

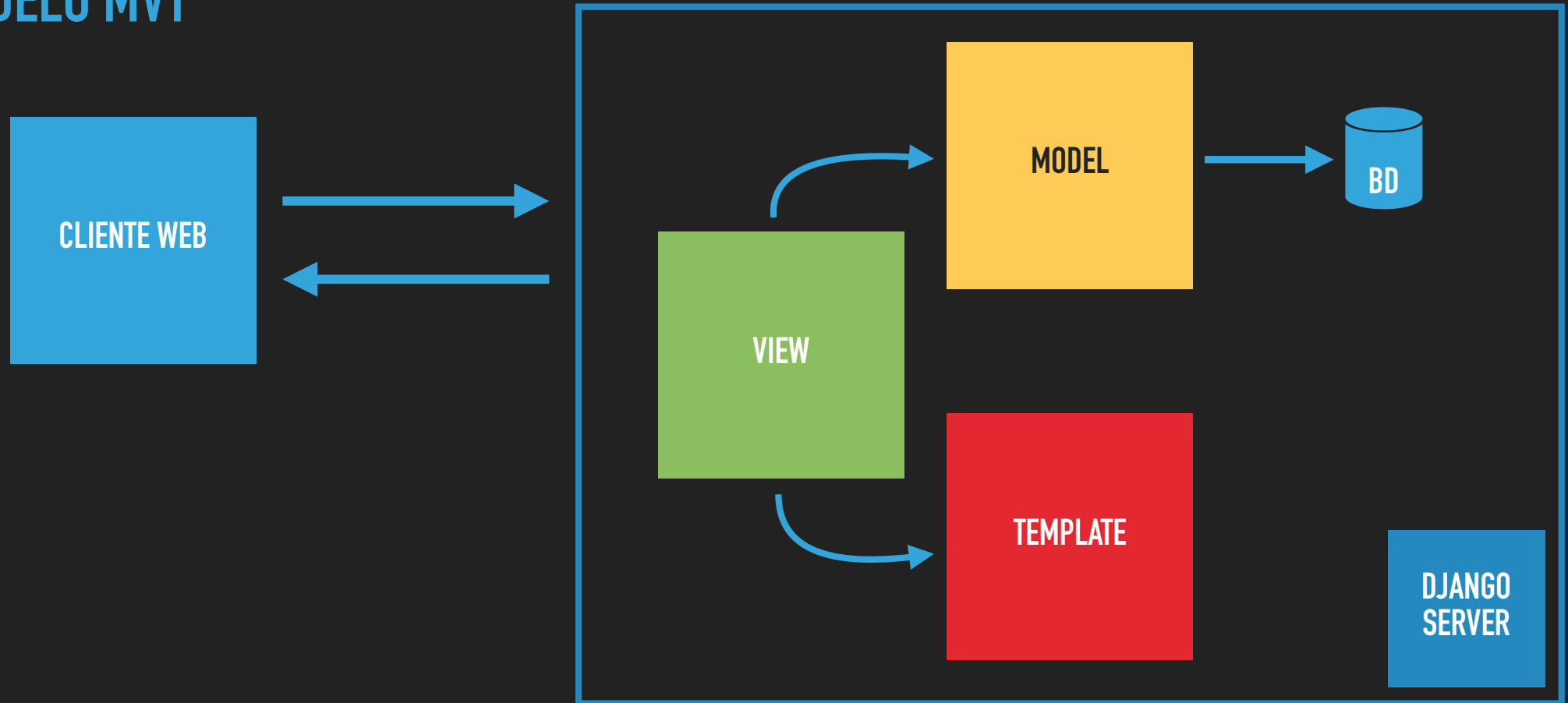
TEMPLATES, MODELOS,

DJANGO

DJANGO APPLICATIONS

TEMPLATES

MODELO MVT



USO DE TEMPLATES EN DJANGO

- ▶ Crear un proyecto y una app.
- ▶ Especificar el directorio de los templates.
- ▶ Crear un template.
- ▶ Crear una vista y mapearla a una url.
- ▶ Templates en acción.

