**Documentación del Proyecto**

**1. Tecnología Cliente-Servidor y Elementos Básicos de TCP/IP**

**1.1. ¿Cómo Funciona la Tecnología Cliente-Servidor?**

Imagina la tecnología cliente-servidor como una conversación entre dos personas en una cafetería. Uno es el **cliente** que hace una solicitud, y el otro es el **servidor** que responde.

* **Cliente:** Piensa en tu navegador web, como Chrome o Firefox. Cuando quieres ver una página web, tu navegador (el cliente) hace una solicitud al servidor. Esta solicitud es como decir: “¡Hola, quiero ver la página de inicio de tu sitio web!”
* **Servidor:** El servidor es como el barista en la cafetería. Recibe la solicitud, la procesa (en este caso, busca la información de la página web), y luego envía una respuesta de vuelta al cliente. Esto es como el barista preparando tu café y entregándotelo.

**¿Cómo funciona esto en detalle?**

1. **Solicitud del Cliente:** Tu navegador envía una solicitud al servidor para obtener la página web.
2. **Procesamiento en el Servidor:** El servidor recibe tu solicitud, encuentra la página web solicitada, y la prepara para enviarla.
3. **Respuesta del Servidor:** El servidor envía la página web de vuelta a tu navegador.
4. **Visualización en el Cliente:** Tu navegador recibe la página web y la muestra en tu pantalla.

**1.2. Elementos Básicos de TCP/IP**

Para que esta conversación entre cliente y servidor funcione, usamos un conjunto de reglas llamadas **TCP/IP**. Aquí están los elementos principales:

* **IP (Protocolo de Internet):** Es como la dirección de tu casa. Cada dispositivo en la red (como tu computadora o teléfono) tiene una dirección IP única para que pueda ser encontrado.
* **TCP (Protocolo de Control de Transmisión):** Asegura que los mensajes lleguen completos y en el orden correcto. Es como un mensajero que verifica que cada paquete de información llegue correctamente.
* **UDP (Protocolo de Datagrama de Usuario):** Similar a TCP, pero no garantiza la entrega. Es como enviar una carta sin asegurarte de que llegue a su destino. Se usa para cosas como transmitir video en vivo.
* **HTTP/HTTPS (Protocolo de Transferencia de Hipertexto/Secure):** HTTP es el lenguaje que los navegadores y servidores usan para hablar entre sí. HTTPS es la versión segura, que cifra la información para protegerla.

**1.3. Front-End vs. Back-End**

Cuando hablamos de desarrollo web, nos referimos a dos partes principales:

* **Front-End:** Es lo que ves en tu navegador. Incluye el diseño y la interacción que tienes con el sitio web. Usamos tecnologías como HTML, CSS y JavaScript para construir esto. Es como el escaparate de una tienda, lo que atrae a los clientes.
* **Back-End:** Es como el área de la tienda que no ves, pero donde ocurre toda la magia. Aquí es donde se maneja la lógica, los datos y el funcionamiento interno. Usamos tecnologías como Python (en Django), PHP o Ruby. Es donde se toman los pedidos y se preparan los productos que luego ves en el escaparate.

**2. Capas de Presentación, Negocios y Datos en un Proyecto Web con Django**

**2.1. Capa de Presentación**

La **capa de presentación** es lo que el usuario ve e interactúa. En Django, esto incluye:

* **Plantillas HTML:** Son los documentos que definen cómo se ve una página web. Las plantillas permiten insertar datos dinámicos (como tu nombre en una página de perfil) usando una sintaxis especial de Django.
* **Vistas:** Las vistas en Django son como los chefs que preparan tu comida. Reciben una solicitud, buscan los datos necesarios y pasan esa información a las plantillas para que se muestre correctamente.
* **Formularios:** Manejan la entrada del usuario, como cuando completas un formulario de registro. Verifican que la información ingresada sea correcta y la envían al servidor.

**2.2. Capa de Negocios**

La **capa de negocios** es donde se encuentra la lógica de la aplicación. Aquí es donde se maneja la inteligencia detrás de la web:

* **Modelos:** Son como el catálogo de productos en una tienda. Definen qué tipo de datos almacenamos y cómo se relacionan entre sí. En Django, los modelos representan las tablas de la base de datos y permiten trabajar con esos datos de manera sencilla.
* **Vistas:** Además de manejar la presentación, también ejecutan la lógica del negocio. Por ejemplo, si un usuario agrega un producto al carrito, la vista actualiza la base de datos y calcula el total.
* **Lógica de Aplicación:** Todo el código que realiza cálculos, verificaciones y otras operaciones importantes. Se encuentra en los modelos, vistas o archivos separados según cómo esté estructurado el proyecto.

**2.3. Capa de Datos**

La **capa de datos** es donde se almacenan y recuperan los datos. En Django, esta capa está formada por:

* **Base de Datos:** Es como el almacén donde guardamos toda la información importante. Django usa un sistema llamado ORM para interactuar con la base de datos, lo que simplifica el trabajo con datos.
* **Migraciones:** Son como las actualizaciones de inventario en una tienda. Permiten ajustar la estructura de la base de datos cuando se realizan cambios en los modelos, asegurando que todo esté en orden.