***Estándar de codificación básica PSR-1***

***1. Visión general***

* Los archivos DEBEN utilizar solamente las etiquetas <?php y <?=.
* Los archivos DEBEN emplear solamente la codificación UTF-8 sin BOM para el código PHP.
* Los archivos DEBERÍAN declarar cualquier estructura (clases, funciones, constantes, etc,...) o realizar partes de la lógica de negocio (por ejemplo, generar una salida, cambio de configuración ini, etc,...) pero NO DEBERÍAN hacer las dos cosas.
* Los espacios de nombres y las clases DEBEN cumplir el estándar PSR-0.
* Los nombres de las clases DEBEN declararse en notación StudlyCaps. [^1]
* Las constantes de las clases DEBEN declararse en mayúsculas con guiones bajos como separadores CONSTANTE\_DE\_CLASE.
* Los nombres de los métodos DEBEN declararse en notación camelCase. [^2]

**Notas:**

[^1] StudlyCaps, es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúscula.

[^2] camelCase, es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúsculas exceptuando la primera palabra.

***2. Archivos***

***2.1. Etiquetas PHP***

El código PHP DEBE utilizar las etiquetas largas <?php ?> o las etiquetas cortas para imprimir salida de información <?= ?>; NO DEBE emplear otras variantes.

***2.2. Codificación de caracteres***

El código PHP DEBE utilizar codificación UTF-8 sin BOM.

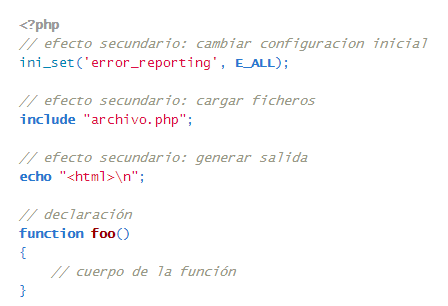
***2.3. Efectos secundarios***

Un archivo DEBERÍA declarar estructuras (clases, funciones, constantes, etc,...) y no causar efectos secundarios, o DEBERÍA ejecutar partes de la lógica de negocio, pero NO DEBERÍA hacer las dos cosas.

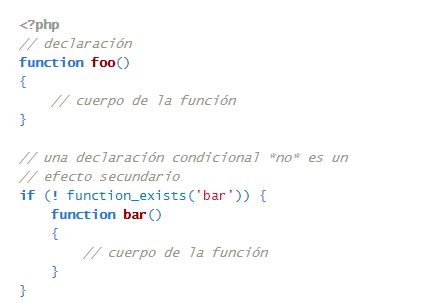
La frase "efectos secundarios" significa: que la ejecución de la lógica de negocio no está directamente relacionado con declarar clases, funciones, constantes, etc, simplemente la de incluir el archivo.

"Efectos secundarios" incluyen, pero no se limitan a: generar salidas, uso explícito de requiere o include, conexiones a servicios externos, modificación de configuraciones iniciales, enviar errores o excepciones, modificar variables globales o estáticas, leer o escribir un archivo, etc.

El siguiente ejemplo muestra un archivo que incluye las dos: declaraciones y efectos secundarios; Un ejemplo de lo que debe evitar:



El siguiente ejemplo es el de un archivo que contiene declaraciones sin efectos secundarios; Un ejemplo que puede seguir:



**3. Espacios de nombres y nombres de las Clases**

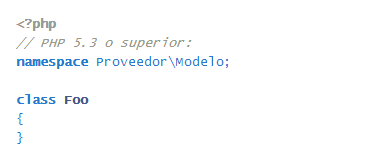
Los espacios de nombres y las clases DEBEN seguir el estándar PSR-0.

Esto significa que cada clase estará en un fichero independiente y está dentro de un espacio de nombres en al menos un nivel: un nombre de proveedor de nivel superior.

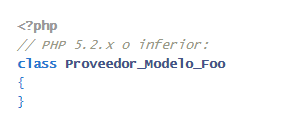
Los nombres de las clases DEBEN declararse con notación StudlyCaps. [^1]

El código escrito para PHP 5.3 o superior DEBE hacer un uso formal de los espacios de nombres.

Por ejemplo:



El código escrito para PHP 5.2.x o inferior DEBERÍA emplear una convención de pseudo-espacios de nombres con prefijos en los nombres de las clases con el formato Proveedor\_.

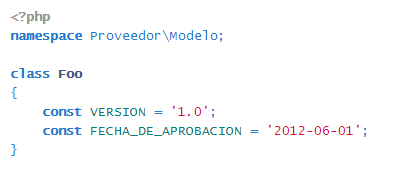


**4. Constantes de Clases, Propiedades y Métodos**

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

**4.1. Constantes**

Las constantes de las clases DEBEN declararse siempre en mayúsculas y separadas por guiones bajos. Por ejemplo:



**4.2. Propiedades**

Esta guía evita intencionadamente cualquier recomendación respecto al uso de las notaciones $StudlyCaps, $camelCase, o $guion\_bajo en los nombres de las propiedades. [^1] [^2]

Cualquiera que sea la convención en nomenclatura, DEBERÍA ser utilizada de forma coherente con un alcance razonable. Este alcance PUEDE ser a nivel de proveedor, a nivel de paquete, a nivel de clase o a nivel de método.

**4.3. Métodos**

Los nombres de los métodos DEBEN declararse en notación camelCase(). [^2]