

```

1  <?php
2  class Dbase{
3      private $host = DB_HOST;
4      private $user = DB_USER;
5      private $password = DB_PASSWORD;
6      private $bdatos = DB_BASE;
7
8      //para las consultas
9      private $dbh;
10     private $stmt;    //querys
11     private $error;
12
13     public function __construct() {
14         $dsn = "mysql:host=" . $this->host . ";dbname=" . $this->bdatos;
15         $opciones = array(
16             PDO::ATTR_PERSISTENT => true,
17             PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION);
18         //instanciamos PDO
19         try {
20             $this->dbh = new PDO($dsn, $this->user, $this->password, $opciones);
21             $this->dbh->exec('set names utf8');
22         } catch (PDOException $e) {
23             $this->error = $e->getMessage();
24             echo $this->error;
25         }
26     }
27
28     /**
29     * Esta funcion prepara el sql para ejecutar con el PDO y su sentencia PREPARE *
30     * @param string $SQL la consulta sql
31     * @return void el sql sanitizado
32     * @version 1.0 enero del 2019
33     * @author Juanito
34     */
35     public function query($sql) {
36         $this->stmt = $this->dbh->prepare($sql);
37     }
38
39     //parametrizacion de la consulta Bind de Pdo :)
40     public function bind($parametro, $valor, $tipo = null) {
41         if (is_null($tipo)) {
42             switch (true) {
43                 case is_int($valor):
44                     $tipo = PDO::PARAM_INT;
45                     break;
46                 case is_bool($valor):
47                     $tipo = PDO::PARAM_BOOL;
48                     break;
49                 case is_null($valor):
50                     $tipo = PDO::PARAM_NULL;
51                     break;
52                 default:
53                     $tipo = PDO::PARAM_STR;
54                     break;
55             }
56         }
57         $this->stmt->bindValue($parametro, $valor, $tipo);
58     }
59     //ejecuta la consulta
60     public function execute() {
61         return $this->stmt->execute();
62     }
63
64     //obtener todos los registros
65     public function registros(){
66         $this->execute();
67         return $this->stmt->fetchAll(PDO::FETCH_OBJ);
68     }
69     //obtener UN SOLO registro
70     public function registro() {
71         $this->execute();
72         return $this->stmt->fetch(PDO::FETCH_OBJ);

```

```
73     }
74     //Cantidad de filas --- para paginacion y otras cosas
75     public function totalFilas() {
76         $this->execute();
77         return $this->stmt->rowCount();
78     }
79
80 }
81
82 ?>
```