

Manual de Estilo de Código

Elemento	Convención	Descripción	Ejemplo
Variables (JS & PHP)	camelCase	mayúscula. Excepción: Las variables de backend (PHP) que interactúan con la BD (como resultados de queries o índices de arreglos) pueden usar snake_case para consistencia con SQL.	nombreUsuario, idHorario, \$totalRegistros
Clases (PHP & JS)	PascalCase	Cada palabra comienza con mayúscula. (Aplica si en el futuro se implementan Clases u Objetos).	SistemaChecador, UsuarioLider
Constantes (PHP)	UPPER_CASE	Todas las letras en mayúscula, separadas por guion bajo.	REGISTROS_POR_PAGINA, HORA_LIMITE
Funciones (JS)	camelCase	Idéntico al de las variables de JavaScript.	iniciarRelojTiempoReal(), buscarSugerencias()
Funciones (PHP)	snake_case	Minúsculas separadas por guion bajo, como se observa en conectar().	obtener_registros_inusuales(), calcular_horas_totales()
Archivos (PHP/JS)	snake_case	Minúsculas separadas por guion bajo.	asignacion_horarios.php, funciones_utiles.js
Parámetros SQL/BD	snake_case	Nombres de tablas y columnas deben ser en minúsculas y usar guion bajo.	id_usuario, hora_inicio, tabla_registros

Formato General y Estructura

A. PHP

1. Apertura y Cierre:

- Siempre usar la etiqueta completa `<?php` y `?>` y nunca la etiqueta corta `<?.`
- Cuando un archivo contiene solo código PHP (como los módulos de funciones o *scripts* que solo manejan lógica POST), es preferible omitir la etiqueta de cierre `?>` al final del archivo para evitar la inyección accidental de espacios en blanco o
- caracteres que causen errores de encabezado.

Manual de Estilo de Código

2. Indentación:

- Usar **4 espacios** para la indentación, nunca tabulaciones.

3. Estructuras de Control:

- La llave de apertura ({) debe estar en la **misma línea** que la declaración de la estructura (if, for, while, function).

Ejemplo:

```
if (isset($_POST['accion'])) {  
    // Lógica aquí  
} else {  
    // Lógica aquí  
}
```

Manual de Estilo de Código

1. Operaciones con Base de Datos:

- La conexión debe establecerse al inicio del *script* usando la función `conectar()`.
- Usar `mysqli_real_escape_string()` para sanear todas las variables de entrada (`$_POST`, `$_GET`) que se usarán en *queries* para prevenir ataques de inyección SQL.

B. JavaScript

1. Indentación y Estructuras:

- Usar 4 espacios para la indentación.
- Las llaves de apertura (`{`) deben estar en la misma línea que la declaración de la función o estructura de control.

2. Punto y Coma:

- Siempre terminar las sentencias con punto y coma (`;`).

3. Uso de `var`, `let`, y `const`:

- Usar `const` por defecto para variables que no serán reasignadas (ej., referencias a elementos del DOM).
- Usar `let` para variables cuyo valor cambiará (ej., contadores).
- Evitar el uso de `var`.

4. Comentarios en Funciones Asíncronas (Fetch/AJAX):

- Los bloques `.then()` y `.catch()` deben tener una indentación clara.

Manual de Estilo de Código

Ejemplo :

```
fetch(`?sugerencias=1&buscar=${query}`)  
  .then(response => response.json())  
  .then(data => {  
    mostrarSugerencias(data);  
  })  
  .catch(error => console.error('Error:', error));
```