

A graphic on the left side of the slide. It features a 3D effect with four overlapping rectangular blocks in purple, orange, yellow, and blue. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these blocks in white. An orange arrow points to the right from the orange block.

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# DJANGO

## Clase 12

Django: Templates - 1

# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 11

### Django: Views - 2

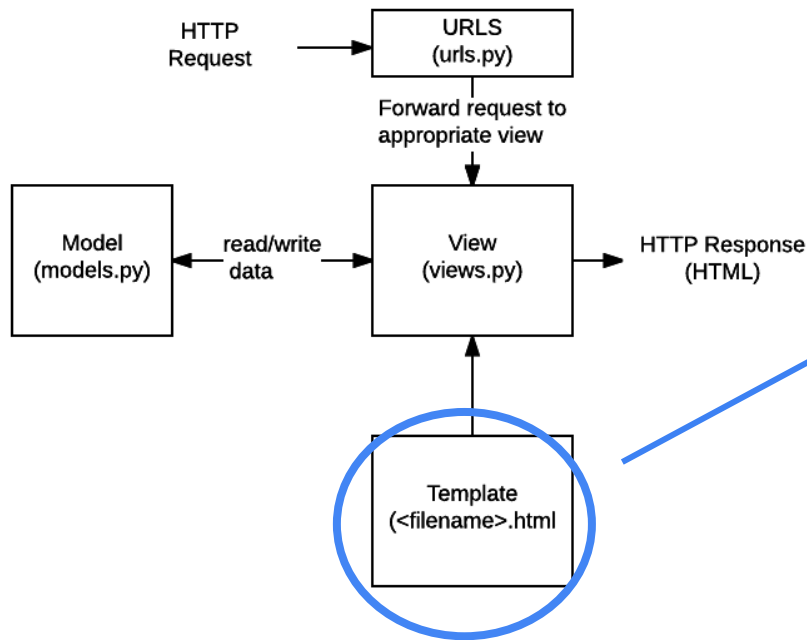
- Url y vistas parametrizadas
- Analizando el HttpRequest
- Tipos de HttpResponse
- Html desde Django

## Clase 12

### Django: Templates - 1

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

# Plantillas (templates)

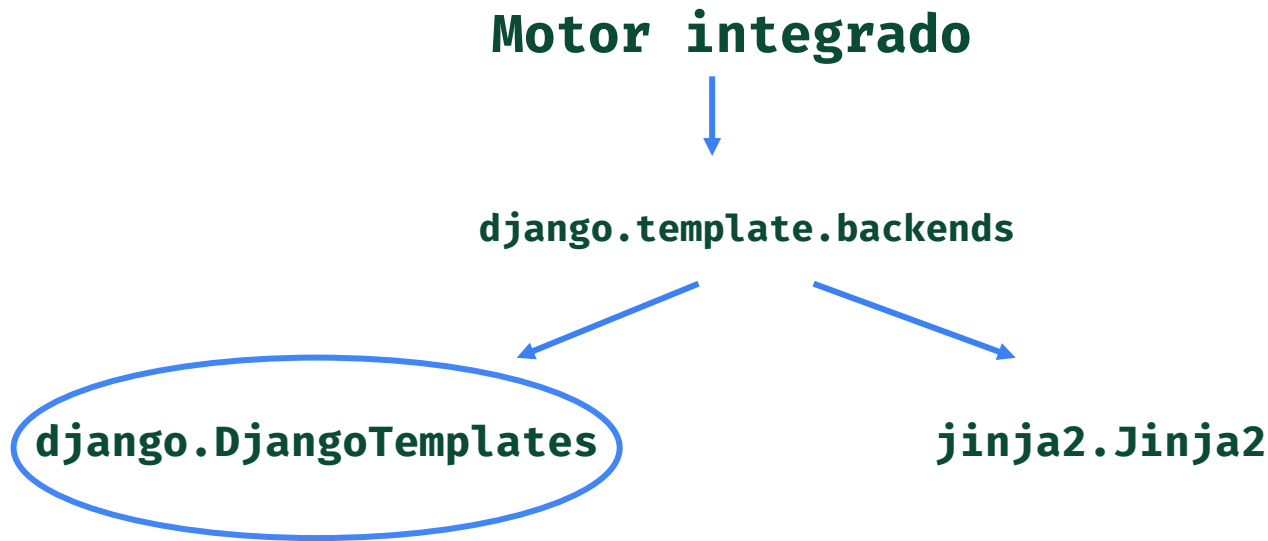


PRESENTACIÓN != LÓGICA DE NEGOCIO

Se enfoca en como debe ser mostrado un documento.

GENERA HTML DINÁMICAMENTE CON UNA SINTÁXIS 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})
```

**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**