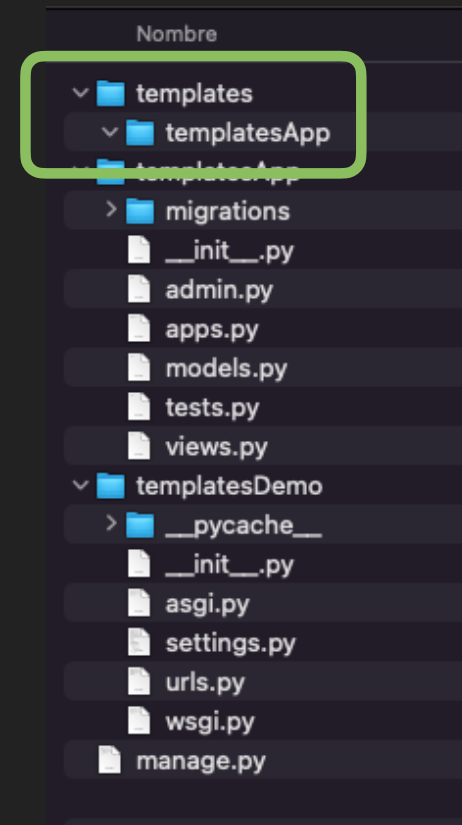
USO DE TEMPLATES EN DJANGO

- ▶ Crear un proyecto y una app.
 - ▶ **`django-admin startproject templatesDemo`**
 - ▶ **`cd templatesDemo`**
 - ▶ **`python3 manage.py startapp templatesApp`**
 - ▶ **`code .`**

USO DE TEMPLATES EN DJANGO

- ▶ Especificar el directorio de los templates.

- ▶ Creamos un directorio “templates”
- ▶ Dentro de ese directorio creamos un directorio por cada app



USO DE TEMPLATES EN DJANGO

- ▶ Agregar la ruta en settings.py
 - ▶ Dentro de `TEMPLATES = []`
 - ▶ `os.path.join(BASE_DIR, 'templates')`
 - ▶ No olvidar `import os`

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'templates')],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```

USO DE TEMPLATES EN DJANGO

- ▶ Crear un template dentro de la carpeta templates/templateApp

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Primer Template</title>

  </head>
  <body>
    <h1>Templates en Acción!</h1>
  </body>
</html>
```


USO DE TEMPLATES EN DJANGO

- ▶ Crear una vista dentro de templatesApp/views.py y la ruta en urls.py

```
from django.shortcuts import render

# Create your views here.
def renderTemplate(request):
    return render(request, 'templatesApp/firstTemplate.html')
```

views.py

```
from templatesApp.views import renderTemplate

urlpatterns = [
    path('admin/', admin.site.urls),
    path('render/', renderTemplate)
]
```

urls.py



<http://127.0.0.1:8000/render/>

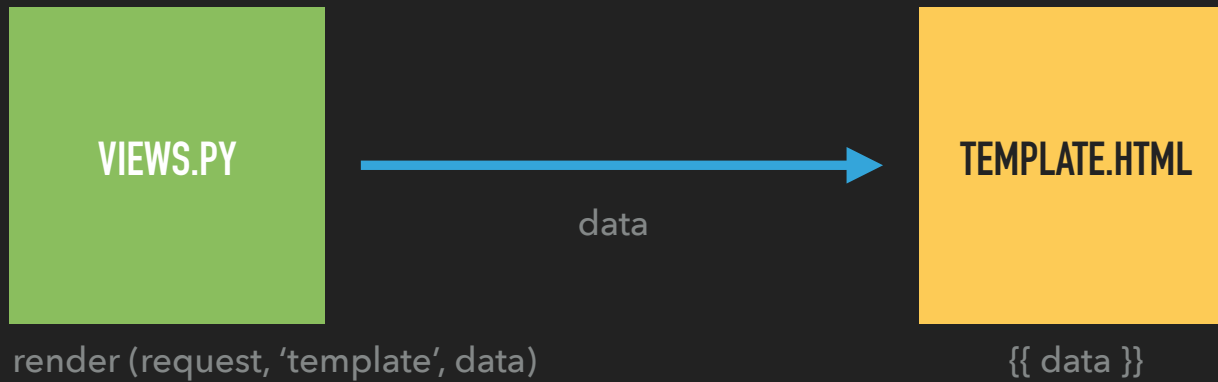
USO DE TEMPLATES TAGS (DTL)

- ▶ `{{ data }}`
- ▶ `{% tagName %}`
 - ▶ `{% csrf_token %}`
 - ▶ `{% x extends y %}`
 - ▶ `{% if %} {% endif %}`
 - ▶ `{% for %} {% endfor %}`

DJANGO APPLICATIONS

TEMPLATES DATA

USO DE TEMPLATES DATA



DJANGO TEMPLATES

USO DE TEMPLATES DATA

