



Tipos de variables

Los nombres de variables comienzan con el carácter \$ y son sensibles a mayúsculas y minúsculas (no así las palabras claves del lenguaje)

En PHP no es necesario definir el tipo de dato que almacena antes de utilizarla, las mismas se crean en el momento de emplearlas. Las variables se declaran cuando se le asigna un valor, por ejemplo:

```
$dia = 24; //Se declara una variable de tipo integer.  
$sueldo = 758.43; //Se declara una variable de tipo double.  
$nombre = "juan"; //Se declara una variable de tipo string.  
$existe = true; //Se declara una variable boolean.
```

También podemos hacer notar que para disponer comentarios de línea debemos utilizar dos caracteres //

Para la impresión de variables utilizaremos inicialmente el comando echo. Un programa completo que inicializa y muestra el contenido de cuatro variables de distinto tipo es:

pagina1.php

Hemos utilizado un comando echo para mostrar los mensajes, otro el contenido de variables y finalmente otro para imprimir un elemento HTML. Este proceso puede acortarse un poco pero para que sea más claro inicialmente tomaremos el camino largo de hacer la impresión de un dato con cada comando echo.

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $dia = 24; //Se declara una variable de tipo integer.
        $sueldo = 758.43; //Se declara una variable de tipo double.
        $nombre = "juan"; //Se declara una variable de tipo string.
        $existe = true; //Se declara una variable boolean.
        echo "Variable entera:";  

        echo $dia;
        echo "<br>";
        echo "Variable double:";  

        echo $sueldo;
        echo "<br>";
        echo "Variable string:";  

        echo $nombre;
        echo "<br>";
        echo "Variable boolean:";  

        echo $existe;
    ?>
</body>

</html>
```

Problema propuesto.

Definir una variable de cada tipo:
integer, double, string y boolean.
Luego imprimirlas en la página, una
por línea.

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php

$edad = 20;
$peso = 90.25;
$nombre = "juan";
$existe = true;
echo "Variable integer:";
echo $edad;
echo "<br>";
echo "Variable double:";
echo $peso;
echo "<br>";
echo "Variable string:";
echo $nombre;
echo "<br>";
echo "Variable boolean:";
echo $existe;

?>

</body>

</html>
```

Variables de tipo string

Una variable de este tipo puede almacenar una serie de caracteres.

```
● ● ●  
$cadena1="Hola";  
$cadena2="Mundo";  
echo $cadena1." ".$cadena2;
```

Para concatenar string empleamos el operador .
(punto)

Tengamos en cuenta que el comando echo de más arriba lo podemos hacer más largo de la siguiente forma:

```
● ● ●  
echo $cadena1;  
echo " ";  
echo $cadena2;
```

A medida que uno haga ejercicios podremos resumir en un solo comando echo la salida de múltiples variables.

Cuando una cadena encerrada entre comillas dobles contiene una variable en su interior, ésta se trata como tal, por lo tanto se utilizará su contenido para el almacenamiento:

A screenshot of a terminal window with a dark background and light text. At the top, there are three colored window control buttons (red, yellow, green). Below them, the terminal prompt is visible. The main area contains the following PHP code:

```
$dia=10;  
$fecha="Hoy es $dia";  
echo $fecha;
```

The code is written in a monospaced font, with variable names in blue and strings in cyan.

En pantalla se muestra: Hoy es 10
Es decir, en la cadena, se sustituye el nombre de
la variable \$dia, con el contenido de la misma.
Una cadena se puede definir con las comillas
simples (pero es importante tener en cuenta que
no se sustituyen las variables si empleamos
comillas simples):



```
$nombre='juan carlos';
```

Estructura condicional (if)

Cuando se pretende que el programa, una vez llegado a un cierto punto, tome un camino concreto en determinados casos y otro diferente si las condiciones de ejecución difieren, se utiliza el conjunto de instrucciones: if, else y elseif. La estructura base de este tipo de instrucciones es la siguiente:

```
● ● ●  
if (Condición)  
{  
    Instrucción a1;  
    Instrucción a2;  
}  
else  
{  
    Instrucción b1;  
    Instrucción b2;  
}
```

Cuando la ejecución llega al punto donde se encuentra la instrucción if, el programa verificará el cumplimiento o no de la condición. Si la condición es verdadera se ejecutarán las instrucciones a1 y a2, de lo contrario, se ejecutarán las instrucciones b1 y b2.

En los casos en que las condiciones sean varias, se pueden utilizar los if de un modo denominado anidado o anillado, como se indica de la manera siguiente:



```
if (Condicion 1)
{
    Instrucción a1;
    Instrucción a2;
}
else
{
    if (Condicion 2)
    {
        Instrucción b1;
        Instrucción b2;
    }
    else
    {
        Instrucción c1;
        Instrucción c2;
    }
}
```

De este modo se pueden introducir tantas condiciones como se quiera dentro de la condición principal. Una variante de este sistema es utilizando la sentencia elseif, que permite en una sola línea introducir una condición adicional:



```
if (Condicion 1)
{
    Instrucción a1;
    Instrucción a2;
}
elseif (Condicion 2)
{
    Instrucción b1;
    Instrucción b2;
}
else
{
    Instrucción c1;
    Instrucción c2;
}
```

Para las condiciones tener en cuenta que disponemos de los siguientes operadores:



```
== para ver si una variable es igual a otra.  
!= distinto.  
>= mayor o igual.  
> mayor.  
<= menor o igual  
< menor
```

La mejor forma de entender esta estructura condicional es por medio de ejemplos. El primero que nos plantearemos es generar un valor aleatorio (es decir lo elige la máquina al azar, como extraer una bolilla de un bolillero de lotería) comprendido entre 1 y 10. Luego mostraremos un mensaje si es menor o igual a 5 o si es mayor a 5. El programa completo es:
[pagina1.php](#)

```
● ● ●

<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <?php
      $valor = rand(1, 10);
      echo "El valor sorteado es $valor<br>";
      if ($valor <= 5) {
        echo "Es menor o igual a 5";
      } else {
        echo "Es mayor a 5";
      }
    ?>
  </body>

</html>
```

Es importante recordar que siempre la condición del if debe ir entre paréntesis. Si la condición del if se verifica verdadera (es decir el número sorteado es menor o igual a 5) ejecuta el primer bloque que se encuentra entre llaves. En caso de verificarse falsa la condición del if se ejecuta el bloque entre llaves que se encuentra después del else.

El valor aleatorio lo generamos llamando a la función rand pasándole el valor mínimo y máximo:



Imprimimos el valor generado a los efectos de controlar el resultado:



```
echo "El valor sorteado es $valor<br>";
```

En el primer problema tenemos solo dos caminos posibles, el valor es menor o igual a cinco o es mayor. En un segundo ejemplo mostraremos como disponer una estructura condicional if anidada.

El problema es el siguiente: Generar un valor aleatorio entre 1 y 100. Luego mostrar si tiene 1,2 o 3 dígitos.

Como podemos observar estamos en presencia de un problema que tiene tres caminos posibles. El valor puede tener 1 dígito, 2 dígitos o 3 dígitos. Si pensamos un poco podremos identificar que para que tenga un dígito debe generarse un valor entre 1 y 9, para que tenga dos dígitos deberá estar comprendido entre 10 y 99, y finalmente para tener 3 dígitos deberá ser el valor 100.

La página con el programa respectivo es:



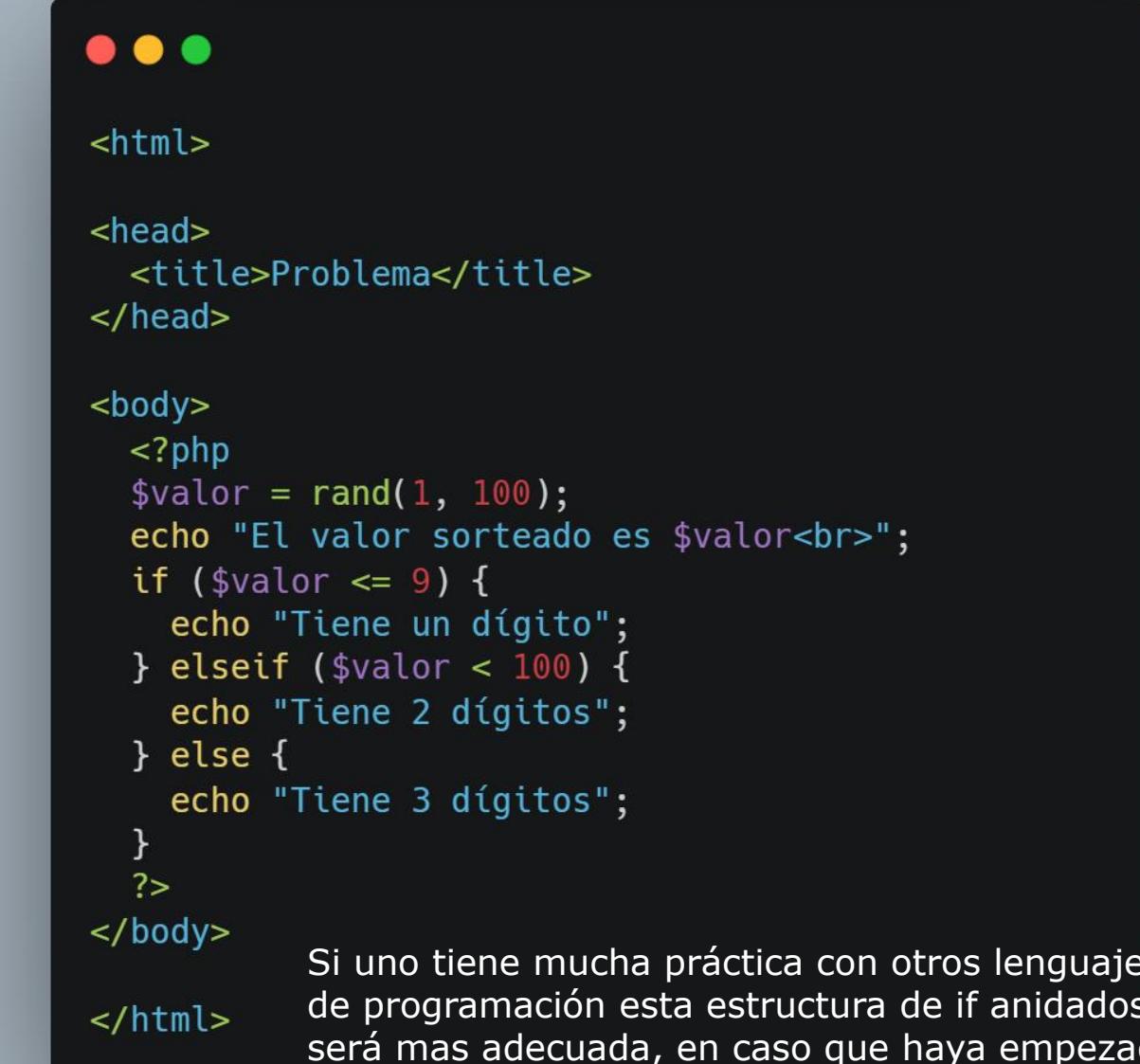
```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <?php
      $valor = rand(1, 100);
      echo "El valor sorteado es $valor<br>";
      if ($valor <= 9) {
        echo "Tiene un dígito";
      } else {
        if ($valor < 100) {
          echo "Tiene 2 dígitos";
        } else {
          echo "Tiene 3 dígitos";
        }
      }
    ?>
  </body>

</html>
```

Es importante notar como la segunda estructura if se encuentra contenida entre las llaves del else del primer if. Es decir que si el valor aleatorio es menor o igual a 9 se ejecuta el bloque del verdadero del primer if y no se ejecuta por lo tanto el if anidado en el else. Por el contrario si la condición del primer if se verifica falsa se ejecuta el bloque del else del primer if, la misma contiene una estructura if con sus bloques del verdadero y falso.

A modo ilustrativo el mismo ejemplo resuelto con la estructura if/elseif será:



```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <?php
      $valor = rand(1, 100);
      echo "El valor sorteado es $valor<br>";
      if ($valor <= 9) {
        echo "Tiene un dígito";
      } elseif ($valor < 100) {
        echo "Tiene 2 dígitos";
      } else {
        echo "Tiene 3 dígitos";
      }
    ?>
  </body>
</html>
```

Si uno tiene mucha práctica con otros lenguajes de programación esta estructura de if anidados le será mas adecuada, en caso que haya empezado a programar hace poco será conveniente que elija y practique la primera forma.

Problema propuesto.

Generar un valor aleatorio entre 1 y 3. Luego imprimir en castellano el número (Ej. si se genera el 3 luego mostrar en la página el string "tres").

Para ver si una variable es igual a cierto valor debemos plantear una condición similar a:

```
if ($valor==3)
{
    //algoritmo
}
```

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php

$num = rand(1, 3);
if ($num == 1) {
    echo "uno";
} else {
    if ($num == 2) {
        echo "dos";
    } else {
        echo "tres";
    }
}

?>

</body>

</html>
```

Estructuras repetitivas (for - while - do/while)

Las estructuras repetitivas son similares al lenguaje C, JavaScript, Java etc.

Estructura for:



```
for([Inicialización de la variable];[Condición];[Incremento o decremento de la variable])
{
    [Instrucciones];
}
```

El primer ejemplo que haremos es mostrar en la página los números del 1 al 100:

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        for ($f = 1; $f <= 100; $f++) {
            echo $f;
            echo "<br>";
        }
    ?>
</body>

</html>
```

Quien no ha visto la estructura for en otro lenguaje pasamos a explicarla:

```
● ● ●  
  
for ($f = 1; $f <= 100; $f++) {  
    echo $f;  
    echo "<br>";  
}
```

El primer argumento del for es la inicialización de una variable, en este caso se inicializa la variable \$f con el valor 1. Este primer argumento del for se ejecuta solo una vez. Luego se ejecuta el segundo argumento que es la condición. Si la misma se verifica como verdadera se ejecuta todo el bloque comprendido entre las llaves de apertura y cerrado. Luego de haberse ejecutado el bloque repetitivo se ejecuta el tercer argumento del for que es el incremento de la variable, en este caso \$f++ incrementa el contenido de la variable \$f en 1 (también podemos poner en lugar de \$f++ la asignación \$f=\$f+1)

Luego del incremento de la variable se ejecuta nuevamente la condición del for (segundo argumento), de validarse nuevamente verdadero pasa a ejecutar el bloque repetitivo. Este ciclo se repite hasta que la condición del for se verifica falsa.

La segunda estructura repetitiva es:

```
● ● ●  
while (condición)  
{  
    [Instrucciones];  
}
```

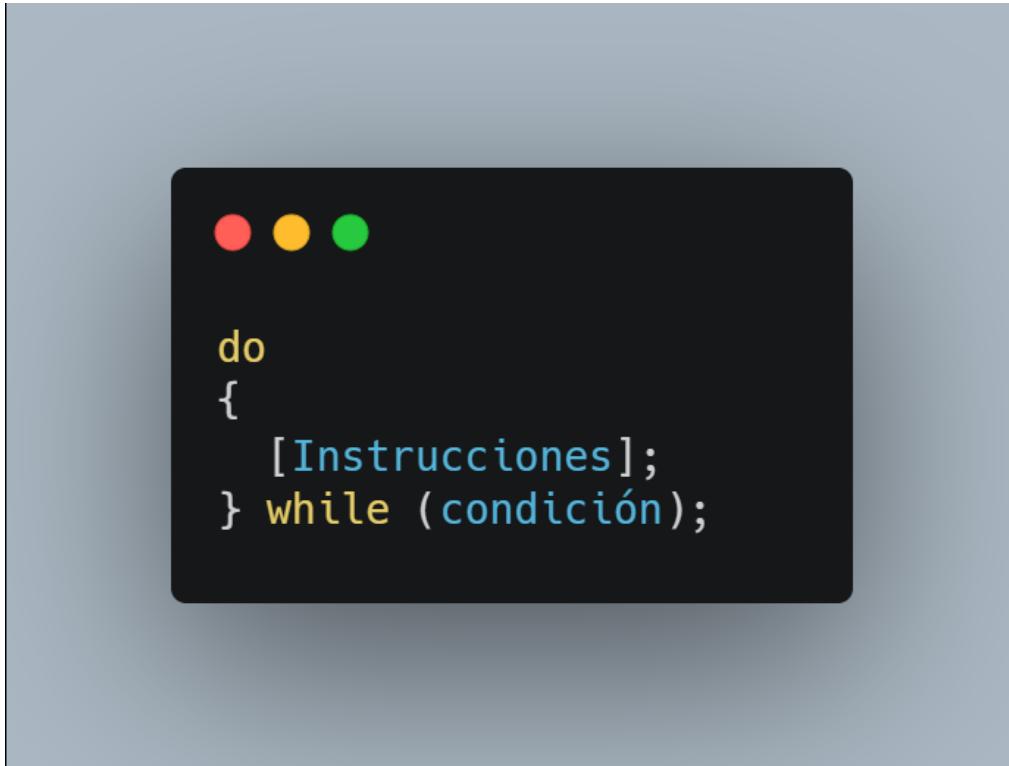
Esta estructura está en casi todos los lenguajes. El bloque se repite mientras la condición del while sea verdadera. La condición del while se verifica antes de ingresar al bloque a repetir. Si la misma se verifica falsa la primera vez no se ejecutará el bloque.

Veamos un ejemplo: Generar un valor aleatorio entre 1 y 100, luego imprimir en la página desde 1 hasta el valor generado (de uno en uno):

```
● ● ●  
<html>  
  <head>  
    <title>Problema</title>  
  </head>  
  
  <body>  
    <?php  
      $valor = rand(1, 100);  
      $inicio = 1;  
      while ($inicio <= $valor) {  
        echo $inicio;  
        echo "<br>";  
        $inicio++;  
      }  
    ?>  
  </body>  
</html>
```

La variable \$inicio tiene el valor 1 antes de ingresar al while. Cada vez que se ejecuta una vez el bloque del while se incrementa \$inicio en uno. Cuando \$inicio supere la variable \$valor finalizará la estructura repetitiva y pasará a ejecutarse la instrucción inmediatamente siguiente a la llave de cerrado del while.
Es importante notar que luego de la condición del while NO disponemos PUNTO y COMA.

Por último tenemos en el lenguaje una estructura repetitiva similar al while llamada do/while, donde la condición se verifica luego de ejecutarse el bloque repetitivo.



Queda como tarea intentar hacer una página empleando esta estructura. Tener en cuenta que al final de la línea del do/while SI LLEVA punto y coma.

Problema propuesto.

Mostrar la tabla de multiplicar del 2. Emplear el for, luego el while y por último el do/while.

