

Banco de preguntas:

SECCIÓN 1- Conceptuales (Básicas)

1. ¿Qué es una librería estándar en un lenguaje de programación?

Una librería estándar es un conjunto de funciones y herramientas que vienen incluidas en un lenguaje de programación y que permiten realizar tareas comunes sin tener que programarlas desde cero.

2. ¿Por qué es mejor usar funciones del lenguaje que crear todo desde cero?

Es mejor usar funciones del lenguaje porque están optimizadas, probadas, ahorran tiempo y reducen la posibilidad de errores.

3. ¿Qué es un componente en programación?

Un componente es una parte modular del programa que realiza una función específica y puede reutilizarse.

4. ¿Qué diferencia hay entre un componente y una librería?

Un componente es una unidad funcional dentro del programa, mientras que una librería es un conjunto de funciones predefinidas que pueden ser utilizadas por distintos componentes.

5. ¿Qué es JSON?

JSON es un formato ligero de intercambio de datos basado en pares clave–valor, que se puede convertir fácilmente a estructuras de datos en diferentes lenguajes.

6. ¿Por qué JSON es tan utilizado en aplicaciones web?

JSON se usa mucho en aplicaciones web porque es ligero, fácil de leer, compatible con casi todos los lenguajes y permite intercambiar datos entre cliente y servidor de forma sencilla.

7. ¿Qué es una validación de datos?

La validación de datos es el proceso de verificar que los valores ingresados cumplen con las reglas definidas (tipo, rango, formato).

8. ¿Qué puede pasar si no validamos datos en una aplicación?

Si no validamos los datos, se pueden registrar valores incorrectos, generar errores en el programa o permitir vulnerabilidades de seguridad.

SECCIÓN 2: Sobre su programa en PHP

9. ¿Qué funciones de strings utilizaste y para qué?

Se usaron funciones de strings:

limpiarTxt() para sanitizar texto (nombre y puesto).

trim() para eliminar espacios al inicio y fin.

strlen() para verificar longitud mínima del nombre.

10. ¿Qué función usaste para validar el correo y cómo funciona?

Utilicé filter_var() con FILTER_VALIDATE_EMAIL, que verifica si el correo cumple con el formato estándar. Para validar correo se usa validarCorreo(\$correo) en PHP y la función validarCorreo() en JS. Comprueba que el correo tenga el formato correcto.

11. ¿Cómo validaste que la edad fuera correcta?

La edad se valida con validarEdad(\$edad) en PHP y en JS con isNaN() y rango entre 18 y 100.

12. ¿Qué función utilizaste para convertir datos a JSON?

Para convertir datos a JSON se usa convertirAJSON(\$empleados) que internamente utiliza json_encode().

13. ¿Qué función utilizaste para procesar nuevamente el JSON?

Para procesar JSON se usa convertirDesdeJSON(\$jsonData) que internamente utiliza json_decode().

14. ¿Por qué al decodificar JSON es importante elegir entre objeto o arreglo?

Al decodificar JSON es importante elegir objeto o arreglo porque cambia la forma de acceder a los datos: con [] si es arreglo asociativo o -> si es objeto.

15. ¿Cómo calculaste los días transcurridos desde la fecha de ingreso?

Se calculan los días transcurridos usando la función calcularDiasDesdeIngreso(\$fecha) que hace la diferencia entre la fecha de ingreso y la fecha actual.

16. ¿Qué función usaste para trabajar con fechas?

Se usó strtotime() dentro de calcularDiasDesdeIngreso() y validarFechaIngreso() para trabajar con fechas.

17. ¿Cómo calculaste el promedio de edad?

El promedio de edad se calcula sumando todas las edades con \$sumaEdad y dividiendo entre el total de registros (\$totalRegistros).

18. ¿Qué función usaste para contar registros?

Para contar registros se usa count(\$empleados).

19. ¿Qué pasaría si el usuario deja un campo vacío?

Si el usuario deja un campo vacío, la validación de JS evita enviar el formulario y la validación PHP lo detiene mostrando un mensaje de error.

20. ¿Qué parte del programa consideras que es un "componente"?

Consideramos como “componente” el conjunto de funciones en funciones.php (limpieza, validaciones, JSON, cálculo de días), ya que son reutilizables.

SECCIÓN 3: Relación con Librerías

21. ¿Qué categoría de librería utilizaste para manejar texto?

Librería de manejo de texto → limpiarTxt(), trim(), strlen().

22. ¿Qué categoría utilizaste para validaciones?

Librería para validaciones → validarCorreo(), validarEdad(), validarFechaIngreso().

