Introducción a Django

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 incluídas

Modelos y BBDD / ORM

- **Control de migraciones**
- **Autenticación / Autorización**
- **Control de sesiones**
- **Gestión Petición / Respuesta**
- **Vistas**
- **Formularios**
- **Plantillas**
- **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 Framework base para extender y ofrecer soluciones.

Front / Back / Desacoplamiento / API...

Vamos a situarnos

- **Paradigma de programación**

  POO (Python, recordad, todo objetos)

- **Patrón de diseño**

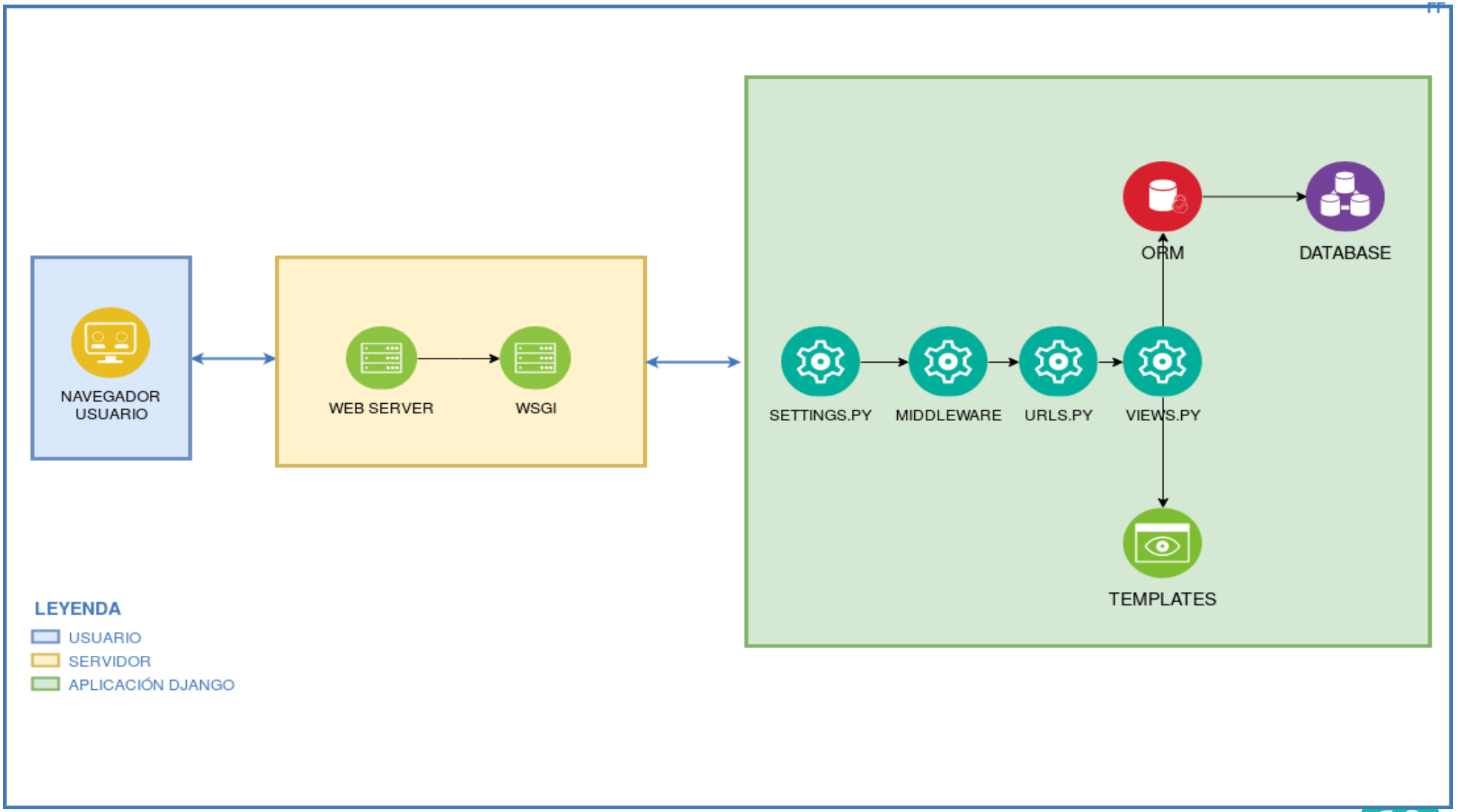
  Sistema > MVC

Pequeña mentira

Patrón MVT – Modelo / Vista / Plantilla

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

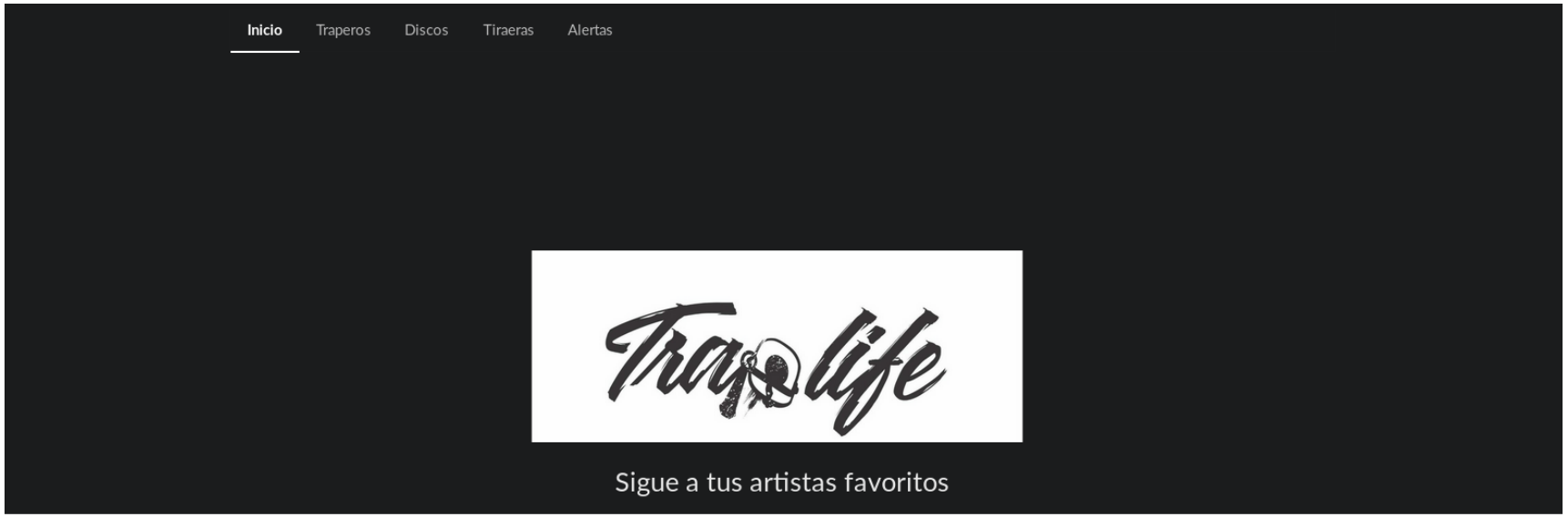
Creación del proyecto

```
$ virtualenv --python=python3 ./django_demo_extrepython

$ cd django_demo_extrepython/

$ source bin/activate

$ pip install django

$ django-admin startproject django_demo_extrepython
```

Requisitos y commit inicial

```
$ cd django_demo_extrepython/
$ pip freeze > requirements.txt
$ git flow init
$ git add .
$ git commit
$ ./manage.py migrate
$ ./manage.py runserver
```

Portada de la aplicación



Listado de traperos

C. Tangana

Nombre: Antón Álvarez Alfaro

Edad: 28

Cecilio G

Nombre: Juan Cecilia Ruiz

Edad: 24

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

```python
