	<p style="text-align: center;"><b>UNIVERSIDAD DON BOSCO</b>  <b>FACULTAD DE INGENIERÍA</b>  <b>ESCUELA DE COMPUTACIÓN</b></p>
<p style="text-align: center;"><b>CICLO I-2021</b></p>	<p style="text-align: center;"><b>Lenguajes Interpretados en el Servidor</b>  <b>Guía de práctica No. 3</b>  <b>Estructuras de control - Sentencias condicionales</b></p>

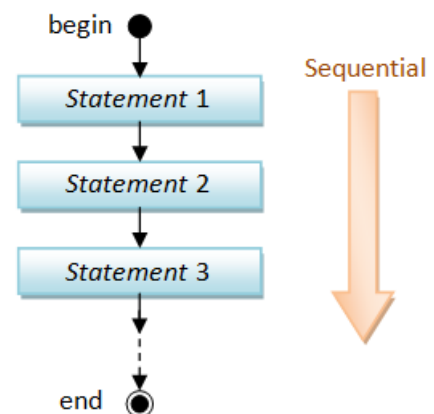
## I. OBJETIVOS

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

- Adquirir amplio dominio en el uso de las sentencias condicionales *if* y *switch*.
- Utilizar sentencias condicionales *if* y sentencias selectivas *switch* para resolver problemas prácticos.
- Lograr un uso adecuado de las sentencias condicionales anidadas.
- Dominar el uso de la sintaxis alternativa de las sentencias condicionales.
- Conseguir un uso correcto del operador condicional *?:*

## 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 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. En esta guía de práctica trabajaremos con las primeras dos, dejando las últimas para la siguiente guía de práctica.

Se podrían clasificar las sentencias condicionales en dos:

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

## 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.

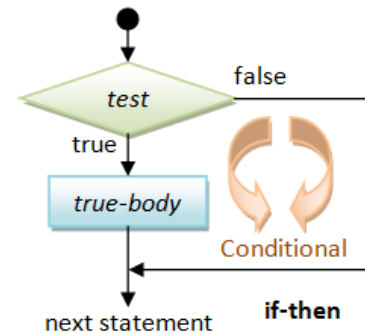
PHP proporciona una estructura **if** que es similar a la de C:

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, expresión se evalúa a su valor condicional (*boolean*). Si expresión 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;
    }
?>
```

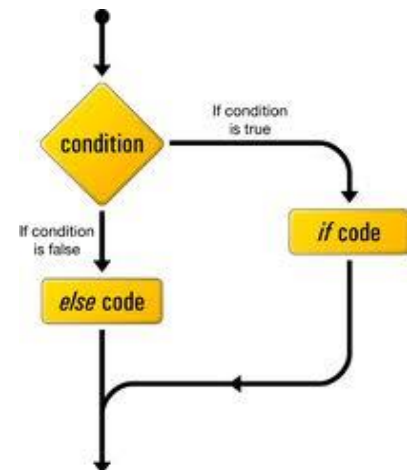
Las sentencias **if** se pueden anidar indefinidamente dentro de otras sentencias **if** o **else**, lo cual proporciona una flexibilidad completa para ejecuciones condicionales en las diferentes partes de tu programa.

## 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, y si hubiera alguna expresión **elseif** - sólo si se evaluara también 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 ;
```

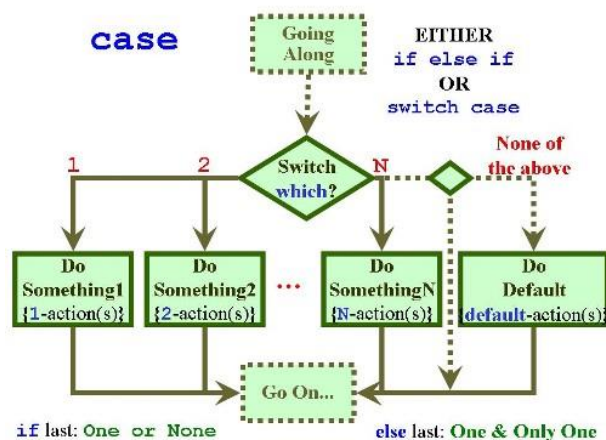
Debe tener en cuenta que si la expresión condicional se evalúa como **TRUE**, se ejecutará expresion1 y si se evalúa **FALSE**, se ejecutará expresion2. Como las expresiones devuelven un valor, usted puede perfectamente asignar la instrucción completa en una variable, como se muestra en el siguiente 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 sentencias 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 #3	1
2	Computadora con WampServer instalado y funcionando correctamente	1
3	Editor PHP Designer 2007 personal o DevPHP, Sublime, Notepad++ , etc.	1
4	Memoria USB	1

