

Capítulo 3: NLP en Python

Capítulo 4: Django Web Framework

Capítulo 5: Aplicación Inteligente

# Django Web Framework

Data Science for Developer

# Objetivos

- Explicar el funcionamiento web framework de Django.
- Desarrollar una aplicación web básica en Django.



# Agenda

- Historia de Django
- Quiénes usan Django
- Beneficios de Django
- Django MVT
- Model Layer
- View Layer
- Template Layer
- Django Admin
- Querysets
- Django Template Language
- Django REST
- Proyecto Final



# Historia de Django

- Creado en el 2003 por desarrolladores del periódico Lawrence Journal-World.
- Objetivo inicial: simplificar la gestión de contenidos del periódico en la web.
- Origen del nombre: Django Reinhardt.
- 2008: Administrado por el Django Software Foundation (DSF).
- Versión Actual: 2.2
- Se basa en Python y permite el desarrollo de Web Apps.



# Quiénes usan Django



Quora



The  
Washington  
Post



# Beneficios de Django



**Rápido**



**Funcionalidad  
Incluída**



**Versátil**



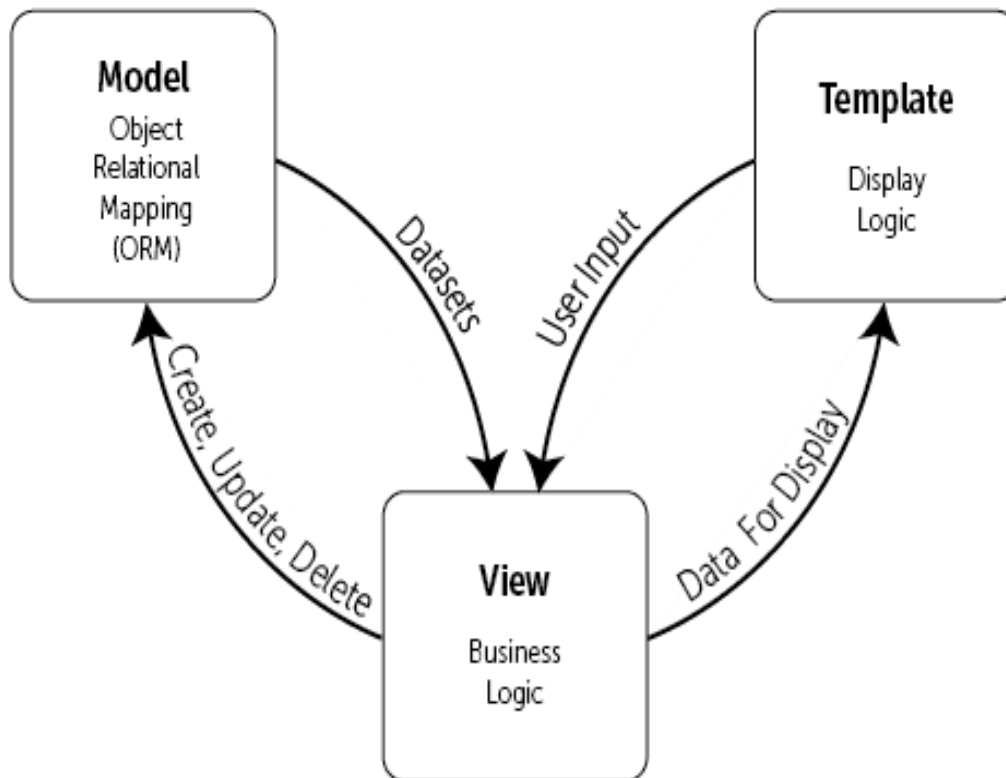
**Seguro**



**Escalable**



# Django MVT



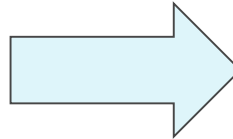
**Model-View-Template  
(MVT)**



# Model Layer: Capa de Datos

```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```



```
CREATE TABLE myapp_person (
    "id" serial NOT NULL PRIMARY KEY,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL
);
```





# View Layer: Capa del Negocio (1)

```
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^articles/2003/$', views.special_case_2003),
    url(r'^articles/([0-9]{4})/$', views.year_archive),
    url(r'^articles/([0-9]{4})/([0-9]{2})/$', views.month_archive),
    url(r'^articles/([0-9]{4})/([0-9]{2})/([0-9]+)/$', views.article_detail),
]
```



## View Layer: Capa del Negocio (2)

