

django

Introducción a Django

<https://civan@bitbucket.org/civan/lilbeef>

Javier Civantos
@civantoz
javiercivantos@gmail.com

Lema de Django

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

Ridículamente rápido

¡LEER LA DOCUMENTACIÓN!

- Entender el paradigma de desarrollo
- Instalar Django
- Iniciar una aplicación propia

Baterías incluidas

- **ORM (Mapeo objeto relacional)**
- **Control de migraciones**
- **Autenticación / Autorización**
- **Control de sesiones**
- **Vistas (Ciclo de Request - Response)**
- **Formularios**
- **Plantillas**
- **Internacionalización**
- **Logger**
- **Caché**
- **Seguridad**

Tranquilizadoramente Seguro

De serie Django te ayuda a evitar...

- **Cross site scripting (XSS) protection**
- **Cross site request forgery (CSRF) protection**
- **SQL injection protection**
- **Clickjacking protection**
- **SSL/HTTPS**
- **Host header validation**
- **Session security**

Extremadamente Escalable

Arquitectura “shared-nothing”

Permite añadir más hardware en:

- Servidores de BBDD
- Servidores Caché
- Servidores Web o de aplicación.

Increíblemente versátil

¡Django NO ES UN CMS!

Django ES UN FRAMWORK base para extender y ofrecer soluciones.

Front / Back / API / Desacoplamiento

Vamos a situarnos

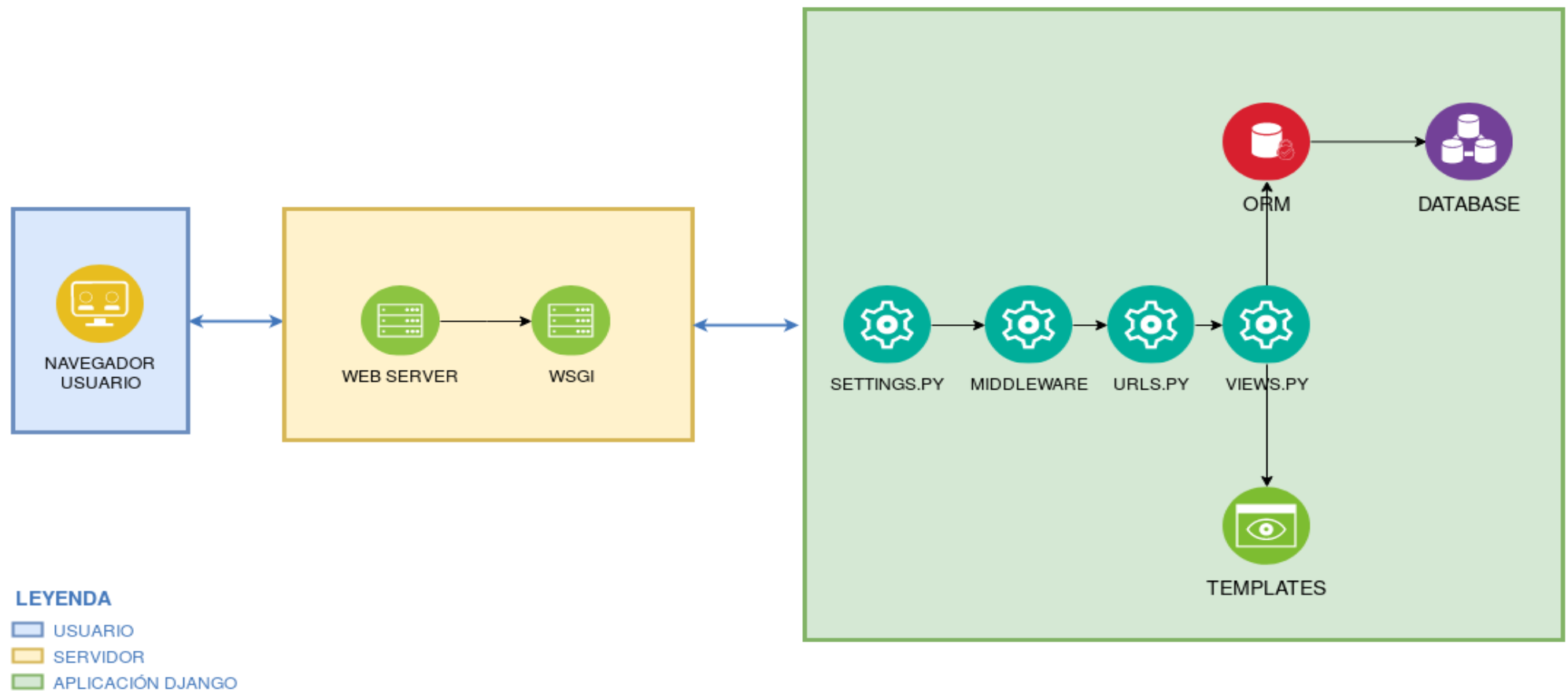
- **Paradigma de programación:**
POO (Python, recordad, todo objetos)
- **Patrón de arquitectura de software:**
Sistema > MVC

