Universidad de la Cuenca del Plata
Autorización Definitiva. Decreto Poder Ejecutivo Nacional N°091/2006

Paradigmas y Lenguajes de Programación III

CARRERA: Ingeniería en Sistemas de Información

MATERIA: Paradigmas y Lenguajes de Programación III COMISIÓN: “U” (única)

PROFESOR: Mgter. Ing. Encina Agustín.

ESTUDIANTE: Gamarra Kiara Barbarella

FECHA: 05-11-2025

Informe de errores encontrados

Task

Archivo editado: Task.php

Sección modificada: Métodos getAllTasks, addTask y completeTask.

Problema detectado:
Los métodos utilizaban cadenas de texto ('pending' y 'completed') como valores para el campo completada. En la base de datos, este campo está definido como un tipo numérico (BOOLEAN o TINYINT), lo que ocasiona errores de compatibilidad y resultados incorrectos al realizar consultas o actualizaciones.

Solución aplicada:
Se reemplazaron los valores de texto por valores numéricos coherentes con el tipo de dato definido en la base de datos:

- 0 para indicar tareas pendientes.
- 1 para indicar tareas completadas.

Además, se actualizó la firma de los métodos para reflejar este cambio en el valor por defecto del parámetro $status.

Problema:
El método hacía referencia a una columna llamada completed_at, la cual no estaba garantizada en la estructura de la base de datos, pudiendo generar el error *"Unknown column 'completed_at' in 'field list'"*.

Solución aplicada:
Se verificó la necesidad del campo y se documentó la instrucción SQL necesaria para agregarlo en la tabla tareas en caso de no existir.

Instrucción aplicada:

ALTER TABLE tareas ADD COLUMN completed_at DATETIME NULL;

Scrip

Archivo editado: script.js

Línea o sección modificada:
Funciones renderTasks, addTask, completeTask y deleteTask.

Problema detectado:
En la función renderTasks, el código hacía referencia al atributo description de los objetos recibidos desde el servidor (task.description). Sin embargo, en la base de datos el campo se denomina

descripcion, lo que provocaba que las tareas se mostraran vacías en la interfaz o que no se renderizaran correctamente.

Solución aplicada:
Se reemplazaron todas las referencias a task.description por task.descripcion para mantener la coherencia con la estructura de datos devuelta por el backend.

Línea o sección modificada:
Llamadas fetch en las funciones addTask, completeTask y deleteTask.

Problema detectado:
Las solicitudes fetch enviaban parámetros codificados manualmente en la cadena del cuerpo (body), lo que podía generar errores en caso de contener caracteres especiales (por ejemplo, tildes o símbolos). Además, el encabezado 'Content-Type' estaba correctamente definido, pero se recomendó asegurar el uso de encodeURIComponent() para todos los valores dinámicos enviados al servidor.

Solución aplicada:
Se reforzó la codificación de los parámetros mediante encodeURIComponent() para evitar errores en los datos transmitidos.

Código modificado (fragmento):

body: `action=add&description=${encodeURIComponent(description)}`

y en las demás funciones:

body: `action=complete&id=${encodeURIComponent(id)}`

body: `action=delete&id=${encodeURIComponent(id)}`

Sección modificada:
Evento 'submit' del formulario (taskForm).

Problema detectado:
El campo de entrada de texto se limpiaba correctamente tras agregar una tarea, pero la lista de tareas no se actualizaba de inmediato si la respuesta del servidor no devolvía un estado de éxito. Esto podía generar una percepción de falta de respuesta.

Solución aplicada:
Se aseguró que la función loadTasks() se ejecute únicamente después de una respuesta exitosa del servidor dentro del método addTask(), garantizando que la lista se actualice dinámicamente con los nuevos datos confirmados por el backend.

SQL

Archivo editado: todo_app.sql

Sección modificada: Definición de la tabla tareas.

Problema detectado:
El campo completed_at, utilizado en el código PHP dentro del método completeTask, no existe en la estructura de la tabla. Esto ocasiona un error al ejecutar la consulta SQL que intenta actualizar dicha columna (Unknown column 'completed_at' in 'field list').

Solución aplicada:
Se añadió el campo completed_at de tipo DATETIME con valor nulo por defecto para registrar la fecha y hora en que una tarea se marca como completada.

Código agregado:

ALTER TABLE `tareas` ADD COLUMN `completed_at` DATETIME NULL AFTER `completada`;

Sección modificada: Datos de ejemplo (inserciones iniciales).

Problema detectado:
Los registros de prueba no incluían el campo completed_at, lo que podía generar inconsistencias al consultar tareas completadas desde el sistema.

Solución aplicada:
Se actualizaron las inserciones existentes para incluir el campo completed_at con valor NULL, manteniendo la coherencia con el nuevo esquema.

Código modificado:

INSERT INTO `tareas` (`id`, `descripcion`, `completada`, `completed_at`, `fecha_creacion`) VALUES

(1, 'Diseñar la estructura MVC', 0, NULL, '2025-11-04 14:15:05'),

(2, 'Implementar la conexión a la BD', 0, NULL, '2025-11-04 14:15:05'),

(3, 'Añadir funcionalidad de eliminar con JS', 0, NULL, '2025-11-04 14:15:05');

Sección modificada: Definición del campo completada.

Problema detectado:
El código PHP y JavaScript requerían diferenciar entre tareas pendientes y completadas utilizando valores numéricos (0 y 1). Si bien el campo completada ya estaba definido como TINYINT(1), no se especificaba explícitamente el valor por defecto, lo que podría causar problemas si el sistema intentaba insertar una tarea sin indicar su estado.

Solución aplicada:
Se reforzó la definición del campo para asegurar que todas las nuevas tareas se creen con valor 0 (pendiente) por defecto.

Código ajustado:

`completada` TINYINT(1) NOT NULL DEFAULT 0,