```
from django.shortcuts import render

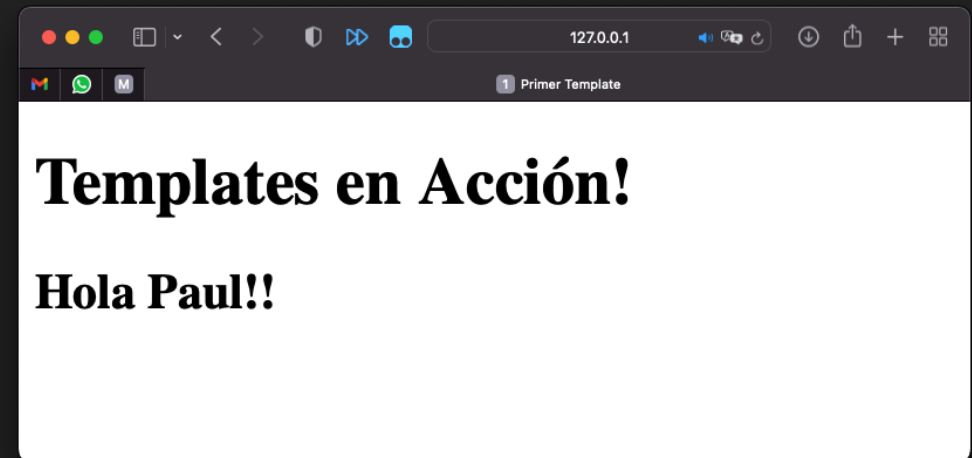
# Create your views here.
def renderTemplate(request):
    data = {"nombre" : "Paul"}
    return render(request, 'templatesApp/firstTemplate.html', data)
```

views.py

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Primer Template</title>

  </head>
  <body>
    <h1>Templates en Acción!</h1>
    <h2>Hola {{nombre}}!! </h2>
  </body>
</html>
```

templatesApp/firstTemplate.html



<http://127.0.0.1:8000/render/>

EJERCICIO TEMPLATES

- ▶ Agregar una vista llamada **infoUsuario** que despliegue la información de un usuario utilizando un template llamado **userInfoTemplate.html** en la ruta <http://127.0.0.1:8000/info/> como se muestra en la imagen.



<http://127.0.0.1:8000/info/>

DJANGO APPLICATIONS

STATIC FILES

STATIC FILES

- ▶ Normalmente haremos uso de archivos estáticos dentro de nuestro proyecto.
- ▶ Generalmente son imágenes, css o archivos js
- ▶ Podemos definir la ruta en donde se almacenarán estos archivos utilizando settings.py

