

UF2_NF2_PRAC_02: Refactorització de codi

Objectius de la pràctica:

- Entendre que és la refactorització de codi.
- Refactoritzar un projecte propi utilitzant els mecanismes de refactorització
- Documentar els passos realitzats amb captures de pantalla i les explicacions necessàries

Enunciat

Recupera el codi de la pràctica del mòdul 7, UF1_PRAC_02 Cursos Udemy. Utilitza un mínim de 5 mecanismes de refactorització (La màxima puntuació de la pràctica serà amb 10 mecanismes o més).

1. Inline Method

He quitado la variable \$connect y he puesto su resultado directamente en la función de mysqli

• Problem

```
function queryEliminarUsuario(string $email): bool {  
    // FIXME hay demasiado if  
    $connect = @connectionSQL();  
    $query = "DELETE FROM usuarios WHERE email = '" . $email . "'";  
    return mysqli_query($connect, $query);  
}
```

• Solution

```
function queryEliminarUsuario(string $email): bool {  
    $query = "DELETE FROM usuarios WHERE email = '" . $email . "'";  
    return mysqli_query(@connectionSQL(), $query);  
}
```

2. Inline Method 2

Pasamos de comprobar la variable \$conn en un if antes del return a comprobarlo con una ternaria en el return

• Problem

```
function connectionSQL() {  
    $servername = "localhost";  
    $username = "root";  
    $password = "";  
    $database = "udemy";  
  
    // Create connection  
    $conn = mysqli_connect($servername, $username, $password, $database);  
  
    // Check connection  
    if (!$conn) {  
        die("Connection failed: " . mysqli_connect_error());  
    }  
    return $conn;  
}
```

• Solution

```
function connectionSQL() {  
    $servername = "localhost";  
    $username = "root";  
    $password = "";  
    $database = "udemy";  
  
    // Create connection  
    $conn = mysqli_connect($servername, $username, $password, $database);  
  
    // Check connection  
    return !$conn ? die("Connection failed: " . mysqli_connect_error()) : $conn;  
}
```

3. Add parameter

Le hacemos posible que se le puedan pasar por parámetro las opciones de conexión, pero dejándole por defecto una configuración para no tener que modificar todo el código

- Problem

```
function connectionSQL() {
    $servername = "localhost";
    $username = "root";
    $password = "";
    $database = "udemy";

    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $database);

    // Check connection
    return !$conn ? die("Connection failed: " . mysqli_connect_error()) : $conn;
}
```

- Solution

```
function connectionSQL($servername = "localhost", $username = "root", $password = "",
    $database = "udemy") {
    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $database);

    // Check connection
    return !$conn ? die("Connection failed: " . mysqli_connect_error()) : $conn;
}
```

4. Extract method y Add parameter

Hemos sacado el foreach de la función main y le hemos pasado los parámetros por la función

- Problema

```
function mostrarHeader() {
    print "<h1>Resultado respuesta</h1>";
}

function main() {
    mostrarHeader();
    $notas = [9, 8, 5, 5];
    foreach ($notas as $nota) {
        print "Resposta: " . $nota . "<br/>";
    }
}

main();
```

- Solucion

```
function mostrarHeader($assignatura) {
    print "<h1>Resultado ". $assignatura . "</h1>";
}

function mostrarNotas($notas) {
    foreach ($notas as $nom => $nota) {
        print "Nota de ". $nom . ": " . $nota . "<br/>";
    }
}

function main() {
    mostrarHeader("Progrmacion");
    mostrarNotas(["Eros" => 9, "Ureña" => 8, "Redu" => 5, "Onetti" => 5]);
}

main();

?>
```

5. Extract variable

Hemos extraído las condiciones fuera y las hemos guardado en variables para facilitar la comprensión al leer

- Problem

```
$sistemaOperativo = "Windows";
$bitsProcessador = "64";
$ownerPC = "Iruña";

if (($sistemaOperativo == "Linux") && ($bitsProcessador == "64") && ($ownerPC == "Gabi")) {
    print ("Legal owner");
}
else {
    print ("Illegal owner");
}
```

- Solution

```
$sistemaOperativo = "Windows";
$bitsCPU = "64";
$ownerPC = "Iruña";

// Conditions
$isSystemLinux = $sistemaOperativo == "Linux";
$isCPU64bits = $bitsCPU == "64";
$isGabiOwner = $ownerPC == "Gabi";

if ($isSystemLinux && $isCPU64bits && $isGabiOwner) {
    print ("Legal");
}
else {
    print ("Illegal");
}
print (" owner");
```

6. Split Temporary Variable

Aunque sean variables temporales hay que darles un nombre correcto para facilitar la lectura y seguimiento del código, además de que evita problemas futuros si nos olvidamos de cambiar el contenido de la variable

- Problem

```
$alto = 5;
$ancho = 8;

$aux = 2 * ($alto + $ancho);
print $aux;
$aux = $alto * $ancho;
print $aux;
```

- Solution

```
$alto = 5;
$ancho = 8;

$perimetro = 2 * ($alto + $ancho);
print $perimetro;
$area = $alto * $ancho;
print $area;
```

7. Replace Temp with Query

Parece no mantener una variable temporal que este ocupando todo el rato espacio, movemos el cálculo a una función y devolvemos el resultado

- Problem

```
$horasTrabajadas = 80;
$horasPrecio = 5;
$propinas = 100;

$sueldo = ($horasPrecio * $horasTrabajadas) + $propinas;
if ($sueldo > 1000) {
    print ("Sueldo aceptable, ". $sueldo . "€");
} elseif ($sueldo == 1000) {
    print ("Mileurista, ". $sueldo . "€");
} else {
    print ("Demasiado justo para vivir, ". $sueldo . "€");
}
```

- Solution

```
function calcularSueldo() {
    $horasTrabajadas = 80;
    $horasPrecio = 5;
    $propinas = 100;
    return ($horasPrecio * $horasTrabajadas) + $propinas;
}

if (calcularSueldo() > 1000) {
    print ("Sueldo aceptable, ". calcularSueldo() . "€");
} elseif (calcularSueldo() == 1000) {
    print ("Mileurista, ". calcularSueldo() . "€");
} else {
    print ("Demasiado justo para vivir, ". calcularSueldo() . "€");
}
```