



Curso de Programación en PHP

Nivel I

Universidad Autónoma de Entre Ríos
Facultad de Ciencia y Tecnología – Oro Verde – v2.1



Capítulo 4: Programación de base de datos

Repaso de la clase PDO

Descubriendo la clase PDOStatement

Consultas de selección a MySQL utilizando PDO

Insertar, modificar y borrar filas en MySQL

Formas de recorrer filas de una consulta

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

Clase PDO y sus métodos

PDO::beginTransaction()

Inicializa una transacción.

PDO::commit()

Ejecuta una transacción.

PDO::rollBack()

Vuelve atrás una transacción (Rolls back) una transacción.

PDO::quote()

Quotes (Pone entre comillas simples un string) para uso en una query (consulta).

PDO::lastInsertId()

Retorna el ID (identificador) de la última fila insertada o secuencia de valores.



Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

Clase PDO y sus métodos

PDO::query()

Ejecuta una instrucción SQL, retornando un conjunto de resultados (resul set) como un objeto de tipo **PDOStatement**.

PDO::prepare()

Prepara una instrucción para ejecución y retorna un objeto de tipo **PDOStatement**.



Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::beginTransaction — Inicia una transacción

Sintaxis

bool PDO::beginTransaction (void)

Descripción

Inicia una transacción y desactiva el modo auto-commit. Devuelve TRUE en caso de éxito o FALSE en caso de error.





Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::commit — Ejecuta la transacción

Sintaxis

bool PDO::commit (void)

Descripción

Este método ejecuta una transacción, retornando más tarde el modo auto-commit en la conexión a la base de datos. Devuelve TRUE en caso de éxito o FALSE en caso de error.





Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::rollBack — Vuelva atrás una transacción

Sintaxis

bool PDO::rollBack (void)

Descripción

Este método vuelve atrás una transacción iniciada con `PDO::beginTransaction()`. Y retorna al modo auto-commit en la conexión a la base de datos. Devuelve `TRUE` en caso de éxito o `FALSE` en caso de error.





Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::lastInsertId — Retorna el ID de la última fila insertada o el valor de la secuencia

Sintaxis

string **PDO::lastInsertId** ([string \$nombre = NULL])

Descripción

Este método devuelve una cadena que contiene el ID de la última fila insertada o el valor de la secuencia. El parámetro del método permite especificar el nombre de la secuencia de objetos solo para el driver de PostgreSQL.





Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::query — Ejecuta una SQL declarada, retornando un conjunto de resultados en un objeto de la clase PDOStatement

Sintaxis

PDOStatement **PDO::query** (string \$SqlDeclarada)

Descripción

Este método ejecuta la consulta SQL declarada y retorna el conjunto de resultados en un objeto de la clase PDOStatement, en caso de error el método devuelve false.





Clase 8: Utilizando PDO con MySQL

Repaso de la clase PDO

PDO::prepare — Prepara una declaración SQL para ejecutar y retorna un objeto de la clase PDOStatement

Sintaxis

PDOStatement **PDO::prepare** (string \$SqlDeclarada [, array \$opciones_control = array()])

Descripción

Este método prepara una declaración SQL para ser ejecuta por el método PDOStatement::execute(), si se ejecuta con éxito devuelve un objeto de la clase PDOStatement y en caso contrario false o PDOException si se produjo un error.





Capítulo 4: Programación de base de datos

~~Repaso de la clase PDO~~

Descubriendo la clase PDOStatement

Consultas de selección a MySQL utilizando PDO

Insertar, modificar y borrar filas en MySQL

Formas de recorrer filas de una consulta

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

Clase PDOStatement y sus métodos

`PDOStatement->execute()`

Ejecuta una instrucción preparada.

`PDOStatement->debugDumpParams()`

Dump a un comando SQL preparado.

`PDOStatement->errorCode()`

Recupera el SQLSTATE asociado con la última operación realizada en la base de datos.

`PDOStatement->errorInfo()`

Recupera información de error extendida asociada con la ultima operación realizada en la base de datos.



Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

Clase PDOStatement y sus métodos

PDOStatement->**bindColumn()**

Sustituye el valor de una columna de la base de datos a una variable PHP.

PDOStatement->**bindParam()**

Sustituye el valor de un parámetro (argumento), a el nombre de variable especificada.

PDOStatement->**bindValue()**

Sustituye un valor como parámetro.

PDOStatement->**closeCursor()**

Cierra el cursor, habilitando la instrucción a ser ejecutada otra vez.

