

Ejercicio 1: Formularios Básicos con Método GET y POST

1. **Crea un formulario básico** en HTML que envíe información de usuario (nombre, email y edad) a un archivo PHP (`procesar.php`) usando el método `GET`.
2. **Muestra la información** recibida en el archivo `procesar.php` utilizando `print_r($_REQUEST)` y `print_r($_GET)`.
3. Modifica el formulario para usar el método `POST` y verifica cómo cambia el resultado al usar `print_r($_POST)`.

Ejercicio 2: Uso de Distintos Controles de Formulario

1. **Añade diferentes tipos de controles** a un formulario, incluyendo:
 - Caja de texto (`text`).
 - Área de texto (`textarea`).
 - Caja de contraseña (`password`).
 - Casillas de verificación (`checkbox`).
 - Botones radio (`radio`).
 - Menú desplegable (`select` con múltiples opciones).
2. Envía la información a un archivo PHP y **muestra el contenido recibido**.
3. Verifica qué ocurre cuando no se seleccionan opciones en las casillas de verificación o botones radio y en caso de que se produzca un evento indeseado, corrige el error (p.ej envía mensaje de error al usuario)

Ejercicio 3: Manejo de Archivos con `input type="file"`

1. Crea un formulario que permita subir un archivo.
2. Configura el formulario para que envíe el archivo a un archivo PHP usando `enctype="multipart/form-data"` y el método `POST`.
3. En el archivo PHP (`subir.php`), guarda el archivo en una carpeta del servidor y muestra un mensaje indicando si la subida fue exitosa.

Ejercicio 4: Formulario con Múltiples Botones de Envío

1. Crea un formulario con dos botones de envío (`input type="submit"`) y define atributos `name` y `value` distintos para cada uno.
2. En el archivo PHP que procesa el formulario, verifica cuál de los botones fue presionado y muestra un mensaje específico.

Ejercicio 5: Validación de Campos y Gestión de Errores

1. Crea un formulario con campos obligatorios (por ejemplo, nombre y email).
2. Al recibir la información en el archivo PHP, verifica que no estén vacíos y que el email tenga un formato válido.

3. Si la validación falla, muestra un mensaje de error; de lo contrario, muestra la información recibida.

Ejercicio 6. Comprobación de números con `is_numeric()` y `ctype_digit()`

1. Crear un formulario que reciba un valor y comprobar si el valor introducido es un número utilizando `is_numeric()` y `ctype_digit()`.

Ejercicio 7. Comprobación de tipos de datos con funciones `is_`

1. Crear un formulario que reciba diferentes tipos de datos y comprobar el tipo utilizando funciones `is_int()`, `is_float()`, `is_bool()`, `is_string()`, etc.

Ejercicio 8. Validación con funciones `ctype_`

1. Crear un formulario que reciba un valor y validar si el valor es un alfabético (`ctype_alpha()`), alfanumérico (`ctype_alnum()`) o solo contiene dígitos (`ctype_digit()`).

Ejercicio 9. Validación con `filter_var()`

1. Crear un formulario que reciba un correo electrónico y una URL y validar estos datos con `filter_var()`.