

TAREA 5.1: ARQUITECTURA DE APLICACIONES WEB

David Rodríguez

2º DAW Desarrollo Web en Contorno Servidor

1- ¿Por qué la arquitectura de cliente/servidor se conoce como arquitectura de dos capas?

Se conoce así por su funcionamiento: requiere de una interfaz de usuario que se instala en el ordenador del cliente/estación de trabajo que envía solicitudes a un servidor que ejecuta operaciones complejas.

2- Elabora una tabla con las ventajas e inconvenientes del sistema de dos capas.

Ventajas	Inconvenientes
<ul style="list-style-type: none">- Buen desempeño en ambientes no dinámicos estables.- Herramientas de desarrollo de sistemas de dos capas robustas y evaluadas.	<ul style="list-style-type: none">- No es el sistema más eficiente.- Difícilmente escalable.- Número de conexiones limitada.- Mala ejecución en organizaciones que cambian constantemente.- Requieren un control excesivo de las versiones y se debe controlar su distribución si se le hacen cambios a la aplicación.- La seguridad del sistema es compleja y requiere administración de bases de datos debido a la cantidad de dispositivos que acceden a ella.- Las herramientas de desarrollo, si bien son robustas, están en constante cambio y la dependencia de alguna de ellas puede complicar o lastrar el escalamiento futuro de la aplicación.

3- Explica brevemente en qué consisten los tres niveles en los que se suele dividir la arquitectura de las aplicaciones web.

- El primer nivel es la capa de cliente, que puede ser una interfaz de navegador web, con la que el usuario interactúa con la máquina. Presenta los datos procesados y devueltos desde la capa de aplicaciones.
- El segundo nivel es una capa de aplicaciones intermedia que es encargada de recibir los parámetros del ordenador cliente y contiene el o los programas que se encargan de procesar la petición. Puede realizar una petición a un servidor en el tercer nivel.
- El tercer nivel es una capa de servidor de datos que recibe peticiones del servidor de aplicaciones para devolver respuestas a una petición a una base de datos. Aloja, recupera, mantiene y asegura la integridad de los datos.

4- Elabora una tabla con las ventajas e inconvenientes del sistema de tres capas. Explica en detalle cada concepto.

Ventajas	Inconvenientes
<ul style="list-style-type: none"> - Las llamadas de la capa cliente/usuario al servidor de aplicaciones son más flexibles, ya que sólo se necesitan enviar parámetros. - La interfaz de cliente no es necesaria para comunicarse con el receptor de datos, por lo que dicha estructura de datos puede ser modificada sin que afecte a la interfaz cliente. - La separación de roles en tres capas hace más fácil el reemplazo, modificación o reutilización de código, módulos o capas sin que afecte a otros módulos. 	<ul style="list-style-type: none"> - Si el entorno de trabajo del sistema de tres capas se ve sometido a un elevado tráfico de datos, este requiere que se administre su balance de carga y que sea robusto ante fallos. - No todos los navegadores actuales son iguales, lo que puede ocasionar problemas de compatibilidad en el funcionamiento con el servidor de aplicaciones. - La estandarización de desarrollo entre diferentes proveedores e internet ha sido lenta y puede no ser completa, lo que en ocasiones obliga a escoger entre un estándar o una solución propia de un proveedor.

5- Compara ASP.NET y PHP: ventajas e inconvenientes.

- ASP.NET:

Es un lenguaje desarrollado por Microsoft usado para desarrollar sitios web, siendo una evolución de ASP, superando sus limitaciones. Es necesario IIS con el framework .NET propios de Microsoft para que funcione. Consiste en una serie de clases .NET utilizadas para crear aplicaciones web tanto del lado del cliente como del lado servidor. Las páginas creadas con ASP.NET funcionan en todo tipo de navegadores.

○ Ventajas:

- Está completamente orientado a objetos.
- Permite personalizar los controles de usuario.
- La capa de aplicación y la de código están bien diferenciadas.
- Facilita el mantenimiento de grandes aplicaciones.
- Mayor seguridad, rapidez y respuesta del servidor.

○ Inconvenientes:

- Alto consumo de recursos.
- Tecnología propietaria de Microsoft, lo que implica pagar una licencia de uso.
- El hospedaje de sitios web puede ser costoso.

- PHP:

Es un lenguaje de programación interpretado de código abierto del lado servidor, rápido y con una gran biblioteca de funciones. Permite incrustar código HTML, y el terminal cliente sólo verá el código HTML ejecutado en su interfaz del navegador.

○ Ventajas:

- Lenguaje, gratuito, rápido, multiplataforma, con gran funcionalidad modular y fácil de aprender.
- Medianamente orientado a objetos, presenta clases y herencia; no requiere definición de tipos de variables ni manejo de bajo nivel.
- Permite conectarse a muchos administradores de bases de datos: MySQL, MariaDB, MS SQL Server, etc.
- Muy documentado.

○ Inconvenientes:

- Al ejecutarse del lado servidor, puede volverse ineficiente ante elevadas cargas de trabajo o número de solicitudes.
- La lectura de código puede complicarse si se mezclan muchas sentencias HTML y PHP.
- Su programación orientada a objetos es muy deficiente de cara a aplicaciones grandes.
- Dificulta la modularización y la organización de la aplicación por capas.

BIBLIOGRAFÍA/WEBGRAFÍA

- Anónimo (2018): *Arquitectura de las aplicaciones web*, en nanopdf.com [recurso electrónico], consulta el 10 de enero de 2022 (https://nanopdf.com/download/21-arquitectura-de-las-aplicaciones-web_pdf).
- Matehuala, Instituto Tecnológico (2019): *Arquitectura de las aplicaciones web*, col. Programación Web, en programacionwebisc.wordpress.com [recurso electrónico], consulta el 10 de enero de 2022 (<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>).