



# FHO

FUNDAÇÃO HERMÍNIO OMETTO

# PÓS-GRADUAÇÃO 2020

- ✓ Aperfeiçoamento
- ✓ MBA
- ✓ Especialização
- ✓ Mestrado

contatopos@fho.edu.br

 (19) 9 9872-0550

Inscrições pelo site  
[www.fho.edu.br](http://www.fho.edu.br)

SIGA NOSSAS REDES SOCIAIS



FHO - Fundação  
Hermínio Ometto



fho\_oficial



FHO - Fundação  
Hermínio Ometto

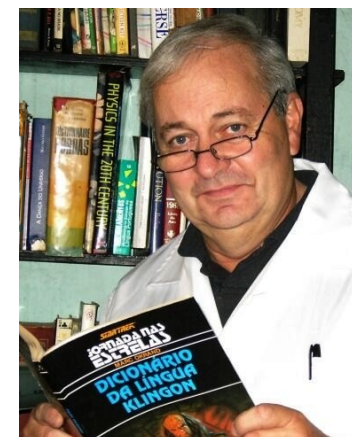


# **DESENVOLVIMENTO WEB COM FRAMEWORK DJANGO**

**ORLANDO SARAIVA JR**

## Citação

“ Na aula você não aprende...  
Na aula você entende! ”



# Roteiro

Quem sou eu ?

O que é um framework ?

O framework web Django

Hands-on

# Orlando Saraiva Jr

Mestre em Tecnologia  
FT - Unicamp - 2013

MBA em Gestão Estratégica de Negócios  
Unifian - 2008

Graduação em Tecnologia em Informática  
CESET - 2005





# Orlando Saraiva Jr

Assistente de Suporte acadêmico  
Unesp Rio Claro desde 2007

Docente na FHO  
desde 2009

Docente na FATEC Araras  
desde 2018



## Framework

Um framework é um conjunto de classes cooperantes que constroem um projeto reutilizável para uma específica classe de software. [Deu89, JF88]

Você customiza um framework, para uma aplicação específica, através da criação de subclasses específicas para a aplicação, sendo essas subclasses específicas das classes abstratas do framework.

Os frameworks sempre tem um particular domínio de aplicação.

## Framework

Um framework é um **conjunto de classes** cooperantes que constroem um projeto reutilizável para uma específica classe de software. [Deu89, JF88]

Você customiza um framework, para uma aplicação específica, através da criação de subclasses específicas para a aplicação, sendo essas subclasses específicas das classes abstratas do framework.

Os frameworks sempre tem um particular **domínio de aplicação**.