### IV. PROCEDIMIENTO

#### Indicaciones:

1. Asegúrese de digitar el código de los siguientes ejemplos que se presentan a continuación. Tenga en cuenta que los editores de código PHP no son compiladores solamente editores por lo tanto los errores de sintaxis lo podrá observar únicamente hasta que se ejecute el script al cargar la página en el navegador de su preferencia.
2. Descargar de la pagina de la universidad los recursos que utilizara en la guía
3. Recuerde crear una carpeta en la carpeta del servidor que está utilizando (www, httdocs), en donde almacenara los recursos de la guía y sus paginas PHP

**Ejemplo 1:** En este ejemplo se evaluará si un año ingresado por un usuario a través de un formulario resulta ser bisiesto o no. El script PHP procesará el año ingresado para determinar si resulta ser bisiesto o no.

#### Archivo 1: bisiesto.php

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <title>Año bisiesto</title>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width,user-scalable=no,initial-
7 scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
8     <meta http-equiv="Content-Script-Type" content="text/javascript" />
```





El año 2019 no es bisiesto  
[Probar otro año](#)

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).

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Calcular Salario</title>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width,user-scalable=no,initial-scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
7   <link rel="stylesheet" href="css/salario.css" />
8   <link rel="stylesheet" href="css/links.css" />
9   <script src="js/modernizr.custom.lis.js"></script>
10 </head>
11 <body>
12 <header id="inset">
13   <h1>Detalle del salario</h1>
14 </header>
15 <section>
16 <article>
17 <?php
18     if(isset($_POST['enviar'])){
19         $empleado = isset($_POST['empleado']) ? $_POST['empleado'] : "";
20         $sueldobase = isset($_POST['sueldobase']) ? $_POST['sueldobase'] : "";
21         if(isset($_POST['hextra'])){
22             $horasextras = isset($_POST['numhorasex']) ? $_POST['numhorasex'] :
23 "0";
24             $pagohextra = isset($_POST['pagohextra']) ? $_POST['pagohextra'] : "0";
25             $sueldoextras = $horasextras * $pagohextra;
26         }
27         else{
28             $horasextras = 0;
29             $sueldoextras = 0;
30         }
31         $sueldoneto = $sueldobase + $sueldoextras;
32         echo "<div>\n<h3>Boleta de pago</h3>\n";
33         echo "<table>\n";
34         echo "\t<tr>\n\t\t<th colspan='2'>\n\t\t\tDetalle del
35 pago\n\t\t</th>\n\t\t<tr>\n\t\t\t
36         echo "\t<tr>\n\t\t\t<td>\n\t\t\tEl empleado es: \n\t\t\t</td>\n\t\t\t<td class='detail'>\n\t\t\t\t,$empleado,\n\t\t\t\t</td>\n\t\t\t<tr>\n\t\t\t\t

```

```

37     echo "<tr>\n\t\t<td>\n\t\t\tEl sueldo base es: \n\t\t\t</td>\n\t\t\t<td class=\"detail\">\n\t\t\t\t", $sueldobase, "\n\t\t\t</td>\n\t\t\t</tr>\n";
38     echo "<tr>\n\t\t\t<td>\n\t\t\t\tLas horas extras son: \n\t\t\t\t</td>\n\t\t\t\t<td class=\"detail\">\n\t\t\t\t\t", $horasextras, "\n\t\t\t\t</td>\n\t\t\t\t</tr>\n";
39     echo "<tr>\n\t\t\t\t<td>\n\t\t\t\t\tEl pago por hora extra es:
40 \n\t\t\t\t\t</td>\n\t\t\t\t\t<td class=\"detail\">\n\t\t\t\t\t\t", $sueldohextras,
41 "\n\t\t\t\t\t</td>\n\t\t\t\t\t</tr>\n";
42     echo "<tr>\n\t\t\t\t\t<th>\n\t\t\t\t\t\tEl sueldo neto devengado es:
43 \n\t\t\t\t\t\t</th>\n\t\t\t\t\t\t<th class=\"number\">\n\t\t\t\t\t\t\t", $sueldoneto,
44 "\n\t\t\t\t\t\t</th>\n\t\t\t\t\t\t</tr>\n";
45     echo "</table>\n</div>\n";
46 }
47 ?>
48 <a href="salario.html" class="a-btn">
49 <span class="a-btn-symbol">i</span>
50 <span class="a-btn-text">Ingresar</span>
51 <span class="a-btn-slide-text">otro empleado</span>
52 <span class="a-btn-slide-icon"></span>
53 </a>
54 </article>
55 </section>
56 </body>
57 </html>

