**Descripción básica del funcionamiento de la Tecnología Cliente-Servidor en "Hassber Web".**

Paso a paso:

1. **Inicio del Usuario:**

* El usuario abre su navegador web (Chrome, Firefox, etc.) y escribe la URL del proyecto "Hassber Web".

1. **Solicitud al Servidor:**

* El navegador (cliente) envía una solicitud HTTP al servidor de "Hassber Web".

1. **Procesamiento en el Servidor:**

* El servidor, utilizando Django, recibe la solicitud y determina qué información necesita enviar de vuelta al cliente.

Si es necesario, el servidor interactúa con la base de datos para obtener o modificar datos.

1. **Respuesta del Servidor:**

* El servidor envía una respuesta HTTP al cliente. Esta respuesta puede incluir una página HTML, datos en formato JSON para ser utilizados por JavaScript, etc.

1. **Renderizado en el Cliente:**

* El navegador del cliente recibe la respuesta y muestra el contenido al usuario.
* JavaScript puede manejar interacciones adicionales sin necesidad de recargar la página, gracias a AJAX.

Elementos Básicos de TCP/IP en "Hassber Web".

1. Direcciones IP:

* Cada dispositivo (cliente y servidor) tiene una dirección IP única, permitiendo la comunicación.

2. TCP:

* Protocolo que garantiza que los datos se entreguen correctamente entre el cliente y el servidor.

3. HTTPS:

* Asegura que la comunicación entre el cliente y el servidor esté cifrada y sea segura.

Diferencias entre Front-End y Back-End en "Hassber Web".

**Front-End** (lo que ve el usuario)**:**

1. HTML/CSS:

* HTML estructura la página web.
* CSS define el estilo y apariencia.

2. JavaScript:

* Añade interactividad y manejo de eventos (clics, formularios, etc.).

3. AJAX:

* Permite que el navegador del cliente solicite y reciba datos del servidor sin recargar la página completa.

**Back-End** (Lo que no ve el usuario)**:**

1. Django:

* Gestiona la lógica de la aplicación.
* Maneja la autenticación de usuarios.
* Controla el acceso a diferentes partes de la aplicación.

2. Base de Datos:

* Almacena información como cuentas de usuario, registros de ventas, productos, etc.
* Realiza operaciones como crear, leer, actualizar y eliminar datos.

3. APIs REST:

* Exponen datos del servidor para que puedan ser consumidos por el front-end.

**Resumen Visual.**

1. Cliente (Navegador):

* Usuario -> Navegador -> Solicitud HTTP -> Servidor

2. Servidor:

* Servidor -> Procesa Solicitud -> Interactúa con Base de Datos -> Envía Respuesta HTTP

3. Cliente (Navegador):

* Navegador -> Recibe Respuesta -> Muestra Contenido al Usuario -> Interactividad con JavaScript