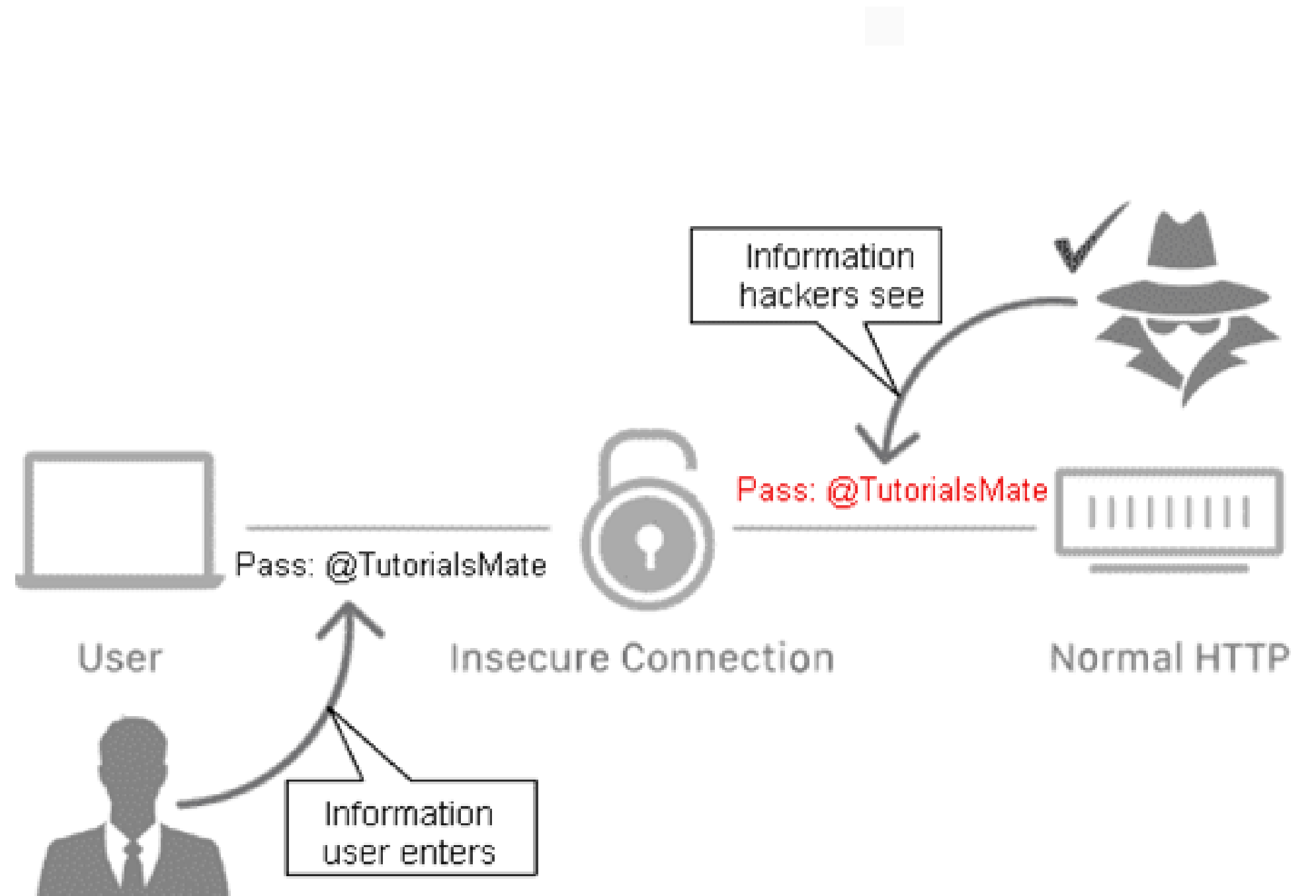




Aplicaciones WEB

HTTP

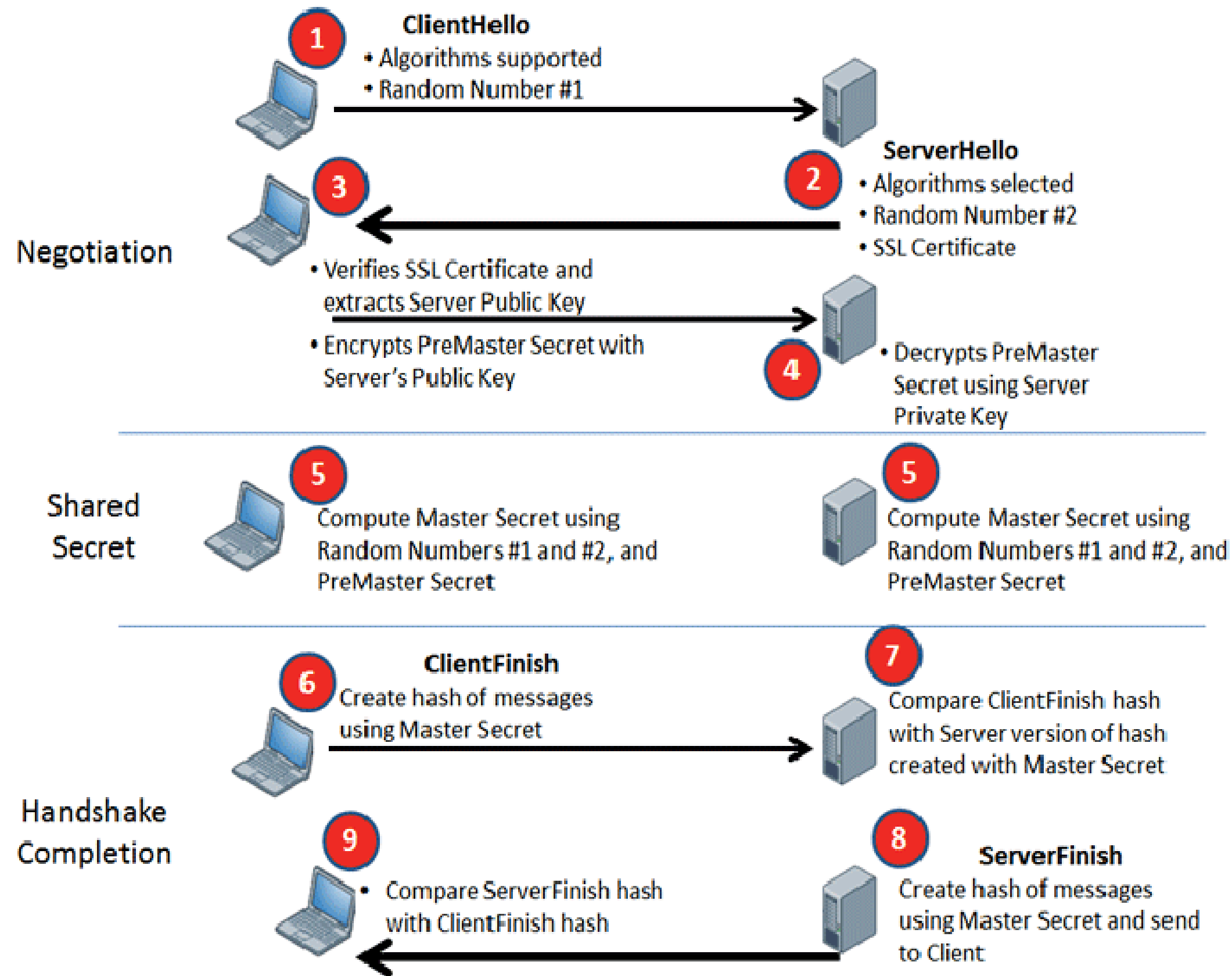
- Llamado “HyperText Transfer Protocol” o protocolo de Transferencia de Hipertexto.
- Base de cualquier intercambio de datos en la Web.
- Evolución:
 - HTTP 0.9 1991 – 1996
 - HTTP 1.0 1996 1999
 - HTTP 1.1 1999 – 2000
 - HTTP 1.2 2000 2015
 - HTTP 2.0 (HTTP/2) 2015 2018.
 - HTTP 3.0 2018 ????
- Incorporación de la seguridad
 - **HTTPS** 1994.
- Modelo Cliente/Servidor.
 - **Cliente**: Solicita los datos y los recibe del Servidor.
 - **Servidor**: Proporciona los datos al cliente.



