| | |
|---|---|
| | **UNIVERSIDAD DON BOSCO**<br>**FACULTAD DE ESTUDIOS TECNOLÓGICOS**<br>**COORDINACIÓN DE COMPUTACIÓN Y MÓVILES** |
| **CICLO I-2023** | **Desarrollo de Aplicaciones Web**<br>**con Software Interpretado en el Servidor**<br>**Guía de práctica No. 2**<br>**Estructuras de Control – Sentencias condicionales** |

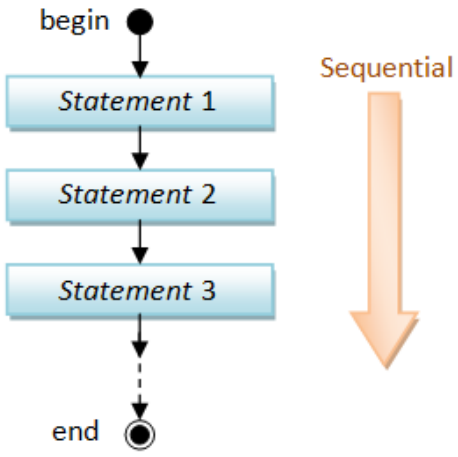
## I. OBJETIVOS

En esta guía de práctica se espera que los estudiantes logren:

- Adquirir amplio dominio en el uso de las sentencias condicionales *if*.
- Utilizar correctamente todas las variantes de la sentencia *if* (*if-else*, *if-else if-else*, *if* anidados).
- Tener conocimiento completo del funcionamiento y utilización de una sentencia *switch*.
- Lograr la habilidad de manejar la característica del efecto de caída en cascada de la sentencia *switch*.
- Desarrolle la capacidad de utilizar eficazmente matrices para almacenar valores y acceder a sus valores en los scripts PHP.
- Sea capaz de utilizar estructuras de control repetitivas para trabajar con datos procedentes de una matriz escalar o asociativa.
- Adquiera el dominio de matrices unidimensionales y multidimensionales utilizando ciclos o lazos repetitivos anidados.

## II. INTRODUCCION TEORICA

Los primeros ejemplos de programación que suelen utilizarse están compuestos por una serie de instrucciones que se ejecutan de forma secuencial. A medida que se va a avanzando en un curso uno cae en la cuenta que lo más habitual es que los problemas que se nos plantean casi nunca se pueden resolver con una ejecución secuencial. Es importante dominar y conocer cierto tipo de instrucciones que permitan modificar el flujo de ejecución de las instrucciones. Estas instrucciones permitirán controlar el flujo lógico en que se ejecutarán las instrucciones dentro de un script. Específicamente, podremos utilizar estructuras de control condicionales que nos permitirán bifurcar el flujo del programa dependiendo de si una condición se evalúa como verdadera o falsa, o repetir un bloque de instrucciones un número determinado de veces para realizar de forma más eficiente y elegante un proceso.




Las estructuras de control se dividen en sentencias condicionales, selectivas y repetitivas. Vamos a examinar todos estos tipos de instrucciones durante este curso.

# Sentencias condicionales

Las sentencias condicionales se pueden clasificar en dos grupos:

a) Sentencias condicionales *if*.
   i. Sentencia if simple.
   ii. Sentencia if-else.
   iii. Sentencia if-elseif-else.

b) Sentencia selectiva *swicht*.

**Sentencia if**

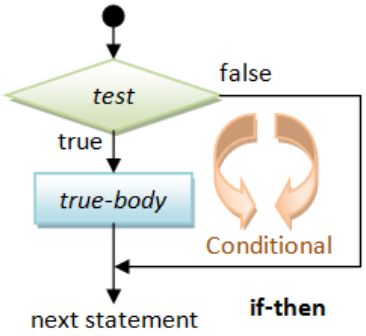
La sentencia ***if*** permite evaluar una expresión condicional y decidir, en base al resultado, si se ejecuta o no un bloque de código. Si la evaluación resulta verdadera (*true*) se ejecuta el bloque de instrucciones y si resulta falsa (*false*) no se ejecuta.
PHP proporciona una estructura *if* que es similar a la de C:

Sintaxis básica:

```
<?php
    if(expresion) //sentencia;
?>
```

Sintaxis extendida:

```
<?php
  if(expresion){
     //bloque de sentencias;
  }
?>
```




La sintaxis básica muestra la construcción de una sentencia ***if*** cuando esta contiene una sola instrucción. La sintaxis extendida muestra una sentencia ***if*** con varias instrucciones dentro del bloque que encierran las llaves del ***if***.

Como se describe en la sección sobre expresiones, `expresion` se evalúa a su valor condicional (*boolean*). Si expresion se evalúa como TRUE, PHP ejecutará la sentencia, y si se evalúa como FALSE, la ignorará y continuará con la siguiente instrucción después del ***if***.