PDOStatement->**columnCount()**

Retorna el numero de columnas en el conjunto de resultados.



Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

Clase PDOStatement y sus métodos

`PDOStatement->fetch()`

Recupera la siguiente fila de un conjunto de resultados (result set).

`PDOStatement->fetchAll()`

Retorna un array conteniendo todas las filas del conjunto de resultados (result set).

`PDOStatement->fetchColumn()`

Retorna una única columna de la siguiente fila de un conjunto de resultados.

`PDOStatement->fetchObject()`

Recupera la siguiente fila y la retorna como un objeto.

`PDOStatement->getAttribute()`

Recupera una atributo.



Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

Clase PDOStatement y sus métodos

`PDOStatement->getColumnMeta()`

Retorna metadatos de una columna en un conjunto de resultados.

`PDOStatement->nextRowset()`

Avanza a la siguiente Rowset en un multi-rowset.

`PDOStatement->rowCount()`

Retorna el numero de filas afectadas por la última instrucción SQL.

`PDOStatement->setAttribute()`

Modifica un atributo.

`PDOStatement->setFetchMode()`

Modifica el valor por defecto para recuperar datos de la base de datos para la instrucción actual.



Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->execute — Ejecuta una declaración preparada

Sintaxis

bool PDOStatement::execute ([array \$parametros_entrada])

Descripción

Este método ejecuta una declaración SQL preparada. Devuelve TRUE en caso de éxito o FALSE en caso de error.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->bindValue — Enlaza un valor a un parámetro

Sintaxis

bool PDOStatement::bindValue (mixed \$parametro, mixed \$valor [, int \$tipoDeDato = PDO::PARAM_STR])

Descripción

Este método permite enlazar un valor a un parámetro numerado o etiquetado de la declaración SQL que se preparo con el método PDO::prepare(). En caso de error devuelve false, caso contrario true.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->debugDumpParams — imprime una instrucción SQL preparada

Sintaxis

void PDOStatement::debugDumpParams (void)

Descripción

Este método permite imprimir una consulta SQL preparada. Mostrará la consulta SQL, el número de parámetros, la lista de parámetros, los tipos de datos, los valores, entre otros.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->fetch — Obtiene la siguiente fila de un conjunto de resultados

Sintaxis

```
mixed PDOStatement::fetch ([ int $fetch_estilo = PDO::FETCH_BOTH [, int  
$cursor_orientation = PDO::FETCH_ORI_NEXT [, int $cursor_offset = 0 ]]] )
```

Descripción

Este método permite obtener la siguiente fila de un conjunto de resultados. El estilo de la fila que se obtiene es un arreglo indexado por el nombre de la columna y también indexado por número de columna (de 0 a N). En caso de error al intentar obtener la fila retorna false.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->fetchAll — Retorna un arreglo que contiene todas las filas de un conjunto de resultados

Sintaxis

```
array PDOStatement::fetchAll ([ int $fetch_estilo = PDO::FETCH_BOTH [, mixed  
$fetch_argument [, array $ctor_args = array() ]]] )
```

Descripción

Este método permite obtener un arreglo con todas las filas de un conjunto de resultados.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->fetchObject — Obtiene la siguiente fila de un conjunto de resultados retornando los valores en un objeto

Sintaxis

mixed **PDOStatement::fetchObject** ([string \$nombreClase = "stdClass" [, array \$ctor_args]])

Descripción

Este método permite obtener la siguiente fila de un conjunto de resultados en una variable de tipo objeto de la clase stdClass. En caso de error devuelve false.





Clase 8: Utilizando PDO con MySQL

Descubriendo la clase PDOStatement

PDOStatement->setFetchMode — Establece el modo de recuperación de los datos para la presente declaración SQL.

Sintaxis

```
bool PDOStatement::setFetchMode ( int $modo )
```

Descripción

Este método permite establecer el modo de recuperación de los datos para la presente declaración SQL. Espera como parámetro un entero que define el modo de recuperación. Se recomienda utilizar las constantes (**PDO::FETCH_***) para la definición del modo. Devuelve true si tuvo éxito y en caso de error devuelve false.





