

Pruebas

Martín Martínez García
IES AGUADULCE

Clases

Clase actividades

Tabla antes de los cambios:

| | id | titulo | descripcion | precio | categoria | duracion | ubicacion | fecha | hora |
|---|----|------------------------------------|---|--------|-----------|----------|------------------------------|------------|----------|
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 2 | Visita a Bodegas con Cata de Vinos | Disfruta de una visita guiada a bodegas locales co... | 80.00 | 1 | 4.00 | Bodegas del Valle | 2024-09-20 | 15:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 3 | Clase de Cocina Local | Aprende a preparar platos típicos con un chef loca... | 60.00 | 3 | 2.00 | Escuela de Cocina Gourmet | 2024-09-22 | 18:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 4 | Excursión en Barco al Atardecer | Disfruta de una tranquila excursión en barco duran... | 75.00 | 2 | 2.00 | Puerto Deportivo | 2024-09-25 | 19:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 5 | Escalada en Roca | Desafia tus habilidades con una sesión de escalada... | 100.00 | 2 | 5.00 | Zona de Escalada de Montaña | 2024-09-30 | 09:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 6 | Masaje Relax | Relájate con un masaje completo en un spa local. | 45.00 | 4 | 1.00 | Spa Wellness | 2024-09-18 | 14:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 7 | Yoga en la Playa | Participa en una sesión de yoga al amanecer en la ... | 30.00 | 4 | 1.50 | Playa de la Arena | 2024-09-19 | 07:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 8 | Senderismo en el Parque Nacional | Explora senderos naturales en un parque nacional c... | 90.00 | 5 | 6.00 | Parque Nacional de la Sierra | 2024-09-28 | 08:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 9 | Observación de Aves | Participa en una excursión para observar aves en s... | 50.00 | 5 | 4.00 | Reserva Natural de los Aves | 2024-10-01 | 06:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 30 | Clase de Yoga Avanzada | Clase más intensa | 25.00 | 0 | 2.00 | Sala B | 2024-12-10 | 12:00:00 |
| <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar | 31 | Clase de Pilates | Clase relajante | 20.00 | 0 | 2.00 | Sala A | 2024-12-10 | 10:00:00 |

Crear:

Método:

```
// Crear una nueva actividad
public static function crearActividad($titulo, $descripcion, $duracion, $ubicacion, $precio, $categoria, $fecha, $hora) {
    // Validaciones
    if (empty($titulo) || empty($descripcion) || $duracion <= 0 || $precio < 0 || empty($ubicacion) || empty($categoria) || empty($fecha) || empty($hora)) {
        return false;
    }

    // Comprobar si la actividad ya existe
    if (!self::comprobarActividadExistente(null, $titulo, $fecha, $hora)) {

        // Si no existe, proceder con la creación
        $conexion = Conexion::conectar();
        $query = "INSERT INTO actividades (titulo, descripcion, duracion, ubicacion, precio, categoria, fecha, hora) VALUES (:titulo, :descripcion, :duracion, :ubicacion, :precio, :categoria, :fecha, :hora)";
        $stmt = $conexion->prepare($query);
        $stmt->bindParam(':titulo', $titulo);
        $stmt->bindParam(':descripcion', $descripcion);
    }
```

Prueba:

```

// Crear una actividad correctamente
public function testCrearActividad()
{
    $resultado = Actividades::crearActividad("Spinning", "Clase activa", 2, "Sala A", 20, "Bienestar", "2024-12-10", "10:00");
    $this->assertTrue($resultado); // Se espera que la actividad sea creada con éxito
}

// Intentar crear actividad con datos inválidos
public function testCrearActividadConDatosInvalidos()
{
    $resultado = Actividades::crearActividad("", "", -1, "", -5, "", "2024-12-10", "10:00");
    $this->assertFalse($resultado); // Se espera que falle debido a los datos no válidos
}

```

Editar:

Método:

```

// Editar una actividad existente
public static function editarActividad($id, $titulo, $descripcion, $duracion, $ubicacion, $precio, $categoria, $fecha, $hora) {
    // Validaciones de entrada
    if (empty($titulo) || empty($descripcion) || $duracion <= 0 || $precio < 0 || empty($ubicacion) || empty($categoria) || empty($fecha) || empty($hora)) {
        return false; // Devuelve false si los parámetros no son válidos
    }

    // Comprobar si la actividad con el ID existe
    if (!self::comprobarActividadExistente($id)) {
        return false; // La actividad no existe
    }

    // Procedemos a actualizar la actividad
    $conexion = Conexion::conectar();
    $query = "UPDATE actividades SET titulo = :titulo, descripcion = :descripcion, duracion = :duracion, ubicacion = :ubicacion, precio = :precio, categoria = :categoria, fecha = :fecha, hora = :hora WHERE id = :id";
    $stmt = $conexion->prepare($query);
    $stmt->bindParam(':titulo', $titulo);
    $stmt->bindParam(':descripcion', $descripcion);
    $stmt->bindParam(':duracion', $duracion);
    $stmt->bindParam(':ubicacion', $ubicacion);
    $stmt->bindParam(':precio', $precio);
    $stmt->bindParam(':categoria', $categoria);
    $stmt->bindParam(':fecha', $fecha);
    $stmt->bindParam(':hora', $hora);
    $stmt->bindParam(':id', $id);
    return $stmt->execute(); // Devuelve el resultado de la ejecución del UPDATE
}

```

Prueba:

```

// Editar una actividad existente
public function testEditarActividad()
{
    // Editar la actividad
    $resultado = Actividades::editarActividad(30, "Clase de Natación", "Clase más intensa", 2, "Sala B", 25, "Bienestar", "2024-12-10", "12:00");
    $this->assertTrue($resultado); // Se espera que la edición sea exitosa
}

// Intentar editar una actividad inexistente
public function testEditarActividadInexistente()
{
    $resultado = Actividades::editarActividad(999, "Clase de Yoga", "Clase relajante", 2, "Sala A", 20, "Bienestar", "2024-12-10", "10:00");
    $this->assertFalse($resultado); // Se espera que falle, no existe la actividad con ID 999
}

```

Eliminar:

Método:

```

// Eliminar una actividad
public static function eliminarActividad($id) {
    // Comprobar si la actividad con el ID existe
    if (!self::comprobarActividadExistente($id)) {
        return false; // La actividad no existe
    }

    // Procedemos a eliminar la actividad
    $conexion = Conexion::conectar();
    $query = "DELETE FROM actividades WHERE id = :id";
    $stmt = $conexion->prepare($query);
    $stmt->bindParam(':id', $id);
    return $stmt->execute();
}

