

Django + Python



Régis da Silva
e
José Aciole

SENAC

5 de novembro de 2013



Introdução

- Python - linguagem de programação de alto nível, interpretada, orientada a objetos e de tipagem dinâmica e forte.
- Django - framework de desenvolvido web.

Um pouco de Python

```
lista = [1,-4,5,-3,-8,12]
def listaValores():
    soma = 0
    for i in lista:
        if i > 0:
            soma += i
        else:
            print i, 'eh negativo.'
    print 'Total', soma

listaValores()
```

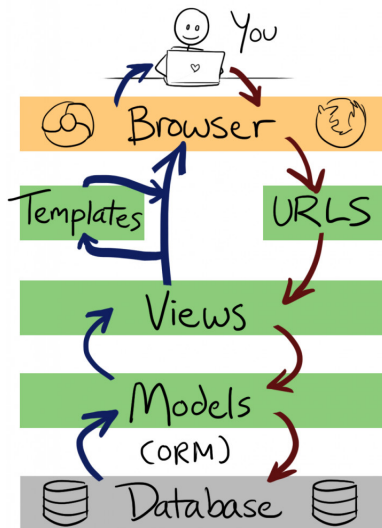
O que é Django

O *Django* é um framework web desenvolvido em Python.
Foi criado por *Adrian Holovaty* numa agência publicitária.
O nome Django foi inspirado no músico de Jazz *Django Reinhardt*.

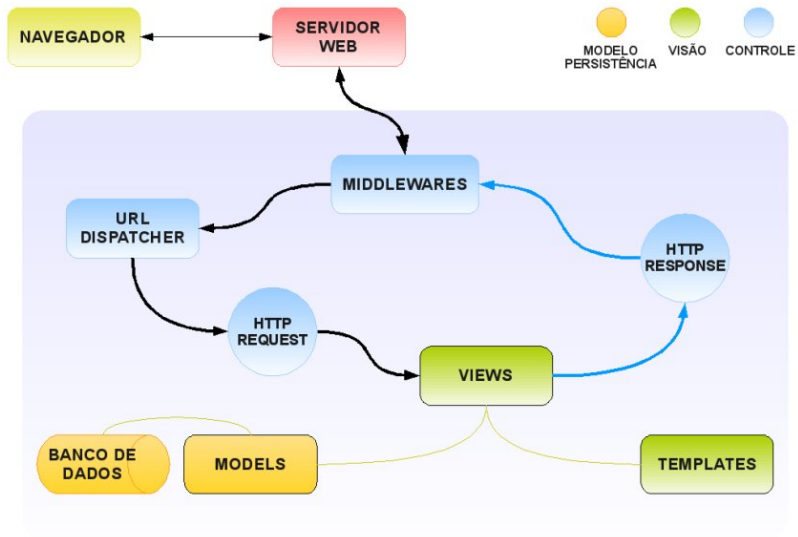
O que é Django

- adota o padrão MVC
- ênfase em reusabilidade e plugabilidade
- desenvolvimento ágil
- baseado no conceito DRY (*Don't Repeat Yourself*) “não se repita”
- mapeamento objeto-relacional ORM
- orientação à objeto: classes, atributos, métodos, herança, ...
- sistema de administração
- sistema de templates
- código aberto (*open source*)

MVC x MTV



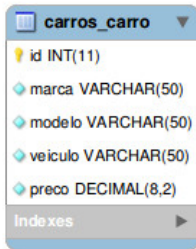
MVC x MTV



Models

- camada de abstração do banco de dados
- definem as entidades do sistema
- cada atributo de classe do modelo representa uma coluna do banco de dados
- é onde acontece o ORM