La estructura for permite incrementar una variable de 2 en 2:

```
for ($f = 2; $f <= 20; $f = $f + 2)
```

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php

echo "Tabla del 2 con el for";
echo "<br>";
for ($f = 2; $f <= 20; $f = $f + 2) {
    echo $f;
    echo "-";
}
echo "<br>";
echo "Tabla del 2 con el while";
echo "<br>";
$f = 2;
while ($f <= 20) {
    echo $f;
    echo "-";
    $f = $f + 2;
}
echo "<br>";
echo "Tabla del 2 con el do/while";
echo "<br>";
$f = 2;
do {
    echo $f;
    echo "-";
    $f = $f + 2;
} while ($f <= 20);

?>

</body>

</html>
```

Envío de datos de un FORMULARIO (controles text y submit)

Una actividad fundamental en PHP es la recolección de datos de un formulario HTML. El proceso para el manejo de FORMULARIOS requiere generalmente dos páginas, una que implementa el formulario y otra que procesa los datos cargados en el formulario.

La estructura mínima de un formulario es la siguiente: para la entrada de un nombre de persona, un objeto text y un botón para el envío del dato al servidor:

`pagina1.html`

```
<html>
  <head>
    <title>Formulario de entrada del dato</title>
  </head>

  <body>
    <form method="post" action="pagina2.php">
      Ingrese su nombre:
      <input type="text" name="nombre">
      <br>
      <input type="submit" value="confirmar">
    </form>
  </body>

</html>
```

Esta página está completamente codificada en HTML, es decir no hay un script en PHP, el formulario contiene elementos HTML puros.

El elemento <form> y </form> nos permite definir un formulario en la página.

Este elemento tiene dos propiedades que debemos inicializar obligatoriamente: action y method.

La propiedad action indica el nombre del archivo que recibirá los datos ingresados por el operador en el formulario y que serán enviados al servidor cuando se presione el botón (submit). La propiedad method indica como se organizan esos datos para enviarlos al servidor, pudiendo ser mediante los métodos post o get (normalmente los datos de un formulario se envían mediante el método post).

Para crear un cuadro de texto para el ingreso del nombre debemos definir un objeto de tipo "text" y darle un nombre:



```
<input type="text" name="nombre">
```

La propiedad type nos permite definir el tipo de control y con la propiedad name indicamos el nombre del control.

Por último todo formulario tiene un botón de tipo submit:



También utilizamos el elemento input pero en la propiedad type indicamos que se trata de un botón de envío de datos. En la propiedad value indicamos el texto que queremos que aparezca en el botón.

Ahora necesitamos una página con un pequeño programa en PHP que procese los datos ingresados en el formulario:

pagina2.php

```
● ● ●  
<html>  
  <head>  
    <title>Captura de datos del form</title>  
  </head>  
  
  <body>  
    <?php  
      echo "El nombre ingresado es:";  
      echo $_POST[ 'nombre' ];  
    ?>  
  </body>  
  
</html>
```

Para acceder al dato en PHP se cuenta con un vector llamado `$_POST` indicando como subíndice el nombre del cuadro de texto que definimos en el formulario (dicho nombre es sensible a mayúsculas y minúsculas)

Es común indicar entre comillas simples el subíndice en lugar de comillas dobles (con comillas dobles también funciona)

En nuestro problema sólo mostramos por pantalla el valor ingresado en el formulario HTML:



```
echo $_POST[ 'nombre' ];
```

Problema propuesto.

Confeccionar un formulario que solicite la carga de un nombre de persona y su edad, luego mostrar en otra página si es mayor de edad (si la edad es mayor o igual a 18)

pagina1.php

```
● ● ●

<html>

    <head>
        <title>Problema</title>
    </head>

    <body>

        <form method="post" action="pagina2.php">
            Ingrese el nombre:
            <input type="text" name="nombre">
            <br>
            Ingrese la edad:
            <input type="text" name="edad">
            <br>
            <input type="submit" value="confirmar">
        </form>

    </body>

</html>
```

pagina2.php



```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
echo $_POST[ 'nombre' ];
echo "<br>";
if ($_POST[ 'edad' ] >= 18) {
    echo "Es mayor de edad";
} else {
    echo "No es mayor de edad";
}
?>

</body>

</html>
```

FORMULARIO (control radio)

El control HTML radio generalmente se utiliza en conjunto con otros controles radio. Cuando agrupamos un conjunto de controles radio solo uno de ellos puede estar seleccionado y si seleccionamos otro se desmarca el actual.

Para analizar este control dispondremos un ejemplo:
Implementar un formulario que solicite la carga de dos enteros, uno en cada text.

Disponer dos controles de tipo radio que nos permitan seleccionar si queremos sumar o restar los dos valores ingresados:

[pagina1.html](#)

```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <form action="pagina2.php" method="post">
      Ingrese primer valor:
      <input type="text" name="valor1">
      <br>
      Ingrese segundo valor:
      <input type="text" name="valor2">
      <br>
      <input type="radio" name="radio1" value="suma">sumar
      <br>
      <input type="radio" name="radio1" value="resta">restar
      <br>
      <input type="submit" name="operar">
    </form>
  </body>

</html>
```

Es importante notar que se trata nuevamente de un archivo HTML puro, que no tiene código PHP. La entrada de los dos números se efectúa en dos controles



```
<input type="text" name="valor1">
<input type="text" name="valor2">
```

Es importante notar que cada text tiene un name DIFERENTE.

Para seleccionar el tipo de operación a efectuar disponemos dos controles de tipo radio:



```
<input type="radio" name="radiol" value="suma">sumar  
<input type="radio" name="radiol" value="resta">restar
```

Ahora vemos que los dos controles de tipo radio tienen el MISMO nombre. Esto es necesario para que el navegador sepa que los dos controles están relacionados (recordar que cuando uno selecciona un radio se debe deseleccionar el otro)

Desde la otra página accederemos al value del control seleccionado.

Por último disponemos un control de tipo submit para el envío de los datos del formulario.

El código de la página que procesa el formulario, llamada:"pagina2.php" (la que indicamos en el elemento FORM del formulario) es:

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        if ($_POST['radio1'] == "suma") {
            $suma = $_POST['valor1'] + $_POST['valor2'];
            echo "La suma es:" . $suma;
        } else {
            if ($_POST['radio1'] == "resta") {
                $resta = $_POST['valor1'] - $_POST['valor2'];
                echo "La resta es:" . $resta;
            }
        }
    ?>
</body>

</html>
```

El vector asociativo `$_POST` tiene tres componentes:



```
$_POST['radio1']
$_POST['valor1']
$_POST['valor2']
```

En la componente `$_POST['radio1']` almacena la cadena "suma" o "resta" según cual se seleccionó en el formulario.

Con dos if verificamos cual operación está seleccionada y procedemos a efectuarla:

```
● ● ●

if ($_POST['radio1'] == "suma") {
    $suma = $_POST['valor1'] + $_POST['valor2'];
    echo "La suma es:" . $suma;
} else {
    if ($_POST['radio1'] == "resta") {
        $resta = $_POST['valor1'] - $_POST['valor2'];
        echo "La resta es:" . $resta;
    }
}
```

Problema propuesto.

Solicitar que se ingrese por teclado el nombre de una persona y disponer tres controles de tipo radio que nos permitan seleccionar si la persona: 1-no tiene estudios, 2-estudios primarios y 3-estudios secundarios.

En la página que procesa el formulario mostrar el nombre de la persona y un mensaje indicando el tipo de estudios que posee.

pagina1.php

```
● ● ●

<html>

  <head>
    <title>Problema</title>
  </head>

  <body>

    <form action="pagina2.php" method="post">
      Ingrese Nombre:
      <input type="text" name="nombre">
      <br>
      Estudios:
      <br>
      <input type="radio" name="radio1" value="sin">Sin estudios
      <br>
      <input type="radio" name="radio1" value="primario">Estudios primarios
      <br>
      <input type="radio" name="radio1" value="secundario">Estudios secundarios
      <br>
      <input type="submit" value="Confirmar">
    </form>

  </body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
echo $_REQUEST['nombre'];
echo "<br>";
if ($_REQUEST['radio1'] == "sin") {
    echo "Sin estudios.";
}
if ($_REQUEST['radio1'] == "primario") {
    echo "Estudios primarios.";
}
if ($_REQUEST['radio1'] == "secundario") {
    echo "Estudios secundarios.";
}

?>

</body>

</html>
```

FORMULARIO (control checkbox)

El control checkbox tiene dos estados:
seleccionado o no seleccionado.

Para analizar este control utilizaremos prácticamente el mismo ejemplo que con el objeto radio:

Implementar un formulario que solicite la carga de dos enteros, uno en cada text. Disponer dos controles de tipo checkbox que nos permitan seleccionar si queremos sumar y/o restar los valores ingresados.

El formulario html tiene el siguiente código:

```
<head>
  <title>Problema</title>
</head>

<body>
  <form action="pagina2.php" method="post">
    Ingrese primer valor:
    <input type="text" name="valor1">
    <br>
    Ingrese segundo valor:
    <input type="text" name="valor2">
    <br>
    <input type="checkbox" name="check1">sumar
    <br>
    <input type="checkbox" name="check2">restar
    <br>
    <input type="submit" name="operar">
  </form>
</body>

</html>
```

Lo nuevo en este problema son los dos controles de tipo checkbox:



```
<input type="checkbox" name="check1">sumar  
<br>  
<input type="checkbox" name="check2">restar  
<br>
```

Es importante notar que cada checkbox tiene un nombre distinto.

Ahora veamos el código de la página que procesa el formulario:



```
<?php
if (isset($_POST['check1'])) {
    $suma = $_POST['valor1'] + $_POST['valor2'];
    echo "La suma es:" . $suma . "<br>";
}
if (isset($_POST['check2'])) {
    $resta = $_POST['valor1'] - $_POST['valor2'];
    echo "La resta es:" . $resta;
}
?>
```

Si el checkbox no está seleccionado en el formulario no se crea una entrada en el vector asociativo `$_REQUEST`, para saber si existe una determinada componente en un vector se emplea la función `isset`, si retorna `true` significa que existe y por lo tanto el checkbox está seleccionado. Disponemos dos `if` a la misma altura ya que los dos controles de tipo checkbox podrían estar seleccionados.

Ejercicio.

Confeccionar un formulario que solicite la carga del nombre de una persona y que permita seleccionar una serie de deportes que practica (futbol, basket, tennis, voley) Mostrar en la página que procesa el formulario la cantidad de deportes que practica.

```
<html>
  <head>
    <title>Problema</title>
  </head>
  <body>
    <form method="post" action="pagina2.php">
      Ingrese el nombre:
      <input type="text" name="nombre">
      <br>
      Deportes que practica:<br>
      <input type="checkbox" name="check1">Fútbol
      <br>
      <input type="checkbox" name="check2">Basquet
      <br>
      <input type="checkbox" name="check3">Tenis
      <br>
      <input type="checkbox" name="check4">Voley
      <br>
      <input type="submit" value="confirmar">
    </form>
  </body>
</html>
```

```
<?php
$cant = 0;
if (isset($_POST['check1'])) {
  $cant++;
}
if (isset($_POST['check2'])) {
  $cant++;
}
if (isset($_POST['check3'])) {
  $cant++;
}
if (isset($_POST['check4'])) {
  $cant++;
}
echo $_POST['nombre'];
echo " practica $cant deportes."
?>
```

FORMULARIO (control select)

Veamos ahora como podemos rescatar desde PHP el elemento seleccionado de un control HTML de tipo select.

Implementar un formulario que solicite la carga de dos enteros, uno en cada "text".

Disponer un control de tipo select que nos permita seleccionar si queremos sumar o restar los dos valores ingresados:

```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <form action="pagina2.php" method="post">
      Ingrese primer valor:
      <input type="text" name="valor1">
      <br>
      Ingrese segundo valor:
      <input type="text" name="valor2">
      <br>
      <select name="operacion">
        <option value="suma">sumar</option>
        <option value="resta">restar</option>
      </select>
      <br>
      <input type="submit" name="operar">
    </form>
  </body>

</html>
```

Lo nuevo que aparece en este formulario es el control de tipo select.



```
<select name="operacion">
    <option value="suma">sumar</option>
    <option value="resta">restar</option>
</select>
```

Cada opción tiene un valor (value). El seleccionado es el que se enviará a la página que procesa el formulario. El texto que aparece dentro del control es el que disponemos entre las marcas option.

Ahora la página que captura los datos ingresados en el formulario es:

```
<?php
    if ($_POST['operacion'] == "suma") {
        $suma = $_POST['valor1'] + $_POST['valor2'];
        echo "La suma es:" . $suma;
    } else {
        if ($_POST['operacion'] == "resta") {
            $resta = $_POST['valor1'] - $_POST['valor2'];
            echo "La resta es:" . $resta;
        }
    }
?>
```

Problema propuesto.

Confeccionar un formulario que solicite el ingreso del nombre de una persona y un control select (en este último permitir la selección de los ingresos mensuales de la persona: 1-1000,1001-3000,>3000)

En la página que procesa el formulario mostrar un mensaje si debe pagar impuestos a las ganancias (si supera 3000)

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

    <form method="post" action="pagina2.php">
        Ingrese el nombre:
        <input type="text" name="nombre">
        <br>
        Seleccione en que rango se ubica sus ingresos:
        <select name="ingresos">
            <option value="1">1-1000</option>
            <option value="2">1001-3000</option>
            <option value="3">>3000</option>
        </select>
        <br>
        <input type="submit" value="confirmar">
    </form>

</body>

</html>
```

```
● ● ●

<?php
    echo $_POST['nombre'];
    if ($_POST['ingresos'] == 3) {
        echo " debe pagar impuestos.";
    } else {
        echo " no debe pagar impuestos a las ganancias.";
    }
?>
```

FORMULARIO (control textarea)

El control HTML "textarea" se diferencia del "text" en que permite el ingreso de muchas líneas.

Problema

Implementar un formulario web para ingresar el nombre de una persona en un control text y mediante un control de tipo textarea se permita el ingreso del curriculum de dicha persona.

```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <form action="pagina2.php" method="post">
      Ingrese nombre:<input type="text" name="nombre"><br>
      Ingrese su curriculum:<br>
      <textarea name="curriculum"></textarea>
      <br>
      <input type="submit" value="Confirmar">
    </form>
  </body>

</html>
```

La sintaxis de este control es bastante diferente a la del control text:

```
<textarea name="curriculum"></textarea>
```

Si queremos que aparezca inicializado con texto debemos disponerlo en:

```
<textarea name="curriculum">Hola Mundo</textarea>
```

La página PHP que procesa los dos datos ingresados en el formulario es:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        echo "El nombre ingresado:" . $_POST['nombre'];
        echo "<br>";
        echo "El curriculum:" . $_POST['curriculum'];
    ?>
</body>

</html>
```

Ejercicio.

Confeccionar una página que muestre un contrato dentro de un textarea, disponer puntos suspensivos donde el operador debe ingresar un texto. La página que procesa el formulario sólo debe mostrar el contrato con las modificaciones que hizo el operador.

Ej. de un contrato puede ser:

En la ciudad de [.....], se acuerda entre la Empresa [.....]
representada por el Sr. [.....] en su carácter de Apoderado,
con domicilio en la calle [.....] y el Sr. [.....],
futuro empleado con domicilio en [.....], celebrar el presente
contrato a Plazo Fijo, de acuerdo a la normativa vigente de los
artículos 90,92,93,94, 95 y concordantes de la Ley de Contrato de Trabajo N° 20.744.



```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Completar contrato:<br>
        <textarea name="contrato" rows="8" cols="80">
En la ciudad de [.....], se acuerda entre la Empresa [.....] representada por el Sr. [.....] en su carácter de Apoderado, con domicilio en la calle [.....] y el Sr. [.....], futuro empleado con domicilio en [.....], celebrar el presente contrato a Plazo Fijo, de acuerdo a la normativa vigente de los artículos 90,92,93,94, 95 y concordantes de la Ley de Contrato de Trabajo N° 20.744.
</textarea>
        <br>
        <input type="submit" value="Confirmar">
    </form>

</body>

</html>
```



```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        echo $_POST['contrato'];
    ?>
</body>

</html>
```

Vectores

Un Array es una colección de valores. Los array pueden ser unidimensionales (vectores), bidimensionales (matrices) y multidimensionales (más de dos dimensiones)

Los arrays se utilizan ampliamente en el lenguaje PHP.

Se utiliza el delimitador [] para acceder a los diferentes elementos del vector.

Se lo puede crear al vuelo, sin tener que declararlo:



```
$dias[0]=31;  
$dias[1]=28;
```

Luego de estas dos líneas, tenemos creado un vector de dos elementos, a los cuales accedemos por un subíndice.



```
echo $dias[0]; //31  
echo $dias[1]; //28
```

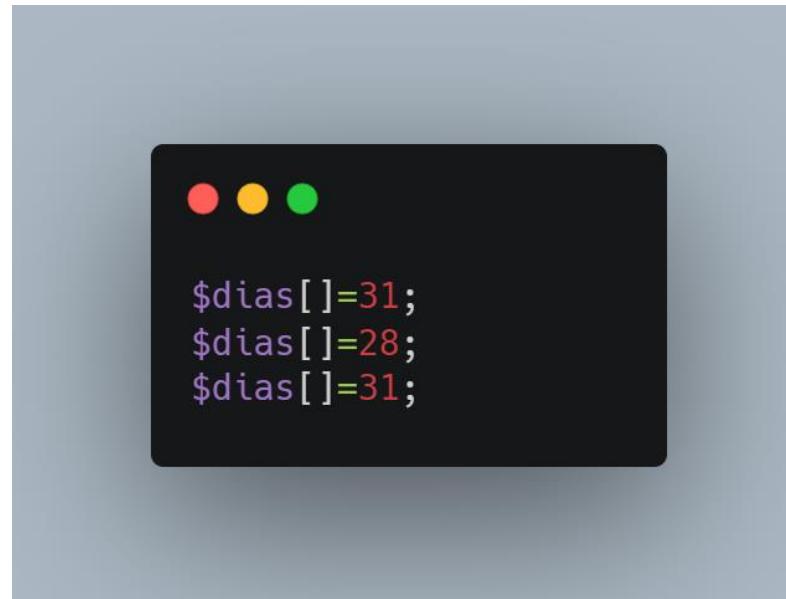
El vector, como podemos ver, puede ir creciendo en forma dinámica, es decir que si ahora hacemos:



Automáticamente comienza a numerarse desde cero.

el vector ahora pasa a tener 3 componentes.

También podemos obviar el subíndice cuando asignamos los valores:



Si necesitamos conocer el tamaño del vector en cualquier momento podemos llamar a la función count.



```
echo count($dias); //3
```

Si queremos imprimir todos los elementos en la página podemos hacer:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

    <?php
        $nombres[] = "juan";
        $nombres[] = "pedro";
        $nombres[] = "ana";
        for ($f = 0; $f < count($nombres); $f++) {
            echo $nombres[$f];
            echo "<br>";
        }
    ?>

</body>

</html>
```

La función sizeof(<nombre del vector>) es equivalente a count
Otra forma de inicializar un vector es definirlo e inicializarlo
simultáneamente:



```
$edades=array("menores","jovenes","adultos");
```

Estamos definiendo el vector edades con tres componentes, numeradas automáticamente de cero a dos.

Acotaciones

Cuando tenemos que recorrer en forma completa un vector en PHP es muy común utilizar la estructura 'foreach'. Veamos el mismo ejemplo anterior para recorrer el vector \$nombres:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$nombres[] = "juan";
$nombres[] = "pedro";
$nombres[] = "ana";
foreach ($nombres as $nombre) {
    echo $nombre;
    echo "<br>";
}
?>

</body>

</html>
```

En cada repetición del 'foreach' la variable \$nombre almacena una componente del vector \$nombres, luego dentro del 'foreach' mostramos el contenido de la variable \$nombre:

```
foreach ($nombres as $nombre) {  
    echo $nombre;  
    echo "<br>";  
}
```

Problema propuesto.

