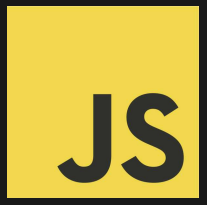


SEMINARIO DE LENGUAJES

SERVIDORES Y CLIENTES

CLASE 8 - SERVER SIDE

Programación Server side en JavaScript

AGENDA

1. Clientes Y Servidores
2. Servidores Web1
3. Servidores Web Dinámicos Y Estáticos
4. Servidores Dinámicos1
5. Programación Del Lado Del Cliente Y Del Lado Del Servidor
6. Programación Del Lado Del Servidor
7. Web Frameworks
8. Web Frameworks En Javascript
9. Resumiendo..
10. Referencias

CLIENTES Y SERVIDORES

Cuando visitamos un sitio web con nuestro navegador predilecto (Firefox, Chrome, Edge, etc.) se produce una conexión vía Internet entre un cliente —quien inicia la conexión y solicita el contenido del sitio web— y un servidor —quien recibe la conexión y envía el contenido solicitado.

CLIENTES Y SERVIDORES

Los clientes hacen peticiones, son dispositivos de distinto tipo. Los servidores son como un negocio, atienden las peticiones de los clientes.

SERVIDORES WEB[1]

Un servidor Web hace referencia a elementos de hardware y software o ambos, que trabajan en forma conjunta.

- Hardware: Almacena el software y los archivos de componentes. Conecta a Internet y soporta el intercambio de datos y dispositivos conectados a la red.
- Software: incluye la forma en que el usuario accede a los recursos. Un servidor HTTP entiende las URLs y HTTP

SERVIDORES WEB²

Los navegadores se comunican con los servidores Web a través de peticiones HTTP. Se incluye la URL que identifica el recurso, el método que identifica la acción requerida y puede incluir información adicional

codificada en parámetros de la URL como en POST.

```
GET https://developer.mozilla.org/en-US/search?q=client+server+overview&
topic=apps&topic=html&topic=css&topic=js&topic=api&topic=webdev HTTP/1.1
Host: developer.mozilla.org
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/52.0.2743.116 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;
q=0.8
Referer: https://developer.mozilla.org/en-US/
Accept-Encoding: gzip, deflate, sdch, br
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7
Accept-Language: en-US,en;q=0.8,es;q=0.6
Cookie: sessionid=6ynxs23n521lu21b1t136rhbv7ezngie;
csrftoken=zIPUJsAZv6pcgCBJSCj1zU6pQZbfMUAT; dwf_section_edit=False;
dwf_sg_task_completion=False; _gat=1; _ga=GA1.2.1688886003.1471911953; ffo=true
```

SERVIDORES WEB^3

```
POST https://developer.mozilla.org/en-US/profiles/hamishwillee/edit HTTP/1.1
Host: developer.mozilla.org
Connection: keep-alive
Content-Length: 432
Pragma: no-cache
Cache-Control: no-cache
Origin: https://developer.mozilla.org
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/52.0.2743.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;
q=0.8
Referer: https://developer.mozilla.org/en-US/profiles/hamishwillee/edit
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,es;q=0.6
Cookie: sessionid=6ynxs23n521lu21b1t136rhbv7ezngie; _gat=1;
csrftoken=zIPUJsAZv6pcgCBJSCj1zU6pQZbfMUAT; dwf_section_edit=False;
dwf_sg_task_completion=False; _ga=GA1.2.1688886003.1471911953; ffo=true

csrfmiddlewaretoken=zIPUJsAZv6pcgCBJSCj1zU6pQZbfMUAT&user-username=hamishwillee&
user-fullname=Hamish+Willee&user-title=&user-organization=&user-
location=Australia&user-locale=en-US&user-timezone=Australia%2FMelbourne&user-
irc_nickname=&user-interests=&user-expertise=&user-twitter_url=&user-
stackoverflow_url=&user-linkedin_url=&user-mozillians_url=&user-facebook_url=
```

