**Peticiones HTTP**

Las **peticiones HTTP** son la forma en que los navegadores y otros clientes web se comunican con los servidores. Cuando abres un sitio web, tu navegador envía una petición HTTP al servidor donde está alojado ese sitio. El servidor responde con los datos solicitados, como el contenido de la página web.

**Conceptos clave**

* **URL (Uniform Resource Locator):** Es la dirección que usas para acceder a un recurso en la web.

**Ejemplo:** [**https://www.ejemplo.com:80/pagina**](https://www.ejemplo.com:80/pagina)**.**

* **Dominio:** Se refiere a una dirección legible por humanos que se utiliza para identificar un sitio web o un recurso en Internet. Es parte de la URL que se usa para acceder a sitios web y otros recursos en la web. **Ejemplo: ejemplo.com**.

**El dominio está compuesto por**

* + **Nombre de dominio**: La parte principal del dominio, que en el ejemplo es ejemplo.
  + **Dominio de nivel superior (TLD)**: La extensión del dominio que sigue al nombre principal, como .com, .org, .net, etc. En el ejemplo, el TLD es .com.
* **Puerto:** Especifica el canal de comunicación. Por defecto, HTTP usa el puerto 80 y HTTPS usa el 443. Si no se especifica, se asume el puerto por defecto.

Se refiere a un número que identifica una aplicación o servicio específico en un servidor. Los puertos ayudan a dirigir el tráfico de red a la aplicación correcta dentro de un servidor. Cada aplicación en un servidor puede escuchar en un puerto diferente, permitiendo que múltiples servicios coexistan en una sola dirección IP. **Ejemplo**: En la URL http://www.ejemplo.com:8080/path/to/resource, **el número de puerto es 8080**.

**Tipos de peticiones**

1. **GET:** Solicita datos del servidor. Por ejemplo, cuando visitas una página web, tu navegador hace una petición GET para obtener el contenido de esa página.
2. **POST:** Envía datos al servidor para crear o actualizar recursos. Por ejemplo, cuando envías un formulario en un sitio web, se hace una petición POST con los datos del formulario.
3. **PUT:** Se utiliza para actualizar un recurso existente o crear uno nuevo en una ubicación específica si no existe. Cuando envías un recurso completo para reemplazar el existente. Por ejemplo, si tienes un usuario y quieres actualizar todos sus datos, enviarías todos los campos de ese usuario.
4. **DELETE:** Solicita la eliminación de un recurso en el servidor.
5. **PATCH:** Se utiliza para actualizar parcialmente un recurso existente. Cuando solo necesitas cambiar algunos atributos de un recurso sin afectar el resto de sus datos. Por ejemplo, si solo quieres actualizar la edad de un usuario, solo envías ese campo.

**Estructura de una petición HTTP**

* **Método:** Indica qué acción se desea realizar (GET, POST, etc.).
* **URL:** La dirección del recurso que se está solicitando. Puede incluir el camino del recurso y, opcionalmente, una cadena de consulta.
* **Cabeceras (Headers):** Las cabeceras son pares clave-valor que proporcionan información adicional sobre la petición. Se utilizan para especificar detalles como el tipo de contenido, la autenticación, el idioma y más cada cabecera se escribe en una línea separada
* **Cuerpo (Body):** Contenido que se envía con algunas peticiones. No todas las peticiones HTTP tienen un cuerpo, pero es común en métodos como POST, PUT y PATCH, donde se envían datos al servidor. l cuerpo puede contener datos en varios formatos, siendo JSON uno de los más utilizados.