```
from django.http import HttpResponse
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>It is now %s.</body></html>" % now
    return HttpResponse(html)
```



# Template Layer: Capa de Presentación

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="style.css" />
  <title>{% block title %}My amazing site{% endblock %}</title>
</head>

<body>
  <div id="sidebar">
    {% block sidebar %}
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/blog/">Blog</a></li>
    </ul>
    {% endblock %}
  </div>

  <div id="content">
    {% block content %}{% endblock %}
  </div>
</body>
</html>
```

- HTML
- CSS
- Javascript
- Y... DTL



# Django Admin

## Django administration

### Site administration

#### AUTHENTICATION AND AUTHORIZATION

**Groups** [+ Add](#) [Change](#)

**Users** [+ Add](#) [Change](#)

#### CORE

**Classrooms** [+ Add](#) [Change](#)

**Education boards** [+ Add](#) [Change](#)

**Schools** [+ Add](#) [Change](#)

**Students** [+ Add](#) [Change](#)

**Subjects** [+ Add](#) [Change](#)

**Teachers** [+ Add](#) [Change](#)

#### Recent actions

##### My actions

[+ IB](#)  
Education board

[+ ICSE](#)  
Education board

[+ CBSE](#)  
Education board



# Querysets

```
from blog.models import Blog
b = Blog(name='Beatles Blog', tagline='All the latest Beatles news.')
b.save()
```

```
>>> all_entries = Entry.objects.all()
```

```
Entry.objects.filter(pub_date__year=2006)
```

```
>>> Entry.objects.filter(
...     headline__startswith='What'
... ).exclude(
...     pub_date__gte=datetime.date.today()
... ).filter(
...     pub_date__gte=datetime.date(2005, 1, 30)
... )
```



# Django Template Language (DTL)

```
{% extends "base_generic.html" %}

{% block title %}{{ section.title }}{% endblock %}

{% block content %}
<h1>{{ section.title }}</h1>

{% for story in story_list %}
<h2>
  <a href="{{ story.get_absolute_url }}">
    {{ story.headline|upper }}
  </a>
</h2>
<p>{{ story.tease|truncatewords:"100" }}</p>
{% endfor %}
{% endblock %}
```

- Variables {{ name }}
- Filtros {{ name|lower }}
- Tags {% if %} {% endif %}
- Comentarios {# #}
- Bloques



# Django REST

django

REST

framework

**Web browsable API**



# Consultas Adicionales Django

- Q&A
- Otros temas de Django
- Dudas





# Proyecto Final

- Desarrollo de una aplicación inteligente:
  - Django
  - Scikit-learn
  - Otros frameworks Python
- Exposición de la aplicación inteligente: (1 h)
  - Objetivos
  - Tecnologías
  - Demo
  - Lecciones aprendidas / recomendaciones



# Proyecto Final I

## Desarrollar una aplicación inteligente en Django y Python

- Formación de grupos para el proyecto final (máx 5 por grupo) - 10 minutos.
- Discutir ideas iniciales del proyecto final - 10 minutos.
- Presentación grupo e ideas iniciales - 10 minutos por grupo.



# Ejercicio N° 4.1: Django

Al finalizar el laboratorio, el alumno logrará:

- Desarrollar un web app sencillo usando Django.



# Proyecto Final

- Desarrollo de una aplicación inteligente:
  - Django
  - Scikit-learn
  - Otros frameworks Python
- Exposición de la aplicación inteligente (1 h)
  - Objetivos
  - Tecnologías
  - Demo
  - Lecciones aprendidas / recomendaciones



# Proyecto Final II

## Desarrollar una aplicación inteligente en Django y Python

- Discutir avances del proyecto final - 10 minutos.
- Presentación avances - 10 minutos por grupo.



# Lectura Adicional

Para obtener información adicional, puede consultar los siguientes enlaces:

- <https://docs.djangoproject.com/en/1.11/>



# Resumen

En este capítulo, usted aprendió que:

- Django es un Web Framework muy popular desarrollado en Python, que permite armar aplicaciones web en tiempos ágiles y con una **estructura optimiza para escalabilidad**.



# Tarea N° 4: Funcionalidad adicional de Django

## 1.11

Revisar la documentación de Django 1.11 y enviar un informe de una hoja máximo.

- Elegir una funcionalidad
- Incluir un trozo de código
- Explicar qué significa en el contexto del desarrollo de web apps.

Enviar por correo al instructor.