El siguiente ejemplo mostraría: $a es mayor que $b, si $a fuera mayor que $b. De no ser así no mostraría nada:

```
<?php
  if($a > $b) print "$a es mayor que $b";
?>
```

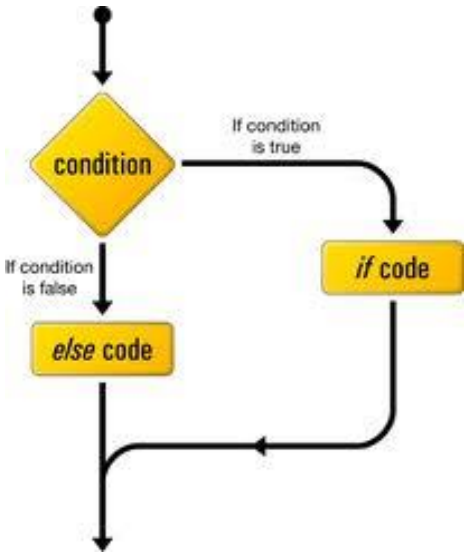
A menudo, se desea tener más de una sentencia ejecutada de forma condicional. Por supuesto, no hay necesidad de encerrar cada sentencia con una cláusula **if**. En vez de eso, se pueden agrupar varias sentencias en un grupo de sentencias. Por ejemplo, este código mostraría $a es mayor que \$b, si $a fuera mayor que $b, y entonces asignaría el valor de $a a $b:

```
<?php
  if($a > $b) {
            print "$a es mayor que $b";
            $b = $a;
        }
?>
```

Las sentencias ***if*** se pueden anidar indefinidamente dentro de otras sentencias ***if*** o ***else***, lo cual proporciona una flexibilidad completa para ejecuciones condicionales en las diferentes partes de tu programa.

**Sentencia if-else**

Hay situaciones en las que se desea ejecutar una o varias sentencias si se cumple una cierta condición, y ejecutar un bloque de sentencias distintas si la condición no se cumple. En estos casos es útil la sentencia *if ...**else***. De esta forma se extiende una sentencia ***if*** para ejecutar una o varias sentencias en caso de que la expresión en la sentencia ***if*** se evalúe como FALSE. Por ejemplo, el siguiente código mostraría $a (su valor) es impar si $a%2 resultara evaluado como 1 o $a es par si fuera evaluado como 0. Note que solamente se pueden obtener esos valores como resultado:




```
<?php
  if($a % 2){
           print "$a es impar";
        }
        else{
           print "$a es par";
        }
?>
```

La sentencia ***else*** se ejecuta solamente si la expresión ***if*** se evalúa como FALSE.

**Sentencia if-elseif-else**

La sentencia ***elseif*** es una combinación de ***if*** y ***else***. Al igual que un ***else***, extiende una sentencia ***if*** para ejecutar una sentencia diferente en caso de que la expresión ***if*** original se evalúa como FALSE. No obstante, a diferencia del ***else***, ejecutará esa expresión alternativa solamente si la expresión condicional ***elseif*** se evalúa como TRUE. Un bloque ***if*** puede contener múltiples ***elseif***, pero únicamente un ***else***. El bloque de instrucciones bajo un ***elseif*** se ejecutará, únicamente si la condición ***if*** y todas las instrucciones ***elseif*** anteriores devuelven FALSE.

Por ejemplo, el siguiente código mostraría a es mayor que b, a es igual a b o a es menor que b:

```
<?php
 if($a > $b){
     echo "$a es mayor que $b";
 }
 elseif($a == $b){
     echo "$a es igual que $b";
 }
 else{
     echo "$a es menor que $b";
 }
?>
```

Puede haber varios ***elseif*** dentro de la misma sentencia ***if***. La primera expresión ***elseif*** (si hay alguna) que se evalúe como TRUE se ejecutaría. En PHP, también se puede escribir '*else if*' (con dos palabras) y el comportamiento sería idéntico al de un '*elseif*' (una sola palabra).

La sentencia ***elseif*** se ejecuta sólo si la expresión ***if*** precedente y cualquier otra expresión ***elseif*** precedente se evalúan como FALSE, y la expresión ***elseif*** actual se evalúa como TRUE.

**Sintaxis alternativa de la sentencia condicional**

PHP ofrece una forma alternativa de construcción de una sentencia condicional. Esta forma alternativa reemplaza el uso de la llave de apertura del bloque ***if*** por dos puntos (:) y la llave de cierre del bloque por la sentencia ***endif***.

