

Instituto Tecnológico Superior Zacatecas Norte

División de Informática

**Carrera de Ingeniería en Sistemas
Computacionales**

Proceso de Software en Equipo.

Tema: Estándar de codificación.

Alumno: Eduardo García Delgado.

N.C: 15010199

Versión 0.1
11/10/2019.

Comentarios.	<p>Cuando se vaya a comentar una línea, el comentario se debe de poner arriba de la línea de código que deberá describir y dejando un espacio entre el comentario y la sintaxis correspondiente. Ejemplos:</p> <p>Para html:</p> <pre> <!-- Método [1] → <div> </div> </pre> <p>Para php:</p> <pre> // este es un ciclo for for(\$i=0;\$i<count(\$arrayC);\$i++) { } </pre> <p>Nota: también se puede hacer uso de comentarios múltiples los cuales serán de la siguiente manera:</p> <pre> /* este es un ciclo for que sirve para verificar cuales de los alumnos ya fueron seleccionados para que se les asigne un asesor */ for(\$i=0;\$i<count(\$arrayC);\$i++) { } </pre> <p>Para javascript:</p> <pre> // Método para validar campos function ValidarAltaProd() { } </pre> <p>Nota: también se puede hacer uso de comentarios múltiples los cuales serán de la siguiente manera:</p> <pre> /* este es un método que sirve para verificar cuales de los campos ya fueron llenado, de igual forma si alguno falta, se mandará una alerta al usuario. */ function ValidarAltaProd() { } </pre>
--------------	---

	<p>Para css:</p> <pre>/* Definición de etiquetas */ #etiqueta { font-family: fantasy; color: white; }</pre> <p>Nota: también se funciona como comentario múltiple al igual que en php y javascript</p> <p>Ejemplo de cómo no se usará:</p> <p>Para html:</p> <pre><div> <!-- Método [1] --> </div></pre> <p>Para php:</p> <pre>for(\$i=0;\$i<count(\$arrayC);\$i++) // este es un ciclo for { }</pre> <p>Para javascript:</p> <pre>function ValidarAltaProd() //Método para validar campos { }</pre> <p>Para css:</p> <pre>#etiqueta /* Definición de etiquetas */ { font-family: fantasy; color: white; }</pre>
Uso de llaves.	<p>Las llaves sólo serán utilizadas en el caso de php, css y javascript, su uso será ubicándose debajo de la declaración del método correspondiente. Ejemplo:</p> <p>Para php:</p> <pre>function sumar(\$i,\$j) { }</pre> <p>Para javascript:</p>

	<pre>function ValidarAltaProd() { }</pre> <p>Para css:</p> <pre>#debajoEncabezado { font-family: fantasy; color: white; }</pre> <p>Ejemplos de cómo no se usará:</p> <p>Para php:</p> <pre>function sumar(\$i,\$j) { }</pre> <p>Para javascript:</p> <pre>function ValidarAltaProd() { }</pre> <p>Para css:</p> <pre>#debajoEncabezado { font-family: fantasy; color: white; }</pre>
Declaración de variables.	<p>Para declarar una variable se debe hacer línea por línea cada variable, deben de empezar con una mayúscula y pueden contener caracteres como guión bajo (_) y números. Si el nombre de la variable es muy largo se puede resumir con las cuatro primeras letras de la palabra, de igual forma si se compone de dos o más palabras el nombre será de las 3 primeras letras de una palabra seguida de un guión bajo y las 3 primeras letras del segundo nombre. Ejemplo:</p> <p>Para php:</p> <pre>\$Nombre = 'Jose'; \$Edad = 10; \$Num_Con = '15010199';</pre> <p>Para javascript:</p> <pre>var Nombre = 'Jose'; var Edad = 10; var Num_Con = '15010199';</pre>

Declaración de clases.