Capítulo 4: Programación de base de datos

~~Repaso de la clase PDO~~

~~Descubriendo la clase PDOStatement~~

Consultas de selección a MySQL utilizando PDO

Insertar, modificar y borrar filas en MySQL

Formas de recorrer filas de una consulta

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

`PDO::query("consulta")`

`// Consulta simple a una tabla de una DB MySQL con PDO`

```
$dsn = 'mysql:dbname=phpn1_db_clase8;host=127.0.0.1;port=3306';
```

```
$usuario = 'root';
```

```
$contrasenia = '';
```

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'SELECT * FROM persona';
```

```
$stmt = $dbh->query($consulta);
```

```
...
```

```
$stmt = null;
```

```
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

- Consulta simple (una tabla) con selección de campos

```
$dbh = new PDO('mysql:dbname=phpn1_db_clase8;host=127.0.0.1', 'root', '');  
$consulta = 'SELECT apellidos, nombres FROM persona';  
$stmt = $dbh->query($consulta);  
...  
$stmt = null;  
$dbh = null;
```

- Consulta simple (varias tablas) seleccionando todos los campos

```
$dbh = new PDO('mysql:dbname=phpn1_db_clase8;host=127.0.0.1', 'root', '');  
$consulta = 'SELECT * FROM persona, suscripcion';  
$consulta .= ' WHERE persona.id_persona = suscripcion.id_persona';  
$stmt = $dbh->query($consulta);  
...  
$stmt = null;  
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

- Consulta a una DB MySQL con una subconsulta en la clausula WHERE

```
$dbh = new PDO('mysql:dbname=phpn1_db_clase8;host=127.0.0.1', 'root', '');
```

```
$consulta = 'SELECT *';  
$consulta .= ' FROM persona';  
$consulta .= ' WHERE persona.id_persona = ';  
$consulta .= '(SELECT id_persona FROM suscripcion';  
$consulta .= ' WHERE suscripcion.id_suscripcion > 500)';
```

```
$stmt = $dbh->query($consulta);
```

```
...
```

```
$stmt = null;  
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

PDO::prepare(), PDO::bindValue() y PDO::execute()

// Consulta con filtro simple y enlazando valores nombrados

```
$dsn = 'mysql:dbname=phpn1_db_clase8;host=127.0.0.1;port=3306';
```

```
$usuario = 'root';
```

```
$contrasenia = '';
```

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'SELECT * FROM persona WHERE id_persona = :id_persona';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':idpersona', 33, PDO::PARAM_INT);
```

```
$stmt->execute();
```

```
$stmt = null;
```

```
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

PDO::prepare(), PDO::bindValue() y PDO::execute()

// Consulta con filtro simple y enlazando valores numerados

```
$dsn = 'mysql:dbname=phpn1_db_clase8;host=127.0.0.1;port=3306';
```

```
$usuario = 'root';
```

```
$contrasenia = '';
```

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'SELECT * FROM persona WHERE id_persona = ?';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindParam(1, 22, PDO::PARAM_INT);
```

```
$stmt->execute();
```

```
$stmt = null;
```

```
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Consultas de selección a MySQL con PDO

PDO::prepare(), PDO::bindColumn() y PDO::execute()

// Consulta simple y enlazando las columnas de la consulta

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'SELECT apellidos, nombres FROM persona';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindColumn('apellidos', $apellidos);
```

```
$stmt->bindColumn('nombres', $nombres);
```

```
$stmt->execute();
```

```
$stmt = null;
```

```
$dbh = null;
```





Capítulo 4: Programación de base de datos

~~Repaso de la clase PDO~~

~~Descubriendo la clase PDOStatement~~

~~Consultas de selección a MySQL utilizando PDO~~

Insertar, modificar y borrar filas en MySQL

Formas de recorrer filas de una consulta

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Insertar filas a MySQL con PDO y enlazando datos nombrados

// Insertar una fila en una tabla MySQL utilizando PDO

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'INSERT INTO tipo_suscripcion';
```

```
$consulta .= ' (id, descripcion, observacion)';
```

```
$consulta .= ' VALUES ';
```

```
$consulta .= '(:id_tipo_suscripcion, :descripcion, :observacion)';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':id_tipo_suscripcion', 1, PDO::PARAM_INT);
```

```
$stmt->bindValue(':descripcion', 'Noticias de PHP', PDO::PARAM_STR);
```

```
$stmt->bindValue(':observacion', 'Noticias del lenguaje PHP', PDO::PARAM_STR);
```

```
$stmt->execute();
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Insertar filas con campos auto incrementales a MySQL con PDO y enlazando datos nombrados

// Insertar una fila en una tabla MySQL utilizando PDO

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'INSERT INTO tipo_suscripcion';
```

```
$consulta .= ' (descripcion, observacion);
```

```
$consulta .= ' VALUES ';
```

```
$consulta .= '(:descripcion, :observacion)';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':descripcion', 'Noticias de HTML', PDO::PARAM_STR);
```

```
$stmt->bindValue(':observacion', 'Noticias del lenguaje HTML', PDO::PARAM_STR);
```

```
$stmt->execute();
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Insertar filas en varias tablas con PDO