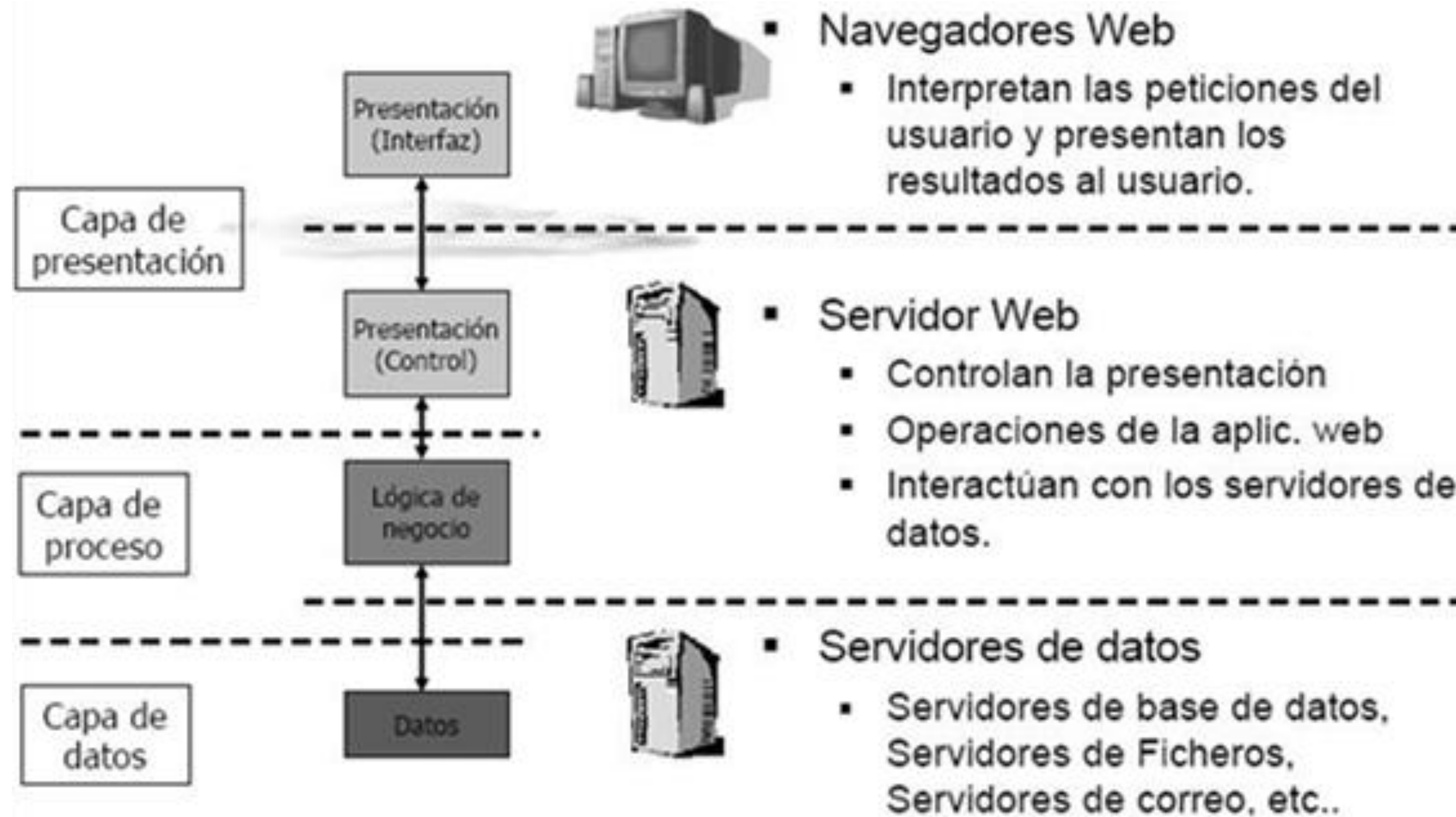
HTTP vs HTTPS

HTTP	HTTPS
<ul style="list-style-type: none">• HTTP stands for 'HyperText Transfer Protocol'.	<ul style="list-style-type: none">• HTTPS stands for 'HyperText Transfer Protocol Secure'.
<ul style="list-style-type: none">• HTTP works at the application layer.	<ul style="list-style-type: none">• HTTPS works at the transport layer.
<ul style="list-style-type: none">• The default port number is 80, for communication.	<ul style="list-style-type: none">• Here, the default port number is 443.
<ul style="list-style-type: none">• No encryption is present in HTTP websites.	<ul style="list-style-type: none">• Both encryption and decryption exist on HTTPS websites.

