

1. ¿Qué es la normalización en bases de datos y por qué es importante aplicarla?

La normalización es un proceso que se aplica al diseño de bases de datos para organizar correctamente la información, separando los datos en diferentes tablas relacionadas entre sí, con el fin de evitar la repetición innecesaria y asegurar que todo esté bien estructurado.

- Evita datos duplicados.
- Usa claves foráneas para unir.
- Facilita el mantenimiento.

2. ¿Qué datos nunca deberías guardar en texto plano en una base de datos cuando creas un sistema de login?

Nunca se debe guardar una contraseña en texto plano (visible) en la base de datos. Lo correcto es cifrarla usando técnicas de hashing, como bcrypt o la función password_hash() en PHP. Esto asegura que, aunque alguien acceda a la base de datos, no podrá ver ni usar las contraseñas reales.

3. ¿Qué harías para evitar que alguien acceda a una página protegida sin haber iniciado sesión?

Se debe validar que exista una sesión activa, por ejemplo, usando \$_SESSION en PHP. Si la sesión no está iniciada, el sistema debe redirigir al login automáticamente. Esto protege el acceso a páginas privadas, como paneles de usuarios o administradores.

4. ¿Cómo manejarías una sesión de usuario en PHP para que no acceda a páginas protegidas si no ha iniciado sesión?

Primero se llama a session_start() en cada página que requiere seguridad. Luego se revisa si hay una variable de sesión (como el ID del usuario). Si no existe, se redirige al login. Esto asegura que solo usuarios autenticados puedan ver ciertas páginas.

5. ¿Qué buenas prácticas aplicarías al diseñar la base de datos de un sistema de usuarios con niveles de acceso?

- Crear una tabla para los usuarios y otra para los roles o niveles de acceso.
- Relacionarlas con una clave foránea.
- Evitar datos repetidos y asegurar campos como correo único.

Esto hace el sistema más organizado, escalable y seguro.

6. ¿Cuál es la diferencia entre una clave primaria y una clave foránea?

- La clave primaria es un identificador único de cada fila como por ejemplo 'id_usuario'.
- La clave foránea enlaza una fila de una tabla con otra, como por ejemplo (id_rol en la tabla de usuarios).

7. ¿Cómo asegurarías que una contraseña no sea visible en la base de datos?

Se usa una técnica llamada hashing con funciones como password_hash() de PHP. Esto transforma la contraseña en un valor cifrado que no se puede revertir fácilmente. Cuando el usuario inicia sesión, se compara usando password_verify().

8. ¿Para qué se utiliza \$_POST y \$_GET en PHP?

- \$_POST se usa cuando se envía información privada o sensible (como contraseñas, formularios).
- \$_GET se usa para pasar parámetros visibles en la URL, como filtros o búsquedas.

En pocas palabras se elige POST cuando no quieres que la información se vea en el navegador.

9. ¿Qué ventajas aporta usar jQuery en vez de JavaScript puro en un proyecto con muchas interacciones en el frontend?

jQuery permite hacer acciones comunes como animaciones, peticiones AJAX o manipulación del DOM de forma más rápida y sencilla que con JavaScript puro. Es muy útil en proyectos donde se quiere reducir el tiempo de desarrollo y mejorar la experiencia del usuario.

- Con jQuery se pueden enviar datos de un formulario sin recargar la página (AJAX).
- Con jQuery se puede hacer que aparezcan o desaparezcan mensajes de validación.
- Ideal para sitios dinámicos y con muchos formularios.