***Guía de estilo de codificación***

Esta guía amplía y extiende el estándar de codificación básica PSR-1.

La objetivo de esta guía es la de reducir la dificultad cuando se lee código de diferentes autores. Lo realiza mediante la enumeración de una serie de reglas común y expresiones sobre cómo dar formato al código PHP.

En el documento original se usa el RFC 2119 para el uso de las palabras MUST, MUST NOT, SHOULD, SOULD NOT y MAY. Para que la traducción sea lo más fiel posible, se traducira siempre MUST como el verbo deber en presente (DEBE, DEBEN), SHOULD como el verbo deber en condicional (DEBERÍA, DEBERÍAN) y el verbo MAY como el verbo PODER.

**1. Visión general**

El código DEBE seguir el estándar PSR-1.

El código DEBE usar 4 espacios como indentación, no tabuladores.

NO DEBE haber un límite estricto en la longitud de la línea; el límite DEBE estar en 120 caracteres; las líneas DEBERÍAN tener 80 caracteres o menos.

DEBE haber una línea en blanco después de la declaración del namespace, y DEBE haber una línea en blanco después del bloque de declaraciones use.

Las llaves de apertura de las clases DEBEN ir en la línea siguiente, y las llaves de cierre DEBEN ir en la línea siguiente al cuerpo de la clase.

Las llaves de apertura de los métodos DEBEN ir en la línea siguiente, y las llaves de cierre DEBEN ir en la línea siguiente al cuerpo del método.

La visibilidad DEBE estar declarada en todas las propiedades y métodos; abstract y final DEBEN estar declaradas antes de la visibilidad; static DEBE estar declarada después de la visibilidad.

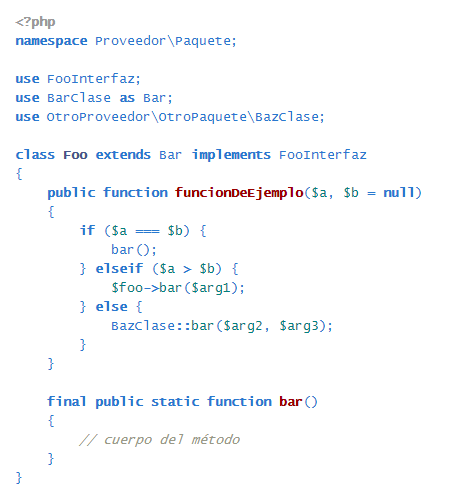
Las palabras clave de las estructuras de control DEBEN tener un espacio después de ellas, las llamadas a los métodos y las funciones NO DEBEN tenerlo.

Las llaves de apertura de las estructuras de control DEBEN estar en la misma línea, y las de cierre DEBEN ir en la línea siguiente al cuerpo.

Los paréntesis de apertura en las estructuras de control NO DEBEN tener un espacio después de ellos, y los paréntesis de cierre NO DEBEN tener un espacio antes de ellos.

**1.1. Ejemplo**

Este ejemplo incluye algunas de las siguientes reglas a modo de visión general rápida:



**2. General**

**2.1 Codificación estándar básica**

El código DEBE seguir las normas expuestas en el estándar PSR-1.

**2.2 Ficheros**

Todos los ficheros PHP DEBEN usar el final de línea Unix LF.

Todos los ficheros PHP DEBEN terminar con una línea en blanco.

La etiqueta de cierre ?> DEBE ser omitida en los ficheros que sólo contengan código PHP.

**2.3. Líneas**

NO DEBE haber un límite estricto en la longitud de la línea.

El límite flexible de la línea DEBE estar en 120 caracteres; los correctores de estilo automáticos DEBEN advertir de ésto, pero NO DEBEN producir errores.

Las líneas NO DEBERÍAN ser más largas de 80 caracteres; las líneas más largas de estos 80 caracteres DEBERÍAN dividirse en múltiples líneas de no más de 80 caracteres cada una.