Definir un vector con los nombres de los días de la semana. Luego imprimir el primero y el último elemento del vector.

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$diaSemana[ ] = "Lunes";
$diaSemana[ ] = "Martes";
$diaSemana[ ] = "Miércoles";
$diaSemana[ ] = "Jueves";
$diaSemana[ ] = "Viernes";
$diaSemana[ ] = "Sábado";
$diaSemana[ ] = "Domingo";
echo "Primer elemento:" . $diaSemana[0];
echo "<br>";
echo "Último elemento:" . $diaSemana[6];
?>

</body>

</html>
```

Creación de un archivo de texto.

Una actividad fundamental es poder registrar información en el servidor (no como hemos estado haciendo hasta el momento generando sólo una página con los datos cargados)

Para la registraciónde datos en el servidor disponemos de dos herramientas que se complementan en muchos casos (archivos de texto y bases de datos)

En este apartado veremos como crear un archivo de texto y añadir datos al mismo. Lo presentaremos al tema resolviendo un problema: Implementación de un libro de visitas.

Para resolver este problema plantearemos dos páginas, un formulario para realizar la carga del nombre del visitante y sus comentarios (disponemos un objeto de tipo "text" y otro de tipo "textarea"):

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese su nombre:
        <input type="text" name="nombre">
        <br>
        Comentarios:
        <br>
        <textarea name="comentarios" rows="10" cols="40"></textarea>
        <br>
        <input type="submit" value="Registrar">
    </form>
</body>

</html>
```

hemos agregado al control textarea, las propiedades rows y cols que dimensionan el mismo en la pantalla:



```
<textarea name="comentarios" rows="10" cols="40"></textarea>
```

Veamos ahora la página (pagina2.php) que graba los datos cargados en el formulario en un archivo:

```
<?php
$ar = fopen("datos.txt", "a") or
die("Problemas en la creacion");
fputs($ar, $_POST['nombre']);
fputs($ar, "\n");
fputs($ar, $_POST['comentarios']);
fputs($ar, "\n");
fputs($ar, "-----");
fputs($ar, "\n");
fclose($ar);
echo "Los datos se cargaron correctamente.";
?>
```

En primer lugar creamos o abrimos el archivo de texto "datos.txt". El segundo parámetro de la función fopen indica la forma de apertura de archivo "**a**" (lo crea o si ya existe el archivo lo abre para añadir datos al final), "**W**" (crea el archivo de texto, si existe borra su contenido) y la última forma de apertura del archivo es "**R**" (abre el archivo para su lectura)

Como en este problema nos interesa que el archivo vaya creciendo con los datos que aportan los visitantes al sitio lo abrimos para añadir, parámetro "a".

La función fopen retorna una referencia al archivo y la almacenamos en una variable llamada \$ar. Si el archivo no se puede abrir, se ejecuta la instrucción que se encuentra luego del operador "or" en nuestro caso llamamos a la función die que finaliza la ejecución del programa PHP mostrando como mensaje el texto que le pasamos a dicha función (por ejemplo si el disco duro del servidor está lleno no se podrá crear el archivo de texto)

```
● ● ●  
$ar = fopen("datos.txt", "a") or  
die("Problemas en la creacion");
```

Para la grabación de datos utilizamos la función fputs que tiene dos parámetros: la referencia al archivo donde grabamos y el string a grabar.



Para el salto de línea en el archivo de texto, usamos los caracteres \n. De esta forma cuando leamos el archivo de texto lo haremos línea a línea.

Cuando dejamos de trabajar con el archivo llamamos a la función fclose.

Hay que tener muy presente que el archivo se almacena en el servidor y no en la máquina de la persona que está navegando. Si nos dirigimos a la carpeta c:\xampp\htdocs encontraremos el archivo "datos.txt", tenga en cuenta que está en la máquina donde se ejecutó el script de PHP (normalmente en un servidor web). Luego veremos como leer el contenido del archivo y mostrarlo en otra página del sitio (En nuestro caso como utilizamos el equipo como cliente/servidor el archivo datos.txt se crea en la misma carpeta donde se alojan nuestras páginas php)

Ejercicio.

Confeccionar un programa en PHP que permita hacer el pedido de pizzas via internet.

El formulario debe ser:

Nombre:[.....]

Direccion:[.....]

Jamon y queso:[x]

Cantidad[...]

Napolitana:[x]

Cantidad[...]

Muzzarella:[x]

Cantidad[...]

[Confirmar]

Para el ingreso del nombre, dirección y cantidad de pizzas de cada tipo disponer objetos de la clase "text".

Disponer tres objetos de tipo "checkbox" para seleccionar los tipos de pizzas.

Por último disponer un botón para el envío de datos: "submit".

Grabar en un archivo de texto llamado "pedidos.txt" cada pedido, separados por una línea de puntos entre cada pedido.

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

    <form method="post" action="pagina2.php">
        Nombre:
        <input type="text" name="nombre" size="30">
        <br>
        Dirección:
        <input type="text" name="direccion" size="40">
        <br>
        Jamon y Queso:
        <input type="checkbox" name="jamonqueso">
        <input type="text" name="cantjamonqueso" size="3">
        <br>
        Napolitana:
        <input type="checkbox" name="napolitana">
        <input type="text" name="cantnapolitana" size="3">
        <br>
        Muzzarella:
        <input type="checkbox" name="muzzarella">
        <input type="text" name="cantmuzzarella" size="3">
        <br>
        <input type="submit" value="Confirmar">
    </form>

</body>

</html>
```

```
<?php
    $ar = fopen("datos.txt", "a") or
        die("Problemas en la creacion");
    fputs($ar, "Nombre:");
    fputs($ar, $_POST['nombre']);
    fputs($ar, "\n");
    fputs($ar, "Dirección:");
    fputs($ar, $_POST['direccion']);
    fputs($ar, "\n");
    if (isset($_POST['jamonqueso'])) {
        fputs($ar, "Cantidad de Jamón y Queso:");
        fputs($ar, $_POST['cantjamonqueso']);
        fputs($ar, "\n");
    }
    if (isset($_POST['napolitana'])) {
        fputs($ar, "Cantidad de Napolitana:");
        fputs($ar, $_POST['cantnapolitana']);
        fputs($ar, "\n");
    }
    if (isset($_POST['muzzarella'])) {
        fputs($ar, "Cantidad de Muzzarella:");
        fputs($ar, $_POST['cantmuzzarella']);
        fputs($ar, "\n");
    }
    fputs($ar, "-----");
    fputs($ar, "\n");
    fclose($ar);
    echo "El pedido se cargó correctamente.";
?>
```

Lectura de un archivo de texto.

Para la lectura de un archivo de texto contamos con la función fgets. Además debemos abrir el archivo para lectura.

La apertura para leer:

```
● ● ●  
$ar=fopen("datos.txt","r") or  
die("No se pudo abrir el archivo");
```

Para leer:

```
● ● ●  
$linea=fgets($ar);
```

Veamos como mostrar por pantalla el contenido del archivo "datos.txt" creado en el punto anterior:

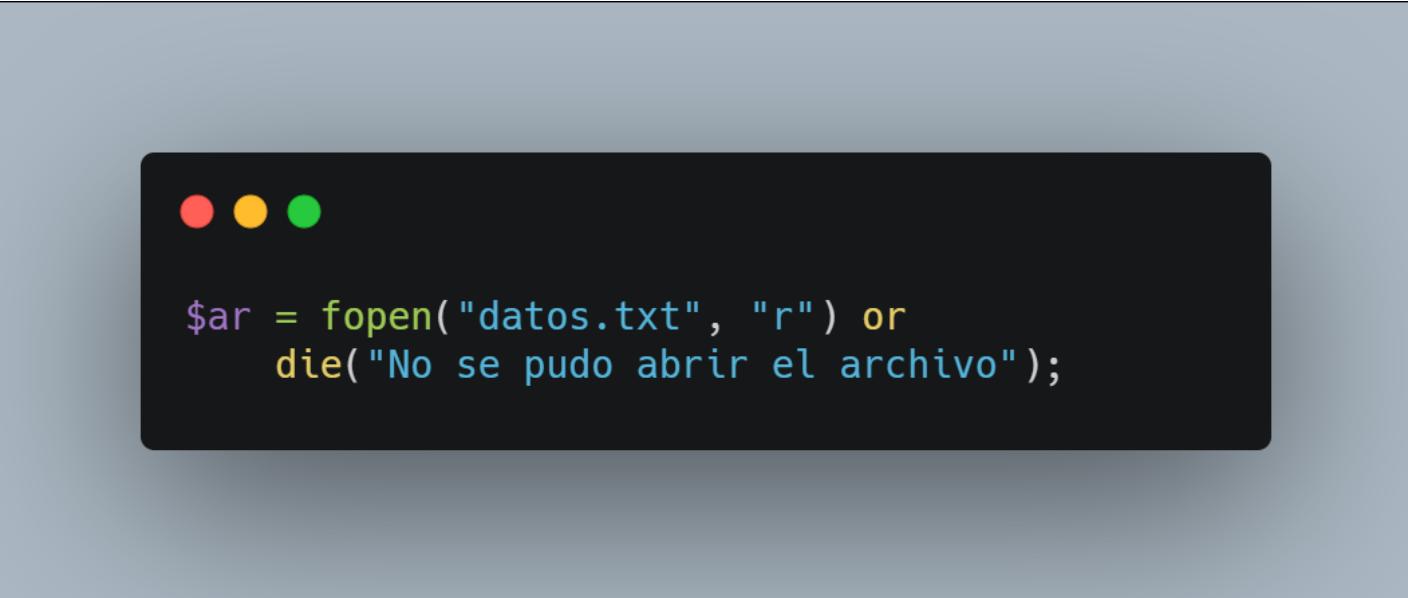
```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $ar = fopen( "datos.txt", "r" ) or
            die( "No se pudo abrir el archivo" );
        while ( !feof($ar) ) {
            $linea = fgets($ar);
            $lineasalto = nl2br($linea);
            echo $lineasalto;
        }
        fclose($ar);
    ?>
</body>

</html>
```

Lo primero que debemos identificar es la forma de apertura del archivo:



```
$ar = fopen("datos.txt", "r") or  
die("No se pudo abrir el archivo");
```

El segundo parámetro de fopen es "r" es decir read (apertura para lectura), si el archivo no existe por ejemplo se ejecuta la función die que finaliza el programa mostrando el string correspondiente.

La función feof retorna true si se ha llegado al final del archivo en caso contrario retorna false. Para que se impriman todas las líneas del archivo se plantea una estructura repetitiva que se ejecuta mientras no se llegue al final de archivo (el operador lógico not en PHP es el carácter !):



Dentro de la estructura repetitiva leemos una línea completa del archivo de texto con la función fgets:

```
● ● ●  
$linea = fgets($ar);
```

La variable \$linea contiene una línea completa del archivo de texto, inclusive el salto de línea (\n). Como el navegador no hace un salto de línea con este carácter, debemos convertir dicho carácter al elemento
 propia de HTML. La función que realiza esta actividad se llama nl2br (new line to br). El resultado se almacena en una nueva variable que es la que realmente imprimimos:

```
● ● ●  
$lineasalto = nl2br($linea);  
echo $lineasalto;
```

Ejercicio.

Confeccionar un programa que muestre el archivo de pedido de pizzas via internet del punto anterior.

Recordemos que creamos el archivo de texto llamado pedidos.txt (grabar la página php en el mismo directorio donde se encuentra el archivo pedidos.txt)

```
<html>
  <head>
    <title>Problema</title>
  </head>
  <body>
    <?php
      $ar = fopen( "pedidos.txt", "r" ) or
        die( "No se pudo abrir el archivo" );
      while ( !feof($ar) ) {
        $linea = fgets($ar);
        $lineasalto = nl2br($linea);
        echo $lineasalto;
      }
      fclose($ar);
    ?>
  </body>
</html>
```

Vectores (asociativos)

Este tipo de vectores no es común a otros lenguajes, pero en PHP son de uso indispensable en distintas situaciones (ya lo empleamos cuando recuperamos información de un formulario accediendo al vector `$_REQUEST` que crea PHP en forma automática, cuando accedamos a datos de una base de datos también lo emplearemos etc.) Un vector asociativo permite acceder a un elemento del vector por medio de un subíndice de tipo string.

Inicialmente uno piensa que esto nos complica las cosas, como veremos más adelante la realidad nos demuestra lo contrario.

Como ejemplo, consideremos que deseamos guardar en un vector el DNI, nombre y dirección de una persona. Empleando un vector con subíndice entero la solución sería:



```
<?php
$registro[] = "20456322";
$registro[] = "Martinez Pablo";
$registro[] = "Colon 134";
?>
```

De esta forma debemos recordar que cuando deseamos mostrar el nombre de la persona debemos acceder al subíndice 1. Esto es sencillo si tenemos un vector con tres elementos, pero que sucede si debemos almacenar 20 datos relacionados en un vector.

Un vector asociativo se define de la siguiente forma:

```
<?php
$registro['dni'] = "20456322";
$registro['nombre'] = "Martinez Pablo";
$registro['direccion'] = "Colon 134";
echo $registro['nombre'];
?>
```

Ahora vemos que para imprimir el nombre de la persona no debemos recordar una posición dentro de un vector sino un nombre de clave. Esto se hace indispensable cuando administramos un conjunto de datos grandes.

En un vector asociativo toda componente está asociada a una clave.

Otras formas de crear un vector asociativo:

```
<?php
$registro = array(
    'dni' => '20456322',
    'nombre' => 'Martinez Pablo',
    'direccion' => 'Colon 134'
);
echo $registro['dni'];
?>
```

Acotaciones

Cuando tenemos que recorrer en forma completa un vector asociativo en PHP podemos utilizar la estructura 'foreach'. Veamos un ejemplo:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$articulo = array(
    'codigo' => 1,
    'descripcion' => 'manzanas',
    'precio' => 30.25
);

foreach ($articulo as $clave => $valor) {
    echo 'Para la clave : ' . $clave . " almacena el valor: " . $valor;
    echo "<br>";
}
?>

</body>

</html>
```

En cada repetición del 'foreach' la variable \$clave almacena el subíndice del vector y la variable \$valor contiene el valor de la componente del vector:

```
● ● ●

foreach ($articulo as $clave => $valor) {
    echo 'Para la clave : ' . $clave . " almacena el valor: " . $valor;
    echo "<br>";
}
```

Resultado:

```
● ● ●

Para la clave :codigo almacena el valor: 1
Para la clave :descripcion almacena el valor: manzanas
Para la clave :precio almacena el valor: 30.25
```

Desarrollaremos una aplicación que solicite la carga de 5 números en un formulario HTML y al presionar un botón se calcule la suma en el servidor.



```
<form action="pagina2.php" method="post">
    Ingrese primer valor:
    <input type="text" name="valor1">
    <br>
    Ingrese segundo valor:
    <input type="text" name="valor2">
    <br>
    Ingrese tercer valor:
    <input type="text" name="valor3">
    <br>
    Ingrese cuarto valor:
    <input type="text" name="valor4">
    <br>
    Ingrese quinto valor:
    <input type="text" name="valor5">
    <br>
    <input type="submit">
</form>
```



```
<?php
$suma = 0;
foreach ($_POST as $clave => $valor) {
    echo "Valor: " . $valor;
    echo "<br>";
    $suma = $suma + $valor;
}
echo "La suma es: " . $suma;
?>
```

Como vemos podemos recorrer en forma completa los 5 elementos del vector '`$_POST`' y acceder tanto a su valor como su clave:

```
● ● ●  
$suma = 0;  
foreach ($_POST as $clave => $valor) {  
    echo "Valor: " . $valor;  
    echo "<br>";  
    $suma = $suma + $valor;  
}
```

Si solo queremos acceder a los valores del vector asociativo y no a sus claves, luego podemos codificar la sintaxis del `foreach`:

```
● ● ●  
foreach ($_POST as $valor) {  
    echo "Valor: " . $valor;  
    echo "<br>";  
    $suma = $suma + $valor;  
}
```

Problema propuesto.

Crear un vector asociativo que almacene las claves de acceso de 5 usuarios de un sistema. Acceder a cada componente por su nombre. Imprimir una componente del vector.



```
<?php
$clave['ana'] = "abc123";
$clave['luis'] = "z67yui";
$clave['carlos'] = "sdf3sdf";
$clave['laura'] = "dsf3k32";
$clave['pedro'] = "axldds23";

echo "La clave de luis es:" . $clave['luis'];
?>
```

Funciones en PHP

La sintaxis para la definición de una función en PHP es:

```
● ● ●  
function [nombre de la función]([parámetros])  
{  
    [algoritmo]  
}
```

Implementaremos una función que muestre un título centrado en pantalla, y la llamaremos posteriormente dos veces:

```
<?php  
function mostrartitulo($men)  
{  
    echo "<h1 style=\"text-align:center\>";  
    echo $men;  
    echo "</h1>";  
}  
  
mostrartitulo("Primer título");  
echo "<br>";  
mostrartitulo("Segundo segundo");  
  
?>
```

Para mostrar el título centrado utilizamos el elemento h1 de HTML y definimos un estilo centrado para el mismo.

Si vemos la función, notamos que lo más trabajoso es definir todos los elementos HTML para crear el título. Es importante notar que en PHP para introducir las dobles comillas dentro de un string debemos anteceder el carácter '\'; para introducir el carácter '\' debemos escribir dos barras \\.

Las llamadas a la función las hacemos por medio de su nombre y pasamos el único parámetro que requiere:

```
mostraritulo("Primer titulo");
echo "<br>";
mostraritulo("Segundo segundo");
```

Las funciones nos permiten tener un programa más ordenado y facilitan la reutilización del código.

Más adelante veremos como hacer archivos con rutinas comunes a muchas páginas.

Una función puede retornar un dato, supongamos que necesitamos una función que nos retorne el promedio de dos valores, el código sería:

```
<?php
function retornarpromedio($valor1, $valor2)
{
    $pro = $valor1 / $valor2;
    return $pro;
}

$v1 = 100;
$v2 = 50;
$p = retornarpromedio($v1, $v2);
echo $p;
?>
```

Cuando una función retorna un dato debemos emplear la palabra clave return seguida del valor que devuelve.

En la llamada a la función el valor retornado se almacena generalmente en una variable:



```
$p = retornarpromedio($v1, $v2);
```

Si queremos que retorne más de un dato debemos emplear parámetros por referencia.