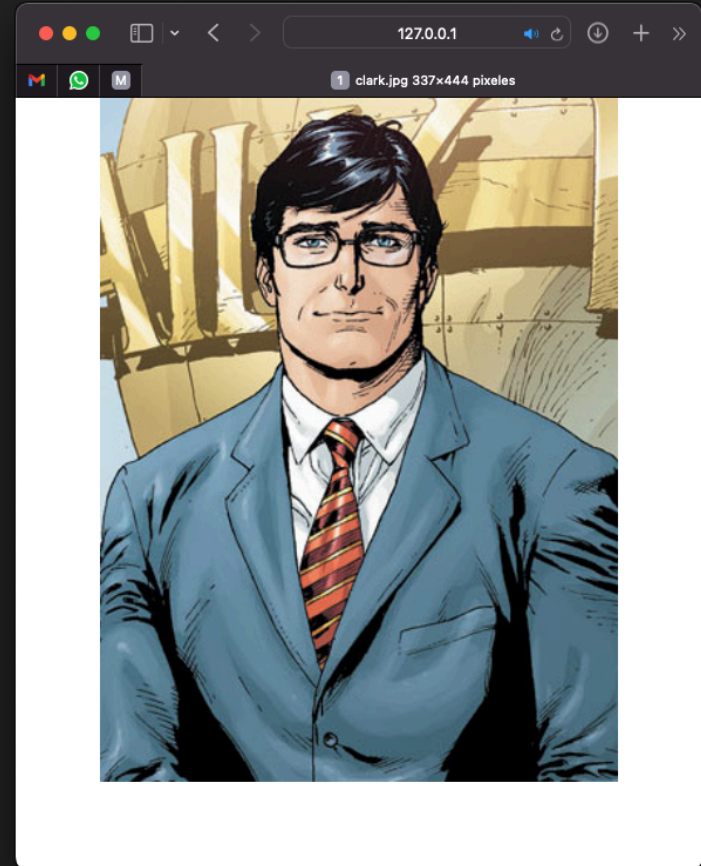
```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static')]
```


STATIC FILES

- ▶ Agreguemos una imagen a **static/images/**
- ▶ Podemos verificar el contenido accediendo a **<http://127.0.0.1:8000/static/images/clark.jpg>**



<http://127.0.0.1:8000/static/images/clark.jpg>

STATIC FILES

- ▶ Para agregar la imagen al template agregamos:

```
<!DOCTYPE html>
{% load static %}

<html>
  <head>
    <meta charset="utf-8">
    <title>Información de Usuario</title>

  </head>
  <body>
    <h1>Información del Usuario</h1>
    
    <ul>
      <li>ID: {{id}}</li>
      <li>Nombre: {{nombre}}</li>
      <li>Email: {{email}}</li>
    </ul>
  </body>
</html>
```

← Incorporamos los archivos estáticos

← Usamos la referencia dentro de la imagen



<http://127.0.0.1:8000/info/>

EJERCICIO STATIC FILES

- ▶ Agregar un archivo css dentro de la carpeta static/css e incorporarlo al proyecto.
- ▶ Agregar algunas reglas de cls para la página de info de usuarios.



