

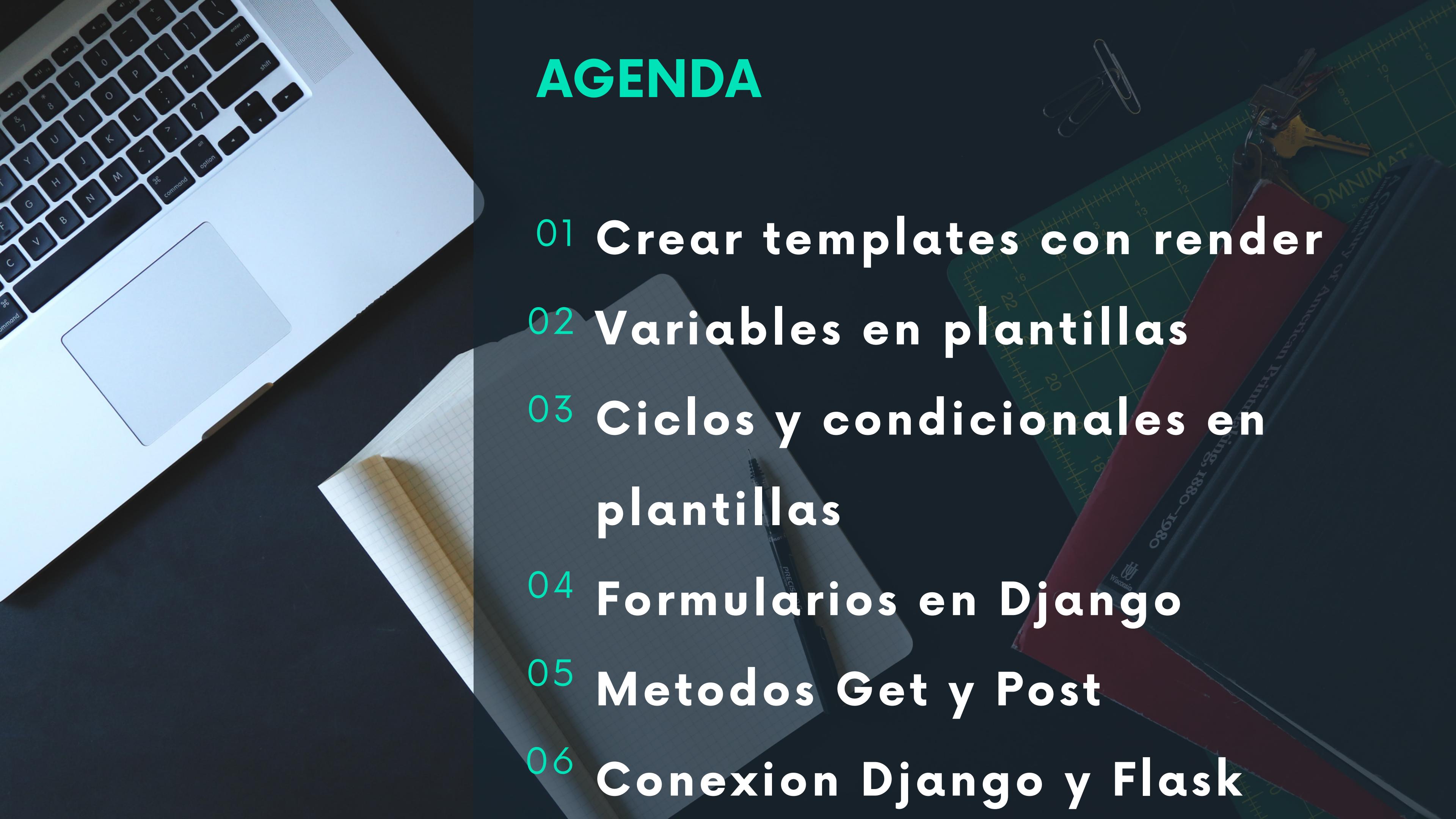


IEEE  
COMPUTER  
SOCIETY

*Universidad de San Carlos de Guatemala  
Student Branch Chapter*

# TALLER DJANGO / FLASK CLASE 6

TUTOR: LUIS GARCIA



# AGENDA

- 01 Crear templates con render
- 02 Variables en plantillas
- 03 Ciclos y condicionales en plantillas
- 04 Formularios en Django
- 05 Metodos Get y Post
- 06 Conexion Django y Flask



CREAR PLANTILLAS

# RENDER

## 1. Importamos render

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

## 2. retornar render en las vistas en views.py

```
return render(request, "agregar.html", {"msg": mensaje})
```

Diagram illustrating the parameters of the render function:

- A white curved arrow points from the word "request" to the first argument of the render function.
- A white curved arrow points from the word "contexto" to the second argument of the render function.
- A white curved arrow points from the text "nombre del html" to the file name "agregar.html" in the third argument of the render function.