Ejemplo:

```
<?php
   if($num % 2 == 0):
      echo $num . " es par";
   else:
      echo $num . " es impar";
?>
```

**Operador condicional**

En PHP se puede utilizar un operador ternario utilizado con frecuencia en el lenguaje C++. Este es el operador "**?:**", o también llamado **operador condicional**. Este es un operador ternario que se utiliza para simplificar la escritura de una sentencia condicional de tipo if…else, cuando esta contiene únicamente una sentencia en el bloque if y una sentencia en el bloque else.

La sintaxis de este operador es la siguiente:

```
(expresion_condicional) ? expresion1 : expresion2 ;
```

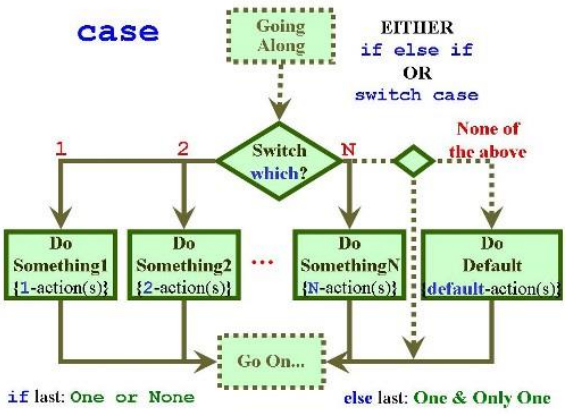
Debe tener en cuenta que si la `expresión_condicional` se evalúa como **TRUE**, se ejecutará `expresion1` y si se evalúa **FALSE**, se ejecutará `expresion2`. Como las expresiones devuelven un valor, usted puede perfectamente asignar la instrucción completa en una variable, como se muestra en el siguiente ejemplo:

```
$condicion = ($nota >= 6.0) ? "Aprobado" : "Reprobado" ;
```

Si la expresión condicional resulta verdadera la variable `$condicion` tendrá el valor de "Aprobado", si se evalúa como falsa, entonces el valor asignado a `$condicion` será `"Reprobado"`.

**Sentencia switch**

La sentencia ***switch*** es similar a una sentencia ***if*** con varias instrucciones ***elseif***. La diferencia está en que en un ***if*** la condición se evalúa tanto por la sentencia ***if*** como por los ***elseif*** que esta contenga. En cambio, en un ***switch***, la condición se evalúa solamente una vez, luego el valor obtenido se compara con una serie de valores de prueba predefinidos. Cada uno de estos valores se presenta con una sentencia ***case*** dentro del ***switch***. Cuando uno de estos valores coincide con el evaluado por la condición se ejecuta el bloque de instrucciones dentro de ese case de forma secuencial. En caso de que ninguno de los valores



de prueba coincida con el devuelto por la condición se ejecutará el bloque de instrucciones bajo la sentencia ***default***, si es que está presente, si no lo está, entonces no se ejecutará ninguna instrucción en el ***switch***.

Nota: Tener en cuenta que al contrario que otros lenguajes de programación, ***continue*** se aplica a ***switch*** y funciona de manera similar a ***break***. Si se tiene un ***switch*** dentro de un bucle y se quiere continuar con el paso siguiente en el bucle externo, usar ***continue 2.***

Los siguientes dos ejemplos son dos modos distintos de escribir el mismo programa, uno usa una serie de sentencias ***if***, y el otro usa la sentencia ***switch***:

Utilizando if:

```
<?php
     if ($i == 0) {
         echo "$i es igual a 0";
     }
     elseif ($i == 1) {
                   echo "$i es igual a 1";
     }
     elseif ($i == 2) {
               echo "$i es igual 2";
     }
?>
```

Utilizando switch:

```
<?php
  switch ($i) {
      case 0:
                   print "i equals 0";
                   break;
      case 1:
       print "i equals 1";
       break;
      case 2:
       print "i equals 2";
       break;
  }
?>
```

Es importante entender cómo se ejecuta la sentencia ***switch*** para evitar errores. La sentencia ***switch*** se ejecuta línea por línea (realmente, sentencia a sentencia). Al comienzo, no se ejecuta código. Sólo cuando se encuentra una sentencia ***case*** con un valor que coincide con el valor obtenido en la evaluación de la expresión ***switch***, PHP comienza a ejecutar las sentencias. PHP continúa ejecutando las sentencias hasta el final del bloque switch, o la primera vez que se encuentre una sentencia ***break***. Si no se escribe una sentencia break al final de una lista de sentencias case, PHP seguirá ejecutando las sentencias del siguiente ***case***. Por ejemplo:

```
<?php
  switch ($i) {
     case 0:
               print "i es igual a 0";
     case 1:
               print "i es igual a 1";
     case 2:
         print "i es igual a 2";
```

```
 }
?>
```

