

django

**The web framework for perfectionists
with deadlines.**



CUSOL (Comunidad Universitaria de Software Libre)



¿Por qué django?

- Rápido
- Totalmente precargado
- Seguro
- Escalable
- Versátil



<http://www.django pony.com/>

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

www.djangoproject.com

Un poco de Historia

Fue desarrollado inicialmente por *Lawrence Journal-World* y actualmente es desarrollado por *Django Software Foundation*

Liberado bajo licencia BSD (de 3 cláusulas)



Usan Django...



Características de Django

- Un servidor web ligero incluido
- ORM (object-relational Mapper)
- Un sistema de formularios
- Un sistema de templates
- Un framework de caching
- Un sistema de internacionalización
- Una interfaz para el framework de test unitarios de Python
- Interfaz de administración



Características de Django

- Extendido (más de 2500 paquetes disponibles)
- Puede correr en combinación con
 - ◆ Apache
 - ◆ Ngnix, Gunicorn
- Soporta: Postgres, MySQL, SQLite, Oracle MS
- NoSQL DB: MongoDB Google App Engine
- Migraciones de base de datos!!!



MVT no MVC

→ Modelo

- ◆ Define la estructura de los datos
- ◆ Se encarga de realizar los queries a la base de datos

→ Vista

- ◆ Define que data va a ser presentada
- ◆ Devuelve respuestas HTTP

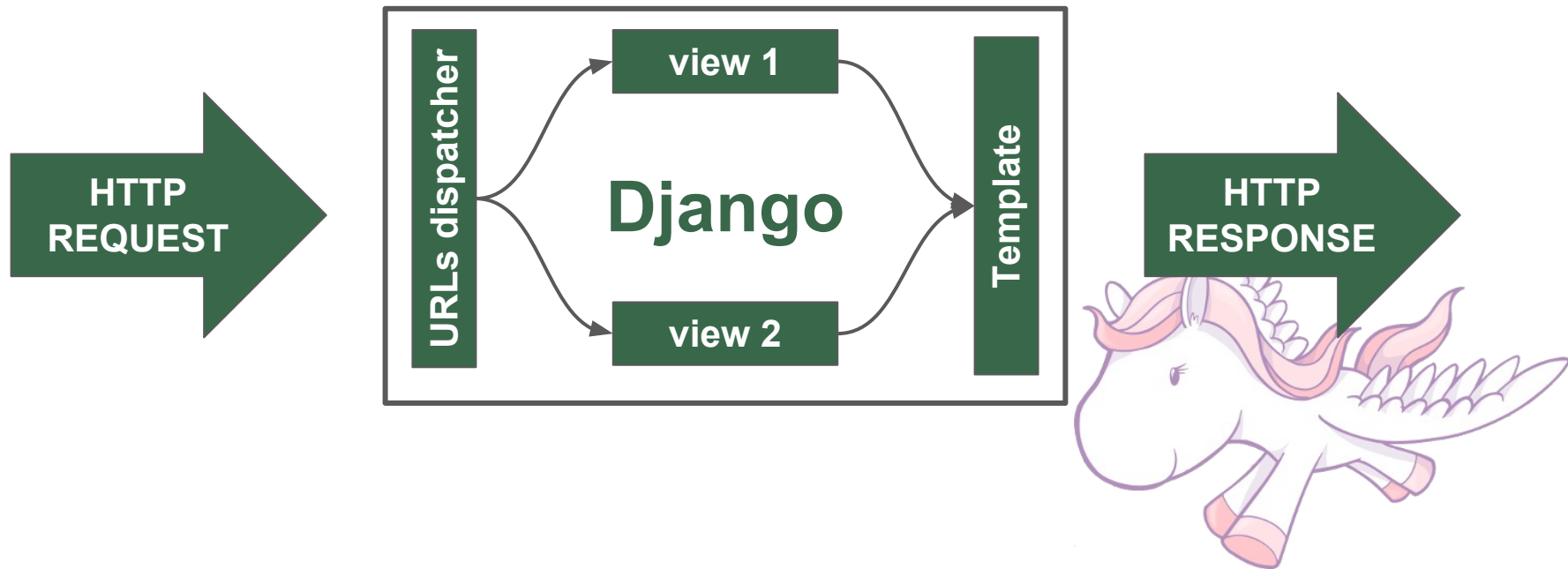
→ Template

- ◆ Devuelve los datos en un formato adecuado

Patrones URLs basados en expresiones regulares



Cómo procesa django una petición



Instalando django

pip, virtualenv, pyenv

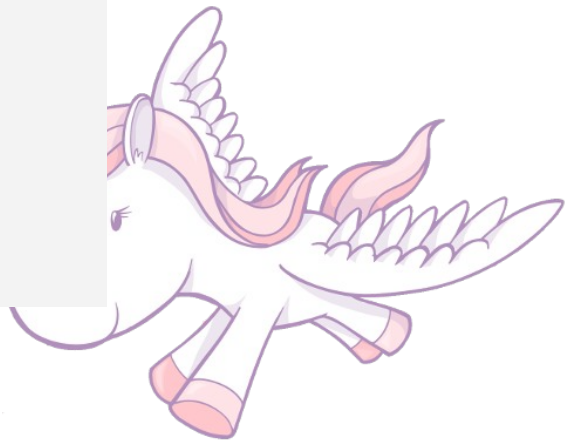
```
$ python3 -m venv mi_proyecto
```

```
$ mkvirtualenv mi_proyecto
```

```
$ workon mi_proyecto
```

```
$ pip install django
```

```
$ pip freeze
```



Estructura del proyecto

```
$ django-admin startproject mi_proyecto
```

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

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

It worked!