HTTPS – Handshake SSL



Arquitectura



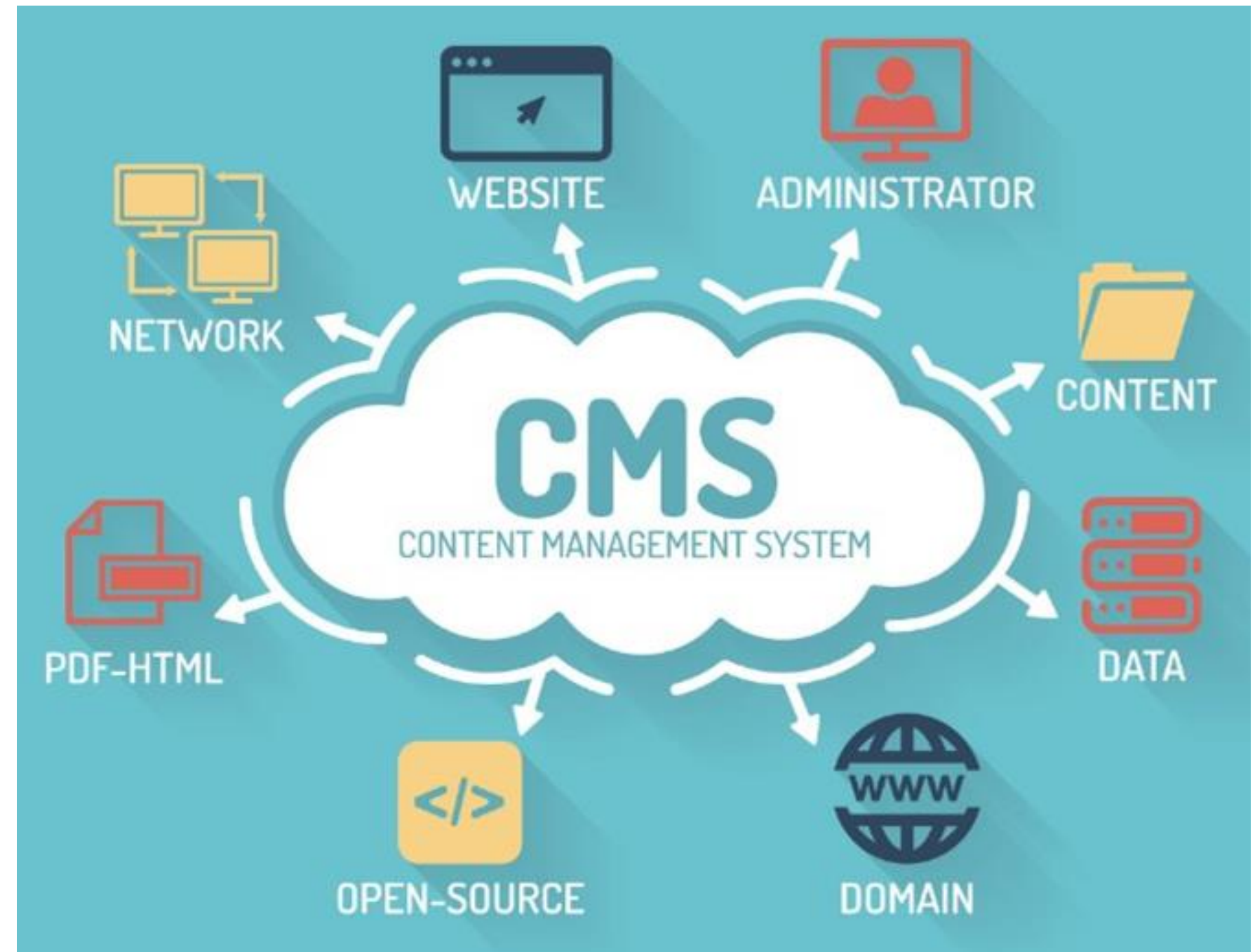
Aplicaciones WEB



<https://flatironschool.com/blog/front-end-vs-back-end-development>

CMS

- Content Management System, es un software de creación y gestión de páginas web.
- Una página web está formada por código, que no es más que una serie de caracteres escritos en un lenguaje de programación: HTML, PHP, Javascript...
- Para crear una página se puede escribir ese código desde cero o «apoyarse» en unas de estas plataformas que dan gran parte de ese código ya escrito.
- Ofrecen un panel muy fácil de usar (backend) para gestionar todo lo que se muestra en la web (frontend).