Para empezar una clase html, se debe de ubicar con la siguiente estructura:

```
<!doctype html>
<html lang="en">

    <head>
        <title> ... </title>
        ....
    </head>

    <body>
        ....
    </body>

</html>
```

Para empezar una clase php, se debe de ubicar con la siguiente estructura:

```
<?php
    ....
?>
```

Para empezar una clase css, se debe de ubicar con la siguiente estructura:

```
#etiqueta_1
{
    ....
}

#etiqueta_2
{
    ....
}

...

#etiqueta_N
{
    ...
}
```

	<p>Para empezar una clase javascript, se debe de ubicar con la siguiente estructura:</p> <pre>function F1() { ... } function F2() { ... } ... function FN() { ... }</pre>
Declaración de métodos.	<p>Los métodos su declaración es similar que las variables se deben de empezar con una mayúscula y pueden contener caracteres como guión bajo (_) y números. Si el nombre del método es muy largo se puede resumir con las cuatro primeras letras de la palabra, de igual forma si se compone de dos o más palabras el nombre será de las 3 primeras letras de una palabra seguida de un guión bajo y las 3 primeras letras del segundo nombre. Ejemplo:</p> <p>Para php / javascript:</p> <pre>function Sumar (\$i, \$j) { } function Agr_Est () { }</pre>
Nomenclatura de fórmulas.	<p>Cuando se va a realizar una expresión matemática, se deben de usar los espacios para ver con claridad los operadores y operandos. Ejemplo:</p> <p>Suma = X + P;</p> <p>AreaTrian = (B * H) / 2;</p>

<p>Indentación.</p>	<p>La indentación se aplica con una tabulación después de la posición de la llave del método o clase, así para todas las líneas de código que pertenezcan a dicho método o clase. Ejemplo;</p> <pre><?php for (\$i = 0; \$i < 10; \$i ++) { if (\$i == 7) { echo ('Es un siete'); } } ?></pre>
<p>Concatenación.</p>	<p>Si se necesita imprimir una variable de php en código html, necesitamos ejecutar php dentro de donde queremos imprimir la variable con '<?php ?>', aquí sí se puede de manera lineal. Ejemplo:</p> <pre><a href=" ../B1.php?ID=<?php echo \$ID;?>&Nombre=<?php echo \$Nombre2;?>"></pre> <p>Dentro de php, se utilizarán espacios entre la cadena a imprimir, el caracter '.' y la variable acompañada por el símbolo '\$'. Ejemplo:</p> <p>Para php:</p> <pre>echo 'El nombre es '. \$Nombre;</pre>
<p>Espacios en blanco.</p>	<p>Entre cada método o sentencia habrá un espacio en blanco con la finalidad de mostrarlo de manera clara. Ejemplo:</p> <pre><?php for (\$i = 0; \$i < 10; \$i ++) { if (\$i == 7) { echo ('Es un siete'); } } do { echo (NO es un siete'); \$i++; } while (\$i != 7); ?></pre>

<p>Líneas modificadas.</p>	<p>En html, para conocer si se modificó una línea, después del cierre con el caracter ">" se deberá dejar un espacio de tabulación y colocar un pequeño comentario "<! -- M -->" indicando que esa línea fue modificada. Ejemplo:</p> <pre><title> T&iacute;tulo Conocido </title> <! -- M --></pre> <p>En php y javascript, para conocer si se modificó una línea, " se deberá dejar un espacio de tabulación al final de la línea y colocar un pequeño comentario "// M" indicando que esa línea fue modificada. Ejemplo:</p> <p>Para php:</p> <pre>echo 'Linea nueva'; //M</pre> <p>Para javascript:</p> <pre>alert ("Linea nueva"); //M</pre> <p>En css, para conocer si se modificó una línea, " se deberá dejar un espacio de tabulación al final de la línea y colocar un pequeño comentario "/* M */" indicando que esa línea fue modificada. Ejemplo:</p> <pre>height: 500px; /* M */</pre>
<p>Líneas añadidas.</p>	<p>En html, para conocer si se añadió una línea, después del cierre con el caracter ">" se deberá dejar un espacio de tabulación y colocar un pequeño comentario "<! -- A -->" indicando que esa línea fue añadida. Ejemplo:</p> <pre><title> T&iacute;tulo Conocido </title> <! -- A --></pre> <p>En php y javascript, para conocer si se añadió una línea, " se deberá dejar un espacio de tabulación al final de la línea y colocar un pequeño comentario "// A" indicando que esa línea fue añadida. Ejemplo:</p> <p>Para php:</p> <pre>echo 'Linea nueva'; //A</pre> <p>Para javascript:</p> <pre>alert ("Linea nueva"); //A</pre> <p>En css, para conocer si se añadió una línea, " se</p>

	<p>deberá dejar un espacio de tabulación al final de la línea y colocar un pequeño comentario “/* A */” indicando que esa línea fue añadida. Ejemplo:</p> <p>height: 500px; /* A */</p>
--	--