Pequeña mentira

**Patrón MVT - Modelo / Vista / Plantilla
(Template)**

**Ordenados y a las cosas por su
nombre.**

Ciclo de Vida de Django



Nuevo proyecto Django

BUENAS PRÁCTICAS

- **Planificación de desarrollo** ⇄⇄!
- **VirtualEnvs**
- **Control de código y versiones (GIT...)**
- **Versiones adecuadas.**
 - Django 2 (o LTS para clientes finales)
 - Python3 para todo lo nuevo (2.7 solo mantenimiento – Planear migración!!!)
 - PIP3 para gestionar dependencias

Creación de proyecto 1/2

```
civan ~$ virtualenv --python=python3 ./lilbeef
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/civan/lilbeef/bin/python3
Also creating executable in /home/civan/lilbeef/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
civan ~$ cd lilbeef/
civan ~/lilbeef$ source bin/activate
civan (p) lilbeef$ ~/lilbeef$ pip install django
Collecting django
  Using cached Django-2.0.3-py3-none-any.whl
Collecting pytz (from django)
  Using cached pytz-2018.3-py2.py3-none-any.whl
Installing collected packages: pytz, django
Successfully installed django-2.0.3 pytz-2018.3
civan (p) lilbeef$ ~/lilbeef$ django-admin startproject lilbeef
civan (p) lilbeef$ ~/lilbeef$ cd li
lib/      lilbeef/
civan (p) lilbeef$ ~/lilbeef$ cd lilbeef/
```

Creación de proyecto 2/2

```
civan [p] lilbeef [~/lilbeef/lilbeef] ./manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
civan [p] lilbeef [~/lilbeef/lilbeef] ./manage.py runserver
Performing system checks...

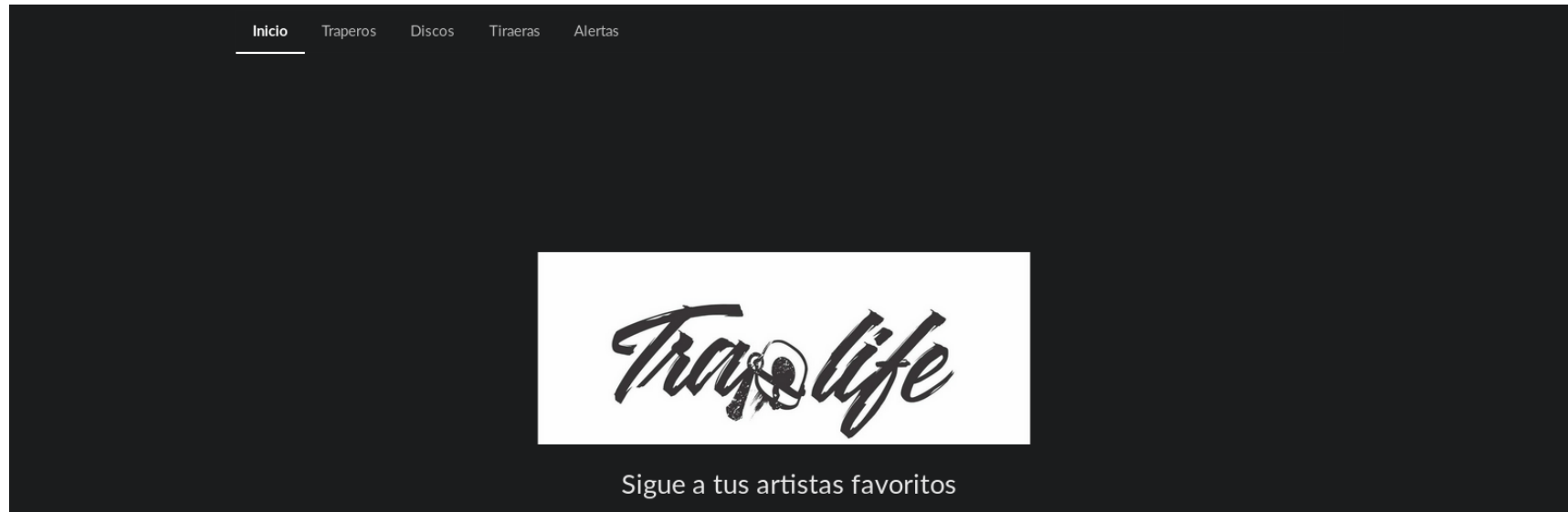
System check identified no issues (0 silenced).
April 02, 2018 - 18:20:20
Django version 2.0.3, using settings 'lilbeef.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Clonar y arrancar el proyecto

