INVESTIGACIÓN PHPUnit

Es un popular framework de pruebas para PHP. Desarrolladores de todo el mundo lo utilizan para escribir y ejecutar pruebas unitarias en su código PHP. Las pruebas unitarias son una parte crucial del desarrollo de software que ayuda a garantizar la calidad y la corrección del código. Con PHPUnit, puede escribir pruebas automatizadas que comprueban el comportamiento de su código, detectan errores y regresiones, y garantizan que los cambios en el código no afecten a la funcionalidad existente.y que sea Diseñado fácil de usar.
-sintaxis simple e intuitiva para definir casos de prueba
-para reducir errores en el código y no tener errores que puedan ser costosos.

A quién va dirigido:
Está dirigido a desarrolladores en php que se inician en este mundo de las pruebas .

Instalación:

1. Crear un nuevo proyecto PHP: Primero, crea un nuevo directorio para tu proyecto PHP. Por ejemplo, podrías crear un directorio llamado "phpunit-intro".

2. Instalar Composer: A continuación, deberá instalar Composer, un popular gestor de paquetes para PHP. Puede descargar Composer desde el sitio web oficial (https://getcomposer.org/download/) o instalarlo usando el gestor de paquetes de su sistema operativo, por ejemplo, brew en macOS.

Características de PHPUnit

- Automatización de pruebas : automatiza tareas de pruebas repetitivas con el mínimo esfuerzo.
- Marco de simulación : admite la creación de objetos simulados para pruebas aisladas.
- Afirmaciones : Métodos integrados para validar los resultados esperados.
- Análisis de cobertura de código : rastrea el porcentaje de código ejecutado durante las pruebas.
- Suites de pruebas : organice y ejecute casos de prueba relacionados juntos.
  Integración: funciona perfectamente con herramientas CI/CD e IDE.

Afirmaciones en PHPUnit

- assertEquals() : comprueba si dos valores son iguales.
- assertTrue() / assertFalse() : valida condiciones booleanas.
- assertNull() : garantiza que una variable sea nula.
- assertContains() : verifica si un elemento existe en una matriz o cadena.
- assertInstanceOf() : confirma que un objeto es de una clase específica.
- assertCount() : válida la cantidad de elementos en una matriz o iterable.

EJEMPLOS:

CRUD DE FICHAJES DE CICLISTAS:
primero debemos crear una carpeta del proyecto que se llame ciclistas luego creamos las carpetas con los archivos :

index. , agregar , editar, eliminar, guardar, database,todos . php
styles.css para su estilo antes de inciar hacer el código creamos nuestra base en mysql

```
CREATE TABLE `ciclistas` (
  `id` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `apellido` varchar(100) NOT NULL,
  `edad` int(11) NOT NULL,
  `pais` varchar(50) NOT NULL,
  `especialidad` varchar(100) NOT NULL,
  `salario` decimal(10,2) NOT NULL,
  `peso` decimal(5,2) NOT NULL,
  `altura` decimal(5,2) NOT NULL,
  `potencia_maxima` int(11) DEFAULT NULL,
  `vo2_max` decimal(5,2) DEFAULT NULL,
  `fecha_contrato` date NOT NULL,
  `equipo anterior` varchar(100) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
```

luego pasamos ahora así crear todos los códigos para cada archivo
conexion.php

después de todo eso asi iniciamos a realizar las pruebas a nuestro crud
y arreglamos nuestro proyecto a esta estructura :

```
ciclistas/
── src/
   ── models/
│    │   ── CiclistaModel.php
│    └── database.php
├    ─ tests/
│    └── CiclistaModelTest.php
├── vendor/
├── composer.json
└── public/
    ├── index.php
    ├── agregar.php
    └── ... más archivos
```