Aquí, si $i es igual a 0, ¡PHP ejecutaría todas las sentecias print! Si $i es igual a 1, PHP ejecutaría las últimas dos sentencias print y sólo si $i es igual a 2, se obtendría la conducta 'esperada' y solamente se mostraría 'i es igual a 2'. Así, es importante no olvidar las sentencias break (incluso aunque pueda querer evitar escribirlas intencionadamente en ciertas circunstancias).
En una sentencia switch, la condición se evalúa sólo una vez y el resultado se compara a cada sentencia case. En una sentencia elseif, la condición se evalúa otra vez. Si tu condición es más complicada que una comparación simple y/o está en un bucle estrecho, un switch puede ser más rápido.
La lista de sentencias de un case puede también estar vacía, lo cual simplemente pasa el control a la lista de sentencias del siguiente case.

```
<?php
  switch ($i) {
     case 0:
     case 1:
     case 2:
               print "i es menor que 3, pero no negativo";
         break;
     case 3:
               print "i es 3";
  }
?>
```

Un caso especial es el default. Este "case" coincide con todo lo que no coincidan los otros case. Por ejemplo:

```
<?php
  switch ($i) {
     case 0:
         print "i es igual a 0";
               break;
     case 1:
               print "i es igual a 1";
         break;
     case 2:
               print "i es igual a 2";
         break;
     default:
               print "i no es igual a 0, 1 o 2";
  }
?>
```

La expresión case puede ser cualquier expresión que se evalúe a un tipo simple, es decir, números enteros o de punto flotante y cadenas de texto. No se pueden usar aquí ni arrays ni objetos a menos que se conviertan a un tipo simple.

**break**
***break*** escapa de las estructuras de control iterante (bucle) actual *for*, *while*, o *switch*.
***break*** acepta un parámetro opcional, el cual determina cuántas estructuras de control hay que escapar.

**Sintaxis alternativa de la sentencia switch**

También la sentencia ***switch*** puede utilizarse con una sintaxis alternativa de dos puntos. La forma de construir una sentencia *switch* con sintaxis alternativa, se muestra a continuación:

```
<?php
switch($i):
    case 0:
        echo "$i es igual a 0";
        break;
    case 1:
        echo "$i es igual a 1";
        break;
    case 2:
        echo "$i es igual a 2";
        break;
    default:
        echo "$i no es igual a 0, 1 ni 2";
endswitch;
?>
```

## III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

| No. | Material | Cantidad |
|---|---|---|
| 1 | Guía de práctica #2: Estructuras de control – Sentencias condicionales | 1 |
| 2 | Computadora con Xampp instalado y funcionando correctamente | 1 |
| 3 | Editor Sublime Text 3 o DevPHP | 1 |
| 4 | Memoria USB o disco flexible | 1 |

## IV. PROCEDIMIENTO

Indicaciones: Asegúrese de digitar el código de los siguientes ejemplos que se presentan a continuación. Tenga en cuenta que ni Sublime Text, ni PHP Designer son compiladores; solamente son editores de código PHP, por lo tanto, los errores de sintaxis los podrá observar únicamente hasta que se ejecute el script al cargar la página en el navegador de su preferencia.

**Ejemplo 1: Se desea calcular el salario o sueldo neto de algún empleado mediante el uso de un formulario de ingreso de la información necesaria para realizar el cálculo. El formulario debe solicitar el nombre del empleado, su sueldo base y, si realiza horas extras, debe ingresarse el número de horas extras trabajadas y el salario por hora extra a pagar. Al enviar los datos del formulario debe mostrarse de forma ordenada toda la información en una boleta de pago, mostrando como dato final el sueldo neto o líquido a pagar, tomando en cuenta las horas extras. En este formulario de ingreso se ha utilizado validación en el cliente de la entrada de datos en los campos de texto del formulario, haciendo uso de JavaScript no intrusivo (investigar sobre este tema, si no está claro).**

**Archivo 1: salario.html**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Preferencia de restaurantes</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,user-scalable=no,initial-
scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
```

