



Actividad Áulica 26

Desafío bug

CARRERA: Ingeniería en sistemas de Información

MATERIA: Paradigmas y Lenguajes de Programación III

COMISIÓN: "U" (única)

PROFESOR: Mgter. Ing. Encina Agustín.

ESTUDIANTE: Szevaga Emanuel,

FECHA: 05-11-2025



Detección de errores

El principal problema se encuentra en la inconsistencia del manejo del estado de la tarea (completada) entre el código PHP y la base de datos MySQL.

1. Inconsistencia de Tipos de Datos (Task.php): La columna completada es TINYINT(1) (espera 0 o 1), pero los métodos en Task.php (como getAllTasks y addTask) intentan usar strings como 'pending' o 'completed'.
2. Columna Faltante (Task.php y todo_app.sql): El método completeTask intenta actualizar una columna llamada completed_at (Línea 28 en Task.php), la cual no existe en el CREATE TABLE de todo_app.sql. Además, script.js espera recibir esta columna para las tareas completadas.

Detalles de las modificaciones

1. Archivo: mvc-todolist-completo/todo_app.sql
 - Problema: La columna completed_at no existía, causando un error de SQL al intentar completar una tarea.
 - Solución Aplicada (Modificación en CREATE TABLE):

Antes:

```
CREATE TABLE `tareas` (
  `id` int(11) NOT NULL,
  `descripcion` varchar(255) NOT NULL,
```



```
'completada' tinyint(1) NOT NULL DEFAULT 0,  
'fecha_creacion' timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Después:

```
CREATE TABLE `tareas` (  
    `id` int(11) NOT NULL,  
    `descripcion` varchar(255) NOT NULL,  
    `completada` tinyint(1) NOT NULL DEFAULT 0,  
    `fecha_creacion` timestamp NOT NULL DEFAULT current_timestamp(),  
    `completed_at` datetime DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

2. Archivo: mvc-todolist-completo/model/Task.php

- Problema: Inconsistencia en el tipo de dato del estado (completada) en las consultas.
- Solución Aplicada: Correcciones en la clase task:

```
<?php  
  
class Task {  
  
    private $pdo;
```



// ... __construct ...

// Solución 3: Usar 0 (pendiente) o 1 (completada) en lugar de strings

```
public function getAllTasks($status = 0) {  
  
    $stmt = $this->pdo->prepare("SELECT * FROM tareas WHERE completada = ?  
        ORDER BY id DESC");  
  
    $stmt->execute([$status]);  
  
    return $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
}
```

// Solución 4: Usar 0 (pendiente) en lugar de string 'pending'

```
public function addTask($description, $status = 0) {  
  
    $stmt = $this->pdo->prepare("INSERT INTO tareas (descripcion, completada)  
        VALUES (?,?)");  
  
    return $stmt->execute([$description, $status]);  
  
}
```

// ... deleteTask ...

// Solución 2: Usar 1 (entero) en lugar de string 'completed'

```
public function completeTask($id) {
```



```
$stmt = $this->pdo->prepare("UPDATE tareas SET completada = 1, completed_at =  
NOW() WHERE id = ?");  
  
return $stmt->execute([$id]);  
  
}  
  
}  
  
?>
```

3. Archivo: mvc-todolist-completo/controller/TaskController.php

- Problema: El controlador pasaba strings ('pending', 'completed') al método getAllTasks, esperando que el modelo interpretara estos valores.
- Solución Aplicada (Corrección en el Bloque GET):

```
} elseif ($_SERVER['REQUEST_METHOD'] === 'GET' && $_GET['action'] === 'getTasks')  
  
{  
  
    // Solución 5: Pasar valores enteros: 0 para pendiente, 1 para completada  
  
    $pendingTasks = $taskModel->getAllTasks(0);  
  
    $completedTasks = $taskModel->getAllTasks(1);  
  
    $response = [  
  
        'status' => 'success',  
  
        'pendingTasks' => $pendingTasks,  
  
        'completedTasks' => $completedTasks  
  
    ];
```



Carrera: Ing. Sistemas de Información
Materia: Paradigmas y Lenguajes de Programación III
Estudiante: Szevaga Emanuel

Profesor: Mgter. Ing. Encina Agustín
Comisión "U" (única)

}