donde se debe descargar en tu carpeta :
composer require --dev phpunit/phpunit
luego acomodamos y organizamos los codigos asi
models:

```php
<?php
namespace App\Models;

class CiclistaModel {
    private $base_de_datos;

    public function __construct($base_de_datos) {
        $this->base_de_datos = $base_de_datos;
    }

    public function getAll($especialidad = null) {
        $where = "";
        $params = [];

        if ($especialidad) {
            $where = " WHERE especialidad = :especialidad";
            $params[':especialidad'] = $especialidad;
        }

        $stmt = $this->base_de_datos->prepare("SELECT * FROM ciclistas" . $where);

        foreach ($params as $key => $value) {
            $stmt->bindValue($key, $value);
        }

        $stmt->execute();
        return $stmt->fetchAll(\PDO::FETCH_ASSOC);
    }

    public function getById($id) {
        $stmt = $this->base_de_datos->prepare("SELECT * FROM ciclistas WHERE id = :id");
        $stmt->bindParam(':id', $id, \PDO::PARAM_INT);
        $stmt->execute();
        return $stmt->fetch(\PDO::FETCH_ASSOC);
    }

    public function create($data) {
        $sql = "INSERT INTO ciclistas (
                nombre, apellido, edad, pais, especialidad,
                salario, peso, altura, potencia_maxima, vo2_max,
                fecha_contrato, equipo_anterior
                ) VALUES (
                :nombre, :apellido, :edad, :pais, :especialidad,
                :salario, :peso, :altura, :potencia_maxima, :vo2_max,
                :fecha_contrato, :equipo_anterior
                )";

        $stmt = $this->base_de_datos->prepare($sql);
```

```php
        return $stmt->execute($data);
    }

    public function update($id, $data) {
        $data[':id'] = $id;
        $sql = "UPDATE ciclistas SET
                nombre = :nombre,
                apellido = :apellido,
                edad = :edad,
                pais = :pais,
                especialidad = :especialidad,
                salario = :salario,
                peso = :peso,
                altura = :altura,
                potencia_maxima = :potencia_maxima,
                vo2_max = :vo2_max,
                fecha_contrato = :fecha_contrato,
                equipo_anterior = :equipo_anterior
                WHERE id = :id";

        $stmt = $this->base_de_datos->prepare($sql);
        return $stmt->execute($data);
    }

    public function delete($id) {
        $stmt = $this->base_de_datos->prepare("DELETE FROM ciclistas WHERE id = :id");
        $stmt->bindParam(':id', $id, \PDO::PARAM_INT);
        return $stmt->execute();
    }

    public function getTotalPresupuesto() {
        $stmt = $this->base_de_datos->query("SELECT SUM(salario) as total FROM ciclistas");
        $result = $stmt->fetch(\PDO::FETCH_ASSOC);
        return (float) ($result['total'] ?? 0);
    }
}
```

la conexión

```php
<?php
$host = 'localhost';
$contrasena = "";
$usuario = "root";
$nombre_base_de_datos = "equipo_ciclistas";

try {
    $base_de_datos = new PDO('mysql:host=' . $host . ';dbname=' . $nombre_base_de_datos, $usuario, $contrasena);

    $base_de_datos->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Error en la conexión a la base de datos: " . $e->getMessage();
    exit();
}
?>
```

test
database test

```php
<?php
namespace App\Tests;

use PHPUnit\Framework\TestCase;
use PDO;
use PDOException;
use App\Models\CiclistaModel;

abstract class DatabaseTest extends TestCase
{
    protected static $base_de_datos;
    protected $model;

    public static function setUpBeforeClass(): void
    {
        try {
            echo "Intentando conectar a la base de datos...\n";
            self::$base_de_datos = new PDO(
                'mysql:host=localhost;dbname=equipo_ciclistas_test',
                'root',
                '',
                [
                    PDO::ATTR_TIMEOUT => 5,
                    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
                ]
            );
            echo "Conexión exitosa\n";


            self::$base_de_datos->exec("CREATE TABLE IF NOT EXISTS ciclistas (
                id INT AUTO_INCREMENT PRIMARY KEY,
                nombre VARCHAR(100) NOT NULL,
                apellido VARCHAR(100) NOT NULL,
                edad INT NOT NULL,
                pais VARCHAR(50) NOT NULL,
                especialidad VARCHAR(100) NOT NULL,
                salario DECIMAL(10,2) NOT NULL,
                peso DECIMAL(5,2) NOT NULL,
                altura DECIMAL(5,2) NOT NULL,
                potencia_maxima INT,
                vo2_max DECIMAL(5,2),
                fecha_contrato DATE NOT NULL,
                equipo_anterior VARCHAR(100),
                created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
            )");
        } catch (PDOException $e) {
            die("Error de conexión: " . $e->getMessage());
        }
```

```
    }

    protected function setUp(): void
    {
        self::$base_de_datos->exec("TRUNCATE TABLE ciclistas");
        self::$base_de_datos->exec("INSERT INTO ciclistas
            (nombre, apellido, edad, pais, especialidad, salario, peso, altura, fecha_contrato)
            VALUES
            ('Test', 'Ciclista', 25, 'Testlandia', 'Escalador', 100000, 65.5, 1.75, '2023-01-01'),
            ('Otro', 'Ciclista', 28, 'Testlandia', 'Sprinter', 150000, 70.0, 1.80, '2023-01-01')");

        $this->model = new CiclistaModel(self::$base_de_datos);
    }

    public static function tearDownAfterClass(): void
    {
        self::$base_de_datos->exec("DROP TABLE IF EXISTS ciclistas");
        self::$base_de_datos = null;
    }
}