Supongamos que necesitamos ahora que una función nos retorne el cuadrado y cubo de un número:

```
<?php
function cuadradocubo($valor, &$cuad, &$cub)
{
    $cuad = $valor * $valor;
    $cub = $valor * $valor * $valor;
}
cuadradocubo(2, $c1, $c2);
echo "El cuadrado de 2 es:" . $c1 . "<br>";
echo "El cubo de 2 es:" . $c2;
?>
```

Es decir, cuando le antecedemos el carácter ampersand al parámetro, es por referencia. El objetivo es asignarle cierto valor al parámetro y posteriormente el dato quedará almacenado en la variable que le pasamos a la función.

```
function cuadradocubo($valor, &$cuad, &$cub)
{
    $cuad = $valor * $valor;
    $cub = $valor * $valor * $valor;
}
```

El parámetro \$cuad se almacena en la variable \$c1 y el parámetro \$cub se almacena en \$c2. Es la forma más adecuada de modificar variables dentro de una función.

Ejercicio.

Confeccionar un formulario que solicite la carga del nombre de usuario y su clave en dos oportunidades. En la página que se procesan los datos del formulario implementar una función que imprima un mensaje si las dos claves ingresadas son distintas.

```
<?php  
  
<form action="pagina2.php" method="post">  
    Ingrese nombre de usuario:  
    <input type="text" name="usuario">  
    <br>  
    Ingrese su clave:  
    <input type="password" name="clave1">  
    <br>  
    Repita la clave  
    <input type="password" name="clave2">  
    <br>  
    <input type="submit" name="Confirmar">  
</form>
```

```
<?php  
  
function verificarClaves($cla1, $cla2)  
{  
    if ($cla1 != $cla2)  
        echo "Las claves ingresadas son distintas";  
}  
  
verificarClaves($_POST['clave1'], $_POST['clave2']);  
  
?>
```

Base de datos (MySQL)

Uno de los empleos principales de PHP es el acceso a una base de datos en el servidor. Las operaciones básicas se hacen empleando como lenguaje el SQL.

PHP implementa distintas funciones según la base de datos a emplear. Existen funciones actualmente para acceder a las siguientes servidores de bases de datos:

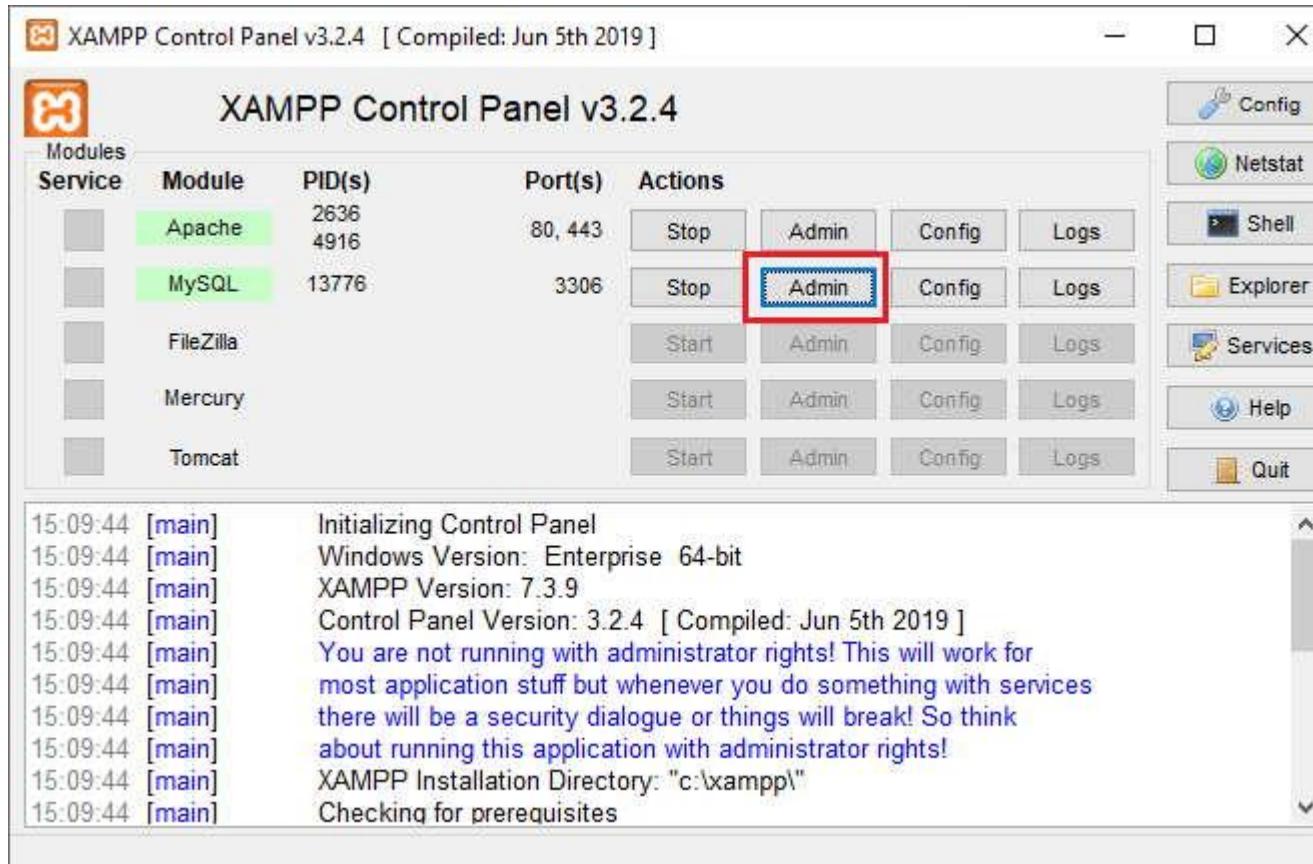
- MySQL
- MariaDB
- Microsoft SQL Server
- Oracle
- PostgreSQL
- SysBase
- FrontBase
- Informix
- InterBase
- Ingres
- mSQL
- dBase
- SQLite

El más empleado en la actualidad en la web es el gestor de base de datos MySQL. Cuando instaló el entorno de XAMPP en un principio para trabajar con PHP, se instaló el MySQL o MariaDB.

Para crear una base de datos el XAMPP instala también un programa codificado en PHP que nos permite interactuar con el MySQL o MariaDB.

Este programa se llama PHPMyAdmin (como veremos nos permite crear las bases de datos, tablas, índices, usuarios etc.)

Para iniciar el PHPMyAdmin debemos presionar el botón "Admin" de MySQL en el panel de control de XAMPP:



Como podemos ver la interfaz del PHPMyAdmin es un programa que se ejecuta en la web:

The screenshot shows the phpMyAdmin configuration interface running in a web browser. The URL is `localhost / 127.0.0.1 | phpMyAdmin`. The main menu on the left includes options like Nueva, calendario, facturación, information_schema, mysql, performance_schema, phpmyadmin, sistema, test, and tutorialesya.

The central area displays several configuration panels:

- Configuraciones generales**: Includes a dropdown for "Cotejamiento de la conexión al servidor" set to "utf8mb4_unicode_ci".
- Configuraciones de apariencia**: Shows the language set to "Español - Spanish" and the theme set to "pmahomme". A "Tamaño de fuente" (Font size) dropdown is also present.
- Servidor de base de datos**: Provides detailed information about the server:
 - Servidor: 127.0.0.1 via TCP/IP
 - Tipo de servidor: MariaDB
 - Conexión del servidor: No se está utilizando SSL
 - Versión del servidor: 10.4.6-MariaDB - mariadb.org binary distribution
 - Versión del protocolo: 10
 - Usuario: root@localhost
 - Conjunto de caracteres del servidor: cp1252 West European (latin1)
- Servidor web**: Lists the web server environment:
 - Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.9
 - Versión del cliente de base de datos: libmysql - mysqld 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675f6d72e5cf0f267f48e \$
 - extensión PHP: mysqli curl mbstring
 - Versión de PHP: 7.3.9
- phpMyAdmin**: Links to various resources:
 - Acerca de esta versión: 4.9.0.1, versión estable más reciente: 4.9.1
 - Documentación
 - Página oficial de phpMyAdmin
 - Contribuir
 - Obtener soporte
 - Lista de cambios
 - Licencia

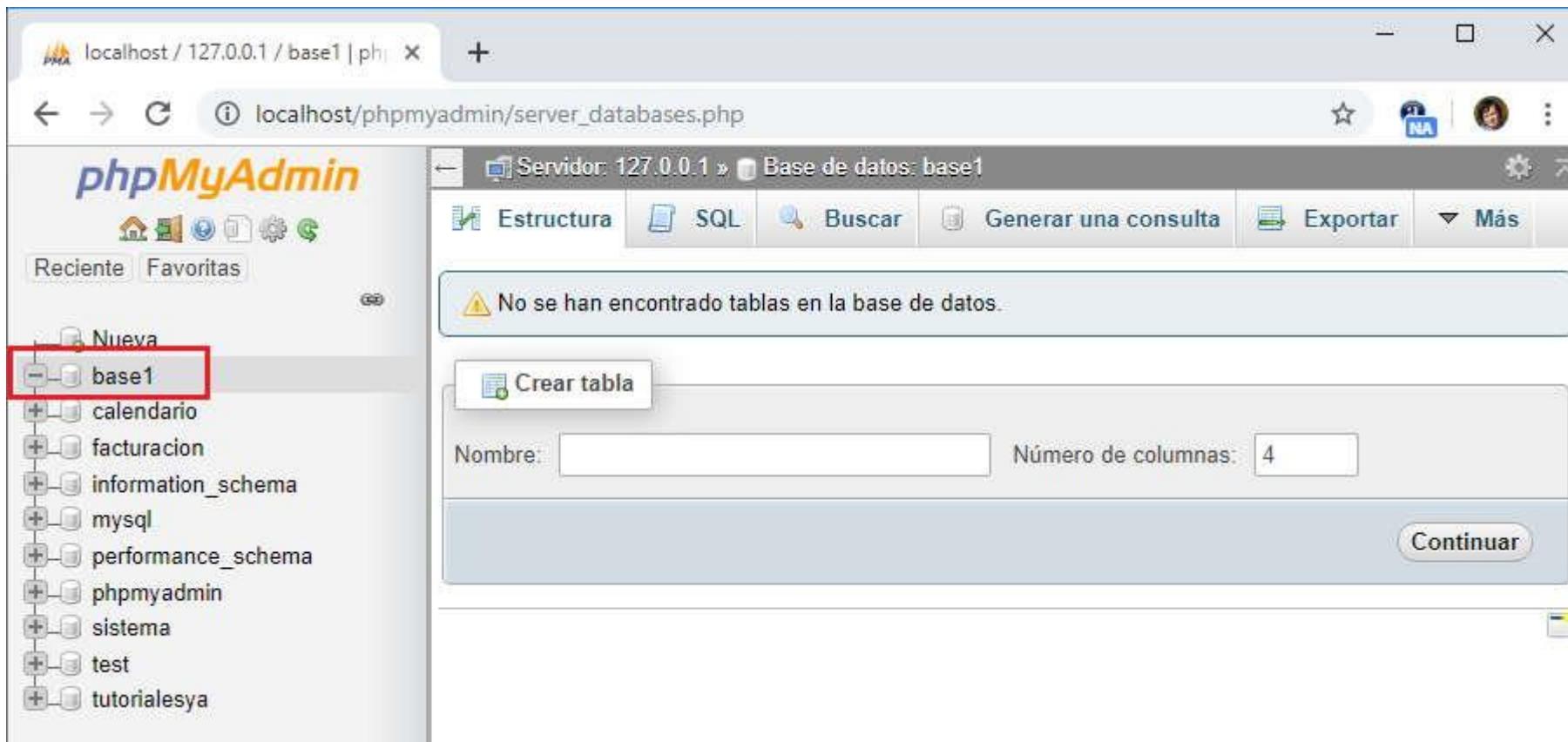
At the bottom, there is a "Consola" tab.

Para crear una base de datos procedemos a seleccionar la pestaña "Base de datos" e ingresamos como nombre "base1" y presionamos el botón crear:

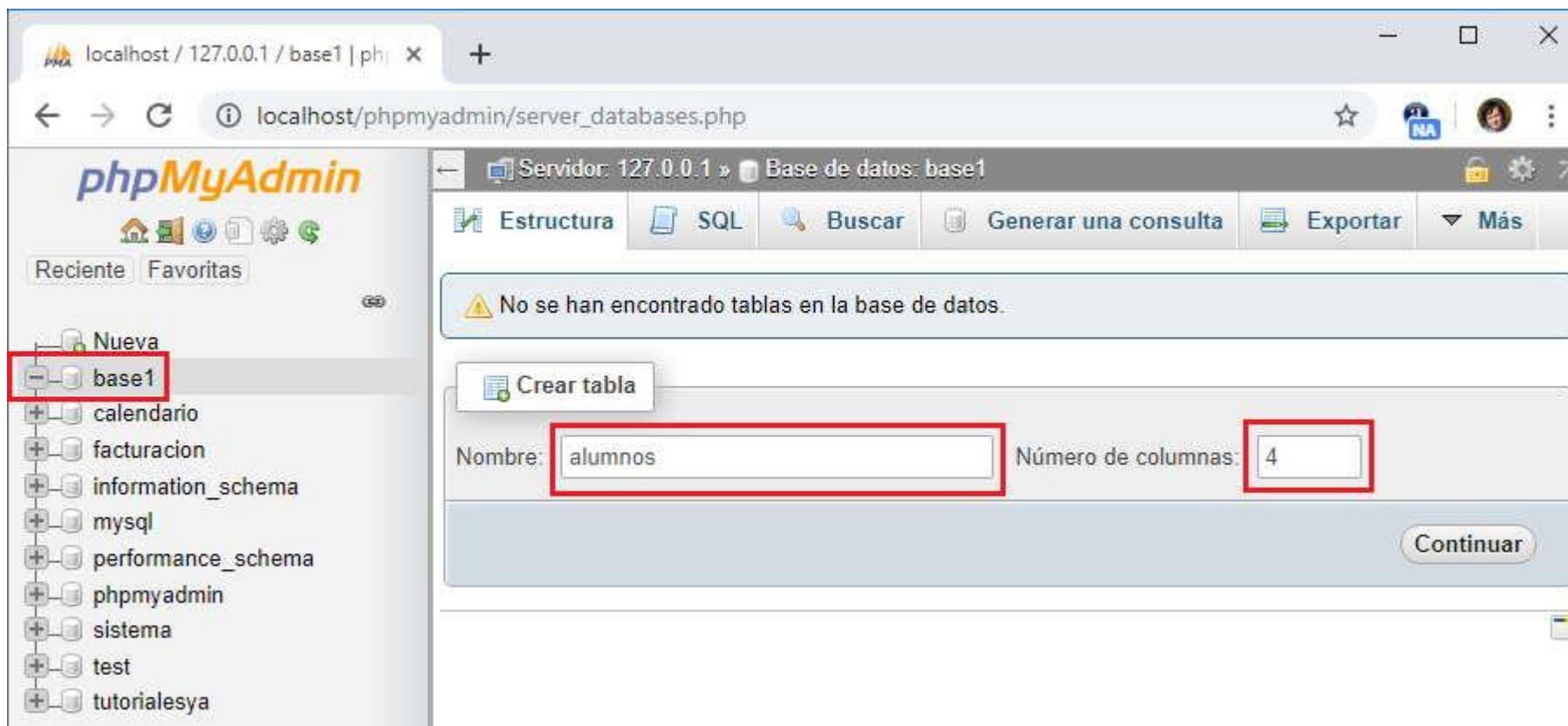
The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar lists existing databases: calendario, facturacion, information_schema, mysql, performance_schema, phpmyadmin, sistema, test, and tutorialesya. The main panel has tabs for 'Bases de datos', 'SQL', 'Estado actual', 'Cuentas de usuarios', and 'Más'. The 'Bases de datos' tab is selected and highlighted with a red box. Below it, there's a 'Crear base de datos' button and a text input field containing 'base1', which is also highlighted with a red box. A dropdown menu next to the input field shows 'Cotejamiento'. At the bottom, a table lists the current databases with their character sets and options to 'Seleccionar privilegios'.

Base de datos	Cotejamiento	Acción
calendario	latin1_swedish_ci	Seleccionar privilegios
facturacion	latin1_swedish_ci	Seleccionar privilegios
information_schema	utf8_general_ci	Seleccionar privilegios

Luego de crear la base de datos podemos ver que aparece en el lado izquierdo:



En el lado derecho de la pantalla podemos ahora ingresar el nombre de una tabla y la cantidad de campos que tendrá (crearemos una tabla llamada alumnos con 4 campos):



La estructura de la tabla es:

```
codigo int auto_increment primary key  
nombre varchar(50)  
mail varchar(70)  
codigocurso int
```

En el PHPMyAdmin ingresamos:

The screenshot shows the 'Estructura' (Structure) tab of the PHPMyAdmin interface for a table named 'alumnos'. The table has four columns: 'codigo', 'nombre', 'mail', and 'codigocurso'. The 'codigo' column is defined as INT and is highlighted with a red box. The 'nombre' column is defined as VARCHAR(50) and is also highlighted with a red box. The 'mail' column is defined as VARCHAR(70) and is highlighted with a red box. The 'codigocurso' column is defined as INT and is highlighted with a red box. Below the table structure, there are sections for 'Comentarios de la tabla:' (Table comments), 'Cotejamiento:' (Check constraints), and 'definición de la PARTICIÓN:' (Partition definition). At the bottom right, there are 'Previsualizar SQL' (Preview SQL) and 'Guardar' (Save) buttons.

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejan
codigo	INT		Ninguno	
nombre	VARCHAR	50	Ninguno	
mail	VARCHAR	70	Ninguno	
codigocurso	INT		Ninguno	

Es importante también hacer notar que en el campo código debemos marcar en Índice el valor "primary" y tildar la columna A_I (Auto_Increment):

Servidor: 127.0.0.1 » Base de datos: base1 » Tabla: alumnos

Examinar Estructura SQL Buscar Insertar Exportar Más

Nombre de la tabla: alumnos Agregar 1 columna(s)

Estructura

Atributos	Nulo	Índice	A.I.	Comentarios	Virtualidad
▼	□	PRIMARY	<input checked="" type="checkbox"/>		
▼	□	---			
▼	□	---			
▼	□	---			

ento:

Por último presionamos el botón guardar y ya tenemos la tabla "alumnos" creada en la base de datos "base1":

The screenshot shows the MySQL Workbench interface for creating a new table named 'alumnos'. The table structure includes four columns: 'codigo' (INT), 'nombre' (VARCHAR(50)), 'mail' (VARCHAR(70)), and 'codigocurso' (INT). The 'Guardar' (Save) button at the bottom right is highlighted with a red box.

Servidor: 127.0.0.1 » Base de datos: base1 » Tabla: alumnos

Examinar Estructura SQL Buscar Insertar Exportar Más

Nombre de la tabla: alumnos Agregar 1 columna(s)

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento
codigo	INT		Ninguno	
nombre	VARCHAR	50	Ninguno	
mail	VARCHAR	70	Ninguno	
codigocurso	INT		Ninguno	

Comentarios de la tabla: Cotejamiento: Motores de almacén:

definición de la PARTICIÓN:

Dividido por: (Expresión o lista de columnas)

Particiones:

Previsualizar SQL Guardar

La tabla almacenará datos de alumnos que desarrollarán cursos de programación en PHP, ASP y JSP.

