

DWEC01. Selección de arquitecturas y herramientas de programación

Ejercicio 1

- 1- Explica en qué consiste el modelo cliente-servidor.

El modelo cliente-servidor, el dispositivo que solicita información se denomina “cliente”, el que responde a la solicitud se denomina “servidor”. Los procesos de cliente y servidor se consideran parte de la capa de la aplicación. El cliente comienza el intercambio solicitando los datos al servidor, quien responde enviando uno o más streams de datos al cliente. Los protocolos de la capa de aplicación describen el formato de las solicitudes y respuestas entre clientes y servidores. Además de la transferencia real de datos, este cambio también puede requerir la autenticación del usuario y la identificación de un archivo de datos que se vaya a transferir.

Un ejemplo podría ser el uso del servicio de correo electrónico de ISP para enviar, recibir y almacenar correo electrónico. El cliente de correo electrónico es una PC doméstica emite una solicitud al servidor de correo electrónico del ISP para que se le envíe todo correo no leído. El servidor responde enviando al cliente el correo electrónico solicitando.

- 2- Describe al menos 5 características de un cliente.

- En el mayor de los casos interactúa con los usuarios finales.
- Se puede conectar varios servidores a la vez.
- Recibe respuestas del servidor.
- Es el primero en iniciar una solicitud o petición.
- Pide peticiones al servidor.

- 3- Describe al menos 5 características de un servidor.

- Recibe las solicitudes de los clientes.
- Puede tener conexión con más de un cliente.
- Recibe la información para procesarla y luego se la envía al cliente.
- Procesa la lógica de la aplicación y realiza validaciones a nivel de bases de datos.
- Procesa los requerimientos de las bases de datos

- 4- ¿Cómo se consigue que un cliente se comunique con un servidor o viceversa?

Para que esta comunicación funcione, la interfaz de usuario, es la capa encargada de hacer posible la interacción entre el usuario y el motor de navegación, esta se comunica con el Sistema Operativo para manejar las sesiones de los usuarios o el almacenamiento de preferencias.

También tenemos el motor de navegación, es la capa que consulta y administra el motor de renderizado (carga las URLs, recarga las páginas, gestiona alertas de JavaScript...)

El motor renderizado, es el encargado de interpretar el código de la página web. Debe ser capaz de mostrar documentos de audios y videos, html o xml, hojas de estilos como CSS e imágenes.

Ejercicio 2

Lee el siguiente artículo que trata sobre los lenguajes de script y contesta a las siguientes preguntas:

- 1- Respecto a los clientes Web, ¿Qué lenguajes se utilizan actualmente y por qué? Menciona sus características.
- JavaScript -> Es el lenguaje mas usado por su rendimiento y su simplicidad. Algunas características son:

- Lenguaje al lado del cliente: Esto se refiere a que se ejecuta en la máquina del propio cliente.
 - Lenguaje orientado a objetos: Un lenguaje orientado a objetos quiere decir que utiliza clases y objetos como estructuras que permite utilizarse de forma simple y son reutilizables durante todo el desarrollo.
 - De tipado débil o no tipado: Esto nos quiere decir que no es necesario especificar el tipo de dato al declarar una variable. Esta característica supone una gran ventaja a la hora de programar que seremos más rápidos.
 - De alto nivel: Que JavaScript sea de alto nivel significa que su sintaxis es fácilmente comprensible por su similitud al lenguaje de las personas. Se llama de “alto nivel” porque su sintaxis esta alejada del nivel máquina.
 - Lenguaje interpretado: Es un lenguaje interpretado porque utiliza un intérprete que permite convertir las líneas de código en el lenguaje de máquina.
 - Muy utilizado por desarrolladores.
- TypeScript -> Se va usando cada vez más pero todavía no ha llegado a su máximo, podemos decir que se trata de un superconjunto de JavaScript, es decir amplía JavaScript con una nueva sintaxis que añade, entre otras cosas, el tipado estático opcional, genéricos, decoradores y elementos de POO como interfaces o property accessors. Algunas características son:
 - Apoyado por fabricantes: Tanto Microsoft como Google apoyan de una forma fuerte este lenguaje.
 - Compilado: Podemos detectar muchos de los errores antes de abordar la ejecución.
 - Tipado: Aporta tipos de datos estáticos encima de JavaScript.
 - Moderno y en constante evolución: Tiene opciones interesantes como por ejemplo unión de types que permite que un tipo soporte dos opciones.
 - Webassembly -> Lenguaje de bajo nivel, con mucha trayectoria por delante. Sus desarrolladores son los grandes navegadores Web. Mejora la velocidad de las aplicaciones. Algunas características son:
 - El código del cliente no es legible, ya que no es JavaScript, es código nativo. Sí, si se puede analizar, pero es casi como analizar un archivo .exe o ejecutable.
 - Se programa en un lenguaje fuertemente tipado o al menos un lenguaje seguro, no como JavaScript.
 - Mejora bastante el rendimiento de las Aplicaciones Web y sobre todo los videojuegos.
- 2- ¿Cuáles son las ventajas y desventajas de los lenguajes en script respecto a los lenguajes de programación tradicionales?
- Mientras que los lenguajes Script son lenguajes específicos de la plataforma web, los lenguajes de programación son multiplataformas.
 - Los lenguajes de Script es su mayoría son interpretados.
 - Los lenguajes Scripts son mas lentos que los lenguajes de programación, ya que estos tienen que ser compilados en el momento.
 - Con los lenguajes de Script se escribe menos código porque son más intensivos en codificación.
 - Con los lenguajes de Script necesitas un Navegador Web para que funcione la aplicación.