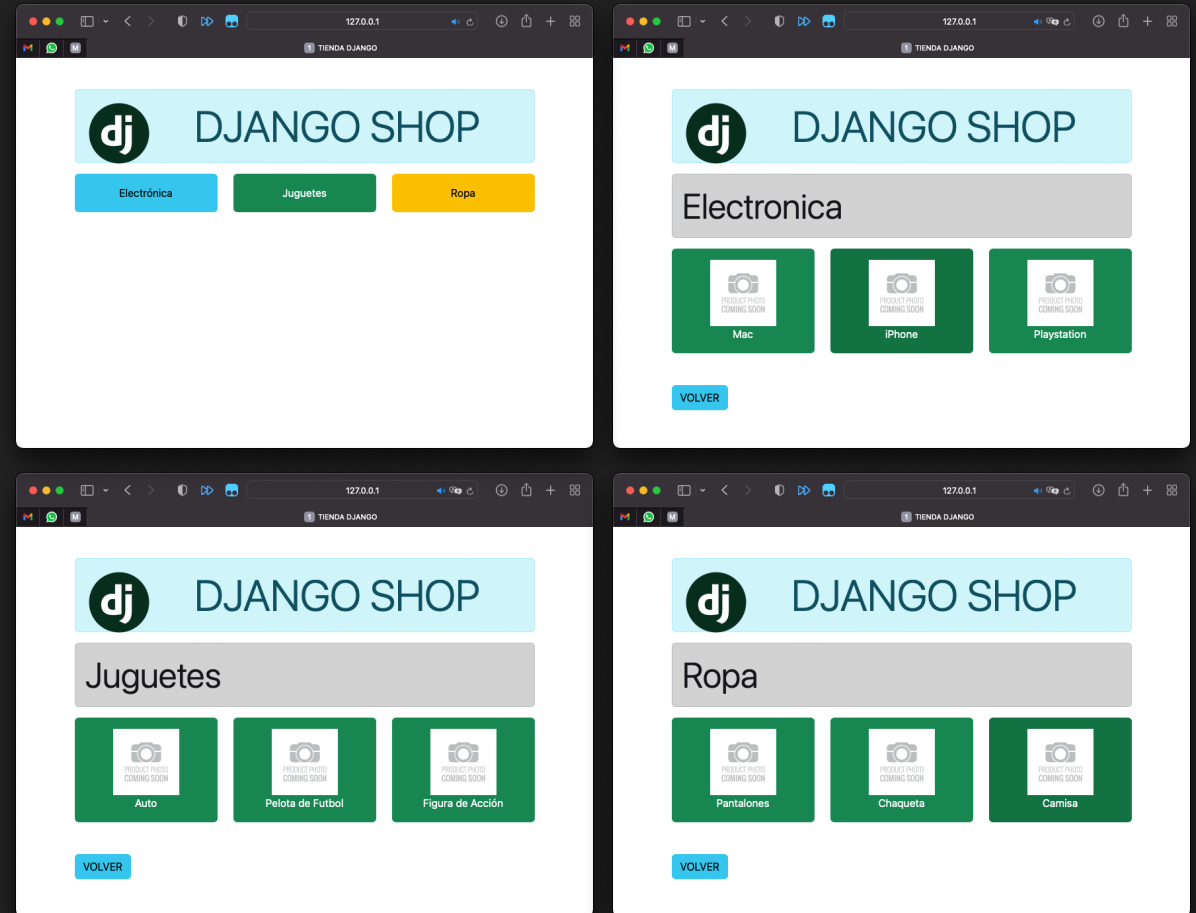
<http://127.0.0.1:8000/info/>

DJANGO APPLICATIONS

CASO DE USO

EJERCICIO PRACTICO TEMPLATES

- ▶ Crear un proyecto llamado ejercicioProductos que contenga una app.
- ▶ Debe tener 2 vistas, una para el índice y otra para los productos.
- ▶ En el índice se deben desplegar accesos a 3 categorías de productos, cada una debe mostrar 3 productos definidos en la vista.
- ▶ Debe incorporar imágenes estáticas (logo y placeholders)

