Documentación del Proyecto: Sistema de Gestión de Tareas

# 1. Introducción

Este proyecto consiste en una aplicación de escritorio para la gestión de tareas personales o de trabajo, desarrollada en Java utilizando NetBeans como entorno de desarrollo. Permite a los usuarios crear, consultar, modificar y eliminar tareas, además de visualizarlas en tablas. Es útil para cualquier persona o equipo que necesite organizar sus pendientes y llevar un control básico de sus actividades.

# 2. Alcance

Lo que hace:

- Permite agregar nuevas tareas indicando nombre, descripción, estado y fecha límite.

- Permite modificar o eliminar tareas existentes.

- Visualiza todas las tareas en tablas.

- Filtra tareas por estado y por fecha límite (opcional, en proceso).

Lo que no hace:

- No permite gestionar múltiples usuarios.

- No ofrece autenticación o control de accesos.

- No genera reportes ni gráficos estadísticos.

- No funciona como aplicación web o móvil.

# 3. Tecnologías usadas

- Java

- NetBeans

- MySQL

- JDBC

- Patrón de diseño MVC (Modelo - Vista - Controlador)

# 4. Requisitos del sistema

Funcionales:

- Crear nuevas tareas.

- Consultar tareas por su ID.

- Modificar tareas existentes.

- Eliminar tareas.

- Mostrar tareas en una tabla.

- Filtrar tareas por estado y/o por fecha límite (opcional).

No funcionales:

- Interfaz sencilla y amigable.

- Respuesta rápida en la gestión de datos.

- Código estructurado bajo MVC para facilitar mantenimiento y escalabilidad.

- Conexión estable con base de datos.

# 5. Análisis

Diagrama de Clases (UML):

@startuml

class Tarea {

- int id

- String nombre

- String descripcion

- String status

- LocalDate limite

}

class conexion {

+ static Connection conn()

}

class tareasDb {

- Connection con

- PreparedStatement ps

+ boolean RegistrarTarea(Tarea t)

+ boolean ModificarTarea(Tarea t)

+ Tarea BuscarTareaPorId(int id)

}

class Principal {

+ void cargarTareas()

+ void btnAgregarActionPerformed()

+ void btnModificarActionPerformed()

+ void btnBuscarActionPerformed()

}

' Relaciones

Principal --> tareasDb : usa

tareasDb --> conexion : usa

tareasDb --> Tarea : manipula

Principal --> Tarea : manipula

@enduml

Diagrama

El contenido generado por IA puede ser incorrecto.

Casos de uso:

- Registrar tarea

- Modificar tarea

- Eliminar tarea

- Buscar tarea por ID

- Mostrar todas las tareas

- Filtrar por estado

- Filtrar por fecha límite (opcional)

Diagrama

El contenido generado por IA puede ser incorrecto.

# 6. Diseño

Wireframes / Mockups (descripción textual):

-Diseño test para las opciones de Mostrar y Eliminar

Diagrama, Rectángulo

El contenido generado por IA puede ser incorrecto.

-Diseño test para la los formularios para agregar y modificar tareas

Imagen que contiene Forma

El contenido generado por IA puede ser incorrecto.

- Ventana principal con:

- Tabla de tareas.

- Campos de texto para capturar nombre, descripción, fecha límite.

- ComboBox para estado de la tarea.

- Botones para agregar, modificar, eliminar, buscar y mostrar todas.

- ComboBox de filtros y botones para filtrar.(no jalo)

# 7. Conclusiones

Aprendizajes:

- Aplicación de Java JDBC para conectarse a MySQL.

- Manejo de fechas con LocalDate y DateTimeFormatter.

- Uso de TableRowSorter para filtros dinámicos en tablas.

- Aplicación del patrón MVC para separar responsabilidades.

Problemas encontrados:

- Manejo de fechas SQL vs LocalDate, resuelto mediante conversiones con toLocalDate().

- Errores al intentar insertar datos nulos en campos DATE en MySQL, solucionado asegurando formatos válidos antes de insertar.

- Filtros en tablas no funcionaban por no tener un TableRowSorter asignado correctamente, se solucionó creando y asociándolo tras cargar los datos.