```
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <link rel="stylesheet" href="css/salario.css" />
    <link rel="stylesheet" href="css/formoid-solid-purple.css" />
    <script src="js/disabledtextfields.js"></script>
    <script src="js/inputFilter.js"></script>
    <script src="js/modernizr.custom.lis.js"></script>
    <script src="js/prefixfree.min.js"></script>
</head>
<body>
<header id="inset">
    <h1>Salario empleado</h1>
</header>
<section>
<article>
<div id="empleado">
<form action="salario.php" method="POST" name="salario" id="salario"
class="formoid-solid-purple">
<div class="title">
    <h2>Cálculo de salario</h2>
</div>
<div class="element-name">
    <label class="title"></label>
    <div class="nameFirst">
        <input type="text" name="empleado" id="txtempleado" maxlength="50"
placeholder="Nombre empleado"
allowed="ABCDEFGHIJKLMNÑOPQRSTUVWXYZabcdefghijklmnñopqrstuvwxyzÁÉÍÓÚáéíóúÜü"
messageid="alfabeticchars" class="large" />
        <span class="icon-place"></span>
    </div>
    <span id="alfabeticchars">Solo acepta caracteres alfabéticos</span>
</div>
<div class="element-number">
    <label class="title"></label>
    <div class="item-cont">
        <input type="text" name="sueldobase" id="sueldobase" maxlength="12"
placeholder="Sueldo base" allowed="0123456789." messageid="numbersonly"
class="large" />
        <span class="icon-place"></span>
    </div>
    <span id="numbersonly">Solo acepta números y punto decimal</span>
</div>
<div class="element-checkbox">
    <label class="title">Horas extras:</label>
    <div class="column column1">
        <input type="checkbox" name="hextra" id="hextrasi" value="no" />
        <span>Habilitar</span>
    </div>
</div>
<div class="element-number">
    <label class="title"></label>
    <div class="item-cont">
        <input type="text" name="numhorasex" id="numhorasex" maxlength="2"
disabled="disabled" placeholder="Número de horas" allowed="0123456789"
messageid="integersonly" />
        <span class="icon-place"></span>
    </div>
    <span id="integersonly">Solo acepta números enteros</span>
</div>
<div class="element-number">
    <label class="title"></label>
    <div class="item-cont">
        <input    type="text"    name="pagohextra"    id="pagohextra"    size="12"
maxlength="6"
```

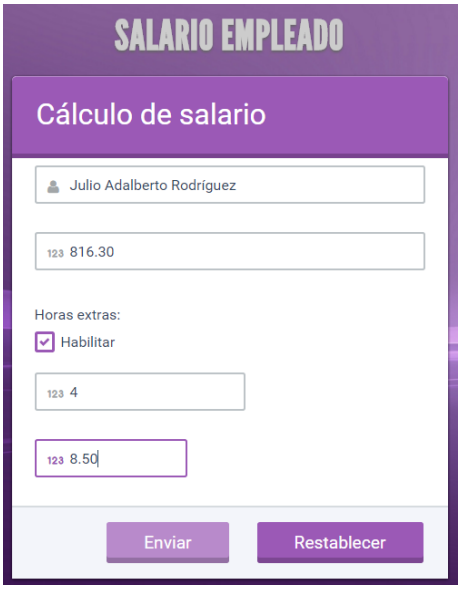
```
disabled="disabled" placeholder="Pago hora extra" allowed="0123456789."
messageid="othernumbersonly" />
        <span class="icon-place"></span>
    </div>
    <span id="othernumbersonly">Solo acepta números y punto decimal</span>
</div>
<div class="submit">
    <input type="submit" name="enviar" id="enviar" value="Enviar" />
    <input type="reset" name="reset" id="reset" value="Restablecer" />
</div>
</form>
</div>
</article>
</section>
</body>
</html>
```

**Archivo 2: salario.php**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Calcular Salario</title>
    <meta charset="utf-8" />
    <meta  name="viewport"  content="width=device-width,user-scalable=no,initial-
scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
    <link rel="stylesheet" href="css/salario.css" />
    <link rel="stylesheet" href="css/links.css" />
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<header id="inset">
    <h1>Detalle del salario</h1>
</header>
<section>
<article>
<?php
    if(isset($_POST['enviar'])){
        $empleado = isset($_POST['empleado']) ? $_POST['empleado'] : "";
        $sueldobase = isset($_POST['sueldobase']) ? $_POST['sueldobase'] : "";
        if(isset($_POST['hextra'])){
            $horasextras = isset($_POST['numhorasex']) ? $_POST['numhorasex'] :
"0";
            $pagohextra = isset($_POST['pagohextra']) ? $_POST['pagohextra'] : "0";
            $sueldohextras = $horasextras * $pagohextra;
        }
        else{
            $horasextras = 0;
            $sueldohextras = 0;
        }
        $sueldoneto = $sueldobase + $sueldohextras;
        echo "<div>\n<h3>Boleta de pago</h3>\n";
        echo "<table>\n";
        echo       "\t<tr>\n\t\t<th       colspan=\"2\">\n\t\t\tDetalle       del
pago\n\t\t</th>\n\t</tr>\n";
        echo   "\t<tr>\n\t\t<td>\n\t\t\tEl   empleado   es:   \n\t\t</td>\n\t\t<td
class=\"detail\">\n\t\t\t",$empleado,"\n\t\t</td>\n\t\t\t</tr>\n";
        echo  "\t<tr>\n\t\t<td>\n\t\t\tEl  sueldo  base  es:  \n\t\t</td>\n\t\t<td
class=\"detail\">\n\t\t\t\$ ", $sueldobase, "\n\t\t</td>\n\t</tr>\n";
        echo "\t<tr>\n\t\t<td>\n\t\t\tLas horas extras son: \n\t\t</td>\n\t\t<td
class=\"detail\">\n\t\t\t",$horasextras,"\n\t\t</td>\n\t</tr>\n";
        echo    "\t<tr>\n\t\t<td>\n\t\t\tEl    pago    por    hora    extra    es:
\n\t\t</td>\n\t\t<td      class=\"detail\">\n\t\t\t\$      ",      $sueldohextras,
"\n\t\t</td>\n\t</tr>\n";
```

```
        echo    "\t<tr>\n\t\t<th>\n\t\t\tEl    sueldo    neto    devengado    es:
\n\t\t</th>\n\t\t<th       class=\"number\">\n\t\t\t\$       ",       $sueldoneto,
"\n\t\t</th>\n\t</tr>\n";
        echo "</table>\n</div>\n";
    }
?>
<a href="salario.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Ingresar</span>
    <span class="a-btn-slide-text">otro empleado</span>
    <span class="a-btn-slide-icon"></span>
</a>
</article>
</section>
</body>
</html>
```

