

Captura de pantalla del funcionamiento del formulario:

Registro de empleados

Nombre completo:

Correo electrónico:

Edad:

Fecha de ingreso:
 dd/mm/aaaa

Puesto o área:

Registro de empleados

Nombre completo:
 gisell hernandez

Correo electrónico:

Edad:

Fecha de ingreso:
 dd/mm/aaaa

Puesto o área:

Todos los campos son obligatorios.

Registro de empleados

Nombre completo:

Correo electrónico:

Edad:

Fecha de ingreso:
 [calendario]

Puesto o área:

Registrar empleado

Correo inválido.

Registro de empleados

Nombre completo:

Correo electrónico:

Edad:

Fecha de ingreso:
 [calendario]

Puesto o área:

Registrar empleado

Edad inválida (18-100).

Reporte de empleados

Nombre	Correo	Edad	Días desde ingreso	Puesto
Edgar Ibarra	ed@gmail.com	22	9	Jefe
Gisell Hernandez	gis@gmail.com	20	9	Empleado

Total de registros: 2

Promedio de edad: 21

Correos inválidos: 0

[Registrar otro empleado](#)

Definición de librería estándar:

Es un conjunto precompilado de módulos, funciones, clases y subrutinas fundamentales que vienen integradas por defecto en un lenguaje de programación. Su objetivo es simplificar el desarrollo al proporcionar herramientas comunes (entrada/salida, matemáticas, gestión de memoria) sin requerir instalaciones externas, garantizando compatibilidad y eficiencia.

Funciones utilizadas:

Funciones de strings:

`trim()` - Elimina espacios al inicio y final. Se usó para eliminar espacios innecesarios al inicio y final del texto ingresado por el usuario.

`strtolower()` - Convierte texto a minúsculas. Para convertir todo el texto a minúsculas y normalizarlo.

`ucwords()` - Convierte la primera letra de cada palabra en mayúscula. Para poner la primera letra de cada palabra en mayúscula (formatear nombre y puesto).

`strlen()` - Devuelve la longitud de una cadena. Para verificar que el nombre tenga una longitud mínima válida.

`htmlspecialchars()` - Convierte caracteres especiales para evitar inyección HTML. Para evitar inyección de código HTML y mejorar la seguridad.

Funciones de filtros:

`filter_var()` - Valida y filtra datos. Se usó para validar datos usando funciones estándar de PHP.

`FILTER_VALIDATE_EMAIL` - Valida correos electrónicos. Para comprobar que el correo tenga formato válido.

`FILTER_VALIDATE_INT` - Valida enteros. Para verificar que la edad sea un número entero válido.

Funciones de fechas:

`strtotime()` - Convierte fecha a timestamp. Para convertir la fecha ingresada a formato timestamp y poder calcular diferencias.

`time()` - Devuelve timestamp actual. Para obtener la fecha actual del sistema.

`floor()` - Redondea hacia abajo (para calcular días). Para obtener el número entero de días transcurridos.

`date()` - Genera fecha actual formateada. Para establecer la fecha máxima permitida en el formulario.

Funciones de JSON:

`json_encode()` - Convierte arreglo a JSON. Para convertir el arreglo de empleados a formato JSON y guardarla.

`json_decode()` - Convierte JSON a arreglo. Para convertir el JSON nuevamente en un arreglo manipulable en PHP.

`JSON_PRETTY_PRINT` - Formato legible. Para que el archivo JSON sea más legible.

Funciones de arreglos:

`count()` - Cuenta elementos del arreglo. Para contar el total de empleados registrados.

`is_array()` - Verifica si variable es arreglo. Para asegurar que los datos decodificados sean un arreglo válido.

`foreach` - Recorre arreglos. Para recorrer la lista de empleados.

`round()` - Redondea promedio. Para calcular el promedio de edad con dos decimales.

Funciones adicionales:

`file_get_contents()` - Leer archivo JSON.

`file_put_contents()` - Guardar archivo JSON.

`file_exists()` - Verificar existencia de archivo.

`header()` - Redirección.

`exit` - Terminar ejecución.

`session_start()` - Manejo de sesiones.

`isset() / empty()` - Verificar variables.

Ventajas el uso del JSON:

El uso de JSON ofrece múltiples ventajas, ya que es un formato ligero, fácil de leer y escribir tanto para humanos como para máquinas, lo que facilita la comprensión y el mantenimiento de los datos. Su estructura basada en pares clave–valor y arreglos permite organizar información de manera clara y ordenada. Además, es ampliamente compatible con la mayoría de los lenguajes de programación y plataformas, lo que lo convierte en un estándar ideal para el intercambio de datos en aplicaciones web y móviles, especialmente en APIs. También destaca por su eficiencia en el uso de ancho de banda y rapidez en el

procesamiento, siendo más liviano que otros formatos, lo que mejora el rendimiento en la transmisión de información.

Problemas encontrados y solución:

Durante el desarrollo de esta práctica me enfrenté a varios errores ya que se aumento el grado de dificultad, uno de ellos fue el guardar el registro de varios empleados, para conseguir esto se optó por realizar un archivo externo en el que se furan guardando todos los registros de los empleados.

Otro de los problemas fue el convertir el dato de la fecha de string a fecha ya que no entendía la manera en la que esto se hacía, por lo que tuve que realizar una investigación para lograr comprender el funcionamiento de cada una de las partes que componen a esta función. Otro de los errores que más se repitió fueron los enlaces entre los archivos ya que olvidaba ponerlos y al final no sabía en que sección iban ubicados. Estos fueron los problemas principales a los que me enfrenté al realizar esta práctica.

Conclusión personal:

Después de realizar esta práctica, obtuve como aprendizaje conocer para qué funciona JSON, así como el uso de sus librerías, y ampliar mi conocimiento sobre las funciones que existen en JavaScript y PHP, lo cual nos ayuda a facilitar el trabajo al momento de programar, ya que nos ahorra una gran cantidad de tiempo y mejora la funcionalidad de nuestros proyectos. Por ejemplo, en mi caso particular, conocí la función para manejar fechas y todo el procedimiento que se debe llevar a cabo para transformarlas de una cadena de caracteres a un formato de fecha. Del mismo modo, comprendí por qué es necesario transformar la información que se recibe por medio de un formulario a formato JSON, ya que es un lenguaje que nuestras computadoras y servidores pueden interpretar; de lo contrario, la información permanece como una cadena de texto y no se puede procesar correctamente. Este ejercicio fue bastante productivo, ya que abarcó muchos conceptos y temas, permitiéndonos obtener una mayor cantidad de conocimientos que serán muy útiles para las prácticas posteriores.