El código del alumno es de tipo numérico (int) y al indicar que es auto_increment se generará automáticamente por el gestor de base de datos.

Los campos nombre y mail son de tipo varchar (podemos almacenar cualquier carácter) y por último el campo codigocurso representa el curso a tomar por el alumno (1=PHP, 2=ASP y 3=JSP)

El campo clave de esta tabla es el código de alumno (es decir no podemos tener dos alumnos con el mismo código, no así el nombre del alumno que puede eventualmente repetirse)

En los próximos conceptos comenzaremos a ver como desde PHP podemos comunicarnos con la base de datos "base1" y acceder a la tabla "alumnos" para ejecutar los comandos SQL más comunes como pueden ser: select, insert, delete, update etc.

INSERT (Alta de registros en una tabla)

Luego de crear una base de datos y sus tablas
(Vamos a trabajar con la base de datos ya creada:
base1, que contiene la tabla alumnos), veremos
como agregar registros.

Para añadir datos en la tabla empleamos el
comando SQL llamado insert.

Necesitamos dos páginas para este proceso, una
será el formulario de carga de datos y la siguiente
será la que efectúe la inserción en la tabla.

Formulario de carga de datos:

pagina1.html

```
● ● ●

<html>

  <head>
    <title>Problema</title>
  </head>

  <body>
    <h1>Alta de Alumnos</h1>
    <form action="pagina2.php" method="post">
      Ingrese nombre:
      <input type="text" name="nombre"><br>
      Ingrese mail:
      <input type="text" name="mail"><br>
      Seleccione el curso:
      <select name="codigocurso">
        <option value="1">PHP</option>
        <option value="2">ASP</option>
        <option value="3">JSP</option>
      </select>
      <br>
      <input type="submit" value="Registrar">
    </form>
  </body>

</html>
```

El formulario es bastante similar a los que venimos desarrollando en puntos anteriores, tal vez lo distinto es cómo emplearemos el control "select" del curso a desarrollar:

```
<select name="codigocurso">
  <option value="1">PHP</option>
  <option value="2">ASP</option>
  <option value="3">JSP</option>
</select>
```

Cada opción tiene su respectivo valor (en este caso los números 1,2 y 3) y los textos a mostrar PHP, ASP y JSP. El dato que se envía a la otra página es el código de curso (esto debido a que definimos la propiedad value).

Ahora veremos como realizar la registraci n de los datos cargados en el formulario, en la tabla alumnos de la base de datos base1:

pagina2.php

```
<html>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexi n");

        mysqli_query($conexion, "insert into alumnos(nombre,mail,codigocurso) values
                                ('$_POST[nombre]','$_POST[mail]','$_POST[codigocurso]')")
            or die("Problemas en el select" . mysqli_error($conexion));

        mysqli_close($conexion);

        echo "El alumno fue dado de alta.";
    ?>
</body>

</html>
```

localhost / 127.0.0.1 / base1 / alu X +

localhost/phpmyadmin/sql.php?db=base1&table=alumnos&pos=0

phpMyAdmin

Servidor: 127.0.0.1 » Base de datos: base1 » Tabla: alumnos

Examinar Estructura SQL Buscar Insertar Más

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0004 segundos.)

SELECT * FROM `alumnos`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones

codigo	nombre	mail	codigocurso
1	diego	diego@gmail.com	3

Editar Copiar Borrar Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

Consola

codigo	nombre	mail	codigocurso
1	diego	diego@gmail.com	3

Problema resuelto.

Copiar el contenido de este cuadro de texto al VS Code y almacenarlo en la carpeta c:\xampp\htdocs con un nombre con extensión php.
luego abrir el navegador (Chrome, FireFox, IExplorer etc.) y en la barra de direcciones tippear:
<http://localhost/pagina1.php> (o el nombre del archivo que le dió)

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <h1>Alta de Alumnos</h1>
    <form action="pagina2.php" method="post">
        Ingrese nombre:
        <input type="text" name="nombre"><br>
        Ingrese mail:
        <input type="text" name="mail"><br>
        Seleccione el curso:
        <select name="codigocurso">
            <option value="1">PHP</option>
            <option value="2">ASP</option>
            <option value="3">JSP</option>
        </select>
        <br>
        <input type="submit" value="Registrar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

        mysqli_query($conexion, "insert into alumnos(nombre,mail,codigocurso) values
                                ('$_POST[nombre]', '$_POST[mail]', $_POST[codigocurso])")
        or die("Problemas en el select" . mysqli_error($conexion));

        mysqli_close($conexion);

        echo "El alumno fue dado de alta.";
    ?>
</body>

</html>
```

Problema propuesto.

Crear en la base de datos "base1" otra tabla llamada "cursos".

La estructura de esta segunda tabla debe ser:

```
codigo int auto_increment primary_key  
nombrecurso varchar(40)
```

Utilizar el PHPMyAdmin para la creación de esta tabla.

Implementar las dos páginas necesarias para efectuar el alta de cursos. Un formulario para ingresar el nombre del curso y otra página donde se efectuará el insert.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<form action="pagina2.php" method="post">
    Ingrese nombre del curso:
    <input type="text" name="nombrecurso" size="40">
    <br>
    <input type="submit" value="Confirmar">
</form>

</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    mysqli_query($conexion, "insert into cursos(nombrecurso) values ('$_POST[nombrecurso]')") or
        die("Problemas en el select" . mysqli_error($conexion));

    mysqli_close($conexion);

    echo "El curso fue dado de alta.";
    ?>
</body>

</html>
```

Listado (selección de registros de una tabla)

```
<html>
<head>
    <title>Problema</title>
</head>
<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "basel") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo,nombre,mail,codigocurso
                                            from alumnos") or
            die("Problemas en el select:" . mysqli_error($conexion));

        while ($reg = mysqli_fetch_array($registros)) {
            echo "Codigo:" . $reg['codigo'] . "<br>";
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Mail:" . $reg['mail'] . "<br>";
            echo "Curso:";

            switch ($reg['codigocurso']) {
                case 1:
                    echo "PHP";
                    break;
                case 2:
                    echo "ASP";
                    break;
                case 3:
                    echo "JSP";
                    break;
            }
            echo "<br>";
            echo "<hr>";
        }

        mysqli_close($conexion);
    ?>
</body>
</html>
```

Problema resuelto.

Copiar el contenido de este cuadro de texto al VS Code y almacenarlo en la carpeta c:\xampp\htdocs con un nombre con extensión php.

Luego abrir el navegador (Chrome, FireFox, IExplorer etc.) y en la barra de direcciones tipear:

<http://localhost/pagina1.php> (o el nombre del archivo que le dió)

página1.php

```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>

    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
      die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select codigo,nombre, mail, codigocurso
                                              from alumnos") or
      die("Problemas en el select:" . mysqli_error($conexion));

    while ($reg = mysqli_fetch_array($registros)) {
      echo "Codigo:" . $reg['codigo'] . "<br>";
      echo "Nombre:" . $reg['nombre'] . "<br>";
      echo "Mail:" . $reg['mail'] . "<br>";
      echo "Curso:";

      switch ($reg['codigocurso']) {
        case 1:
          echo "PHP";
          break;
        case 2:
          echo "ASP";
          break;
        case 3:
          echo "JSP";
          break;
      }
      echo "<br>";
      echo "<hr>";
    }

    mysqli_close($conexion);
    ?>

  </body>

</html>
```

Problema propuesto.

Confeccionar un programa que recupere los datos de la tabla cursos de la base de datos base1.
Mostrar el código de curso y su nombre.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
die("Problemas en el select:" . mysqli_error($conexion));

while ($reg = mysqli_fetch_array($registros)) {
    echo "Codigo:" . $reg['codigo'] . "<br>";
    echo "Nombre del curso:" . $reg['nombrecurso'] . "<br>";
    echo "<hr>";
}

mysqli_close($conexion);
?>

</body>

</html>
```

Consulta (selección de registros de una tabla)

pagina1.html

```
● ● ●

<html>

  <head>
    <title>Problema</title>
  </head>

  <body>
    <form action="pagina2.php" method="post">
      Ingrese el mail del alumno a consultar:
      <input type="text" name="mail">
      <br>
      <input type="submit" value="buscar">
    </form>
  </body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "basel") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo,nombre,codigocurso
                                            from alumnos where mail='$_POST[mail]'"') or
            die("Problemas en el select:" . mysqli_error($conexion));

        if ($reg = mysqli_fetch_array($registros)) {
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Curso:";

            switch ($reg['codigocurso']) {
                case 1:
                    echo "PHP";
                    break;
                case 2:
                    echo "ASP";
                    break;
                case 3:
                    echo "JSP";
                    break;
            }
        } else {
            echo "No existe un alumno con ese mail.";
        }
        mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema resuelto.

Copiar el contenido de este cuadro de texto al VS Code y almacenarlo en la carpeta c:\xampp\htdocs con un nombre con extensión php.

Luego abrir el navegador (Chrome, FireFox, IExplorer etc.) y en la barra de direcciones tipear:
http://localhost/pagina1.php (o el nombre del archivo que le dió)



The screenshot shows a dark-themed code editor window with three circular window controls at the top. The code inside the editor is as follows:

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno a consultar:
        <input type="text" name="mail">
        <br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo,nombre,codigocurso
                                            from alumnos where mail='$_POST[mail]'" ) or
            die("Problemas en el select:" . mysqli_error($conexion));

        if ($reg = mysqli_fetch_array($registros)) {
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Curso:" ;
            switch ($reg['codigocurso']) {
                case 1:
                    echo "PHP";
                    break;
                case 2:
                    echo "ASP";
                    break;
                case 3:
                    echo "JSP";
                    break;
            }
        } else {
            echo "No existe un alumno con ese mail." ;
        }
        mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que permita ingresar el nombre de un alumno en un formulario, luego mostrar los datos del mismo (tener en cuenta que puede haber más de un alumno con el mismo nombre)

pagina1.php

```
● ● ●  
  
<html>  
  
<head>  
    <title>Problema</title>  
</head>  
  
<body>  
    <form action="pagina2.php" method="post">  
        Ingrese nombre del/los alumno/s:  
        <input type="text" name="nombre">  
        <br>  
        <input type="submit" value="buscar">  
    </form>  
</body>  
  
</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo,nombre, codigocurso
                                            from alumnos where nombre='$_POST[nombre]'" ) or
            die("Problemas en el select:" . mysqli_error($conexion));

        $cant = 0;
        while ($reg = mysqli_fetch_array($registros)) {
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Curso:" ;
            switch ($reg['codigocurso']) {
                case 1:
                    echo "PHP";
                    break;
                case 2:
                    echo "ASP";
                    break;
                case 3:
                    echo "JSP";
                    break;
            }
            $cant++;
            echo "<hr>";
        }

        if ($cant == 0) {
            echo "No existen alumnos con dicho nombre";
        }

        mysqli_close($conexion);
    ?>
</body>

</html>
```

DELETE (Baja de un registro en una tabla)

`pagina1.html`

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno a borrar:
        <input type="text" name="mail">
        <br>
        <input type="submit" value="buscar y borrar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select codigo from alumnos
                                                where mail='$_POST[mail]'" ) or
        die("Problemas en el select:" . mysqli_error($conexion));
    if ($reg = mysqli_fetch_array($registros)) {
        mysqli_query($conexion, "delete from alumnos where mail='$_POST[mail]'" ) or
            die("Problemas en el select:" . mysqli_error($conexion));
        echo "Se efectuó el borrado del alumno con dicho mail.";
    } else {
        echo "No existe un alumno con ese mail.";
    }
    mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que permita ingresar el nombre de un curso por teclado y posteriormente efectúe el borrado de dicho registro en la tabla cursos. Mostrar un mensaje si no existe el curso.

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre del curso a borrar:
        <input type="text" name="nombrecurso">
        <br>
        <input type="submit" value="Borrar">
    </form>
</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo from cursos
                                                where nombrerecurso='$_POST[nombrerecurso]'") or
            die("Problemas en el select:" . mysqli_error($conexion));
        if ($reg = mysqli_fetch_array($registros)) {
            mysqli_query($conexion, "delete from cursos where nombrerecurso='$_POST[nombrerecurso]'") or
                die("Problemas en el select:" . mysqli_error($conexion));
            echo "Se efectuó el borrado del curso ingresado.";
        } else {
            echo "No existe un curso con dicho nombre.";
        }
        mysqli_close($conexion);
    ?>
</body>

</html>
```

DELETE (Baja de todos los registros de una tabla)

```
● ● ●

<html>

  <head>
    <title>Problema</title>
  </head>

  <body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
      die("Problemas con la conexión");

    mysqli_query($conexion, "delete from alumnos") or
      die("Problemas en el select:" . mysqli_error($conexion));

    echo "Se efectuó el borrado de todos los alumnos.';

    mysqli_close($conexion);
    ?>
  </body>

</html>
```

Problema propuesto.

Efectuar el borrado de todos los registros de la tabla cursos.



```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        mysqli_query($conexion, "delete from cursos") or
            die("Problemas en el select:" . mysqli_error($conexion));
        echo "Se efectuó el borrado de todos los cursos.";
        mysqli_close($conexion);
    ?>
</body>

</html>
```

UPDATE (Modificación de un registro de una tabla)

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno:
        <input type="text" name="mail"><br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php

$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select * from alumnos
                                         where mail='$_POST[mail]'" ) or
die("Problemas en el select:" . mysqli_error($conexion));
if ($reg = mysqli_fetch_array($registros)) {
    ?>

    <form action="pagina3.php" method="post">
        Ingrese nuevo mail:
        <input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
        <br>
        <input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
        <input type="submit" value="Modificar">
    </form>

    <?php
} else
    echo "No existe alumno con dicho mail";
?
</body>

</html>
```

pagina3.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die( "Problemas con la conexión");

        mysqli_query($conexion, "update alumnos
                                set mail='$_POST[mailnuevo]'
                                where mail='$_POST[mailviejo]'" ) or
            die( "Problemas en el select:" . mysqli_error($conexion));
        echo "El mail fue modificado con exito";
    ?>
</body>

</html>
```

Problema propuesto.

Efectuar la modificación del nombre del curso de la tabla "cursos". Para la búsqueda ingresar el código de curso.

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el codigo del curso a modificar:
        <input type="text" name="codigo"><br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select * from cursos
                                         where codigo=$_POST[codigo]") or
die("Problemas en el select:" . mysqli_error($conexion));
if ($reg = mysqli_fetch_array($registros)) {
    ?>

    <form action="pagina3.php" method="post">
        Ingrese el nuevo nombre del curso:
        <input type="text" name="nombrecurso" value="<?php echo $reg['nombrecurso'] ?>">
        <input type="hidden" name="codigo" value="<?php echo $reg['codigo'] ?>">
        <br>
        <input type="submit" value="Modificar">
    </form>

<?php
} else
    echo "No existe un curso con dicho codigo";
?>
</body>

</html>
```

pagina3.php

```
● ● ●

<html>

    <head>
        <title>Problema</title>
    </head>

    <body>
        <?php
            $conexion = mysqli_connect("localhost", "root", "", "base1") or
                die("Problemas con la conexión");

            mysqli_query($conexion, "update cursos
                                set nombrecurso='$_POST[nombrecurso]'
                                where codigo=$_POST[codigo]" ) or
                die("Problemas en el select:" . mysqli_error($conexion));
            echo "Se modificó el nombre del curso";
        ?>
    </body>

</html>
```

INSERT (y consulta de otra tabla)

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre:
        <input type="text" name="nombre"><br>
        Ingrese mail:
        <input type="text" name="mail"><br>
        Seleccione el curso:
        <select name="codigocurso">
            <?php
            $conexion = mysqli_connect("localhost", "root", "", "base1") or
                die("Problemas con la conexión");

            $registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
                die("Problemas en el select:" . mysqli_error($conexion));
            while ($reg = mysqli_fetch_array($registros)) {
                echo "<option value=\"$reg[codigo]\">$reg[nombrecurso]</option>";
            }
            ?>
        </select>
        <br>
        <input type="submit" value="Registrar">
    </form>
</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        mysqli_query($conexion, "insert into alumnos(nombre,mail,codigocurso)
                                    values ('$_POST[nombre]','$_POST[mail]','$_POST[codigocurso]')") or
            die("Problemas en el select" . mysqli_error($conexion));
        mysqli_close($conexion);
        echo "El alumno fue dado de alta.";
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar el alta de la tabla alumnos
empleando controles de tipo "radio" para la
selección del curso.

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre:
        <input type="text" name="nombre"><br>
        Ingrese mail:
        <input type="text" name="mail"><br>
        Seleccione el curso:<br>
        <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
            die("Problemas en el select:" . mysql_error());
        while ($reg = mysqli_fetch_array($registros)) {
            echo "<input type=\"radio\" name=\"radiol\" value=\"$reg[codigo]\">" .
        $reg['nombrecurso'] . "<br>";
        }
        ?>
        <input type="submit" value="Registrar">
    </form>
</body>

</html>
```

pagina2.php

```
● ● ●

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"insert into alumnos(nombre,mail,codigocurso) values
('$_POST[nombre]','$_POST[mail]','$_POST[radio1]')") or
    die("Problemas en el select".mysqli_error($conexion));

echo "El alumno fue dado de alta.";
?>
</body>
</html>
```

Listado (selección de registros de varias tablas - INNER JOIN)

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select alu.codigo as codigo,
                                                nombre,
                                                mail,
                                                codigocurso,
                                                nombrecurso
                                                from alumnos as alu
                                                inner join cursos as cur on cur.codigo=alu.codigocurso") or
        die("Problemas en el select:" . mysqli_error($conexion));

    while ($reg = mysqli_fetch_array($registros)) {
        echo "Codigo:" . $reg['codigo'] . "<br>";
        echo "Nombre:" . $reg['nombre'] . "<br>";
        echo "Mail:" . $reg['mail'] . "<br>";
        echo "Curso:" . $reg['nombrecurso'] . "<br>";
        echo "<hr>";
    }
    mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que permita ingresar el código de un alumno y nos muestre su nombre, mail y nombre del curso en el cual está inscripto.
Hacer un formulario donde se ingrese el código de alumno y otra página donde se muestren los datos respectivos. Mostrar un mensaje si no existe el código de alumno ingresado.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el codigo del alumno:
        <input type="text" name="codigo">
        <br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select nombre,
                                                mail,
                                                nombrecurso
                                            from alumnos as alu
                                            inner join cursos as cur on cur.codigo=alu.codigocurso
                                            where alu.codigo=$_REQUEST[codigo]) or
            die("Problemas en el select:" . mysqli_error($conexion));

        if ($reg = mysqli_fetch_array($registros)) {
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Mail:" . $reg['mail'] . "<br>";
            echo "Curso:" . $reg['nombrecurso'] . "<br>";
        } else {
            echo "No existe un alumno con ese código.";
        }
        mysqli_close($conexion);
    ?>
</body>

</html>
```

Función count de SQL.