```

ciclista modeltest

```
<?php
namespace App\Tests;

use PHPUnit\Framework\TestCase;
use App\Models\CiclistaModel;

class CiclistaModelTest extends DatabaseTest
{
    public function testGetAll()
    {
        $result = $this->model->getAll();
        $this->assertCount(2, $result);
        $this->assertEquals('Test', $result[0]['nombre']);
    }

    public function testGetByEspecialidad()
    {
        $result = $this->model->getAll('Escalador');
        $this->assertCount(1, $result);
        $this->assertEquals('Escalador', $result[0]['especialidad']);
    }

    public function testGetById()
    {
        $id = self::$base_de_datos->query("SELECT id FROM ciclistas LIMIT 1")->fetchColumn();
        $result = $this->model->getById($id);
        $this->assertEquals('Test', $result['nombre']);
    }

    public function testCreate()
    {
        $newCiclista = [
            ':nombre' => 'Nuevo',
            ':apellido' => 'Ciclista',
            ':edad' => 30,
            ':pais' => 'Testlandia',
            ':especialidad' => 'Contrarrelojista',
            ':salario' => 200000,
            ':peso' => 68.0,
            ':altura' => 1.78,
            ':potencia_maxima' => 400,
            ':vo2_max' => 80.0,
            ':fecha_contrato' => '2023-01-01',
            ':equipo_anterior' => 'Test Team'
        ];

        $result = $this->model->create($newCiclista);
        $this->assertTrue($result);
```

```php
        $count = self::$base_de_datos->query("SELECT COUNT(*) FROM ciclistas")->fetchColumn();
        $this->assertEquals(3, $count);
    }

    public function testUpdate()
    {
        $id = self::$base_de_datos->query("SELECT id FROM ciclistas WHERE nombre = 'Test'")->fetchColumn();

        $updateData = [
            ':nombre' => 'Test Actualizado',
            ':apellido' => 'Ciclista',
            ':edad' => 26,
            ':pais' => 'Testlandia',
            ':especialidad' => 'Escalador',
            ':salario' => 110000,
            ':peso' => 66.0,
            ':altura' => 1.75,
            ':potencia_maxima' => 420,
            ':vo2_max' => 82.0,
            ':fecha_contrato' => '2023-01-01',
            ':equipo_anterior' => 'Test Team'
        ];

        $result = $this->model->update($id, $updateData);
        $this->assertTrue($result);

        $updated = $this->model->getById($id);
        $this->assertEquals('Test Actualizado', $updated['nombre']);
    }

    public function testDelete()
    {
        $id = self::$base_de_datos->query("SELECT id FROM ciclistas WHERE nombre = 'Test'")->fetchColumn();
        $result = $this->model->delete($id);
        $this->assertTrue($result);

        $count = self::$base_de_datos->query("SELECT COUNT(*) FROM ciclistas")->fetchColumn();
        $this->assertEquals(1, $count);
    }

    public function testGetTotalPresupuesto()
    {
        $total = $this->model->getTotalPresupuesto();
        $this->assertEquals(250000, $total);
    }
}
```

composer.json

```json
{
    "name": "ciclistas1/ciclistas",
    "description": "Sistema de gestión de equipo de ciclistas",
    "type": "project",
    "require": {
        "php": "^8.0",
        "ext-pdo": "*"
    },
    "require-dev": {
        "phpunit/phpunit": "^11.5"
    },
    "autoload": {
        "psr-4": {
            "App\\": "src/",
            "App\\Tests\\": "tests/"
        }
    },
    "autoload-dev": {
        "psr-4": {
            "App\\Tests\\": "tests/"
        }
    }

}

