

A graphic on the left side of the slide features four horizontal bars of increasing height from bottom to top, colored purple, blue, yellow, and orange. The orange bar is the tallest and has a white arrow pointing to the right. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across the bars in white.

Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Reunión 8

Django: Templates

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Reunión 11

Django: Views - 2

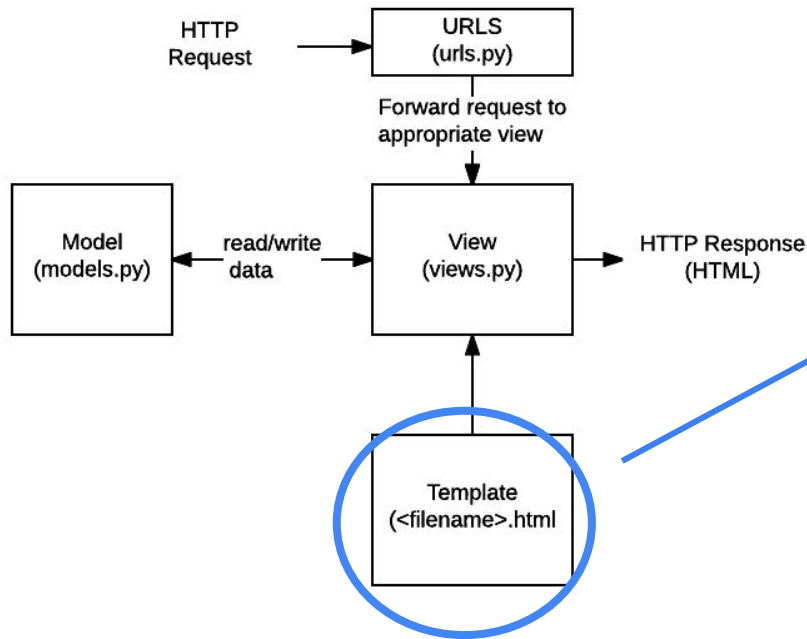
- Url y vistas parametrizadas
- Analizando el HttpRequest
- Tipos de HttpResponse
- Html desde Django

Reunión 12

Django: Templates - 1

- Template backend
- Template loader
- Configuración
- Integración con Vistas
- Contexto
- Variables

Plantillas (templates)

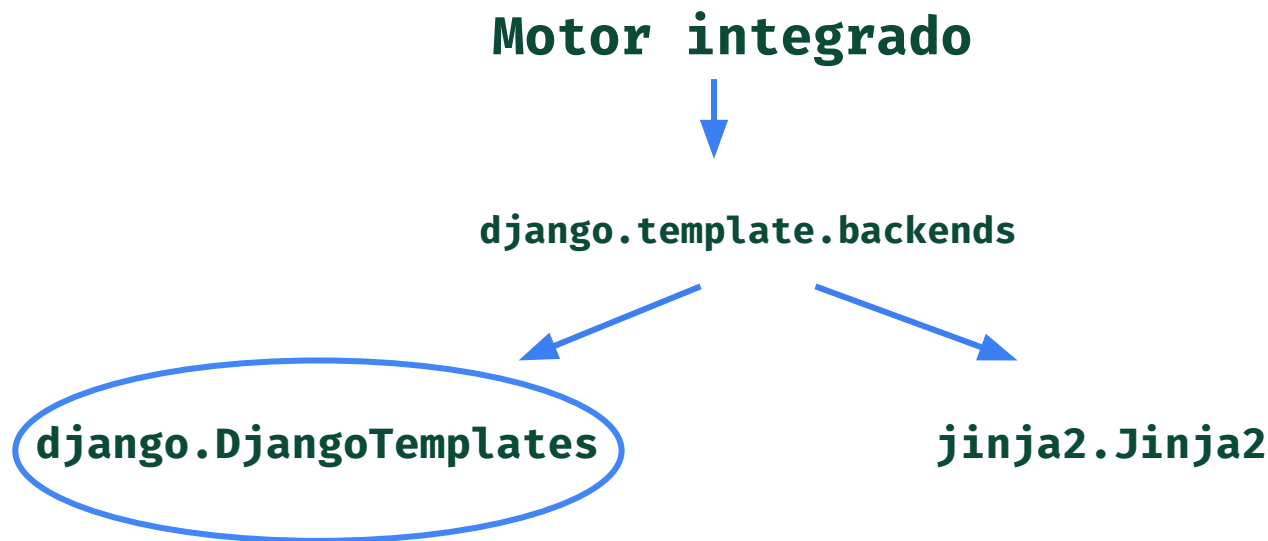


PRESENTACIÓN != LÓGICA DE NEGOCIO

Se enfoca en cómo debe ser mostrado un documento.