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select count(*) as cantidad from alumnos") or
            die("Problemas en el select:" . mysqli_error($conexion));
        $reg = mysqli_fetch_array($registros);
        echo "La cantidad de alumnos inscriptos son :" . $reg['cantidad'];
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que muestre por pantalla los nombres de todos los cursos y al final la cantidad total de cursos.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select nombrecurso from cursos") or
die("Problemas en el select:" . mysqli_error($conexion));

while ($reg = mysqli_fetch_array($registros)) {
    echo "Nombre del curso:" . $reg['nombrecurso'] . "<br>";
    echo "<hr>";
}

$registros = mysqli_query($conexion, "select count(*) as cantidad from cursos") or
die("Problemas en el select:" . mysqli_error($conexion));

$reg = mysqli_fetch_array($registros);
echo "La cantidad de cursos son :" . $reg['cantidad'];

mysqli_close($conexion);
?>

</body>

</html>
```

UPDATE (Modificación de un registro trabajando con dos tablas)

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno:
        <input type="text" name="mail"><br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select * from alumnos
                                                where mail='$_POST[mail]'" ) or
        die("Problemas en el select:" . mysqli_error($conexion));
        if ($regalu = mysqli_fetch_array($registros)) {
            ?>
            <form action="pagina3.php" method="post">
                <input type="hidden" name="mailviejo" value="<?php echo $regalu['mail'] ?>">
                <select name="codigocurso">
                    <?php
                        $registros = mysqli_query($conexion, "select * from cursos") or
                        die("Problemas en el select:" . mysqli_error($conexion));
                        while ($reg = mysqli_fetch_array($registros)) {
                            if ($regalu['codigocurso'] == $reg['codigo'])
                                echo "<option value=\"$reg[codigo]\" selected>$reg[nombrecurso]</option>";
                            else
                                echo "<option value=\"$reg[codigo]\">$reg[nombrecurso]</option>";
                        }
                    ?>
                </select>
                <br>
                <input type="submit" value="Modificar">
            </form>
        <?php
    } else
        echo "No existe alumno con dicho mail";
    ?>
</body>

</html>
```

pagina3.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "update alumnos
                                                set codigocurso=$_POST[codigocurso]
                                                where mail='$_POST[mailviejo]'" ) or
            die("Problemas en el select:" . mysqli_error($conexion));
        echo "El curso fue modificado con exito";
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar la modificación del mail, nombre y curso de la tabla "alumnos". Ingresar por teclado el código de alumno para su búsqueda.

pagina1.php

```
● ● ●  
<html>  
  
<head>  
    <title>Problema</title>  
</head>  
  
<body>  
    <form action="pagina2.php" method="post">  
        Ingrese el codigo del alumno:  
        <input type="text" name="codigo"><br>  
        <input type="submit" value="buscar">  
    </form>  
</body>  
  
</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select * from alumnos
                                                where codigo=$_POST[codigo]") or
            die("Problemas en el select:" . mysqli_error($conexion));
        if ($regalu = mysqli_fetch_array($registros)) {
            ?>
            <form action="pagina3.php" method="post">
                Nombre: <input type="text" name="nombre" value="<?php echo $regalu['nombre'] ?>"><br>
                Mail: <input type="text" name="mail" value="<?php echo $regalu['mail'] ?>"><br>
                <input type="hidden" name="codigo" value="<?php echo $regalu['codigo'] ?>">
                <select name="codigocurso">
                    <?php
                        $registros = mysqli_query($conexion, "select * from cursos") or
                            die("Problemas en el select:" . mysqli_error());
                        while ($reg = mysqli_fetch_array($registros)) {
                            if ($regalu['codigocurso'] == $reg['codigo'])
                                echo "<option value=\"$reg[codigo]\" selected>$reg[nombrecurso]</option>";
                            else
                                echo "<option value=\"$reg[codigo]\">$reg[nombrecurso]</option>";
                        }
                    ?>
                </select>
                <br>
                <input type="submit" value="Modificar">
            </form>
        <?php
    } else
        echo "No existe alumno con dicho código";
    ?>
</body>

</html>
```

pagina3.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "update alumnos
                                                set digocurso=$_POST[codigocurso],
                                                nombre='$_POST[nombre]',mail='$_POST[mail]'
                                                where codigo=$_POST[codigo]") or
            die("Problemas en el select:" . mysqli_error($conexion));
        echo "El curso fue modificado con exito";
    ?>
</body>

</html>
```

Cláusula Group By de SQL.

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "basel") or
        die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select count(alu.codigo) as cantidad,
                                                nombrecurso
                                                from alumnos as alu
                                                inner join cursos as cur on cur.codigo=alu.codigocurso
                                                group by alu.codigocurso") or
        die("Problemas en el select:" . mysqli_error($conexion));
    while ($reg = mysqli_fetch_array($registros)) {
        echo "Nombre del curso:" . $reg['nombrecurso'] . "<br>";
        echo "Cantidad de inscriptos:" . $reg['cantidad'] . "<br>";
        echo "<hr>";
    }
    mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que muestre el nombre del curso, la cantidad de inscriptos y todos los inscriptos a dicho curso. Repetir esto con todos los cursos.

Es decir, en la página debe aparecer algo parecido a:

Nombre del curso:PHP

Cantidad de inscriptos:3

Nombres: Martinez Luis - Rodriguez Pablo - Gonzalez Ana

Nombre del curso:JSP

Cantidad de inscriptos:2

Nombres: Hernandez Hector - Roca Marta

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select count(alu.codigo) as cantidad,
                                                nombrecurso,
                                                codigocurso
                                         from alumnos as alu
                                         inner join cursos as cur on cur.codigo=alu.codigocurso
                                         group by alu.codigocurso") or
        die("Problemas en el select:" . mysqli_error($conexion));
    while ($reg = mysqli_fetch_array($registros)) {
        echo "Nombre del curso:" . $reg['nombrecurso'] . "<br>";
        echo "Cantidad de inscriptos:" . $reg['cantidad'] . "<br>";
        $registros2 = mysqli_query($conexion, "select nombre
                                                from alumnos
                                                where codigocurso=$reg[codigocurso]") or
            die("Problemas en el select:" . mysqli_error($conexion));
        while ($reg2 = mysqli_fetch_array($registros2)) {
            echo $reg2['nombre'] . "-";
        }
        echo "<hr>";
    }
    mysqli_close($conexion);
    ?>
</body>

</html>
```

Parámetros en un hipervínculo.

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <A href="pagina2.php?tabla=2">Tabla del 2</A> <br>
    <A href="pagina2.php?tabla=3">Tabla del 3</A> <br>
    <A href="pagina2.php?tabla=5">Tabla del 5</A>
</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        echo "Listado de la tabla del $_GET[tabla] <br>";
        for ($f = 1; $f <= 10; $f++) {
            $valor = $f * $_GET['tabla'];
            echo $valor . "-";
        }
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que muestre una página con todos los nombres de la tabla "cursos". Los nombres deben aparecer como hipervínculos a otra página que deberá mostrar todos los alumnos inscriptos a dicho curso. Como parámetro en el hipervínculo, pasar el código de curso.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

    <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
        die("Problemas en el select:" . mysqli_error($conexion));

    while ($reg = mysqli_fetch_array($registros)) {
        echo "Codigo:" . $reg['codigo'] . "<br>";
        echo "Nombre del curso:<a href=\"pagina2.php?codigo=$reg[codigo]\\">" . $reg['nombrecurso'] . "</a><br>";
        echo "<hr>";
    }

    mysqli_close($conexion);
    ?>

</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "basel") or
    die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select nombre from alumnos where codigocurso=$_GET[codigo]") or
    die("Problemas en el select:" . mysqli_error($conexion));

echo "Alumnos inscriptos a dicho curso<br>";
while ($reg = mysqli_fetch_array($registros)) {
    echo $reg['nombre'] . " - ";
}

mysqli_close($conexion);
?>

</body>

</html>
```

Paginación de registros.

pagina1.php

```
<?php
if (isset($_REQUEST['pos']))
    $inicio = $_REQUEST['pos'];
else
    $inicio = 0;
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select alu.codigo as
                                    codigo,
                                    nombre,
                                    mail,
                                    codigocurso,
                                    nombrecurso
                           from alumnos as alu
                           inner join cursos as cur on cur.codigo=alu.codigocurso
                           limit $inicio,2") or
die("Problemas en el select:" . mysqli_error($conexion));
$impresos = 0;
while ($reg = mysqli_fetch_array($registros)) {
    $impresos++;
    echo "Codigo:" . $reg['codigo'] . "<br>";
    echo "Nombre:" . $reg['nombre'] . "<br>";
    echo "Mail:" . $reg['mail'] . "<br>";
    echo "Curso:" . $reg['nombrecurso'] . "<br>";
    echo "<hr>";
}
if ($inicio == 0)
    echo "anteriores ";
else {
    $anterior = $inicio - 2;
    echo "<a href=\"pagina1.php?pos=$anterior\">Anteriores </a>";
}
if ($impresos == 2) {
    $proximo = $inicio + 2;
    echo "<a href=\"pagina1.php?pos=$proximo\">Siguientes</a>";
} else
    echo "siguientes";
mysqli_close($conexion);
?>

</body>
</html>
```

Problema propuesto.

Confeccionar un programa que muestre los registros de la tabla "cursos" con páginas de 3 registros.

pagina1.php

```
<?php
if (isset($_GET['pos'])) {
    $inicio = $_GET['pos'];
} else {
    $inicio = 0;
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$conexion = mysqli_connect("localhost", "root", "", "basel") or
    die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select codigo,
                                         nombrecurso
                                         from cursos
                                         limit $inicio,3") or
    die("Problemas en el select:" . mysqli_error($conexion));
$impresos = 0;
while ($reg = mysqli_fetch_array($registros)) {
    $impresos++;
    echo "Codigo:" . $reg['codigo'] . "<br>";
    echo "Nombre:" . $reg['nombrecurso'] . "<br>";
    echo "<hr>";
}
mysqli_close($conexion);
if ($inicio == 0)
    echo "anteriores ";
else {
    $anterior = $inicio - 3;
    echo "<a href=\"pagina1.php?pos=$anterior\">Anteriores </a>";
}
if ($impresos == 3) {
    $proxima = $inicio + 3;
    echo "<a href=\"pagina1.php?pos=$proxima\">Siguientes</a>";
} else
    echo "siguientes";
?>

</body>

</html>
```

Subir un archivo al servidor (Upload)

pagina1.html

```
● ● ●

<html>

  <head>
    <title>Problema</title>
  </head>

  <body>
    <form action="pagina2.php" method="post" enctype="multipart/form-data">
      Seleccione el archivo:
      <input type="file" name="foto"><br>
      <input type="submit" value="Enviar">
    </form>
  </body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    copy($_FILES['foto']['tmp_name'], $_FILES['foto']['name']);
    echo "La foto se registro en el servidor.<br>";
    $nom = $_FILES['foto']['name'];
    echo "<img src=\"$nom\">";
    ?>
</body>

</html>
```

Cuando se ejecuta esta página, ya está almacenado en el servidor el archivo, en una carpeta temporal. Ahora nos hace falta copiar el mismo a la carpeta donde se encuentra nuestra página (en definitiva nuestro sitio de internet). Para esto llamamos a la función copy:

```
copy($_FILES['foto']['tmp_name'],$_FILES['foto']['name']);
```

La matriz \$_FILES almacena el nombre del archivo guardado en el directorio temporal (\$_FILES['foto']['tmp_name']) y el nombre del archivo original (\$_FILES['foto']['name'])

Por último mostramos en la página el archivo de la imagen que se almacenó en el servidor:

```
$nom=$_FILES['foto']['name'];
echo "<img src=\"$nom\">";
```

Otras cosas interesantes que tiene la matriz \$_FILES:

`$_FILES['foto']['name']` El nombre original del fichero en
la máquina cliente.

`$_FILES['foto']['type']` El tipo de archivo (si el navegador
lo proporciona). Un ejemplo podría ser "image/gif".

`$_FILES['foto']['size']` El tamaño en bytes del fichero recibido.

`$_FILES['foto']['tmp_name']` El nombre del archivo temporal que se
utiliza para almacenar en el servidor el archivo recibido.

Problema propuesto.

Confeccionar un programa que permita hacer multiples upload con una página (por ejemplo que permita seleccionar hasta 3 archivos)

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post" enctype="multipart/form-data">
        Seleccione el archivo 1:
        <input type="file" name="foto1"><br>
        Seleccione el archivo 2:
        <input type="file" name="foto2"><br>
        Seleccione el archivo 2:
        <input type="file" name="foto3"><br>
        <input type="submit" value="Enviar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
if ($_FILES['foto1']['tmp_name'] != "") {
    copy($_FILES['foto1']['tmp_name'], $_FILES['foto1']['name']);
    $nom = $_FILES['foto1']['name'];
    echo "<img src=\"$nom\">";
}
if ($_FILES['foto2']['tmp_name'] != "") {
    copy($_FILES['foto2']['tmp_name'], $_FILES['foto2']['name']);
    $nom = $_FILES['foto2']['name'];
    echo "<img src=\"$nom\">";
}
if ($_FILES['foto3']['tmp_name'] != "") {
    copy($_FILES['foto3']['tmp_name'], $_FILES['foto3']['name']);
    $nom = $_FILES['foto3']['name'];
    echo "<img src=\"$nom\">";
}

?>

</body>

</html>
```

Creación y lectura de una cookie

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body <?php if (isset($_COOKIE['color'])) echo " style=\"$COOKIE[color]\" ?>>
    <form action="pagina2.php" method="post">
        Seleccione de que color desea que sea la página de ahora en más:<br>
        <input type="radio" value="rojo" name="radio">Rojo<br>
        <input type="radio" value="verde" name="radio">Verde<br>
        <input type="radio" value="azul" name="radio">Azul<br>
        <input type="submit" value="Crear cookie">
    </form>
</body>

</html>
```

pagina2.php

```
<?php
if ($_REQUEST['radio'] == "rojo")
    setcookie("color", "#ff0000", time() + 60 * 60 * 24 * 365, "/");
elseif ($_REQUEST['radio'] == "verde")
    setcookie("color", "#00ff00", time() + 60 * 60 * 24 * 365, "/");
elseif ($_REQUEST['radio'] == "azul")
    setcookie("color", "#0000ff", time() + 60 * 60 * 24 * 365, "/");
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    Se creó la cookie.
    <br>
    <a href="pagina1.php">Ir a la otra página</a>
</body>

</html>
```

La llamada a la función setcookie debe hacerse antes de imprimir cualquier etiqueta HTML, de lo contrario no funcionará.

Como podemos observar, la creación de la cookie se hace llamando a la función setcookie:

```
<?php
if ($_REQUEST['radio'] == "rojo")
    setcookie("color", "#ff0000", time() + 60 * 60 * 24 * 365, "/");
elseif ($_REQUEST['radio'] == "verde")
    setcookie("color", "#00ff00", time() + 60 * 60 * 24 * 365, "/");
elseif ($_REQUEST['radio'] == "azul")
    setcookie("color", "#0000ff", time() + 60 * 60 * 24 * 365, "/");
?>
```

El nombre de la cookie se llama "color" y el valor que almacenamos depende de que control de tipo radio esté seleccionado en la página anterior. La fecha de expiración de la cookie la calculamos fácilmente llamando a la función time() que nos retorna la fecha actual en segundos y le sumamos el producto $60*60*24*365$ (60 segundos * 60 minutos * 24 horas * 365 días) es decir que la cookie existirá en la máquina del visitante hasta el año próximo.

Cuando indicamos como directorio la sintaxis "/" significa que la cookie se crea a nivel del sitio y con cuquier petición a dicho sitio, el navegador enviará la cookie al servidor.

Por último dispusimos en esta página un hipervínculo a la página anterior, para ver que, de ahora en más, cada vez que ejecutemos la pagina1.php, el color de fondo de la misma dependerá del valor de la cookie registrada.

Problema propuesto.

Crear un formulario que solicite la carga del nombre de usuario. Cuando se presione un botón crear una cookie para dicho usuario. Luego cada vez que ingrese al formulario mostrar el último nombre de usuario ingresado.

pagina1.php

```
● ● ●

<html>

<head>
    <title>Formulario de entrada del dato</title>
</head>

<body>

    <form method="post" action="pagina2.php">
        Ingrese su nombre:
        <input type="text" name="nombre" value=<?php if (isset($_COOKIE['usuario'])) echo $_COOKIE['usuario'] ?>>
        <br>
        <input type="submit" value="confirmar">
    </form>

</body>

</html>
```

pagina2.php

```
<?php
setcookie("usuario", $_REQUEST['nombre'], time() + 60 * 60 * 24 * 365, "/");
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>

    Se creó la cookie.
    <br>
    <a href="pagina1.php">Ir a la otra página</a>

</body>

</html>
```

INSERT (Alta de registros en una tabla)

Luego de crear una base de datos y sus tablas (Vamos a trabajar con la base de datos ya creada: base1, que contiene la tabla alumnos), veremos como agregar registros.

Para añadir datos en la tabla empleamos el comando SQL llamado insert.

Necesitamos dos páginas para este proceso, una será el formulario de carga de datos y la siguiente será la que efectúe la inserción en la tabla.

Formulario de carga de datos:

```
<html>
  <head>
    <title>Problema</title>
  </head>

  <body>
    <h1>Alta de Alumnos</h1>
    <form action="pagina2.php" method="post">
      Ingrese nombre:
      <input type="text" name="nombre"><br>
      Ingrese mail:
      <input type="text" name="mail"><br>
      Seleccione el curso:
      <select name="codigocurso">
        <option value="1">PHP</option>
        <option value="2">ASP</option>
        <option value="3">JSP</option>
      </select>
      <br>
      <input type="submit" value="Registrar">
    </form>
  </body>

</html>
```

Borrado de una cookie

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese su mail:
        <input type="text" name="mailusuario" value=<?php if (isset($_COOKIE['mail'])) echo $_COOKIE['mail']; ?>>
        <br>
        <input type="radio" name="opcion" value="recordar">
        Recordar en esta computadora el mail ingresado.
        <br>
        <input type="radio" name="opcion" value="norecordar">
        No recordar.
        <br>
        <input type="submit" value="confirmar">
    </form>
</body>

</html>
```

pagina2.php

```
● ● ●

<?php
if ($_POST['opcion'] == "recordar")
    setcookie("mail", $_POST['mailusuario'], time() + (60 * 60 * 24 * 365), "/");
elseif ($_POST['opcion'] == "norecordar")
    setcookie("mail", "", time() - 1000, "/");
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    if ($_POST['opcion'] == "recordar")
        echo "cookie creada";
    elseif ($_POST['opcion'] == "norecordar")
        echo "cookie eliminada";
    ?>
    <br>
    <a href="pagina1.php">Ir a la otra página</a>
</body>

</html>
```

Problema propuesto.

