



Introdução ao Django

Luís Henrique Borges

@luishborges
contato@luishenrique.org

Maio de 2012

Quero que vocês saibam no final:

- O que é o Django
- Paradigmas e métodos do Django
- API (shell) do Django
- Como criar e configurar um projeto
- Como criar uma aplicação
- Como modelar sua aplicação
- Como mapear uma URL
- Como fazer um método view
- Como criar um template
- Como fazer um formulário
- Habilitar a interface Admin

Definições importantes

- **Framework**

- Abstração de códigos comuns
- Funcionalidades genéricas a serem especificadas e definidas para cada app
- Diferente de biblioteca (independência do código)

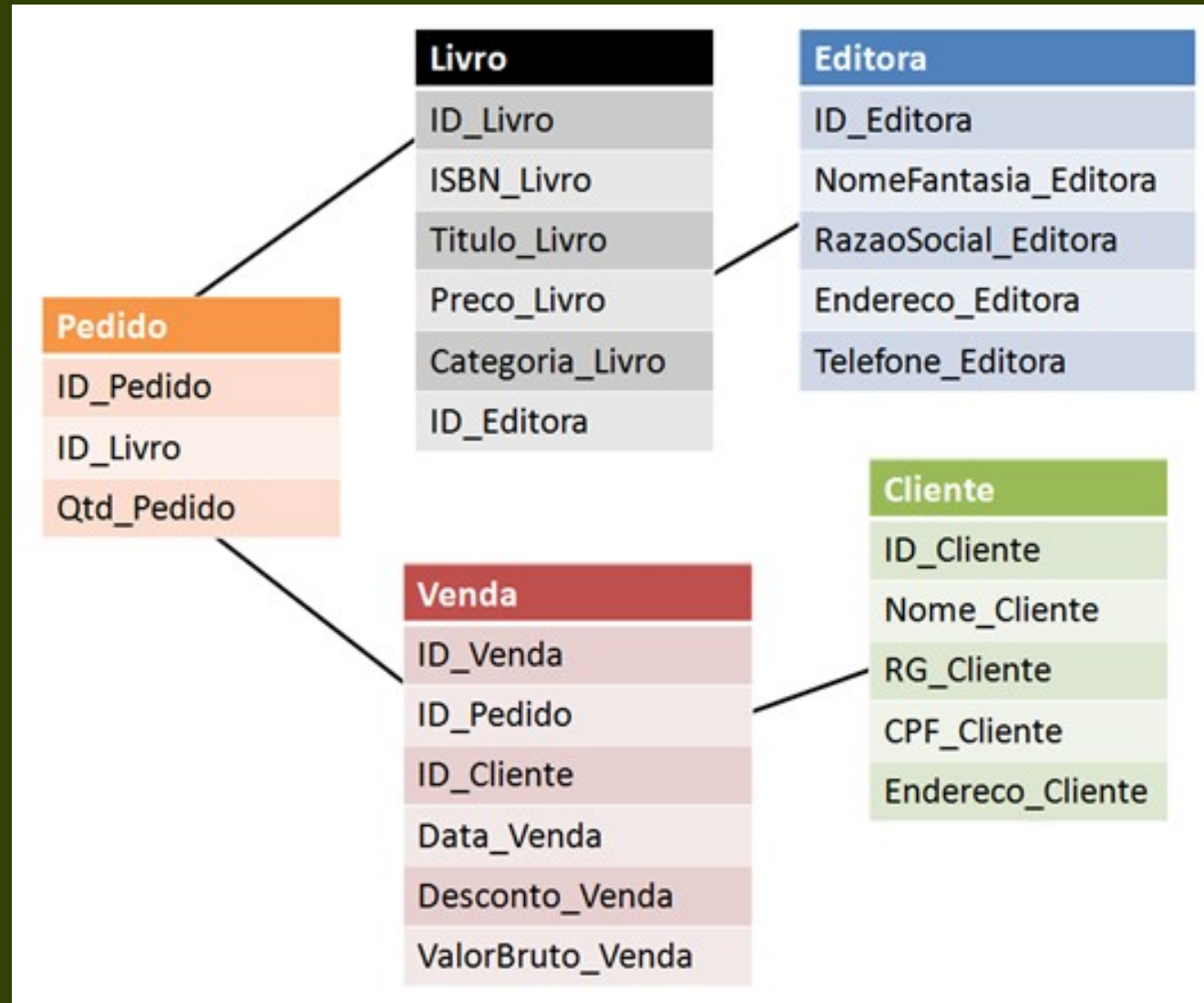
- **Banco de Dados**

- Coleções de informações
- Controlado e mantido por um SGBD (Sistema Gerenciador de Banco de Dados)
- Alguns SGBDs: PostgreSQL, MySQL, Oracle, Microsoft SQL Server, Fireboard e BD2