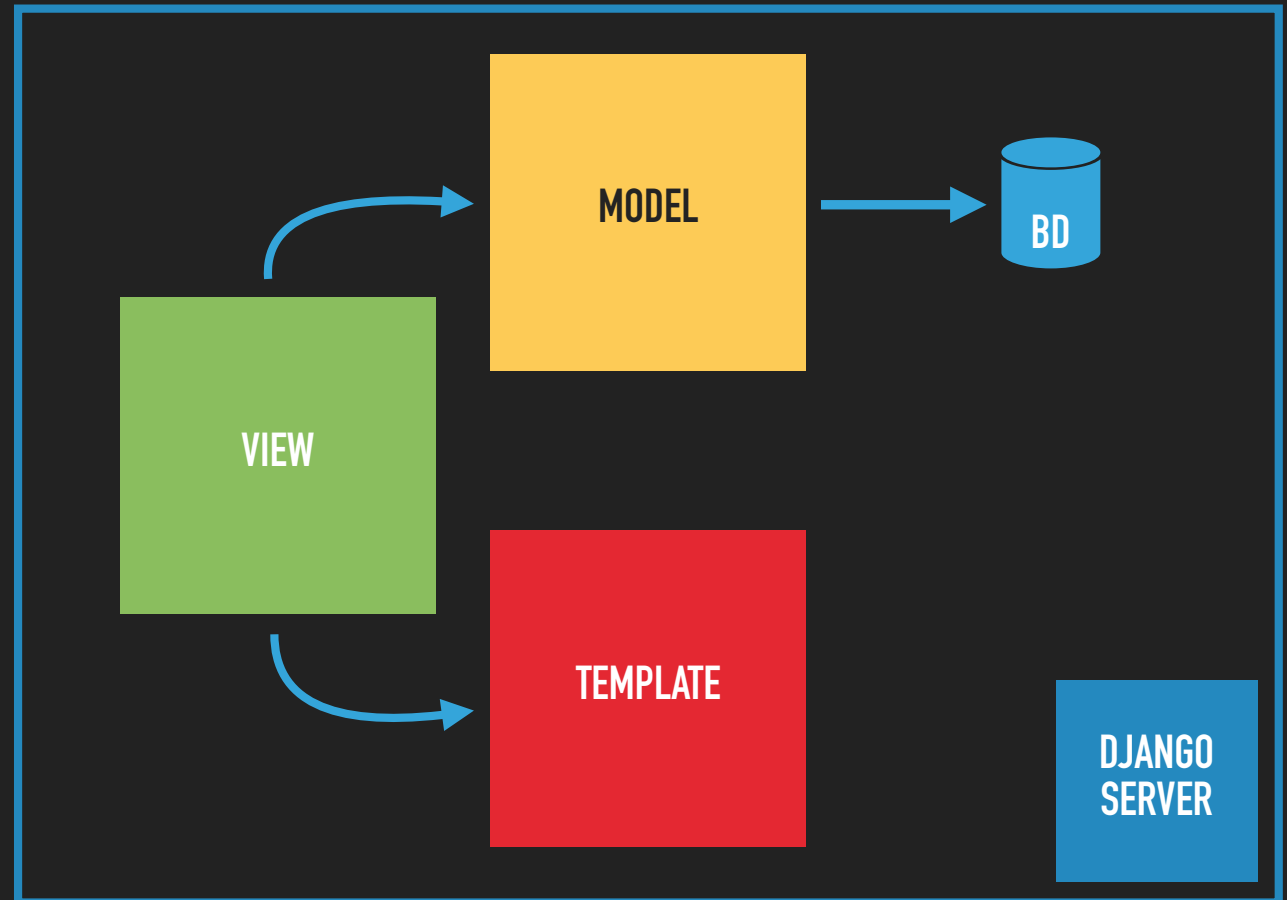


DJANGO APPLICATIONS

MODELS

USO DE MODELOS EN DJANGO

- ▶ Crear el Proyecto
- ▶ Configurar la Base de Datos
- ▶ Crear la clase model
- ▶ Realizar las migraciones
- ▶ Usar el modelo en la vista



CONFIGURAR MYSQL

► Crear la BD en mysql

```
#pip install PyMySQL
import pymysql
pymysql.install_as_MySQLdb()

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'DJANGO_DEMO_EMP',
        'USER': 'root',
        'PASSWORD': 'pass'
    }
}
```

settings.py

CONFIGURAR MYSQL

▶ Crear Migraciones

▶ **`python3 manage.py makemigrations`**

▶ Ejecutar Migraciones

▶ **`python3 manage.py migrate`**

▼ Tables

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- empApp_employee

CREAR MODELS

```
# Create your models here.  
class Employee(models.Model):  
    nombre = models.CharField(max_length=50)  
    email = models.CharField(max_length=50)  
    fono = models.CharField(max_length=15)
```

