Despliegue de App Web

2ºCFGS DAW UNIDAD 1

ACTIVIDAD 1:

1. **¿Qué arquitectura web hemos visto en clase? ¿Qué modelo sigue? ¿Cuáles son las características básicas de ese modelo?**

Hemos visto la arquitectura WWW que proviene del modelo cliente servidor.

Sus características básicas son:

**Separación de funciones**: El sistema se divide en dos roles principales: el cliente, que solicita servicios, y el servidor, que los proporciona.

**Comunicación a través de una red**: Los clientes y servidores interactúan a través de una red (como Internet o una red local) usando protocolos como HTTP o TCP/IP.

**Procesamiento centralizado en el servidor**: El servidor aloja y ejecuta la lógica de negocio, procesa solicitudes y maneja bases de datos, mientras que el cliente presenta la interfaz al usuario.

**Múltiples clientes**: Un servidor puede atender a varios clientes simultáneamente, lo que lo hace escalable.

**Interdependencia**: El cliente depende del servidor para acceder a los recursos o servicios; sin la conexión o servidor disponible, el cliente no puede funcionar plenamente.

**Seguridad y control centralizado**: El servidor suele gestionar aspectos clave como la autenticación, la autorización y la protección de datos.

1. **¿Qué entendemos cuando hablamos de HTTP? ¿Cómo funciona la comunicación de este protocolo? ¿Cuáles son sus métodos más utilizados? ¿En qué se diferencia de HTTPS?**

HTTP es el protocolo que se usa para la transferencia de archivos entre el servidor web y el cliente.

HTTP funciona de la siguiente manera, un usuario accede a una URL, el cliente WEB descodifica la URL, separando sus diferentes partes, se abre una conexión TCP o UDP, dependiendo de la versión de HTTP y luego el servidor devuelve la respuesta al cliente. Por último, se cierra la conexión TCP o UDP.

El HTTPs es lo mismo pero los datos viajan cifrados, de modo que no son legibles por si una persona quiere interceptarlos.

1. **Define las distintas capas que podemos encontrar en un modelo básico de arquitectura web ¿Se puede definir alguna capa más? ¿Cuál es la principal diferencia en la funcionalidad de un modelo de 3 capas y uno de 4?**

En un modelo básico de arquitectura web, tenemos 3 capas. La capa de presentación es la parte del front-end, es la encargada de la parte de la navegabilidad, validación de datos de entrada, etc. Se trata de lo que se presenta al usuario.

La capa de negocio es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifican que las reglas establecidas se cumplen.

Por último, la capa de acceso a datos es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio.

La diferencia con el modelo de 4 capas es que se incluye una capa más entre el front-end y el back-end. La capa de servicios: permite desacoplar la interfaz de usuario del resto de capas, permitiendo que las funcionalidades de nuestra aplicación sean accesibles por otras aplicaciones u servicio.

1. **Haz una breve investigación sobre alguno de los modelos de arquitectura web que se han desarrollado y comenta las ventajas que introdujeron.**

Uno de los modelos más influyentes en la arquitectura web es el modelo cliente-servidor. En este modelo, los usuarios (clientes) interactúan con un servidor a través de navegadores web u otras aplicaciones. El servidor aloja y procesa la lógica de la aplicación, así como los datos.

Ventajas del modelo cliente-servidor:

Centralización del control: El servidor gestiona los datos y la lógica de la aplicación, lo que facilita su administración, mantenimiento y actualización.

Escalabilidad: El servidor puede manejar múltiples clientes simultáneamente, y es posible aumentar su capacidad según la demanda.

Seguridad: Al centralizar los datos, es más fácil implementar controles de seguridad, como firewalls y autenticación.

Facilidad de distribución de actualizaciones: Los cambios en la aplicación se realizan en el servidor, sin necesidad de actualizar los clientes manualmente.