```

phpunit.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<phpunit bootstrap="vendor/autoload.php"
         colors="true"
         verbose="true">
    <testsuites>
        <testsuite name="Ciclistas Test Suite">
            <directory>tests</directory>
            <exclude>tests/DatabaseTest.php</exclude>
        </testsuite>
    </testsuites>

    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
        </whitelist>
    </filter>
       <logging>
        <log type="testdox-html" target="report.html"/>
    </logging>
    <coverage>
        <report>
            <html outputDirectory="coverage-report"/>
        </report>
    </coverage>
</phpunit>
```

Crea una base de datos separada para pruebas:

CREATE DATABASE equipo_ciclistas_test;

se debe instalar esto
composer install

y luego php vendor\bin\phpunit para ejecutar las pruebas asi

C:\xampp\htdocs\ciclistas1>php vendor/bin/phpunit
PHPUnit 11.5.19 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.12
Configuration: C:\xampp\htdocs\ciclistas1\phpunit.xml

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

Time: 00:00.146, Memory: 8.00 MB

OK, but there were issues!
Tests: 7, Assertions: 12, PHPUnit Deprecations: 1.

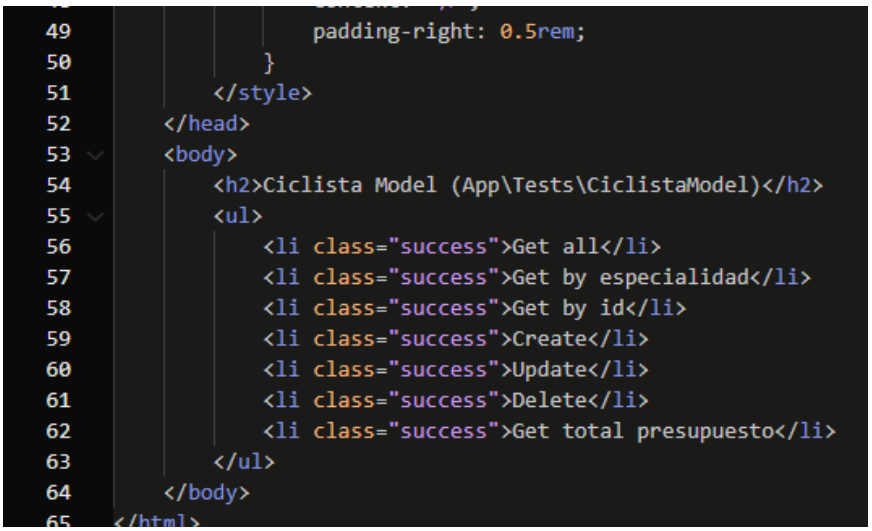
y si quiere el html de las pruebas sólo tiene que ejecutar el siguiente código
php vendor/bin/phpunit --testdox-html report.html

y se le genera el código en su proyecto

Ciclista Model (App\Tests\CiclistaModel)

- ✓ Obtener todo
- ✓ Obtener por especialidad
- ✓ Obtener por id
- ✓ Crear
- ✓ Actualizar
- ✓ Borrar
- ✓ Obtener presupuesto total

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8"/>
        <title>Test Documentation</title>
        <style>
            body {
                text-rendering: optimizeLegibility;
                font-family: Source SansSerif Pro, Arial, sans-serif;
                font-variant-ligatures: common-ligatures;
                font-kerning: normal;
                margin-left: 2rem;
                background-color: #fff;
                color: #000;
            }

            body > ul > li {
                font-size: larger;
            }

            h2 {
                font-size: larger;
                text-decoration-line: underline;
                text-decoration-thickness: 2px;
                margin: 0;
                padding: 0.5rem 0;
            }

            ul {
                list-style: none;
                margin: 0 0 2rem;
                padding: 0 0 0 1rem;
                text-indent: -1rem;
            }

            .success:before {
                color: #4e9a06;
                content: '✓';
                padding-right: 0.5rem;
            }

            .defect {
                color: #a40000;
            }

            .defect:before {
                color: #a40000;
                content: 'X';
```



```
                padding-right: 0.5rem;
            }
        </style>
    </head>
    <body>
        <h2>Ciclista Model (App\Tests\CiclistaModel)</h2>
        <ul>
            <li class="success">Get all</li>
            <li class="success">Get by especialidad</li>
            <li class="success">Get by id</li>
            <li class="success">Create</li>
            <li class="success">Update</li>
            <li class="success">Delete</li>
            <li class="success">Get total presupuesto</li>
        </ul>
    </body>
</html>
```

