Creado por:

Isabel Maniega

Backend

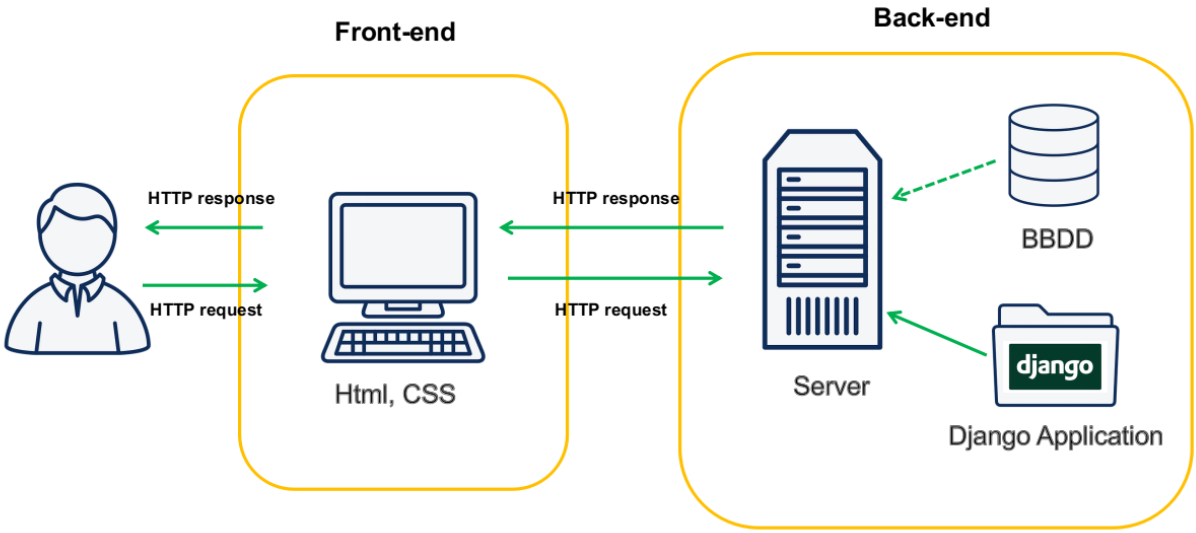
Front-end/ Frontend es la parte de interactua con usuario, pulsar botones, registrar valores, etc...

Back-end /Backend es la parte interna donde se encuentra "cerebro" de la aplicación

Full-Stack es un framework con el cuál podemos realizar una WEB completa con su parte Frontend y su parte Backend

Interfaz es la conexión funcional entre dos sistemas, programas o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles, permitiendo el intercambio de la información.

REST / API-REST es una interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.



Un usuario realiza una petición a través del front (HTTP request) al backend, esa petición es procesada por el Backend y da una respuesta al front (HTTP response)

El **frontend** esta formado normalmente plantillas HTML, diseño usa CSS.

En nuestro caso no vamos a trabajar con ninguno de ellos.

Algunos de los frameworks para diseño web son:

- bootstrap: https://getbootstrap.com/docs/5.2/examples/ (https://getbootstrap.com/docs/5.2/examples/)
- tailwindcss: https://tailwindcss.com/ (https://tailwindcss.com/)

En el caso de **Backend**, tenemos varias alternativas para realizar un backend:

- **Django:** Aplicación Full-Stack, framework desarrollo WEB, gestión de usuario, etc

https://www.djangoproject.com/ (https://www.djangoproject.com/)

https://www.django-rest-framework.org/tutorial/quickstart/ (https://www.django-rest-framework.org/tutorial/quickstart/)

- **Flask:** Framework para desarrollo de Backend.

https://flask.palletsprojects.com/en/2.2.x/ (https://flask.palletsprojects.com/en/2.2.x/)

- **FastAPI:** Framework para desarrollo de Backend, esté está desarrollado para Machine Learning.

https://fastapi.tiangolo.com/tutorial/query-params/ (https://fastapi.tiangolo.com/tutorial/query-params/)

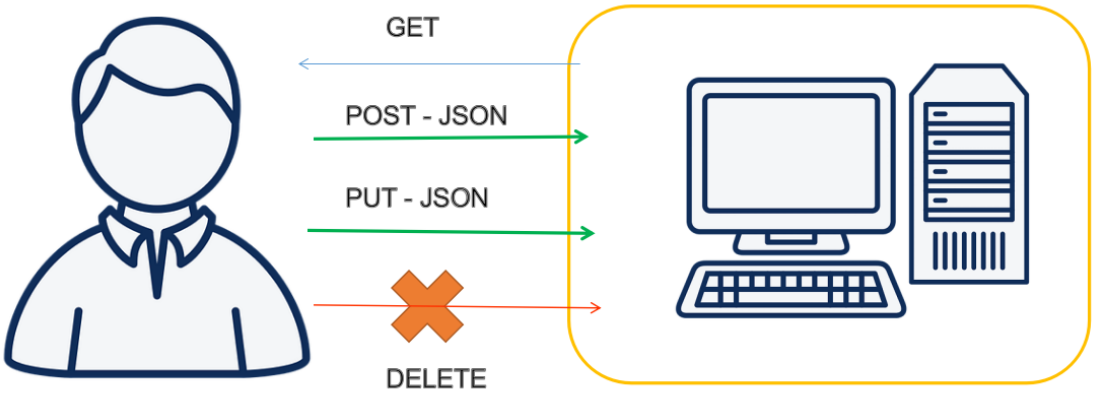
Todo ellos pueden o están conectados a bases de datos (SQL, NoSQL), carpetas con las plantillas de HTML, CSS, imágenes, archivo, etc, estos está ubicado en el SERVIDOR.

Principales métodos HTTP

Recordar que HTTP es un protocolo de comunicación en el cual todos debemos seguir una estructura.

Cuando se realizan las distintas operaciones en HTTP (HTTP request/response), tenemos 4 métodos bien diferenciados:

- **Método GET:** leer o mostrar la información
- **Método POST:** para crear o insertar la información
- **Método PUT:** para actualizar o editar la información
- **Método DELETE:** para eliminar la información



Las principales respuestas por parte del servidor son:

- **1xx:** Información de respuesta. La petición es recibida y se encuentra en progreso
- **2xx:** Satisfactorio. La petición ha sido recibida y aceptada satisfactoriamente.
- **3xx:** Redirección. Pasos adiccionales deben ser realizados en orden de petición.
- **4xx:** Error del cliente. Se realiza una mala sintaxis o una mala petición.
- **5xx:** Error del Servidor. La respuesta ha sido fallida e invalidada la petición.

Ejemplos más usados:

- **200 OK:** solicitud se ha realizado con éxito.
- **201 Created:** solicitud fue creada con éxito y creado un nuevo recurso.
- **400 Bad Request:** sintaxis inválida.
- **404 Not Found:** el servidor no encuentra el contenido solicitado.
- **500 Internal Server Error:** error en el servidor.

Creado por:

Isabel Maniega