



DJANGO: O FRAMEWORK WEB PARA PERFECCIONISTAS COM PRAZOS.



11/05/2013

RENNE ROCHA

Engenheiro Eletricista - Unicamp

Desenvolvedor Python Nokia Siemens Networks

Organizador do Caipira Ágil

<http://caipiraagil.com/>

Fundador do Laboratório Hacker de Campinas

<http://lhc.net.br/>

O QUE É DJANGO?

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

PRINCIPAIS FUNCIONALIDADES

OBJECT-RELATIONAL MAPPER (ORM)

Seus modelos de dados são definidos **totalmente** em Python.

Fornece uma API para acesso ao banco de dados.

Independente de sistema de gerenciamento de banco de dados.

INTERFACE DE ADMINISTRAÇÃO AUTOMÁTICA

Interface pronta para ser utilizada em produção para entrada, consulta, atualização e exclusão de dados.

C R U D.

URLS AMIGÁVEIS

Esquema limpo e elegante para definição de URLs

Definidos em um módulo Python que faz o mapeamento de padrões de URL (expressões regulares) com funções Python (views).

SISTEMA DE TEMPLATES

Linguagem de templates amigável para designers.

Facilita separação de código e apresentação.

INTERNACIONALIZAÇÃO

Suporte a aplicações multi-idíomas.

Permite a especificação de trechos para tradução e definir funcionalidades específicas por idioma.

QUEM UTILIZA???



Instagram
Fast beautiful photo sharing

 Entrar



Conheça o Instagram

É um modo **rápido, lindo e divertido** de compartilhar sua vida com amigos usando uma série de fotos.

Tire uma foto, escolha um filtro para transformar sua aparência e publique no Instagram. Compartilhe no Facebook, no Twitter e também no Tumblr - é muito fácil. É o compartilhamento de fotos, reinventado.

Ah é, mencionamos que é de graça?



DISQUS

For Websites

Showcase

Sign-in

Elevating the discussion, anywhere
on the web.

Get this on your site

Try now by jumping into
the discussion. Click to
reveal the new comment.

3 comments

★ 63



Leave a message...

Best ▾

Community

My Disqus

Share



Show One New Comment



Marilyn Monroe • 3 minutes ago

What just happened? Where are we anyway?

Works with...

WordPress

tumblr

Blogger

Drupal

HTML & JS



notícias ▾

esportes ▾

entretenimento ▾

tecnologia ▾

vídeos ▾

ASSINE JÁ

CENTRAL

E-MAIL ▾

ENTRAR >

São Paulo e Timão duelam por vaga na decisão; siga lances



Inter joga pelo título contra o Juventude; siga



Galo faz 3º no Tombense; siga em tempo real

Santa abre contra Sport; siga a final

- Ceará abre o marcador contra o Fortaleza; acompanhe
- Atlético-PR e Coritiba fazem 1º jogo da