- Eso significa que, para montar una web con un CMS como WordPress, seguramente el más conocido, no es necesario tener conocimientos de programación.
- Los CMS trabajan con plantillas o themes que aportan el diseño, colores y estilo (css) a la web.



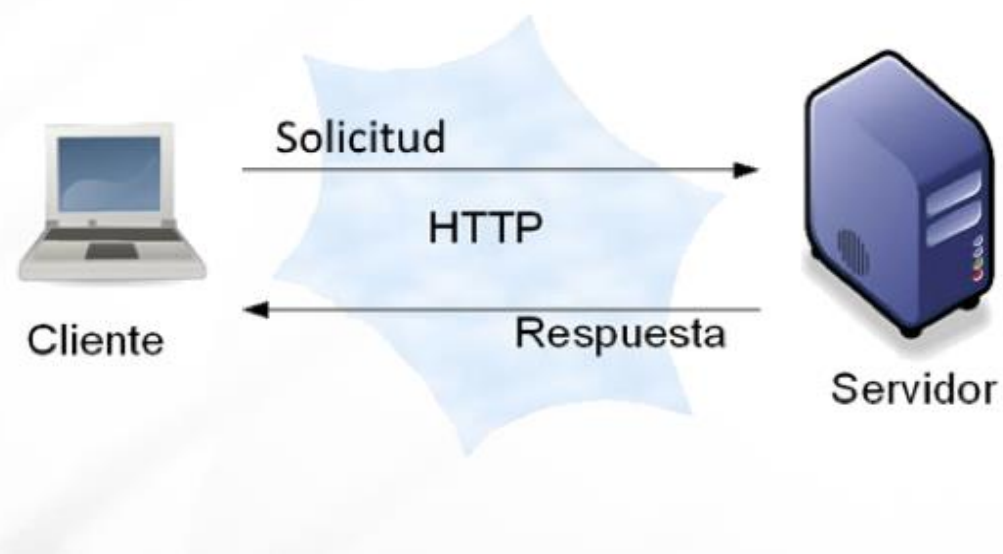
Como son los mensajes

MENSAJE DE PETICION (CLIENTE)

GET / HTTP/1.1

Host: www.eltiempo.es

Accept-Language: es



MENSAJE DE RESPUESTA (SERVIDOR)

HTTP/1.1 200 OK

Date: Sat, 09 Oct 2020 14:28:02 GMT

Server: Apache

Last-Modified: Tue, 01 Dec 2019 20:18:22 GMT

Accept-Ranges: bytes

Content-Length: 32178

[LINEA VACIA]

Content-Type: text/html <!DOCTYPE html...

(aquí empiezan los 32178 bytes de la web solicitada)