# RENDER

## 3. Configurar la ruta de los template en settings.py

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['C:/Users/juana/OneDrive/Desktop/ProyectosDjango/Proyecto1/Proyecto1/plantillas'],
        'APP_DIRS': True,
        'OPTIONS': {
```

directorio donde  
tengamos los templates



# USAR VARIABLES EN PLANTILLAS



# IMPRIMIR VARIABLES EN TEMPLATES

`{{ identificador }}`

## JERARQUIA DE LLAMADAS

1. Diccionario
2. Atributo
3. Objeto
4. Índice de lista



CICLOS Y  
CONDICIONALES EN  
PLANTILLAS



## CONDICIONAL IF

```
{% if condicion %}
```

```
//sentencias
```

```
{% elif condicion %}
```

```
//sentencias
```

```
{% else %}
```

```
//sentencias
```

```
{% endif %}
```

## CICLO FOR

```
{% for elem in lista %}
```

```
//sentencias
```

```
{% endfor %}
```



# FORMULARIOS EN DJANGO



# Crsf token

Es una medida de seguridad que se ejecuta en cada POST que impide ataque a la web por secuestro de sesion

Lo que hace el Crsf token es crear un token diferente cada vez que el usuario este en la pagina web



METODO GET Y POST

# VERIFICAR EL METODO QUE SE ESTA ENVIANDO

Para especificar a nuestra vista que metodo se esta enviando debemos usar el condicional if y el request

Metodo que se  
esta usando



```
if request.method == "GET":  
  
    return render(request, "agregar.html", {})  
  
elif request.method == "POST":
```

# RECOLECTAR DATOS DEL FORM

Para recolectar los datos de un formulario tambien se hace uso del request

Identificador



```
pelicula = request.POST["pelicula"]  
genero = request.POST["genero"]  
descripcion = request.POST["descripcion"]
```

Variable donde se guarda el valor



Metodo que se envio en el form



# CONEXIÓN DJANGO Y FLASK



# REQUESTS

Es una biblioteca HTTP para python. Permite Hacer solicitudes HTTP/1.1 con mucha facilidad. Esto funciona automaticamente gracias a urllib3

## Instalacion

pip install requests

## Uso

Metodo que maneja  
el endpoint

