**Documentación de la Implementación del Backend**

**1. Descripción General**

Este proyecto implementa un formulario en el frontend que valida los datos ingresados por el usuario y luego los envía al servidor mediante una solicitud HTTP POST para su procesamiento. La lógica de validación y el envío de datos se maneja en **JavaScript**, mientras que el procesamiento de los datos en el servidor se realiza con **PHP**.

**2. Archivos Involucrados**

1. **main.js**: Archivo principal que inicializa y ejecuta la validación del formulario.
2. **validacion\_formulario.js**: Contiene la clase ValidacionFormulario que maneja la validación de los datos del formulario y su envío al backend.
3. **envio\_email.php:** Contiene la clase envioEmail, que utiliza PHPMailer para enviar correos electrónicos de confirmación a los usuarios
4. **datosFormulario.php:** Maneja las solicitudes POST para procesar y almacenar los datos del formulario y envía un correo electrónico de confirmación.

**3. Flujo de Trabajo**

**3.1. Inicialización y Validación (main.js)**

* Se importa la clase ValidacionFormulario desde validacion\_formulario.js.
* Se selecciona el formulario y el contenedor del mensaje en el DOM.
* Se instancia ValidacionFormulario con los elementos seleccionados.
* Se llama al método validarFormulario() que inicia la validación del formulario cuando se envía.

**3.2. Clase ValidacionFormulario (validacion\_formulario.js)**

La clase ValidacionFormulario realiza las siguientes tareas:

* **Constructor**: Inicializa los elementos del formulario, el contenedor del mensaje y el mensaje de contestación al usuario.
* **Métodos**:
  + getValores(): Recopila los valores de los campos de texto, email y textarea.
  + getValoresradioButton(): Obtiene el valor del radio button seleccionado.
  + getValorCheckbox(): Obtiene el valor del checkbox seleccionado.
  + validarFormulario(): Valida que todos los campos del formulario estén completos y, si es así, envía los datos al servidor.
  + mostrarAlerta(): Muestra un mensaje de éxito o error según la validación.
  + limpiarFormulario(): Resetea el formulario después de enviarlo.
  + enviarDatos(): Envía los datos al servidor utilizando fetch y maneja la respuesta del servidor.

**3.3. Procesamiento de Datos (datosFormulario.php)**

El archivo datosFormulario.php se encarga de procesar los datos recibidos. Recibe los datos a través de una solicitud POST y los procesa según sea necesario.

El script podría incluir funciones como la validación en el lado del servidor, almacenamiento en una base de datos o envío de correos electrónicos.

**3.4. Envío de email automático (envio\_email.php)**

El archivo envio\_email.php se encarga de enviar un email automático al correo que el usuario especifica en el formulario, el correo solo será enviado en caso de que la consulta SQL sea correcta y se haya ejecutado sin ningún inconveniente.

**4. Ejemplo de Funcionamiento**

* El usuario completa el formulario en la interfaz.
* Al enviar el formulario, validacion\_formulario.js valida los campos.
* Si la validación es exitosa, los datos se envían a datosFormulario.php usando el método POST.
* El servidor procesa los datos y responde con un JSON que indica el éxito o el fallo de la operación.
* En caso de éxito, inserta los datos en la base de datos SQL y envía un email al usuario.

**5. Consideraciones de Seguridad**

* **Validación**: Aunque se realiza la validación en el lado del cliente, es crucial validar nuevamente los datos en el servidor para evitar ataques como la inyección SQL.
* **Manejo de Errores**: Es importante manejar adecuadamente los errores tanto en el cliente como en el servidor para brindar retroalimentación clara al usuario.