**PHP Data Objects (PDO)**

PDO es la capa de abstracción de objetos para poder manipular los datos de las bases de datos de forma estándar.

SGBD: tenemos dos categorías : gestores SQL y gestores NO SQL

BD relacionales: Oracle, PostgreSql, Mysql

BD no relacionales: MongoDB

Con PDO podemos utilizar cualquier controlador Mysql, MongoDB, etc

PHP ofrece dos API para comunicarnos con el SGBD Mysql: PDO,MYSQLI

# **La clase PDO**

Representa una conexión entre PHP y un servidor de bases de datos.

**PDO** {

public [\_\_construct](https://www.php.net/manual/es/pdo.construct.php) ( string $dsn [, string $username [, string $password [, array $options ]]] )

public [beginTransaction](https://www.php.net/manual/es/pdo.begintransaction.php) ( void ) : bool

public [commit](https://www.php.net/manual/es/pdo.commit.php) ( void ) : bool

public [errorCode](https://www.php.net/manual/es/pdo.errorcode.php) ( void ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public [errorInfo](https://www.php.net/manual/es/pdo.errorinfo.php) ( void ) : array

public [exec](https://www.php.net/manual/es/pdo.exec.php) ( string $statement ) : int

public [getAttribute](https://www.php.net/manual/es/pdo.getattribute.php) ( int $attribute ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public static [getAvailableDrivers](https://www.php.net/manual/es/pdo.getavailabledrivers.php) ( void ) : array

public [inTransaction](https://www.php.net/manual/es/pdo.intransaction.php) ( void ) : bool

public [lastInsertId](https://www.php.net/manual/es/pdo.lastinsertid.php) ([ string $name = **NULL** ] ) : string

public [prepare](https://www.php.net/manual/es/pdo.prepare.php) ( string $statement [, array $driver\_options = array() ] ) : [PDOStatement](https://www.php.net/manual/es/class.pdostatement.php)

public [query](https://www.php.net/manual/es/pdo.query.php) ( string $statement ) : [PDOStatement](https://www.php.net/manual/es/class.pdostatement.php)

public [quote](https://www.php.net/manual/es/pdo.quote.php) ( string $string [, int $parameter\_type = PDO::PARAM\_STR ] ) : string

public [rollBack](https://www.php.net/manual/es/pdo.rollback.php) ( void ) : bool

public [setAttribute](https://www.php.net/manual/es/pdo.setattribute.php) ( int $attribute , [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed) $value ) : bool }

# **La clase PDOStatement**

Representa una sentencia preparada y, después de la ejecución de la instrucción, un conjunto de resultados asociados.

/\* Métodos \*/

public [bindColumn](https://www.php.net/manual/es/pdostatement.bindcolumn.php) ( [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $column , [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) &$param [, int $type [, int $maxlen [, [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $driverdata ]]] ) : bool

public [bindParam](https://www.php.net/manual/es/pdostatement.bindparam.php) ( [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $parameter , [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) &$variable [, int $data\_type = PDO::PARAM\_STR [, int $length [, [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed) $driver\_options ]]] ) : bool

public [bindValue](https://www.php.net/manual/es/pdostatement.bindvalue.php) ( [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $parameter , [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $value [, int $data\_type = PDO::PARAM\_STR ] ) : bool

public [closeCursor](https://www.php.net/manual/es/pdostatement.closecursor.php) ( void ) : bool

public [columnCount](https://www.php.net/manual/es/pdostatement.columncount.php) ( void ) : int

public [debugDumpParams](https://www.php.net/manual/es/pdostatement.debugdumpparams.php) ( void ) : void

public [errorCode](https://www.php.net/manual/es/pdostatement.errorcode.php) ( void ) : string

public [errorInfo](https://www.php.net/manual/es/pdostatement.errorinfo.php) ( void ) : array

public [execute](https://www.php.net/manual/es/pdostatement.execute.php) ([ array $input\_parameters ] ) : bool

public [fetch](https://www.php.net/manual/es/pdostatement.fetch.php) ([ int $fetch\_style [, int $cursor\_orientation = PDO::FETCH\_ORI\_NEXT [, int $cursor\_offset = 0 ]]] ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public [fetchAll](https://www.php.net/manual/es/pdostatement.fetchall.php) ([ int $fetch\_style [, [mixed](https://www.php.net/manual/es/language.pseudo-types.php#language.types.mixed) $fetch\_argument [, array $ctor\_args = array() ]]] ) : array

public [fetchColumn](https://www.php.net/manual/es/pdostatement.fetchcolumn.php) ([ int $column\_number = 0 ] ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public [fetchObject](https://www.php.net/manual/es/pdostatement.fetchobject.php) ([ string $class\_name = "stdClass" [, array $ctor\_args ]] ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public [getAttribute](https://www.php.net/manual/es/pdostatement.getattribute.php) ( int $attribute ) : [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed)

public [getColumnMeta](https://www.php.net/manual/es/pdostatement.getcolumnmeta.php) ( int $column ) : array

public [nextRowset](https://www.php.net/manual/es/pdostatement.nextrowset.php) ( void ) : bool

public [rowCount](https://www.php.net/manual/es/pdostatement.rowcount.php) ( void ) : int

