

INTRODUCCIÓN

---

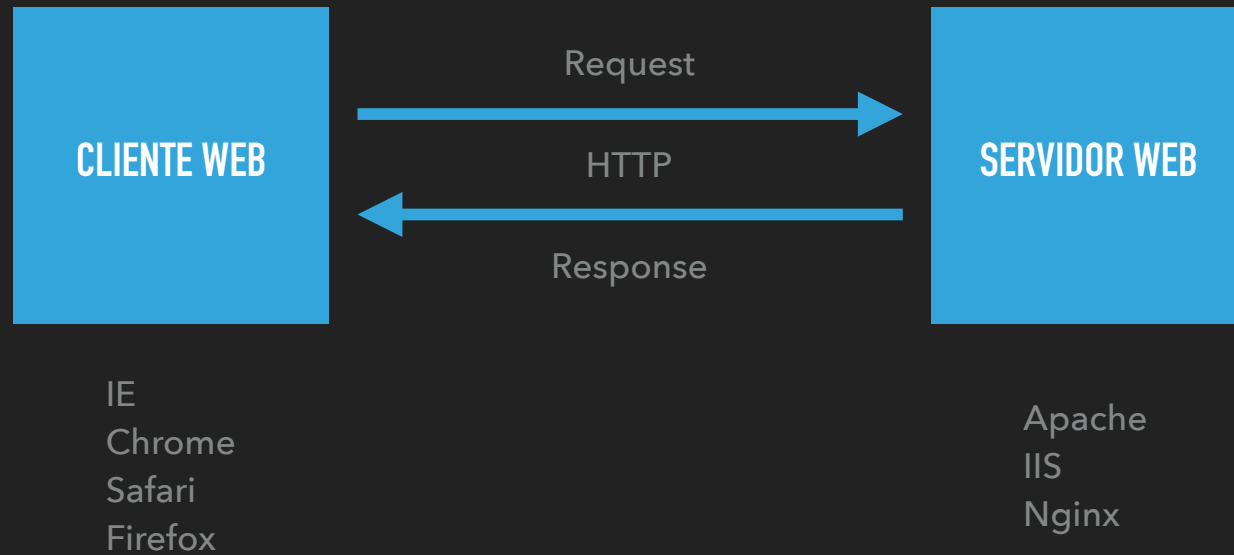
# WEB APPLICATIONS

WEB APPLICATIONS

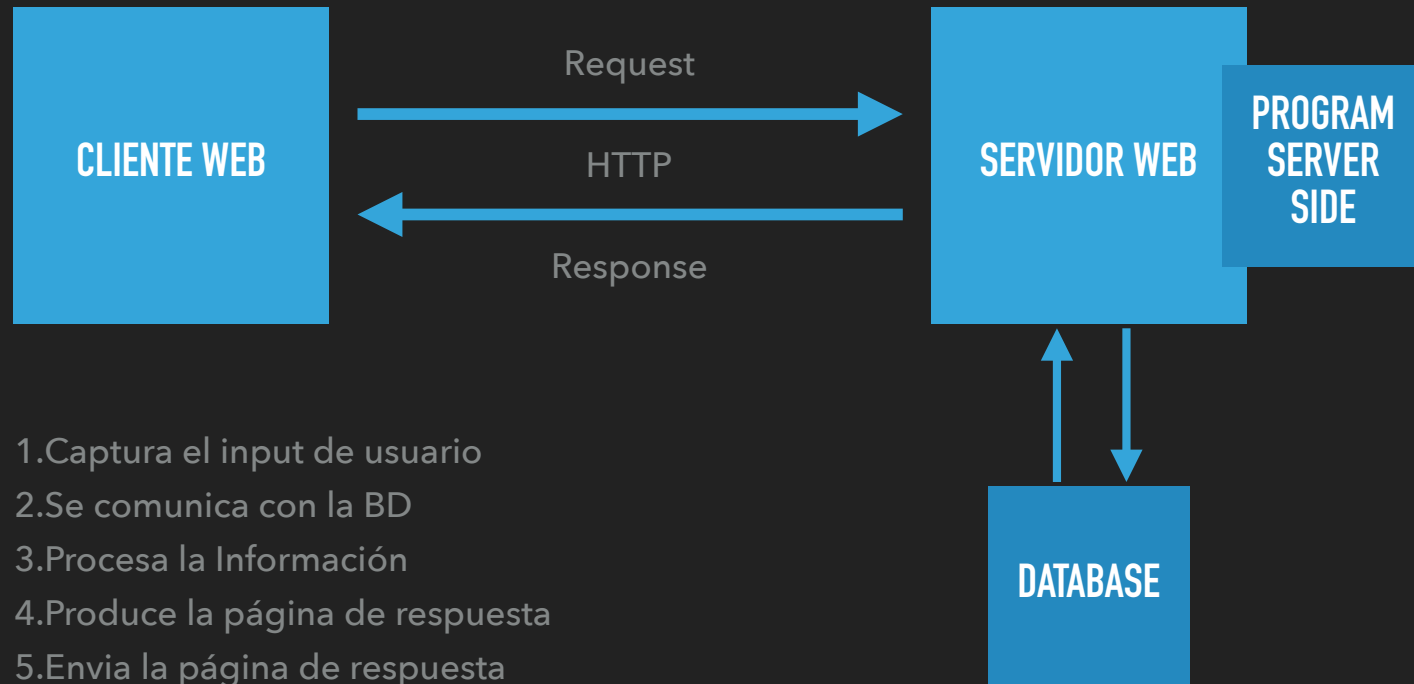