ahora el public eliminar

```php
<?php
include '../src/database.php';

if (isset($_GET['id'])) {
    $id = intval($_GET['id']);

    try {

        $stmt = $base_de_datos->prepare("SELECT nombre, apellido FROM
ciclistas WHERE id = ?");
        $stmt->execute([$id]);
        $ciclista = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($ciclista) {

            $stmt = $base_de_datos->prepare("DELETE FROM ciclistas
WHERE id = ?");
            $stmt->execute([$id]);

            $mensaje = "Ciclista {$ciclista['nombre']}
{$ciclista['apellido']} eliminado correctamente";
            header("Location: index.php?success=" .
urlencode($mensaje));
        } else {
            header("Location: index.php?error=Ciclista no encontrado");
        }
        exit();
    } catch (PDOException $e) {
```

```
        die("Error al eliminar: " . $e->getMessage());
    }
} else {
    header("Location: index.php");
    exit();
}
?>
```

editar

```
<?php
include_once '../src/database.php';

if (!isset($_GET['id'])) {
    header("Location: index.php");
    exit();
}

$id = $_GET['id'];
$stmt = $base_de_datos->prepare("SELECT * FROM ciclistas WHERE id = ?");
$stmt->execute([$id]);
$ciclista = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$ciclista) {
    header("Location: index.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Editar Ciclista</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Editar Ciclista: <?php echo $ciclista['nombre'] . ' ' . $ciclista['apellido']; ?></h1>

        <form action="guardar.php" method="post">
            <input type="hidden" name="id" value="<?php echo $ciclista['id']; ?>">

            <div class="form-grid">
                <div class="form-group">
                    <label for="nombre">Nombre:</label>
                    <input type="text" id="nombre" name="nombre" value="<?php echo
$ciclista['nombre']; ?>" required>
                </div>

                <div class="form-group">
                    <label for="apellido">Apellido:</label>
                    <input type="text" id="apellido" name="apellido" value="<?php echo
$ciclista['apellido']; ?>" required>
```

```
                </div>

                <div class="form-group">
                    <label for="edad">Edad:</label>
                    <input type="number" id="edad" name="edad" min="18" max="50" value="<?php echo
$ciclista['edad']; ?>" required>
                </div>

                <div class="form-group">
                    <label for="pais">País:</label>
                    <input type="text" id="pais" name="pais" value="<?php echo $ciclista['pais'];
?>" required>
                </div>

                <div class="form-group">
                    <label for="especialidad">Especialidad:</label>
                    <select id="especialidad" name="especialidad" required>
                        <option value="Contrarrelojista" <?php echo $ciclista['especialidad'] ==
'Contrarrelojista' ? 'selected' : ''; ?>>Contrarrelojista</option>
                        <option value="Escalador" <?php echo $ciclista['especialidad'] ==
'Escalador' ? 'selected' : ''; ?>>Escalador</option>
                        <option value="Sprinter" <?php echo $ciclista['especialidad'] == 'Sprinter'
? 'selected' : ''; ?>>Sprinter</option>
                        <option value="Rodador" <?php echo $ciclista['especialidad'] == 'Rodador' ?
'selected' : ''; ?>>Rodador</option>
                        <option value="Gregario" <?php echo $ciclista['especialidad'] == 'Gregario'
? 'selected' : ''; ?>>Gregario</option>
                        <option value="Clasicómano" <?php echo $ciclista['especialidad'] ==
'Clasicómano' ? 'selected' : ''; ?>>Clasicómano</option>
                    </select>
                </div>

                <div class="form-group">
                    <label for="salario">Salario Anual (USD):</label>
                    <input type="number" id="salario" name="salario" min="50000" step="0.01"
value="<?php echo $ciclista['salario']; ?>" required>
                </div>
            </div>

            <h2>Ficha Técnica</h2>

            <div class="form-grid">
                <div class="form-group">
                    <label for="peso">Peso (kg):</label>
                    <input type="number" id="peso" name="peso" min="40" max="100" step="0.1"
value="<?php echo $ciclista['peso']; ?>" required>
                </div>

                <div class="form-group">
                    <label for="altura">Altura (m):</label>
                    <input type="number" id="altura" name="altura" min="1.50" max="2.10" step="0.01"
value="<?php echo $ciclista['altura']; ?>" required>
                </div>

                <div class="form-group">
                    <label for="potencia_maxima">Potencia Máxima (W):</label>
                    <input type="number" id="potencia_maxima" name="potencia_maxima" min="200"
max="600" value="<?php echo $ciclista['potencia_maxima']; ?>">
                </div>
```

```
                <div class="form-group">
                    <label for="vo2_max">VO2 Máximo (ml/kg/min):</label>
                    <input type="number" id="vo2_max" name="vo2_max" min="40" max="90" step="0.1"
value="<?php echo $ciclista['vo2_max']; ?>">
                </div>

                <div class="form-group">
                    <label for="fecha_contrato">Fecha de Contrato:</label>
                    <input type="date" id="fecha_contrato" name="fecha_contrato" value="<?php echo
$ciclista['fecha_contrato']; ?>" required>
                </div>

                <div class="form-group">
                    <label for="equipo_anterior">Equipo Anterior:</label>
                    <input type="text" id="equipo_anterior" name="equipo_anterior" value="<?php echo
$ciclista['equipo_anterior']; ?>">
                </div>
            </div>

            <div class="form-actions">
                <button type="submit" class="btn">Actualizar Ciclista</button>
                <a href="index.php" class="btn-cancelar">Cancelar</a>
            </div>
        </form>
    </div>

    <script>

        const pesoInput = document.getElementById('peso');
        const alturaInput = document.getElementById('altura');

        function calcularIMC() {
            const peso = parseFloat(pesoInput.value);
            const altura = parseFloat(alturaInput.value);

            if (peso && altura) {
                const imc = peso / (altura * altura);
                console.log('IMC actualizado:', imc.toFixed(2));
            }
        }

        pesoInput.addEventListener('input', calcularIMC);
        alturaInput.addEventListener('input', calcularIMC);
    </script>
</body>
</html>
```