```

1- Resultado de la **página salario.html** en el navegador haciendo uso del servidor web:

2- Resultado del **script salario.php** al hacer clic en el **botón Enviar** del formulario del paso anterior:

Detalle del pago	
El empleado es:	Carlos Perez
El sueldo base es:	\$ 800
Las horas extras son:	5
El pago por hora extra es:	\$ 67.5
El sueldo neto devengado es:	\$ 867.5

**Ejemplo 3:** En el siguiente ejemplo se ilustra cómo detectar el idioma en el cliente (navegador del usuario). Para probar el ejemplo, será necesario que cambie el idioma predeterminado del navegador, agregando un nuevo idioma si es necesario y subiéndolo hasta el primer nivel para que este nuevo idioma sea el predefinido. Pregunte al docente cómo hacer esto si acaso no lo sabe Ud.

#### Archivo 1: idioma.php

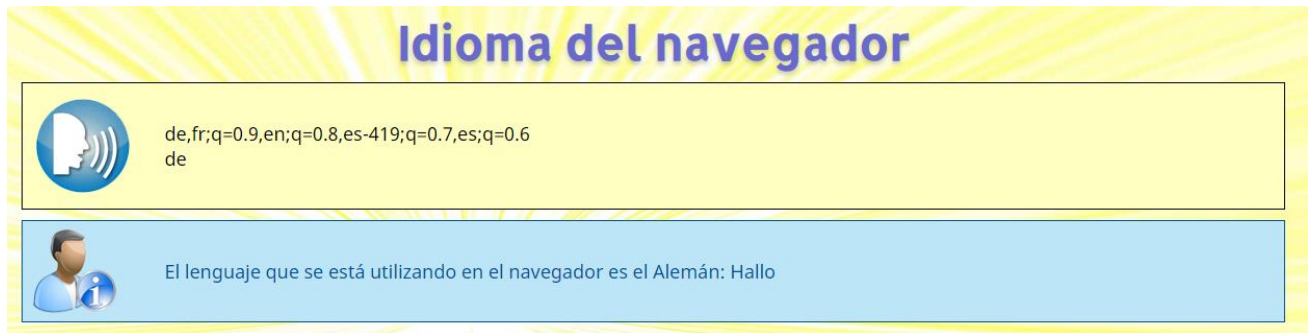
```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <title>Detector de Lenguaje del Navegador</title>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width,user-scalable=no,initial-
7 scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
8     <meta http-equiv="Content-Script-Type" content="text/javascript" />
9     <link rel="stylesheet" href="css/idioma.css" />
10 </head>
11 <body>
12 <div id="wrapper">
13 <header>
14     <h1 class="emboss">Idioma del navegador</h1>
15 </header>
16 <section>
17 <?php
18     //Antes de nada introducimos mensajes en forma de variables
19     $espanol = "Hola";
20     $ingles = "Hello";
21     $aleman = "Hallo";
22     $frances = "Bonjour";
23     $italiano = "Ciao";
24     $portugues = "Olá";
25     //Ahora leemos del navegador cuál es su lengua oficial
26     $completo = $_SERVER["HTTP_ACCEPT_LANGUAGE"];
27     $idioma = substr($ _SERVER["HTTP_ACCEPT_LANGUAGE"],0,2);
28     // $idioma=substr($HTTP_ACCEPT_LANGUAGE,0,2);
29     //Formulamos las posibilidades que se pueden dar
30     echo "<p>" . $completo . "<br>";
31     echo $idioma . "</p>\n";
32     if($idioma == "es"){
33         echo '<p class="msgOff">';
34         printf ("El lenguaje que se está utilizando en el navegador es el Español: %s </p>\n", $espanol);
35     }
36     elseif($idioma == "fr"){
37         echo '<p class="msgOff">';
38         printf ("El lenguaje que se está utilizando en el navegador es el Francés: %s </p>\n", $frances);
39     }
40     elseif($idioma == "en"){
41         echo '<p class="msgOff">';
42         printf ("El lenguaje que se está utilizando en el navegador es el Inglés: %s </p>\n", $ingles);
43     }
44     elseif($idioma == "de"){
45         echo '<p class="msgOff">';
46         printf ("El lenguaje que se está utilizando en el navegador es el Alemán: %s </p>", $aleman);
47     }
48     elseif($idioma == "it"){
49         echo '<p class="msgOff">';
50         printf ("El lenguaje que se está utilizando en el navegador es el Inglés: %s </p>\n", $italiano);
51     }
52     elseif($idioma == "pt"){
53         echo '<p class="msgOff">';
54         printf ("El lenguaje que se está utilizando en el navegador es el Alemán: %s </p>", $portugues);
55     }
56     else{
57         echo '<p class="msgOff">';
58         echo "El idioma del navegador es desconocido.</p>\n";
59     }
60 ?>
```

```

61 </section>
62 </div>
63 </body>
64 <script src="js/modernizr.custom.lis.js"></script>
65 <script src="js/switchclass.js"></script>
66 </html>