---

# CONCEPTOS BÁSICOS

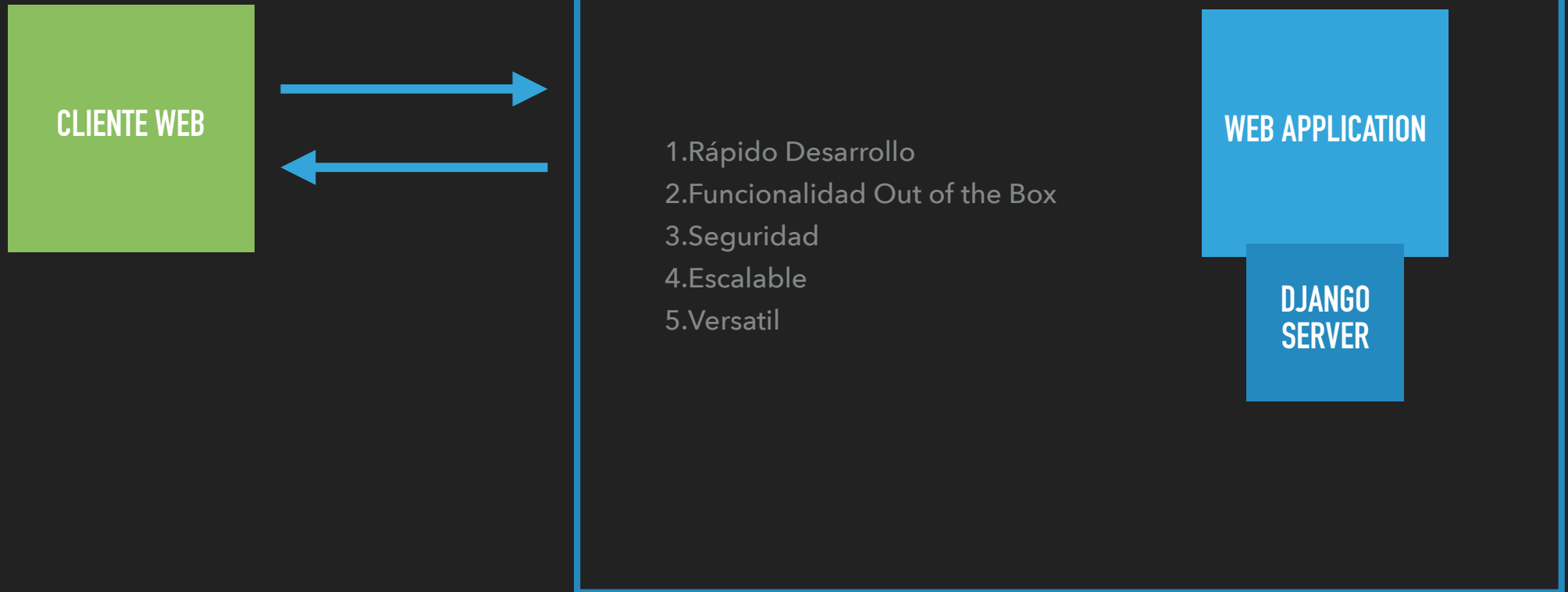
## COMO FUNCIONA EL MODELO CLIENTE / SERVIDOR ?



