**STANDART DE CODIFICACIÓN**

**EMPRESA-TIPO PRODUCTO – PERSONA – TALLA -PRODUCTO – ORDEN DE COMPRA- BODEGA**

|  |  |
| --- | --- |
| Purpose | Guia de implementación para programas desarrollados en lenguaje PHP |
| Counting Standard | * No contar las líneas en blanco ni comentarios en una o multiples lineas * Ser coherente con lo que se pone en cada línea física. |
| Program Headers | Comenzar todos los programas con una cabecera descriptiva. |
| Header Format | **Clases**  /\*\*  \* Nombre de la clase: clsNombreClase  \* Función: Obtener el nombre de la clase  \* Creado por: juan García Fecha: 01-marz-2016  \*/  **Metodos**  /\*\*  \* Nombre de la Metodo: NombreMetodo  \* Función: Obtener el nombre de la clase  \* Retorna : Una Cadena  \* Creado por: juan García Fecha: 01-marz-2016  \* Modificado por:  \*/ |
| Contents | Proporcionar un resumen de los contenidos. |
| Contents  Example | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /\* Contents:  /\* Incluir librerias  /\* Declaración de la clase  /\* Definición de atributos  /\* Declaración del constructor  /\* Declaración de setters and getters  /\* Declaración de metodos |
| Reuse Instructions | * Describir cómo se utiliza el programa. Proporcionar el formato de declaración, los valores de parámetros y tipos, y los límites de los parámetros. * Proporcione advertencias de valores ilegales, condiciones de desbordamiento, u otras condiciones que podrían dar lugar a un funcionamiento inadecuado. |
| Reuse Example | /\* Recursos: Mensajes deben ser genéricos. (duda) |
| Identifiers | Utilice nombres descriptivos de todas las variables, nombres de funciones, constantes y otros identificadores. Evite las abreviaturas o variables de una sola letra. |

