

## TAREA RECUPERACIÓN PDO+CRUD 1(13/04/2021)

- Primero debemos de crear las carpetas necesarias en nuestra raíz del proyecto, como son las carpetas public, sql y clases
- Creamos el proyecto con el composer

```
Juan José@DESKTOP-91UP6BN c:\xampp\htdocs\php\repasomayo\trabajo1crud
# composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [juan jos/trabajo1crud]:
```

```
This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [juan jos/trabajo1crud]: juanjose/trabajo1crud
Description []: Primer trabajo recuperacion pdo
Author [, n to skip]: juanjoseparralopez
Invalid author string. Must be in the format: John Smith <john@example.com>
Author [, n to skip]: juanjoseparralopez@gmail.com
Invalid author string. Must be in the format: John Smith <john@example.com>
Author [, n to skip]: n
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []: GNU

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? yes
Search for a package:
Would you like to define your dev dependencies (require-dev) interactively [yes]? yes
Search for a package:

{
    "name": "juanjose/trabajo1crud",
    "description": "Primer trabajo recuperacion pdo",
    "type": "project",
    "license": "GNU",
    "require": {}
}

Do you confirm generation [yes]?
```

Ahora debemos de añadirle al composer.json el autoload y se los debe de quedar asi:

```
composer.json X
composer.json > {} require
1  {
2      "name": "juanjose/trabajo1crud",
3      "description": "Primer trabajo recuperacion pdo",
4      "type": "project",
5      "license": "GNU",
6      "require": {},
7      "config": {
8          "optimize-autoloader": true
9      },
10     "autoload": {
11         "psr-4": {
12             "Clases\\": "clases"
13         }
14     }
15 }
```

Para terminar con el composer.json debemos de hacer mediante el terminal un composer require fakerphp/faker para poder usar el faker en nuestro proyecto.

```
composer.json X
composer.json > ...
1  {
2      "name": "juanjose/trabajo1crud",
3      "description": "Primer trabajo recuperacion pdo",
4      "type": "project",
5      "license": "GNU",
6      "require": {
7          "fakerphp/faker": "^1.14"
8      },
9      "config": {
10         "optimize-autoloader": true
11     },
12     "autoload": {
13         "psr-4": {
14             "Clases\\": "clases"
15         }
16     }
17 }
```

Ahora nos disponemos a crear la base de datos y a darle permisos al usuario creado en clase.

```
Juan José@DESKTOP-91UP6BN C:\xampp\htdocs\php\repasomayo\trabajo1crud
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.17-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database trabajo1;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> grant all on repaso1.* to urepaso@'localhost';
Query OK, 0 rows affected (0.004 sec)
```

Creamos el .config con los datos usados anteriormente. Así debe de quedar

```
.config
1  ;parametros de conexion
2  [opciones]
3  host=localhost
4  bbdd=trabajo1
5  usuario=urepaso
6  pass=secret0
```

Ahora creamos dentro de la carpeta clases el archivo Conexión.php

```
clases > Conexion.php > ...
1  <?php
2  namespace Clases;
3  use PDO;
4  use PDOException;
5
6  class Conexion{
7      protected static $conexion;
8
9      public function __construct()
10     {
11         if(self::$conexion==null){
12             self::crearConexion();
13         }
14     }
15     public static function crearConexion(){
16         $opciones=parse_ini_file('../.config');
17         $mibase=$opciones["bbdd"];
18         $miUser=$opciones["usuario"];
19         $miPass=$opciones["pass"];
20         $miHost=$opciones["host"];
21         $dns="mysql:host=$miHost;dbname=$mibase;charset=utf8mb4";
22         try {
23             self::$conexion=new PDO($dns,$miUser,$miPass);
24             self::$conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
25         } catch (PDOException $ex) {
26             die("Error al conectar a la BBDD,mensaje".$ex->getMessage());
27         }
28     }
29 }
```

Ahora crearemos dentro de la misma carpeta el archivo Datos.php donde usaremos el faker para guardar datos aleatorios dentro de nuestra base de datos

```

clases > Datos.php > Datos > crearUsuarios
1  <?php
2
3  namespace Clases;
4  require '../vendor/autoload.php';
5  use Faker\Factory;
6  use Clases\Users;
7  class Datos{
8      public $faker;
9
10     public function __construct($tabla,$cantidad)
11     {
12         $this->faker=Factory::create('es_ES');
13         switch ($tabla) {
14             case 'users':
15                 $this->crearUsuarios($cantidad);
16                 break;
17         }
18     }
19
20     public function crearUsuarios($n){
21         //creamos un usuario admin
22         $usuario = new Users();
23         $usuario->borrarTodo();
24         $usuario->setNombre("Paco");
25         $usuario->setApellidos("Perez Gil");
26         $usuario->setUsername("admin");
27         $usuario->setMail("admin@mail.com");
28         $pass=hash('sha256', "secret0");
29         $usuario->setPass($pass);
30         $usuario->create();
31         //creamos el resto
32         for ($i=0; $i < $n; $i++) {
33             $usuario->setNombre($this->faker->firstName());
34             $usuario->setApellidos($this->faker->lastName(). " ". $this->faker->lastName());
35             $usuario->setUsername($this->faker->unique()->userName());
36
37             $usuario->create();
38         }