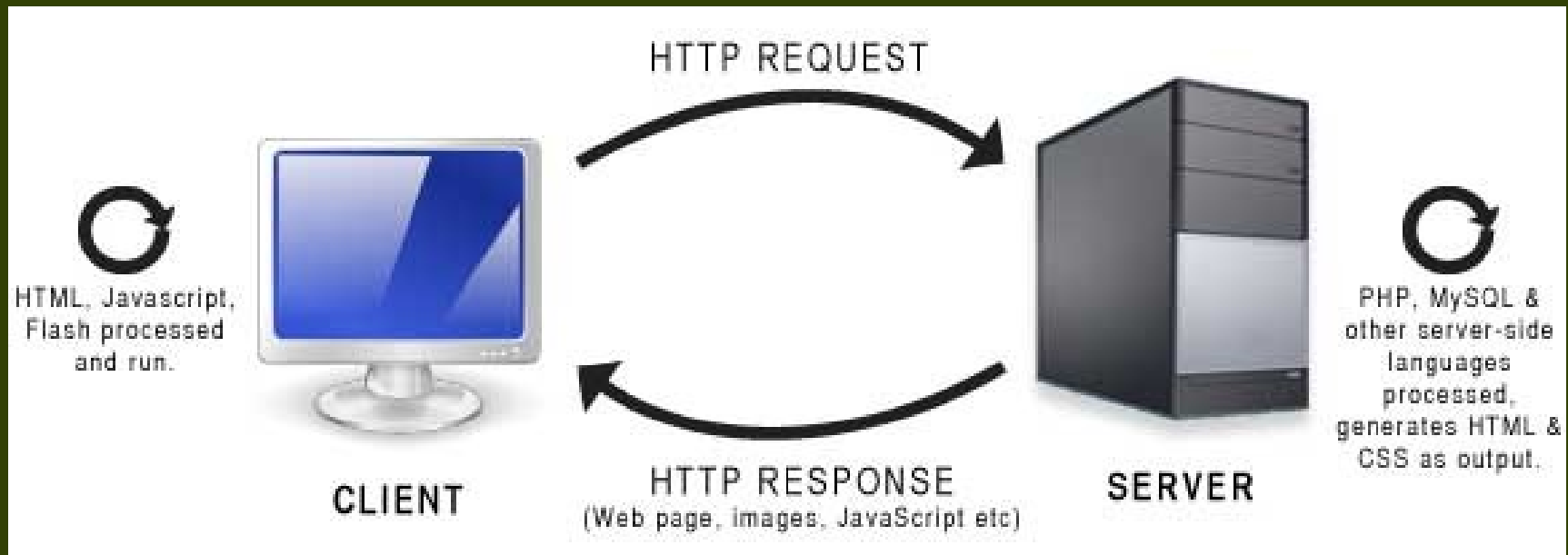
Definições

- **Server-side**
 - Linguagens de programação dinâmicas
 - Executam no servidor
 - Exemplos: PHP, ASP, Coldfusion, Java, .NET, Perl
- **Cliente-side**
 - Linguagens que executam no lado do cliente (browser)
 - Exemplos: Javascript, HTML, XHTML
- **CMS**
 - Sistema Gerenciador de Conteúdo
 - Exemplos: Joomla, Wordpress, Drupal, Plone.
- **Expressão Regular**
 - Identificação de cadeias de caracteres (padrões).

Banco de Dados Relacional



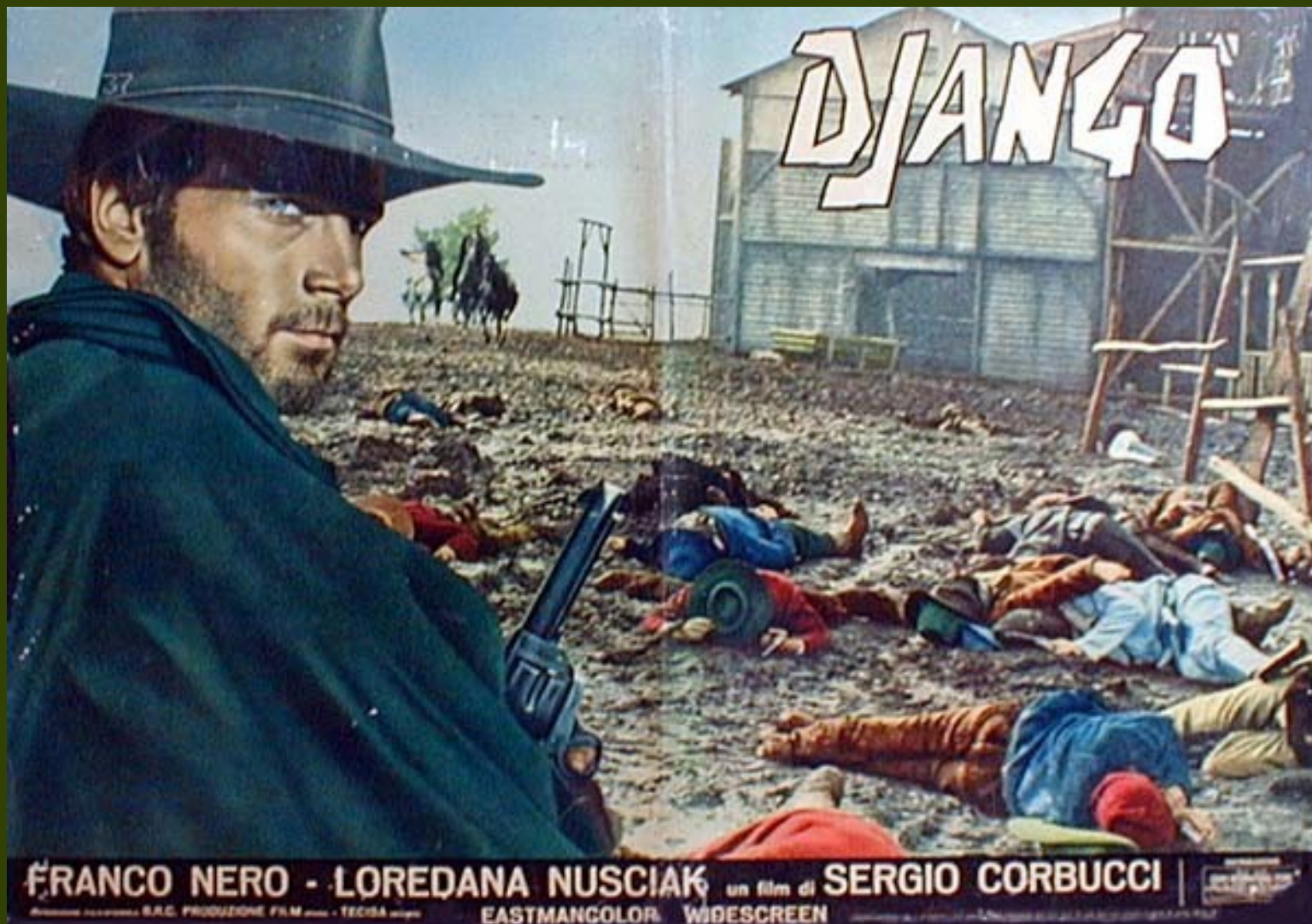
Modelo de Requisição



<http://demosthenes.info/assets/images/server-side.jpg>

Hoje nós vamos falar de Django

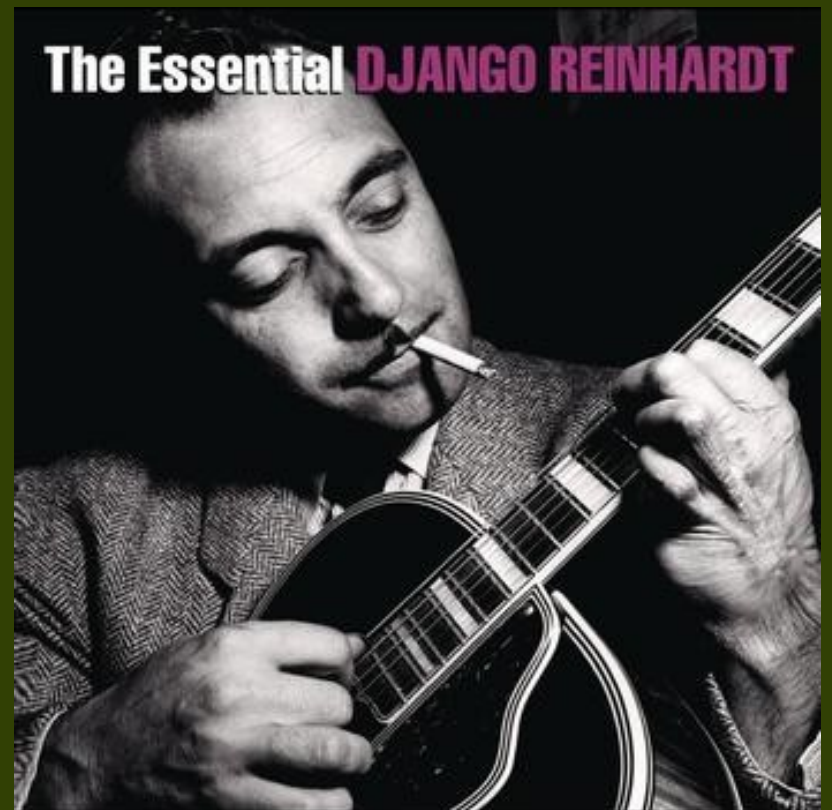
Não desse Django...