## BACK END PROGRAMMING



## DJANGO SERVER



## EMPRESAS QUE UTILIZAN DJANGO



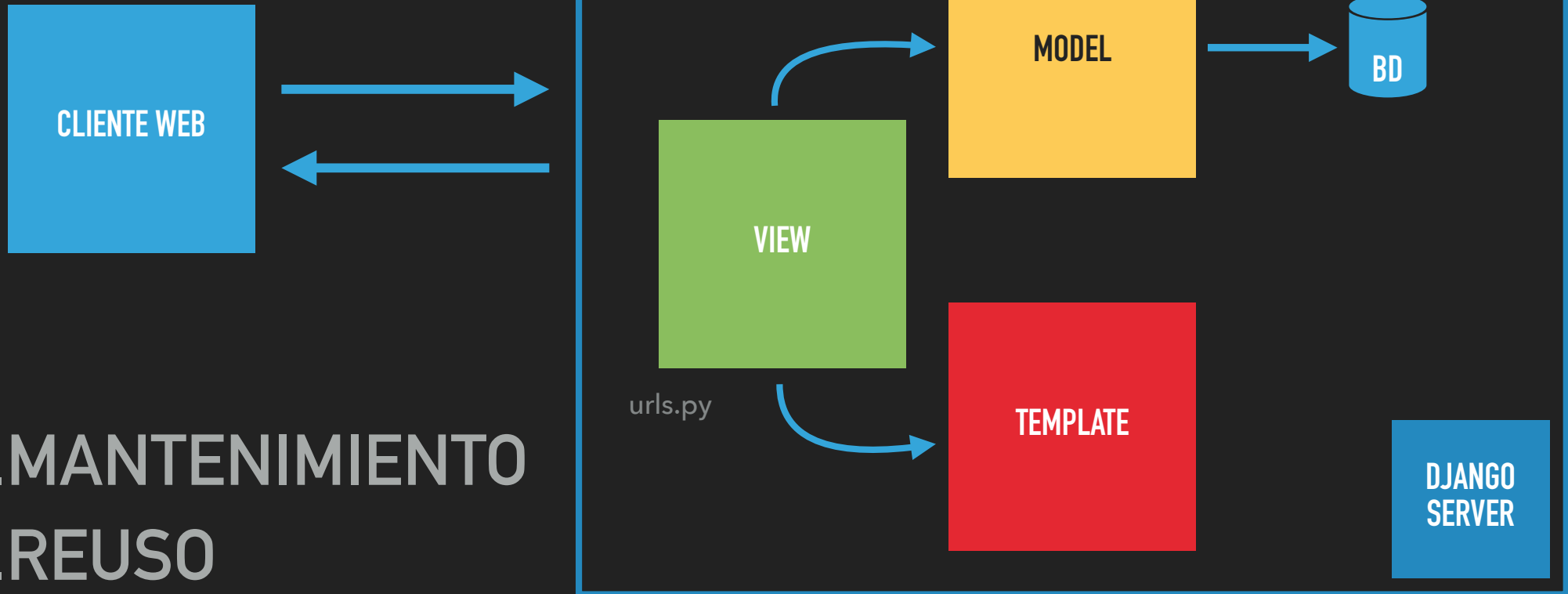
TEXTO

---

**DOCUMENTACIÓN Y SITIO OFICIAL**

**[HTTPS://WWW.DJANGOPROJECT.COM](https://www.djangoproject.com)**

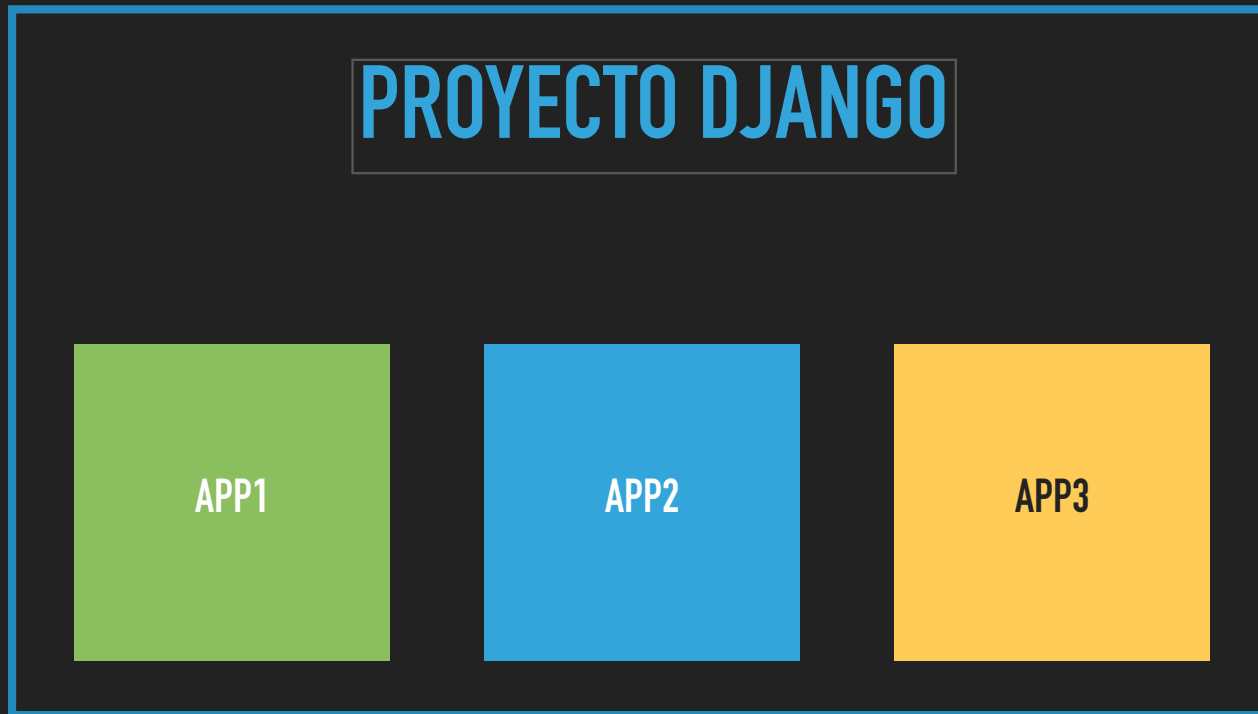
## MODELO MVT



- 1.MANTENIMIENTO
- 2.REUSO



# PROYECTO Y APLICACIONES EN DJANGO



## PREGUNTAS... VERDADERO O FALSO?

- ▶ En el modelo MVT la vista (view) es el encargado de renderizar la interfaz.
- ▶ Un proyecto Django puede tener varias aplicaciones.
- ▶ Enviar el request es parte del trabajo del servidor.
- ▶ El model en el MVT es el encargado de conectar a la BD.