23. ¿Qué categoría utilizaste para manejo de fechas?

Librería de manejo de fechas → strtotime() y calcularDiasDesdelIngreso().

24. ¿Qué categoría utilizaste para trabajar con arreglos?

Librería para arreglos → funciones de PHP como count() y operaciones con \$empleados[].

25. ¿Qué ventaja te dio usar funciones estándar en vez de crear tu propia validación?

La ventaja de usar funciones estándar es ahorrar tiempo, evitar errores y asegurar que las validaciones y conversiones sean correctas.

SECCIÓN 4: Sobre JSON

26. ¿Qué estructura tiene un JSON?

Un JSON tiene estructura de objetos con pares clave–valor y puede contener arreglos.

27. ¿Qué son clave y valor?

Clave = nombre del dato; Valor = información asociada a esa clave.

28. ¿Qué diferencia hay entre JSON y un arreglo asociativo en PHP?

JSON es texto usado para intercambiar datos entre sistemas, mientras que un arreglo asociativo es una estructura de datos interna de PHP.

29. ¿En qué situaciones reales se utiliza JSON?

JSON se usa en APIs, almacenamiento en archivos, intercambio cliente-servidor y bases de datos NoSQL.

30. ¿Qué pasaría si el JSON está mal estructurado?

Si un JSON está mal estructurado, json_decode() no lo procesa y se generan errores en la aplicación.

SECCIÓN 5: Sobre Minecraft (Abstracción)

31. ¿Qué representa el módulo de captura en tu construcción?

El módulo de captura representa el formulario donde se ingresan los datos.

32. ¿Qué representa la biblioteca dentro del mundo?

La biblioteca representa las funciones y librerías que soportan las operaciones (PHP, JS).

33. ¿Cómo representaron las validaciones?

Las validaciones se representan con mecanismos que bloquean o permiten la entrada según las condiciones.

34. ¿Qué parte representa el procesamiento?

El procesamiento está representado por la lógica que transforma y guarda los datos (PHP).

35. ¿Qué parte representa el reporte final?

El reporte final está representado por la tabla HTML que muestra los empleados registrados.

36. ¿Cómo relacionas redstone con lógica de programación?

La redstone representa la lógica de programación (condicionales, validaciones).

37. ¿Qué representa una puerta que se abre solo si la señal es correcta?

Una puerta que se abre solo si la señal es correcta representa una validación condicional.

38. ¿Qué representa el conjunto de módulos conectados?

El conjunto de módulos conectados representa la arquitectura modular del sistema.

39. ¿Qué parte de tu construcción representa un componente reutilizable?

Un componente reutilizable es un módulo de funciones de funciones.php.

40. ¿Cómo representa Minecraft el concepto de arquitectura modular?

Minecraft representa arquitectura modular mediante bloques independientes que se combinan para formar un sistema completo.

SECCIÓN 6: Preguntas de Profundización (Para alumnos fuertes)

41. ¿Podrías haber hecho esta práctica sin usar librerías estándar? ¿Cómo cambiaría el programa?

Sí, se podría hacer sin librerías, pero sería más largo, complejo y propenso a errores.

42. ¿Qué riesgos existen al no validar correos electrónicos correctamente?

Se aceptan datos inválidos, puede fallar el sistema o permitir ataques de inyección.

43. ¿Cómo mejorarías el sistema para hacerlo más profesional?

Mejoras para profesionalizar el sistema: separar lógica en módulos, agregar interfaz más amigable, manejar errores de forma centralizada.

44. ¿Cómo guardarías múltiples empleados en un archivo JSON externo?

Guardar múltiples empleados en un JSON externo usando
file_put_contents("empleados.json", convertirAJSON(\$empleados)).

45. ¿Qué mejoras de seguridad agregarías?

Mejoras de seguridad: sanitizar entradas, validar datos, proteger archivos JSON contra acceso externo.

46. ¿Cómo separarías el sistema en módulos reales?

Separar el sistema en módulos reales: captura (formulario), validación, procesamiento, almacenamiento y reporte.

47. ¿Qué patrón de diseño podrías aplicar aquí?

MVC (Modelo-Vista-Controlador).

48. ¿Cómo escalarías este sistema para 10,000 registros?

Usar base de datos en lugar de JSON, paginación, optimización de consultas.

49. ¿Qué diferencia hay entre procesamiento del lado del cliente y del servidor?

Cliente valida antes de enviar (JS), servidor valida y procesa (PHP).

50. ¿Qué aprendiste realmente sobre componentes y librerías con esta práctica?

Las librerías facilitan tareas repetitivas y los componentes permiten organizar código modular y reutilizable.