Resultado en el navegador haciendo uso del servidor web:





**Ejemplo 2: El siguiente script PHP muestra cómo utilizar la función date() para manipular la fecha y hora del sistema. Debe notar que se está estableciendo a nivel de código la zona horaria adecuada para la ubicación del país. Se hace uso de la instrucción *switch* para asignar los valores del día, el mes y la hora del sistema.**

**Archivo 1: fechahora.php**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Fecha y hora del Sistema</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,user-scalable=no,initial-
scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <link rel="stylesheet" href="css/fechahora.css" />
</head>
<body>
```

```
<?php
   //Establecer la zona horaria para el servidor web,
   //de modo que concuerde con la hora del país donde
   //se está ejecutando el script.
   date_default_timezone_set("America/El_Salvador");
   $dia=date("w");
   switch($dia){
      case 0: $dia="Domingo";break;
      case 1: $dia="Lunes";break;
      case 2: $dia="Martes";break;
      case 3: $dia="Miércoles";break;
      case 4: $dia="Jueves";break;
      case 5: $dia="Viernes";break;
      case 6: $dia="Sábado";break;
   }
   $mes=date("n");
   switch($mes){
      case 1:  $mes="Enero";break;
      case 2:  $mes="Febrero";break;
      case 3:  $mes="Marzo";break;
      case 4:  $mes="Abril";break;
      case 5:  $mes="Mayo";break;
      case 6:  $mes="Junio";break;
      case 7:  $mes="Julio";break;
      case 8:  $mes="Agosto";break;
      case 9:  $mes="Septiembre";break;
      case 10: $mes="Octubre";break;
      case 11: $mes="Noviembre";break;
      case 12: $mes="Diciembre";break;
   }
   $numero = date("j");
   $anio = date("Y");
   $hora = date("g:i  a");
   printf("<header>\n");
   printf("\t<div class=\"box box3 shadow3\">\n");
   printf("\t\t<p>Hoy es %s, %d de %s del %d \n</p>\n", $dia, $numero, $mes, $anio);
   printf("\t</div>\n");
   printf("</header>\n");
   printf("<section>\n");
   printf("<article>\n");
   printf("\t<div class=\"box box4 shadow4\">\n");
   printf("\t\t\t<p>Son las: %s \n\t\t</p>\n", $hora);
   printf("\t</div>\n");
   printf("</article>\n");
   printf("</section>\n");
?>
</body>
<script src="js/modernizr.custom.lis.js"></script>
<script src="js/prefixfree.min.js"></script>
</html>
```

**Resultado en el navegador usando el servidor web:**


**Ejemplo 3: El siguiente ejemplo permite realizar una encuesta a los visitantes del sitio averiguando de qué forma se han enterado del sitio. El sitio únicamente reporta el medio que el usuario ha seleccionado, no realiza conteos. Esto será posible hasta que veamos archivos o bases de datos.**

**Archivo1: encuesta.php**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Encuesta</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <link rel="stylesheet" href="css/encuesta.css" />
    <link rel="stylesheet" href="css/cd-dropdown-menu.css" />
    <link rel="stylesheet" href="css/common.css" />
    <link rel="stylesheet" href="css/demo.css" />
    <link rel="stylesheet" href="css/icons.css" />
    <link rel="stylesheet" href="css/links.css" />
    <script src="js/modernizr.custom.lis.js"></script>
    <script src="js/prefixfree.min.js"></script>
</head>
<body>
<header>
   <h1 class="extruded">
      Favor responder la pregunta
   </h1>
</header>
<section>
<article>
<?php
   if(isset($_POST['enviar'])):
      $contact = $_POST['contacto'];
         switch($contact):
            case 'client':
                 echo "<p>Cliente que visita constantemente el sitio.</p>";
                 break;
            case 'television':
                 echo "<p>Cliente referido por anuncio televisivo.</p>";
                 break;
            case 'phonebook':
                 echo "<p>Cliente referido mediante guía telefónica.</p>";
                 break;
        case 'social':
             echo "<p>Cliente referido a través de redes sociales.</p>";
             break;
            case 'friend':
```

```
                echo "<p>Cliente referido por amigos.</p>";
                break;
            //No parece necesario porque el ingreso es por medio de un campo select
         default:
                echo "<p>No se puede especificar cómo contactó el cliente con
nuestro sitio web.</p>";
          endswitch;
          $link  = "<a href=\"". $_SERVER['PHP_SELF'] ."\" class=\"a-btn\">";
      $link .= "\t<span class=\"a-btn-symbol\">i</span>";
      $link .= "\t<span class=\"a-btn-text\">Volver</span>";
      $link .= "\t<span class=\"a-btn-slide-text\">al formulario</span>";
      $link .= "\t<span class=\"a-btn-slide-icon\"></span>";
      $link .= "</a>";
      echo $link;
   else:
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
   <select name="contacto" id="cd-dropdown" class="cd-select">
      <option value="-1" selected>(Selecciones una opción)</option>
      <option value="client" class="icon-chrome">Soy cliente frecuente</option>
      <option      value="television"      class="icon-safari">Publicidad      por
televisión</option>
      <option value="phonebook" class="icon-IE">Directorio telefónico</option>
      <option    value="social"    class="icon-firefox">A    través    de    redes
sociales</option>
      <option value="friend" class="icon-opera">Por sugerencia de amigo(s)</option>
   </select>
   <input    type="submit"    name="enviar"    id="enviar"    class="styled-button"
value="Enviar" />
</form>
<?php
   endif;
?>
</article>
</section>
</body>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="js/jquery.dropdown.js"></script>
<script>
   $(function() {
      $('#cd-dropdown').dropdown({
         gutter : 5,
         stack : false,
         slidingIn : 100
      });

   });
</script>
</html>
```