guardar.php

```php
<?php
include_once '../src/database.php';
include_once '../src/models/CiclistaModel.php';

$model = new App\Models\CiclistaModel($base_de_datos);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    if ($_POST['edad'] < 18 || $_POST['edad'] > 50) {
        die("La edad debe estar entre 18 y 50 años");
    }

    if ($_POST['salario'] < 50000) {
        die("El salario mínimo es de $50,000");
    }


    $data = [
        ':nombre' => htmlspecialchars($_POST['nombre']),
        ':apellido' => htmlspecialchars($_POST['apellido']),
        ':edad' => intval($_POST['edad']),
        ':pais' => htmlspecialchars($_POST['pais']),
        ':especialidad' => htmlspecialchars($_POST['especialidad']),
        ':salario' => floatval($_POST['salario']),
        ':peso' => floatval($_POST['peso']),
        ':altura' => floatval($_POST['altura']),
        ':potencia_maxima' => isset($_POST['potencia_maxima']) ? intval($_POST['potencia_maxima']) :
null,
        ':vo2_max' => isset($_POST['vo2_max']) ? floatval($_POST['vo2_max']) : null,
        ':fecha_contrato' => $_POST['fecha_contrato'],
        ':equipo_anterior' => isset($_POST['equipo_anterior']) ?
htmlspecialchars($_POST['equipo_anterior']) : null
    ];

    try {
        if (isset($_POST['id']) && !empty($_POST['id'])) {
            $success = $model->update(intval($_POST['id']), $data);
            $message = $success ? "Ciclista actualizado correctamente" : "Error al actualizar";
        } else {
            $success = $model->create($data);
            $message = $success ? "Ciclista creado correctamente" : "Error al crear";
        }

        header("Location: index.php?success=" . urlencode($message));
        exit();
    } catch (PDOException $e) {

        error_log("Error en guardar.php: " . $e->getMessage());


        die("Ocurrió un error al guardar los datos. Por favor intenta nuevamente.");
    }
} else {
    header("Location: index.php");
    exit();
}
```

