

Exemples de proves unitàries amb PHPUnit

operacions.php

Aquest fitxer defineix la classe Operations, que implementa dues operacions matemàtiques bàsiques: suma i divisió. També inclou validacions per garantir que els arguments siguin vàlids.

```
<?php

class Operations
{
    public function __construct()
    {
    }

    public function add($n1, $n2)
    {
        if ($n1 == NULL || $n2 == NULL || !is_numeric($n1) || !is_numeric($n2)) throw
new InvalidArgumentException('Value are not numeric');
        return $n1 + $n2;
    }

    public function divide($n1, $n2)
    {
        if ($n1 === NULL || $n2 === NULL || !is_numeric($n1) || !is_numeric($n2))
throw new InvalidArgumentException('Value are not numeric');
        if ($n2 == 0) throw new DivisionByZeroError();
        return $n1 / $n2;
    }
}
```

OperationsTest.php

Aquest fitxer defineix una sèrie de proves unitàries per a la classe Operations utilitzant PHPUnit. Es comprova que les funcions de la classe Operations es comportin de manera correcta, tant en condicions normals com en casos excepcionals.

```
<?php

use PHPUnit\Framework\TestCase;

class OperationsTest extends TestCase
{
    private $op;

    public function setup(): void
    {
        $this->op = new Operations();
    }

    public function testAddWithTwoValues()
    {

```

```

        $this->assertEquals(7, $this->op->add(2, 5));
    }

    public function testAddWithNullValues()
    {
        $this->expectException(InvalidArgumentException::class);
        $this->op->add(NULL, NULL);
    }

    public function testAddWithNoNumericValues()
    {
        $this->expectException(InvalidArgumentException::class);
        $this->op->add('a', 'hello');
    }

    public function testDivideWithTwoValues()
    {
        $this->assertEquals(1, $this->op->divide(5, 5));
    }

    public function testDivideWithNullValues()
    {
        $this->expectException(InvalidArgumentException::class);
        $this->op->divide(NULL, NULL);
    }

    public function testDivideWithNoNumericValues()
    {
        $this->expectException(InvalidArgumentException::class);
        $this->op->divide('a', 'hello');
    }

    public function testDivideBetweenZero()
    {
        $this->expectException(DivisionByZeroError::class);
        $this->op->divide(5, 0);
    }
}

```

Resultat esperat:

```

(base) C:\xampp\htdocs\ejemplo\S9>vendor\bin\phpunit tests\
Xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug.client_host/xdebug.client_port).
PHPUnit 11.5.0 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

.....                                         7 / 7 (100%)

Time: 00:01.099, Memory: 8.00 MB

OK (7 tests, 7 assertions)

```

login.php

Aquest fitxer conté una funció que permet validar les credencials d'un usuari per iniciar sessió.

```
<?php

function login($username, $password, $connection) {
    // Preparar la consulta SQL per obtenir el hash de la contrasenya
    $stmt = $connection->prepare("SELECT password FROM users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    // Si es troba un usuari, verificar la contrasenya
    if ($result->num_rows === 1) {
        $row = $result->fetch_assoc();
        $hashedPassword = $row['password'];

        // Verificar la contrasenya introduïda contra el hash emmagatzemat
        if (password_verify($password, $hashedPassword)) {
            return true; // Inici de sessió correcte
        }
    }

    return false; // Credencials incorrectes
}
?>
```

LoginIntegrationTest.php

Aquest fitxer conté proves d'integració escrites amb PHPUnit per verificar que la funció login del fitxer login.php funciona correctament amb una base de dades real.

```
<?php

use PHPUnit\Framework\TestCase;

class LoginIntegrationTest extends TestCase {
    private $connection;

    // Configuració inicial: establir connexió amb la base de dades
    protected function setUp(): void {
        $servername = "localhost";
        $username = "root";
        $password = ""; // La teva contrasenya de MySQL
        $dbname = "login_test";

        $this->connection = new mysqli($servername, $username, $password, $dbname);

        if ($this->connection->connect_error) {
            die("Connexió fallida: " . $this->connection->connect_error);
        }
    }
}
```

```

        // Assegurar-se que la taula té dades de prova
        $this->setUpTestData();
    }

    // Dades de prova inicials
    private function setUpTestData() {
        $this->connection->query("TRUNCATE TABLE users");

        $hashedPassword = password_hash("1234", PASSWORD_DEFAULT);
        $this->connection->query("INSERT INTO users (username, password) VALUES
('admin', '$hashedPassword')");

        $hashedPasswordUser = password_hash("userpass", PASSWORD_DEFAULT);
        $this->connection->query("INSERT INTO users (username, password) VALUES
('user', '$hashedPasswordUser')");
    }

    // Tancar connexió després de les proves
    protected function tearDown(): void {
        $this->connection->close();
    }

    // Prova d'inici de sessió amb credencials vàlides
    public function testLoginSuccess() {
        require_once __DIR__ . '/../login.php';

        $this->assertTrue(login('admin', '1234', $this->connection)); // Credencials
correctes
    }

    // Prova d'inici de sessió amb credencials incorrectes
    public function testLoginFailure() {
        require_once __DIR__ . '/../login.php';

        $this->assertFalse(login('admin', 'wrongpassword', $this->connection)); //
Contrasenya incorrecta
    }

    // Prova d'inici de sessió amb usuari inexistent
    public function testLoginNonExistentUser() {
        require_once __DIR__ . '/../login.php';

        $this->assertFalse(login('nonexistent', '1234', $this->connection)); //
Usuari inexistent
    }
}
?>

