Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Iztapalapa

Actividades 17: Análisis de Diseño

Autor
Martínez Buenrostro Jorge Rafael

Profesor
Eduardo Filemón Vázquez Santacruz

12 de diciembre de 2023

El patrón de diseño MVT (Modelo, Vista, Template) es una variante del patrón de diseño MVC (Modelo-Vista-Controlador) que se utiliza en el desarrollo de aplicaciones web, especialmente en el contexto de frameworks como Django, que es un marco web de alto nivel para el desarrollo rápido de aplicaciones en Python.

A continuación, se describe cada componente del patrón MVT:

Modelo (Model):

El Modelo representa la capa de datos y la lógica de negocios. Se encarga de gestionar y manipular los datos, así como de realizar operaciones relacionadas con la lógica de la aplicación. En Django, los modelos se definen mediante clases Python que representan las tablas de la base de datos y definen la estructura y comportamiento de los datos.

Vista (View):

La Vista es responsable de presentar los datos al usuario y de manejar las interacciones del usuario. En Django, las vistas son funciones o clases que toman una solicitud HTTP y devuelven una respuesta HTTP. Las vistas pueden renderizar plantillas (***templates***) para presentar datos al usuario.

Plantilla (Template):

La Plantilla se encarga de la presentación visual y define la estructura y el formato de la interfaz de usuario. En Django, las plantillas son archivos que contienen código HTML con la capacidad de incrustar expresiones y bloques de control de flujo utilizando la sintaxis de las plantillas de Django. Las plantillas se utilizan para generar el contenido dinámico que se envía al usuario.

El flujo de trabajo típico en un patrón MVT en Django sería el siguiente:

1. El usuario realiza una solicitud HTTP, como visitar una página en un navegador.
2. La solicitud es manejada por una vista en Django, que puede realizar operaciones en el modelo y luego renderizar una plantilla.
3. La plantilla se llena con datos provenientes del modelo y se presenta al usuario como una respuesta HTTP.
4. El usuario ve la interfaz de usuario generada y puede interactuar con ella.
5. Si el usuario realiza acciones que generan nuevas solicitudes, el ciclo se repite.

En resumen, el patrón MVT en Django comparte muchos conceptos con el patrón MVC, pero la nomenclatura es adaptada para reflejar las peculiaridades de los marcos de desarrollo web y la forma en que Django organiza y maneja las capas de una aplicación.