agregar.php

```php
<?php  './src/database.php'; ?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Agregar Ciclista</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Agregar Nuevo Ciclista</h1>

        <form action="guardar.php" method="post">
            <div class="form-grid">
                <div class="form-group">
                    <label for="nombre">Nombre:</label>
                    <input type="text" id="nombre" name="nombre" required>
                </div>

                <div class="form-group">
                    <label for="apellido">Apellido:</label>
                    <input type="text" id="apellido" name="apellido" required>
                </div>

                <div class="form-group">
                    <label for="edad">Edad:</label>
                    <input type="number" id="edad" name="edad" min="18" max="50" required>
                </div>

                <div class="form-group">
                    <label for="pais">País:</label>
                    <input type="text" id="pais" name="pais" required>
                </div>

                <div class="form-group">
                    <label for="especialidad">Especialidad:</label>
                    <select id="especialidad" name="especialidad" required>
                        <option value="">Seleccione...</option>
                        <option value="Contrarrelojista">Contrarrelojista</option>
                        <option value="Escalador">Escalador</option>
                        <option value="Sprinter">Sprinter</option>
                        <option value="Rodador">Rodador</option>
                        <option value="Gregario">Gregario</option>
                        <option value="Clasicómano">Clasicómano</option>
                    </select>
                </div>

                <div class="form-group">
                    <label for="salario">Salario Anual (USD):</label>
                    <input type="number" id="salario" name="salario" min="50000" step="0.01" required>
                </div>
            </div>

            <h2>Ficha Técnica</h2>

            <div class="form-grid">
                <div class="form-group">
                    <label for="peso">Peso (kg):</label>
                    <input type="number" id="peso" name="peso" min="40" max="100" step="0.1" required>
                </div>

                <div class="form-group">
                    <label for="altura">Altura (m):</label>
                    <input type="number" id="altura" name="altura" min="1.50" max="2.10" step="0.01" required>
                </div>

                <div class="form-group">
```

```
                    <label for="potencia_maxima">Potencia Máxima (W):</label>
                    <input type="number" id="potencia_maxima" name="potencia_maxima" min="200" max="600">
                </div>

                <div class="form-group">
                    <label for="vo2_max">VO2 Máximo (ml/kg/min):</label>
                    <input type="number" id="vo2_max" name="vo2_max" min="40" max="90" step="0.1">
                </div>

                <div class="form-group">
                    <label for="fecha_contrato">Fecha de Contrato:</label>
                    <input type="date" id="fecha_contrato" name="fecha_contrato" required>
                </div>

                <div class="form-group">
                    <label for="equipo_anterior">Equipo Anterior:</label>
                    <input type="text" id="equipo_anterior" name="equipo_anterior">
                </div>
            </div>

            <div class="form-actions">
                <button type="submit" class="btn">Guardar Ciclista</button>
                <a href="index.php" class="btn-cancelar">Cancelar</a>
            </div>
        </form>
    </div>

    <script>

        document.querySelector('form').addEventListener('submit', function(e) {
            const peso = parseFloat(document.getElementById('peso').value);
            const altura = parseFloat(document.getElementById('altura').value);
            const imc = peso / (altura * altura);

            if (imc < 18 || imc > 25) {
                if (!confirm('El IMC calculado está fuera del rango recomendado (18-25). ¿Deseas continuar?'))
{
                    e.preventDefault();
                }
            }
        });
    </script>
</body>
</html>
```