```

Per verificar que la funció login del fitxer login.php funciona correctament amb una base de dades real, prèviament caldrà executar el següent codi sql per crear una base de dades, amb una taula d'usuaris i dos usuaris de prova:

```
CREATE DATABASE login_test;
```

```
USE login_test;
```

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL,  
    password VARCHAR(255) NOT NULL  
);
```

```
INSERT INTO users (username, password) VALUES  
( 'admin', '1234'),  
( 'user', 'password');
```

Resultat esperat:

```
(base) C:\xampp\htdocs\ejemplo\S9_int_hash>vendor\bin\phpunit tests\  
Xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug.clie  
nt_host/xdebug.client_port).  
PHPUnit 11.5.1 by Sebastian Bergmann and contributors.  
  
Runtime:      PHP 8.2.12  
  
...                                                  3 / 3 (100%)  
  
Time: 00:00.591, Memory: 8.00 MB  
  
OK (3 tests, 3 assertions)
```

roles.php

Aquest fitxer defineix una funció anomenada canAccessPage(\$page), que s'utilitza per determinar si un usuari amb un rol específic pot accedir a una pàgina concreta.

```
<?php  
session_start();  
  
function canAccessPage($page) {  
    $permissions = [  
        'Administrador' => ['admin.php'],  
        'Usuari' => ['usuari.php'],  
        'Visitant' => ['visitant.php']  
    ];  
  
    $currentRole = $_SESSION['rol'] ?? 'Visitant'; // Per defecte "Visitant"  
    return in_array($page, $permissions[$currentRole] ?? []);  
}  
?>
```

Aquest fitxer conté proves automatitzades per comprovar si la funció canAccessPage funciona correctament.

```
<?php

use PHPUnit\Framework\TestCase;

class RoleAccessTest extends TestCase {
    protected function setUp(): void {
        $_SESSION = []; // Reiniciar $_SESSION abans de cada test
    }

    // Prova que un administrador només pot accedir a admin.php
    public function testAdminAccess() {
        require_once __DIR__ . '/../roles.php';

        $_SESSION['rol'] = 'Administrador';
        $this->assertTrue(canAccessPage('admin.php'));
        $this->assertFalse(canAccessPage('usuari.php'));
        $this->assertFalse(canAccessPage('visitant.php'));
    }

    // Prova que un usuari només pot accedir a usuari.php
    public function testUserAccess() {
        require_once __DIR__ . '/../roles.php';

        $_SESSION['rol'] = 'Usuari';
        $this->assertTrue(canAccessPage('usuari.php'));
        $this->assertFalse(canAccessPage('admin.php'));
        $this->assertFalse(canAccessPage('visitant.php'));
    }

    // Prova que un visitant només pot accedir a visitant.php
    public function testVisitorAccess() {
        require_once __DIR__ . '/../roles.php';

        $_SESSION['rol'] = 'Visitant';
        $this->assertTrue(canAccessPage('visitant.php'));
        $this->assertFalse(canAccessPage('admin.php'));
        $this->assertFalse(canAccessPage('usuari.php'));
    }

    // Prova d'accés amb un rol inexistent
    public function testInvalidRole() {
        require_once __DIR__ . '/../roles.php';

        $_SESSION['rol'] = 'InvalidRole';
        $this->assertFalse(canAccessPage('admin.php'));
        $this->assertFalse(canAccessPage('usuari.php'));
        $this->assertFalse(canAccessPage('visitant.php'));
    }
}
```

```

    }

    // Prova d'accés a una pàgina inexistent
    public function testInvalidPage() {
        require_once __DIR__ . '/../roles.php';

        $_SESSION['rol'] = 'Administrador';
        $this->assertFalse(canAccessPage('nonexistent.php'));

        $_SESSION['rol'] = 'Usuari';
        $this->assertFalse(canAccessPage('nonexistent.php'));

        $_SESSION['rol'] = 'Visitant';
        $this->assertFalse(canAccessPage('nonexistent.php'));
    }
}

```

Resultat esperat:

```

(base) C:\xampp\htdocs\ejemplo\S9_rol>vendor\bin\phpunit tests\
Xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug.client_host/xdebug.client_port).
PHPUnit 11.5.1 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12

.....                                         5 / 5 (100%)

Time: 00:00.014, Memory: 8.00 MB

OK (5 tests, 15 assertions)

```