

Preguntas Teóricas a resolver

Presentaciones.

1. ¿Aplicación Web = Sitio Web? ¿Por qué?
2. ¿A qué nos referimos cuando hablamos de “cliente” en desarrollo web?
3. ¿A qué nos referimos cuando hablamos de lenguaje del lado del cliente? Mencione ejemplos.
4. ¿A qué nos referimos cuando hablamos de lenguaje del lado del servidor? Mencione ejemplos.
5. ¿Puedo conectarme a una base de datos utilizando HTML?

Web Estáticas y Dinámicas.

1. ¿Qué es una página web dinámica? Mencione sus características.
2. Nombre 3 ejemplos de ellas.
3. ¿Qué es una página web estática? Mencione sus características.
4. Nombre 3 ejemplos de ellas.
5. En la actualidad, en qué caso utilizaría cada uno de ellas.

Instalación

1. ¿Qué aplicación utilizaron para instalar el servidor Web? Y ¿Por qué?
2. ¿Cómo puedo saber que versión de PHP tengo instalado?
3. Indique en qué directorio del servidor tengo que alojar mi sitio/sistema para poder utilizarlo/probarlo.
4. ¿Qué archivo de configuración tengo que editar para mostrar/ocultar los errores en tiempo de ejecución? Y ¿Cuáles son las distintas opciones para eso?
5. ¿Cuáles son las variables que debo modificar para no tener grandes sobresaltos a la hora de implementar un sistema/sitio web?

DESARROLLO

Presentaciones

1. No, una aplicación web no es lo mismo que un sitio web. Un sitio web es principalmente informativo y este mismo está compuesto por páginas web estáticas o dinámicas. En cambio, una aplicación web es un programa interactivo que permite a los usuarios realizar tareas específicas y manejar datos, funcionando en un navegador web.
2. En desarrollo web, el término "cliente" generalmente se refiere al dispositivo y software (usualmente un navegador web) que un usuario utiliza para acceder y interactuar con sitios web o aplicaciones web. Este entorno ejecuta el código recibido del servidor, como HTML, CSS, y JavaScript.
3. Cuando hablamos de "lenguaje del lado del cliente" en el desarrollo web, nos referimos a los lenguajes de programación que se ejecutan en el navegador del usuario, en lugar de en el servidor. Estos lenguajes se utilizan principalmente para desarrollar la interfaz y la interactividad de las páginas web, permitiendo que estas respondan a las acciones del usuario sin necesidad de recargar la página o comunicarse constantemente con el servidor. Ejemplos de esto son HTML, CSS, etc.
4. Cuando hablamos de "lenguaje del lado del servidor" en el desarrollo web, nos referimos a los lenguajes de programación que se ejecutan en el servidor, procesando las peticiones enviadas por el cliente (normalmente un navegador) y generando respuestas adecuadas, como páginas web dinámicas o datos en formatos específicos. Estos lenguajes también gestionan tareas como la interacción con bases de datos, la lógica de negocio, y la autenticación de usuarios. Ejemplos de estos son Python, PHP, etc.
5. No, no se puede conectarse directamente a una base de datos utilizando solo HTML. HTML es un lenguaje de marcado usado para estructurar y presentar contenido en la web, pero no tiene capacidades de programación o interacción directa con bases de datos.

Web Estáticas y Dinámicas

1. Una página web dinámica es aquella que muestra contenido que puede cambiar en respuesta a diferentes contextos o condiciones. A diferencia de las páginas web estáticas, que siempre muestran el mismo contenido cada vez que son cargadas, las páginas dinámicas pueden adaptarse y mostrar información diferente según la interacción del usuario, datos del servidor, o cualquier otro tipo de entrada. Estas páginas son generadas en tiempo real, usualmente mediante lenguajes de programación del lado del servidor. Sus características son Interactividad, Contenido personalizado, Conexión con bases de datos, Generación en tiempo real, Uso de scripts del lado del servidor.
2. Ejemplos de páginas web dinámicas pueden ser tiendas en línea, redes sociales (por ej. instagram), portales de noticias, etc.
3. Una página web estática es aquella que muestra el mismo contenido fijo cada vez que se accede a ella. El contenido de una página estática no cambia a menos que sea manualmente actualizado por el desarrollador. Estas páginas son básicamente archivos HTML que son servidos por el servidor web tal como están almacenados. Sus características son contenido fijo, carga rápida, facilidad de desarrollo y hospedaje, seguridad, etc.
4. Tres ejemplos pueden ser:
Sitio web personal: Un sitio web que solo contiene información sobre una persona, como su currículum, información de contacto y tal vez algunas fotos.
Portafolio en línea: Un sitio web que muestra el trabajo y proyectos de un diseñador,

fotógrafo o artista.

Página de inicio de empresa: Un sitio web que presenta información sobre una empresa, sus servicios y su equipo.

5. En la actualidad utilizaría:

Web estática: Las páginas web estáticas son ideales cuando el contenido del sitio no necesita actualizaciones frecuentes y permanece relativamente constante a lo largo del tiempo. Por ejemplo, un sitio web personal que muestra un portafolio de trabajo. También Si necesitas una presencia en línea básica y no tienes la necesidad de funcionalidades complejas o interactivas, una página web estática puede ser la mejor opción.