Ejercicio 3

- 3- ¿Qué herramientas has utilizado para realizar este ejercicio y para qué has utilizado cada una de ellas?

La herramienta que he utilizado es el Visual Studio Code para todo el tema de la programación.

Utilizo esta herramienta porque es con la que esto mas familiarizado y con la que me veo mas cómodo trabajando.

Google Chrome, lo he utilizado para testear que todo el código generado funcionara correctamente.

Ejercicio 4

- 1- Firefox vs. Chrome Dev Tools. Prueba e investiga sobre las herramientas Firefox y Chrome Dev Tools, documentando las similitudes y diferencias que has encontrado entre ellas.

A simple vista son muy parecidas y las dos tienen casi las mismas opciones, pero diría que la de Firefox se ve un poco más completa y profesional, en cuanto al diseño también se ve todo más claro.

Algunas diferencias que he encontrado es que Firefox Dev es que tiene “las herramientas de la fuente” y el “analizador de animaciones”, esto para un diseñador le viene muy bien.

Firefox te viene todo en español en cambio Chrome te viene en inglés.

Como bien dije antes la consola de Firefox parece ser la más profesional, ya que también nos muestra otros tipos de problemas como; los de red, CSS, errores de seguridad y mensajes de registros y poder interactuar con ellos a través de la interfaz de la consola.

- 2- Realiza una comparativa entre los frameworks/Librerías de JavaScript, más conocidos, concluyendo cuál de ellos te gustaría estudiar y la razón.

Angular, React.js, Vue.js, Node.js

- Angular -> Es uno de los frameworks open source más avanzados. Para crear sitios web de una sola página con elementos interactivos. Permite la actualización en tiempo real de un sitio web desde diferentes dispositivos. Cualquier vista se cambia de forma simultánea en la aplicación móvil y el sitio web. Además, cuenta con apoyo comunitario muy grande.
- React.js -> No es realmente un framework, sino una librería, dos de las aplicaciones web más utilizadas en el mundo están desarrolladas con React.js. Facebook e Instagram, entre muchas otras, utilizan la capacidad interactiva que proporciona esta librería. Si lo que quieres desarrollar una web robusta, dinámica y escalable con un rendimiento fiable React.js te lo dará. Otra de las características es la capacidad de traer elementos reutilizables.
- Vue.js -> Es un framework progresivo de Javascript para crear interfaces de usuario, con un lenguaje particular que nos permite crear aplicaciones de forma rápida, agradable, sencilla y muy práctica.
- Node.js -> Es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en Javascript.
Node es un entorno controlado por eventos diseñados para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos. Sus características principales.
 - Velocidad: Al estar construido sobre el motor de JavaScript de Google Chrome, por eso su biblioteca es muy rápida en la ejecución de código.
 - Sin búfer: Generan los datos en trozos (chunks), nunca los almacenan en búfer.
 - Un subproceso escalable: Node.js utiliza un modelo de un solo subproceso con bucle de eventos.

Sobre los que más me gustaría estudiar y la razón son:

Angular creo que sería un framework interesante, ya que ahora está en crecimiento y se está demandando cada vez más.

Por otro lado, este año he tenido la suerte de conocer React y que no sabía nada sobre él y la verdad que es bastante potente y seguiré avanzando por mi cuenta con él, y que también está en una época de crecimiento y cada vez se demanda más.