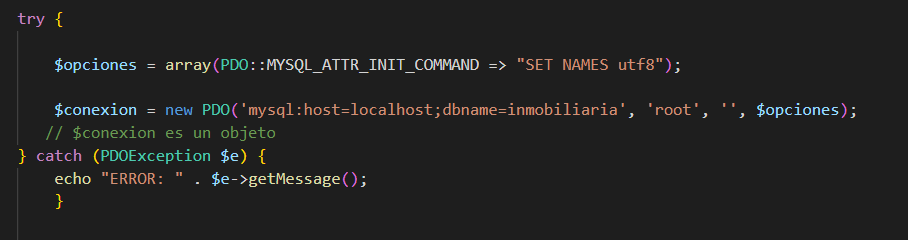
public [setAttribute](https://www.php.net/manual/es/pdostatement.setattribute.php) ( int $attribute , [mixed](https://www.php.net/manual/es/language.pseudo-types.php" \l "language.types.mixed) $value ) : bool

public [setFetchMode](https://www.php.net/manual/es/pdostatement.setfetchmode.php) ( int $mode ) : bool

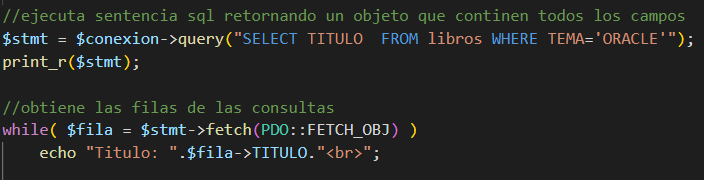
}

Vamos a realizar un ejercicio utilizando la B.D. Inmobiliaria

Declaramos una instancia de un objeto de la clase **PDO** para manejar la conexión. Al objeto debemos pasarle los parámetros de la conexión:



### Consulta de datos con PDO

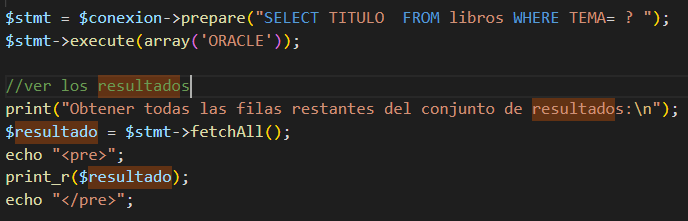


Utilizamos sentencias preparadas, para evitar inyección sql.

En lugar de concatenar datos de usuario directamente a la consulta, se utiliza un *marcador de posición en* su lugar. Los datos se envían por separado, lo que significa que no hay posibilidad de que el motor de SQL confunda los datos del usuario para un conjunto de instrucciones

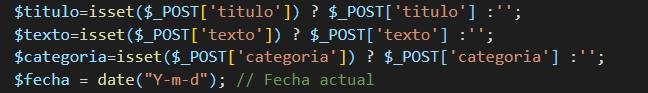
Una **instancia de PDOStatement** se crea cuando se llama a **PDO->prepare()**, y con ese objeto creado se llama a métodos como bindParam() para pasar valores o execute() para ejecutar sentencias

public **PDOStatement::execute** ([ array $input\_parameters ] ) : bool

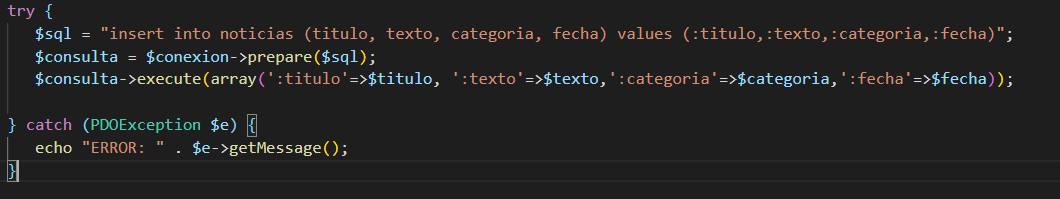


### Insertar registros en un B.D.

1. Recojo datos del formulario



1. Inserto en la B.D



1. Visualizo último rgtro. insertado



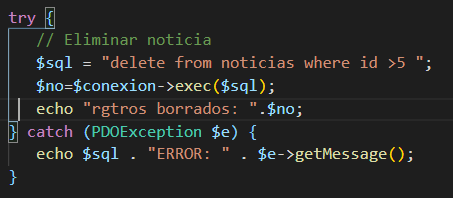
### Borrar registros en un B.D.

PDO:exec (Ejecuta una sentencia SQL y devuelve el número de filas afectadas)

public **PDO::exec** ( string $statement ) : int

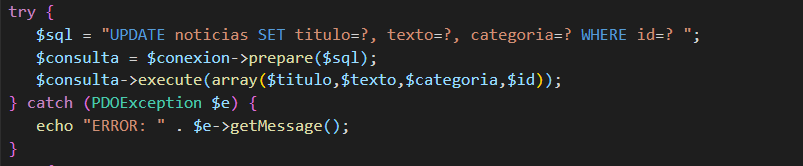
PDOStatement::execute  (Ejecuta una sentencia preparada)

public **PDOStatement::execute** ([ array $input\_parameters ] ) : bool



### Actualizar un registro en una B.D.

### Los datos que voy a actualizar los recogeré de un formulario



Otra opción mas simplificada