Web dinámica: Si necesitas que los usuarios interactúen con tu sitio web regularmente, como en redes sociales, foros, o tiendas en línea donde los usuarios puedan dejar reseñas y comentarios, entonces una página web dinámica es necesaria para gestionar estas interacciones en tiempo real. También si tu sitio web requiere actualizaciones frecuentes de contenido, como un blog, noticias, inventario de productos en una tienda en línea, o información que cambia regularmente.

Instalacion

1. La aplicación que utilizamos para instalar el servidor web fue "XAMPP", porque es XAMPP es un paquete de software libre que simplifica el proceso de desarrollo de aplicaciones web al proporcionar un entorno completo y listo para usar, no es necesario instalar cada una de las herramientas que incluye (Apache, MySQL, PHP y Perl) por separado, lo que ahorra tiempo y evita errores de configuración.

2. Se puede saber la versión de PHP ejecutando un archivo PHP, creando un archivo PHP con el siguiente código:

```
<?php  
echo 'Versión de PHP: ' . phpversion();  
?>
```

Luego hay que subir este archivo a la raíz del sitio web.

Después, abrir un navegador web y acceder a la dirección completa del archivo.

El código anterior va a mostrar la versión de PHP sin detalles adicionales

O también si es que tenemos permisos para acceder al servidor a través de SSH, podemos usar la línea de comandos para verificar la versión de PHP. Ahora hay que abrir una terminal y ejecuta el siguiente comando:

```
php -v
```

En lo personal me resulta más sencilla esta ultima forma.

3. En servidores Apache como el que utilizamos, el directorio raíz suele ser C:\inetpub\wwwroot, en servidores Windows.
4. Para mostrar u ocultar los errores en tiempo de ejecución en PHP, puedes modificar la configuración utilizando el archivo **php.ini**:

Para editar la configuración de visualización de errores, hay que buscar la directiva **display_errors** dentro del archivo **php.ini**.

- **display_errors = On**: Esto mostrará los errores en tiempo de ejecución en la salida del navegador web.
- **display_errors = Off**: Esto ocultará los errores en tiempo de ejecución y los enviará al archivo de registro de errores del servidor (si está configurado).

Es importante recordar la capacidad de modificar la configuración de PHP a través de archivos **.htaccess** puede estar desactivada en la configuración del servidor por razones de seguridad.

Si se desea mostrar u ocultar errores en tiempo de ejecución a nivel del servidor web Apache, se puede configurar esto a través del archivo de configuración **httpd.conf** o en archivos **.htaccess** específicos del directorio donde se encuentra tu aplicación.

- Para habilitar la visualización de errores en tiempo de ejecución en Apache, puedes agregar o modificar las siguientes líneas en tu archivo de configuración:
`php_flag display_errors on`
`php_value error_reporting E_ALL`
- Para deshabilitar la visualización de errores en tiempo de ejecución:
`php_flag display_errors off`

5.

1. Configuración del servidor:

- Asegúrate de que la configuración del servidor (como PHP, Apache, MySQL, etc.) cumpla con los requisitos de tu aplicación.
- Verifica que las versiones de software sean compatibles con tu sistema.

2. Seguridad:

- Implementa medidas de seguridad adecuadas, como firewalls, certificados SSL, y reglas de acceso.
- Asegúrate de que los datos sensibles estén protegidos mediante cifrado y almacenamiento seguro.
- Actualiza regularmente el software y parchea cualquier vulnerabilidad conocida.

3. Respallos (backups):

- Implementa medidas de seguridad adecuadas, como firewalls, certificados SSL, y reglas de acceso.
- Asegúrate de que los datos sensibles estén protegidos mediante cifrado y almacenamiento seguro.

4. Escalabilidad:

- Diseña tu sistema para que pueda escalar según las necesidades futuras, tanto en términos de tráfico como de funcionalidad.
- Utiliza técnicas como la distribución de carga y el aprovisionamiento automático para manejar picos de tráfico.

5. Monitoreo y diagnóstico:

- Implementa herramientas de monitoreo para supervisar el rendimiento del servidor, la disponibilidad del sitio y la integridad de los datos.
- Configura alertas para ser notificado sobre problemas potenciales antes de que afecten a los usuarios.

6. Pruebas exhaustivas:

- Implementa herramientas de monitoreo para supervisar el rendimiento del servidor, la disponibilidad del sitio y la integridad de los datos.
- Configura alertas para ser notificado sobre problemas potenciales antes de que afecten a los usuarios.

7. Documentación:

- Implementa herramientas de monitoreo para supervisar el rendimiento del servidor, la disponibilidad del sitio y la integridad de los datos.
- Configura alertas para ser notificado sobre problemas potenciales antes de que afecten a los usuarios.

8. Pruebas posteriores a la implementación:

- Después de la implementación, verifica nuevamente que todo funcione correctamente.
- Realiza pruebas exhaustivas en el entorno de producción.