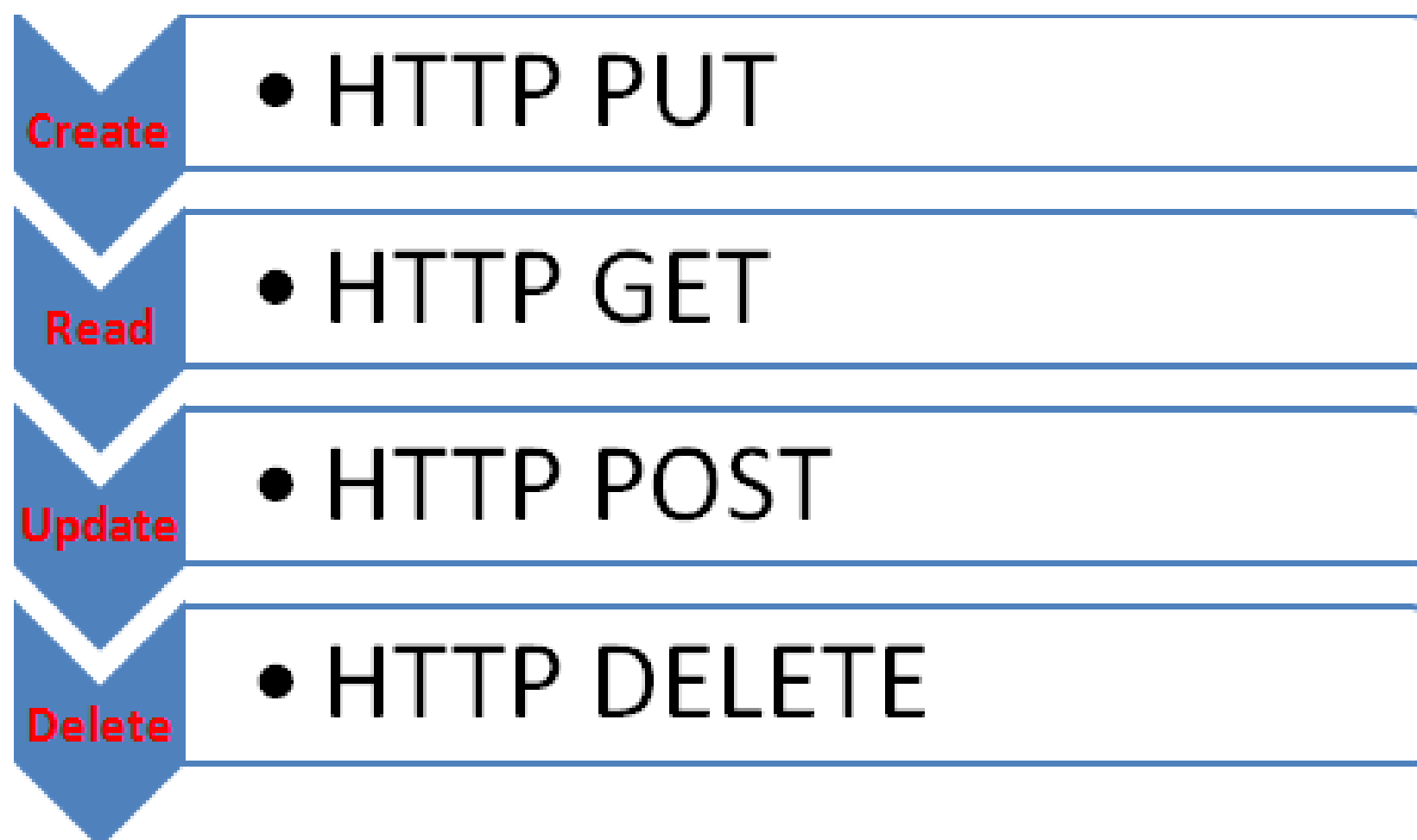
HTTP Métodos

- **PUT**

- Crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición.
- Si no existe y se crea de forma satisfactoria, entonces el servidor envía el código **201 (Created)** .
- Si el elemento existe y es modificado de forma satisfactoria, entonces el servidor de origen debe enviar una respuesta **200 (OK)** o una respuesta **204 (en-US) (No Content)**
- Ejemplo
 - **PUT /nuevo.html HTTP/1.1**

- **GET**

- Obtiene información del servidor, trayendo los datos que están en el servidor al cliente.
- Si la petición es correcta, tendremos en la línea el código **200 OK**.
- Si la petición es errónea, tendremos un mensaje de error del servidor especificado (**404 Not Found, 400 Bad Request...**)
- Ejemplo:
 - **GET /test/web.html HTTP/1.1**



HTTP Métodos

• POST

- Envía información desde el cliente para actualizar o añadir en el servidor.
- Habitualmente se utilizan formularios.
- Si la petición es correcta, el mensaje será **201 Created**
- Si la petición es errónea, tendremos en la línea de estado el mensaje de error del servidor especificado (**404 Not Found**, **400 Bad Request...**)
- Ejemplo: **POST /formulario_post.php HTTP/1.1**