SERVIDOR DJANGO

---

# INSTALACIÓN

# REQUERIMIENTOS

### ▶ Python3

- ▶ <https://www.python.org/downloads/>

### ▶ PIP

- ▶ <https://bootstrap.pypa.io/get-pip.py>

- ▶ `python3 get-pip.py`

### ▶ Django

- ▶ `pip install django`

## REQUERIMIENTOS

```
>_:  
python3 --version  
python3 -m django --version
```

## OTRAS APLICACIONES

```
pip install mysqlclient
```

### ▶ MySQL

▶ <https://www.mysql.com/downloads/>

### ▶ Visual Studio Code

▶ <https://code.visualstudio.com/download>

### ▶ Mysql Workbench

▶ <https://www.mysql.com/products/workbench/>

INSTALANDO DJANGO

---

## CONFIGURAR PYTHON CON MYSQL

```
>_ pip install mysqlclient
```

SERVIDOR DJANGO

---

**HOLA MUNDO**



## CREAR EL PRIMER PROYECTO CON DJANGO

### ► Crear Proyecto:

```
>_
```

```
django-admin startproject holaMundoDjango  
cd holaMundoDjango  
code .
```

#### HOLAMUNDODJANGO

✓ holaMundoDjango

🔗 `__init__.py`

🔗 `asgi.py`

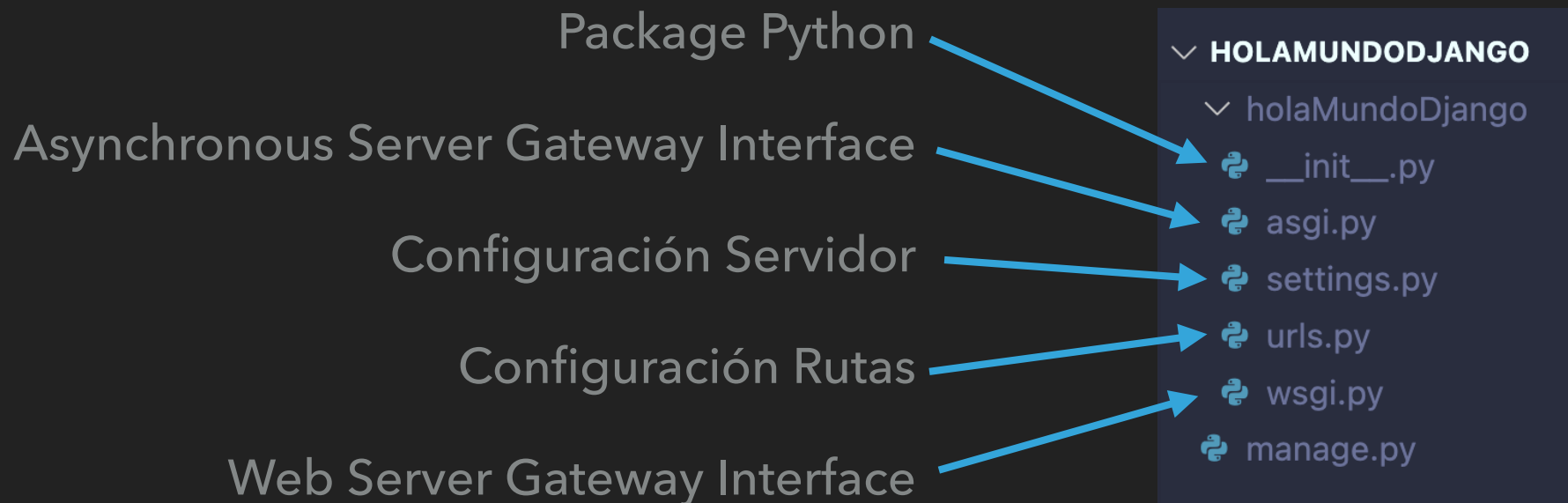
🔗 `settings.py`

🔗 `urls.py`

🔗 `wsgi.py`

🔗 `manage.py`

## CONFIGURAR PYTHON CON MYSQL

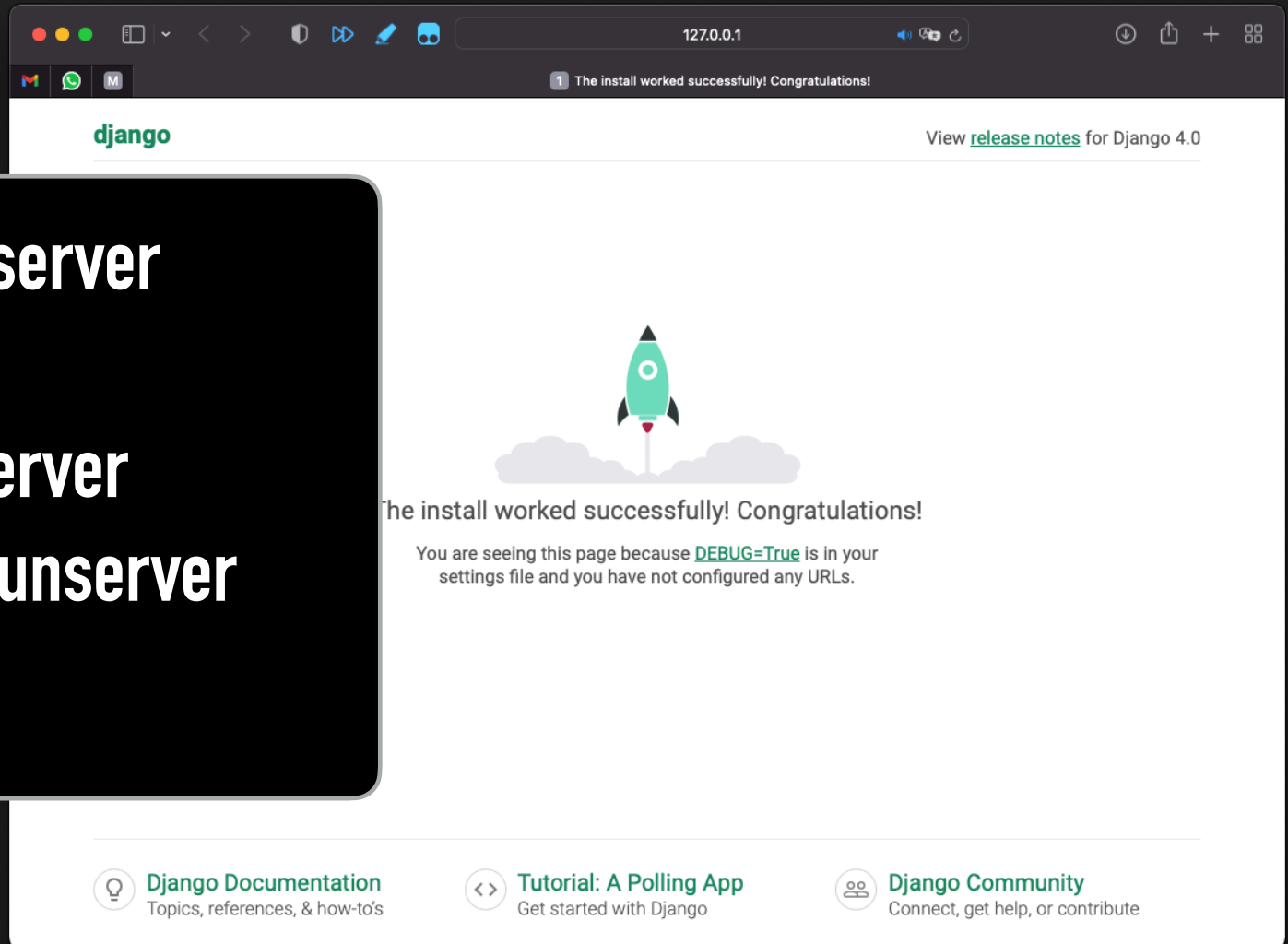


NOTA: Diferencia entre ASGI y WSGI

HOLA MUNDO DJANGO

## LANZAR APP

```
python3 manage.py runserver  
(Alternativas)  
python manage.py runserver  
python.exe manage.py runserver
