

## **Informe de Proyecto: Actividad 2 - Evolución de la API REST**

**Fecha:** 17 de enero de 2026

**Materia:** Desarrollo de Aplicaciones Web

**Entregable:** Actividad 2 - API REST

**Estudiante:** José Daniel Avendaño Morales

**Docente:** Diego Fernando Zárate Pineda

**Programa:** Ingeniería de Sistemas

### **1. Introducción y Objetivo**

El objetivo de esta actividad fue extender las capacidades de la API desarrollada en la fase anterior. Se buscó mejorar la experiencia del usuario mediante la implementación de búsquedas dinámicas, una gestión de estados más flexible y la garantía de integridad de los datos a través de validaciones del lado del servidor.

### **2. Descripción de Funcionalidades Implementadas**

#### **A. Endpoint de Búsqueda de Tareas (GET /api/tareas/buscar)**

Se añadió un endpoint especializado para la localización de recursos específicos.

- **Funcionamiento:** Permite al usuario enviar una palabra clave (término) como parámetro de consulta.
- **Lógica de Negocio:** La API filtra la lista de tareas y devuelve todas aquellas que contengan la palabra clave en su título o en su descripción.
- **Criterio de Calidad:** La búsqueda es insensible a mayúsculas y minúsculas (*case-insensitive*) para mayor precisión.

#### **B. Reversión de Estado a Pendiente (PATCH /api/tareas/{id}/pendiente)**

Se implementó una nueva operación parcial para el manejo de estados.

- **Funcionamiento:** Permite que una tarea previamente marcada como "completada" regrese al estado de "pendiente".
- **Lógica de Negocio:** El campo `IsCompleted` se establece en `false` para el identificador de tarea (ID) proporcionado en la URL.

#### **C. Validaciones de Datos Obligatorias**

Se mejoró la robustez de la API mediante la restricción de entradas de datos.

- **Acción:** Se configuró el sistema para que el título de la tarea sea estrictamente obligatorio.
- **Implementación:** Se utilizaron **Data Annotations** en el modelo de datos para asegurar que no se procesen peticiones de creación (POST) o edición (PUT) si el título está vacío.

### 3. Resolución de Incidencias Técnicas

Durante la fase de compilación con dotnet run, se identificó un error crítico:

- **Problema:** Presencia de etiquetas de texto extrañas (cite\_start, cite\_end) dentro del código fuente de C#.
- **Causa:** Error de sintaxis por copiado de metadatos no pertenecientes al lenguaje.
- **Solución:** Se realizó una limpieza integral de los archivos TaskModel.cs y TareasController.cs, eliminando etiquetas de referencia y restaurando la estructura lógica de los atributos de C#, logrando una compilación exitosa.

### 4. Guía de Validación en Postman

Para verificar el cumplimiento de los criterios de evaluación, se realizaron las siguientes pruebas:

1. **Validación de Integridad:** Intento de POST con título vacío. Resultado: 400 Bad Request.
2. **Búsqueda Funcional:** GET /api/tareas/buscar?termino=clase. Resultado: Retorno exitoso de tareas filtradas.
3. **Gestión de Estado:** PATCH /api/tareas/1/pendiente. Resultado: Cambio verificado del campo isCompleted a false.

### 5. Instrucciones para la Entrega

- **Código Fuente:** Actualizado en el repositorio público de Git con un README.md que detalla los cambios de la Actividad 2.  
  
**Link:** <https://github.com/vallrack/apirestfullparte2>
- **Documentación:** El presente archivo PDF resume todas las modificaciones técnicas.