models.py

<https://docs.djangoproject.com/en/4.0/ref/models/fields/>

AGREGAR MODEL A LAS VISTAS, RUTAS Y TEMPLATES

```
from empApp.models import Employee

# Create your views here.
def employeeData(request):
    empleados = Employee.objects.all()
    data = {'empleados' : empleados}
    return render(request, 'empleados.html', data)
```

views.py

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

from empApp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('empleados/', views.employeeData)
]
```

urls.py

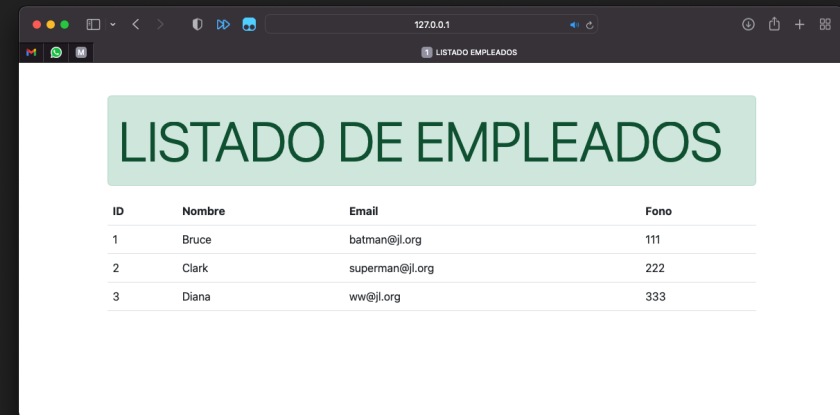
```
<body class="container mt-5">
    <div class="alert alert-success display-1">LISTADO DE EMPLEADOS</div>