Criando um modelo



Criando um modelo

```
# -*- coding: utf-8 -*-

from django.db import models

class Carro(models.Model):
    marca = models.CharField('Marca', max_length=50)
    modelo = models.CharField('Modelo', max_length=50)
    veiculo = models.CharField('Veículo', max_length=50)
    preco = models.DecimalField('Preço', max_digits=8, decimal_places=2)

    class Meta:
        verbose_name = 'carro'
        verbose_name_plural = 'carros'

    def __unicode__(self):
        return self.veiculo
```

settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql', # Add 'postgresql_psycopg2' or 'sqlite3' if needed  
        'NAME': 'carros', # Or path to database file if using sqlite3:  
        # The following settings are not used with sqlite3:  
        'USER': 'root',  
        'PASSWORD': '123456',  
        'HOST': 'localhost', # Empty for localhost through named pipes - unix only  
        'PORT': '', # Set to empty string for named pipes only  
    }  
}
```

URL dispatcher

- utiliza URLs limpas e elegantes
- Django permite que você defina as URLs da maneira que quiser

urls.py

```
from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    url(r'^admin/', include(admin.site.urls)),
    url(r'^$', 'carros.views.index', name='home'),
)
```

Views

Uma *view* é simplesmente uma função *python* que recebe uma *requisição* e retorna uma *resposta* web.

Equivale ao *controller* de outros frameworks.

views.py

```
from django.shortcuts import render_to_response
from carros.models import Carro

def index(request):
    return render_to_response('carros/index.html', {
        'carros': Carro.objects.all().order_by('id')
    })
```

Templates

- O uso de *template* no *Django* proporciona facilidade e flexibilidade
- um *template* contém variáveis e *tags*, quando o *template* é avaliado essas variáveis são substituídas por valores
- Equivale ao *view* de outros frameworks.

Variáveis

Acessando o valor de uma variável

```
{{ variavel }}
```

Acessando atributos

```
{{ carro.marca }}
```

Tags

```
{% tag %}
```

- tags podem criar texto na saída
- podem aplicar *loops* ou lógica

Sintaxe mais completa

```
{% tag %}  
...  
{% endtag %}
```

Comentários

```
{# comentario #}
```

Exibindo dados

```
{% for current in carros %}  
<table>  
<tr>  
<td>{{current.marca}}</td>  
<td>{{current.modelo}}</td>  
<td>{{current.veiculo}}</td>  
<td>{{current.preco}}</td>  
</tr>  
</table>  
{% endfor %}
```

Exibindo dados



A screenshot of a web browser window. The address bar shows the URL 127.0.0.1:8000. The page title is "Carros". Below the title is a table with four columns: "Marca", "Modelo", "Veículo", and "Preço". The table contains six rows of data.

Marca	Modelo	Veículo	Preço
Audi	A3	Audi A3 Sport	94700,00
Chevrolet	Agile	Agile Flex câmbio automático	45990,00
Honda	Fit	Honda Fit Sport	28300,00
Fiat	500	Fiat 500 Flex	42840,00
Fiat	Palio	Palio Weekend	42540,00

Forms

- o *forms* é uma biblioteca do *django* para criação de formulários
- é possível gerar um formulário automaticamente através de um *model*
- possui regras de validação automática para cada tipo de campo

Programas e bibliotecas auxiliares

- virtualenv - para isolar os ambientes de cada projeto
- pip - para instalação de pacotes
- south - para migrações de modelos de dados
- git - versionamento
- fabric - deploy

Programas usados

SO	Linux (Ubuntu 13.10)
Linguagem	Python
Framework	Django
SGBD	MySQL
Front-End do SGBD	Workbench
Editor	Sublime Text 3
Compilador	Shell do sistema (terminal)
Versionamento	Git
Apresentação	LaTeX

Sites feitos com Django

<http://www.djangosites.org/tag/djangobrasil/>

<http://blog.glaucocustodio.com/2013/01/10/comecando-com-django-instalando-o-django-no-windows/>

<http://blog.glaucocustodio.com/2012/08/09/desenvolvimento-em-camadas-com-django/>

<http://www.djangobrasil.org/>

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

<http://osantana.me/2008/03/03/ambiente-isolado-para-python-com-virtualenv/>

<http://www.djangosites.org/>

http://imasters.com.br/artigo/10824/desenvolvimento/aplicacoes_rapidas_para_web_com_django/

Obrigado

<https://github.com/rg3915/django>