Prueba de conocimiento en programación:

Lenguaje: PHP, JavaScript, SQL

**Primer componente: Básico.**

**Ejercicio 1: PHP Basics**

Escribe un programa en PHP que tome un número entero como entrada y determine si es un número primo. El programa debe imprimir "Es primo" si el número es primo, y "No es primo" si no lo es.

**Ejercicio 2: JavaScript Basics**

Escribe una función en JavaScript que reciba un arreglo de números como entrada y devuelva la suma de los números pares en el arreglo.

**Ejercicio 3: PHP + MySQL**

Crea una tabla llamada "usuarios" en una base de datos MySQL con los siguientes campos:

* id (entero, clave primaria)
* nombre (cadena de texto)
* correo (cadena de texto)

Escribe un script en PHP que inserte un nuevo usuario en la tabla "usuarios" utilizando datos de entrada del usuario. El programa debe solicitar al usuario que ingrese su nombre y correo electrónico, y luego insertar esos valores en la tabla.

**Ejercicio 4: JavaScript + DOM Manipulation**

Crea una página HTML con un botón. Escribe una función en JavaScript que, al hacer clic en el botón, cambie el color de fondo de la página al azar.

**Ejercicio 5: PHP + MySQL**

Escribe una consulta en PHP para seleccionar todos los usuarios de la tabla "usuarios" cuyo nombre comienza con la letra "A" y ordenarlos alfabéticamente de forma ascendente. Imprime los resultados en forma de una lista ordenada.

**Segundo componente: Orientado a objetos**

**Ejercicio 1: PHP Object-Oriented Programming (OOP)**

Crea una clase en PHP llamada "Empleado" con los siguientes atributos: nombre, salario y puesto. Implementa un método llamado obtenerSalarioAnual() que devuelva el salario anual del empleado (asumiendo que el salario mensual se proporciona).

**Ejercicio 2: JavaScript Advanced Concepts**

Escribe una función en JavaScript que reciba una cadena de texto como entrada y devuelva la misma cadena, pero con las palabras en orden inverso. Por ejemplo, si la entrada es "Hola, cómo estás", la función debería devolver "estás cómo Hola,".

**Ejercicio 3: PHP + MySQL Advanced**

Supongamos que tienes dos tablas en una base de datos MySQL: "productos" y "ventas". La tabla "productos" tiene los campos id, nombre y precio. La tabla "ventas" tiene los campos id, producto\_id, cantidad y fecha.

* Escribe una consulta en PHP que devuelva el nombre del producto más vendido y la cantidad total de ventas para ese producto.

**Ejercicio 4: JavaScript + API Integration**

Utilizando JavaScript, crea una página web que muestre la información climática actual de una ciudad específica utilizando una API de clima (puedes utilizar OpenWeatherMap u otra de tu elección). La página debe mostrar el nombre de la ciudad, la temperatura actual y una descripción del clima.

**Ejercicio 5: PHP + Security**

Escribe una función en PHP que valide si una contraseña cumple con los siguientes criterios:

* Tiene al menos 8 caracteres.
* Contiene al menos una letra mayúscula y una letra minúscula.
* Contiene al menos un número.
* La función debe devolver true si la contraseña cumple con los criterios, y false si no los cumple.

**Tercer componente Modelo Vista Controlador**

**Ejercicio 1: MVC en PHP**

Imagina que estás construyendo un sistema de registro de usuarios en PHP utilizando el patrón MVC. A continuación, se describen las responsabilidades de cada componente:

**Modelo (UsuarioModelo):**

Se encarga de interactuar con la base de datos para realizar operaciones relacionadas con los usuarios (crear, leer, actualizar, eliminar).

* Contiene los métodos crearUsuario($datos), obtenerUsuario($id), actualizarUsuario($id, $datos), eliminarUsuario($id), entre otros.

**Vista (UsuarioVista):**

Se encarga de mostrar la interfaz de usuario para el registro de usuarios.

* Contiene un formulario con campos como nombre, correo electrónico, contraseña, etc.
* Al enviar el formulario, llama al controlador para procesar los datos.

**Controlador (UsuarioControlador):**

Recibe las solicitudes del usuario provenientes de la vista.

* Valida y procesa los datos enviados por la vista.
* Interactúa con el modelo para realizar las operaciones necesarias en la base de datos.
* Retorna las respuestas adecuadas a la vista para mostrar mensajes de éxito o error.

Tu tarea consiste en implementar las clases UsuarioModelo, UsuarioVista y UsuarioControlador utilizando el patrón MVC. Además, debes crear una página que muestre el formulario de registro y que, al enviarlo, utilice el controlador para almacenar los datos del usuario en la base de datos.

**Ejercicio 2: MVC en JavaScript**

Imagina que estás desarrollando una aplicación de lista de tareas en JavaScript utilizando el patrón MVC. A continuación, se describen las responsabilidades de cada componente:

**Modelo (TareaModelo):**

* Representa la estructura de una tarea, con atributos como título, descripción, fecha de creación, etc.
* Se encarga de realizar operaciones relacionadas con las tareas, como agregar una nueva tarea, obtener todas las tareas, marcar una tarea como completada, etc.

**Vista (TareaVista):**

* Muestra la interfaz de usuario para ver y manipular las tareas.
* Muestra una lista de tareas con su título y estado de completado.
* Permite agregar nuevas tareas, marcar tareas como completadas y eliminar tareas.

**Controlador (TareaControlador):**

* Recibe las acciones del usuario provenientes de la vista, como agregar una tarea, marcar una tarea como completada, etc.
* Interactúa con el modelo para realizar las operaciones correspondientes en las tareas.
* Actualiza la vista para reflejar los cambios realizados en las tareas.

Tu tarea consiste en implementar las clases TareaModelo, TareaVista y TareaControlador utilizando el patrón MVC. Además, debes crear una página web que muestre la lista de tareas, permita agregar nuevas tareas, marcar tareas como completadas y eliminar tareas.

Guardar los archivos de manera independiente, y al final realizar un solo comprimido y enviarlo al correo de donde fue enviado este documento.