

1. Arquitectura de la web y herramientas de desarrollo

Todas las explicaciones debes realizarlas con tus propias palabras, no se puede copiar y pegar de Internet.

1. ¿Cuál es la relación entre Internet y World Wide Web?

Internet es el conjunto de redes y la infraestructura por la cual está todo conectado y WWW es el sistema o el camino por donde se mueve la información.

2. Explica para qué sirven cada uno de los siguientes protocolos en el contexto de la web.

- a. **TCP** → crea un camino seguro entre dos computadoras por dónde poder pasar información.
- b. **IP** → es el nombre identificativo numérico que tiene que tener un dispositivo para poder estar conectado a la red de internet (siempre que la IP sea pública o esté conectado a una).
- c. **HTTP** → es un protocolo de envío de información
- d. **HTTPS** → el mismo protocolo que HTTP pero resistente a malwares
- e. **TLS** → protocolo de encriptación que hace más seguro el tráfico de datos normalmente por Internet

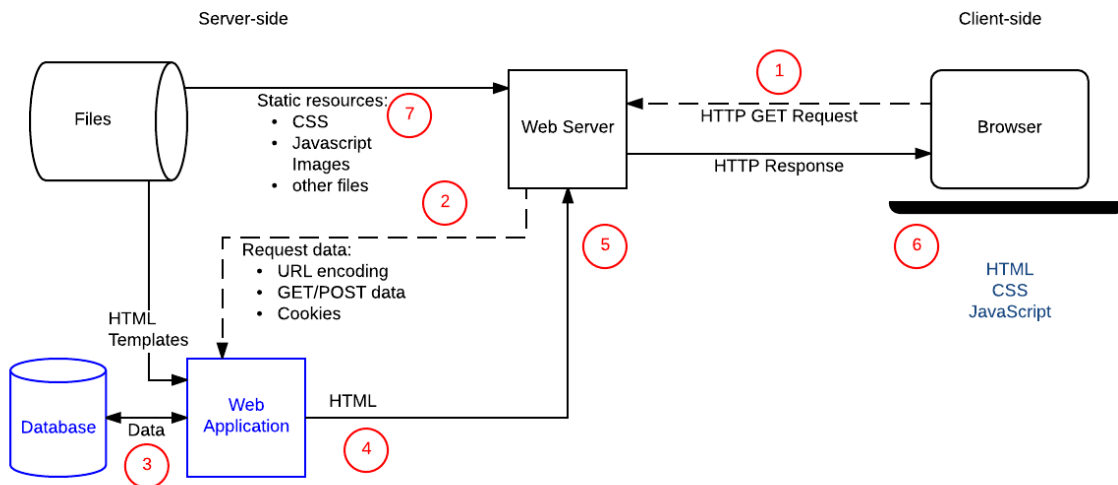
3. Explica en qué consiste el hipertexto y qué papel juega en la web.

Un hipertexto es un texto que contiene un enlace a otra web y juega un papel muy importante ya que tienes un texto muy resumido de otra página web la cual no hace falta que cargue hasta que hagas click en dicho texto.

4. Explica en qué consiste el modelo cliente-servidor en el que se basa la web.

Es un modelo de reparto de tareas dónde el cliente formula peticiones y el servidor devuelve el resultado de la petición.

5. Explica y paso a paso una petición web dinámica. Utiliza el siguiente diagrama que se ha visto en clase:



1. El cliente hace una petición tipo GET al servidor web con extensión “.(lenguaje de programación usado en backend)”.
2. Envía la petición al servidor de aplicaciones dónde ejecuta el código necesario para buscar la información en la base de datos.
3. Devuelve los datos necesarios de la base de datos a la aplicación web.
4. Transforma la información en html.
5. Se envía al servidor web.
6. Carga en el cliente como respuesta al GET inicial.
7. Cargar estilos, imágenes, javascript y otros archivos restantes.

6. Explica el modelo de capas de la programación en el cliente:

Comportamiento (JS, WASM)	Añade animación
Presentación (CSS)	Añade estilos
Estructura (HTML)	Contenido estructurado
Contenido (texto, imágenes, etc.)	

7. Nombra, y explica con tus palabras las mejores prácticas del desarrollo web cliente vistas en clase.

Compatibilidad entre navegadores → que la página se pueda abrir preferentemente en los 4 navegadores más usados

Diseño web responsive → que se adapta a todos los tamaños de pantalla

Performance → diseñarlo con una interfaz amigable, fácil de usar y de carga rápida para no desesperar al usuario.

Accesibilidad → hacer la web de manera que puedan usarla el mayor número de personas posible, ya sean personas con discapacidades o de todo tipo de edades.

Internacionalización → web traducida a diferentes idiomas y culturas.

Privacidad y seguridad → hacer segura la página web.

8. Elige y explica 5 limitaciones que tiene javascript y, en general, la programación en el cliente.

- Cientes que no soportan js
- El usuario puede desactivarlo
- Al ejecutarse en un sandbox no tiene acceso total a la información del cliente.
- Hay que usar Babel para una máxima compatibilidad
- El código siempre es visible en el navegador

9. Explica las 3 formas vistas en clase de incluir un script en el HTML. Pon un ejemplo ejemplos de cada caso.

- Cargar un archivo .js externo con la etiqueta `<script>`
- Añadir un bloque de código directamente
- Añadir el código js a un elemento html

10. Hemos visto en clase que la etiqueta `<script>` debería ponerse dentro de la etiqueta `<head>`, pero es una práctica habitual ponerla justo antes del cierre de la etiqueta `</body>`. ¿Cuál es la razón de esta práctica?

Para usarlo después de una acción que te interesa que cargue antes que el `<script>`

11. Justifica la existencia de los atributos `async` y `defer` de la etiqueta `<script>` y cuál es su diferencia.

- async:** el js se ejecuta nada más cargar.
- defer:** carga pero se ejecuta al acabar de cargar el DOM.

Las siguientes actividades se tienen que hacer, pero no documentar.

12. Instala Visual Studio Code en tu equipo y añade la extensión de LiveServer. Llama al profesor cuando lo tengas.