## Framework para Web



Pyramid™



Falcon

Bottle



GRAILS



Flask

web development,  
one drop at a time

LARAVEL



RAILS



spring

ZEND



YII FRAMEWORK



PHP  
FRAMEWORKS

django

CODEIGNITER



CAKEPHP



SYMFONY



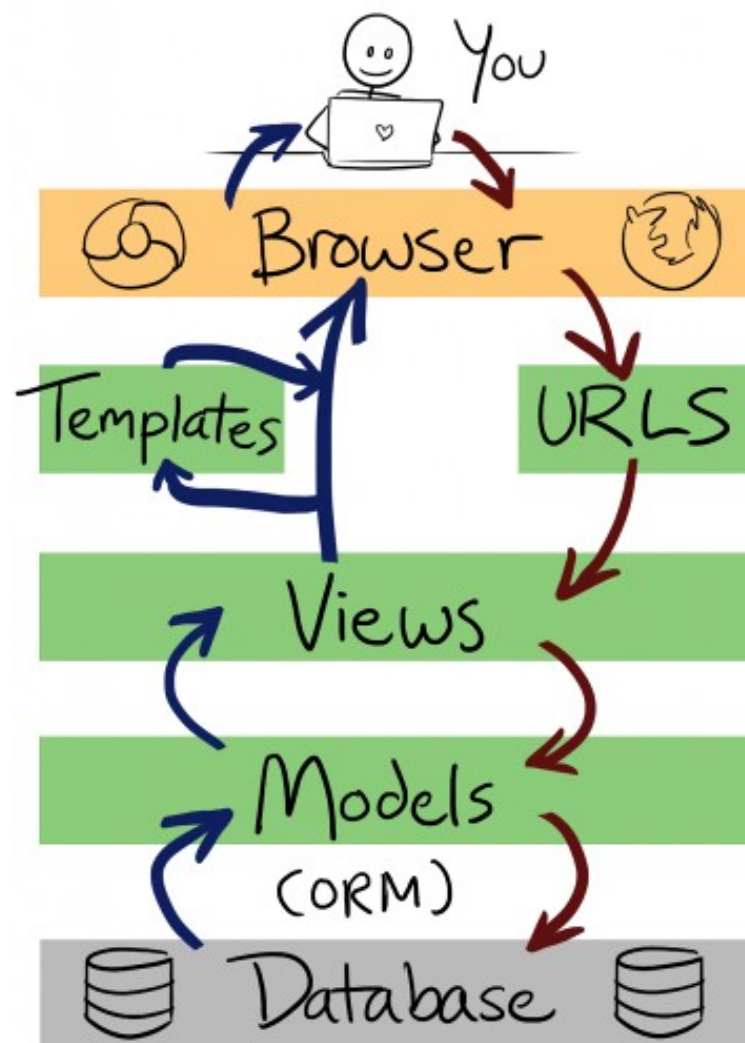
Pylons

# Django

Como o Django foi desenvolvido no ambiente movimentado de uma redação, ele foi desenhado para tornar as tarefas comuns do desenvolvimento Web rápidas e fáceis.

# django

# Django Fluxo MTV



## Citação

Talk is cheap.  
Show me the code.



## Feriado v. 0.1

```
(venv) pip install django==2.2.10
```

```
(venv) django-admin startproject feriados
```

```
__init.py__
```

```
manage.py
```

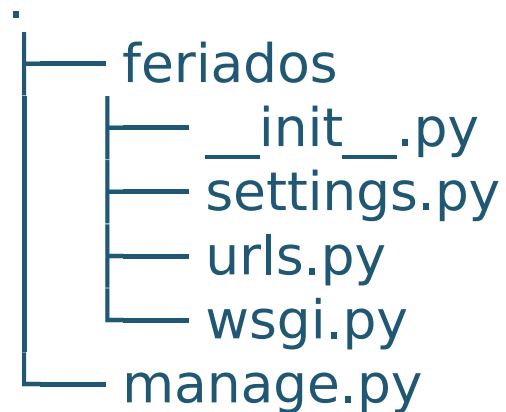
```
settings.py
```

```
urls.py
```

```
wsgi.py
```

```
nome_do_projeto/
```

## Feriado v. 0.1



(venv) cd feriados

(venv) python manage.py startapp feriado



# Feriado v. 0.1

## feriados/urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('feriado.urls')),
]
```

# Feriado v. 0.1

## feriados/feriado/views.py

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

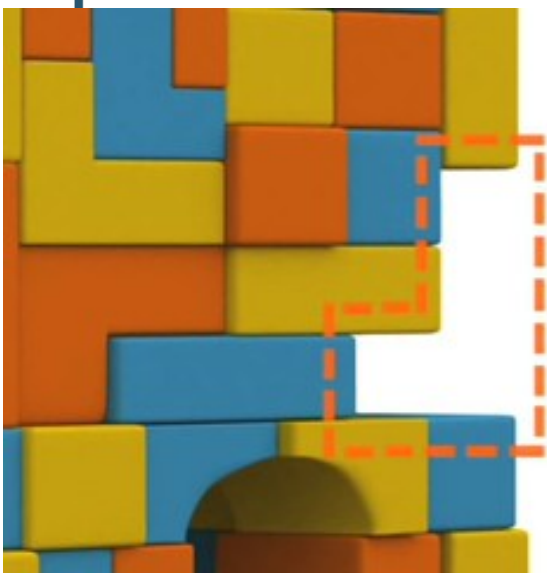
```
def natal(request):  
    return HttpResponse("Não é natal.")
```

# Feriado v. 0.1

## feriado/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.natal),
]
```

## E os testes ?



# Feriado v. 0.1

## /feriado/tests.py

```
from django.test import TestCase
```

```
class NatalTest(TestCase):
```

```
    def setUp(self):
```

```
        self.resp = self.client.get('/')
```

```
    def test_200_response(self):
```

```
        self.assertEqual(200, self.resp.status_code)
```

```
    def test_texto(self):
```

```
        self.assertContains(self.resp, 'natal')
```

## Feriado v. 0.2

Crie um diretório **templates** no diretório **feriado/**

**feriados/feriado/templates/natal.html**

```
<!DOCTYPE html>
<html lang="pt_br">
  <head>
    <meta charset="UTF-8">
    <title>Feriado de Natal</title>
  </head>
  <body>
    <h1>Não é natal</h1>
  </body>
</html>
```



## Feriado v. 0.2

### **feriados/feriado/views.py**

```
from django.shortcuts import render
```

```
def natal(request):  
    return render(request, 'natal.html')
```

## Feriado v. 0.2

### settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'feriado', # <== Incluir essa linha aqui  
]
```

## Feriado v. 0.3

### Editar a view

