SESIÓN - 32

Conexión a la base de datos con PDO

Desarrollo:

1. Conexión a la base de datos con PDO.

En primer lugar antes de crear nuestra conexión es verificar que tenemos PDO disponible para trabajar; para ello creamos el siguiente archivo y colocamos el código necesario para verificar que este habilitado. Crea el archivo con el nombre de phpinfo.php y escribe este código:

```
<?php
        phpinfo();
?>
```

Al ejecutarlo en el servidor esta es la información que te debe de salir:

PDO

PDO support	enabled
PDO drivers	mysql, sqlite

pdo_mysql

PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.11-dev - 20120503 - $Id: 40933630edef551dfaca71298a83fad8d03d62d4 $

Si todo esta correcto entonces ahora si procedemos a crear nuestra conexión con la clase PDO. Pero antes que nada crearemos una base de datos con el nombre *dbjobs* en nuestro servidor con la siguiente tabla *cetpagoslocal* que tendrá 3 campos (*id, denominacion, direccion*); una vez hecho esto nos dirigimos a nuestra carpeta creada y proceder a implementar el código de conexión.

Para realizar una nueva conexión se debe crear una instancia del objeto PDO. Este constructor acepta una serie de parámetros de conexión que pueden ser específicos para cada sistema de bases de datos.

Si no se logra establecer la conexión se producirá una excepción (PDOException). Si la conexión es exitosa, una instancia de PDO será devuelta. La conexión permanece activa por todo el ciclo de vida del objeto PDO. Para cerrar la conexión, se debe destruir el objeto asegurándose que toda referencia sea eliminada, o bien, PHP cerrará la conexión automáticamente cuando el programa finalice.

```
<html>
 <head>
  <title>PDO-prueba</title>
 </head>
 <body>
  <?php
  class Conectar
  {
    protected $pdo;
     public function __CONSTRUCT()
            {
              try
               {
                 $this->pdo = new PDO('mysql:host=localhost;dbname=dbjobs', 'root', '');
                 $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                      }
                      catch(Exception $e)
                      {
                       $e->getMessage() ;
                      }
            }
  }
  ?>
 </body>
</html>
```

Como podrán observar es muy sencillo el código de conexión:

```
$this->pdo = new PDO('mysql:host=localhost;dbname=dbjobs', 'root', '');
```

Creamos con el "new" una instancia de clase PDO (lo que nos permitirá usar varias a la vez), definimos el tipo de base de datos con "mysql:", el nombre del servidor "localhost" (en este caso), el nombre de la base de datos "dbname=dbjobs", y después el nombre de usuario y la contraseña del servidor de bases de datos (si las tuviera configuradas o las que están por defecto) "*'root', ' '*".

Si la aplicación no captura la excepción lanzada por el constructor de PDO, la acción predeterminada que toma el servidor es la de finalizar el script y mostrar información de seguimiento. Esta información probablemente revelará todos los detalles de la conexión a la base de datos, incluyendo el nombre de usuario y la contraseña.

Cuando se utiliza PDO o algún otro código que dispare una excepción (el termino en inglés es throw) se necesita agrupar el código en un bloque try-catch, si todo marcha bien nuestro código se ejecuta sin error, en dado caso de que salga algo mal agarramos el error en el catch.

```
catch(Exception $e)
    {
        $e->getMessage() ;
    }
```

Debemos de tener cuidado de que:

```
$e->getMessage() ;
```

No esté siempre visible, ya que si se presenta un error nuestro usuario de conexión o nuestra contraseña puede ser mostrada, el uso de getMessage() es para que nosotros podamos depurar nuestro código, entonces lo que debemos de hacer es mostrar otro mensaje, quedando nuestro código para ser usado de la siguiente manera :

```
<html>
 <head>
   <title>PDO-prueba</title>
 </head>
 <body>
  <?php
   class Conectar
   {
    protected $pdo;
     public function __CONSTRUCT()
            {
              try
                {
                    $this->pdo = new PDO('mysql:host=localhost;dbname=dbjobs', 'root', '');
                    $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                    echo 'conexión exitosa';
                  }
                  catch(Exception $e)
                  {
                    echo 'Fallo de conexión...';
                  }
            }
   }
  ?>
 </body>
</html>
```

Una vez realizada con éxito una conexión a la base de datos, será devuelta una instancia de la clase PDO al script. La conexión permanecerá activa durante el tiempo de vida del objeto PDO. Para cerrar la conexión, es necesario destruir el objeto asegurándose de que todas las referencias a él existentes sean eliminadas (esto se puede hacer asignando NULL a la variable que contiene el objeto). Si no se realiza explícitamente, PHP cerrará automáticamente la conexión cuando el script finalice.

```
<?php
 $this->pdo = new PDO('mysql:host=localhost;dbname=dbjobs', 'root', '');
// Utilizar la conexión aquí
// Ya se ha terminado; se cierra
$pdo = null;
?>
```