GENERA HTML DINÁMICAMENTE CON UNA SINTAXIS ESPECIAL

Motor de Plantillas (templates backend)



Configuración (backend)

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            # ... some options here ...  
        },  
    },  
]
```

Lista de directorios en los que
el motor debe buscar archivos
fuente de plantillas

Indica si el motor debe buscar
plantillas dentro de las
aplicaciones instaladas (en la
carpeta templates)

Configuración (loader)

DjangoTemplates → django.template.loaders.filesystem.Loader

```
TEMPLATES = [{  
    'BACKEND': 'django.template.backends.django.DjangoTemplates',  
    'DIRS': [BASE_DIR / 'templates'],  
    'OPTIONS': {  
        'loaders': [  
            ('django.template.loaders.cached.Loader', [  
                'django.template.loaders.filesystem.Loader',  
                'django.template.loaders.app_directories.Loader',  
                'path.to.custom.Loader',  
            ]),  
        ],  
    },  
}]
```

```
TEMPLATES = [{  
    'BACKEND': 'django.template.backer  
    'DIRS': [BASE_DIR / 'templates'],  
}]
```

Debug=False y OPTIONS['loaders'] sin especificar

Debug=True → Lee y compila sus plantillas cada vez que se procesan

Lenguaje de Plantillas de Django (DTL)

Una plantilla se representa con un contexto



Cuando se renderiza (del lado del servidor)



Se remplazan las
variables por valores

Variables → `{{ variable }}`



Se ejecutan
las sentencias

Sentencias → `{% tag %}`
 contenido
 `{% endtag %}`

(próxima clase)

Ejemplo: Plantilla

```
hola_mundo > templates > <> index.html
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hola Mundo!!</title>
8  </head>
9  <body>
10     <h1 style="color:blue">HOLA GENTE! hoy es {{hoy}}</h1>
11 </body>
12 </html>
```

Ejemplo: Url y Vista

```
urls.py  X  index.html
hola_mundo > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.index, name="index"),
6  ]
```

```
from datetime import datetime
from django.http import HttpResponse
from django.template import loader

def index(request):
    template = loader.get_template('index.html')
    context = {"hoy": datetime.now}
    return HttpResponse(template.render(context, request))
```



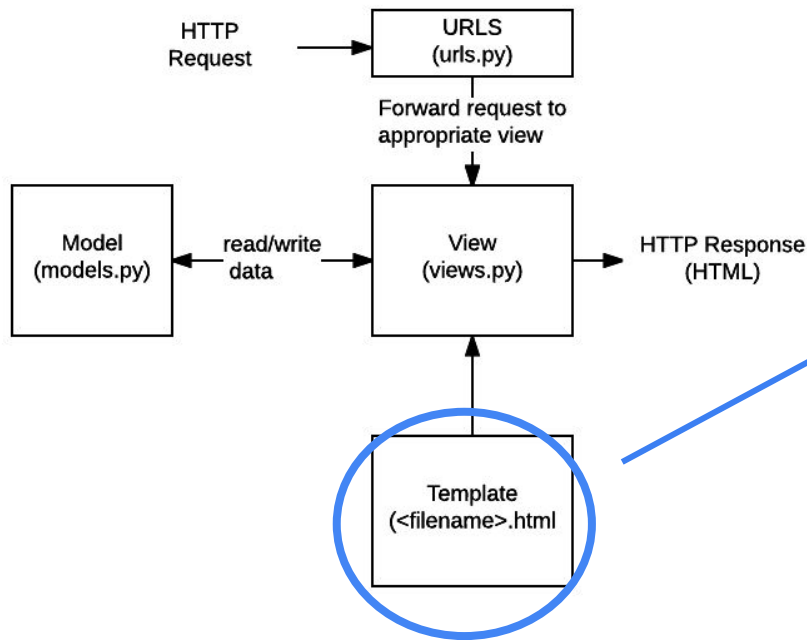
```
from datetime import datetime
from django.shortcuts import render

