PHP 2

Creación de un archivo de texto.

pagina1.html

```
<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>
```

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

```
<html>

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

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

</html>
```

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.

Ejercicio 12

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 "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.

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:

pagina1.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>
```

Ejercicio 13

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)

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 titulo centrado en patalla, y la llamaremos posteriormente dos veces:

```
<html>

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

<body>

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

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

  ?>

</body>

</html>
```

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

```
<html>

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

<body>

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

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

```
  echo $p;
  ?>

</body>

</html>
```

Ejercicio 14

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

Base de datos (MySQL)

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
- SQLlite

El más empleado en la actualidad en la web es el gestor de base de datos MySQL.

INSERT (Alta de registros en una tabla)

Luego de crear una base de datos y sus tablas, 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>
```

Ahora veremos como realizar la registración de los datos cargados en el formulario, en la tabla alumnos de la base de datos.

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

  mysqli_close($conexion);

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

</html>
```

La función mysqli_connect se conecta a una base de datos de tipo MySql, el primer parámetro es la dirección donde se encuentra el gestor de base de datos (en este caso en el mismo servidor por lo que indicamos esto con "localhost"), el segundo parámetro es el nombre de usuario de la base de datos ("root" en nuestro caso, que es el usuario por defecto que crea MySql para el administrador), seguidamente indicamos la clave del usuario root (por defecto al instalar el XAMPP se crea con clave vacía) y por último indicamos el nombre de la base de datos a conectarnos (en nuestro ejemplo ya creamos la base de datos llamada: base1 que tiene la tabla alumnos)
En caso de haber algún error en la llamada a la función la misma retorna false por lo que se ejecuta la instrucción seguida del operador or, en nuestro caso llamamos a la función die que detiene la ejecución del programa y muestra el mensaje por pantalla.

El paso más importante es la codificación del comando SQL insert(debemos llamar a la función mysqli_query pasando como primer parámetro la referencia a la conexion y el segundo parámetro es un string con el comando insert):

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

La sintaxis del comando insert es bastante sencilla, indicamos el nombre de la tabla y los campos de la tabla a cargar. Luego debemos indicar en el mismo orden los valores a cargar en cada campo (dichos valores los rescatamos del formulario anterior).

Los campos de tipo varchar SQL requiere que encerremos entre comillas simples, esto sucede para el nombre y el mail; en cambio, para el codigocurso esto no debe ser así.

Otra cosa a tener en cuenta es que los subíndices de los vectores asociativos no deben ir entre simples comillas ya que se encuentran dentro de un string, sino se producirá un error.

Ejercicio 15

Crear en la base de datos "base1" otra tabla llamada "cursos".
La estructura de esta segunda tabla debe ser:

```
codigo int auto_increment primery_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.

Listado (selección de registros de una tabla)

Para recuperar datos desde MySQL o MariaDB debemos emplear el comando select:

```
select codigo,nombre,mail,codigocurso from alumnos
```

Debemos pasar desde PHP un string con este comando para que MySQL lo ejecute y retorne todas las filas de la tabla alumnos.

Veremos entonces como recuperar los datos almacenados en la tabla alumnos de la base de datos "base1".

El programa que muestra los registros en una página es:

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,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>
```

Si el comando SQL es correcto, en la variable $registros se almacena una referencia a los datos rescatados de la tabla alumnos. Ahora debemos ir mostrando registro a registro los datos extraídos:

```
 while ($reg = mysqli_fetch_array($registros)) {
```

Ejercicio 16

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.

DELETE (Baja de un registro en una tabla)

Tenemos el archivo "pagina2.php" que se encarga de buscar el mail ingresado en el formulario y en caso que exista se procede a borrarlo:

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

Ejercicio 17

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.

UPDATE (Modificación de un registro de una tabla)

De las actividades con una tabla esta es la más larga. Vamos a resolverlo implementando tres páginas, la primera un formulario de consulta del mail de

un alumno, la segunda otro formulario que nos permita cargar su mail modificado y la última registrará el cambio en la tabla.

El formulario de consulta del mail del alumno es similar a problemas anteriores:

```
<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>
```

La segunda página es la más interesante y con conceptos nuevos:

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='$_REQUEST[mail]'") or
    die("Problemas en el select:" . mysqli_error($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>
```

Un concepto importante es como enviar el mail del primer formulario a la tercer página, esto se logra con los controles de tipo "hidden", este tipo de control no se visualiza en el formulario pero se envía al presionar el botón submit.

Si queremos que el control text se cargue con el mail ingresado en el formulario anterior debemos cargar la propiedad value con dicho valor:

```
     <input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
```

Por último la pagina3.php es la que efectúa la modificación de la tabla propiamente dicha. Con el mail ingresado en la pagina1.php, el mail modificado en la pagina2.php se efectúa el update.

```
<html>

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

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

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

</html>
```

Ejercicio 18

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