django

Origem

LAWRENCE
JOURNAL-WORLD®



O que é o Django

“Superframework” web

Escrito em Python

Voltado para o desenvolvimento ágil

Facilita tarefas simples (e chatas)

Quem usa Django

Globo.com
Nytimes.com
Guardian.co.uk
Nasa
addons.mozilla.org
Disqus
Instagram
Pinterest

www.djangosites.org

Por que usar Django

Produtividade

Aplicações eficientes

Código elegante

Metodologia ágil de desenvolvimento

Robustez e segurança

Django vs Rails

https://docs.google.com/View?docid=dcn8282p_1hg4sr9

Django no mercado



<http://www.indeed.com/jobtrends?q=rails%2C+django%2C+jsp&l=>

Considerações

- Não é um CMS
- Muitos conhecem Django antes do Python
- Django ajuda mas não é perfeito
- Não espere que a ferramenta faça tudo
- Testes não são perda de tempo
- Cuidado com a indentação
- Ainda não roda sobre sobre Python 3.x

Características

- MVC (MTV)
- ORM
- Interface Admin
- Templates
- Mapeamento de URLs
- Cache
- Internacionalização
- DRY, TDD
- Segurança, formulários, e-mail, feed, paginação, e muitos outros componentes.

Contextualizando: a legibilidade do código como uma vantagem de usar framework.

Quem nunca tentou entender um
código mal feito?

Pior ainda quando é seu próprio código...

VIDA DE PROGRAMADOR

.COM.BR

/* SUGESTÃO
ENVIADA POR GUSTAVO */



#179

PRECISAMOS COMPRAR UMA
FERRAMENTA PARA OFUSCAR
O CÓDIGO-FONTE



NÃO PRECISA...

COMO ASSIM,
NÃO PRECISA?



O ESTAGIÁRIO JÁ FAZ
CÓDIGO OFUSCADO
DIRETO...



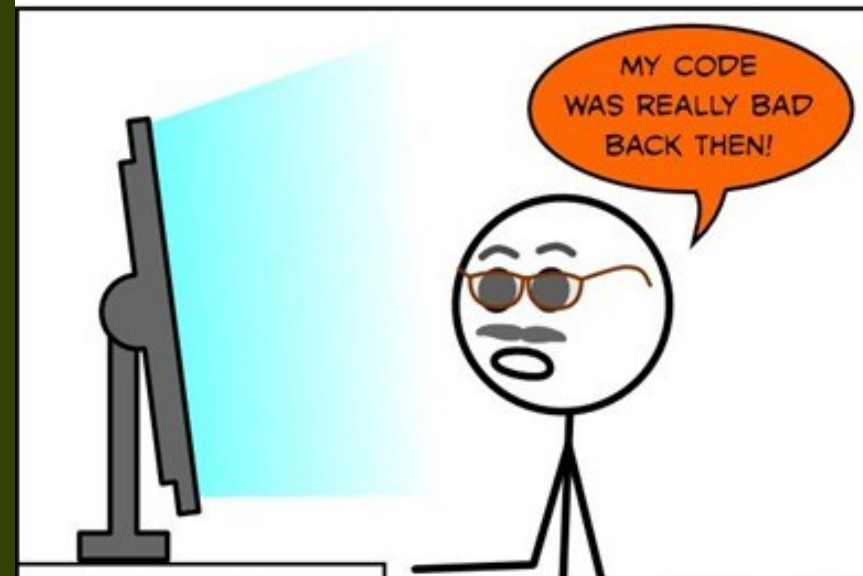
COMIC #44 - BAD CODE

MY CODE
WAS REALLY BAD
BACK THEN!



TIME PASSES...

MY CODE
WAS REALLY BAD
BACK THEN!



[HTTP://TWITCH.COM/44/](http://TWITCH.COM/44/)

Twitch by Eric Burke

```

<!-- InstanceBeginEditable name="EditRegion1" -->
<td width="62%" valign="top"><div align="center"><font size="5" face="Verdana, Arial, Helvetica, sans-serif">Dog
Clothes<strong> </strong></font> <br>
<br>
</div>
</td>
<table width="109%" height="908" border="0" align="center" cellpadding="0" cellspacing="0">

<td valign="top" height="113" width="191" bgcolor="#FFFFFF">
<div align="center"><a href="http://www.pawprintzpetboutique.com/dog-clothes-shirts-sleeves-personality.htm"></a></div></td>
<td width="191" height="113" valign="top" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><a href="dog-clothes-sweaters.htm"></a></div></td>
<td width="191" height="113" valign="top" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><a href="http://www.pawprintzpetboutique.com/dog-clothes-shirts.htm"></a></div></td>
</tr>
<td valign="top" height="60" width="191" bgcolor="#FFFFFF"> <div align="center"><b><font size="2" face="Geneva, Arial, Helvetica,
san-serif"></font><font color="ffcc66" size="2" face="Geneva, Arial, Helvetica, san-serif"><b><a
href="http://www.pawprintzpetboutique.com/dog-clothes-shirts-sleeves-personality.htm">Personality T's</a></b></font><font color="FFCC66"
face="Geneva, Arial, Helvetica, san-serif" size="2"></font></b></div></td>
<td width="191" height="60" valign="top" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><b><font color="ffcc66" face="Geneva, Arial, Helvetica, san-serif" size="2"><a
href="http://www.pawprintzpetboutique.com/dog-clothes-sweaters.htm">Sweaters</a></font></b></div></td>
<td width="191" height="60" valign="top" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><b><font size="2" face="Geneva, Arial, Helvetica, san-serif"><a
href="http://www.pawprintzpetboutique.com/dog-clothes-shirts.htm">
Shirts</a></font></b></div></td>
</tr>
<td valign="top" height="120" width="191" bgcolor="#FFFFFF"> <div align="center"><a
href="http://www.pawprintzpetboutique.com/dog-clothes-sweat.htm"></a></div></td>
<td valign="top" height="120" width="191" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><a href="http://www.pawprintzpetboutique.com/dog-clothes-pajamas.htm"></a></div></td>
<td valign="top" height="120" width="191" bordercolor="#7C2953" bgcolor="#FFFFFF">
<div align="center"><a href="http://www.pawprintzpetboutique.com/dog-clothes-coats.htm"></a></div></td>
</tr>
<td valign="top" height="60" width="191" bgcolor="#FFFFFF"> <div align="center"><font color="ffcc66" size="2" fac
Helvetica, san-serif"><b><a href="http://www.pawprintzpetboutique.com/dog-clothes-sweat.htm">Sweatshirts

