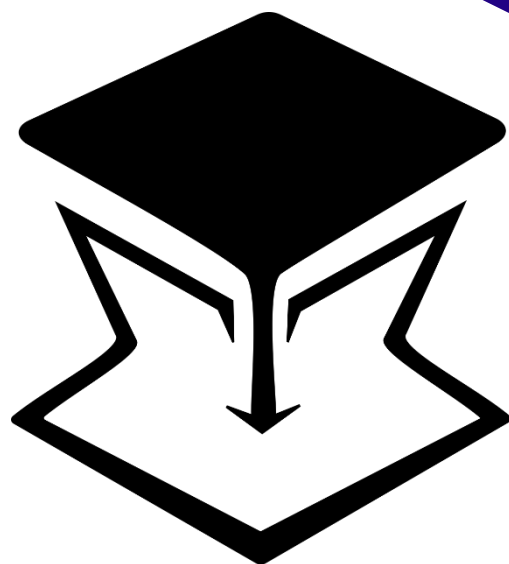




SIGMA

Estándar de Programación SIGMA

3-BYTE Team

Adrián PORTILLO, Glenda FLORES, Franco ROSAS

Laboratorio de Desarrollo de Software

Unidad Académica Río Gallegos

Universidad Nacional de la Patagonia Austral

3-BYTE
TEAM



Este documento refleja los estándares de codificación del lenguaje PHP (PSR) publicada por PHP Framework Interop Group.

El mismo no solo busca definir la nomenclatura de las variables, objetos, métodos y funciones, sino que también hace referencia al orden y legibilidad del código escrito.

Tabla de contenido

<i>Nombres de Ficheros</i>	5
Ejemplo:	5
<i>Extensiones de Ficheros</i>	5
<i>Organización de los Ficheros</i>	5
<i>Ficheros fuente PHP</i>	5
<i>Comentarios de comienzo</i>	5
Ejemplo:	5
<i>Sentencias namespace y use</i>	5
Ejemplo:	6
<i>Declaraciones de Clases e Interfaces</i>	6
Ejemplo:	6
<i>Indentación</i>	6
<i>Longitud de la línea</i>	6
<i>Rompiendo líneas</i>	6
Ejemplo:	7
Comentarios	7
Formatos de los comentarios de implementación	7
Comentarios de bloque	7
Ejemplo:	7
<i>Comentarios de una línea</i>	7
Ejemplo:	7
<i>Comentarios de fin de línea</i>	8
<i>Comentarios de documentación</i>	8
Ejemplo:	8
Declaraciones	8
<i>Cantidad por línea</i>	8
Ejemplo:	8
<i>Inicialización</i>	8
Ejemplo:	9
Sentencias	9
<i>Sentencias simples</i>	9
Ejemplo:	9
<i>Sentencias compuestas</i>	9

Ejemplo:	9
<i>Sentencias if, if-else, elseif</i>	9
Ejemplo:	10
<i>Sentencias for, while, do-while</i>	10
Ejemplo:	10
<i>Sentencias switch</i>	10
Ejemplo:	11
Espacios en blanco y líneas en blanco	11
Convenciones de nombres	11
<i>Clases</i>	11
<i>Métodos</i>	11
<i>Variables</i>	11
<i>Constantes</i>	11
Ejemplos de Código	12
<i>Ejemplo de archivo fuente PHP:</i>	12

Estándar de Programación

Nombres de Ficheros

Los nombres de los ficheros en PHP deben ser claros y descriptivos. Utilizar el formato PascalCase para nombrar clases, y los archivos deben seguir el nombre de la clase que contienen.

Ejemplo:

Si la clase se llama UserManager, el archivo debe ser UserManager.php.

Extensiones de Ficheros

Todos los archivos PHP deben tener la extensión .php

Organización de los Ficheros

Los ficheros deben estar organizados según el módulo o funcionalidad que representan. Deben estar estructurados en directorios claros como controllers/, models/, views/, services/, repositories/.

Ficheros fuente PHP

Cada archivo PHP debe iniciar con la apertura <?php sin cierre de la etiqueta ?>, excepto en archivos que contienen solo HTML. Se recomienda evitar el uso de etiquetas cortas <?.

Comentarios de comienzo

Cada archivo debe incluir comentarios que describan el propósito del archivo, el autor y la última fecha de modificación.

Ejemplo:



Sentencias namespace y use

Se debe definir el namespace al principio del archivo, seguido de las declaraciones use.

Ejemplo:

```
1 namespace App\Controllers;  
2 use App\Models\User;
```

Declaraciones de Clases e Interfaces

Las clases y las interfaces deben seguir la convención PascalCase. Las clases deben comenzar con una letra mayúscula, y no deben utilizarse guiones bajos (_).