```

Resultado del **script idioma.php** en el navegador haciendo uso del servidor web:



**Ejemplo 4:** 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

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <title>Fecha y hora del Sistema</title>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width,user-scalable=no,initial-
7 scale=1.0,maximum-scale=1.0,minimum-scale=1.0" />
8     <meta http-equiv="Content-Script-Type" content="text/javascript" />
9     <link rel="stylesheet" href="css/fechahora.css" />
10 </head>
11 <body>
12 <?php
13     //Establecer la zona horaria para el servidor web,
14     //de modo que concuerde con la hora del país donde
15     //se está ejecutando el script.
16     date_default_timezone_set("America/El_Salvador");
17     $dia=date("w");
18     switch($dia){
19         case 0: $dia="Domingo";break;
20         case 1: $dia="Lunes";break;
21         case 2: $dia="Martes";break;
22         case 3: $dia="Miércoles";break;
23         case 4: $dia="Jueves";break;
24         case 5: $dia="Viernes";break;
25         case 6: $dia="Sábado";break;
26     }
27     $mes=date("n");

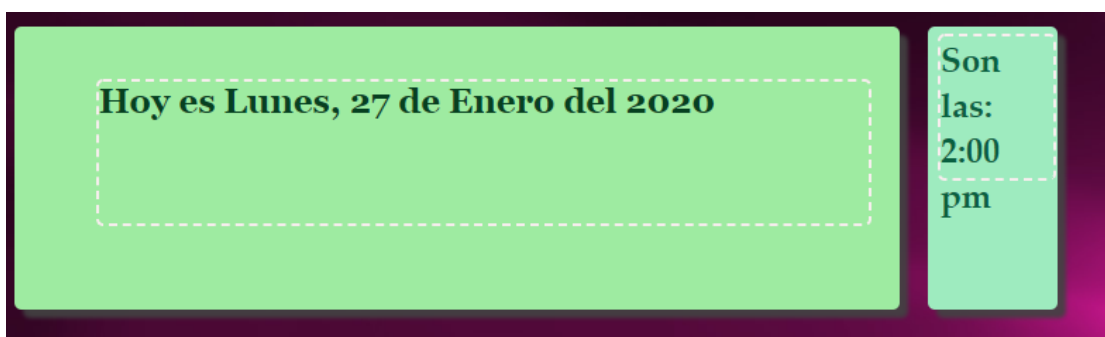
```

```

28     switch($mes){
29         case 1: $mes="Enero";break;
30         case 2: $mes="Febrero";break;
31         case 3: $mes="Marzo";break;
32         case 4: $mes="Abril";break;
33         case 5: $mes="Mayo";break;
34         case 6: $mes="Junio";break;
35         case 7: $mes="Julio";break;
36         case 8: $mes="Agosto";break;
37         case 9: $mes="Septiembre";break;
38         case 10: $mes="Octubre";break;
39         case 11: $mes="Noviembre";break;
40         case 12: $mes="Diciembre";break;
41     }
42     $numero = date("j");
43     $anio = date("Y");
44     $hora = date("g:i a");
45     printf("<header>\n");
46     printf("\t<div class=\"box box3 shadow3\">\n");
47     printf("\t\t<p>Hoy es %s, %d de %s del %d \n</p>\n", $dia, $numero, $mes, $anio);
48     printf("\t</div>\n");
49     printf("</header>\n");
50     printf("<section>\n");
51     printf("<article>\n");
52     printf("\t<div class=\"box box4 shadow4\">\n");
53     printf("\t\t\t<p>Son las: %s \n\t\t</p>\n", $hora);
54     printf("\t</div>\n");
55     printf("</article>\n");
56     printf("</section>\n");
57 ?>
58 </body>
59 <script src="js/modernizr.custom.lis.js"></script>
60 <script src="js/prefixfree.min.js"></script>
61 </html>

```

Resultado del **script fechahora.php** en el navegador usando el servidor web:



**Ejemplo 5:** 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

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <title>Comparar números</title>
5      <meta charset="utf-8" />
6      <meta name="viewport" content="width=device-width,user-scalable=no,initial-scale=1.0,
7      maximum-scale=1.0,minimum-scale=1.0" />
8      <meta http-equiv="Content-Script-Type" content="text/javascript" />
9      <link href="http://fonts.googleapis.com/css?family=Muli" rel="stylesheet">
10     <link rel="stylesheet" href="css/font.css" />
11     <link rel="stylesheet" href="css/formstyle.css" />