```

<http://www.circuit-riders.net/articles-learn-html.php>

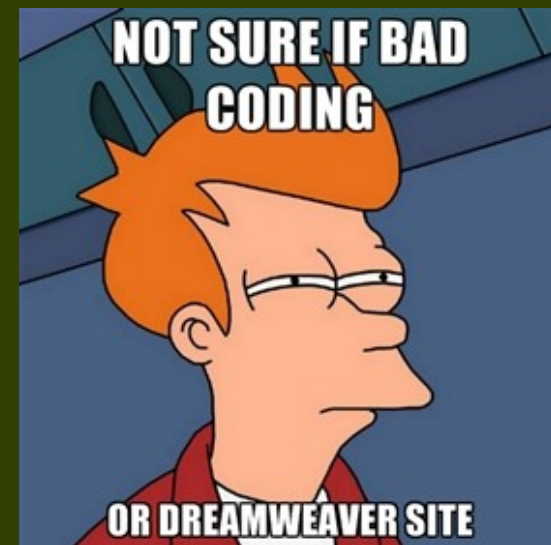
Ferramentas que sujam e lazy programmers

```

private string shortCity(string city)
{
    if (city == "Stockholm")
    {
        return "&nbsp;S";
    }
    if (city == "G\x00f6teborg")
    {
        return "&nbsp;G";
    }
    if (city == "Ume\x00e5")
    {
        return "&nbsp;U";
    }
}

```

<http://blog.dileno.com>



<http://www.testically.org/tags/clean-code-developer/>

No Django, *geralmente*, o código é mais limpo e legível.


```

{% extends "foobar/baz.html" %}

{% load news %}

{% block title %}{{ title }}{% endblock %}

{% comment %}foo{% endcomment %}

{% block content %}

    {{ category.name }}

    <h1>Articles</h1>

{% for article in articles %}
    <div class="article">
        <h2>{{ article.title }}</h2>

        <div class="content">
            {{ article.content|markdown }}
            -written by {{ author }}
        </div>
    </div>
{% endfor %}
{% endblock %}

```

```

2 from django.shortcuts import get_object_or_404
3 from django.contrib.auth.models import User
4 from django.template import RequestContext
5 from django.utils import simplejson
6 from django import forms
7 from potluck.uploadify import views, models
8 from potluck.status_updates.models import StatusUpdate
9 from potluck.status_updates.forms import NewStatusUpdateForm
10
11 def status_update_detail(request, object_id, y
12     status_update = StatusUpdate.objects.get(id=object_id)
13     return render_to_response('status_updates/status_update_detail.html',
14                               {'status_update': status_update,
15                               'request': request})
15
16 def uploaded_received_handler(sender, data, **kwargs):
17     if file:
18         new_file_upload = UploadedFile.objects.create(
19             file=file,

```

Características do Django

Paradigma MVC

- Model – View – Controller
- Modularização do código
- Separação entre regras de negócio e interface com usuário.

MVC

PHP sem MVC

```
<?php endif ?>

<?php
$usuario = query(
    "SELECT u.*, e.* FROM usuarios u
    INNER JOIN enderecos_usuarios e ON u.idUsuario = e.idUsuario
    WHERE u.idUsuario = " . (int)$_SESSION['idUsuario'];
$usuario = $usuario->fetch_assoc();
?>

<div class="row">

<?php if (usuarioLogado()){ ?>
```

CakePHP (MVC)

```
354  /**
355   * view method
356   *
357   * @param string $id
358   * @return void
359   */
360  public function admin_view($id = null) {
361      $this->User->id = $id;
362      if (!$this->User->exists()) {
363          throw new NotFoundException(__('Invalid user'));
364      }
365      $this->set('user', $this->User->read(null, $id));
366  }
```