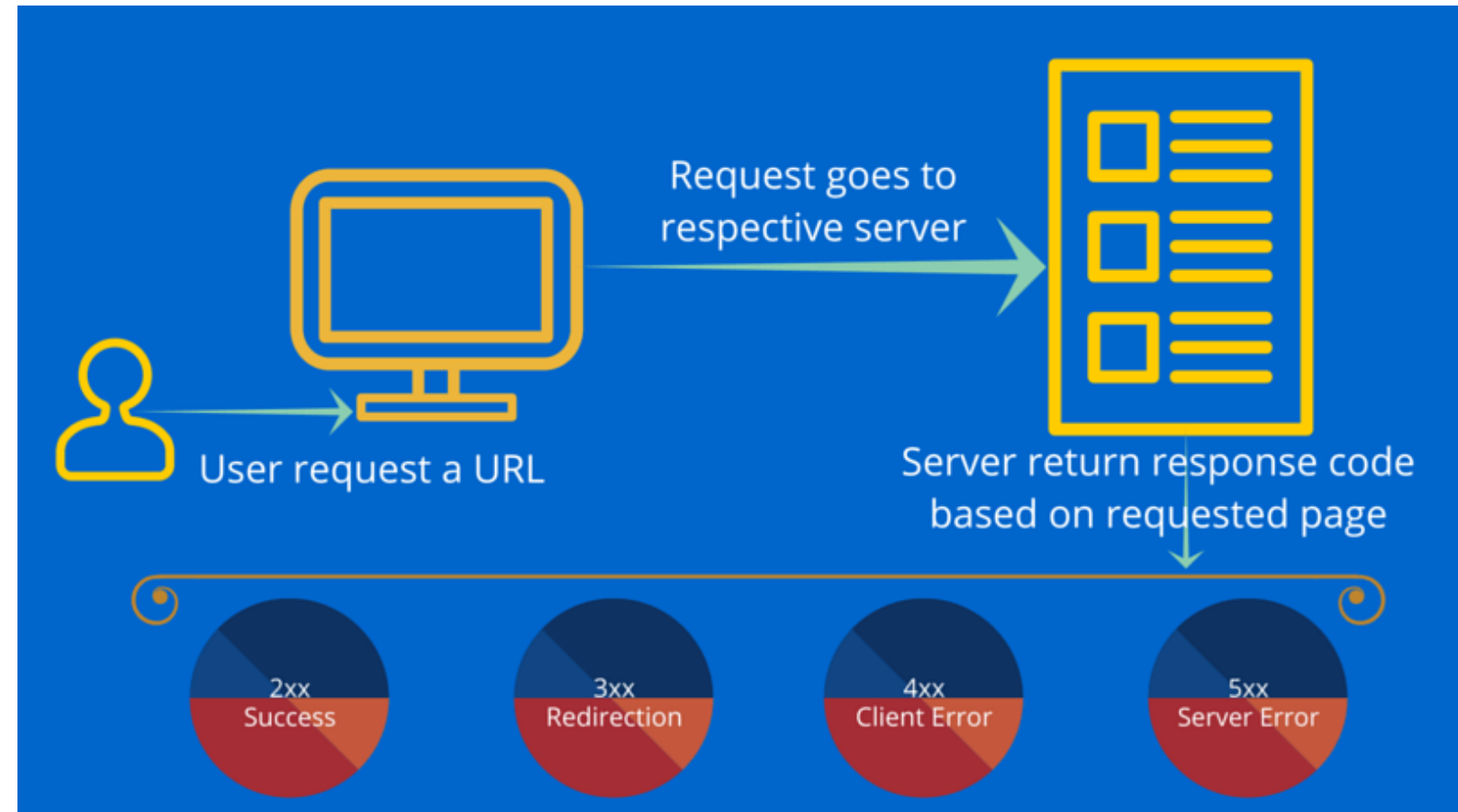
• DELETE

- Elimina el recurso
- Si el método se aplica con éxito, hay varios códigos de estado de respuesta posibles:
 - **202 (Accepted)**, es probable que la acción tenga éxito, pero aún no se ha llevado a cabo.
 - **204 (No Content)** si la acción ha sido promulgada y no se debe proporcionar más información.
 - **200 (OK)** si la acción ha sido ejecutada y el mensaje de respuesta incluye una representación que describe el estado.
 - Ejemplo: **DELETE /file.html HTTP/1.1**



<https://geekflare.com/es/http-status-code-infographics/>

Errores



<https://geekflare.com/es/http-status-code-infographics/>

CÓDIGOS DE ESTADO HTTP
RYTE

Code	Message
200	OK
301	Moved Permanently
302	Moved Temporarily
404	Not Found
500	Internal Server Error
503	Service Unavailable

404
PAGE NOT FOUND

Un código de estado es una parte de la respuesta devuelta por el servidor cuando un cliente (por ejemplo, un navegador) llama a una URL.

