Estándar de Codificación - PHP.

Propósito	Guiar a los desarrolladores en la escritura de códigos PHP.
Estándar de Conteo	• Contar cada línea física como un LOC. • No contar etiquetas de inicio y fin de PHP. • No contar líneas vacías y líneas de comentarios. • No contar líneas con una sola llave.
Lenguaje	Define el lenguaje (Idioma) para desarrollar los códigos.
Formato General	`Ingles.`
Nombrado de Archivos	Define la nomenclatura de nombrado de los archivos físicos donde se desarrollan los códigos.
Formato General – Clase	• `El nombre debe ser igual al nombre de la clase especificada en el diagrama de clases.` • `Cada clase debe realizarse en un solo archivo de código.`
Formato General – Módulo – Vista	• `frmA` • `A: nombre del caso de uso (Palabras concatenadas).`
Formato General – Módulo – Modelo	`Nombre del caso de uso (Palabras concatenadas).`
Otros	• `Evitar la mezcla de códigos PHP con códigos HTML5, a menos que sea absolutamente necesario.` • `Cada bloque de código debe de estar escrito en archivos independientes.`
Encabezados	Define la estructura de los encabezados que deben de tener los archivos donde se desarrollan los códigos.
Formato General	`/*******************************************************` `Name:` `Autor name:` `Modification autor name:` `Creation date:` `Modification date:` `Description:` `*******************************************************/`
Estructura de Contenidos	Define, de manera resumida, la estructura de los contenidos de archivos de códigos.
Formato General – Clase	`//Etiqueta de inicio de código PHP` `//Importar paquetes/clases` `//Declaración de clases` `  //Declaración de atributos/propiedades` `  //Declaración de constructor de clase` `  //Declaración de comportamientos/métodos` `  //Declaración de destructor de clase` `//Etiqueta de fin de código PHP`
Formato General – Módulo	`//Etiqueta de inicio de código PHP` `//Importar paquetes/clases` `//Declaraciones de sentencias`

	`//Etiqueta de fin de código PHP`
Instrucciones de Reúso	Define como se prepara el código para reúso. Provee la declaración de formatos, valores, tipos y límites de parámetros. Provee alertas de valores no permitidos, condiciones de sobrecarga u otras condiciones que puedan resultar operaciones con resultados inapropiados.
Formato General	`public function verifyPasswordToDataBase($vpassword)` `//Purpose: It verifies the user password in the database` `//Limitations: The connection to database must be established` `//Returns -1, 0 or 1, according to checkout in database`
Identificadores	Define de manera descriptiva los nombres de todas las variables, constantes y otros identificadores. Evitar las abreviaturas o variables de una sola letra.
Formato General	`public class clspUser{};` `//Principal class name` `public class clscUser{};` `//Collection class name` `public $idUser;` `//Attributes/properties (Variable) class` `const cPi = 3.14159;` `//Attributes/properties (Constant) class` `public function __construct(){}` `//Constructor class` `public function addToDataBase(){}` `//Behavior/Method class` `public function verifyPasswordToDataBase($vpassword){}` `//Behavior/Method (Parameter) class` `public function __destruct(){}` `//Destructor class` `function showDataUser(){}` `//Function` `$vuserName = "ecabrera";` `//Variable in Behavior/Method class or Module` `$vcivilStatus= new clspCivilStatus();` `//Variable in Behavior/Method class or Module` `define("cPi"', "3.14159");` `//Constant in Module`
Comentarios	Define la forma de documentar códigos para que los lectores puedan comprender las operaciones. Los comentarios deben de explicar tanto el propósito como las conductas de los códigos.
Buenos Comentarios	`public $idIndividual;` `//Set/get the individual identifier.` `public function addToDataBase()` `//It adds a new individual to database` `while ($vrow=$vmySql->ObtenerArrayFilas())` `//Have all records been read?` `$vsum = 0;` `//Set the variable sum to zero`
Malos Comentarios	`public $idIndividual;` `//Individual identifier.` `public function addToDataBase()` `//It adds a individual` `while ($vrow=$vmySql->ObtenerArrayFilas())` `//Verifies if is end rows` `$vsum = 0;` `//Sum is equal to zero`
Espacios en Blanco	Escribir códigos con suficiente espacio de tal forma que no estén amontonados. Separar cada sección de código con al menos un espacio.
Formato General	`public function __get($vproperty)` `//Get value property` `{` `if( isset($vproperty) ){` `//doesn´t the property exist?`

	throw new Exception("Property doesn't exist: $vvalue"); } else{ //It returns the property value return $this->vproperty; } } public function __set($vproperty, $vvalue) //Set value property { if(isset($vproperty)){ //doesn´t the property exist? throw new Exception("Property doesn't exist: $vvalue"); } else{ //It sets the property value $this->vproperty=$vvalue; } }
Sangría	Define la forma de poner sangrías en cada nivel de llaves desde su apertura. La apertura y cierre de llaves deben de estar alineadas unas con otras.
Formato General	try{ //It sets user password string (No coded) $this->password=$this->passwordString; //It sets sql statement in order to add new user $vsql ="INSERT INTO c_user(id_user, id_userType, fldname, fldpassword, fldpasswordString) "; $vsql.="VALUES('" . $this->idUser . "'"; $vsql.=", " . $this->userType->idUserType; $vsql.=", '" . $this->name . "'"; $vsql.=", '" . md5($this->password) . "'"; $vsql.=", '" . $this->passwordString . "')"; //It opens the connection to database $vmySql= new MySql(); $vmySql->AbrirConexion(0); if ($vmySql->EjecutarSql($vsql)){ //was the sql statement executed whitout problems? if ($vmySql->ObtenerNumeroFilasAfectadas()!=1){ //Was there a problem adding the user to database? throw new Exception ("Ocurrió un error al registrar los datos del usuario, intente de nuevo", 0); } } else{ throw new Exception ("Ocurrió un error al registrar los datos del usuario", -1); }

	`` //It closes the connection to database $vmySql->CerrarConexion(); //It frees memory to CPU unset($vsql, $vmySql); //It returns 1, user sucessfully added return 1; } catch (Exception $vexcepcion){ //It catches exception //It returns exception code catched return $vexcepcion->getCode(); } ```
Escritura	Define la forma de escritura de las palabras que componen las variables y las constantes.
Formato General	``` const cPi = 3.14159; $varea= cPi * ($vradio * $vradio); $vtotalArea+=$varea; ```