Confeccionar una página que simule ser la de un periódico. La misma debe permitir configurar qué tipo de titular deseamos que aparezca al visitarla, pudiendo ser:

Noticia política, Noticia económica o Noticia deportiva.

Mediante tres objetos de tipo radio, permitir seleccionar qué titular debe mostrar el periódico. Almacenar en una cookie el tipo de titular que desea ver el cliente. La primera vez que visita el sitio deben aparecer los tres titulares.

Disponer un hipervínculo a una tercer página que borre la cookie creada.

pagina1.php

```
● ● ●

<html>
<head>
    <title>Problema</title>
</head>

<body>

<?php
if (isset($_COOKIE['noticias']))
    echo $_COOKIE['noticias'] . "<br>";
else {
    ?>
    <form method="post" action="pagina2.php">
        Configuracion del titular del sitio:<br>
        <input type="radio" name="radio1" value="Noticias Politicas">Noticias Políticas.<br>
        <input type="radio" name="radio1" value="Noticias Economicas">Noticias Económicas.<br>
        <input type="radio" name="radio1" value="Noticias Deportivas">Noticias Deportivas.<br>
        <input type="submit" value="Confirmar">
    </form>
    <?php
}
?>
    <a href="pagina3.php">Borrar cookies</a>
</body>

</html>
```

pagina2.php

```
● ● ●

<?php
setcookie("noticias", $_POST['radiol'], time() + (60 * 60 * 24 * 365), "/");
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <h2>Se configuró correctamente</h2>
    <a href="pagina1.php">Ir a la otra página</a>
</body>

</html>
```

pagina3.php

```
<?php
setcookie("noticias", "", time() - 100, "/");
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <h2>Se borró correctamente</h2>
    <a href="pagina1.php">Ir a la otra página</a>
</body>

</html>
```

Cookie de sesión

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        if (isset($_COOKIE['usu']))
            echo "Cookie de sesión creada. Su valor es $_COOKIE[usu]";
        else
            echo "No existe cookie de sesión";
    ?>
    <br>
    <a href="pagina2.php">Crear cookie de sesión</a>
</body>

</html>
```

pagina2.php

```
<?php
setcookie("usu", "diego", 0);
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    Cookie de sesión creada.<br>
    <a href="pagina1.php">Retornar a la página anterior.</a>
</body>

</html>
```

Variables de sesión (\$_SESSION)

pagina1.html

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre de usuario:
        <input type="text" name="campousuario"><br>
        Ingrese clave:
        <input type="password" name="campoclave"><br>
        <input type="submit" value="confirmar">
    </form>
</body>

</html>
```

pagina2.php

```
<?php
session_start();
$_SESSION['usuario'] = $_POST['campousuario'];
$_SESSION['clave'] = $_POST['campoclave'];
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    Se almacenaron dos variables de sesión.<br><br>
    <a href="pagina3.php">Ir a la tercer página donde se recuperarán
        las variables de sesión</a>
</body>

</html>
```

Problema propuesto.

Confeccionar un formulario que solicite ingresar el mail de un alumno. Si el mail existe en la tabla alumnos, rescatar su nombre y almacenarlo en una variable de sesión. Además disponer un hipervínculo a una tercera página que verifique si existe la variable de sesión y de la bienvenida al alumno, en caso contrario mostrar un mensaje indicando que no puede visitar esta página (para saber si una variable de sesión está definida llamamos a la función isset)

```
if (isset($_SESSION['nombre']))
```

```
....
```

pagina1.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno a consultar:
        <input type="text" name="mail">
        <br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>
```

pagina2.php

```
● ● ●

<?php
session_start();
$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

registros = mysqli_query($conexion, "select codigo, nombre, codigocurso
from alumnos where mail='$_POST[mail]'") or
die("Problemas en el select:" . mysqli_error($conexion));

if ($reg = mysqli_fetch_array($registros)) {
    $_SESSION['usuario'] = $reg['nombre'];
}

?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <a href="pagina3.php">Ingresar a pagina principal</a>;
</body>

</html>
```

pagina3.php

```
<?php
session_start();
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    if (isset($_SESSION['usuario'])) {
        echo "Bienvenido " . $_SESSION['usuario'];
    } else {
        echo "No tiene permitido visitar esta página.";
    }

    ?>
</body>

</html>
```

Incluir un archivo externo (require_once)

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        require_once( "pagina2.php" );
        cabeceraPagina("Titulo principal de la página");
        echo "<br><br><center>Este es el cuerpo de la página<br><br></center>";
        piePagina("Pie de la página");
    ?>
</body>

</html>
```

pagina1.php

```
<?php
function cabeceraPagina($tit)
{
    echo "<h1 style=\"width:100%;background:#ffff00;text-align:center\">$tit</h1>";
}

function piePagina($tit)
{
    echo "<p style=\"width:100%;background:#cccccc;text-align:center\">$tit</p>";
}
?>
```

Problema propuesto.

Confeccionar una librería que contenga una función llamada retornarConexion, la misma debe llamar a las funciones mysqli_connect y retornar la variable que generó la función mysqli_connect.

Tener en cuenta que a esta función la deben implementar en el archivo "pagina2.php".

En el archivo "pagina1.php" incluir la librería que contiene la función retornarConexion. Luego imprimir todos los registros de la tabla alumnos.

pagina1.php

```
○ ○ ○

<html>

<head>
    <title>Problema</title>
</head>

<body>
<?php

require_once("pagina2.php");
$conexion = retornarConexion();

registros = mysqli_query($conexion, "select alu.codigo as codigo,
                                              nombre,
                                              mail,
                                              codigocurso,
                                              nombrecurso
                                         from alumnos as alu
                                         inner join cursos as cur on cur.codigo=alu.codigocurso") or
die("Problemas en el select:" . mysqli_error($conexion));
while ($reg = mysqli_fetch_array($registros)) {
    echo "Codigo:" . $reg['codigo'] . "<br>";
    echo "Nombre:" . $reg['nombre'] . "<br>";
    echo "Mail:" . $reg['mail'] . "<br>";
    echo "Curso:" . $reg['nombrecurso'] . "<br>";
    echo "<hr>";
}
mysqli_close($conexion);
?>
</body>

</html>
```

pagina2.php

```
<?php
function retornarConexion()
{
    $conexion=mysqli_connect("localhost","root","","base1") or
        die("Problemas con la conexión");
    return $conexion;
}
?>
```

Redireccionar a otra página (header)

pagina1.html



The screenshot shows a dark-themed code editor window with three circular status indicators at the top left. The code itself is written in HTML and includes a form for entering a URL and a redireccionar button.

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese dirección de sitio web (ej www.google.com):
        <input type="text" name="direccion" size="30"><br>
        <input type="submit" value="Redireccionar">
    </form>
</body>

</html>
```

pagina2.php



A screenshot of a terminal window with a black background and three colored window control buttons (red, yellow, green) at the top. The terminal displays the following PHP code:

```
<?php
header("Location: http://$_POST[direccion]");
?>
```

Problema propuesto.

Confeccionar un programa que solicite el ingreso de una clave en un formulario. La segunda página debe verificar si ingresó el string "z80" y mostrar un mensaje de bienvenida, en caso contrario, esta página debe redireccionar a la primera página nuevamente.

Cuando tenemos que redireccionar a una página que está en el mismo sitio, sólo es necesario disponer su nombre:

```
<?php  
if ($_REQUEST['direccion']<>"z80")  
    header("Location: pagina1.php");  
?>
```

Otra cosa interesante que podemos hacer es pasar como parámetro en la segunda página un código de error:

```
<?php  
if ($_REQUEST['direccion']<>"z80")  
    header("Location: pagina1.php?error=1");  
?>
```

Y luego en la primera página mostrar un mensaje de error si es que la página recibe este parámetro:

```
<?php  
if (isset($_REQUEST['error']))  
    echo "Ingreso clave incorrecta";  
?>
```

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php {
        if (isset($_POST['error']))
            echo "Ingresó clave incorrecta<br>";
    }
    ?>
    <form method="post" action="pagina2.php">
        Ingrese el clave:
        <input type="password" name="clave">
        <br>
        <input type="submit" value="confirmar">
    </form>

</body>

</html>
```

pagina2.php

```
<?php
if ($_POST['clave'] != "z80") {
    header("Location: pagina1.php?error=1");
}

?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <h2>Bienvenido</h2>
</body>

</html>
```

Creación de imágenes dinámicas desde PHP

pagina1.php

```
<?php
$ancho=100;
$alto=30;
$imagen=imageCreate($ancho,$alto);
$amarillo=ImageColorAllocate($imagen,255,255,0);
ImageFill($imagen,0,0,$amarillo);
$rojo=ImageColorAllocate($imagen,255,0,0);
$valoraleatorio=rand(100000,999999);
ImageString($imagen,5,25,5,$valoraleatorio,$rojo);
for($c=0;$c<=5;$c++)
{
    $x1=rand(0,$ancho);
    $y1=rand(0,$alto);
    $x2=rand(0,$ancho);
    $y2=rand(0,$alto);
    ImageLine($imagen,$x1,$y1,$x2,$y2,$rojo);
}
Header ("Content-type: image/jpeg");
ImageJPEG ($imagen);
ImageDestroy($imagen);
?>
```

Acotaciones

Si utiliza el entorno de XAMPP la extensión 'gd2' que permite gráficos dinámicas se encuentra activa por defecto, pero si utiliza el entorno del WAMPServer debe activarlo.

Para el correcto funcionamiento de las funciones gráficas con WAMPSEVER debemos activar las extensiones de PHP que nos permiten esta funcionalidad, para esto debemos activar desde el WampServer la extensión php_gd2:



Problema propuesto.

Confeccionar una imagen dinámica que represente un botón. Utilizar las funciones vistas para imprimir el texto del botón y para llenar regiones del gráfico, emplear la función imageFilledRectange. ImageFilledRectangle crea un rectángulo lleno con color "col" en la imagen "imagen", comenzando con la coordenada superior izquierda (x1, y1) y finalizando en la coordenada inferior derecha (x2, y2).

imagefilledrectangle (imagen,x1,y1,x2,y2,col)

Recordar que debe ser PHP puro, por lo que deberá borrar todo el código HTML que se encuentra fuera de las marcas:

<?php ?>

```
<?php
$ancho=100;
$alto=30;
$imagen=imageCreate($ancho,$alto);
$grisoscuro=ImageColorAllocate($imagen,100,100,100);
ImageFill($imagen,0,0,$grisoscuro);
$grisclaro=ImageColorAllocate($imagen,200,200,200);
ImageFilledRectangle($imagen,5,5,95,25,$grisclaro);
$rojo=ImageColorAllocate($imagen,200,0,0);
ImageString($imagen,10,10,5,"Boton 1",$rojo);
Header ("Content-type: image/jpeg");
ImageJPEG ($imagen);
ImageDestroy($imagen);
?>
```

Agregar imágenes dinámicas en un archivo HTML

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina3.php" method="post">
        Dígitos verificadores:
        <br>
        Ingrese valor:
        <input type="text" name="numero">
        <br>
        <input type="submit" value="Verificar">
    </form>
</body>

</html>
```

pagina2.php

```
<?php
$ancho=100;
$alto=30;
$imagen=imageCreate($ancho,$alto);
$amarillo=ImageColorAllocate($imagen,255,255,0);
ImageFill($imagen,0,0,$amarillo);
$rojo=ImageColorAllocate($imagen,255,0,0);
$valoraleatorio=rand(100000,999999);
session_start();
$_SESSION['numeroaleatorio']=$valoraleatorio;
ImageString($imagen,5,25,5,$valoraleatorio,$rojo);
for($c=0;$c<=5;$c++)
{
    $x1=rand(0,$ancho);
    $y1=rand(0,$alto);
    $x2=rand(0,$ancho);
    $y2=rand(0,$alto);
    ImageLine($imagen,$x1,$y1,$x2,$y2,$rojo);
}
Header ("Content-type: image/jpeg");
ImageJPEG ($imagen);
ImageDestroy($imagen);
?>
```

pagina3.php

```
<?php
session_start();
?>
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        if ($_SESSION['numeroaleatorio'] == $_POST['numero'])
            echo "Ingresó el valor correcto";
        else
            echo "Incorrecto";
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un formulario que pida ingresar la dirección de un sitio de internet y mediante un control de tipo select permita dar un puntaje a la misma (cargar los valores de 0 a 5). Luego, en la segunda página, imprimir el nombre del sitio y un gráfico creado en forma dinámica con el puntaje obtenido (hacer un círculo por cada punto).

Un poco de ayuda para este problema:

En el archivo "pagina2.php" debemos disponer la etiqueta HTML img para mostrar la imagen y debemos pasar a este archivo el valor que nos llegó del formulario, es decir, el puntaje seleccionado:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
echo "La direccion: $_REQUEST[direccion] tiene";
?>

</body>
</html>
```

Por último, queda como actividad, dibujar tantos círculos como indica el parámetro puntos:

Recordemos que para recuperar el parámetro puntos lo accedemos mediante el vector asociativo \$_REQUEST:

\$_REQUEST['puntos']

Para dibujar los círculos rellenos debemos emplear la función:

```
imagefilledellipse ( <manejador de imagen>, <centro en x>,
                     <centro en y>, <ancho del elipse>,
                     <alto del elipse>,<manejador de color>)
```

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>

    <form method="post" action="pagina2.php">
        Ingrese sitio web:
        <input type="text" name="direccion">
        <br>
        <select name="puntos">
            <option value="0">0</option>
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
        </select>
        <br>
        <input type="submit" value="confirmar">
    </form>

</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    echo "La direccion: $_POST[direccion] tiene";
    ?>
    
</body>

</html>
```

pagina3.php

```
<?php
$ancho=100;
$alto=30;
$imagen=imageCreate($ancho,$alto);
$amarillo=ImageColorAllocate($imagen,255,255,0);
ImageFill($imagen,0,0,$amarillo);
$rojo=ImageColorAllocate($imagen,255,0,0);
$x=10;
for($c=0;$c<$_REQUEST[ 'puntos' ];$c++)
{
    imagefilledellipse($imagen,$x,15,6,6,$rojo);
    $x=$x+15;
}
Header ("Content-type: image/jpeg");
ImageJPEG ($imagen);
ImageDestroy($imagen);
?>
```

Administración de fechas y horas (función date)

```
● ● ●  
<html>  
  
<head>  
    <title>Problema</title>  
</head>  
  
<body>  
    <?php  
        echo "La fecha de hoy es:";  
        $fecha = date("d/m/Y");  
        echo $fecha;  
        echo "<br>";  
        echo "La hora actual es:";  
        $hora = date("H:i:s");  
        echo $hora;  
        echo "<br>";  
    ?>  
    <a href="pagina2.php">Siguiente problema</a>  
</body>  
  
</html>
```

Los caracteres de formato utilizados en las dos llamadas de la función date son:

d día del mes con dos dígitos "01" al "31"

m mes con dos dígitos "01" al "12"

Y año con cuatro dígitos

Para la hora, los caracteres que serán sustituidos son:

H hora con dos dígitos "00" a "23"

i minutos con dos dígitos "00" a "59"

s segundos con dos dígitos "00" a "59"

Los caracteres de formato que veíamos son los más comunes, pero tenemos otras variantes posibles. Si queremos los días y meses sin el cero delante y el año con dos dígitos tenemos entonces:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        echo "La fecha de hoy es:";
        $fecha = date("j/n/y");
        echo $fecha;
        echo "<br>";
    ?>
    <a href="pagina3.php">Siguiente problema</a>
</body>

</html>
```

Los caracteres que ahora tenemos son:
j día del mes "1" al "31"
n mes "1" al "12"
y año con dos dígitos

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $dato = date("L");
        if ($dato == 1)
            echo "Año bisiesto";
        else
            echo "Año no bisiesto";
        echo "<br>";
        echo "Día de la semana:";
        $dato = date("w");
        switch ($dato) {
            case 0:
                echo "domingo";
                break;
            case 1:
                echo "lunes";
                break;
            case 2:
                echo "martes";
                break;
            case 3:
                echo "miércoles";
                break;
            case 4:
                echo "jueves";
                break;
            case 5:
                echo "viernes";
                break;
            case 6:
                echo "sábado";
                break;
        }
    ?>
</body>

</html>
```

Los caracteres son:

L "1" o "0", según si el año es bisiesto o no
w día de la semana, en número, de "0" (domingo) a "6" (sábado)

Problema propuesto.

Confeccionar un programa que permita registrar una queja en un archivo de texto. Disponer un formulario que permita almacenar el nombre del visitante (objeto HTML de tipo "text"), la descripción de la queja (objeto de tipo "textarea"). Almacenar en el archivo de texto la fecha y hora de la queja.

El archivo "pagina1.php" debe ser el formulario de ingreso de datos.

El archivo "pagina2.php" debe registrar la información en el archivo de datos.