```

Prueba:

```

// Eliminar una actividad existente
public function testEliminarActividad()
{
    // Crear una actividad

    // Eliminar la actividad
    $resultado = Actividades::eliminarActividad(31);
    $this->assertTrue($resultado); // Se espera que la actividad sea eliminada correctamente
}

// Intentar eliminar una actividad inexistente
public function testEliminarActividadInexistente()
{
    $resultado = Actividades::eliminarActividad(999);
    $this->assertFalse($resultado); // Se espera que falle, no existe la actividad con ID 999
}

```

Obtener todas las actividades:

Método:

```

// Obtener todas las actividades
public static function obtenerActividades() {
    $conexion = Conexion::conectar();
    $query = "SELECT * FROM actividades";
    $stmt = $conexion->prepare($query);
    $stmt->execute();
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

```

Prueba:

```

// Obtener todas las actividades
public function testObtenerActividades()
{

    $actividades = Actividades::obtenerActividades();

    // Verifica que se devuelvan exactamente dos actividades
    $this->assertCount(2, $actividades);
}

```

Obtener por id:

Método:

```

// Obtener una actividad por ID
public static function obtenerActividadPorId($id) {
    $conexion = Conexion::conectar();
    $query = "SELECT * FROM actividades WHERE id = :id";
    $stmt = $conexion->prepare($query);
    $stmt->bindParam(':id', $id);
    $stmt->execute();
    $actividad = $stmt->fetch(PDO::FETCH_ASSOC);
    if ($actividad) {
        return $actividad;
    }
    return false;
}

```

Prueba:

```

// Obtener actividad por ID
public function testObtenerActividadPorId()
{

    $actividad = Actividades::obtenerActividadPorId(30);

    // Verifica que la actividad devuelta tenga el id 1
    $this->assertNotFalse($actividad); // Se espera que no devuelva false
    $this->assertEquals(30, $actividad['id']); // Se espera que devuelva la actividad con ID 1
}

```

Ejecución pruebas:










```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit tests\Modelo\ActividadesTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```






```
.....                                     8 / 8 (100%)
```

```
Time: 00:00.048, Memory: 4.00 MB
```

```
OK (8 tests, 9 assertions)
```

```
C:\xampp\htdocs\proyecto>
```

| | | | | | | | | | |
|--------------------------|--|--|--|-----------------------|---|-------|---|----------------------------------|---------------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 9 Observación de Aves | Participa en una excursión para observar aves en s... | 50.00 | 5 | 4.00 Reserva Natural de los Aves | 2024-10-01 06:00:00 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 30 Clase de Natación | Clase más intensa | 25.00 | 0 | 2.00 Sala B | 2024-12-10 12:00:00 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 32 Spinning | Clase activa | 20.00 | 0 | 2.00 Sala A | 2024-12-10 10:00:00 |

 ☐ Seleccionar todo Para los elementos que están marcados:  Editar  Copiar  Borrar  Exportar

Como se puede comprobar se ha creado el registro 32 (spinning), se ha eliminado el 31 (clase de pilates) y se ha editado el 30 a clase de natación

Clase categorías

Tabla antes de los cambios:

|  | | | | id | nombre |
|---|--|--|--|----|-------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 1 | Cultura |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 2 | Aventura |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 3 | Gastronomía |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 4 | Relajación |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 5 | Naturaleza |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 35 | test |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 36 | test2 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 37 | test3 |

 ☐ Seleccionar todo Para los elementos que están marcados:  Editar

☐ Mostrar todo | Número de filas: Filtrar filas:

Crear:

Método:

```
public static function crearCategoria($nombre) {  
    // Validaciones  
    if (empty($nombre)) {  
        return false; // Nombre vacío no es válido  
    }  
  
    // Comprobar si la categoría ya existe  
    if (self::comprobarCategoriaExistente(null, $nombre)) {  
        return false; // La categoría ya existe  
    }  
  
    // Si no existe, proceder con la creación  
    $conexion = Conexion::conectar();  
    $stmt = $conexion->prepare("INSERT INTO categorias (Nombre) VALUES (:nombre)");  
    $stmt->bindParam(':nombre', $nombre);  
    return $stmt->execute();  
}
```

Prueba:

```
// Test para la creación de una categoría  
public function testCrearCategoria()  
{  
    // Crear la categoría  
    $resultado = Categorias::crearCategoria('test4');  
  
    // Verificar que la categoría se haya creado correctamente  
    $this->assertTrue($resultado);  
}  
  
// Test para intentar crear una categoría con un nombre duplicado  
public function testCrearCategoriaDuplicada()  
{
```


Editar:

Método:

```
// Modificar una categoría existente
public static function modificarCategoria($id, $nombre) {
    // Validaciones
    if (empty($nombre)) {
        return false; // Nombre vacío no es válido
    }

    // Comprobar si la categoría con el ID existe
    if (!$self::comprobarCategoriaExistente($id)) {
        return false; // La categoría no existe
    }

    // Procedemos a modificar la categoría
    $conexion = Conexion::conectar();
    $stmt = $conexion->prepare("UPDATE categorias SET nombre = :nombre WHERE id = :id");
    $stmt->bindParam(':nombre', $nombre);
    $stmt->bindParam(':id', $id);
    return $stmt->execute();
}
```

Prueba:

```
// Test para modificar una categoría existente
public function testModificarCategoria()
{
    // Modificar la categoría
    $resultado = Categorias::modificarCategoria(35, 'Prueba categoría');

    // Verificar que la modificación fue exitosa
    $this->assertTrue($resultado);
}

// Test para intentar modificar una categoría que no existe
public function testModificarCategoriaNoExistente()
{
    $resultado = Categorias::modificarCategoria(999, 'Categoría inexistente');

    // Verificar que no se puede modificar una categoría inexistente
    $this->assertFalse($resultado);
}
```

Eliminar:

Método:

```
// Eliminar una categoría por ID
public static function eliminarCategoria($id) {
    // Comprobar si la categoría con el ID existe
    if (!self::comprobarCategoriaExistente($id)) {
        return false; // La categoría no existe
    }

    // Procedemos a eliminar la categoría
    $conexion = Conexion::conectar();
    $stmt = $conexion->prepare("DELETE FROM categorias WHERE id = :id");
    $stmt->bindParam(':id', $id);
    return $stmt->execute();
}
```

Prueba:

```
// Test para eliminar una categoría existente
public function testEliminarCategoria()
{
    // Eliminar la categoría
    $resultado = Categorias::eliminarCategoria(37);

    // Verificar que la categoría fue eliminada correctamente
    $this->assertTrue($resultado);
}

// Test para intentar eliminar una categoría que no existe
public function testEliminarCategoriaNoExistente()
{
    $resultado = Categorias::eliminarCategoria(999); // ID no existente
    // Verificar que no se puede eliminar una categoría inexistente
    $this->assertFalse($resultado);
}
```

Obtener todas las categorías:

Método:

```
// Obtener todas las categorías
public static function obtenerCategorias() {
    $conexion = Conexion::conectar();
    $stmt = $conexion->prepare("SELECT * FROM categorias");
    $stmt->execute();
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