Resultado en el navegador haciendo uso del servidor web:



**Ejemplo 4: El siguiente ejemplo ilustra cómo utilizar el operador condicional para encontrar el mayor de tres números ingresados a través de un formulario. Los campos de texto donde se ingresan los números son validados con JavaScript para que sólo acepten valores enteros (de 0 a 9), mostrando un mensaje en caso de querer introducir cualquier otro carácter que no sea numérico.**

**Archivo 1: compararnumeros.html**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Comparar números</title>
    <meta charset="utf-8" />
    <meta  name="viewport"  content="width=device-width,user-scalable=no,initial-
scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <link href="http://fonts.googleapis.com/css?family=Muli" rel="stylesheet">
    <link rel="stylesheet" href="css/font.css" />
    <link rel="stylesheet" href="css/formstyle.css" />
    <script src="js/validar.js"></script>
</head>
<body>
<header>
    <h1>Comparación de números</h1>
</header>
<section>
<article id="formarea">
<form name="frmpalabras" method="POST" action="compararnum.php">
    <fieldset>
        <label for="number1">Primer número:</label>
        <input type="text" name="numero1" id="number1" placeholder="Primer número"
/>
        <br />
        <span id="numbersOnly1">Ingrese números enteros</span>
        <label for="number2">Segundo número:</label>
```

```
        <input type="text" name="numero2" id="number2" placeholder="Segundo número"
/>
        <br />
        <span id="numbersOnly2">Ingrese números enteros</span>
        <label for="number3">Tercer número:</label>
        <input type="text" name="numero3" id="number3" placeholder="Tercer número"
/>
        <br />
        <span id="numbersOnly3">Ingrese números enteros</span>
        <input type="submit" name="conteo" id="count" value="Enviar" />
    </fieldset>
</form>
</article>
</section>
</body>
</html>
```

**Archivo 2: compararnum.php**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Contador de palabras</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Muli" />
    <link rel="stylesheet" href="css/font.css" />
    <link rel="stylesheet" href="css/formstyle.css" />
    <script src="js/prefixfree.min.js"></script>
</head>
<body>
<header>
    <h1>
        Resultados de la comparación de <?php echo $num1 = isset($_POST['numero1'])
?
$_POST['numero1'] : 0; ?>,
        <?php echo $num2 = isset($_POST['numero3']) ? $_POST['numero2'] : 0; ?> y
        <?php echo $num3 = isset($_POST['numero3']) ? $_POST['numero3'] : 0; ?>
    </h1>
</header>
<section>
<article>
    <p class="msg">
        <?php
            $elmayor = ($num1 > $num2) ? $num1 : $num2;
            $elmayor = ($elmayor > $num3) ? $elmayor : $num3;
            printf("El número mayor es: %d", $elmayor);
        ?>
    </p>
<article>
</section>
</body>
</html>
```

