



Guía Práctica de Ejercicios N° 7

En esta séptima práctica emplearemos los conceptos de la programación POO en el lenguaje PHP para comenzar con el estudio de acceso a base de datos utilizando capas de abstracción. Comenzando con una introducción a las extensiones del lenguaje y principalmente a las de base de datos. Luego presentaremos el motor de base de datos MySQL con todas sus destacadas características. A continuación de MySQL veremos el lenguaje de consultas estructurado que nos permitirá hacer una interacción con los datos de las bases de datos. Finalmente, daremos la definición, la estructura y el funcionamiento del acceso a base de datos utilizando PDO (**PHP Data Object**) como capa de abstracción.

Objetivo: Presentar las extensiones de base de datos de PHP. Presentar el motor de base de datos MySQL. Estudiar y repasar el lenguaje de consultas estructurado SQL. Y comprender con claridad el uso de la capa de abstracción PDO en PHP5.

Introducción: En nuestra actualidad vemos que el desarrollo de Internet ha sido inminente y con ello las aplicaciones Web, por lo tanto, se hace indispensable el uso de un lenguaje que permita desarrollar aplicaciones Web como PHP, entre otros. Teniendo en cuenta la situación anterior es que veremos la como desarrollar aplicaciones Web que se ejecutarán del lado del servidor utilizando uno de los mejores lenguajes del ambiente del Software Libre.

Construcciones (palabras clave), funciones y/o métodos que estudiaremos en esta unidad.

Métodos de la clase PDO, PDOStatement y PDOException del lenguaje que veremos.

PDO::__construct crea una instancia de la clase que representa una conexión.

PDO::exec ejecuta una SQL declarada principalmente con sentencias para modificar y/o eliminar filas de una tabla, retornando un valor entero con la cantidad de filas afectadas.

PDO::getAvailableDrivers obtiene en un arreglo todos los controladores de PDO que actualmente se encuentran disponibles.

PDO::errorCode obtiene el SQLSTATE asociado con la última operación sobre el manejador de la base de datos.

PDO::errorInfo obtiene información extendida del error asociado con la última operación sobre el manejador de la base de datos.

PDO::getAttribute obtiene el valor de un atributo de la conexión a la base de datos.

PDO::setAttribute establece el valor de un atributo del manejador de la base de datos.



Ejercicios

Preparar y verificar nuestro entorno de desarrollo PHP

Antes de comenzar con la práctica, asegúrese de tener instalada la extensión PDO para PHP.

1) – Verifique y realice la instalación/activación de ser necesario de la extensión PDO en el servidor Web (configuración PHP).

2) – Verifique e Instale de ser necesario el conector PDO de MySQL en el sistema.

Obtener y establecer valores en atributos de la conexión a la base de datos

3) – Crear un script que realice una conexión a la base de datos y obtenga los valores de los siguientes atributos de la conexión:

- PDO::ATTR_AUTOCOMMIT
- PDO::ATTR_CONNECTION_STATUS
- PDO::ATTR_DRIVER_NAME
- PDO::ATTR_ERRMODE
- PDO::ATTR_SERVER_INFO
- PDO::ATTR_SERVER_VERSION

```
$pdo = new PDO('mysql:host=127.0.0.1;','root','');  
echo $pdo->getAttribute(PDO::ATTR_AUTOCOMMIT) . '<br>';  
echo $pdo->getAttribute(PDO::ATTR_CONNECTION_STATUS) . '<br>';  
echo $pdo->getAttribute(PDO::ATTR_DRIVER_NAME) . '<br>';  
echo $pdo->getAttribute(PDO::ATTR_ERRMODE) . '<br>';  
echo $pdo->getAttribute(PDO::ATTR_SERVER_INFO) . '<br>';  
echo $pdo->getAttribute(PDO::ATTR_SERVER_VERSION) . '<br>';
```

4) – Crear otro script para establecer nuevos valores a los siguientes atributos de la conexión a la base de datos:

- PDO::ATTR_ERRMODE (pasa del valor 0 a 1, recuerde utilizar constantes)
- PDO::ATTR_CASE (pasa del valor 0 a 2, recuerde utilizar constantes)

```
$pdo = new PDO('mysql:host=127.0.0.1;','root','');  
$pdo->setAttribute(PDO::ATTR_ERRMODE, 1);
```



```
$pdo->setAttribute(PDO::ATTR_CASE, 2);
```

Crear base de datos y tablas en MySQL utilizando PDO

5) – Crear una base de datos nueva con el nombre **phpn1_db_clase7**. Utilice la sentencia CREATE TABLE en la declaración de la consulta que se ejecutará desde el método exec() de la clase PDO.

```
$pdo->exec('CREATE DATABASE phpn1_db_clase7');
```