```
response = requests.get("http://localhost:4000/")
```



Url del  
servidor

# PASOS

01

Crear las rutas del API  
(Flask)

02

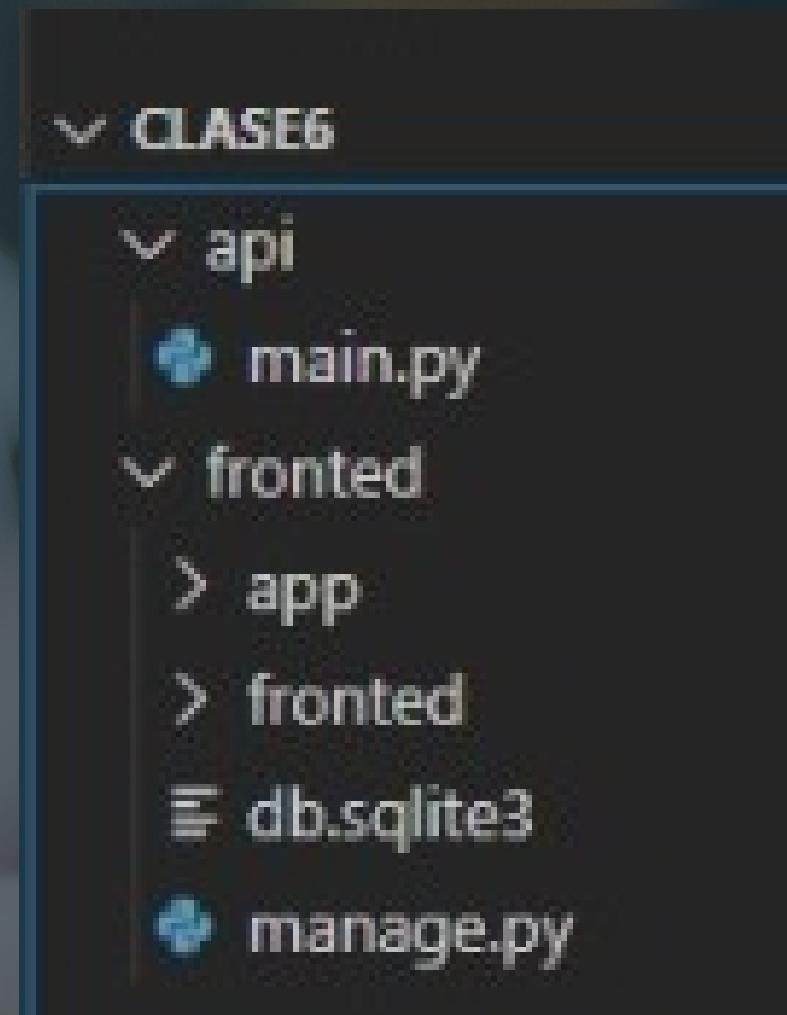
Crear el proyecto  
Django

"django-admin  
startproject fronted"

03

Crear un app dentro del  
proyecto

"py manage.py startapp  
app"



04

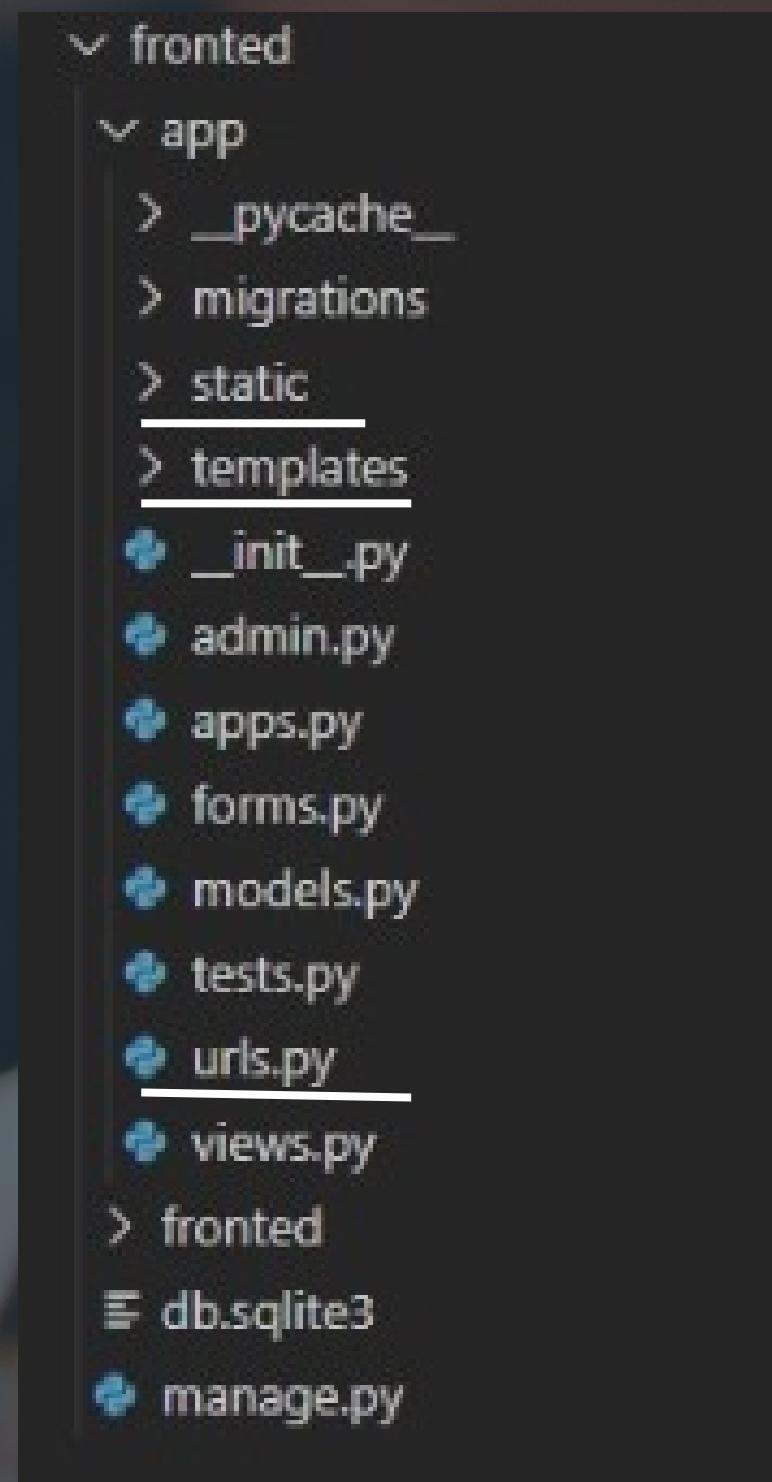
Crear carpeta  
"templates" y "static"  
dentro de app

05

Agregar la app en  
settings.py del fronted

06

Crear urls.py dentro de  
app



Archivo settings.py

Paso 5

```
INSTALLED_APPS = [  
    'app.apps.AppConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages'
```

07

Crear las vistas en  
views.py dentro de app



Aqui se usa el  
requests para hacer  
la peticion

08

Crear la ruta en urls.py  
en app

Paso 6

09

Incluir el archivo  
anterior en urls.py de  
fronted

```
path("", include("app.urls"))
```



# GRACIAS