Prueba:

```
// Obtener todas las categorías
public function testObtenerCategoria()
{
    $resultado = Categorias::obtenerCategorias();

























    // Verifica que se devuelvan exactamente dos actividades
    $this->assertCount(8, $resultado);
}
```


Ejecución pruebas:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit tests\Modelo\CategoriasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.....                                     7 / 7 (100%)
Time: 00:00.027, Memory: 4.00 MB

OK (7 tests, 7 assertions)
C:\xampp\htdocs\proyecto>
```


| ← T → | | | | ▼ id | nombre |
|--------------------------|--|--|--|------|------------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 1 | Cultura |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 2 | Aventura |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 3 | Gastronomía |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 4 | Relajación |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 5 | Naturaleza |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 35 | Prueba categoria |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 36 | test2 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 38 | test4 |


☐ Seleccionar todo
 Para los elementos que están marcados

Como se puede comprobar se ha creado el registro 38 (test4), se ha eliminado el 37 (test 3) y se ha editado el 35 a clase de natación

Clase favoritos

Tabla antes de los cambios:

| id | usuario_id | actividad_id |
|---|------------|--------------|
| Operaciones sobre los resultados de la consulta | | |
|  Crear vista | | |

Crear:

método:

```
// Agregar una actividad a los favoritos
public static function agregarFavorito($usuario_id, $actividad_id)
{
    // Comprobar si el usuario y la actividad existen y son favoritos
    if (!self::comprobarUsuarioExistente($usuario_id) || !self::comprobarActividadExistente($actividad_id) || self::comprobarFavoritoExistente($usuario_id, $actividad_id)) {
        return false;
    }

    // Si no es favorito, proceder a agregarlo
    $conexion = Conexion::conectar();
    $query = "INSERT INTO favoritos (usuario_id, actividad_id) VALUES (:usuario_id, :actividad_id)";
    $stmt = $conexion->prepare($query);
    $stmt->bindParam(":usuario_id", $usuario_id);
    $stmt->bindParam(":actividad_id", $actividad_id);
    return $stmt->execute();
}
```

Prueba:

```
// Test para agregar un favorito correctamente
public function testAgregarFavorito()
{
    // Agregar a favoritos
    $resultado = Favoritos::agregarFavorito(5, 1);

    // Verificar que se ha agregado correctamente
    $this->assertTrue($resultado);
}

// Test para intentar agregar un favorito que ya existe
public function testAgregarFavoritoExistente()
{
    // Intentar agregar el mismo favorito de nuevo
    $resultado = Favoritos::agregarFavorito(5, 1);

    // Verificar que no se ha agregado porque ya existe
    $this->assertFalse($resultado);
}
```

Eliminar:

Método:

```
// Eliminar una actividad de los favoritos
public static function eliminarFavorito($usuario_id, $actividad_id)
{
    // Comprobar si el usuario y la actividad existen
    // Comprobar si el usuario y la actividad existen y son favoritos
    if ( !$this::comprobarFavoritoExistente($usuario_id, $actividad_id) ) {
        return false;
    }

    // Si es un favorito, proceder a eliminarlo
    $conexion = Conexion::conectar();
    $query = "DELETE FROM favoritos WHERE usuario_id = :usuario_id AND actividad_id = :actividad_id";
    $stmt = $conexion->prepare($query);
    $stmt->bindParam(':usuario_id', $usuario_id);
    $stmt->bindParam(':actividad_id', $actividad_id);
    return $stmt->execute();
}
```

Prueba:

```
// Test para eliminar un favorito correctamente
public function testEliminarFavorito()
{
    // Agregar a favoritos
    Favoritos::agregarFavorito(5, 2);

    // Eliminar el favorito
    $resultado = Favoritos::eliminarFavorito(5, 2);

    // Verificar que el favorito fue eliminado
    $this->assertTrue($resultado);
}
```

Ver si una actividad está en favoritos:

Método:

```
// Comprueba si una actividad es favorita para un usuario
public static function esFavorito($usuario_id, $actividad_id)
{
    // Comprobar si el usuario y la actividad existen
    if (!self::comprobarUsuarioExistente($usuario_id) || !self::comprobarActividadExistente($actividad_id)) {
        return false; // Si no existe el usuario o la actividad, no puede ser favorito
    }

    return self::comprobarFavoritoExistente($usuario_id, $actividad_id);
}
```

Prueba:

```
// Test para comprobar si una actividad es favorita
public function testEsFavorito()
{
    // Verificar que ahora es favorito
    $this->assertTrue(Favoritos::esFavorito(5, 1));
}
```

Resultado:

| <div><div><div>↩</div><div>T</div><div>→</div></div><div>▼</div></div> | | | | | id | usuario_id | actividad_id |
|---|---|---|--|--|----|------------|--------------|
| <div><div><div><div><div></div></div></div><div><div>✎</div><div>Editar</div></div><div><div><div></div><div>+</div></div><div><div>📄</div><div>Copiar</div></div><div><div><div></div><div>−</div></div><div><div>🗑</div><div>Borrar</div></div></div></div><div>1</div></div></div> | 5 | 1 | | | | | |

A partir de este punto voy a realizar menos comentarios en los métodos referente a la inserción, obtención y eliminación de datos, ya que es la misma metodología a la existente en las 3 clases vistas y mi proyecto cuenta con 5 clases más. Dicho esto prodigo con la clase comentarios:

Comentarios

Esta tabla al igual que la anterior estaba vacia antes de realizar las pruebas

Creación:

```
// Test para crear un comentario correctamente
public function testCrearComentario()
{
    $usuario_id = 5;
    $actividad_id = 1;
    $comentario = "Este es un comentario de prueba";
    $puntuacion = 5;









    // Crear el comentario
    $resultado = Comentarios::crearComentario($usuario_id, $actividad_id, $comentario, $puntuacion);

    // Verificar que el comentario fue creado correctamente
    $this->assertTrue($resultado);
}

// Test para intentar crear un comentario con un usuario y actividad inexistentes
public function testCrearComentarioConUsuarioInexistente()
{
    $usuario_id = 999; // ID de usuario no existente
    $actividad_id = 100;
    $comentario = "Este es un comentario de prueba";
    $puntuacion = 5;

    // Intentar crear un comentario con un usuario que no existe
    $resultado = Comentarios::crearComentario($usuario_id, $actividad_id, $comentario, $puntuacion);