Resultado en el navegador usando el servidor web:



## V. DISCUSION DE RESULTADOS

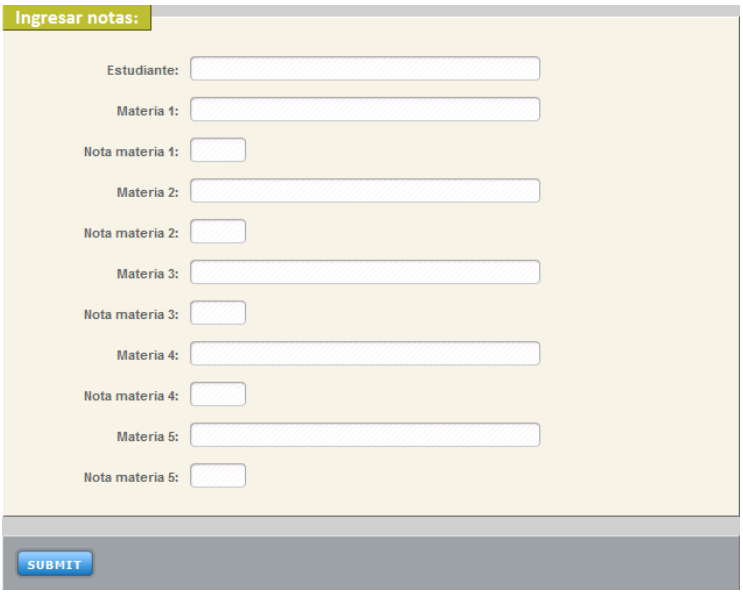
1. Cree un script que a partir de la fecha de nacimiento ingresada en un campo tipo number de formulario calcule el número de días que la persona que la ha ingresado ha vivido. Tome en cuenta los años bisiestos para obtener el cálculo exacto de días.
2. Realice un script que le permita determinar si un carácter ingresado por el usuario a través de un formulario, es una vocal (mayúscula, minúscula o acentuada), una consonante (mayúscula o minúscula), un número (del 0 al 9) o símbolos (.,;:()"""!¡¿?#$%&, etc). En caso de no ser ninguno de estos caracteres, debe mostrar un mensaje que indique que "**el carácter ingresado no se puede procesar**".

## VI. INVESTIGACION COMPLEMENTARIA

1. El número de unidades valorativas que puede cursar un estudiante en cada ciclo es determinado de acuerdo a su CUM:
   i. Si el CUM del estudiante es mayor o igual que 7.5, podrá cursar un máximo de 32 unidades valorativas (UV),
   ii. Si el CUM del estudiante es mayor o igual que 7.0, pero menor que 7.5, podrá cursar un máximo de 24 unidades valorativas (UV).
   iii. Si el CUM es mayor o igual que 6.0 y menor que 7.0, podrá cursar un máximo de 20 unidades valorativas.
   iv. Si el CUM acumulado es menor que 6.0, podrá cursar un máximo de 16 unidades valorativas.

Se pide que realice un formulario en donde se ingresen las notas de 5 materias que el estudiante ha cursado durante el primer ciclo de estudios y a partir de estas calcular el CUM y en base al valor obtenido indique el número de unidades valorativas que puede cursar el siguiente ciclo.

Puede crear un formulario como el siguiente:




## VII. BIBLIOGRAFIA


- Cabezas Granados, Luis Miguel. PHP 6 Manual Imprescindible. Editorial Anaya Multimedia. 1ª edición. Madrid, España. 2010.
- Doyle, Matt. Fundamentos de PHP Práctico. Editorial Anaya Multimedia. 1ª edición. Madrid, España. 2010.
- Gutiérrez, Abraham / Bravo, Ginés. PHP 5 a través de ejemplos. Editorial Alfaomega RAMA. 1ra edición. México. Junio 2005.
- Tim Converse / Steve Suehring. LA BIBLIA DE PHP 6 y MySQL. Editorial Anaya Multimedia. 1a. Edición. Madrid, España. 2009.
- John Coggeshall. La Biblia de PHP 5. 1ra Edición. Editorial Anaya Multimedia. Madrid España. http://www.php.net/manual/en