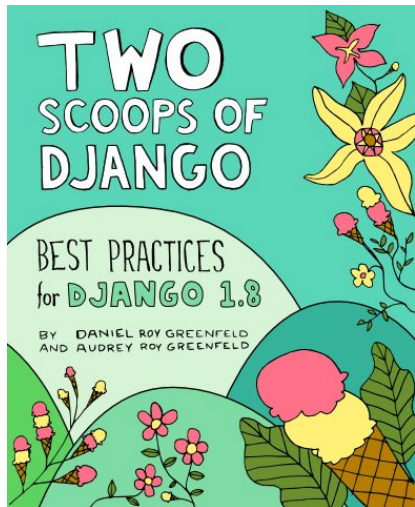
Today

Past 7 days

This month

This year

Próximos passos...



<https://www.djangoproject.com/>

Meus contatos

E-mail: lucasmatheus3492@gmail.com