```



SERVIDOR DJANGO

---

**CREAR UNA APP**

## CREAR NUESTRA PRIMERA APP

- ▶ Agregaremos una aplicación al proyecto

```
python3 manage.py startapp firstApp
```

- ▶ Agregaremos la aplicación a settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'firstApp'  
]
```

## CREAR NUESTRA PRIMERA APP

- ▶ Agregaremos una aplicación al proyecto

```
python3 manage.py startapp firstApp
```

- ▶ Agregaremos la aplicación a settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'firstApp'  
]
```

SERVIDOR DJANGO

---

**VISTAS**

## CREAR VISTAR Y RUTAS

- ▶ **Agregar una vista nueva:**

**Podémos agregar 2 tipos de vistas:**

- ▶ **Vistas basadas en funciones**
- ▶ **Vistas basadas en clases**



## VISTAS BASADAS EN FUNCIONES

- ▶ Agregar una vista nueva (archivo firstApp/views.py):

**Agregamos el import**  
**from django.http import HttpResponse**

```
from django.shortcuts import render
from django.http import HttpResponse
# Create your views here.

def display(request):
    return HttpResponse("<h1>Hola Mundo!</h1>")
```

# VISTAS BASADAS EN FUNCIONES

- ▶ Agregar una ruta para nuestra vista en el archivo urls.py (del proyecto):

```
from firstApp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path("hola/", views.display)
]
```