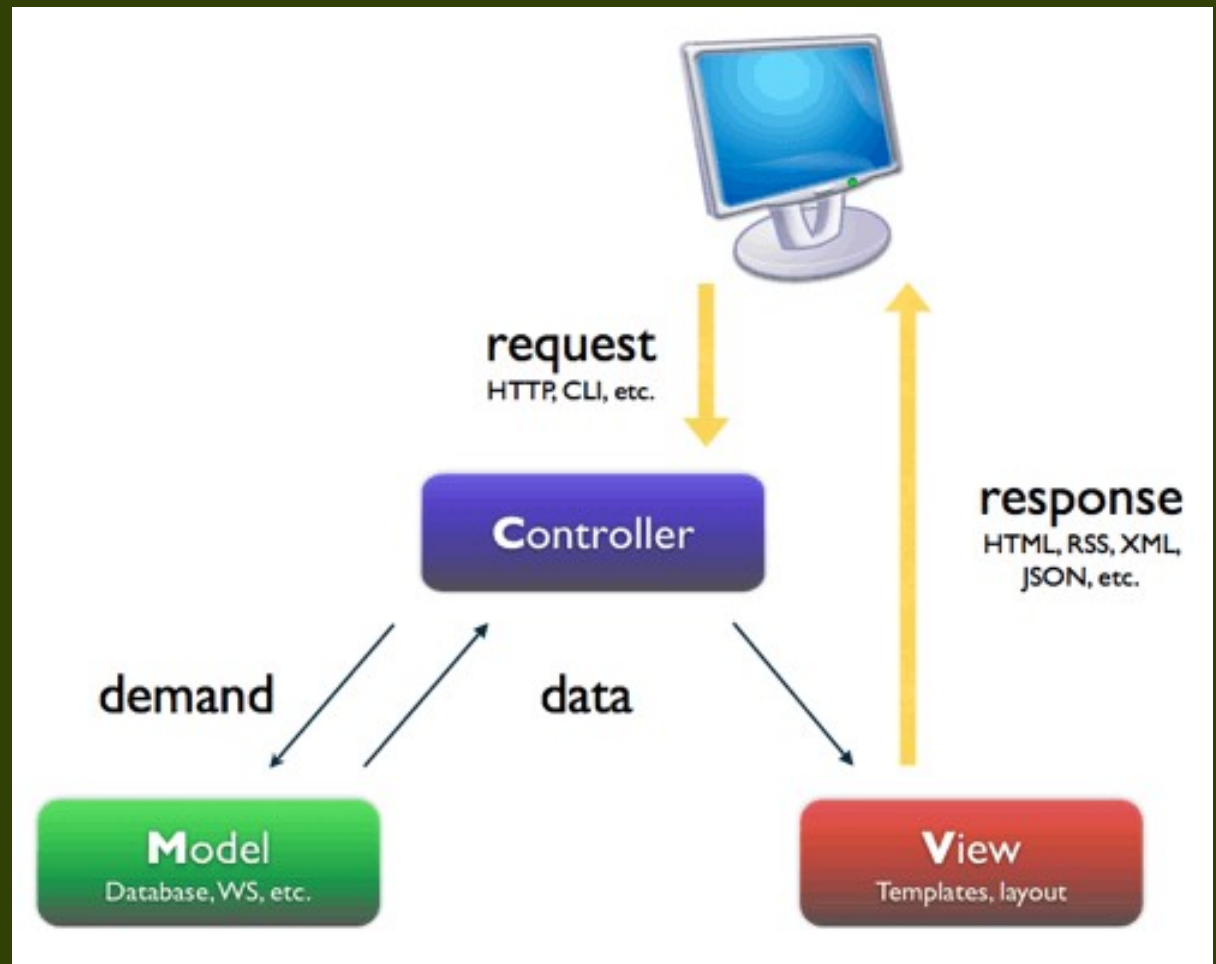
Django MVC = MTV



Model
View
Controller

No Django:

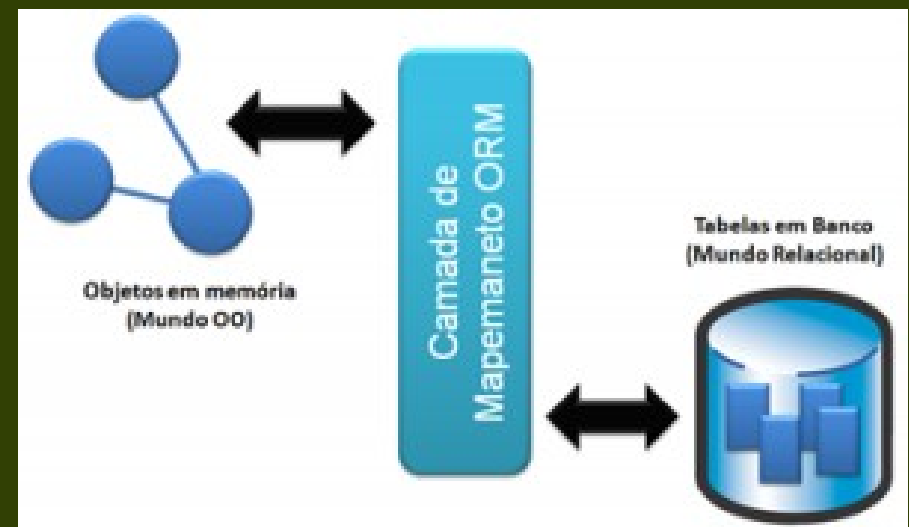
Model
Template (view)
View (controller)



<http://www.fernandovalente.com.br/wordpress/2011/01/11/mvc-model-view-controller/>

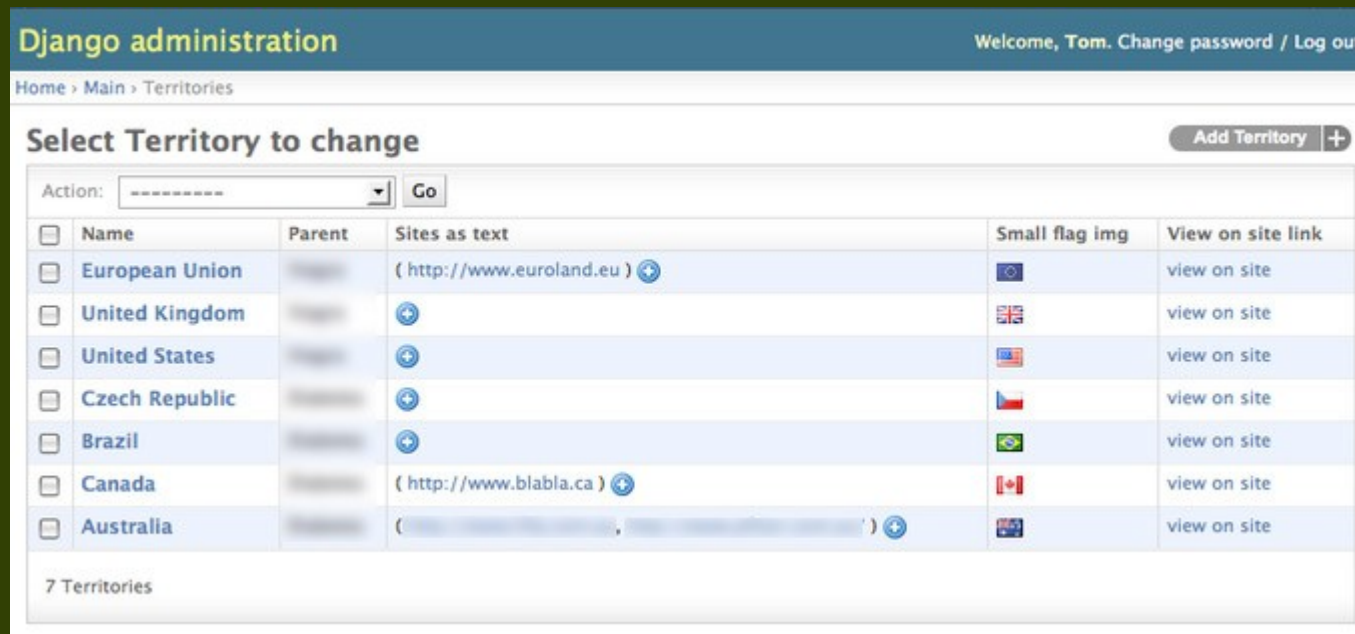
ORM

- Modelo Objeto Relacional
- Para Bancos de Dados relacionais
- Faz o trabalho de persistência
- Abstração do SQL
- Mundo Orientado a Objeto x Relacional
- SQLAlchemy, Hibernate, Doctrine



Django admin

- Parte poderosa do framework
- Interface pronta para produção
- Baseada nos metadados da sua app



DRY



Filme "Tempos Modernos" (Charles Chaplin)