Por último el archivo "pagina3.php" debe imprimir el archivo con todas las quejas registradas hasta el momento. Disponer un hipervínculo en el archivo "pagina1.php" para poder ver todas las quejas registradas.

pagina1.php

```
<form action="pagina2.php" method="post">
    Ingrese su nombre:
    <input type="text" name="nombre">
    <br>
    Queja:
    <br>
    <textarea name="queja" rows="10" cols="40">
    </textarea>
    <br>
    <input type="submit" value="Registrar">
</form>
<a href="pagina3.php">Quejas registradas</a>
</body>

</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$ar = fopen("datos.txt", "a") or
    die("Problemas en la creacion");
fputs($ar, $_POST['nombre']);
fputs($ar, "\n");
fputs($ar, $_POST['queja']);
fputs($ar, "\n");
fputs($ar, "Fecha y Hora:");
$fecha = date("d/m/Y");
fputs($ar, $fecha);
fputs($ar, " ");
$hora = date("H:i:s");
fputs($ar, $hora);
fputs($ar, "\n");
fputs($ar, "-----\n");
fclose($ar);
echo "Los datos se cargaron correctamente.";

?>

</body>

</html>
```

pagina3.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

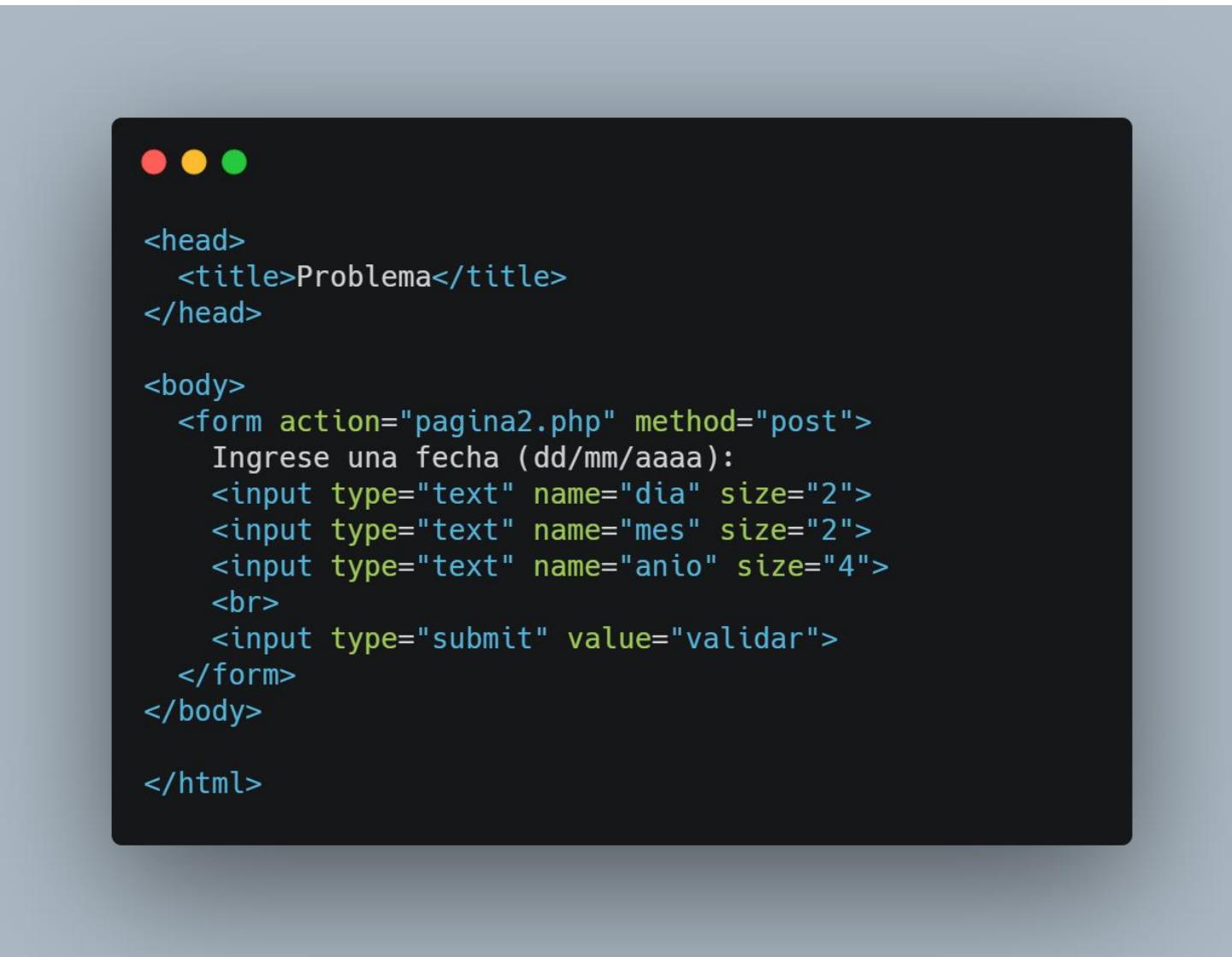
<body>

<?php
$ar = fopen("datos.txt", "r") or
die("No se pudo abrir el archivo");
while (!feof($ar)) {
    $linea = fgets($ar);
    $lineasalto = nl2br($linea);
    echo $lineasalto;
}
fclose($ar);
?>

</body>

</html>
```

Validación de una fecha ingresada por teclado (checkdate)



The image shows a screenshot of a Mac OS X desktop environment. A dark gray window is open, representing a web browser or developer tool. In the top-left corner of the window, there are three colored circular icons: red, yellow, and green. The window contains the following HTML code:

```
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese una fecha (dd/mm/aaaa):
        <input type="text" name="dia" size="2">
        <input type="text" name="mes" size="2">
        <input type="text" name="anio" size="4">
        <br>
        <input type="submit" value="validar">
    </form>
</body>

</html>
```



```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        if (checkdate($_POST['mes'], $_POST['dia'], $_POST['anio']))
            echo "La fecha ingresada es correcta";
        else
            echo "La fecha no es válida";
    ?>
</body>

</html>
```

El programa modificado quedará entonces:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    if (
        is_numeric($_POST['dia']) &&
        is_numeric($_POST['mes']) &&
        is_numeric($_POST['anio'])
    ) {
        if (checkdate($_POST['mes'], $_POST['dia'], $_POST['anio']))
            echo "La fecha ingresada es correcta";
        else
            echo "La fecha no es válida";
    } else
        echo "La fecha no es válida";
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un formulario que solicite la carga de una fecha, disponer tres controles HTML de tipo "select" para elegir el día, el mes y el año. Validar si se trata de una fecha válida en la segunda página.

pagina1.php

```
● ● ●
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Seleccion la fecha(dd/mm/aaaa):
        <select name="dia">
            <option values="1">1</option>
            <option values="2">2</option>
            <option values="3">3</option>
            <option values="4">4</option>
            <option values="5">5</option>
            <option values="6">6</option>
            <option values="7">7</option>
            <option values="8">8</option>
            <option values="9">9</option>
            <option values="10">10</option>
            <option values="11">11</option>
            <option values="12">12</option>
            <option values="13">13</option>
            <option values="14">14</option>
            <option values="15">15</option>
            <option values="16">16</option>
            <option values="17">17</option>
            <option values="18">18</option>
            <option values="19">19</option>
            <option values="20">20</option>
            <option values="21">21</option>
            <option values="22">22</option>
            <option values="23">23</option>
            <option values="24">24</option>
            <option values="25">25</option>
            <option values="26">26</option>
            <option values="27">27</option>
            <option values="28">28</option>
            <option values="29">29</option>
            <option values="30">30</option>
            <option values="31">31</option>
        </select>
        <select name="mes">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
            <option value="9">9</option>
            <option value="10">10</option>
            <option value="11">11</option>
            <option value="12">12</option>
        </select>
        <select name="anio">
            <option value="2003">2003</option>
            <option value="2004">2004</option>
            <option value="2005">2005</option>
            <option value="2006">2006</option>
            <option value="2007">2007</option>
            <option value="2008">2008</option>
            <option value="2009">2009</option>
            <option value="2010">2010</option>
            <option value="2011">2011</option>
            <option value="2012">2012</option>
            <option value="2013">2013</option>
            <option value="2014">2014</option>
            <option value="2015">2015</option>
            <option value="2016">2016</option>
            <option value="2017">2017</option>
            <option value="2018">2018</option>
            <option value="2019">2019</option>
            <option value="2020">2020</option>
        </select>
        <br>
        <input type="submit" value="validar">
    </form>
</body>
</html>
```

pagina2.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
<?php

    if (checkdate($_POST['mes'], $_POST['dia'], $_POST['anio']))
        echo "La fecha ingresada es correcta";
    else
        echo "La fecha no es válida";

    ?>
</body>

</html>
```

Carga de una fecha en una tabla de MySQL

pagina1.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre:
        <input type="text" name="nombre"><br>
        Ingrese mail:
        <input type="text" name="mail"><br>
        Ingrese la fecha de nacimiento (dd/mm/aaaa):
        <input type="text" name="dia" size="2">
        <input type="text" name="mes" size="2">
        <input type="text" name="anio" size="4">
        <br>
        Seleccione el curso:
        <select name="codigocurso">
            <?php
                $conexion = mysqli_connect("localhost", "root", "", "base1") or
                    die("Problemas con la conexión");

                $registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
                    die("Problemas en el select:" . mysqli_error($conexion));
                while ($reg = mysqli_fetch_array($registros)) {
                    echo "<option value=\"$reg[codigo]\">$reg[nombrecurso]</option>";
                }
            ?>
        </select>
        <br>
        <input type="submit" value="Registrar">
    </form>
</body>

</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "basel") or
            die("Problemas con la conexión");

        $fechanacimiento = $_POST['anio'] . "-" . $_POST['mes'] . "-" . $_POST['dia'];
        mysqli_query($conexion, "insert into alumnos(nombre,mail,codigocurso,fechanac) values
            ('$_POST[nombre]','$_POST[mail]',
            $_POST[codigocurso],'$fechanacimiento')") or
            die("Problemas en el select" . mysqli_error($conexion));
        mysqli_close($conexion);
        echo "El alumno fue dado de alta.";
    ?>
    <br>
    <a href="pagina3.php">ver listado de alumnos</a>
</body>

</html>
```

pagina3.php

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $conexion = mysqli_connect("localhost", "root", "", "base1") or
            die("Problemas con la conexión");

        $registros = mysqli_query($conexion, "select alu.codigo as codigo,nombre,mail,codigocurso,fechanac,
            nombrecurso from alumnos as alu
            inner join cursos as cur on cur.codigo=alu.codigocurso") or
            die("Problemas en el select:" . mysqli_error($conexion));
        while ($reg = mysqli_fetch_array($registros)) {
            echo "Codigo:" . $reg['codigo'] . "<br>";
            echo "Nombre:" . $reg['nombre'] . "<br>";
            echo "Mail:" . $reg['mail'] . "<br>";
            echo "Fecha de Nacimiento:" . $reg['fechanac'] . "<br>";
            echo "Curso:" . $reg['nombrecurso'] . "<br>";
            echo "<hr>";
        }
        mysqli_close($conexion);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un programa que permita efectuar el alta en la tabla alumnos. Para el ingreso de la fecha de nacimiento del alumno, disponer 3 controles de tipo "select" (en el día disponer valores entre 1 y 31, en el mes valores comprendidos entre 1 y 12 y por último en el año disponer valores comprendidos entre 1900 y 2020) Validar la fecha en la página que se efectúa el insert propiamente dicho.

Recomendación: Plantear una estructura repetitiva de PHP para la creación del control "select" donde se selecciona el año de nacimiento.

pagina1.php

```
<html>
<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese nombre:
        <input type="text" name="nombre"><br>
        Ingrese mail:
        <input type="text" name="mail"><br>
        Seleccione la fecha de nacimiento(dd/mm/aaaa):
        <select name="dia">
            <option values="1">1</option>
            <option values="2">2</option>
            <option values="3">3</option>
            <option values="4">4</option>
            <option values="5">5</option>
            <option values="6">6</option>
            <option values="7">7</option>
            <option values="8">8</option>
            <option values="9">9</option>
            <option values="10">10</option>
            <option values="11">11</option>
            <option values="12">12</option>
            <option values="13">13</option>
            <option values="14">14</option>
            <option values="15">15</option>
            <option values="16">16</option>
            <option values="17">17</option>
            <option values="18">18</option>
            <option values="19">19</option>
            <option values="20">20</option>
            <option values="21">21</option>
            <option values="22">22</option>
            <option values="23">23</option>
            <option values="24">24</option>
            <option values="25">25</option>
            <option values="26">26</option>
            <option values="27">27</option>
            <option values="28">28</option>
            <option values="29">29</option>
            <option values="30">30</option>
            <option values="31">31</option>
        </select>
        <select name="mes">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
            <option value="9">9</option>
            <option value="10">10</option>
            <option value="11">11</option>
            <option value="12">12</option>
        </select>
        <select name="anio">
            <?php
                for ($f = 1900; $f <= 2020; $f++) {
                    echo "<option value=\"$f\">$f</option>";
                }
            ?>
        </select>
        <br>
        Seleccione el curso:
        <select name="codigocurso">
            <?php
                $conexion = mysqli_connect("localhost", "root", "", "base1") or
                    die("Problemas con la conexión");
                $registros = mysqli_query($conexion, "select codigo,nombrecurso from cursos") or
                    die("Problemas en el select:" . mysqli_error($conexion));
                while ($reg = mysqli_fetch_array($registros)) {
                    echo "<option value=\"$reg[codigo]\\$reg[nombrecurso]>$reg[nombrecurso]</option>";
                }
            ?>
        </select>
        <br>
        <input type="submit" value="Registrar">
    </form>
</body>
</html>
```

pagina2.php

```
<html>

<head>
    <title>Problema</title>
</head>

<body>
<?php
if (checkdate($_POST['mes'], $_POST['dia'], $_POST['anio'])) {
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
        die("Problemas con la conexión");

    $fechanacimiento = $_POST['anio'] . "-" . $_POST['mes'] . "-" . $_POST['dia'];

    mysqli_query($conexion, "insert into alumnos(nombre,mail,codigocurso,fechanac) values
        ('$_POST[nombre]','$_POST[mail]',
        $_POST[codigocurso],'$fechanacimiento')") or
        die("Problemas en el select" . mysqli_error($conexion));
    mysqli_close($conexion);
    echo "El alumno fue dado de alta.";
} else {
    echo "Fecha incorrecta";
}
?>
</body>

</html>
```

Formateo de datos en una página (printf)

página1.php

El siguiente ejemplo muestra el contenido de una variable entera con distintos formatos, lo mismo hacemos para una variable de tipo double:

```
● ● ●

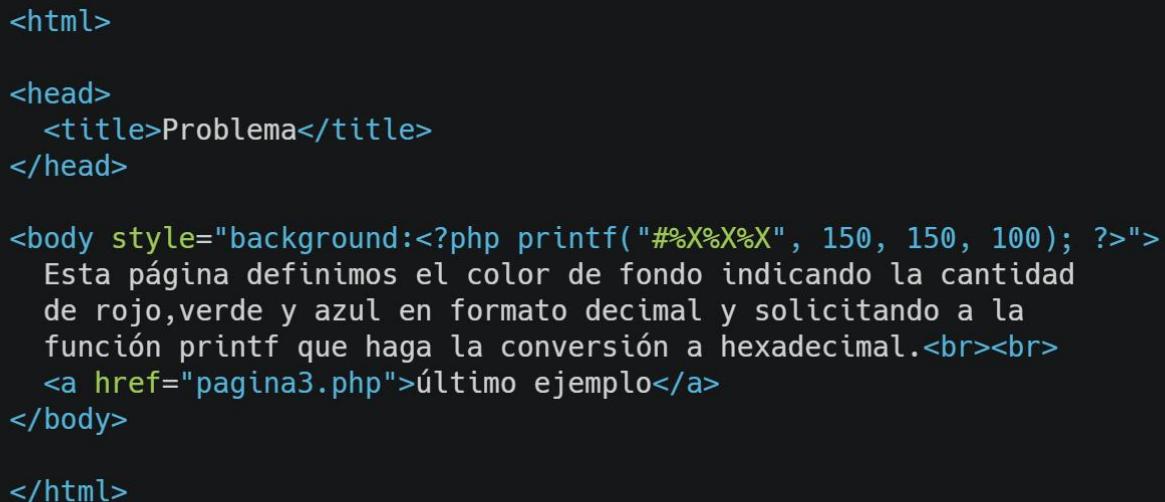
<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
    $entero = 255;
    printf("Valor entero en formato decimal %d <br>", $entero);
    printf("Valor entero en formato hexadecimal con letras minúsculas %x<br>", $entero);
    printf("Valor entero en formato hexadecimal con letras mayúsculas %X<br>", $entero);
    printf("Valor entero en formato binario %b<br>", $entero);
    printf("Valor entero en formato octal %o<br>", $entero);
    $letra = 65;
    printf("Valor entero como carácter ascii %c<br>", $letra);
    echo "<br>";
    $real = 10.776;
    printf("Impresión de un valor de tipo double %f <br>", $real);
    printf("Impresión de un valor de tipo double indicando la cantidad de decimales a imprimir %0.2f <br>", $real);
    ?>
    <br>
    <A href="pagina2.php">Algunas utilidades de estas conversiones</A>
</body>

</html>
```

Veamos una utilidad de la función printf formateando a tipo de dato hexadecimal:



```
<html>
<head>
    <title>Problema</title>
</head>

<body style="background:<?php printf( "#%X%X%X", 150, 150, 100); ?>">
    Esta página definimos el color de fondo indicando la cantidad
    de rojo,verde y azul en formato decimal y solicitando a la
    función printf que haga la conversión a hexadecimal.<br><br>
    <a href="pagina3.php">Último ejemplo</a>
</body>

</html>
```

La función printf puede formatear n datos en una única llamada, como ocurre en este caso:

```
<body style="background:<?php printf("#%X%X%X",150,150,100); ?>">
```

Cuando tenemos los tres valores en formato decimal, para crear un color, la función printf nos facilita la tarea de generar el color definitivo en hexadecimal.

Por último, con la función printf, podemos determinar el número de caracteres que va a ocupar o en su defecto se rellenará con ceros:

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <?php
        $dia = 6;
        $mes = 5;
        $anio = 2019;
        printf("%02d/%02d/%d", $dia, $mes, $anio);
    ?>
</body>

</html>
```

Problema propuesto.

Confeccionar un formulario que solicite la carga del nombre de una persona, su mail y un importe que desea donar.

En la página que procesa los datos ingresados, mostrar el importe a donar rellenando con ceros, por ejemplo: \$0000170 dolares.

pagina1.php

```
<html>
<head>
<title>Problema</title>
</head>
<body>

<form action="pagina2.php" method="post">
Ingrese nombre:
<input type="text" name="nombre" size="40">
<br>
Ingrese mail:
<input type="text" name="mail" size="70">
<br>
Ingrese importe a donar:
<input type="text" name="importe" size="10">
<br>

<input type="submit" value="Confirmar">
</form>

</body>
</html>
```

pagina2.php

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
echo "Nombre:". $_POST['nombre'];
echo "<br>";
echo "Mail:". $_POST['mail'];
echo "<br>";
echo "Importe a donar:" ;
printf("%07d",$_POST['importe']);
?>
</body>
</html>
```

Formato de datos y salida a un string (sprintf)

Así como vimos que la función printf nos permite tener un control más fino sobre cómo un dato debe imprimirse en una página, la función sprintf nos permite formatear la salida de un dato hacia un string y no a la página HTML.

Tipos de conversión utilizadas por la función sprintf (tener en cuenta que son los mismos caracteres de control que se aplican a la función printf):

%b Formatea un entero como un número binario.

%d Formatea un entero como un número decimal con signo.

%u Formatea un entero como un número decimal sin signo.

%o Formatea un entero como un número octal.

%x Formatea un entero como un número hexadecimal en minúsculas.

%X Formatea un entero como un número hexadecimal en mayúsculas.

%c Formatea un entero como un carácter ASCII.

%f Formatea un double con una cantidad de decimales.

%s Formatea un string.

Un ejemplo utilizando la función sprintf:

```
<html>
<head>
    <title>Problema</title>
</head>
<body>
    <?php
        function retornarColorHexa($rojo, $verde, $azul)
        {
            $color = sprintf("#%02X%02X%02X", $rojo, $verde, $azul);
            return $color;
        }
    ?>

    <table>
        <tr>
            <td style="background:<?php echo retornarColorHexa(255, 0, 0) ?>">Cuadro rojo</td>
            <td style="background:<?php echo retornarColorHexa(0, 255, 0) ?>">Cuadro verde</td>
            <td style="background:<?php echo retornarColorHexa(0, 0, 255) ?>">Cuadro azul</td>
        </tr>
    </table>
</body>
</html>
```

Implementamos una función a la cual le enviamos 3 valores decimales y nos retorna un string que especifica un color en hexadecimal.

Problema propuesto.

Almacenar en un string la tabla de caracteres ASCII y luego imprimirla.

```
● ● ●

<html>

<head>
    <title>Problema</title>
</head>

<body>

<?php
$tabla = "";
for ($f = 32; $f <= 255; $f++) {
    $tabla = $tabla . sprintf("%c", $f);
}
echo $tabla;
?>

</body>

</html>
```