def index(request):
    return render(request, "index.html", {"hoy": datetime.now})
```

¿Qué es DTL?

El **L**enguaje de **P**lantillas de **D**jango (**D**jango **T**emplate **L**anguage) es un lenguaje que está diseñado para lograr un equilibrio entre potencia y facilidad. Está pensado para que aquellos que están acostumbrados a trabajar con HTML se sientan cómodos. Si tiene conocimiento de otros lenguajes de plantillas basadas en texto como Smarty o Jinja2, se sentirá como en casa con este lenguaje.

Recordemos



Se **enfoca** en cómo debe ser **mostrado** un documento. **NO** se programa lógica de negocio.

Variables

```
{{ variable }}
```



Tags (etiquetas)

Simples



```
csrf_token:  
{% csrf_token %}
```

Argumentos



```
url:  
{% url 'nombre-url' v1 v2 %}  
  
cycle:  
<li style='color:{% cycle 'red' 'blue' %}'>  
  item  
</li>  
  
include:  
{% include "cac/template.html" %}
```

Apertura y Cierre



```
if:  
{% if variable == 1 %}  
  <h1>Hello!</h1>  
{% endif %}  
  
for:  
<ul>  
  {% for x in frutas %}  
    <li>  
      {{ x }}  
    </li>  
  {% endfor %}  
</ul>
```

Tag: Filter

```
{% filter upper %}  
<h1>hola alejandro, como andas?</h1>  
{% endfilter %}
```

```
{% filter first | upper %}  
<h1>hola alejandro, como andas?</h1>  
{% endfilter %}
```

En variables



```
<h1>Hola {{ nombre|upper }}, como andas?</h1>
```

```
<h1>Hola {{ nombre|first|upper }}, como andas?</h1>
```

Tag: Comment

Comentarios

```
<h1>Buenas gente!</h1>
{% comment %}
<h1>Odio a todo el mundo</h1>
{% endcomment %}
```

Comentarios con descripción

```
<h1>Buenas gente!</h1>
{% comment "este era el mensaje original" %}
<h1>Odio a todo el mundo</h1>
{% endcomment %}
```

Comentarios pequeños

```
<h1>Hola gente{# los odio #}!</h1>
```


Herencia de Plantillas (tags)

block

base.html

```
<!DOCTYPE html>
<html Lang=en">
<head>
  <link rel="stylesheet" href="style.css">
  <title>{% block titulo %}Mi sitio{% endblock %}</title>
</head>
<body>
<header id="navbar">
</header>
<main id="contenido">
  {% block contenido %}{% endblock %}
</main>
</body>
</html>
```

extends

blog.html

```
{% extends "base.html" %}
{% block titulo %}Mi blog{% endblock %}
{% block contenido %}
  {% for entry in blog_entries %}
    <h2>{{ entry.title }}</h2>
    <p>{{ entry.body }}</p>
  {% endfor %}
{% endblock %}
```

Herencia de Plantillas (enfoque recomendado)

```
app/  
  templates/  
    app/  
      base.html  
      index.html  
      ...  
  static/  
    app/  
      js/  
        main.js  
        ...  
      css/  
        base.css  
        ...  
      img/
```

Gestión de archivos estáticos (js, css, etc)

INSTALLED_APPS



django.contrib.staticfiles

settings.py



STATIC_URL = '/static/'

templates



```
{% load static %}  

```

STATICFILES_DIRS



```
STATICFILES_DIRS = [  
    BASE_DIR / "static",  
    "C:/Users/user/static_extra",  
]
```

Prefijos



```
STATICFILES_DIRS = [  
    # ... ("downloads", "/opt/webfiles/stats"),  
]
```

```
<a href="{% static 'downloads/polls_20101022.tar.gz' %}">
```

Asociar con
enfoque utilizado
en plantillas

**No te olvides de completar la
asistencia y consultar dudas**

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

TODO EN EL AULA VIRTUAL