    // Verificar que no se puede crear el comentario
    $this->assertFalse($resultado);
}
```

| | | id | usuario_id | actividad_id | comentario | puntuacion | fecha |
|---|--|--|------------|--------------|---------------------------------|------------|---------------------|
| <input type="checkbox"/> |  Editar  Copiar  Borrar | 1 | 5 | 1 | Este es un comentario de prueba | 5 | 2024-12-09 22:22:35 |
|  <input type="checkbox"/> Seleccionar todo | | Para los elementos que están marcados:  Editar  Copiar  Borrar  Exportar | | | | | |

Eliminación:

| | | id | usuario_id | actividad_id | comentario | puntuacion | fecha |
|--------------------------|--|----|------------|--------------|---------------------------------|------------|---------------------|
| <input type="checkbox"/> |  Editar  Copiar  Borrar | 2 | 5 | 1 | Este es un comentario de prueba | 5 | 2024-12-09 22:23:28 |
| <input type="checkbox"/> |  Editar  Copiar  Borrar | 3 | 5 | 1 | Este es un comentario de prueba | 5 | 2024-12-09 22:23:51 |

```

// Test para obtener los comentarios de una actividad
public function testObtenerComentariosPorActividad()
{
    // Definir una actividad de prueba que tenga comentarios
    $actividad_id = 1;

    // Obtener los comentarios asociados a esta actividad
    $comentarios = Comentarios::obtenerComentariosPorActividad($actividad_id);

    // Verificar que se ha devuelto un arreglo
    $this->assertIsArray($comentarios);

    // Verificar que los comentarios corresponden a la actividad y tienen datos
    if (count($comentarios) > 0)
    {
        $this->assertArrayHasKey('usuario', $comentarios[0]); // Verifica que el nombre del usuario está presente
        $this->assertArrayHasKey('comentario', $comentarios[0]); // Verifica que el comentario está presente
    }
}

```

```

// Test para obtener los comentarios de una actividad que no existe
public function testObtenerComentariosPorActividadNoExistente()
{
    // Actividad que no existe
    $actividad_id = 999;

    // Obtener los comentarios asociados a esta actividad
    $comentarios = Comentarios::obtenerComentariosPorActividad($actividad_id);

    // Verificar que se devuelve un arreglo vacío
    $this->assertEmpty($comentarios);
}

```

```

// Test para obtener la puntuación promedio de una actividad
public function testObtenerPuntuacionPromedio()
{
    // Definir una actividad de prueba con comentarios
    $actividad_id = 1;

    // Obtener la puntuación promedio de la actividad
    $puntuacionPromedio = Comentarios::obtenerPuntuacionPromedio($actividad_id);

    // Verificar que la puntuación promedio es un número
    $this->assertIsFloat($puntuacionPromedio);
    $this->assertGreaterThanOrEqual(0, $puntuacionPromedio); // Asegurarse de que la puntuación es positiva o cero
}

```

```

// Test para obtener la puntuación promedio de una actividad que no tiene comentarios
public function testObtenerPuntuacionPromedioActividadSinComentarios()
{
    // Actividad sin comentarios
    $actividad_id = 999;

    // Obtener la puntuación promedio de la actividad
    $puntuacionPromedio = Comentarios::obtenerPuntuacionPromedio($actividad_id);

    // Verificar que la puntuación promedio es 0
    $this->assertEquals(0, $puntuacionPromedio);
}

// Test para obtener un comentario específico por su ID
public function testObtenerComentarioPorId()
{
    // Definir un comentario que existe en la base de datos
    $comentario_id = 2;

    // Obtener el comentario por su ID
    $comentario = Comentarios::obtenerComentarioPorId($comentario_id);

    // Verificar que el comentario ha sido encontrado
    $this->assertNotNull($comentario);
    $this->assertArrayHasKey('comentario', $comentario); // Verifica que el campo 'comentario' está presente
}

```

```

// Test para intentar obtener un comentario que no existe
public function testObtenerComentarioPorIdNoExistente()
{
    // Comentario que no existe
    $comentario_id = 999;

    // Intentar obtener el comentario por su ID
    $comentario = Comentarios::obtenerComentarioPorId($comentario_id);

    // Verificar que no se ha encontrado el comentario
    $this->assertFalse($comentario);
}

```

```

C:\xampp\htdocs\proyecto>vendor\bin\phpunit tests\Modelo\ComentariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.....
00%)

Time: 00:00.034, Memory: 4.00 MB

OK (9 tests, 14 assertions)

C:\xampp\htdocs\proyecto>

```

Como se puede observar las pruebas han ido bien, es decir, esta clase al igual que el resto funciona como se esperaba

Clase notificaciones:

```
SELECT * FROM `notificaciones`
```

| id | usuario_id | mensaje | fecha | leida |
|----|------------|---------|-------|-------|
|----|------------|---------|-------|-------|

Operaciones sobre los resultados de la consulta

Crear vista

Crear:

```
// Test para agregar una notificación
public function testAgregarNotificacion()
{
    $usuario_id = 5;
    $mensaje = "Prueba de notificación";

    $resultado = Notificaciones::agregarNotificacion($usuario_id, $mensaje);

    $this->assertTrue($resultado); // Verificar que la inserción fue exitosa
}
```

Marcar como leídas

```
// Test para marcar una notificación como leída
public function testMarcarComoLeida()
{
    $notificacion_id = 1; // Asume que esta notificación existe en la base de datos

    $resultado = Notificaciones::marcarComoLeida($notificacion_id);

    $this->assertTrue($resultado); // Verificar que la operación fue exitosa
}
```

| | id | usuario_id | mensaje | fecha | leída |
|--------------------------|----|------------|------------------------|---------------------|-------|
| <input type="checkbox"/> | 1 | 5 | Prueba de notificación | 2024-12-09 23:03:14 | 1 |

```
// Test para marcar todas las notificaciones como leídas
public function testMarcarTodasComoLeidas()
{
    $usuario_id = 5;

    $resultado = Notificaciones::marcarTodasComoLeidas($usuario_id);

    $this->assertTrue($resultado); // Verificar que todas las notificaciones fueron marcadas como leídas
}
```

| | id | usuario_id | mensaje | fecha | leída |
|--------------------------|----|------------|------------------------|---------------------|-------|
| <input type="checkbox"/> | 1 | 5 | Prueba de notificación | 2024-12-09 23:03:20 | 1 |
| <input type="checkbox"/> | 2 | 5 | Prueba de notificación | 2024-12-09 23:04:24 | 1 |
| <input type="checkbox"/> | 3 | 5 | Prueba de notificación | 2024-12-09 23:09:29 | 1 |

```
// Test para contar las notificaciones no leídas
public function testContarNoLeidas()
{
    $usuario_id = 5;

    $conteo = Notificaciones::contarNoLeidas($usuario_id);

    $this->assertIsInt($conteo); // Verificar que se devuelve un número entero
    $this->assertGreaterThanOrEqual(0, $conteo); // Asegurarse de que el conteo es positivo
}
```

Eliminar:

```
// Test para eliminar una notificación
public function testEliminarNotificacion()
{
    $notificacion_id = 1; // Asume que esta notificación existe en la base de datos

    $resultado = Notificaciones::eliminarNotificacion($notificacion_id);