12     <script src="js/validar.js"></script>
13     <!-- Para las versiones anteriores del IE9 -->
14     <!--[if lt IE 9]>
15         <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
16     <![endif]-->
17 </head>
18 <body>
19 <header>
20     <h1>Comparación de números</h1>
21 </header>
22 <section>
23 <article id="formarea">
24 <form name="frmpalabras" method="POST" action="compararnum.php">
25     <fieldset>
26         <label for="number1">Primer número:</label>
27         <input type="text" name="numero1" id="number1" placeholder="Primer número" />
28         <br />
29         <span id="numbersOnly1">Ingrese números enteros</span>
30         <label for="number2">Segundo número:</label>
31         <input type="text" name="numero2" id="number2" placeholder="Segundo número" />
32         <br />
33         <span id="numbersOnly2">Ingrese números enteros</span>
34         <label for="number3">Tercer número:</label>
35         <input type="text" name="numero3" id="number3" placeholder="Tercer número" />
36         <br />
37         <span id="numbersOnly3">Ingrese números enteros</span>
38         <input type="submit" name="conteo" id="count" value="Enviar" />
39     </fieldset>
40 </form>
41 </article>
42 </section>
43 </body>
44 </html>
```

## Archivo 2: compararum.php

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <title>Contador de palabras</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Muli" />
7     <link rel="stylesheet" href="css/font.css" />
8     <link rel="stylesheet" href="css/formstyle.css" />
9     <script src="js/prefixfree.min.js"></script>
10    <!-- Para las versiones anteriores del IE9 -->
11    <!--[if lt IE 9]>
12        <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
13    <![endif]-->
14 </head>
15 <body>
16 <header>
17     <h1>
18         Resultados de la comparación de <?php echo $num1 = isset($_POST['numero1']) ?
19 <?php echo $num2 = isset($_POST['numero3']) ? $_POST['numero2'] : 0; ?>,
20     <?php echo $num2 = isset($_POST['numero3']) ? $_POST['numero2'] : 0; ?> y
21     <?php echo $num3 = isset($_POST['numero3']) ? $_POST['numero3'] : 0; ?>
22     </h1>
23 </header>
24 <section>
25 <article>
26     <p class="msg">
27         <?php
28             $elmayor = ($num1 > $num2) ? $num1 : $num2;
29             $elmayor = ($elmayor > $num3) ? $elmayor : $num3;
30             printf("El número mayor es: %d", $elmayor);
31         ?>
32     </p>
33 </article>
34 </section>
35 </body>
36 </html>
```

1. Resultado del **archivo compararum.html** en el navegador usando el servidor web:



**COMPARACIÓN DE NÚMEROS**

PRIMER NÚMERO:	25
SEGUNDO NÚMERO:	32
TERCER NÚMERO:	17
<input type="button" value="ENVIAR"/>	

2. Resultado del **script compararnum.php**, después de haber hecho clic en el botón Enviar, del punto anterior



## 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:



**Ingresar notas:**

Estudiante:

Materia 1:

Nota materia 1:

Materia 2:

Nota materia 2:

Materia 3:

Nota materia 3:

Materia 4:

Nota materia 4:

Materia 5:

Nota materia 5:

**SUBMIT**

## 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>