```
from django.shortcuts import render
```

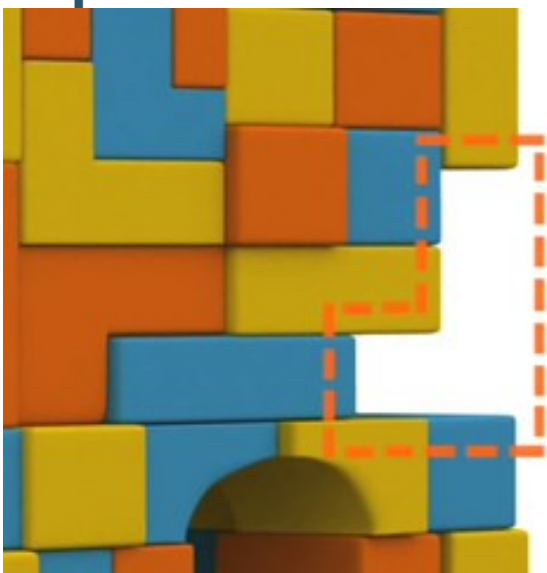
```
def natal(request):  
    contexto = {'natal': True,  
               'carnaval': False}  
    return render(request, 'natal.html', contexto)
```

# Feriado v. 0.3

## Editar o template

```
....  
<body>  
    {% if natal %}  
    <h1>É natal</h1>  
    {% else %}  
        <h1>Não é natal</h1>  
    {% endif %}  
</body>  
....
```

## E os testes ?



## Feriado v. 0.3

### /feriado/tests.py

```
from django.test import TestCase
```

```
class NatalTest(TestCase):
```

```
    def setUp(self):
```

```
        self.resp = self.client.get('/')
```

```
    def test_200_response(self):
```

```
        self.assertEqual(200, self.resp.status_code)
```

```
    def test_texto(self):
```

```
        self.assertContains(self.resp, 'natal')
```

```
    def test_template_natal(self):
```

```
        self.assertTemplateUsed(self.resp, 'natal.html')
```



## Feriado v. 0.4

Um modelo é a fonte única e definitiva de informações sobre seus dados.

Cada modelo é uma classe Python, uma subclasse de `django.db.models.Model`.

Cada atributo representa um campo do banco de dados.

Fonte: <https://docs.djangoproject.com/en/3.0/topics/db/models/>

## Feriado v. 0.4

### **models.py ( do app )**

```
from django.db import models
```

```
class FeriadoModel(models.Model):  
    nome = models.CharField('Feriado',max_length=50)  
    dia = models.IntegerField('Data')  
    mes = models.IntegerField('Mês')  
  
    def __str__(self):  
        return self.nome
```

## Feriado v. 0.4

Execute:

```
python manage.py makemigrations  
python manage.py migrate
```

O que aconteceu ?

## Feriado v. 0.4 - Teste Unitário

```
from feriado.models import FeriadoModel  
from datetime import datetime
```

```
class FeriadoModelTest(TestCase):  
    def setUp(self):  
        self.feriado = 'Natal'  
        self.mes = 12  
        self.dia = 25  
        self.cadastro = FeriadoModel(  
            nome=self.feriado,  
            dia=self.dia,  
            mes=self.mes,  
        )  
        self.cadastro.save()
```

## Feriado v. 0.4 - Teste Unitário ( continuação )

```
def test_created(self):  
    self.assertTrue(FeriadoModel.objects.exists())
```

```
def test_modificado_em(self):  
    self.assertIsInstance(self.cadastro.modificado_em,  
datetime)
```

```
def test_nome_feriado(self):  
    nome = self.cadastro.__dict__.get('nome', '')  
    self.assertEqual(nome, self.feriado)
```

```
def test_dia_feriado(self):  
    dia = self.cadastro.__dict__.get('dia', '')  
    self.assertEqual(dia, self.dia)
```

## Feriado v. 0.4 ( template )

```
<!DOCTYPE html>
<html lang="pt_br">
  <head>
    <meta charset="UTF-8">
    <title>Feriado de Natal</title>
  </head>
  <body>
    {% if feriado %}
      <h1>{{ nome }}</h1>
    {% else %}
      <h1>Hoje não é feriado nenhum.</h1>
    {% endif %}
  </body>
</html>
```



## Feriado v. 0.4 ( views.py do app )

```
from django.shortcuts import render
from .models import FeriadoModel
from datetime import date

def feriado_index(request):
    hoje = date.today()
    feriado_ou_nao = FeriadoModel.objects.filter(
        dia=hoje.day).filter(mes=hoje.month)
    if len(feriado_ou_nao) == 0:
        contexto = {'feriado': False}
    else:
        nome = feriado_ou_nao[0].nome
        contexto = {'feriado': True, 'nome': nome}
    return render(request, 'feriado.html', contexto)
```

## Feriado v. 0.4 ( urls.py do app )

```
from django.urls import path  
from . import views
```

```
urlpatterns = [  
    path("", views.feriado_index),  
]
```

**Orlando Saraiva Jr**

**saraiva@fho.edu.br**

**<https://orlandosaraivajr.github.io/>**



FUNDAÇÃO  
HERMÍNIO OMETTO