Con la ayuda de un código de estado, el servidor indica al cliente si la solicitud (por ejemplo, "envíame la página www.newyorker.com") se ha procesado correctamente o si se ha producido un error.

https://es.ryte.com/wiki/C%C3%B3digos_de_Estado_Http

Parámetros en Aplicaciones WEB

■ GET and POST HTTP request

```
GET /foo?par1=val1&par2=val2 HTTP/1.1  
User-Agent: Mozilla/5.0  
Host: Host  
Accept: */*
```

```
POST /foo HTTP/1.1  
User-Agent: Mozilla/5.0  
Host: Host  
Accept: */*  
Content-Length: 19  
  
par1=val1&par2=val2c
```


Cabeceras petición HTTP

- Las Cabeceras HTTP son los parámetros que se envían en una petición o respuesta HTTP al cliente o al servidor para proporcionar información esencial sobre la transacción en curso. Estas cabeceras proporcionan información mediante la sintaxis 'Cabecera: Valor' y son enviadas automáticamente por el navegador o el servidor Web.
- Una cadena de caracteres, que no diferencia mayúsculas ni minúsculas, seguida por dos puntos (':') y un valor cuya estructura depende de la cabecera.
- La cabecera completa, incluido el valor, ha de ser formada en una única línea, y puede ser bastante larga.

```
POST / HTTP/1.1
```

```
Host: localhost:8000
```

```
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
```

```
Accept: text/html,application/xhtml+xml,...,*/*;q=0.8
```

```
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
```

```
Connection: keep-alive
```

```
Upgrade-Insecure-Requests: 1
```

```
Content-Type: multipart/form-data; boundary=-12656974
```

```
Content-Length: 345
```

Request headers

General headers

Entity headers

```
-12656974
```

```
(more data)
```

<https://developer.mozilla.org/es/docs/Web/HTTP/Messages>

Cabecera Respuesta HTTP

- Las cabeceras HTTP para respuestas siguen también la misma estructura como cualquier otra cabecera: una cadena de texto, que no diferencia entre mayúsculas y minúsculas, seguida por dos puntos (':') y un valor cuya estructura depende del tipo de cabecera.
- Toda la cabecera incluido su valor, se ha de expresar en una única línea.

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 10 Aug 2016 13:17:18 GMT
Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"
Keep-Alive: timeout=5, max=999
Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT
Server: Apache
Set-Cookie: csrftoken=.....
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding
X-Frame-Options: DENY
```

Response headers

Entity headers

General headers

(body)

<https://developer.mozilla.org/es/docs/Web/HTTP/Messages>

Cookies

- Información enviada por un sitio web y almacenada en el navegador del usuario por un tiempo determinado.
- Si se ha realizado un curso desde un inicio o una nueva aplicación se pueden realizar con la misma contraseña o no en su sistema
- Sus principales funciones son:
 - Recordar accesos: Cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo, una cookie no identifica a una persona, sino a una combinación de computadora de la clase de computación-navegador-usuario.
 - Conocer información sobre los hábitos de navegación, e intentos de spyware (programas espía), por parte de agencias de publicidad y otros.
- Información enviada y recibida en las Cabeceras HTTP.

GET /ejemplo.html HTTP/1.1

Host: www.mysite.com

Cookie: colorPreference=blue;
sessionToken=**48745487**

NUEVA PETICION CLIENTE

HTTP/1.0 200 OK

Content-type: text/html

Set-Cookie: colorPreference=blue

Set-Cookie: sessionToken=**48745487**; Expires=Thu, 01 Jan 2031 19:22:10 GMT

RESPUESTA DEL SERVIDOR

Cookies





Lorem ipsum Dolor sit amet, consectetur Adipiscing Elit. Etiam eget quam lacus.