```
$fecha = date("Y-m-d H:i:s");
```

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$dbh->beginTransaction();
```

```
$consultaPersona = 'INSERT INTO persona (apellidos, nombres, correo_electronico)  
VALUES (?, ?, ?)';
```

```
$stmtPersona = $dbh->prepare($consultaPersona);
```

```
$consultaSuscripcion = 'INSERT INTO suscripcion (fecha, id_persona,  
id_tipo_suscripcion) VALUES (?, ?, ?)';
```

```
$stmtSuscrip = $dbh->prepare($consultaSuscripcion);
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Insertar filas en varias tablas con PDO (continuación ...)

```
if (!$stmtPersona->execute(array('Garcia', 'Juan', 'jgarcia@gugler.com.ar'))) {  
    $dbh->rollBack();  
    print_r($stmtPersona->errorInfo());  
} else if (!$stmtSuscrip->execute(array($fecha, $dbh->lastInsertId(), 1))) {  
    $dbh->rollBack();  
    print_r($stmtSuscrip->errorInfo());  
} else {  
    $dbh->commit();  
}
```

```
$stmtPersona = null;  
$stmtSuscripcion = null;  
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Modificar filas a MySQL con PDO y enlazando datos nombrados

// Modificar una fila en una tabla MySQL con PDO

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'UPDATE persona';
```

```
$consulta .= " SET nombres = :nombres";
```

```
$consulta .= ' WHERE id_persona = :id_persona';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':id_persona', 1, PDO::PARAM_INT);
```

```
$stmt->bindValue(':nombres', 'Juan', PDO::PARAM_STR);
```

```
$stmt->execute();
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Modificar varias filas a MySQL con PDO y enlazando datos nombrados

// Modificar una o más filas en una tabla MySQL con PDO

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'UPDATE suscripcion';
```

```
$consulta .= ' SET id_tipo_suscripcion = :nuevo_id_tipo_suscripcion';
```

```
$consulta .= ' WHERE id_tipo_suscripcion = :id_tipo_suscripcion';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':id_tipo_suscripcion', 3, PDO::PARAM_INT);
```

```
$stmt->bindValue(':nuevo_id_tipo_suscripcion', 5, PDO::PARAM_INT);
```

```
$stmt->execute();
```





Clase 8: Utilizando PDO con MySQL

Insertar, modificar y borrar filas en PHP

- Borrar filas en MySQL con PDO y enlazando datos nombrados

```
// Borrar una o más filas en una tabla MySQL
```

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$consulta = 'DELETE FROM suscripcion';
```

```
$consulta .= ' WHERE id_suscripcion > :id_suscripcion';
```

```
$stmt = $dbh->prepare($consulta);
```

```
$stmt->bindValue(':id_suscripcion', 888, PDO::PARAM_INT);
```

```
$stmt->execute();
```





Capítulo 4: Programación de base de datos

~~Repaso de la clase PDO~~

~~Descubriendo la clase PDOStatement~~

~~Consultas de selección a MySQL utilizando PDO~~

~~Insertar, modificar y borrar filas en MySQL~~

Formas de recorrer filas de una consulta

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

Formas de recorrer filas de una consulta

- Buscar y obtener registros de una consulta en un array numerado y/o asociativo

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$stmt = $dbh->prepare("SELECT * FROM tipo_suscripcion");
```

```
$stmt->execute();
```

```
while ($unaFila = $stmt->fetch())  
{  
    echo 'Identificador: ' . $unaFila[0] . '<br/>';  
    echo 'Descripción: ' . $unaFila['descripcion'] . '<br/>';  
    echo 'Observación: ' . $unaFila['observacion'] . '<br/>';  
}
```

```
$stmt = null;
```

```
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Formas de recorrer filas de una consulta