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')
    trapero = models.ForeignKey('Trapero', on_delete=models.CASCADE, related_name='disco_trapero')

    def __str__(self):
        """Conversion de objeto a cadena humana."""
        return '{0} - {1}'.format(self.nombre, self.trapero.aka)
```

Modelos de datos

```python
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

```python
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 ↔ Respose**

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 basda 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 basda 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

```
{% for disco in discos %}
  <h3 class="ui header"><a href="{% url 'disco_id' disco.id %}">{{ disco.nombre }}</a></h3>
  <p>Artista: <a href="{% url 'trapero_id' disco.trapero.id %}">{{ disco.trapero.aka }}</a></p>
  <p>Fecha lanzamiento: {{ disco.fecha_publicacion }}</p>
  <a href="{% url 'disco_id' disco.id %}"><img class="trapero" src="{{ disco.portada.url }}" width="100px"></img></a>
  <div class="ui divider"></div>
{% empty %}
  <h3>Actualmente no hay discos registrados</h3>
{% endfor %}
```

- **VAMOS EL EDITOR**

S|NGULAR BOOTCAMP!

- **Bootcamp en S|NGULAR Badajoz.**
- **Alumnos con carrera terminada.**
- **Primer empleo.**
- **Posibilidad de crecimiento.**

FIN DE LA TORTURA

- **¡MUCHAS GRACIAS POR ASISTIR!**
- **¿DUDAS?**
- **CONTACTO:**
  javiercivantos@gmail.com
  @civantoz