

Al conectar con la base de datos, podemos especificar que el tratamiento de errores se produzca de una de las siguientes maneras:

`PDO::ERRMODE_SILENT`: Similar a los modos antiguos de conexión, hay que comprobar lo devuelto en cada consulta para ver si es `===false`.

`PDO::ERRMODE_WARNING`: En caso de error se lanza un error PHP de tipo Warning

`PDO::ERRMODE_EXCEPTION`: En caso de error se lanza una excepción.

Con este último modo, las funciones de conexión o consulta, en lugar de devolver verdadero o falso en caso de error, lanzan una excepción que puede ser capturada mediante instrucciones **try-catch**.

```
try {
    $db->query('sql invalido');
} catch(PDOException $ex) {
    echo "Ocurrió un error<br>"; //user friendly message
    echo $ex->getMessage();
}
```

El código dentro de la sentencia **try** será ejecutado con normalidad, pero si en algún momento se dispara una excepción (error) de clase `PDOException`, el control del programa pasará al bloque **catch**, almacenando en `$ex` el objeto excepción con metainformación del error.

En cualquier caso, podemos ignorar totalmente el tratamiento de excepciones. En este caso, al producirse alguna ocurrirá un error general de “excepción no capturada”.

Este último modo, aunque es algo más complejo de entender, permite realizar un tratamiento de errores más complejo, nos permite no tener funciones que devuelven resultados de más de un tipo a la vez (booleano `false`, o un array con el resultado), y es el modo estándar de tratamiento de errores que se usa en otros lenguajes fuertemente orientados a objetos como Java o C#.