    $this->assertTrue($resultado); // Verificar que la notificación fue eliminada
}
```

| | | | id | usuario_id | mensaje | fecha | leida |
|--------------------------|--|--|--|------------|--------------------------|---------------------|-------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 2 | 5 Prueba de notificación | 2024-12-09 23:04:24 | 1 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 3 | 5 Prueba de notificación | 2024-12-09 23:09:29 | 1 |

```
// Test para obtener notificaciones filtradas
public function testObtenerNotificacionesFiltradas()
{
    $usuario_id = 5;
    $fecha = date('Y-m-d'); // Usar la fecha actual para el filtro

    $notificaciones = Notificaciones::obtenerNotificacionesFiltradas($usuario_id, $fecha);

    $this->assertIsArray($notificaciones); // Verificar que se devuelve un arreglo

    if (!empty($notificaciones)) {
        $this->assertArrayHasKey('usuario_nombre', $notificaciones[0]); // Verificar que el nombre del usuario está presente
        $this->assertArrayHasKey('mensaje', $notificaciones[0]); // Verificar que el mensaje está presente
    }
}

// Test para obtener notificaciones filtradas sin parámetros
public function testObtenerNotificacionesFiltradasSinParametros()
{
    $notificaciones = Notificaciones::obtenerNotificacionesFiltradas();

    $this->assertIsArray($notificaciones); // Verificar que se devuelve un arreglo

    if (!empty($notificaciones)) {
        $this->assertArrayHasKey('usuario_nombre', $notificaciones[0]); // Verificar que el nombre del usuario está presente
        $this->assertArrayHasKey('mensaje', $notificaciones[0]); // Verificar que el mensaje está presente
    }
}
```


Clase reservas

✔ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas)

```
SELECT * FROM `reservas`
```

id itinerario_id usuario_id fecha_reserva estado monto

Operaciones sobre los resultados de la consulta

 Crear vista

Pruebas:

| ID | Método Probado | Descripción | Datos | Resultado Esperado |
|----|----------------------------|--|--|----------------------------------|
| | | Caso | Entrada | |
| 1 | crearReserva | Crear una reserva con datos válidos. | itinerario_id=20, usuario_id=7, monto=100.50 | Devuelve un ID numérico. |
| 2 | crearReserva | Crear una reserva Con itinerario_id nulo. | itinerario_id=null, usuario_id=7, monto=100.50 | Devuelve false. |
| 3 | obtenerReservas PorUsuario | Obtener reservas de un usuario que tiene reservas. | usuario_id=7 | Devuelve un arreglo de reservas. |
| 4 | obtenerReservas PorUsuario | Obtener reservas de un usuario inexistente. | usuario_id=9999 | Devuelve un arreglo vacío. |

| | | | | |
|----|-------------------------|---|---|---|
| | | | Reserva | |
| 5 | obtenerReserva PorId | Obtener una reserva existente por su ID. | id=último ID creado dinámicamente | Devuelve los datos de la reserva. |
| 6 | obtenerReserva PorId | Obtener una reserva inexistente por su ID. | reserva_id=9999 reserva_id= | Devuelve null. |
| 7 | Actualizar Estado | Actualizar el estado de una reserva con un estado válido. | último ID creado dinámicamente, nuevoEstado= 'confirmada' reserva_id= | Devuelve true. |
| 8 | Actualizar Estado | Actualizar el estado de una reserva con un estado inválido. | último ID creado dinámicamente, nuevoEstado= 'inexistente' | Devuelve false. |
| 9 | Cancelar Reserva | Cancelar reserva existente. | reserva_id= último ID creado dinámicamente | Devuelve true. |
| 10 | Cancelar Reserva | Cancelar reserva inexistente. | reserva_id=9999 | Devuelve false. |
| 11 | Eliminar Reserva | Eliminar reserva existente. | reserva_id= último ID creado dinámicamente | Devuelve true. |
| 12 | Eliminar | Intentar eliminar | reserva_id=9999 | Devuelve false. |

| | | | | |
|----|--------------------------------------|--|------------------------------------|---|
| | Reserva | reserva inexistente. | | |
| 13 | Obtener CosteTotal Actividades | Obtener el costo total actividades asociadas a un itinerario existente con actividades. | itinerario_id=20 | Devuelve un número flotante mayor que 0. |
| 14 | Obtener CosteTotal Actividades | Obtener el costo total actividades asociadas a un itinerario sin actividades. | itinerario_id=9999 | Devuelve 0. |
| 16 | Actualizar Presupuesto | Actualizar presupuesto de itinerario existente. | itinerario_id=20, monto=50.00 | Devuelve true. |
| 17 | Actualizar Presupuesto | Intentar actualizar presupuesto de itinerario inexistente. | itinerario_id=9999, monto=30.00 | Devuelve false. |
| 18 | Obtener Presupuesto | Obtener presupuesto de itinerario existente. | itinerario_id=20 | Devuelve un número flotante mayor o igual que 0. |
| 19 | Obtener Presupuesto | Intentar obtener presupuesto de itinerario inexistente. | itinerario_id=9999 | Devuelve 0. |

Ejecución pruebas:

Test Crear Reserva




```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testCrearReserva tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.022, Memory: 4.00 MB

OK (1 test, 2 assertions)
```







Como se puede ver se ha creado solo la reserva cuyo id no era nulo:






| | | | id | itinerario_id | usuario_id | fecha_reserva | estado | monto | |
|--------------------------|--|--|--|---------------|------------|---------------|---------------------|-----------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 18 | 20 | 7 | 2024-12-14 13:25:37 | pendiente | 100.50 |

Test Obtener Reservas Por Usuario

Se puede apreciar que ha creado la reserva en cuestión y la ha encontrado

Opciones extra

| | | | id | itinerario_id | usuario_id | fecha_reserva | estado | monto | |
|--------------------------|--|--|--|---------------|------------|---------------|---------------------|-----------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 18 | 20 | 7 | 2024-12-14 13:25:37 | pendiente | 100.50 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 19 | 20 | 7 | 2024-12-14 14:01:49 | pendiente | 50.00 |

 ☐ Seleccionar todo Para los elementos que están marcados:  Editar  Copiar  Borrar  Exportar

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtenerReservasPorUsuario tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.045, Memory: 4.00 MB

OK (1 test, 4 assertions)
```

Test Obtener Reserva Por Id

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtenerReservaPorId tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.023, Memory: 4.00 MB

OK (1 test, 4 assertions)
```

Encuentra la reserva con el id proporcionado, no encuentra la reserva inexistente

Test actualizar estado

| | | | | | | | | | |
|--------------------------|--|--|--|----|----|---|---------------------|-----------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 21 | 20 | 7 | 2024-12-14 14:06:00 | pendiente | 150.00 |
|--------------------------|--|--|--|----|----|---|---------------------|-----------|--------|

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testActualizarEstado tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.021, Memory: 4.00 MB

OK (1 test, 2 assertions)

C:\xampp\htdocs\proyecto>
```

| | | | | | | | | | |
|--------------------------|--|--|--|----|----|---|---------------------|------------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 21 | 20 | 7 | 2024-12-14 14:06:00 | confirmada | 150.00 |
|--------------------------|--|--|--|----|----|---|---------------------|------------|--------|

Actualiza a confirmada por que es un estado existente

Cancelar reserva:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testCancelarReserva tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.