```

Ahora debemos de crear el archivo Users.php que hereda de conexión y es donde haremos los métodos con las consultas a la base de datos.

```
Users.php X crearDatos.php
classes > Users.php > Users
1 <?php
2 namespace Clases;
3 use Clases\Conexion;
4 use PDOException;
5 use PDO;
6
7 class Users extends Conexion{
8     private $id;
9     private $nombre;
10    private $apellidos;
11    private $username;
12    private $mail;
13    private $pass;
14
15
16    public function __construct()
17    {
18        parent::__construct();
19    }
20
21    //CRUD-----
22    public function create(){
23        $con="insert into users('nombre','apellidos','username','mail','pass') values(:n,:a,:u,:m,:p)";
24        $stmt=parent::$conexion->prepare($con);
25        try {
26            $stmt->execute([
27                'n'=>$this->nombre,
28                'a'=>$this->apellidos,
29                'u'=>$this->username,
30                'm'=>$this->mail,
31                'p'=>$this->pass,
32            ]);
33        } catch (PDOException $ex) {
34            die("Error al insertar el usuario".$ex->getMessage());
35        }
36    }
37
38    public function borrarTodo(){
39        $con="delete from users";
40        $stmt=parent::$conexion->prepare($con);
41        try {
42            $stmt->execute();
43        } catch (PDOException $ex) {
44            die("Error al borrar TODO".$ex->getMessage());
45        }
46    }
47
```

```
public function validarUsuario(string $nombre, string $pass) : bool{
    $con="select * from users where username=:u AND pass=:p";
    $stmt=parent::$conexion->prepare($con);
    try{
        $stmt->execute([
            'u'=>$nombre,
            'p'=>$pass,
        ]);
    }catch(PDOException $ex){
        die("Error al validar usuario".$ex->getMessage());
    }
    $fila=$stmt->fetch(PDO::FETCH_OBJ);

    return ($fila!=null)? true: false;
}

public function getNombre()
{
    return $this->nombre;
}
```

Mas los getters y setters.

Después creamos dentro de carpeta public crearDatos.php para poder ejecutar el archivo Datos.php y que el faker genere aleatorios los 50 usuarios.

```
public > crearDatos.php > ...
1  <?php
2
3  require '../vendor/autoload.php';
4
5  use Clases\Datos;
6
7  $usu = new Datos('users', 50);
8  echo "<h2>Usuarios Creados</h2>";
9  |
```

Por último creamos el login.php para poder comprobar si esta correctamente creados los datos y el login funciona correctamente

```
1  <?php
2      session_start();
3      require "../vendor/autoload.php";
4      require "../clases/Users.php";
5      use Clases\Users;
6      function mostrarError($texto){
7          $_SESSION['error']=$texto;
8          header("Location:{"$_SERVER['PHP_SELF']}");
9      }
10
11     if(isset($_POST['login'])){
12         $nombre=trim($_POST['nombre']);
13         $pass=trim($_POST['pass']);
14         if(strlen($nombre)==0 || strlen($pass)==0){
15             mostrarError("Rellene los campos!!!!");
16         }
17         $passBuena=hash('sha256',$pass);
18         $usuario=new Users();
19         if($usuario->validarUsuario($nombre, $passBuena)){
20             $usuario=null;
21             die("Me he validado con exito.");
22         }else{
23             $usuario=null;
24             mostrarError("Nombre o pass incorrectos");
25         }
26     }else{
27
28     ?>
```

```

Users.php  crearDatos.php  Datos.php  login.php X
public > login.php > mostrarError
29 <!DOCTYPE html>
30 <html lang="es">
31 <head>
32   <meta charset="UTF-8">
33   <meta http-equiv="X-UA-Compatible" content="IE=edge">
34   <meta name="viewport" content="width=device-width, initial-scale=1.0">
35   <!-- CSS only -->
36   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e0JMYsd
37   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" integrity="sha512-1BBXm8fW90
38   <title>Login</title>
39 </head>
40 <body style="background-color: bisque;">
41 <h3 class="text-center mt-3">Posts Almería S.A.</h3>
42
43 <div class="container mt-3 mb-4">
44   <?php
45     if (isset($_SESSION['error'])) {
46       echo "<p class='my-3 p-3 bg-dark text-danger font-weight-bold'>{$_SESSION['error']}</p>";
47       unset($_SESSION['error']);
48     }
49   >
50   <div class="card text-white bg-secondary mb-3 m-auto mt-5" style="max-width: 48rem;">
51     <div class="card-header text-center">Login</div>
52     <div class="card-body">
53       <form name="login" action="<?php echo $_SERVER['PHP_SELF'] >" method="POST">
54         <div class="form-group">
55           <label for="nu">Nombre de Usuario</label>
56           <input type="text" class="form-control" id="nu" placeholder="Ingresa su nombre" name="nombre" required>
57         </div>
58
59         <div class="form-group mt-3">
60           <label for="np">Password</label>
61           <input type="password" class="form-control" id="np" placeholder="Password" name="pass" required>
62         </div>
63
64         <div class="mt-3">
65           <button type="submit" class="btn btn-primary" name="login"><i class="fas fa-sign-in-alt mr-2"></i>Login</button>
66         </div>
67       </form>
68     </div>
69   </div>
70 </div>
71 </body>
72 </html>
73 <?php
74   }
75 >

```