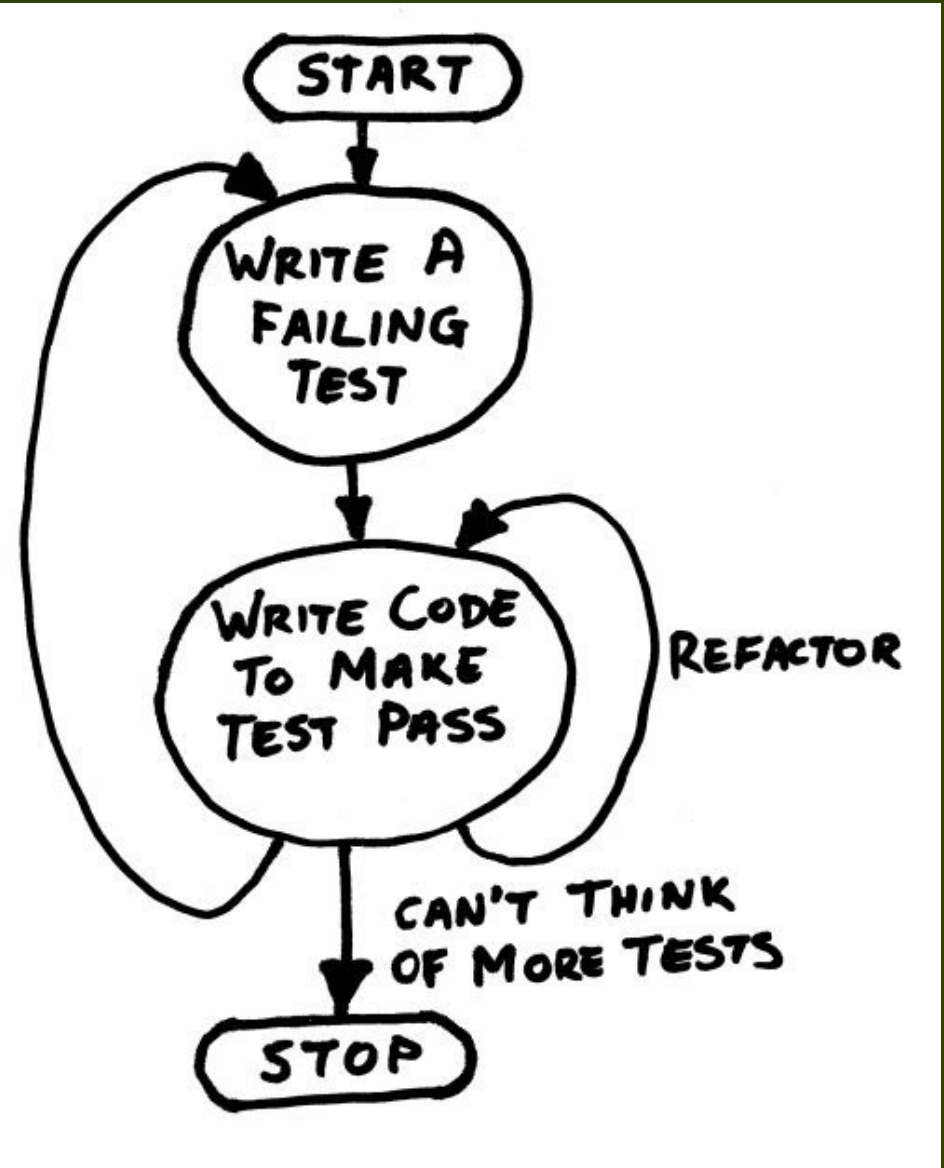
CTRL + C
CTRL + V

?

Don't Repeat Yourself

TDD

Desenvolvimento Orientado a Testes



Templates

É um documento de texto ou uma string, criada com a linguagem de templates do Django. Pode conter blocos.

```
<ul>
{% for item in mylist %}
    <li>{{ item }}</li>
{% endfor %}
</ul>
```

```
Meu nome é {{ first_name }}.
Nasci no dia {{ birthday|date:'d/m/Y' }}.
```

Templates - Herança

Criamos um template padrão, `base.html`.

Deixaremos blocos que terão conteúdos diferentes nas páginas.

Herança:

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

Projetos e Aplicações

Projeto: uma instância do Django. Pode conter várias aplicações.

Aplicação: qualquer trecho de código que possa ser reutilizado. Pode fazer parte de vários projetos.

Mão na massa!

Criando um projeto

Entre na pasta que você vai salvar seu projeto e execute o comando:

```
$ cd minicurso-django (sua pasta)
```

```
$ django-admin.py startproject gerenciador
```


Criando um projeto

Arquivos criados:

```
$ cd gerenciador
```

```
$ ls  
gerenciador  manage.py
```

```
$ ls gerenciador  
__init__.py  settings.py  urls.py  wsgi.py
```

Arquivos do projeto

`__init__.py`: arquivo vazio, indica que o projeto pode ser como um pacote.

`manage.py`: utilitário para auxiliar nas tarefas específicas do projeto.

`settings.py`: arquivo de configuração do seu projeto. Conexão com banco de dados, envio de e-mails, idiomas, aplicações, etc.

`urls.py`: definições de rotas e URLs do projeto.

Testando

Welcome to the Django

```
$ python manage.py runserver
```

Validating models...

0 errors found

Django version 1.4, using settings 'gerenciador.settings'

Development server is running at <http://127.0.0.1:8000/>

Quit the server with CONTROL-C.

It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASE_*` settings in `mysite/settings.py`.
- Start your first app by running `python mysite/manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured `STATICFILES_FINDERS`.

Configure seu projeto

```
$ cd gerenciador
```

```
$ gedit settings.py (use seu editor favorito)
```

Configure seu projeto - BD

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': 'bancodedados.db',  
        'USER': '',  
        'PASSWORD': '',  
        'HOST': '',  
        'PORT': '',  
    }  
}
```

Gerenciando seu projeto

```
$ python manage.py help
```

Algumas opções disponíveis:

`syncdb`: cria tabelas e índices relacionados aos models de todas apps.

`shell`: prompt interativo com ambiente do projeto e suas configurações carregadas.

`runserver`: executa o servidor de desenvolvimento. Padrão na porta 8000.

Criando uma aplicação

```
$ django-admin.py startapp agenda
```

```
$ ls agenda
```

```
__init__.py  models.py  tests.py  views.py
```

Arquivos da aplicação

`__init__.py`: diretório é um pacote Python.

`models.py`: definição dos elementos Model da nossa aplicação.

`views.py`: programação das regras de negócio, as funcionalidades da aplicação. Camada controller do MVC.

`tests.py`: definições de testes unitários da nossa aplicação.

Configure seu projeto - APP

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'agenda',  
    # 'django.contrib.admin',  
    # 'django.contrib.admindocs',  
)
```

Hello World

```
$ gedit gerenciador/urls.py
```

```
urlpatterns = patterns('',  
    ...  
    (r'^$', 'agenda.views.index'),  
)
```

Hello World

```
$ gedit agenda/views.py
```

```
from django.http import HttpResponse
```

```
def index(request):  
    return HttpResponse("Hello World!")
```

<http://127.0.0.1:8000/>

Rodar runserver novamente ou
atualizar a página caso tenha
deixado o terminal executando.