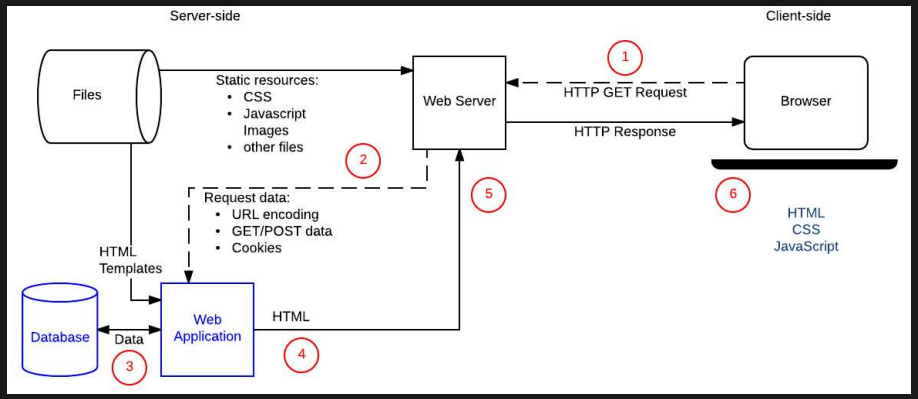
SERVIDORES WEB DINÁMICOS Y ESTÁTICOS

- Estáticos: consiste en una computadora con el software para servir peticiones HTTP. Se envían los archivos hosteados al cliente.
- Dinámicos:es un servidor estático + servidor de aplicaciones + una fuente de datos. La respuesta al cliente depende del contenido en la base de datos o de un JSON o cualquier otra fuente de datos.

SERVIDORES DINÁMICOS[1]

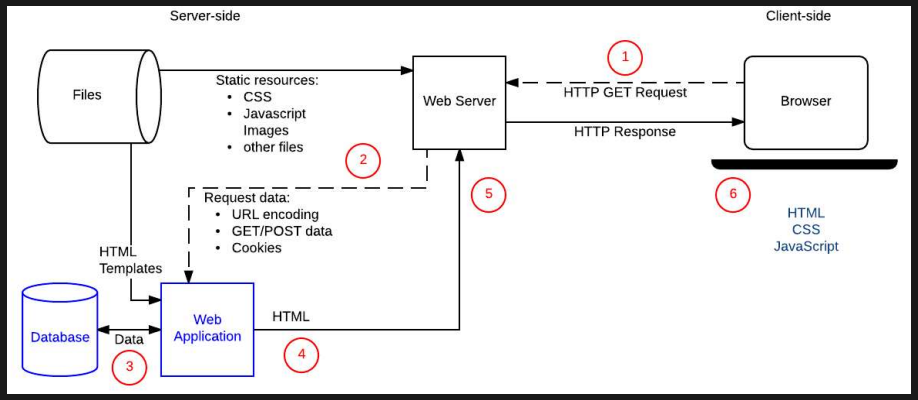
Son más versátiles y más complejos de administrar. Servidores para e-commerce, wikis, blogs, redes sociales y muchos usos más. Al pensar en una aplicación Web, su objetivo es necesario pensar que servidor Web se requiere y cuales se van a utilizar entre los disponibles.

SERVIDORES DINÁMICOS[2]



Los navegadores envían peticiones HTTP al servidor quien las procesa y devuelve las respuestas HTTP apropiadas. Las peticiones de recursos estáticos son gestionadas de la misma manera que para los sitios estáticos (CSS, JS, imágenes, PDF, ..).

SERVIDORES DINÁMICOS[3]



Las peticiones de recursos dinámicos son reenviadas (2) al código del lado-servidor (Web Application). Para las "peticiones dinámicas" el servidor interpreta la petición, lee de la base de datos la información requerida (3), combina los datos recuperados con las plantillas HTML (4), y envía de vuelta una respuesta que contiene el HTML generado (5,6).

PROGRAMACIÓN DEL LADO DEL CLIENTE Y DEL LADO DEL SERVIDOR

- Tienen diferentes objetivos y aspectos a considerar.
- En general no usan el mismo lenguaje de programación (con excepción de JS)
- Se ejecutan en distintos SO

PROGRAMACIÓN DEL LADO DEL SERVIDOR

La programación server-side permite almancenar y recuperar información en forma eficiente, brindar una experiencia de usuario personalizada, acceso controlado al contenido, almacenar información de sesión/estado, brindar notificaciones y comunicación,

análisis de datos, entre otras.

WEB FRAMEWORKS

- Colecciones de funciones, reglas, objetos y otras componentes de software diseñados para resolver problemas comunes.
- Permiten acelerar el desarrollo y simplificar tareas.
- Los frameworks del lado del cliente simplifican el diseño y la presentación. Por ejemplo REACT, Angular, View, entre otros.
- Los frameworks del lado del servidor simplifican tareas comunes como sesiones, soporte de usuarios y autenticación, plantillas, etc.

WEB FRAMEWORKS EN JAVASCRIPT

NodeJS es un servidor Web y Express es un framework escrito en JS para NodeJS. Más flexible, rápido y minimalista.

Permite crear APIs y aplicaciones Web fácilmente, provee manejo de rutas, archivos estáticos, motor de plantillas, integración con base de datos, manejo de errores, middlewares, etc.

RESUMIENDO..

Server-side JS - Mapa Conceptual

REFERENCIAS

Introduction to server-side programming languages

Introducción al lado servidor

ExpressJS

NPM

ExpressJS Middleware

ExpressJS - Motor de plantillas EJS Server-side JS - Mapa Conceptual