

Notas HTTP

Introducción al protocolo HTTP

- **HTTP**: Protocolo de Transferencia de Hipertexto, operando en la capa de aplicación para comunicación cliente-servidor.
- Basado en el modelo **petición-respuesta**, es **sin estado (stateless)**, lo que implica que no guarda información de conexiones previas.
- Se utiliza principalmente con **TCP/IP** aunque es compatible con otras capas de transporte confiables.

Evolución del protocolo

- **HTTP/0.9 (1991)**: Primera versión.
- **HTTP/1.0 (1996)** y **HTTP/1.1 (1997)**: Se introducen cabeceras y conexión persistente.
- **HTTP/2 (2015)**: Basado en SPDY, mejora latencia.
- **HTTP/3 (2018)**:
 - Usa QUIC en lugar de TCP, más rápido y elimina bloqueos de cabecera (head-of-line blocking).
 - Amplia adopción en navegadores modernos.

URL (Uniform Resource Locator)

- **Formato**:
`scheme://[userinfo@]host[:port]/path[?query][#fragment]`.
 - Ejemplo:
`http://google.es:8080/ejemplo/index.html?q=text#something`.
- Relación con URI (Uniform Resource Identifier) y URN (Uniform Resource Name):
 - URI es un término general, mientras que URL y URN son tipos específicos.

Métodos HTTP

- **GET**: Recupera recursos. No modifica el servidor.
- **POST**: Envío de datos para procesar. Usado en formularios.
- **PUT**: Sustituye o crea un recurso.
- **DELETE**: Elimina un recurso.
- Otros: **HEAD**, **OPTIONS**, **PATCH**, **TRACE**, **CONNECT**.
- Clasificación:
 - **Safe methods**: No alteran el estado del servidor (ej.: GET, HEAD).
 - **Idempotent methods**: Resultados consistentes en múltiples peticiones (ej.: GET, PUT).

Estructura de una petición HTTP

- Incluye:
 - Método (GET, POST, etc.).
 - URL del recurso.

- Versión de HTTP.
- Cabeceras y cuerpo opcional.

Ejemplo de GET:

```
GET /en/html/dummy?name=MyName&male=yes HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
```

Respuesta HTTP

- Estructura:
 - **Versión:** Versión del protocolo.
 - **Código de estado:** Indica resultado (ej.: 200 OK, 404 Not Found).
 - **Cabeceras:** Información sobre la respuesta.
 - **Cuerpo:** Datos del recurso solicitado.
- **Categorías de códigos de estado:**
 - **1xx:** Informativos (ej.: 100 Continue).
 - **2xx:** Éxito (ej.: 200 OK).
 - **3xx:** Redirección (ej.: 301 Moved Permanently).
 - **4xx:** Error del cliente (ej.: 404 Not Found).
 - **5xx:** Error del servidor (ej.: 500 Internal Server Error).

HTTPS

- Versión segura de HTTP utilizando **SSL/TLS** para cifrar datos.
- Opera en el puerto 443.

Herramientas y ejercicios

- **Postman:** Herramienta para probar y analizar peticiones HTTP.
- Uso práctico: Inspeccionar formularios y mensajes HTTP con herramientas de navegador como **Chrome DevTools**.