



Taller. JS - Administrador de tareas

Diego Fernando Cáceres Bonilla - 201811820

Sergio Mauricio Ballen Sedano - 201821143

Universidad Pedagógica y Tecnológica de Colombia (UPTC)

Ingeniería de Sistemas y Computación

Jairo Riaño

Noviembre 26, 2023



Objetivo: Desarrollar habilidades avanzadas en diseño y desarrollo web, enfocándose en la creación de páginas adaptables utilizando frameworks como Bootstrap. Se centrará en la implementación de la lógica de la aplicación mediante JavaScript, la interpretación y aplicación eficiente de JSON como formato de objetos y la manipulación dinámica del DOM.

Además, se implementará un enfoque avanzado utilizando un motor de vistas como EJS para dinamizar y optimizar los contenidos de la página, lo que resultará en una experiencia de usuario más interactiva y eficiente.

1. REQUISITOS FUNCIONALES Y NO FUNCIONALES:

Requisitos Funcionales:

1. Agregar Nueva Tarea:
 - 1.1. El sistema debe permitir a los usuarios agregar nuevas tareas.
 - 1.2. Los usuarios deben proporcionar al menos un título para la tarea.
 - 1.3. Se deben recopilar detalles adicionales, como descripción, fecha de vencimiento y estado de la tarea (pendiente, en progreso, completada).
2. Visualizar Tareas:
 - 2.1. El sistema debe permitir a los usuarios ver la lista de todas las tareas existentes.
 - 2.2. Las tareas deben mostrarse de manera clara con sus detalles, como título, descripción, fecha de vencimiento y estado.
3. Eliminar Tarea:
 - 3.1. El sistema debe permitir a los usuarios eliminar una tarea específica de la lista.
 - 3.2. La eliminación de una tarea debe ser irreversible.

Requisitos No Funcionales:

1. Usabilidad:
 - 1.1. La interfaz de usuario debe ser intuitiva y fácil de usar.
 - 1.2. La aplicación debe ser accesible para usuarios con diferentes niveles de habilidad técnica.
2. Rendimiento:
 - 2.1. La aplicación debe tener un rendimiento eficiente, incluso con grandes cantidades de datos.



2.2. Las operaciones comunes, como agregar, visualizar, editar y eliminar tareas, deben realizarse de manera rápida.

3. Escalabilidad:

3.1. El sistema debe ser escalable para manejar un aumento en la cantidad de usuarios y tareas sin degradación significativa del rendimiento.

4. Compatibilidad:

4.1. El sistema debe ser compatible con diferentes navegadores web y dispositivos para garantizar una experiencia consistente para todos los usuarios.

2. DISEÑO UML

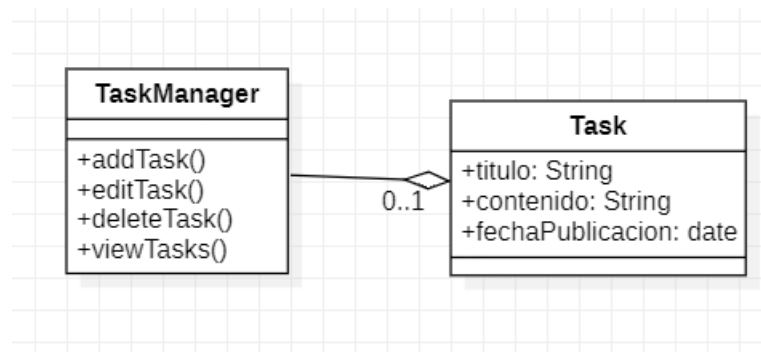


Ilustración 1. Diagrama UML de software. Fuente: Autor

4. MOCKUPS DEL SISTEMA

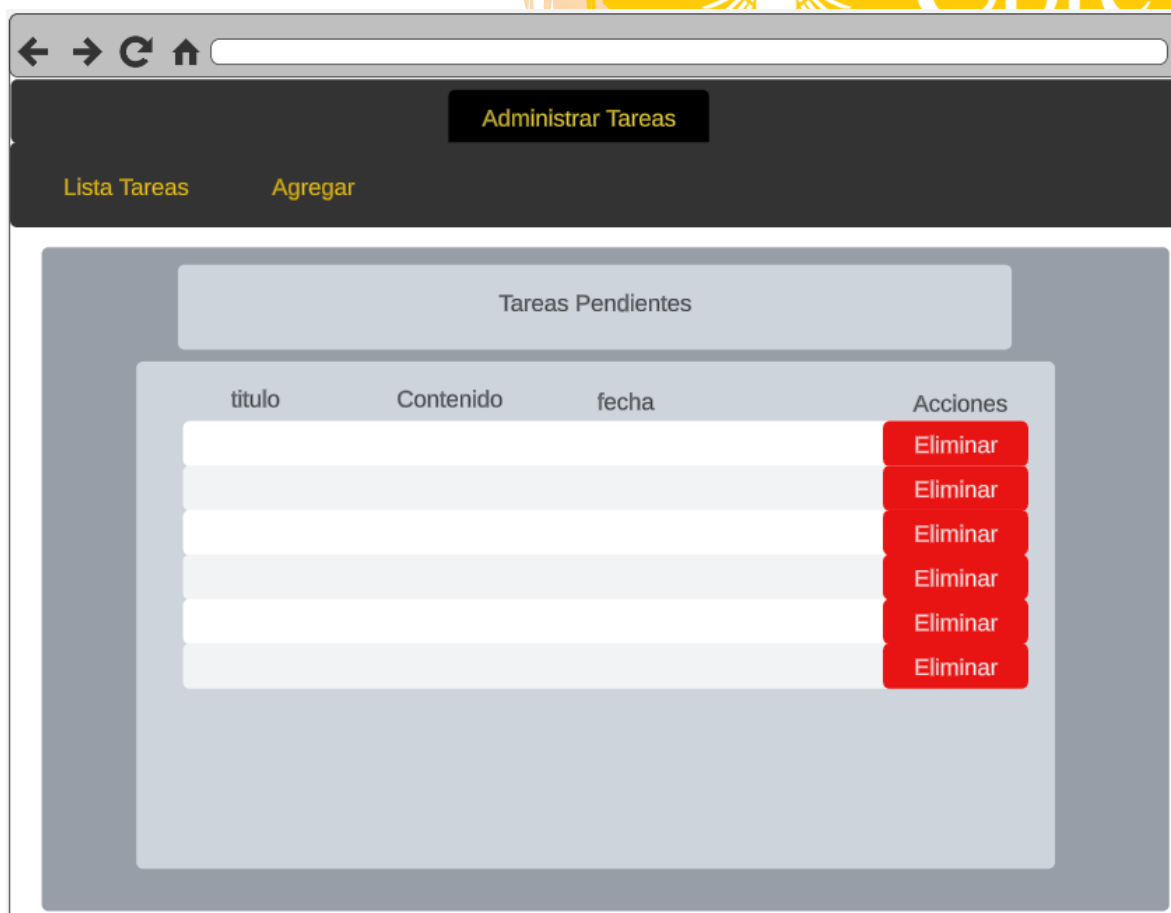


Ilustración 2. Página de Visualización de las tareas. Fuente: Autor



← → ↻ ⬆

Administrar Tareas

Lista Tareas Agregar

Ingrese Nueva Tarea

título

Ingrese Descripcion

Crear

Ilustración 3. Página de registro de Tareas. Fuente: Autor