



TAREA DE RECUPERACIÓN

Pablo López Roda 2ºDAW

Utilizando la tabla "user" del proyecto comenzado en clase, haremos lo siguiente:

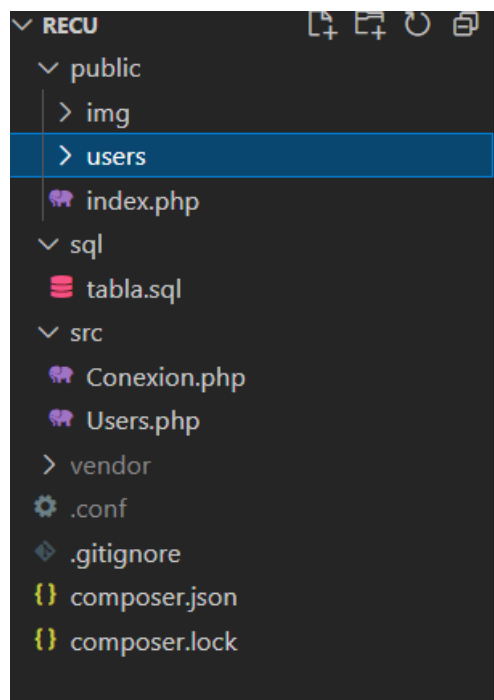
1. Crearemos una estructura de proyecto con la carpetas usuales (public, src o clases, sql)

Con el comando: `composer init` se crea el `src` y `composer`.

Con el comando: `composer install` creamos el `composer.lock` y le añadimos el Faker con el comando copiado en el `packagist`

Creamos la carpeta `public` con su `index.php` y sus carpetas `img` donde introduciremos las imágenes de los perfiles y la carpeta `users` donde introduciremos todos los elementos del CRUD.

También la carpeta `sql` donde introduciremos la tabla `users`



2. Crearemos un archivo de configuración con los parámetros de conexión

```
Conexion.php X buser.php euser.php index.php register.php Users.php
src > Conexion.php > Conexion > crearconexion
1  <?php
2  namespace Recu;
3  use PDO;
4  use PDOException;
5
6  class Conexion{
7      protected static $conexion;
8
9      public function __construct(){
10         if(self::$conexion==null){
11             self::crearconexion();
12         }
13     }
14
15     public static function crearconexion(){
16         $fichero=dirname(__DIR__,1)."/.conf";
17         $opciones=parse_ini_file($fichero);
18         $dbname=$opciones['dbname'];
19         $host=$opciones['host'];
20         $usuario=$opciones['user'];
21         $pass=$opciones['pass'];
22         $dns="mysql:host=$host;dbname=$dbname;charset=utf8mb4";
23         try{
24             self::$conexion=new PDO($dns, $usuario, $pass);
25             self::$conexion->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
26         }
27         catch(PDOException $ex){
28             die("Error en la conexion: ".$ex->getMessage());
29         }
30     }
31 }
```

3. Utilizaremos autocarga de clases y **Faker** implementado con **composer**

```
composer.json > ...
1  {
2      "name": "pablo/recu",
3      "description": "Recuperacion",
4      "type": "project",
5      "license": "Free",
6      "autoload": {
7          "psr-4": {
8              "Recu\\": "src/"
9          }
10     },
11     "authors": [
12         {
13             "name": "Pablo Lopez Roda",
14             "email": "pablolr45@gmail.com"
15         }
16     ],
17     "require": {
18         "fakerphp/faker": "^1.18"
19     }
20 }
21
```

- Meteremos 50 usuarios de prueba, el primero un usuario **admin** con pass conocida

En el index.php de public llamamos al método crearUsuarios creando así 50 usuarios, éste método es llamado desde la clase users de la carpeta src

```
public > index.php
1 <?php
2     require dirname(__DIR__)."/vendor/autoload.php";
3     use Recu\Users;
4     (new Users)->crearUsuarios(50);
5 ?>
6 <!DOCTYPE html>
7 <html lang="es">
8 <head>
9     <meta charset="UTF-8">
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYh
13    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.0.0-beta2/css/all.min.css" integrity="sha512-YWzhKL
14    <title>Inicio</title>
15 </head>
16 <body style="background-color:#ff9800">
17 <h3 class="text-center mt-2">Inicio</h3>
18 <div class="container mt-2">
19     <div class="d-flex justify-content-around">
20         <a href="users/login.php" class="btn btn-success">Login</a>
21         <a href="users/register.php" class="btn btn-success">Registrar</a>
22         <a href="users/index.php" class="btn btn-secondary">Acceso Invitado</a>
23     </div>
24 </div>
25 </body>
26 </html>
```

```
public function crearUsuarios($cant){
    if (!$this->hayUsuarios()){
        (new Users)->setUsername("admin")
        ->setEmail("admin@gmail.com")
        ->setPassword(hash('sha256', 'secret0'))
        ->setImg("/img/users/admin.jpg")
        ->setPerfil(1)
        ->create();
        (new Users)->setUsername("usuario")
        ->setEmail("usuario@gmail.com")
        ->setPassword(hash('sha256', 'usuario'))
        ->setImg("/img/users/users.png")
        ->setPerfil(0)
        ->create();
        $faker= Faker\Factory::create('es_ES');
        for ($i =0; $i < $cant-2; $i++){
            (new Users)->setUsername($faker->unique()->username)
            ->setEmail($faker->unique()->freeEmail)
            ->setPassword($faker->sha256())
            ->setImg("/img/users/users.png")
            ->setPerfil($faker->numberBetween(0, 1))
            ->create();
        }
    }
}

public function hayUsuarios(): bool{
    $q = "select * from users";
    $stmt = parent::$conexion->prepare($q);
    try{
        $stmt->execute();
    }catch (PDOException $ex){
        die("Erro al comprobar si hay usuarios: ".$ex->getMessage());
    }
    parent::$conexion = null;
    return ($stmt->rowCount() != 0);
}
```

