



PHP

PDO - Clase 2

PDO

PHP Data Object



- Métodos
- Binding de parámetros
- Transacciones
- Try / Catch



Métodos

PDO tiene muchos métodos útiles para interactuar con la db.

```
<?php
```

```
    $db = new PDO($host, $db_user, $db_pass);  
    $stmt = $db->prepare('SELECT * FROM movies);  
    $stmt->execute();
```

```
    // obtener el número de filas  
    $rows = $stmt->rowCount();
```

```
    // id de la última fila insertada  
    $lastId = $db->lastInsertId();
```

```
?>
```



Binding de parámetros

Bindear es asociar un alias dentro de nuestro query a una estructura que nos permita reutilizar ese alias, cambiando su valor en cada ejecución de la consulta.

Binding (por array)

Bindea cada parámetro. En este caso **id** y **title**.

```
<?php
    $stmt = $db->prepare("SELECT * FROM movies WHERE
    id=? AND title=?");

    $stmt->execute([
        1,
        "Avatar"
]);

    $results = $stmt->fetch(PDO::FETCH_ASSOC);
```

Binding (por valor)

Podemos especificar el tipo de cada parámetro enviado.

```
<?php
    $stmt = $db->prepare("SELECT * FROM movies WHERE id =
    ? AND title = ?");

    $stmt->bindValue(1, 10, PDO::PARAM_INT);
    $stmt->bindValue(2, "El rey león", PDO::PARAM_STR);

    $stmt->execute();

    $results = $stmt->fetch(PDO::FETCH_ASSOC);
```

Binding (por valor)

Podemos especificar el tipo de cada parámetro enviado.

```
<?php
    $stmt = $db->prepare("SELECT * FROM movies WHERE
    title LIKE ?");

    $stmt->bindValue(1, "%guerra%", PDO::PARAM_STR);

    $stmt->execute();

    $results = $stmt->fetch(PDO::FETCH_ASSOC);
```

Binding (por valor)

Podemos especificar el tipo de cada parámetro enviado.

```
<?php
    $stmt = $db->prepare("SELECT * FROM movies WHERE
    id=? AND title=?");

    $id = 1;
    $titulo = "Avatar";
    $stmt->bindValue(1, $id, PDO::PARAM_INT);
    $stmt->bindValue(2, $titulo, PDO::PARAM_STR);

    $stmt->execute();

    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
```


Placeholders

Para evitar el **?** confuso, podemos utilizar placeholders:

```
<?php
    $stmt = $db->prepare("SELECT * FROM movies WHERE
    id=:id AND title=:titulo");

    $id = 1;
    $titulo = "Avatar";
    $stmt->bindValue(':id', $id, PDO::PARAM_INT);
    $stmt->bindValue(':titulo', $titulo, PDO::PARAM_STR);

    $stmt->execute();

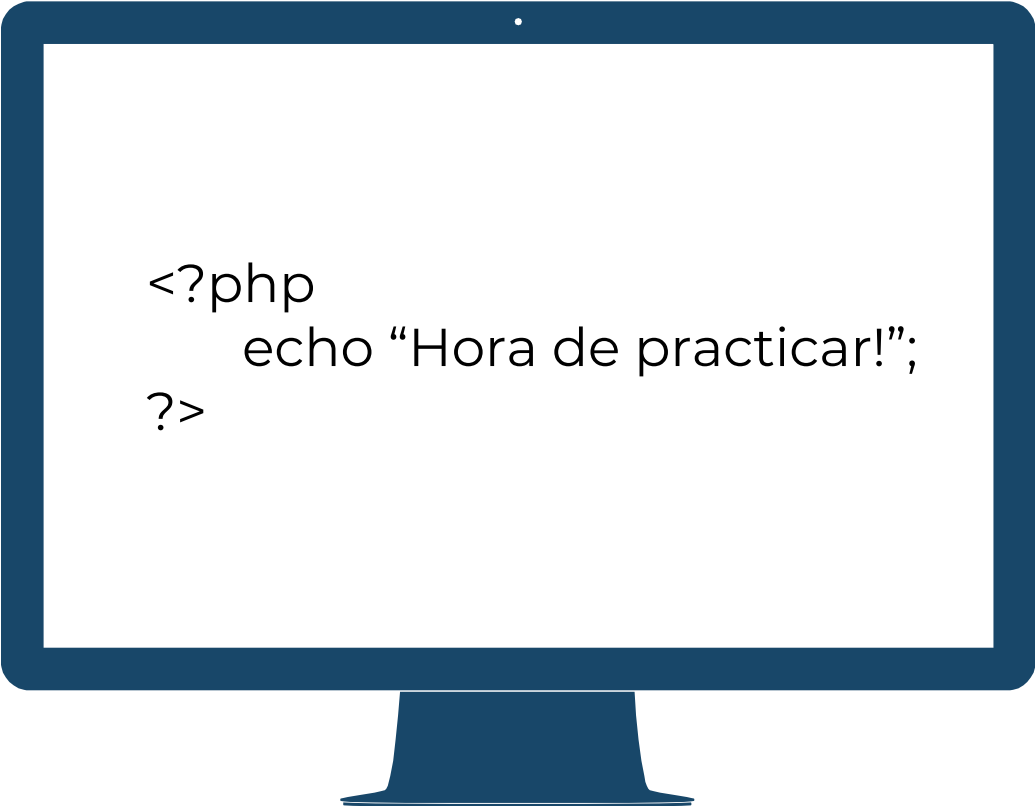
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Binding (por referencia)

En este caso la variable es **pasada por referencia** no por valor, será evaluada en el momento del **execute()**.