.
    1 / 1 (100%)

Time: 00:00.021, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

| | | | | | | |
|--|----|----|---|---------------------|-----------|--------|
|  Borrar | 21 | 20 | 7 | 2024-12-14 14:06:00 | cancelada | 150.00 |
|--|----|----|---|---------------------|-----------|--------|

Test actualizar presupuesto:

| | | | | | | | | | |
|--------------------------|--|--|--|----|---|-------------|--------|---------------------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 20 | 7 | Verano 2025 | FWfrfW | 2024-09-09 20:47:32 | 200.00 |
|--------------------------|--|--|--|----|---|-------------|--------|---------------------|--------|




```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testActualizarPresupuesto tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.

1 / 1 (100%)








Time: 00:00.020, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

| | | | | | | | | | |
|--------------------------|--|--|--|----|---|-------------|--------|---------------------|--------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 20 | 7 | Verano 2025 | FWfrfW | 2024-09-09 20:47:32 | 250.00 |
|--------------------------|--|--|--|----|---|-------------|--------|---------------------|--------|

Como se puede comprobar suma correctamente el monto al presupuesto

Test obtener coste total actividades:

|  | | | | id | itinerario_id | actividad_id |
|---|--|--|--|----|---------------|--------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 29 | 20 | 1 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 28 | 20 | 4 |

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtenerCosteTotalActividades tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.

1 / 1 (100%)

Time: 00:00.021, Memory: 4.00 MB

OK (1 test, 3 assertions)
```

La prueba funciona bien porque comprueba que el coste de las actividades asociadas a un itinerario es mayor a 0

Test obtener Presupuesto:

La prueba funciona bien al igual que en el caso anterior, ya que también comprueba que el presupuesto de un itinerario existente es mayor a 0:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtener
Presupuesto tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.023, Memory: 4.00 MB

OK (1 test, 4 assertions)
C:\xampp\htdocs\proyecto>
```

Test Eliminar Reserva:

Eliminacion de la reserva id 21:

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0006 segundos.)

`SELECT * FROM `reservas``

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: ▼

Opciones extra

| | | | | id | itinerario_id | usuario_id | fecha_reserva | | | |
|--------------------------|---|--------|---|--------|---|------------|---------------|----|---|----------------|
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 18 | 20 | 7 | 2024-12-14 13: |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 19 | 20 | 7 | 2024-12-14 14: |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 21 | 20 | 7 | 2024-12-14 14: |

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testEliminarReserva tests\Modelo\ReservasTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.021, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

</

```
// Confirma una reserva
public static function confirmarReserva($reserva_id)
{
    return self::actualizarEstado($reserva_id, 'confirmada');
}

// Marca una reserva como pagada
public static function pagarReserva($reserva_id)
{
    return self::actualizarEstado($reserva_id, 'pagada');
}
```

Estos 2 métodos no los pruebo por que son exactamente iguales a cancelar reserva, el cual se ha visto que funciona correctamente

Clase usuario:

Estado actual:

| <div><div>←</div><div>→</div></div> | | | id | nombre | clave | telefono | email | tipo | |
|-------------------------------------|---|---|---|--------|--------|--|-----------|-----------------|---------------|
| <input type="checkbox"/> | <div><div><div><div></div></div></div><div>Editar</div></div> | <div><div><div><div></div></div></div><div>Copiar</div></div> | <div><div><div><div></div></div></div><div>Borrar</div></div> | 5 | Martin | \$2y\$10\$pb/1cV9ZhcT6eATfUqyor.gQoeQrITZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | <div><div><div><div></div></div></div><div>Editar</div></div> | <div><div><div><div></div></div></div><div>Copiar</div></div> | <div><div><div><div></div></div></div><div>Borrar</div></div> | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |

Pruebas:

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|-----------------------------|---|--|--|--|
| 1 | testCrearUsuario | Crear un usuario con datos válidos. | Crear un usuario válido con nombre, clave y tipo. | nombre = 'Carlos', clave = 'contraseña_segura', tipo_usuario = 'cliente' | Devuelve true, el usuario se crea correctamente. |
| 2 | testCrearUsuario | Intentar crear un usuario con nombre vacío. | Intentar crear un usuario con nombre vacío. | nombre = '', clave = 'contraseña_segura', tipo_usuario = 'cliente' | Devuelve false, ya que el nombre es vacío. |
| 3 | testObtenerUsuarios | Obtener todos los usuarios. | Obtener la lista de usuarios registrados en la base de datos. | Ninguno (sin parámetros) | Devuelve un array con los usuarios existentes. |
| 4 | testObtenerUsuarioPorNombre | Obtener un usuario por nombre que existe. | Buscar un usuario llamado 'Carlos' que existe en la base de datos. | nombre = 'Carlos' | Devuelve el usuario con nombre 'Carlos'. |

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|-----------------------------|--|---|--|--|
| 5 | testObtenerUsuarioPorNombre | Intentar obtener un usuario por nombre que no existe. | Buscar un usuario llamado 'NoExiste' que no está en la base de datos. | nombre = 'NoExistente' | Devuelve null, no existe un usuario con ese nombre. |
| 7 | testActualizarUsuario | Actualizar los datos de un usuario existente. | Actualizar el nombre, email y teléfono de un usuario con ID 8. | id = 8, nuevo_nombre = 'Antonio', nuevo_email = 'antonio@mail.com', nuevo_telefono = '657915800' | Devuelve true, el usuario se actualiza correctamente. |
| 8 | testActualizarUsuario | Intentar actualizar los datos de un usuario inexistente. | Intentar actualizar un usuario con ID 9999 que no existe. | id = 9999, nuevo_nombre = 'NoExiste', nuevo_email = 'noexiste@mail.com', nuevo_telefono = '0000000000' | Devuelve false, el usuario no existe para actualizar. |
| 9 | testActualizarContraseña | Actualizar la contraseña de un usuario existente. | Actualizar la contraseña del usuario con ID 8. | usuario_id = 8, nueva_contrasena = 'nueva_contrasena_segura' | Devuelve true, la contraseña se actualiza correctamente. |
| 10 | testActualizarContraseña | Intentar actualizar la contraseña de un usuario inexistente. | Intentar actualizar la contraseña de un usuario con ID | usuario_id = 9999, nueva_contrasena = 'nueva_contrasena_segura' | Devuelve false, no se puede actualizar la contraseña de un |

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|---------------------|---|---|---------------|--|
| | | | 9999 que no existe. | | usuario inexistente. |
| 11 | testEliminarUsuario | Eliminar un usuario existente. | Eliminar un usuario con ID 8. | id = 8 | Devuelve true, el usuario se elimina correctamente. |
| 12 | testEliminarUsuario | Intentar eliminar un usuario inexistente. | Intentar eliminar un usuario con ID 9999 que no existe. | id = 9999 | Devuelve false, no se puede eliminar un usuario inexistente. |

Ejecución pruebas:

Test crear usuario

| | | | | id | nombre | clave | telefono | email | tipo |
|--------------------------|--|--|--|----|--------|--|-----------|-----------------|---------------|
| <input type="checkbox"/> | | | | 5 | Martin | \$2y\$10\$pb/1cV9ZhCT6eATfUqyor.gQoeQrITZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | | | | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testCrearUs
uario tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.019, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

| | | | | id | nombre | clave | telefono | email | tipo |
|--------------------------|--|--|--|----|--------|--|-----------|-----------------|---------------|
| <input type="checkbox"/> | | | | 5 | Martin | \$2y\$10\$pb/1cV9ZhCT6eATfUqyor.gQoeQrITZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | | | | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |
| <input type="checkbox"/> | | | | 8 | Carlos | contraseña_segura | NULL | NULL | cliente |

Crea un usuario cuando se pasan los valores correctos, si falta alguno no lo crea

Test obtener usuarios

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtener
Usuarios tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.017, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

Test obtener usuario por nombre:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtener
UsuarioPorNombre tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
1 / 1 (100%)

Time: 00:00.021, Memory: 4.00 MB

OK (1 test, 3 assertions)

C:\xampp\htdocs\proyecto>
```