0 que foi feito até agora

- Criação de um projeto
- Criação de uma aplicação
- Configurações no arquivo `settings.py` na lista `INSTALLED_APPS`
- Apontamos uma URL para uma função `view`
- Implementamos uma função `view` para teste
- Testamos no navegador

Gerenciando os dados

```
$ gedit agenda/models.py
```

```
...
```

```
class ItemAgenda(models.Model):  
    data = models.DateField()  
    hora = models.TimeField()  
    titulo = models.CharField(max_length=90)  
    descricao = models.TextField()  
    local = models.CharField(max_length=80)
```

Crie o Banco de Dados

```
$ python manage.py syncdb
```

```
Creating tables ...  
Creating table auth_permission  
Creating table auth_group_permissions  
Creating table auth_group  
Creating table auth_user_user_permissions  
Creating table auth_user_groups  
Creating table auth_user  
Creating table django_content_type  
Creating table django_session  
Creating table django_site  
Creating table agenda_itemagenda
```

You just installed Django's auth system, which means you don't have any superusers defined.

Would you like to create one now? (yes/no): yes

Username (leave blank to use 'lh'):

E-mail address: luish@comp.ufla.br

Password:

Password (again):

Superuser created successfully.

Installing custom SQL ...

Installing indexes ...

Installed 0 object(s) from 0 fixture(s)

Aproveite a API

```
$ ./manage.py shell
```

```
>>> from agenda.models import ItemAgenda
>>> ItemAgenda.objects.all()
[]
```

```
>>> i = ItemAgenda(data='2012-05-12', hora='08:00:00', titulo='Minicurso Django', descricao='Minicurso integrante do evento CompWeekend', local='UFLA - DCC04')
>>> i.save()
```

```
>>> ItemAgenda.objects.all()
[<ItemAgenda: ItemAgenda object>]
```

Aproveite a API

```
$ gedit agenda/models.py
```

```
class ItemAgenda(...):
```

```
...
```

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


Aproveite a API

```
$ python manage.py shell
```

```
>>> from agenda.models import ItemAgenda
```

```
>>> ItemAgenda.objects.all()
```

```
[<ItemAgenda: Minicurso Django>]
```

Criando o template base

```
$ gedit templates/base.html
```

```
<html>
  <head>
    {% block cabecalho %}
    <title>Agenda</title>
    {% endblock %}
  </head>
  <body>
    {% block conteudo %}
    {% endblock %}
  </body>
</html>
```

Criando a página inicial:
Lista de tarefas

Página inicial: template

```
$ gedit templates/index.html
```

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

```
{% block conteudo %}
```

```
<a href="/novo">Adicionar novo item</a>
```

```
<ul>
```

```
{% for item in itens %}
```

```
    <li>{{ item.data }} - {{ item.titulo }}</li>
```

```
{% empty %}
```

```
    <li>Sem itens</li>
```

```
{% endfor %}
```

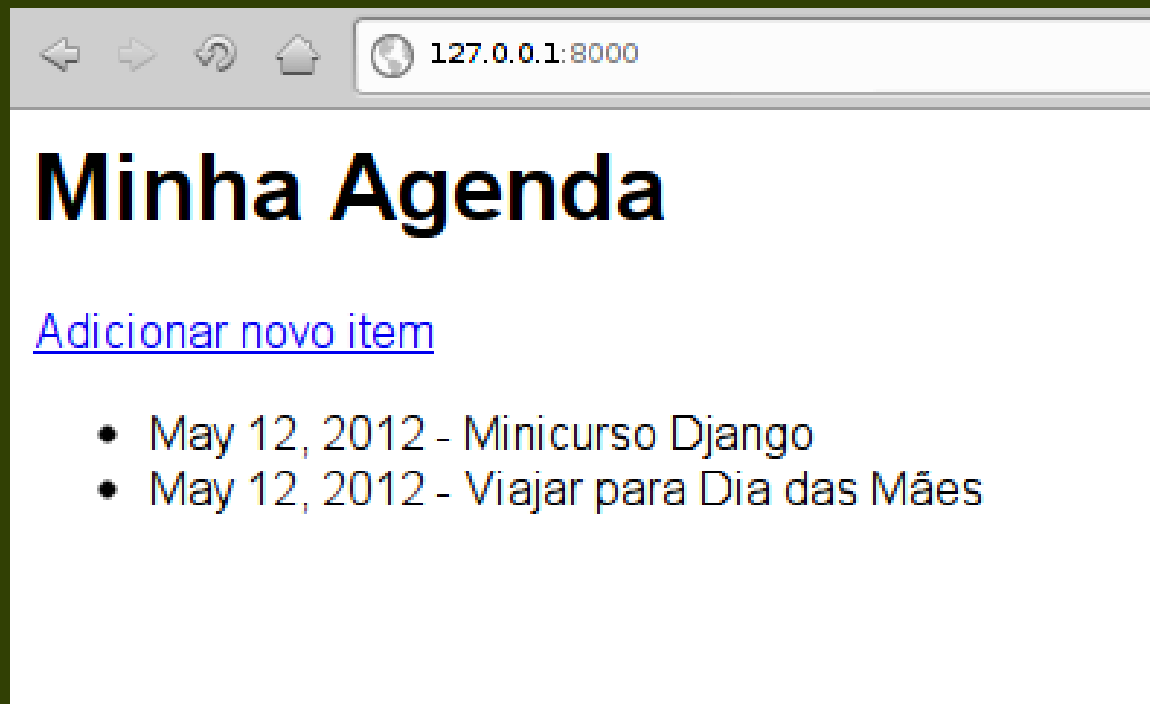
```
</ul>
```

```
{% endblock %}
```

Página inicial: view index

```
from django.shortcuts import render_to_response
from agenda.models import ItemAgenda

def index(request):
    itens = ItemAgenda.objects.all()
    return render_to_response('index.html',
{'itens': itens})
```



Interface Admin

Usando a interface admin #1

```
$ gedit gerenciador/urls.py
```

Descomentar linhas (retirar '# '):

```
from django.contrib import admin  
admin.autodiscover()
```

```
...
```

```
url(r'^admin/', include(admin.site.urls)),
```


Usando a interface admin #2

```
$ gedit gerenciador/settings.py
```

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.admin',  
    'agenda',  
)
```

Usando a interface admin #3

```
$ python manage.py syncdb
```

```
Creating tables ...
```

```
Creating table django_admin_log
```

```
Installing custom SQL ...
```

```
Installing indexes ...
```

```
Installed 0 object(s) from 0 fixture(s)
```

Usando a interface admin #4

```
$ gedit agenda/admin.py (criar arquivo)
```

```
from django.contrib import admin  
from agenda.models import ItemAgenda  
  
admin.site.register(ItemAgenda)
```

Usando a interface admin

<http://127.0.0.1:8000/admin>

Rodar runserver novamente

Criando um formulário para o usuário
inserir itens na agenda

Forms