6) – Crear una tabla nueva en la base de datos **phpn1_db_clase7** con el nombre **tipo_documento** con los siguientes campos:

- id_tipo_documento INT NOT NULL AUTO_INCREMENT PRIMARY KEY
- descripcion VARCHAR(30) NOT NULL
- observacion VARCHAR(50) NOT NULL

```
$pdo->exec('USE phpn1_db_clase7');
```

```
$query = "
```

```
CREATE TABLE tipo_documento (
```

```
id_tipo_documento INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
descripcion VARCHAR(30) NOT NULL,
```

```
observacion VARCHAR(50) NOT NULL
```

```
)
```

```
";
```

```
$pdo->exec($query);
```

7) – Crear otra tabla nueva en la base de datos **phpn1_db_clase7** con el nombre **persona** que posea los siguientes campos:

- id_persona int NOT NULL AUTO_INCREMENT PRIMARY KEY
- id_tipo_documento int NOT NULL
- apellidos varchar(30) NOT NULL
- nombres varchar(50) NOT NULL
- correo_electronico varchar(150) NOT NULL

```
$pdo->exec('USE phpn1_db_clase7');
```

```
$query = "
```

```
CREATE TABLE persona (
```

```
id_persona INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
id_tipo_documento INT NOT NULL,
```



```
apellidos VARCHAR(30) NOT NULL,  
nombres VARCHAR(50) NOT NULL,  
correo_electronico VARCHAR(150) NOT NULL  
)  
";  
$pdo->exec($query);
```

8) – Crear un índice para el campo **id_tipo_documento** de la tabla **persona** de la base de datos **phpn1_db_clase7** con el nombre **idx_personas_1**.

```
$pdo->exec('USE phpn1_db_clase7');  
$pdo->exec("CREATE INDEX idx_personas_1 ON persona(id_tipo_documento)");
```

9) – Alterar la estructura de la tabla **persona** de la base de datos **phpn1_db_clase7** con la siguiente sentencia SQL:

```
ALTER TABLE persona ADD CONSTRAINT fk_id_tipo_documento FOREIGN KEY  
(id_tipo_documento) REFERENCES tipo_documento (id_tipo_documento);
```

```
$pdo->exec('USE phpn1_db_clase7');  
$query = "  
ALTER TABLE persona  
ADD CONSTRAINT fk_id_tipo_documento FOREIGN KEY (id_tipo_documento)  
REFERENCES tipo_documento (id_tipo_documento)  
";  
$pdo->exec($query);
```

Insertar datos en las tablas utilizando la capa de abstracción PDO

10) – Realice un script que permita insertar datos en la tabla **tipo_documento** de la base de datos **phpn1_db_clase7**. Las filas a insertar definen los tipos de documentos de una persona como por ejemplo: DNI, LC, LE y Pasaporte.

```
$pdo->exec('USE phpn1_db_clase7');  
$pdo->exec("INSERT INTO tipo_documento (descripcion, observacion) VALUES('DNI','DNI')");  
$pdo->exec("INSERT INTO tipo_documento (descripcion, observacion) VALUES('LC','LC')");  
$pdo->exec("INSERT INTO tipo_documento (descripcion, observacion) VALUES('LE','LE')");  
$pdo->exec("INSERT INTO tipo_documento (descripcion, observacion) VALUES('Pasaporte','Pasaporte')");
```

11) – Modifique el script del punto anterior para insertar sus datos personales en la tabla **persona** de la base de datos **phpn1_db_clase7**.

```
$pdo->exec('USE phpn1_db_clase7');  
$pdo->exec("INSERT INTO persona (id_tipo_documento, apellidos, nombres, correo_electronico) values(1,
```



```
'Sbarbaro', 'Martin', 'martin@gugler.com.ar')");
```

POO y la capa de abstracción PDO

12) – Realice un script (DataBase.php) que contenga la Clase DataBase para realizar el manejo de base de datos utilizando la capa de abstracción PDO. Dentro de la clase declare y desarrolle un método para conectarse a la base de datos creada en esta guía.

Realice un script (index.php) para probar la conexión con la clase DataBase.

class DataBase

```
{  
  
    const CONTROLADOR = 'mysql';  
  
    const SERVIDOR = '127.0.0.1';  
  
    const USUARIO = 'root';  
  
    const CONTRASENIA = '1234';  
  
    const BASE_DATO = 'phpn1_db_clase7';  
  
  
    private $_conexion;  
  
    public function __construct()  
    {  
        $dsn = self::CONTROLADOR . ':host=' . self::SERVIDOR . ';dbname=' . self::BASE_DATO;  
        $this->_conexion = new PDO($dsn, self::USUARIO, self::CONTRASENIA);  
    }  
  
    public function getConexion()  
    {  
        return $this->_conexion;  
    }  
}
```