Ejemplo:

```
1 class UserManager {  
2     // ...  
3 }
```

Indentación

La indentación debe ser de 4 espacios, no tabulaciones.

Longitud de la línea

Cada línea de código debe tener un máximo de 80 caracteres. Si una línea excede este límite, debe romperse de manera lógica.

Rompiendo líneas

Cuando sea necesario romper una línea larga, alinear las partes divididas para mantener legibilidad.

Ejemplo:

```
1  $userData = $userRepository->findUserByCriteria(  
2      $criteria, $orderBy, $limit  
3  );
```

Comentarios

Formatos de los comentarios de implementación

Los comentarios deben ser claros y concisos, explicando la intención del código, no simplemente repitiendo lo que el código hace.

Comentarios de bloque

Usar comentarios de bloque para explicar secciones amplias de código. Se utilizan `/* */`.

Ejemplo:

```
1  /*  
2      * Esta clase gestiona la autenticación de los usuarios,  
3      * incluyendo login y logout.  
4  */
```

Comentarios de una línea

Para comentarios cortos, usar `//`.

Ejemplo:

```
1  // Verificar si el usuario está autenticado  
2
```

Comentarios de fin de línea

No se recomienda poner comentarios al final de una línea de código, excepto en casos extremadamente simples y claros.

Comentarios de documentación

Se utilizará PHPDoc para documentar clases, métodos y funciones.

Ejemplo:

```
1  /**
2   * Verifica si el usuario está autenticado.
3   *
4   * @param int $userId ID del usuario.
5   * @return bool True si el usuario está autenticado, false en caso contrario.
6   */
7  public function isAuthenticated(int $userId): bool {
8      // ...
9  }
```

Declaraciones

Cantidad por línea

Se debe declarar una única variable por línea.

Ejemplo:

```
1  $nombre = 'Adrián';
2  $edad = 28;
```

Inicialización

Las variables deben inicializarse en el momento de la declaración siempre que sea posible.

Ejemplo:

```
1  $usuario = null;
```

Sentencias

Sentencias simples

Las sentencias simples, como las asignaciones o invocaciones de funciones, deben terminar con un punto y coma.

Ejemplo:

```
1  $usuario = $userManager->getUser();  
2
```

Sentencias compuestas

Las sentencias compuestas deben usar llaves {} aunque contengan una única línea.

Ejemplo:

```
1  if ($usuario) {  
2      $this->login($usuario);  
3  }
```

Sentencias if, if-else, elseif

Colocar el else o elseif en la misma línea que la llave de cierre de if.

Ejemplo:

```
1  if ($usuario) {  
2    // código  
3  } else {  
4    // código  
5  }
```

Sentencias for, while, do-while

Se deben usar siempre llaves {} con bucles, incluso si solo hay una línea de código dentro.

Ejemplo:

```
1  for ($i = 0; $i < 10; $i++) {  
2    echo $i;  
3  }
```

Sentencias switch

Cada case debe terminar con break, a menos que explícitamente se desee el flujo continuo.

Ejemplo:

```
1  switch ($estado) {  
2      case 'activo':  
3          // código  
4          break;  
5      case 'inactivo':  
6          // código  
7          break;  
8      default:  
9          // código  
10 }
```

Espacios en blanco y líneas en blanco

Debe haber una línea en blanco entre bloques de código, métodos y declaraciones de clases para mejorar la legibilidad.

Convenciones de nombres

Clases

Las clases deben usar el formato PascalCase.

Métodos

Los nombres de métodos deben seguir el formato camelCase.

Ejemplo: `getUserById()`

Variables

Las variables deben seguir el formato camelCase.

Ejemplo: `$nombreUsuario`

Constantes

Las constantes deben escribirse en mayúsculas separadas por guiones bajos (_).

Ejemplo: `const MAX_USUARIOS = 100;`

Ejemplos de Código

Ejemplo de archivo fuente PHP:

```
1  <?php
2
3  namespace App\Models;
4
5  class User {
6      private $nombre;
7      private $edad;
8
9      /**
10       * Constructor de la clase User.
11       *
12       * @param string $nombre Nombre del usuario.
13       * @param int $edad Edad del usuario.
14       */
15     public function __construct(string $nombre, int $edad) {
16         $this->nombre = $nombre;
17         $this->edad = $edad;
18     }
19
20     /**
21      * Obtener el nombre del usuario.
22      *
23      * @return string
24      */
25     public function getNombre(): string {
26         return $this->nombre;
27     }
28 }
```

Referencias

- <https://docs.phpdoc.org/> - Documentación PHPDOC
- <https://www.php-fig.org/psr/psr-1/> - Estándar PSR-1
- <https://www.php-fig.org/psr/psr-4/> - Estándar PSR-4
- <https://www.php-fig.org/psr/psr-12/> - Estándar PSR-12