





PDO

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





Métodos

PDO tiene muchos métodos útiles para interactuar con la db.

```
<?php
```

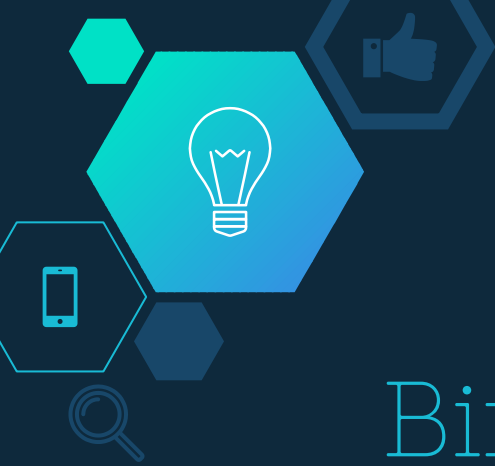
```
    $db = new PDO($host, $db_user, $db_pass);  
    $stmt = $db->query('SELECT * FROM peliculas');
```

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

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

```
?>
```





Bindear

Bindear es ligar o referenciar a otro símbolo más largo y complicado, y que se usa frecuentemente.





Binding de array

Bindea cada parámetro en este caso \$id y \$name como strings.

```
<?php
    $stmt = $db->prepare("SELECT * FROM peliculas
    WHERE id=? AND titulo=?");
    $id = 1;
    $titulo = "Avatar";
    $stmt->execute(array($id, $titulo));
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>
```



Binding de valores

Podemos especificar el tipo de cada parámetro enviado.

```
<?php
    $stmt = $db->prepare("SELECT * FROM peliculas
    WHERE id=? AND titulo=?");
    $id = 1;
    $titulo = "Avatar";
    $stmt->bindValue(1, $id, PDO::PARAM_INT);
    $stmt->bindValue(2, $titulo, PDO::PARAM_STR);
    $stmt->execute();
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>
```



Placeholders

Para evitar el "?" confuso podemos utilizar;

```
<?php
    $stmt = $db->prepare("SELECT * FROM peliculas
    WHERE id=:id AND titulo=:titulo");
    $id = 1;
    $titulo = "Avatar";
    $stmt->bindValue(':id', $id, PDO::PARAM_INT);
    $stmt->bindValue(':titulo', $titulo,
    PDO::PARAM_STR);
    $stmt->execute();
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>
```

Binding con parámetros

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 peliculas  
    WHERE id=:id AND titulo=:titulo");
```

```
    $id = 1;
```

```
    $titulo = "Avatar";
```

```
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);
```

```
    $stmt->bindParam(':titulo', $titulo,  
    PDO::PARAM_STR);
```

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

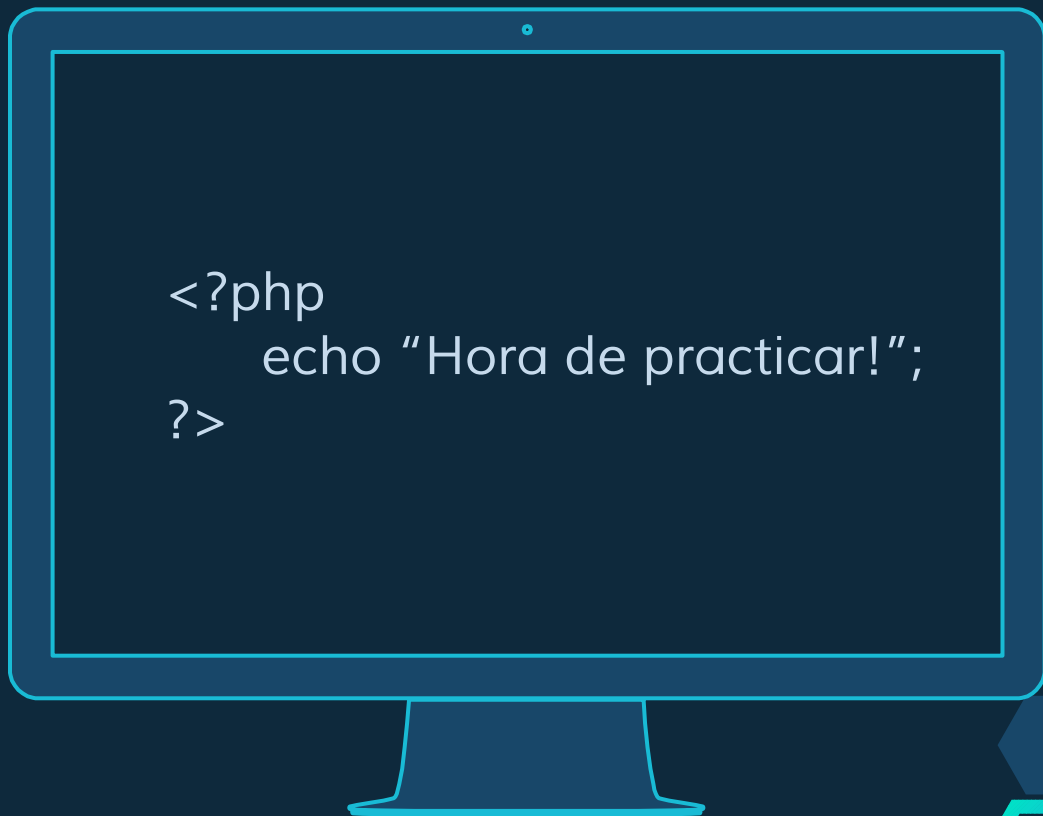
```
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

```
?>
```




¡A practicar!

