

## Estándares de Codificación EPE

Los estándares a utilizar en el desarrollo de este sistema por parte de **Chilombia**, serán los siguientes:

### 1) Estándares para programación en PHP:

- **PSR-0:** Este es el primero de los estándares y nos especifica las siguientes reglas:

#### Puntos Obligatorios

- Los namespaces y las clases deben tener la siguiente estructura (\*)
  - Cada namespace debe tener un namespace superior ("Vendor name").
  - Cada namespace puede tener tantos sub-namespaces como se quiera.
  - Los nombres de los namespaces o clases deben ser separados por un guion bajo (\_).
  - Todos los archivos deben tener la extensión .php.
  - Los nombres de los namespaces o clases deben ser ordenadas alfabéticamente.
- **PSR-1, PSR-2, PSR-3:**

#### Convenciones

- Los archivos deben utilizar solamente
- Los archivos sólo deben utilizar una codificación UTF-8.
- Las funciones sólo deben retornar un sólo valor.
- Los nombres de espacio y las clases deben seguir las reglas del PSR-0.
- Los nombres de clases deben ser escritas utilizando la técnica **StudlyCaps**.
- Las constantes deben ser definidas en MAYÚSCULAS y utilizando guion bajo (\_) como separador.
- Métodos y funciones deben ser escritos utilizando la técnica **camelCase**.
- Debemos de validar que la función que vamos a crear no exista utilizando la función **function\_exists()**.
- Las llaves deben de estar abajo solamente en las clases y métodos.
- La indentación debe ser con un tabulador establecido a 4 espacios.
- Las constantes true, false y null deben ser escritos en minúsculas.
- El número de caracteres por línea deben ser de 80 columnas aunque también esta aceptado que sean hasta 120.



Empacando para estudiar



- Ya no debes utilizar la palabra reservada var para declarar una propiedad, debes utilizar public, private, static o protected.
- Debe haber un espacio después de cada estructura de control (if, for, foreach, while, switch, try...catch, etc.).

Información extraída de:

- 1) <https://www.codejobs.biz/es/blog/2013/02/19/estandares-de-codificacion-en-php-psr0-psr1-psr2-y-psr3>
- 2) [https://github.com/jatubio/5minutos\\_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4](https://github.com/jatubio/5minutos_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4)

## 2) Estándares para programación en JavaScript:

### - General

- Los ficheros tienen que ser nombrados en minúsculas utilizando la forma 'underscore'.
- El objeto global se declara según la forma 'UPPER\_CASE' con guión bajo como separador y precedido del signo de dolar (\$).
- Los subobjetos principales que categorizan las funcionalidades, se declaran según la forma 'UPPER\_CASE' con guión bajo como separador.
- Las variables tienen que ser nombradas utilizando la forma 'camelCase'.
- Los constructores tienen que ser nombrados utilizando la forma 'StudlyCaps'.
- Las constantes, de haberlas, tienen que ser nombradas en 'UPPER\_CASE' con guión bajo como separador.
- Las propiedades tienen que ser nombradas utilizando 'camelCase'.
- Los métodos tienen que ser nombrados utilizando 'camelCase'.

### - Nombres de Fichero

- Los ficheros se nombrarán siempre en minúsculas para evitar errores con aquellas plataformas 'sensitive-case'. Los ficheros tienen que concluir con la extensión .js, y no

deben incluir signos de puntuación excepto – (guión medio) o \_ (guión bajo), siendo preferible este último a la hora de separar palabras.

#### - Objeto Global

- El objeto global es aquel que envuelve a toda la aplicación y que evita la contaminación del entorno global. Para denotar su jerarquía, se declara en mayúsculas precedido del signo de dolar (\$) y su nombre suele coincidir con el de la aplicación.

#### - Subojetos globales

- Los subojetos globales son aquellas subdivisiones principales del proyecto que actúan como categorías superiores. Suelen segmentar el código en bibliotecas o bloques más manejables.

#### - Variables

- Las variables tienen que declararse utilizando la forma 'camelCase'. No deben ir precedidas de un guion bajo para indicar que se tratan de variables privadas.
- Cuando se declaren más de una variable de forma consecutiva, solo se utilizará una única sentencia 'var'.
- Las variables que vayan a contener una instancia de jQuery, se identificarán **anteponiéndoles un signo de dolar (\$)**.

#### - Constructores

- Los constructores (también llamados clases) tienen que nombrarse siguiendo la forma 'StudlyCaps' para diferenciarlos a simple vista del resto de estructuras.



Empacando para estudiar



### - Constantes

- Las constantes se nombran en mayúsculas y separando palabras con un guión bajo. Evitamos el uso del comando @const para evitar errores en Internet Explorer.

### - Propiedades

- Las propiedades se declaran utilizando 'camelCase' al igual que las variables.

### - Métodos

- Los métodos de los objetos tienen que ser nombrados utilizando 'camelCase', al igual que las variables y propiedades.

Información extraída de:

- 1) <http://www.etnassoft.com/2012/10/23/guia-de-estilo-nomenclatura-en-archivos-javascript/>
- 2) <https://github.com/ogranada-xk/js-coding-standards>