|  |  |
| --- | --- |
| Identifier Example | ----------------------------------------------------------------  ------------------------**Capa cliente**----------------------------  ----------------------------------------------------------------  **// Ejemplo identificadores**  Ejemplo de cómo deben iniciar las variables según su alcance.  **Publicas:** Inician en pub - pub $pubStrEsPublica  **Privadas:** Inician en prv - $prvStrEsPrivada  **Parámetros de métodos:** Inician con pv $pvStrParametroValor  **Locales dentro de un método/función:** Inician con v - $vStrLocal  **/\* Ejemplo de** **variables**  **/\* variables**  a. El Nombre debe hacer referencia a lo que representa la variable  b. El nombre de las variables se compone de tres (3) partes, debe tener el estilo camelcase, sin tildes y sin espacio en blanco entre ellas.    Partes constitutivas del nombre de las variables   * **xxx** alcance de la variable. * **yyy** referencia al tipo de dato. * **descripción** explicación corta de la información que contiene la variable   **Ejemplo:**  · Nombre teórico de una variable: **xxxYyyDescripcion**  · Nombre de la variable que almacena la compañía dentro de una clase: $**pubIntIdCia**  · Declaración de una variable local  **/\*Declaración correcta**  **private $prvStrNotas;**  **/\*Declaración incorrecta**  **private $notas;**  /\* más ejemplos de variables:  pubInt – número pequeño o entero - $pubIntNumEntero  pubLng – Enteros largos $pubLngEnteroLargo  PubStr – cadenas medianas o largas $pubStrCadenaLarga  pubChr – cadenas cortas - $pubChrCadenaCorta  pubBln – Booleanas - $pubBlnVerdaderoOFalso  pubDtm - Fecha – $pubDtmDescripcion    **/\* Clases:**  . El nombre de las clases se compone de dos (2) partes, deben utilizar el estilo camelCase con la primera letra en minúscula, sin tildes y sin espacio en blanco entre ellas.  Partes constitutivas del nombre de las clases   * **xxx cls-** Identificador de una clase * **descripción** explicación corta de la información que contiene la clase.     **Ejemplo:**  · Nombre teórico de una Clase: **xxxDescripcion**  · Nombre de una clase persona que son clientes de una compañía:  clsClientes.php    **/\*Declaración correcta**  **<?php**  **class** clsClientes **{**  **}**  **?>**    **/\*Declaración incorrecta**  **<?php**  **class c**lientes **{**  **}**  **?>**  **/\* Formularios:**  . El nombre de los formularios se compone de dos (2) partes, deben utilizar el estilo camelCase con la primera letra en minúscula, sin tildes y sin espacio en blanco entre ellas.  Partes constitutivas del nombre de un formulario   * **xxx frm-** Identificador de una formulario * **descripción** explicación corta de la información que contiene la clase.   **Ejemplo:**  · Nombre teórico de una Clase: **xxxDescripcion**  · Nombre de un formulario que se utiliza para ingresar los datos de los clientes de una compañía:  frmClientes  frmClientes.php  **/\* Funciones:**  . El nombre de una función o método dentro de una clase o formulario, se compone de una (1) parte, deben utilizar el estilo camelCase con la primera letra en minúscula, sin tildes y sin espacio en blanco entre ellas.  . La descripción debe de contener un verbo infinitivo.  Partes constitutivas del nombre de una función o método.   * **descripción** explicación corta de la información que contiene la función.   Ejemplo de declaración de una función:  · Nombre teórico de una función: d**escripción**  · Nombre de una función que calcula la edad de una persona al ingresar su fecha de nacimiento:  calcularEdad  **/\*Declaración correcta**  **public function calcularEdad($****pvDtmFechaNacimiento) {**  **return $pvDtmFechaNacimiento;**  **}**    **/\*Declaración incorrecta**  **public function calcularedad($pvDtmFechaNacimiento) {**  **return $pvDtmFechaNacimiento;**  **}**    ----------------------------------------------------------------  ----------------------Capa base de datos -----------------------  ----------------------------------------------------------------  **Nombre de los** **procedimientos almacenados**.  . El nombre de las procedimientos almacenados se compone de tres (3) partes, inician con el identificador sp, seguido de guion bajo, la descripción del funcionamiento, seguido de guion bajo y la sentencia SQL que se va a aplicar.  **Ejemplo:**  sp\_clientes\_leer  sp\_clientes\_insertar  **/\*Declaración correcta**  **Create procedure** sp\_clientes\_leer**()**  **Begin**  **End**  **/\*Declaración incorrecta**  **Create procedure** clientes\_leer**()**  **Begin**  **End**  **Parámetros** **de los procedimientos almacenados.**  . El nombre de las Parámetros se compone de dos (2) partes, inician con el identificador p, seguido de guion bajo y la descripción  **Parámetros :** Inician con p + Guion bajo + descripción  Ejemplo:  **/\*Declaración correcta**  **p\_razon\_social**  **/\*Declaración incorrecta**  razonsocial  **Variables de los procedimientos almacenados.**  . El nombre de las variables se compone de dos (2) partes, inician con el identificador v, seguido de guion bajo y la descripción  **Locales:** Inician con v + Guion bajo + descripción  Ejemplo:  **/\*Declaración correcta**  **v\_razon\_social**  **/\*Declaración incorrecta**  Razón\_social  **/\* Funciones:**  . El nombre de las funciones se compone de dos (2) partes, inician con el identificador f, seguido de guion bajo y la descripción  Inician con f + Guion bajo + descripción  Ejemplo:  **/\*Declaración correcta**  f\_remover\_hora  **/\*Declaración incorrecta**  remuevehora  **/\*Nombre de tablas**  . El nombre de las tablas se compone de tres (3) partes, inician con el identificador t, un numero de consecutivo, seguido de guion bajo y la descripción.  **Ejemplo:**  Nombre de una tabla compañía  T001\_compania  **/\*Nombre de los campos de la tabla**  . El nombre de las columnas de una tabla se compone de tres (3) partes, inician con el identificador f, el número de consecutivo de la tabla, seguido de guion bajo y la descripción.  **Ejemplo:**  Nombre de los campos de la tabla t001\_compania  F001\_id\_compania  F001\_descripcion  /\* Otros/ Generales     1. Tamaño del nombre de las clases, métodos, objetos de clases, menor o igual a 30 caracteres. 2. Tamaño del nombre de los objetos de base de datos, menor a 28 caracteres. 3. No se pueden utilizar tildes, ni la ñ al nombrar los objetos. 4. No se debe utilizar la combinación de palabras en inglés y en español al asignar un nombre en el proyecto. Debe ser en inglés o en español. |
| Comments | * Documentar el código para que el lector pueda comprender su funcionamiento. * Los comentarios deben explicar el objeto y el comportamiento del código. * Comento declaraciones de variables para indicar su propósito. * Los comentarios para los bucles, ciclos, clases y métodos, deberán estar contenido en las líneas predecesoras. |
| Good Comment | // Ejemplo de un buen comentario de una sola linea y multiples lineas.  v\_razon\_social //Almacena la razón social del tercero |
| Bad Comment | // Ejemplo de un mal comentario de una sola linea y multiples lineas.  //Almacena la razón social del tercero  v\_razon\_social |
| Major Sections | Precede principales secciones del programa por un comentario de bloque que describe el proceso que se realiza en la siguiente sección |

|  |  |
| --- | --- |
| Example | /\* Comentario correcto  /\*  \* Almacena la razón social del tercero  \* y concatena la sucursal del tercero cliente  \*/  /\* Comentario incorrecto  Por qué solo debería de ir en una sola línea.  /\*  \* Almacena la razón social del tercero  \*/ |
| Blank Spaces | * Cada programa separado se debe construir con al menos un espacio. |
| Indenting | * La indentación o tabulación debe ser de 4 espacios respetando la jerarquía de la declaración de las líneas de código. |
| Indenting  Example | // Ejemplo de una buena indentación  class puntoDosClass extends personaClass {  public function calcularEdad($edad) {  return $edad;  }  }  // Ejemplo de una mala indentación  class puntoDosClass extends personaClass {  public function calcularEdad($edad) {  return $edad;  }  } |
| Capitalization | /\* Constantes  . El nombre de las variables se compone de cuatro (4) partes, deben el estilo camel case con la primera letra en minúscula, sin tildes y sin espacio en blanco entre ellas.  Partes constitutivas del nombre de las constantes   * xxx alcance de las constantes. * yyy referencia al tipo de dato. * Const Identificador de constante. * descripción explicación corta de la información que contiene la constante. |
| Capitalization Example | // Ejemplo de Capitalización  Ejemplo de una constante privada:  · Nombre teórico de una Constante: xxxYyyConstDescripcion  · Nombre de la constante que almacena la cantidad de días que tiene una semana:  $prvIntConstNumDiasSemana = 7 |