Estructura de la aplicación

python manage.py startapp mi_aplicacion

mi_app/

 __init__.py

 admin.py

 apps.py

 migrations/

 __init__.py

 models.py

 test.py

 views.py



Vistas

views.py

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

```
def index(request):
```

```
    return HttpResponse("Hola este es el index de mi aplicación")
```



URLs

mi_app/urls.py

```
from django.conf.urls import url

from . import views

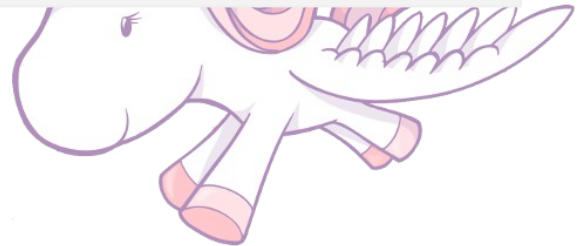
urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

mi_proyecto/urls.py

```
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    url(r'^mi_app/', include('mi_app.urls')),
    url(r'^admin/', admin.site.urls),
]
```

Expresiones regulares!!!



Clases en Python

```
>>> class Potato(object):
    def __init__(self, name, lastname):
        self.name = name
        self.lastname = lastname

    def say_hi(self):
        print('Hi I'm a potato')

    @property
    def full_name(self):
        return "{} {}".format(name, lastname)
```



Vistas basadas en Clases!! (CBV)

views.py

```
from django.http import HttpResponse
from django.views.generic import View
```

```
class MyView(View):
    def get(self, request):
        return HttpResponse('Hola este es el index de mi aplicación')
```

```
urlpatterns = [
    url(r'^$', MyView.as_view()),
]
```



Base de datos

Postgres, MySQL, SQLite

```
INSTALLED_APPS = [...]
```

```
$ python manage.py migrate
```



Modelos

models.py

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

```
class Question(models.Model):
```

```
    question_text = models.CharField(max_length=200)
```

```
    pub_date = models.DateTimeField('date published')
```

```
class Choice(models.Model):
```

```
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
```

```
    choice_text = models.CharField(max_length=200)
```

```
    votes = models.IntegerField(default=0)
```



Modelos (blank, null, __str__)

models.py

...

```
class Question(models.Model):  
    question_text = models.CharField(max_length=200)  
    pub_date = models.DateTimeField('date published')  
  
    def __str__(self):  
        return self.question_text
```



Modelos

```
mi_proyecto/settings.py
INSTALLED_APPS = [...
                    'polls.apps.Polls.Config',
                    ...
                ]
```

```
$ python manage.py makemigrations polls
```

```
$ python manage.py sqlmigrate polls 0001
```

```
$ python manage.py migrate
```



Django Admin

```
$ python manage.py createsuperuser
```

<http://127.0.0.1:8000/admin/>

admin.py

```
from django.contrib import admin
```

```
from .models import Question
```

```
admin.site.register(Question)
```



Django Shell, Queries

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

```
>>> from polls.models import Question, Choice
>>> Question.objects.all()
>>> from django.utils import timezone
>>> q = Question(question_text="Cual es el mejor lenguaje",
                  pub_date=timezone.now())
>>> q.save()
>>> q.question_text
>>> Question.objects.all()
>>> Question.objects.filter(pub_date__year=2015)
```



Vistas

```
from django.template import loader
```

```
def index(request):
```

```
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
```

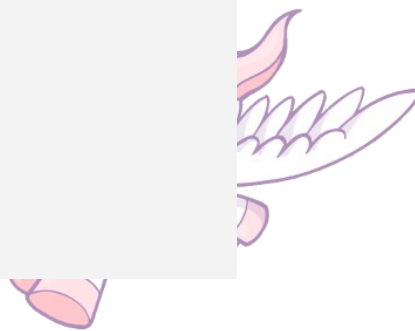
```
    template = loader.get_template('polls/index.html')
```

```
    context = {
```

```
        'latest_question_list': latest_question_list,
```

```
    }
```

```
    return HttpResponse(template.render(context, request))
```



Vistas CBV

```
from django.views.generic import ListView

def Index(ListView):
    model = Question
    queryset = Question.objects.order_by('-pub_date')[:5]
    template_name = 'polls/index.html'
    context_object_name = 'latest_question_list'
```



Templates

```
<html>
  {% if latest_question_list %}
    <ul>
      {% for question in latest_question_list %}
        <li><a href="/polls/{{ question.id }}/"/>{{ question.question_text }}</a></li>
      {% endfor %}
    </ul>
  {% else %}
    <p>No polls are available.</p>
  {% endif %}
</html>
```


Lecturas Recomendadas

- PEP8, escribir como un pythonista python.org/dev/peps/pep-0008/
- Django tutorial docs.djangoproject.com/en/1.9/intro/
- Django wiki docs.djangoproject.com/
- Django girls tutorial tutorial.djangogirls.org/
- Templates github.com/pydanny/cookiecutter-django
- Crispy forms
- CBV
- Rest (Angular.js)
- Tests!!



Únete a CUSOL!!