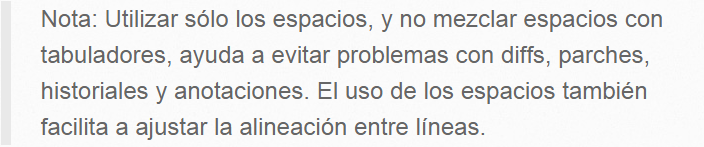
NO DEBE haber espacios en blanco al final de las líneas que no estén vacías.

PUEDEN añadirse líneas en blanco para mejorar la lectura del código y para indicar bloques de código que estén relacionados.

NO DEBE haber más de una sentencia por línea.

**2.4. Indentación**

El código DEBE usar una indentación de 4 espacios, y NO DEBE usar tabuladores para la indentación.



**2.5. Palabras clave y true/false/null.**

Las Palabras clave de PHP DEBEN estar en minúsculas.

Las constantes de PHP true, false y null DEBEN estar en minúsculas.

**3. Espacio de nombre y declaraciones use**

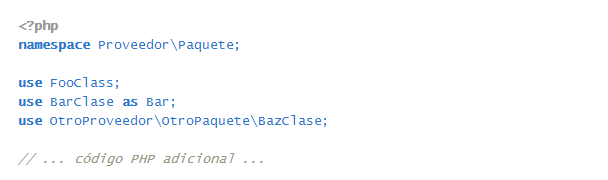
Cuando esté presente, DEBE haber una línea en blanco después de la declación del namespace.

Cuando estén presentes, todas las declaraciones use DEBEN ir después de la declaración del namespace.

DEBE haber un use por declaración.

DEBE haber una línea en blanco después del bloque de declaraciones use.

Por ejemplo:



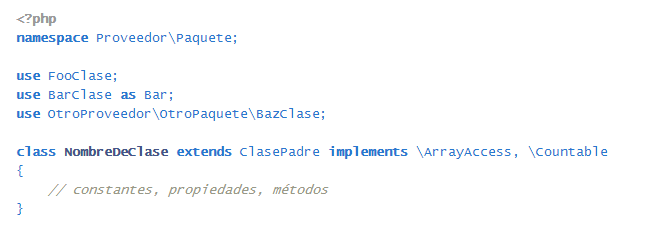
**4. Clases, propiedades y métodos**

El término "clase" hace referencia a todas las clases, interfaces o traits.

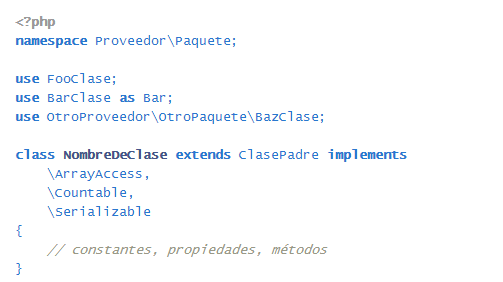
**4.1. Extensiones e implementaciones**

Las palabras clave extends e implements DEBEN declararse en la misma línea del nombre de la clase.

La llave de apertura de la clase DEBE ir en la línea siguiente; la llave de cierre DEBE ir en la línea siguiente al cuerpo de la clase.



La lista de implements PUEDE ser dividida en múltiples líneas, donde las líneas subsiguientes serán indentadas una vez. Al hacerlo, el primer elemento de la lista DEBE estar en la línea siguiente, y DEBE haber una sola interfaz por línea.



**4.2. Propiedades**

La visibilidad DEBE ser declarada en todas las propiedades.

La palabra clave var NO DEBE ser usada para declarar una propiedad.

NO DEBE declararse más de una propiedad por sentencia.

Los nombres de las propiedades NO DEBERÍAN usar un guión bajo como prefijo para indicar si son privadas o protegidas.

Una declaración de propiedas tendrá el siguiente aspecto.