```
civan ~ virtualenv --python=python3 ./lilbeef
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/civan/lilbeef/bin/python3
Also creating executable in /home/civan/lilbeef/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
civan ~ cd lilbeef/
civan ~/lilbeef source bin/activate
civan (p) lilbeef ~/lilbeef git clone https://civan@bitbucket.org/civan/lilbeef.git
Clonando en 'lilbeef'...
remote: Counting objects: 331, done.
remote: Compressing objects: 100% (281/281), done.
remote: Total 331 (delta 102), reused 156 (delta 43)
Recibiendo objetos: 100% (331/331), 5.71 MiB | 2.13 MiB/s, listo.
Resolviendo deltas: 100% (102/102), listo.
civan (p) lilbeef ~/lilbeef cd lilbeef/
civan ^ master (p) lilbeef ~/lilbeef/lilbeef pip install -r requirements.txt
Collecting Django==2.0.3 (from -r requirements.txt (line 1))
  Using cached Django-2.0.3-py3-none-any.whl
Collecting pytz==2018.3 (from -r requirements.txt (line 2))
  Using cached pytz-2018.3-py2.py3-none-any.whl
Installing collected packages: pytz, Django
Successfully installed Django-2.0.3 pytz-2018.3
civan ^ master (p) lilbeef ~/lilbeef/lilbeef ./manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
April 02, 2018 - 17:53:46
Django version 2.0.3, using settings 'lilbeef.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Portada de la aplicación



Listado de traperos



C. Tangana

Nombre: Antón Álvarez Alfaro

Edad: 28

Cecilio G

Nombre: Juan Cecilia Ruiz

Edad: 24

Settings.py

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '-o25smqkv-8p^f8v+4ol68lw1-==@^w0+q(d@4542t=taqbrr_k'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    #gestionamos aplicaciones propias
    'traperos'
]
```

- **VAMOS EL EDITOR**

Urls.py

```
from django.contrib import admin
from django.urls import include, path
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('traperos.urls'))
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

- **VAMOS EL EDITOR**

Modelos de datos

```
class Trapero(models.Model):
    """Modelo con informacion sobre los traperos."""

    aka = models.CharField(max_length=100)
    nombre_real = models.CharField(max_length=250, blank=True, null=True)
    ano_nacimiento = models.IntegerField(default=1980)
    foto = models.FileField(upload_to='fotos/', default='logos/anonymous.jpg')

    @property
    def edad(self):
        """Devuelve la edad aproximada usando el ano actual como base."""
        ano_actual = datetime.datetime.now().year
        return '{0}'.format(ano_actual - self.ano_nacimiento)

    def __str__(self):
        """Conversion de objeto a cadena humana."""
        return '{0}'.format(self.aka)
```