Rafael Marques põe Bota à frente do Flu na final da Taça Rio; acompanhe



'Dança': astros chegam para gravar vinheta



Jô avisa Almir sobre câmera



Renata trai o noivo com Tito

PUBLICIDADE

2 QTOS. - ÁREA DE LAZER NOS MELHORES BAIRROS



globo.tv ESPORTE ESPETÁCULAR



▶ ASSISTA AGORA

12:19

WINGSUIT A MAIS DE 300KM/H

Brasileiro tenta quebrar recorde mundial voando com 99 pessoas



SPORTV

NEWS

MULTI SHOW

gnt

Mais vídeos >

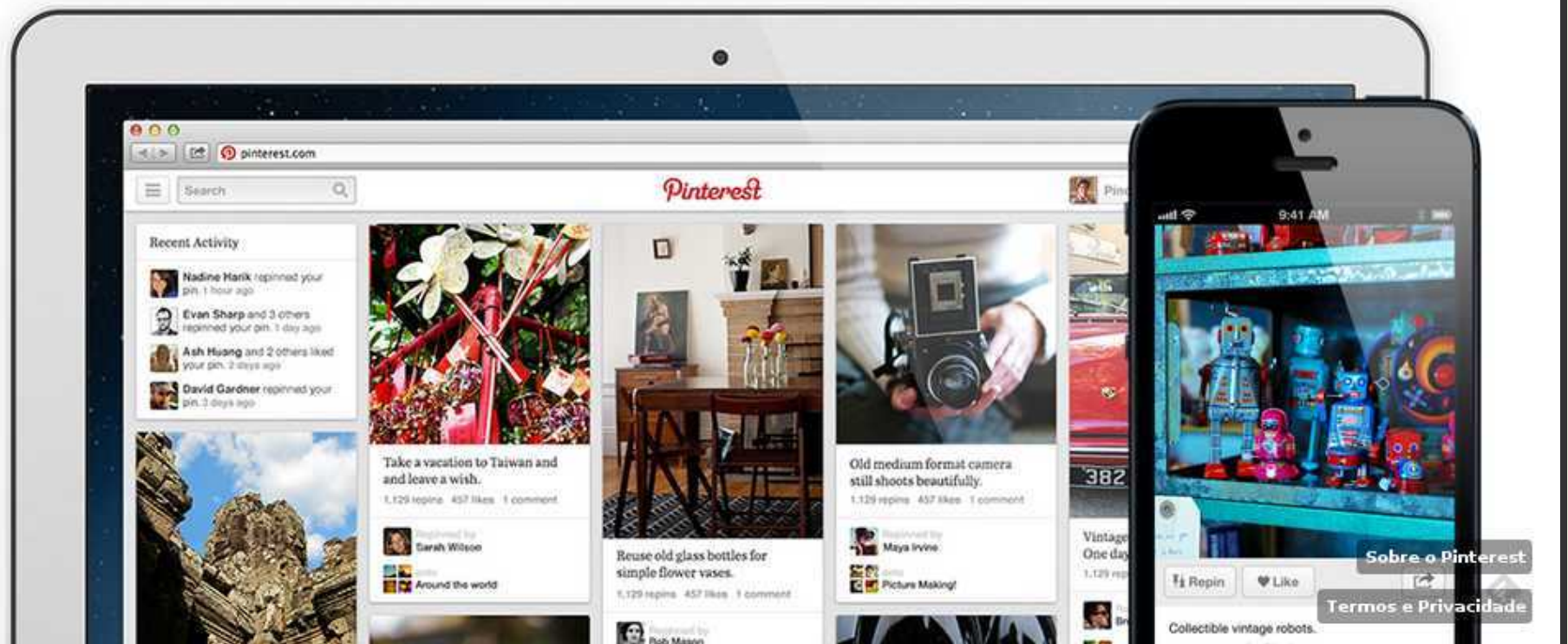




Alguns (milhares) de seus itens preferidos.

Entrar

Participe do Pi



world's biggest issues?
g to solve them?



the **guardian**

Website of the year | 5 May 2013 | Last updated two minutes ago

Your search terms...

The Guardian ▾

Search

the **guardian**

Weather

London



20°C

9°C

[News](#) | [Sport](#) | [Comment](#) | [Culture](#) | [Business](#) | [Money](#) | [Life & style](#) | [Travel](#) | [Environment](#) | [Tech](#) | [TV](#) | [Video](#) | [Dating](#) | [Offers](#) | [Jobs](#)

News

[UK](#) | [World](#) | [Development](#) | [US](#) | [Politics](#) | [Media](#) | [Education](#) | [Society](#) | [Science](#) | [Women](#) | [Law](#) | [Data](#) | [Football](#) | [Observer](#)

Breaking news: 'University of John Lewis' to offer staff work-based degree scheme



UK to pay compensation for imperial rule crimes

Last updated two minutes ago



Exclusive:

Payments to Kenyans tortured during 1950s Mau Mau insurgency could open door for other