Test actualizar usuario:

| | | | | id | nombre | clave | telefono | email | tipo |
|--------------------------|--|--|--|----|--------|--|-----------|-----------------|---------------|
| <input type="checkbox"/> | | | | 5 | Martin | \$2y\$10\$pb/1cV9ZhT6eATfUqyor.gQoeQrITZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | | | | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |
| <input type="checkbox"/> | | | | 8 | Carlos | contraseña_segura | NULL | NULL | cliente |

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testActuali
zarUsuario tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
1 / 1 (100%)

Time: 00:00.027, Memory: 4.00 MB

OK (1 test, 2 assertions)

C:\xampp\htdocs\proyecto>
```

Opciones extra

| | | | | id | nombre | clave | telefono | email | tipo |
|--------------------------|--|--|--|----|---------|--|-----------|------------------|---------------|
| <input type="checkbox"/> | | | | 5 | Martin | \$2y\$10\$pb/1cV9ZhT6eATfUqyor.gQoeQrITZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | | | | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |
| <input type="checkbox"/> | | | | 8 | Antonio | contraseña_segura | 657915800 | antonio@mail.com | cliente |

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

Test actualizar contraseña:

Opciones extra

| | | | id | nombre | clave | telefono | email | tipo | |
|--------------------------|--------|--------|--------|--------|---------|--|-----------|------------------|---------------|
| <input type="checkbox"/> | Editar | Copiar | Borrar | 5 | Martin | \$2y\$10\$pb/1cV9ZhcT6eATfUqyor.gQoeQrlTZ0CDiozQ19cmb... | 657915800 | martin@mail.com | administrador |
| <input type="checkbox"/> | Editar | Copiar | Borrar | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNougoKdnFpe/oq804wit2agA... | NULL | NULL | cliente |
| <input type="checkbox"/> | Editar | Copiar | Borrar | 8 | Antonio | contraseña_segura | 657915800 | antonio@mail.com | cliente |

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testActualizarContrasena tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.026, Memory: 4.00 MB

OK (1 test, 2 assertions)

☐ Editar Copiar Borrar 8 Antonio nueva_contrasena_segura 657915800 antonio@mail.com cliente

Test eliminar usuario:

Opciones extra

| | | | | id | nombre | clave |
|--------------------------|--------|--------|--------|----|--------|----------------------------------|
| <input type="checkbox"/> | Editar | Copiar | Borrar | 5 | Martin | \$2y\$10\$pb/1cV9ZhcT6eATfUqyor. |
| <input type="checkbox"/> | Editar | Copiar | Borrar | 7 | Juan | \$2y\$10\$VPesJtD90an8f27EVrFNc |

☐ Seleccionar todo Para los elementos que están marcados: Editar

C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testEliminarUsuario tests\Modelo\UsuariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.












.
1 / 1 (100%)

Time: 00:00.027, Memory: 4.00 MB

OK (1 test, 2 assertions)

Clase Itinerarios

Opciones extra

| | | | | | | | | |
|---|--|--|--|--------|--|---|--|--|
|   |  | id | usuario_id | nombre | descripcion | fecha_creacion | presupuesto | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 20 | 7 | Verano 2025 FWfrfW | 2024-09-09 20:47:32 | 250.00 |
|  | <input type="checkbox"/> Seleccionar todo | Para los elementos que están marcados: | | |  Editar |  Copiar |  Borrar |  Exportar |

Pruebas

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|---------------------------------|---|---|--|--|
| 1 | testCrearOActualizar Itinerario | Crear un itinerario con datos válidos. | Crear un itinerario nuevo. | usuariold = 1, nombre = 'Vacaciones', descripción = 'Viaje a la playa', presupuesto = 1000 te. | Devuelve true, el itinerario se crea correctamente. |
| 2 | testCrearOActualizar Itinerario | Actualizar un itinerario existente. | Actualizar un itinerario existente con el mismo nombre. | usuariold = 1, nombre = 'Vacaciones', descripción = 'Viaje al Caribe', presupuesto = 2000 te. | Devuelve true, el itinerario se actualiza correctamente. |
| 3 | testCrearOActualizar Itinerario | Intentar crear un itinerario con datos incompletos. | Crear un itinerario sin presupuesto. | usuariold = 1, nombre = 'Aventura', descripción = 'Excursión', presupuesto = null | Devuelve false, el itinerario no se crea debido a datos incompletos. |
| 4 | testObtenerItinerario PorId | Obtener un itinerario por su ID. | Obtener el itinerario con ID 2. | id = 2 | Devuelve el itinerario con ID 2. |

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|-------------------------------------|--|---|-------------------------------------|--|
| 5 | testObtenerActividadesPorItinerario | Obtener actividades asociadas a un itinerario. | Obtener actividades del itinerario con ID 2. | itinerario_id = 2 | Devuelve un array con las actividades asociadas al itinerario. |
| 6 | testAgregarActividad | Agregar una actividad a un itinerario existente. | Agregar la actividad con ID 5 al itinerario con ID 2. | itinerario_id = 2, actividad_id = 5 | Devuelve true, la actividad se agrega correctamente al itinerario. |
| 7 | testEliminarActividad | Eliminar una actividad de un itinerario. | Eliminar la actividad con ID 5 del itinerario con ID 2. | itinerario_id = 2, actividad_id = 5 | Devuelve true, la actividad se elimina correctamente del itinerario. |
| 8 | testEliminarItinerario | Eliminar un itinerario existente. | Eliminar el itinerario con ID 3. | id = 3 | Devuelve true, el itinerario se elimina correctamente. |
| 9 | testEliminarItinerario | Intentar eliminar un itinerario inexistente. | Intentar eliminar el itinerario con ID 9999. | id = 9999 | Devuelve false, el itinerario no se puede eliminar porque no existe. |
| 10 | testContarActividades | Contar el número de actividades | Contar actividades del | itinerario_id = 2 | Devuelve un número |

| ID | Método Probado | Descripción | Caso | Datos Entrada | Resultado Esperado |
|----|---------------------------------|--|--|---|--|
| 11 | testObtenerItinerariosFiltrados | asociadas a un itinerario. | itinerario con ID 2. | | mayor o igual a 0. |
| | | Obtener itinerarios filtrados por usuario y fecha de creación. | Filtrar itinerarios del usuario con ID 1 y fecha = '2024-12-01'. | usuario_id = 1, fecha_creacion = '2024-12-01' | Devuelve un array con los itinerarios filtrados por usuario y fecha. |

Test crear o actualizar itinerario

Opciones extra

| | id | usuario_id | nombre | descripcion | fecha_creacion | presupuesto |
|---|----|------------|-------------|-------------|---------------------|-------------|
| <input type="checkbox"/> Editar Copiar Borrar | 20 | 7 | Verano 2025 | FWfrfW | 2024-09-09 20:47:32 | 250.00 |

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testCrearOActulizarItinerario tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.029, Memory: 4.00 MB

OK (1 test, 3 assertions)
C:\xampp\htdocs\proyecto>
```