5. Haremos un formulario de "login" plenamente operativo.

```
Conexion.php  buser.php  euser.php  index.php  register.php  login.php X

public > users > login.php > ...
1  <?php
2  session_start();
3  require dirname(__DIR__, 2) . "/vendor/autoload.php";
4  use Recu\Users;
5
6  $error = false;
7
8  $INTENTOS=5;
9  $TIEMPO=60;
10
11 function estaVacio($c, $v)
12 {
13     if (strlen($v) < 5) {
14         $_SESSION[$c] = "Este campo debe conterner al menos 5 carcateres";
15         return true;
16     }
17     return false;
18 }
19 if (isset($_POST['btnLogin'])) {
20     $un = trim($_POST['username']);
21     $p = trim($_POST['password']);
22
23     if (estaVacio("username", $un)) $error = true;
24     if (estaVacio("password", $p)) $error = true;
25
26     if (!$error) {
27         $pass=hash('sha256', $p);
28         if((new Users)->comprobarUsuario($un, $pass)){
29             setcookie('errorV', 1, time()-100);
30             $_SESSION['user']=$un;
31             header("Location:index.php");
32         }else{
33             if(isset($_COOKIE['errorV'])){
34                 $cont=++$_COOKIE['errorV'];
35                 if($cont==5){
36                     setcookie('errorV', $cont, time()+60);
37                 }else{
38                     setcookie('errorV', $cont, time()+60);
39                 }
38 }
```

```
src > Users.php > Users
1  <?php
2  namespace Recu;
3  use PDOException;
4  use PDO;
5  use Faker;
6
7  class Users extends Conexion{
8      private $id;
9      private $username;
10     private $email;
11     private $password;
12     private $img;
13     private $perfil;
14
15     public function __construct(){
16         parent::__construct();
17     }
18
19     public function create(){
20         $q= "insert into users(username,email, password, img, perfil) values(:u,:e,:pa,:i,:pe)";
21         $stmt=parent::$conexion->prepare($q);
22         try{
23             $stmt->execute([
24                 ':u'=>$this->username,
25                 ':e'=>$this->email,
26                 ':pa'=>$this->password,
27                 ':i'=>$this->img,
28                 ':pe'=>$this->perfil,
29             ]);
30         }catch(PDOException $ex){
31             die("Error al guardar el usuario: ".$ex->getMessage());
32         }
33     }
34
35     public function read(){
36         $q = "select * from users where id=:i";
37         $stmt = parent::$conexion->prepare($q);
38         try{
39             $stmt->execute([
40                 ':i'=>$this->id,
41             ]);
42             $result=$stmt->fetchAll(PDO::FETCH_ASSOC);
43             return $result;
44         }catch(PDOException $ex){
45             die("Error al leer el usuario: ".$ex->getMessage());
46         }
47     }
48 }
```

```

Users.php 7 Users

public function read(){
    $q = "select * from users where id=:i";
    $stmt = parent::$conexion->prepare($q);
    try{
        $stmt->execute([
            ':i' => $id
        ]);
    }catch (PDOException $ex){
        die("Error al borrar el usuario: ".$ex->getMessage());
    }
    parent::$conexion = null;
    return $stmt->fetch(PDO::FETCH_OBJ);
}

public function update(){
    $q = "update users set username=:u, email=:e, password=:pa, perfil=:pe, img=:i where id=:id";
    $stmt = parent::$conexion->prepare($q);
    try{
        $stmt->execute([
            ':u' => $this->username,
            ':e' => $this->email,
            ':pa' => $this->password,
            ':i' => $this->img,
            ':pe' => $this->perfil,
            ':id' => $id
        ]);
    }catch (PDOException $ex){
        die("Error al guardar el usuario: ".$ex->getMessage());
    }
    parent::$conexion = null;
}

public function delete(){
    $q = "delete from users where id=:i";
    $stmt = parent::$conexion->prepare($q);
    try{
        $stmt->execute([

```

```

▼ users
  🐘 buser.php
  🐘 cerrar.php
  🐘 duser.php
  🐘 euser.php
  🐘 index.php
  🐘 login.php
  🐘 register.php
  🐘 index.php
▼ sql

```