    {%if empleados %}

    <table class="table">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nombre</th>
                <th>Email</th>
                <th>Fono</th>
            </tr>
        </thead>
        <tbody>
            {% for emp in empleados %}
            <tr>
                <td scope="row">{{ emp.id }}</td>
                <td>{{ emp.nombre }}</td>
                <td>{{ emp.email }}</td>
                <td>{{ emp.fono }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
    {% else %}
        <div class="alert alert-info">No hay empleados</div>
    {% endif%}
</body>
```

empleados.html

AGREGAR MODEL A LAS VISTAS



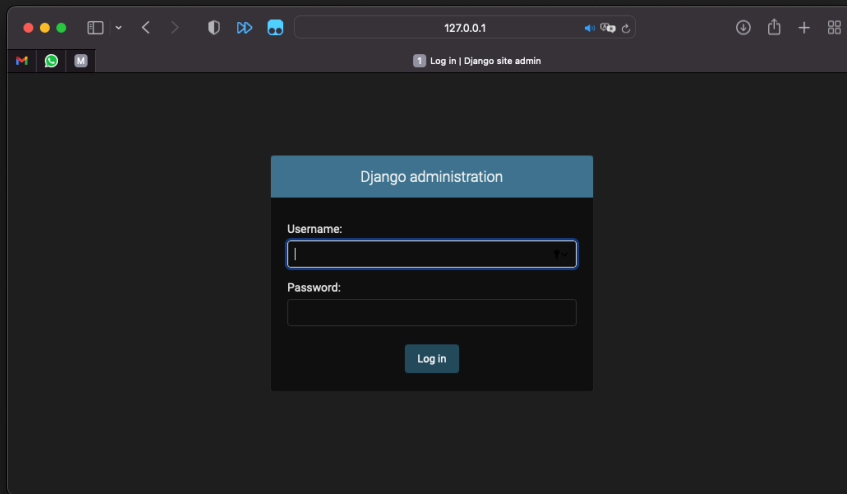
DJANGO APPLICATIONS

DJANGO ADMIN

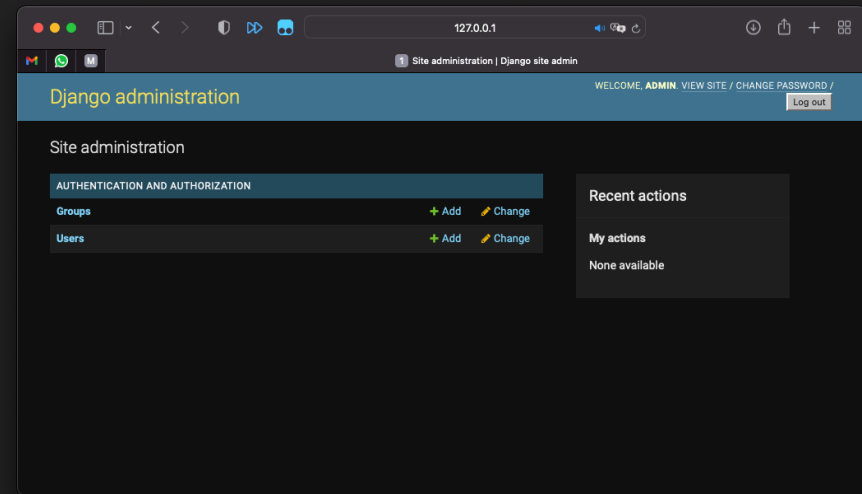
DJANGO MODELS

DJANGO ADMIN

`python3 manage.py createsuperuser`



`http://127.0.0.1:8000/admin/`

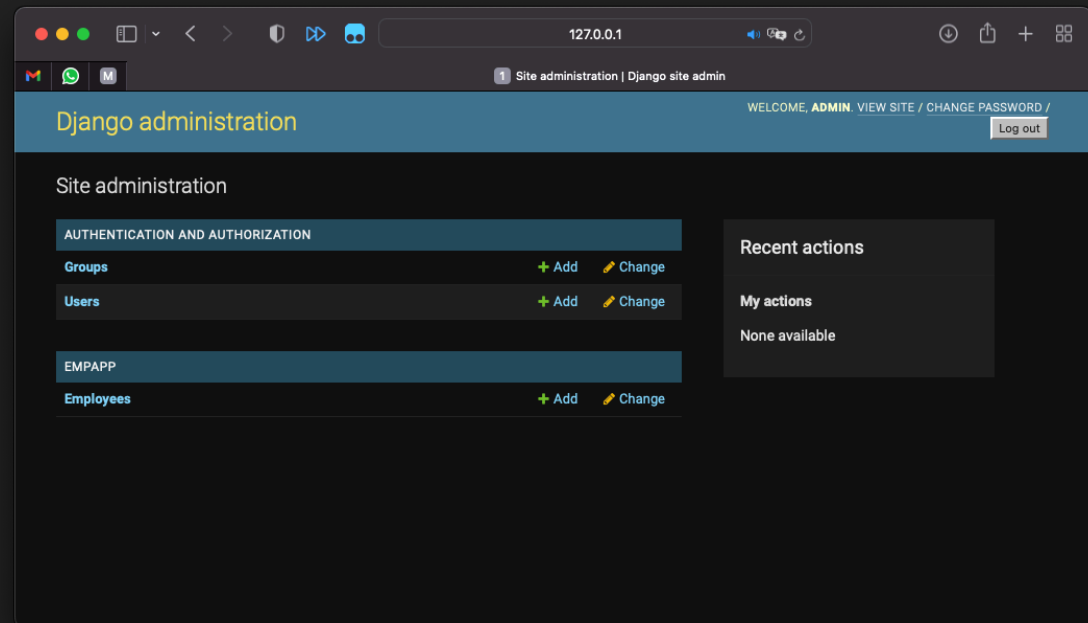


REGISTRAR UN NUEVO MODELO EN DJANGO ADMIN

```
from django.contrib import admin
from empApp.models import Employee

# Register your models here.
admin.site.register(Employee)
```

admin.py



DESPLEGAR ATRIBUTOS EN DJANGO ADMIN

```
from django.contrib import admin
from empApp.models import Employee

class EmployeeAdmin(admin.ModelAdmin):
    list_display = ['nombre', 'email', 'fono']

# Register your models here.
admin.site.register(Employee, EmployeeAdmin)
```

admin.py