# VISTAS BASADAS EN FUNCIONES

► Agregaremos una segunda vista y su ruta

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

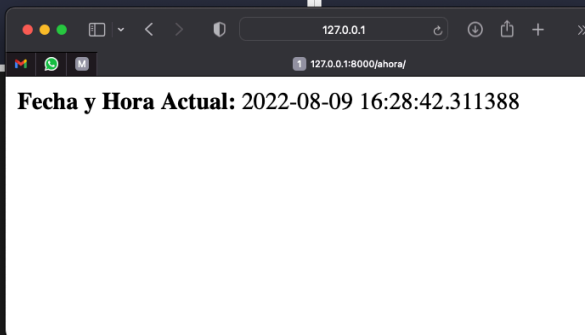
```
def display(request):
    return HttpResponse("<h1>Hola Mundo!</h1>")
```

```
def displayDateTime(request):
    dt = datetime.datetime.now()
    s = "<b>Fecha y Hora Actual: </b>" + str(dt)
    return HttpResponse(s)
```

```
from django.contrib import admin
from django.urls import path
```

```
from firstApp import views
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("hola/", views.display),
    path("ahora/", views.displayDateTime),
]
```



## VISTAS BASADAS EN FUNCIONES

- ▶ Agregamos una segunda aplicación al proyecto

```
python3 manage.py startapp secondApp
```

- ▶ Agregamos la aplicación a settings.py
- ▶ Agregamos una vista en views.py (en secondApp)

# VISTAS BASADAS EN FUNCIONES

- Agregaremos las rutas de una de las siguientes formas

```
from firstApp import views as app1
from secondApp import views as app2

urlpatterns = [
    path('admin/', admin.site.urls),
    path("hola/", app1.display),
    path("ahora/", app1.displayDateTime),
    path("saludo/", app2.saludo),
]
```

```
from firstApp.views import display
from firstApp.views import displayDateTime
from secondApp.views import saludo

urlpatterns = [
    path('admin/', admin.site.urls),
    path("hola/", display),
    path("ahora/", displayDateTime),
    path("saludo/", saludo),
]
```

# RUTAS A NIVEL DE APLICACIONES

- ▶ Podemos tener las rutas a nivel de cada aplicación en vez de una sola para todo el proyecto.
- ▶ Para ello creamos un archivo `urls.py` dentro de cada proyecto y solamente dejamos las rutas que le corresponden.

```
from firstApp.views import display
from firstApp.views import displayDateTime

urlpatterns = [
    path('admin/', admin.site.urls),
    path("hola/", display),
    path("ahora/", displayDateTime),
]
```

```
from django.contrib import admin
from django.urls import path

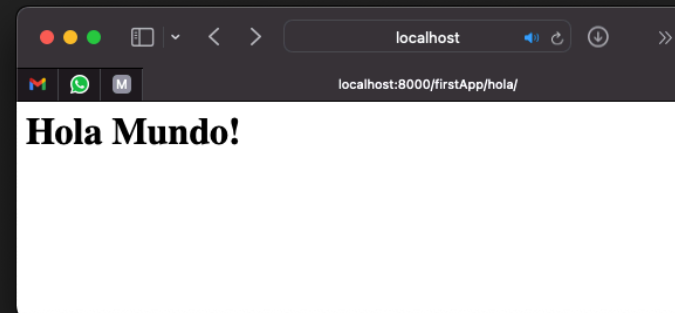
from secondApp.views import saludo

urlpatterns = [
    path('admin/', admin.site.urls),
    path("saludo/", saludo),
]
```

# RUTAS A NIVEL DE APLICACIONES

- Luego en el archivo `urls.py` del proyecto incorporamos las referencias a estos archivos.

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("firstApp/", include('firstApp.urls')),  
    path("secondApp/", include('secondApp.urls')),  
]
```



SERVIDOR DJANGO

---

**ENTORNOS VIRTUALES**



## ENTORNOS VIRTUALES

- ▶ Los entornos virtuales nos permiten tener ambientes separados para distintos proyectos, y así evitar conflictos de versiones entre los distintos paquetes que ocupemos en los proyectos.

## COMANDOS PARA ENTORNOS VIRTUALES

- ▶ `pip3 install virtualenv`
- ▶ `which virtualenv`
- ▶ `python3 -m venv env`
- ▶ `source env/bin/activate`
- ▶ `which python3`
- ▶ `pip3 install Django`
- ▶ `pip3 install mysqlclient`
- ▶ `env/bin/django-admin.py startproject <projectname>`
- ▶ `python3 manage.py migrate`
- ▶ `python3 manage.py runserver`
- ▶ `django`
- ▶ `deactivate`
- ▶ `rm -rf venv`

<https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/>