index.php

```
<?php include_once '../src/database.php';
include_once '../src/models/CiclistaModel.php';

$model = new App\Models\CiclistaModel($base_de_datos);
$especialidad = $_GET['especialidad'] ?? null;
$ciclistas = $model->getAll($especialidad);
$totalPresupuesto = $model->getTotalPresupuesto(); ?>



<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gestión de Equipo de Ciclistas</title>
    <link rel="stylesheet" href="./styles.css">
```

```
</head>
<body>
    <div class="container">
        <h1>Equipo de Ciclistas ineos</h1>

        <div class="header-actions">
            <a href="agregar.php" class="btn">Agregar Ciclista</a>
            <div class="total-presupuesto">
                <?php
                $stmt = $base_de_datos->query("SELECT COUNT(*) as total_ciclistas, SUM(salario) as
total_salarios FROM ciclistas");
                $totales = $stmt->fetch(PDO::FETCH_ASSOC);
                echo "Ciclistas: " . $totales['total_ciclistas'] . " | Presupuesto Total: $" .
number_format($totales['total_salarios'], 2);
                ?>
            </div>
        </div>

        <div class="filtros">
            <form method="get" action="">
                <select name="especialidad" onchange="this.form.submit()">
                    <option value="">Todas las especialidades</option>
                    <?php
                    $especialidades = $base_de_datos->query("SELECT DISTINCT especialidad FROM ciclistas");
                    while ($esp = $especialidades->fetch(PDO::FETCH_ASSOC)) {
                        $selected = ($_GET['especialidad'] ?? '') == $esp['especialidad'] ? 'selected' : '';
                        echo "<option value='{$esp['especialidad']}'
$selected>{$esp['especialidad']}</option>";
                    }
                    ?>
                </select>
            </form>
        </div>

        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nombre</th>
                    <th>Edad</th>
                    <th>País</th>
                    <th>Especialidad</th>
                    <th>Salario</th>
                    <th>Ficha</th>
                    <th>Acciones</th>
                </tr>
            </thead>
            <tbody>
                <?php
                $where = "";
                if (isset($_GET['especialidad']) && !empty($_GET['especialidad'])) {
                    $where = " WHERE especialidad = '" . $_GET['especialidad'] . "'";
                }

                $stmt = $base_de_datos->query("SELECT * FROM ciclistas $where ORDER BY nombre ASC");

                if ($stmt->rowCount() > 0) {
                    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                        echo "<tr>
                            <td>{$row['id']}</td>
                            <td>{$row['nombre']} {$row['apellido']}</td>
                            <td>{$row['edad']}</td>
                            <td>{$row['pais']}</td>
                            <td>{$row['especialidad']}</td>
                            <td>$" . number_format($row['salario'], 2) . "</td>
                            <td>
                                <a href='#' class='btn-ficha' onclick='mostrarFicha({$row['id']})'>Ver
Ficha</a>
                                <div id='ficha-{$row['id']}' class='ficha-tecnica'>
                                    <h3>Ficha Técnica de {$row['nombre']} {$row['apellido']}</h3>
```

```
                                <div class='ficha-grid'>
                                    <div><strong>Edad:</strong> {$row['edad']}</div>
                                    <div><strong>País:</strong> {$row['pais']}</div>
                                    <div><strong>Especialidad:</strong> {$row['especialidad']}</div>
                                    <div><strong>Salario:</strong> $" . number_format($row['salario'], 2) .
"</div>
                                    <div><strong>Peso:</strong> {$row['peso']} kg</div>
                                    <div><strong>Altura:</strong> {$row['altura']} m</div>
                                    <div><strong>IMC:</strong> " . number_format($row['peso'] /
($row['altura'] * $row['altura']), 2) . "</div>
                                    <div><strong>Potencia Máx:</strong> " . ($row['potencia_maxima'] ??
'N/A') . " W</div>
                                    <div><strong>VO2 Máx:</strong> " . ($row['vo2_max'] ?? 'N/A') . "
ml/kg/min</div>
                                    <div><strong>Contrato desde:</strong> {$row['fecha_contrato']}</div>
                                    <div><strong>Equipo anterior:</strong> " . ($row['equipo_anterior'] ??
'Ninguno') . "</div>
                                </div>
                                <button onclick='ocultarFicha({$row['id']})' class='btn'>Cerrar</button>
                            </div>
                        </td>
                        <td class='acciones'>
                            <a href='editar.php?id={$row['id']}' class='btn-editar'>Editar</a>
                            <a href='eliminar.php?id={$row['id']}' class='btn-eliminar' onclick='return
confirm(\"¿Estás seguro de eliminar a {$row['nombre']} {$row['apellido']}?\")'>Eliminar</a>
                        </td>
                    </tr>";
                }
            } else {
                echo "<tr><td colspan='8'>No hay ciclistas registrados</td></tr>";
            }
            ?>
        </tbody>
    </table>
</div>

<script>
    function mostrarFicha(id) {
        document.getElementById('ficha-' + id).style.display = 'block';
    }

    function ocultar Ficha(id) {
        document.getElementById('ficha-' + id).style.display = 'none';
    }
</script>
</body>
</html>
```