```
$ gedit agenda/forms.py
```

```
from django import forms
```

```
class FormItemAgenda(forms.Form):  
    titulo = forms.CharField(max_length=80)  
    data = forms.DateField(  
        widget = forms.DateInput(format='%d/  
%m/%Y'),  
        input_formats = ['%d/%m/%y', '%d/%m/  
%Y'])  
    hora = forms.TimeField()  
    descricao = forms.CharField()  
    local = forms.CharField()
```

Função na view (controller)

```
$ gedit agenda/views.py
```

```
def novo(request):  
    if request.method == 'POST':  
        form = FormItemAgenda(request.POST, request.FILES)  
  
        if form.is_valid():  
            dados = form.cleaned_data  
            item = ItemAgenda(  
                data=dados['data'],  
                hora = dados['hora'],  
                titulo = dados['titulo'],  
                descricao = dados['descricao'],  
                local = dados['local'])  
            item.save()  
            return render_to_response('salvo.html', {})  
        else:  
            form = FormItemAgenda()  
  
    return render_to_response('novo.html', {'form': form},  
context_instance=RequestContext(request))
```

View

```
$ gedit templates/novo.html
```

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

```
{% block conteudo %}
```

```
<h2>Adicionar item na agenda</h2>
```

```
<form action="" method="post">
```

```
    {% csrf_token %}
```

```
    {{ form.as_p }}
```

```
    <input type="submit" value="Adicionar"
```

```
class="btn" />
```

```
</form>
```

```
{% endblock %}
```

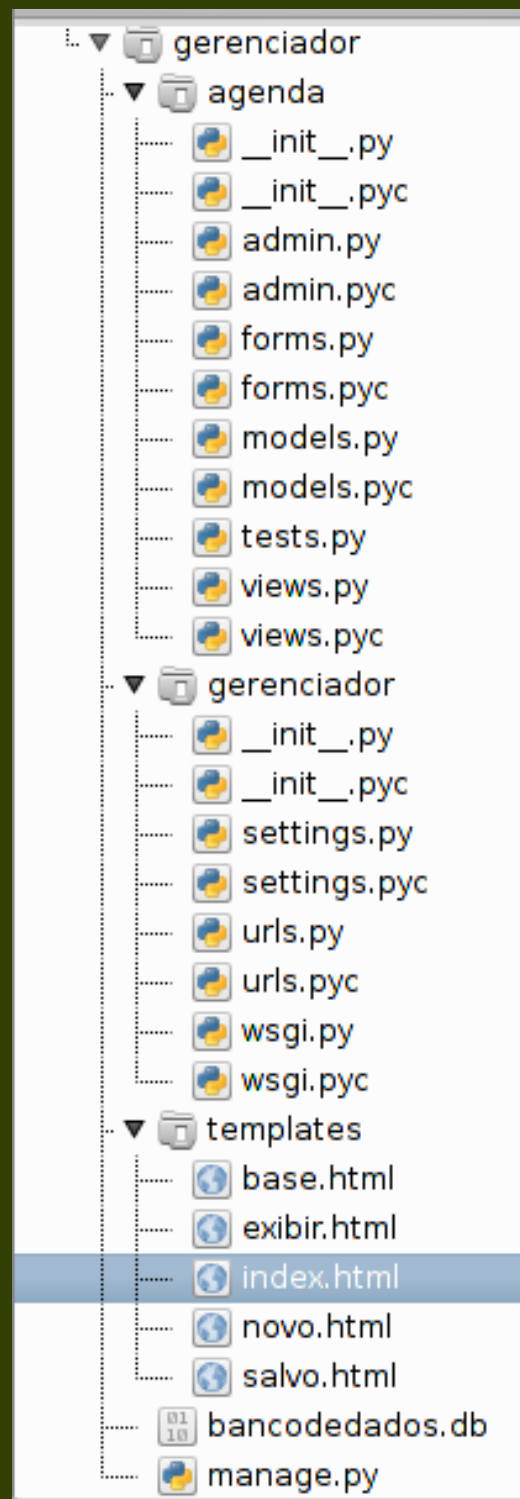

URL

```
$ gedit gerenciador/urls.py
```

```
...
```

```
    (r'^$', 'agenda.views.index'),  
    (r'^novo/$', 'agenda.views.novo'),
```

Estrutura de diretório até agora



Vamos agora fazer a funcionalidade de visualizar um item da agenda.

Visualizar item #1: URL

```
(r'^exibir/(?P<num_item>\d+)/$',  
'agenda.views.exibir'),
```

Exemplo:

/exibir/2

Visualizar item #2: view

```
from django.shortcuts import get_object_or_404
```

```
def exhibir(request, num_item):  
    item = get_object_or_404(ItemAgenda,  
pk=num_item)  
    return render_to_response('exibir.html',  
{ 'item': item })
```

Visualizar item #3: template

`templates/exibir.html`

(No próximo slide)

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

```
{% block conteudo %}
```

```
<h2>{{ item.titulo }}</h2>
```

```
<p>{{ item.descricao }}</p>
```

```
<ul>
```

```
    <li><strong>Data: </strong>{{ item.data |  
date:'d/m/Y' }} - {{ item.horalttime:'h:i' }}</li>
```

```
    <li><strong>Local: </strong>{{ item.local }}</li>  
</ul>
```

```
<p>
```

```
    <a href="/">Voltar</a>
```

```
    <a href="/remover/{{ item.id }}">Remover</a>
```

```
</p>
```

```
{% endblock %}
```


Baseado no **exibir**, criem agora a funcionalidade de **excluir** um item da agenda.

Dica: o método para exclusão de um item é o `delete()`.

Resultado final da aplicação

Minha Agenda

Versão 0.1

- 12/05/2012 às 08:00 - [Minicurso Django](#)
- 15/05/2012 às 09:00 - [Reunião com chefe](#)

Total de 2 itens

[Adicionar novo item](#)

Minha Agenda

Versão 0.1

Adicionar item na agenda

Título:

Data:

Hora:

Descricao:

Local:

[Adicionar](#)

Minha Agenda

Versão 0.1

Minicurso Django

Minicurso integrante do evento CompWeekend

- **Data:** 12/05/2012 - 08:00
- **Local:** UFLA - DCC04

[Voltar](#)

[Remover](#)

Bônus: Ordenação

ASC

```
ItemAgenda.objects.all().order_by('date')
```

DESC

```
ItemAgenda.objects.all().order_by('-date')
```

Bônus: Limite

```
# LIMIT 5
```

```
ItemAgenda.objects.all()[5]
```

```
# OFFSET 5 LIMIT 5
```

```
ItemAgenda.objects.all()[5:10]
```

Dúvidas?

Obrigado. :)

Referências



<https://docs.djangoproject.com/>
<http://docs.djangobrasil.org/>
<http://djbook.ru/>