Modelos de datos

```
class Disco(models.Model):
    """Modelo con informacion sobre los discos."""

    nombre = models.CharField(max_length=250)
    fecha_publicacion = models.DateField()
    portada = models.FileField(upload_to='portadas/', default='img/disco.jpg')
    traperero = models.ManyToManyField('Traperero', related_name='disco_traperero')

    def __str__(self):
        """Conversion de objeto a cadena humana."""
        return '{0}'.format(self.nombre)
```

Modelos de datos

```
class Tiraera(models.Model):
    """Modelo con informacion sobre las tiraeras entre traperos."""

    titulo = models.CharField(max_length=250)
    fecha_inicio = models.DateField()
    descripcion = models.TextField()
    trapero_tira = models.ForeignKey('Trapero', on_delete=models.CASCADE, related_name='trapero_tirador')
    trapero_recibe = models.ForeignKey('Trapero', on_delete=models.CASCADE, related_name='trapero_recibor')

    def __str__(self):
        """Conversion de objeto a cadena humana."""
        return '{0}'.format(self.titulo)
```

Modelos de datos

```
class Alerta(models.Model):  
    """Modelo para recibir alertas de usuarios a través del formulario."""  
  
    titulo = models.CharField(max_length=250)  
    fecha_alerta = models.DateField(auto_now_add=True)  
    descripcion = models.TextField()  
    trapero = models.ForeignKey('Trapero', on_delete=models.CASCADE, related_name='alertas_traperos')  
  
    def __str__(self):  
        """Conversion de objeto a cadena humana."""  
        return '{0}'.format(self.titulo)
```

Petición basada en función

URL

```
path('respuesta/', views.respuesta_simple, name='respuesta'),
```

Vistas

```
def respuesta_simple(request):  
    """Ejemplo de respuestas simple."""  
    return HttpResponse("Bienvenido a Lil Beef")
```

- Request ↔ Response

Petición atajo “render”

URL

```
path('', views.index, name='index'),
```

Vistas

```
def index(request):  
    """Vista para mostrar el indice de la aplicacion."""  
    traperos = Trapero.objects.all().order_by('?')[:2]  
    discos = Disco.objects.order_by('-fecha_publicacion')[:2]  
    tiraeras = Tiraera.objects.order_by('-fecha_inicio')[:2]  
    context = {'traperos': traperos, 'discos': discos, 'tiraeras': tiraeras}  
    return render(request, 'index/index.html', context)
```

Petición basada en clases genéricas

URL

```
path('discos/', views.DiscoListView.as_view(), name='discos'),
```

Vistas

```
class DiscoListView(ListView):  
    """ListView Disco."""  
  
    model = Disco  
    template_name = 'discos/listado_todos_discos.html' # Default: <app_label>/<model_name>_list.html  
    context_object_name = 'listado_todos_discos' # Default: object_list  
    paginate_by = 3  
    queryset = Disco.objects.all() # Default: Model.objects.all()
```


Petición basada en clases (CRUD)

URL

```
path('alertas/', views.AlertaCreate.as_view(), name='alertas'),
```

Vistas

```
class AlertaCreate(CreateView):  
    """Vista de creacion de Alertas."""  
  
    model = Alerta  
    template_name = 'alertas/alertas_form.html'  
    success_url = '/alertas/gracias'  
    fields = ['titulo', 'descripcion', 'trapero', ]
```

Plantillas

```
<div class="ui special cards">
  {% for disco in listado_todos_discos %}
    <div class="card">
      <div class="blurring dimmable image">=
      <div class="content">
        {% for trapero in disco.trapero.all %}
          <p>Artista: <a href="{% url 'trapero_id' trapero.pk %}">{{ trapero.aka }}</a></p>
        {% endfor %}
        <div class="meta">
          <span class="date">Título: {{ disco.nombre }}</span>
        </div>
      </div>
    </div>
  {% empty %}
    <h2>No hay discos disponibles</h2>
  {% endfor %}
</div>
```

- **VAMOS EL EDITOR**

FIN DE LA TORTURA

- **¡MUCHAS GRACIAS POR ASISTIR!**

- **¿DUDAS?**

- **CONTACTO:**

javiercivantos@gmail.com

@civantoz