Opciones extra

| | id | usuario_id | nombre | descripcion | fecha_creacion | presupuesto |
|---|----|------------|-------------|-----------------|---------------------|-------------|
| <input type="checkbox"/> Editar Copiar Borrar | 20 | 7 | Verano 2025 | FWfrfW | 2024-09-09 20:47:32 | 250.00 |
| <input type="checkbox"/> Editar Copiar Borrar | 21 | 7 | Vacaciones | Viaje al Caribe | 2024-12-15 18:20:25 | 2000.00 |

Crea el itinerario, lo actualiza después, pero no crea el itinerario al que le faltan datos

Test Obtener itinerario por id

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtener
ItinerarioPorId tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.026, Memory: 4.00 MB

OK (1 test, 2 assertions)

C:\xampp\htdocs\proyecto>
```

Test agregar Actividad

Opciones extra

| <div><div><div>←</div><div>T</div><div>→</div></div></div> | | | | | id | itinerario_id | actividad_id |
|--|--|--|--|----|----|---------------|--------------|
| <div><div><div></div></div></div> | <div><div><div></div></div></div> Editar | <div><div><div></div></div></div> Copiar | <div><div><div></div></div></div> Borrar | 29 | 20 | 1 | |
| <div><div><div></div></div></div> | <div><div><div></div></div></div> Editar | <div><div><div></div></div></div> Copiar | <div><div><div></div></div></div> Borrar | 28 | 20 | 4 | |

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testAgregar
Actividad tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.

.
    1 / 1 (100%)

Time: 00:00.025, Memory: 4.00 MB










OK (1 test, 1 assertion)

C:\xampp\htdocs\proyecto>
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)]

☐ Mostrar todo | Número de filas: 25 ▾

Opciones extra

| <div><div><div>←</div><div>T</div><div>→</div></div><div></div></div> | | | | | | id | itinerario_id | actividad_id |
|---|--|--|--|----|----|----|---------------|--------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 29 | 20 | 1 | | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 28 | 20 | 4 | | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 36 | 21 | 5 | | |

Test eliminar actividad:








```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testEliminarA tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
 1 / 1 (100%)
```

```
Time: 00:00.028, Memory: 4.00 MB
```

```
OK (1 test, 1 assertion)
```

Opciones extra

|  | | | | id | itinerario_id | actividad_id |
|---|--|--|--|----|---------------|--------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 29 | 20 | 1 |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 28 | 20 | 4 |

Test contar actividades:

```
public function testContarActividades()
{
    $itinerarioid = 21;
    $resultado = Itinerarios::contarActividades($itinerarioid);
    $this->assertIsInt($resultado);
    $this->assertGreaterThanOrEqual(0, $resultado);
}
```

Comprueba que el itinerario actualmente no tiene actividades:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testContarActividades tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
 1 / 1 (100%)
```

```
Time: 00:00.029, Memory: 4.00 MB
```

```
OK (1 test, 3 assertions)
```

En efecto, funciona correctamente

Test Obtener Itinerarios Filtrados

Obtiene el itinerario creado en esa fecha y por ese usuario:

```
public function testObtenerItinerariosFiltrados()
{
    $usuarioId = 7;
    $fechaCreacion = '2024-09-09';
    $resultado = Itinerarios::obtenerItinerariosFiltrados($usuarioId, $fechaCreacion);
    $this->assertIsArray($resultado);
}
```

```
public function testObtenerItinerariosFiltrados()
{
    $usuarioId = 7;
    $fechaCreacion = '2024-09-09';
    $resultado = Itinerarios::obtenerItinerariosFiltrados($usuarioId, $fechaCreacion);
    $this->assertIsArray($resultado);
}
```

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testObtener
ItinerariosFiltrados tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
  1 / 1 (100%)

Time: 00:00.025, Memory: 4.00 MB

OK (1 test, 1 assertion)
```

Test eliminar Itinerario:

```
C:\xampp\htdocs\proyecto>vendor\bin\phpunit --filter testElimina
rItinerario tests\Modelo\ItinerariosTest.php
PHPUnit 9.5.0 by Sebastian Bergmann and contributors.
```

```
.
  1 / 1 (100%)

Time: 00:00.026, Memory: 4.00 MB

OK (1 test, 2 assertions)
```

Opciones extra

| | | | | id | usuario_id | nombre | descripcion | fecl |
|--------------------------|--|--|--|----|------------|----------------|-------------|------|
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 20 | 7 | Verano 2025 | FWfrrW | 202 |

☐ Seleccionar todo
 Para los elementos que están marcados:  Editar

He estado intentando realizar pruebas en los diferentes controladores durante varios días, pero me he encontrado con conflictos relacionados con las cabeceras al utilizar sesiones en las pruebas con PHPUnit. No estoy seguro de cómo proceder, ya que nunca hemos trabajado con PHPUnit para probar cabeceras y sesiones. He estado investigando sobre el tema, pero aún no he logrado encontrar una solución.

De todas formas, se puede comprobar que los métodos funcionan correctamente, y en la web todo está operando sin problemas.