PREGUNTAS

**1.¿Qué comando se utiliza para iniciar un nuevo proyecto en Django? Describe también la estructura básica de carpetas y archivos que se genera al ejecutar este comando.**

El comando para iniciar un nuevo proyecto en Django es:

* django-admin startproject nombre\_del\_proyecto

Este comando crea una nueva carpeta con el nombre del proyecto y genera la estructura básica de carpetas y archivos necesarios para comenzar a trabajar con Django. La estructura básica es la siguiente:

* **Nombre\_del\_proyecto.**
  + **manage.py**: Un script que sirve como punto de entrada para diversas tareas de administración del proyecto, como correr el servidor de desarrollo, migraciones de bases de datos, etc.
  + **nombre\_del\_proyecto:** (subcarpeta con el mismo nombre que el proyecto):
    - **init.py**: Indica que esta carpeta debe ser tratada como un paquete Python.
    - **settings.py**: Archivo de configuración del proyecto (base de datos, aplicaciones instaladas, middleware, etc.).
    - **urls.py**: Define las URL que manejan las diferentes vistas dentro del proyecto.
    - **asgi.py**: Configuración de la interfaz ASGI para la ejecución de servidores asíncronos.
    - **wsgi.py**: Configuración de la interfaz WSGI para la ejecución de servidores web.

**2. Explica en qué consiste el Modelo Vista Controlador (MVC) y cómo se diferencia del Modelo Vista Template (MVT) utilizado en Django.**

El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa una aplicación en tres componentes principales:

* **Modelo**: Maneja la lógica de la aplicación y los datos (base de datos, reglas de negocio).
* **Vista**: Presenta los datos al usuario, normalmente a través de una interfaz de usuario.
* **Controlador**: Interpreta las entradas del usuario, consulta los modelos, y selecciona una vista adecuada para mostrar al usuario.

En Django, se utiliza un patrón similar, llamado **Modelo Vista Template (MVT)**. Aunque es similar a MVC, hay una ligera diferencia:

* **Modelo (Model)**: Igual que en MVC, maneja los datos y la lógica del negocio.
* **Vista (View)**: En MVT, la vista se encarga de la lógica de negocio relacionada con la presentación de datos. La vista en Django no es lo mismo que la vista en MVC.
* **Template**: Corresponde a lo que en MVC se llama "Vista". Es el archivo HTML que se utiliza para renderizar los datos y mostrarlos al usuario.

### 3.Describe los pasos necesarios para configurar y mostrar una imagen estática en una aplicación Django. Incluye la estructura de directorios y el código necesario.

1. **Configurar archivos estáticos**: En el archivo settings.py, debes definir la ubicación de los archivos estáticos:

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

BASE\_DIR / "static",

]

1. **Crear la carpeta de archivos estáticos**: Dentro del directorio principal del proyecto, crea una carpeta llamada static/. Dentro de esta carpeta, puedes organizar tus archivos estáticos como imágenes, CSS y JavaScript.

static/

images/

logo.png

1. **Cargar el archivo estático en la plantilla**: Para usar la imagen en tu plantilla HTML, primero asegúrate de cargar las etiquetas estáticas al inicio del archivo:

{% load static %}

Luego, puedes referenciar la imagen usando:

<img src="{% static 'images/logo.png' %}" alt="Logo">

### 4.¿Cuál es la función de las vistas en un proyecto Django y cómo se relacionan con las URLs? Proporciona un ejemplo de código en Django que defina una vista básica y la asocie con una URL.

Las vistas en Django son funciones o clases que manejan la lógica de la aplicación. Reciben solicitudes HTTP, procesan la información necesaria y devuelven una respuesta HTTP. La función de las vistas es, esencialmente, actuar como intermediarias entre los modelos y las plantillas.

Las vistas están relacionadas con las URLs a través de un archivo urls.py, donde se define qué vista debe ser llamada cuando una URL específica es accedida.

**Ejemplo de código:**

1. **Definir una vista en views.py:**

from django.http import HttpResponse

def saludo(request):

return HttpResponse("¡Hola, mundo!")

1. **Asociar la vista con una URL en urls.py:**

from django.urls import path

from . import views

urlpatterns = [

path('saludo/', views.saludo, name='saludo'),

]

Ahora, cuando el usuario visite http://tu-dominio/saludo/, se ejecutará la vista saludo y se mostrará la cadena "Hola, mundo".

**5.Enumera y describe dos modelos de desarrollo de software diferentes al MVC, mencionando sus principales características y en qué contextos suelen ser utilizados.**

1. **Modelo en Cascada (Waterfall)**:
   * **Características**: Este es un modelo secuencial donde cada fase del desarrollo debe ser completada antes de que la siguiente comience. Las fases típicas incluyen requisitos, diseño, implementación, pruebas, y mantenimiento.
   * **Contextos de uso**: Se utiliza en proyectos con requisitos bien definidos y poco propensos a cambios. Es común en entornos regulados como la industria aeroespacial o médica.
2. **Desarrollo Ágil (Agile)**:
   * **Características**: Se enfoca en la entrega incremental de pequeñas partes de un proyecto. Promueve la colaboración continua con el cliente, la flexibilidad para cambiar requisitos, y ciclos de desarrollo cortos y repetitivos.
   * **Contextos de uso**: Es ideal para proyectos donde los requisitos pueden cambiar con frecuencia o donde se busca una rápida adaptación al mercado. Es popular en el desarrollo de software comercial y startups.