Hasta transacciones





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 cumple:

- 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.



A decorative graphic on the left side of the slide consists of a cluster of hexagons in various shades of blue and cyan. Some hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. A large, solid cyan hexagon is positioned in the center of this cluster.

Transacciones

Una transacción se implementa en 3 pasos:

- `beginTransaction`
- `commit`
- `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  
    $sth = $db->exec("DELETE FROM peliculas  
    WHERE titulo = 'Avatar'");
```

```
    $db->commit();
```

```
?>
```



exec

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

```
$sth = $db->exec("DELETE FROM peliculas WHERE  
titulo = 'Avatar'");
```



commit

```
<?php
```

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

```
    $sth = $db->exec("DELETE FROM peliculas  
    WHERE titulo = 'Avatar'");
```

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

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

```
?>
```



rollBack

```
<?php
```

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

```
    $sth = $db->exec("DELETE FROM peliculas  
    WHERE titulo = 'Avatar'");
```

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

```
?>
```

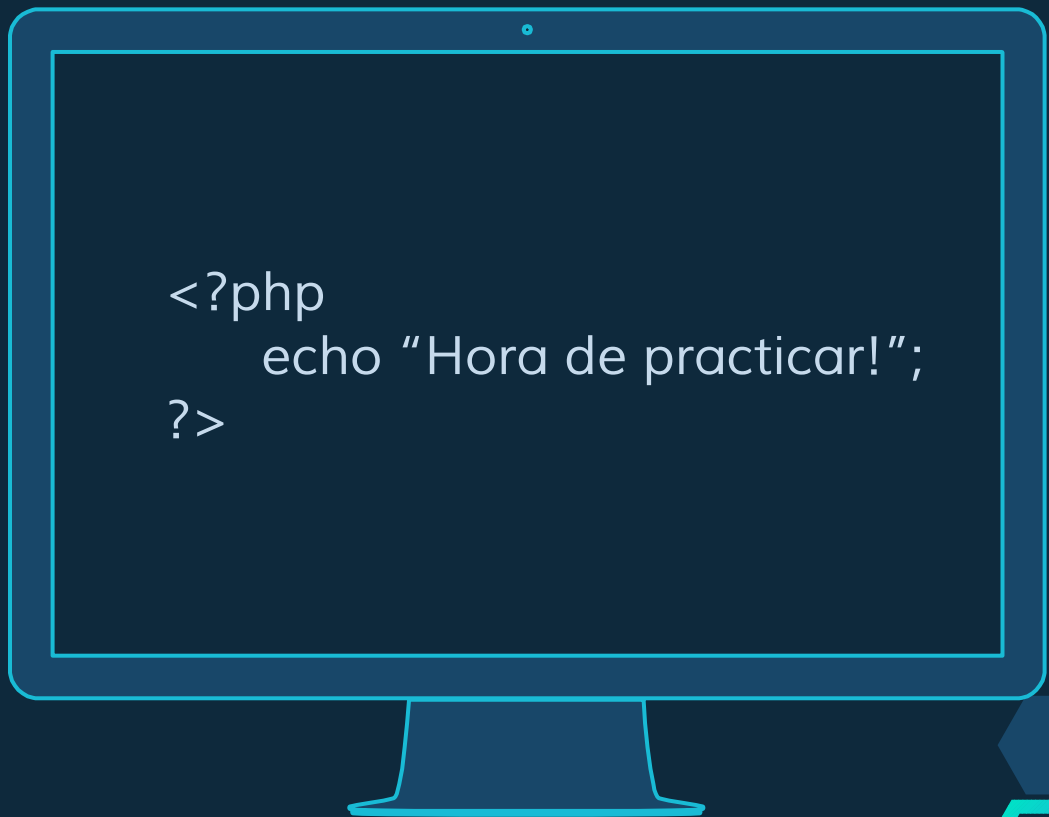

Con manejo de errores

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

        $db->exec("DELETE FROM peliculas WHERE
titulo = 'Avatar'");
        $db->commit();
    }
    catch(PDOException $ex) {
        // Houston tenemos un problema
        $db->rollBack();
        echo $ex->getMessage();
    }
?>
```



¡A practicar!



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





Gracias!

¿Preguntas?