victims

290 comments

Huw Bennett: official policy was to cover up

Hugh Laurie From House to the blues



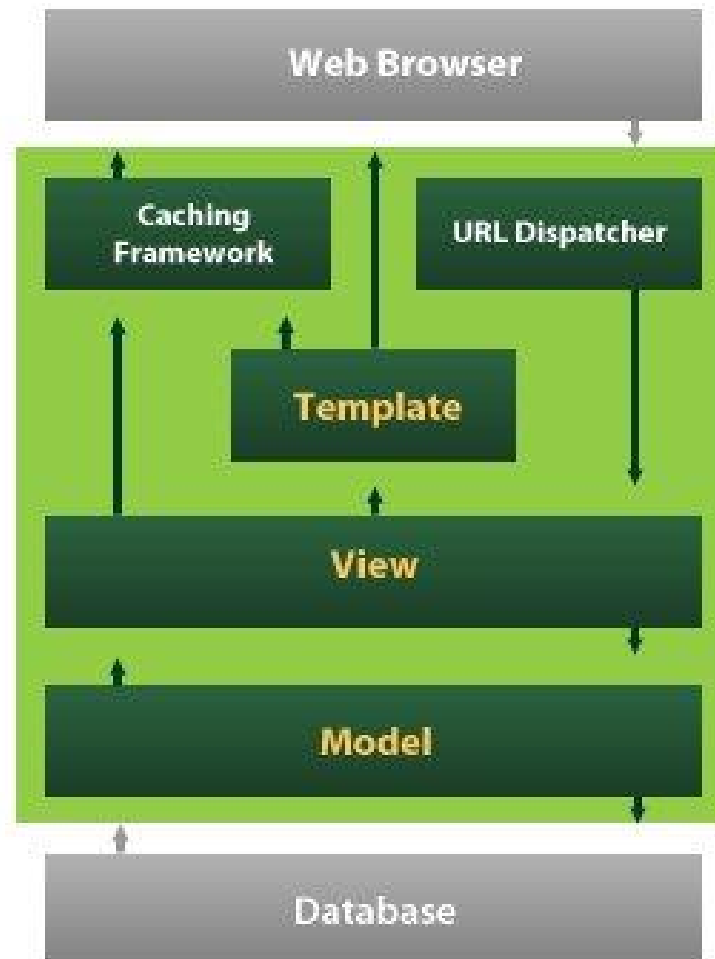
guardianjobs



Online dating

ARQUITETURA

django



MODEL-VIEW-TEMPLATE

Model - seus dados

View - lógica para processar seus dados

Template - apresentando seus dados para o usuário

MODEL

```
from django.contrib.auth.models import User
from django.db import models

class Palestrante(models.Model):
    username = models.ForeignKey(User)
    twitter = models.CharField(max_length=20)

class Trilha(models.Model):
    nome = models.CharField(max_length=50, unique=True)
    data = models.DateTimeField()

class Palestra(models.Model):
    trilha = models.ForeignKey(Trilha)
    titulo = models.CharField(max_length=255)
    palestrantes = models.ManyToManyField(Palestrante)
    resumo = models.TextField()
```

REPRESENTAÇÃO NO BD

```
BEGIN;
CREATE TABLE "schedule_palestrante" (
  "id" integer NOT NULL PRIMARY KEY,
  "username_id" integer NOT NULL,
  "twitter" varchar(20) NOT NULL
)
;
CREATE TABLE "schedule_trilha" (
  "id" integer NOT NULL PRIMARY KEY,
  "nome" varchar(50) NOT NULL UNIQUE,
  "data" datetime NOT NULL
)
;
CREATE TABLE "schedule_palestra_palestrante" (
  "id" integer NOT NULL PRIMARY KEY,
  "palestra_id" integer NOT NULL,
  "palestrante_id" integer NOT NULL REFERENCES "schedule_palestrante" ("id"),
  UNIQUE ("palestra_id", "palestrante_id")
)
;
CREATE TABLE "schedule_palestra" (
  "id" integer NOT NULL PRIMARY KEY,
  "trilha_id" integer NOT NULL REFERENCES "schedule_trilha" ("id"),
  "titulo" varchar(255) NOT NULL,
  "resumo" text NOT NULL
)
;
COMMIT;
```

API

```
In [1]: renne = Palestrante.objects.get(twitter='rennerocha')

In [2]: trilha = Trilha.objects.create(nome='Dev Roots',
...:      data=datetime.datetime(2013, 5, 11))

In [3]: palestra = Palestra.objects.create(
...:      titulo='Django: o framework web.',
...:      trilha=trilha, resumo='Django')

In [4]: Trilha.objects.filter(
...:      palestra__titulo='Django: o framework web.')
Out[4]: [<Trilha: Dev Roots>]

In [5]: palestra.palestrantes.all()
Out[5]: []

In [6]: palestra.palestrante.add(renne)

In [7]: palestra.palestrante.all()
Out[7]: [<Palestrante: renne>]
```

Nunca escreva SQL novamente...
...não ser que seja necessário.