- Buscar y obtener registros de una consulta en un array de objetos

```
$dbh = new PDO($dsn, $usuario, $contrasenia);
```

```
$stmt = $dbh->prepare("SELECT * FROM tipo_suscripcion");
```

```
$stmt->execute();
```

```
while ($unaFila = $stmt->fetchObject())
```

```
{
```

```
    echo 'Identificador: ' . $unaFila->id_tipo_suscripcion . '<br/>';
```

```
    echo 'Descripci&oacute;n: ' . $unaFila->descripcion . '<br/>';
```

```
    echo 'Observaci&oacute;n: ' . $unaFila->observacion . '<br/>';
```

```
}
```

```
$stmt = null;
```

```
$dbh = null;
```





Clase 8: Utilizando PDO con MySQL

Formas de recorrer filas de una consulta

- Buscar y obtener todos los registros de una consulta en un array de objetos

```
$dbh = new PDO($dsn, $usuario, $contrasenia);  
$stmt = $dbh->prepare("SELECT * FROM tipo_suscripcion");  
$stmt->execute();
```

```
$result = $stmt->fetchAll(PDO::FETCH_OBJ);
```

```
foreach ($result as $unaFila)  
{  
    echo 'Identificador: ' . $unaFila->id_tipo_suscripcion . '<br/>';  
    echo 'Descripción: ' . $unaFila->descripcion . '<br/>';  
    echo 'Observación: ' . $unaFila->observacion . '<br/>';  
}
```

```
$stmt = null;  
$dbh = null;
```





Capítulo 4: Programación de base de datos

~~Repaso de la clase PDO~~

~~Descubriendo la clase PDOStatement~~

~~Consultas de selección a MySQL utilizando PDO~~

~~Insertar, modificar y borrar filas en MySQL~~

~~Formas de recorrer filas de una consulta~~

Conexión a múltiples bases de datos MySQL



Clase 8: Utilizando PDO con MySQL

- Múltiples conexiones a la base de datos

```
// Se realiza la primer conexión a la base de datos
$dsn = 'mysql:dbname=phpn1_db_clase7;host=127.0.0.1;port=3306';
$usuario = 'root';
$contrasenia = '';
```

```
$dbhClase7 = new PDO($dsn, $usuario, $contrasenia);
```

```
// Se realiza la segunda conexión a la base de datos
$dsn = 'mysql:dbname=phpn1_db_clase8;host=127.0.0.1;port=3306';
$usuario = 'root';
$contrasenia = '';
```

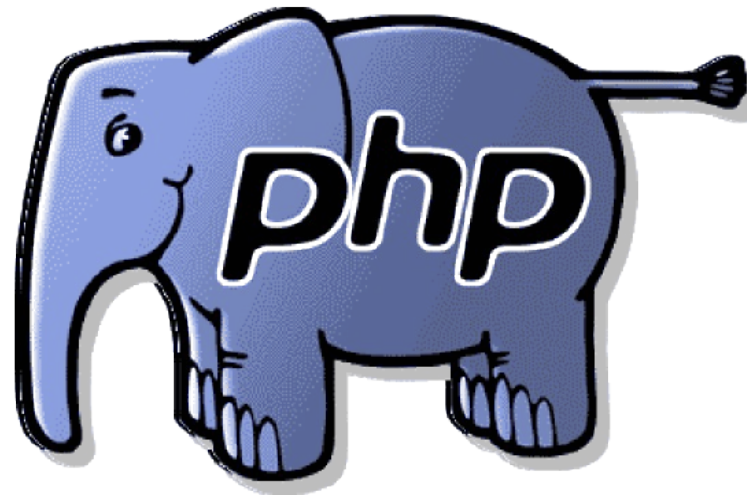
```
$dbhClase8 = new PDO($dsn, $usuario, $contrasenia);
```

```
...
```

```
$dbhClase7 = null;
$dbhClase8 = null;
```



¿Dudas?



¿Consultas?



Información de contacto

Web:

<http://www.gugler.com.ar>

<http://campusvirtual.gugler.com.ar>

<http://www.facebook.com/gugler.com.ar>

<http://www.twitter.com/cgugler>

Mail:

contacto@gugler.com.ar

academica@gugler.com.ar

administracion@gugler.com.ar



Capítulo 4: Programación de base de datos

Constantes de clase para recuperar filas de una consulta a MySQL con PDO

PDO::FETCH_ASSOC

Retorna las columnas dentro un arreglo utilizando el nombre del campo como índice.

PDO::FETCH_BOTH
(predeterminado)

Retorna las columnas dentro un arreglo utilizando el nombre del campo y números como índice.

PDO::FETCH_OBJ

Retorna las columnas dentro un objeto utilizando las propiedades del mismo.

[Volver](#)