```
<?php
$stmt = $db->prepare("SELECT * FROM movies WHERE id=:id
AND title=:title");
$id = 1;
$title = "Avatar";
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->bindParam(':title', $title, PDO::PARAM_STR);
$title = "Buscando a Nemo"; //Se termina insertando este
valor
$stmt->execute();
$results = $stmt->fetch(PDO::FETCH_ASSOC);
```

¡A practicar!



```
<?php  
    echo "Hora de practicar!";  
?>
```



Transacciones

Una vez realizada una conexión a través de PDO, es necesario comprender cómo PDO gestiona las transacciones antes de comenzar a realizar consultas.

Una transacción ayuda a cumplir con:

- Atomicidad.
- Consistencia.
- Aislamiento.
- Durabilidad.

(ACID por sus siglas en inglés).



Transacciones

Garantiza que cualquier trabajo llevado a cabo en una transacción, incluso si se hace por etapas, sea aplicado a la base de datos de forma segura, y **sin interferencia de otras conexiones**, cuando sea consignado.

Mejora la eficiencia de las actualizaciones.



Transacciones

Una transacción se implementa en 2 pasos:

1. `beginTransaction`
 2. `commit` o `rollBack`
- 



beginTransaction

```
<?php
    $db = new PDO($host, $db_user, $db_pass);

    // desactiva el modo auto-commit
    $db->beginTransaction();

    //código a ejecutar en la base de datos
    $stmt = $db->exec(" DELETE FROM movies
    WHERE title = 'Avatar' ");

    $db->commit();
```



exec

Ejecuta una sentencia SQL y devuelve el número de filas afectadas por la sentencia.



```
$stmt = $db->exec("DELETE FROM movies WHERE  
title = 'Avatar'");
```





commit

```
<?php
    $db = new PDO($host, $db_user, $db_pass);

    // desactiva el modo auto-commit
    $db->beginTransaction();

    $stmt = $db->exec("DELETE FROM movies WHERE title
    = 'Avatar'");

    // commit de la transacción
    $db->commit();

    // Desde aquí ya se habilita el auto-commit
    nuevamente
```





rollBack

```
<?php
    $db = new PDO($host, $db_user, $db_pass);
    $db->beginTransaction();

    $stmt = $db->exec("DELETE FROM movies
    WHERE title = 'Avatar'");

    // hace roll back
    $db->rollBack();

?>
```



Con manejo de errores

```
<?php
    $db->beginTransaction();

    try {
        $db->exec("DELETE FROM movies WHERE
title = 'Avatar'");
        $db->commit();
    }
    catch(PDOException $Exception) {
        // Houston tenemos un problema
        $db->rollBack();
        echo $Exception->getMessage();
    }
```

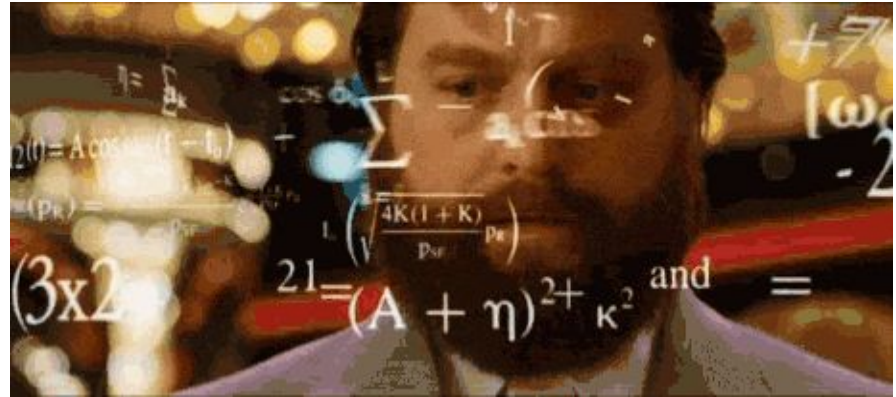
¡A practicar!



```
<?php  
    echo "Hora de practicar!";  
?>
```



¡Gracias!



¿Preguntas?