VIEW

```
from django.http import Http404

def detalhe_palestra(request, palestra_id):
    try:
        palestra = Palestra.objects.get(pk=palestra_id)
    except Palestra.DoesNotExist:
        raise Http404
    return render_to_response('palestras/detalhes.html', {'palestra': palestra})
```

OUTROS RETORNOS

Não precisamos retornar apenas HTML...

```
import csv
from django.http import HttpResponse

def some_view(request):
    # Create the HttpResponse object with the appropriate CSV header.
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = 'attachment; filename="somefilename.csv"'

    writer = csv.writer(response)
    writer.writerow(['First row', 'Foo', 'Bar', 'Baz'])
    writer.writerow(['Second row', 'A', 'B', 'C', '"Testing"', "Here's a quote"])

    return response
```


TEMPLATE

base.html

```
<html>
  <head>
    <title>{% block title %}{% endblock %}</title>
  </head>
  <body>
    {% block content %}{% endblock %}
  </body>
</html>
```

TEMPLATE

palestras.html

```
{% extends 'base.html' %}

{% block title %}Dev Camp - Palestras{% endblock %}

{% block content %}
<h1>Palestras</h1>
<ul>
    {% for palestra in palestras %}
    <li>{{ palestra.titulo }}<br>
        {{ palestra.resumo }}</li>
    {% empty %}
    <li>Nenhuma palestra :( </li>
    {% endfor %}
</ul>
{% endblock %}
```

URLS

```
from django.conf.urls import patterns, url
from django.views.generic.simple import direct_to_template

urlpatterns = patterns('',
    (r'^$', direct_to_template, {'template': 'index.html'}),
    url(r'^palestras/$', 'schedule.views.palestras', name='palestras'),
    url(r'^palestra/(?P<palestra_id>\d+)/$', 'schedule.views.detalhe_palestra', name='detalhe_palestra'),)
)
```

FORMULÁRIOS

Biblioteca de manipulação de formulários.

Geração automática do HTML do formulário.

Valida dados submetidos.

Reexibe o formulário caso existam erros de validação.

FORMULÁRIO SIMPLES

```
from django import forms

class ContactForm(forms.Form):
    subject = forms.CharField(max_length=100)
    message = forms.CharField()
    sender = forms.EmailField()
    cc_myself = forms.BooleanField(required=False)

    def clean_message(self, value):
        bad_words_list = ['fi*****ta', 'va*****er', ]
        if value in bad_words_list:
            raise ValidationError('Sem palavrões!')
```

FORMULÁRIO SIMPLES

```
from django.shortcuts import render
from django.http import HttpResponseRedirect

def contact(request):
    if request.method == 'POST': # If the form has been submitted...
        form = ContactForm(request.POST) # A form bound to the POST data
        if form.is_valid(): # All validation rules pass
            # Process the data in form.cleaned_data
            # ...
            return HttpResponseRedirect('/thanks/') # Redirect after POST
    else:
        form = ContactForm() # An unbound form

    return render(request, 'contact.html', {
        'form': form,
    })
```

MODEL FORM

```
from django import forms
from schedule.models import Palestra

class PalestraForm(forms.ModelForm):

    class Meta:
        model = Palestra
```

INICIANDO UM PROJETO

```
$ django-admin startproject devcamp
```

```
devcamp/  
  manage.py  
devcamp/  
  __init__.py  
  settings.py  
  urls.py  
  wsgi.py
```


APLICAÇÕES

O ambiente montado é um **projeto**.

Agora você precisa criar **aplicações** dentro deste projeto.

Filosofia do Unix: *Write programs that do one thing and do it well*

CRIANDO UMA APLICAÇÃO

```
$ django-admin startapp schedule
```

```
devcamp/  
  manage.py  
devcamp/  
  schedule/  
    __init__.py  
    models.py  
    tests.py  
    views.py
```

APLICAÇÕES - EXEMPLOS

- South - migrações de bancos de dados
- django-registration - registro de contas de usuários
- django-piston - aplicações REST
- django-debug-toolbar - barra de debug para desenvolvedores

QUERO SABER MAIS

- Documentação oficial
<https://www.djangoproject.com/>
- Listas de discussão
django-users
django-brasil
- IRC (Freenode)
#django-br
#django
- Vídeos da DjangoCon
<http://pyvideo.org/>

OBRIGADO!

@RENNEROCHA

<http://rennerocha.com/>
renne@rennerocha.com