

PROYECTO 1 ESTRUCTURAS DE DATOS

PROGRAMA GESTOR DE TAREAS

SARA ISABEL SEPÚLVEDA PÉREZ

SAMUEL MEJIA ORTIZ

ANTONIO MORENO



RESUMEN DE CONTENIDOS

1.OBJETIVOS

2.DESCRIPCIÓN

3. ESTRUCTURAS

4.ARQUITECTURA

5. FUNCIONALIDADES

6. PERSISTENCIA

7. CONCLUSIONES

OBJETIVOS

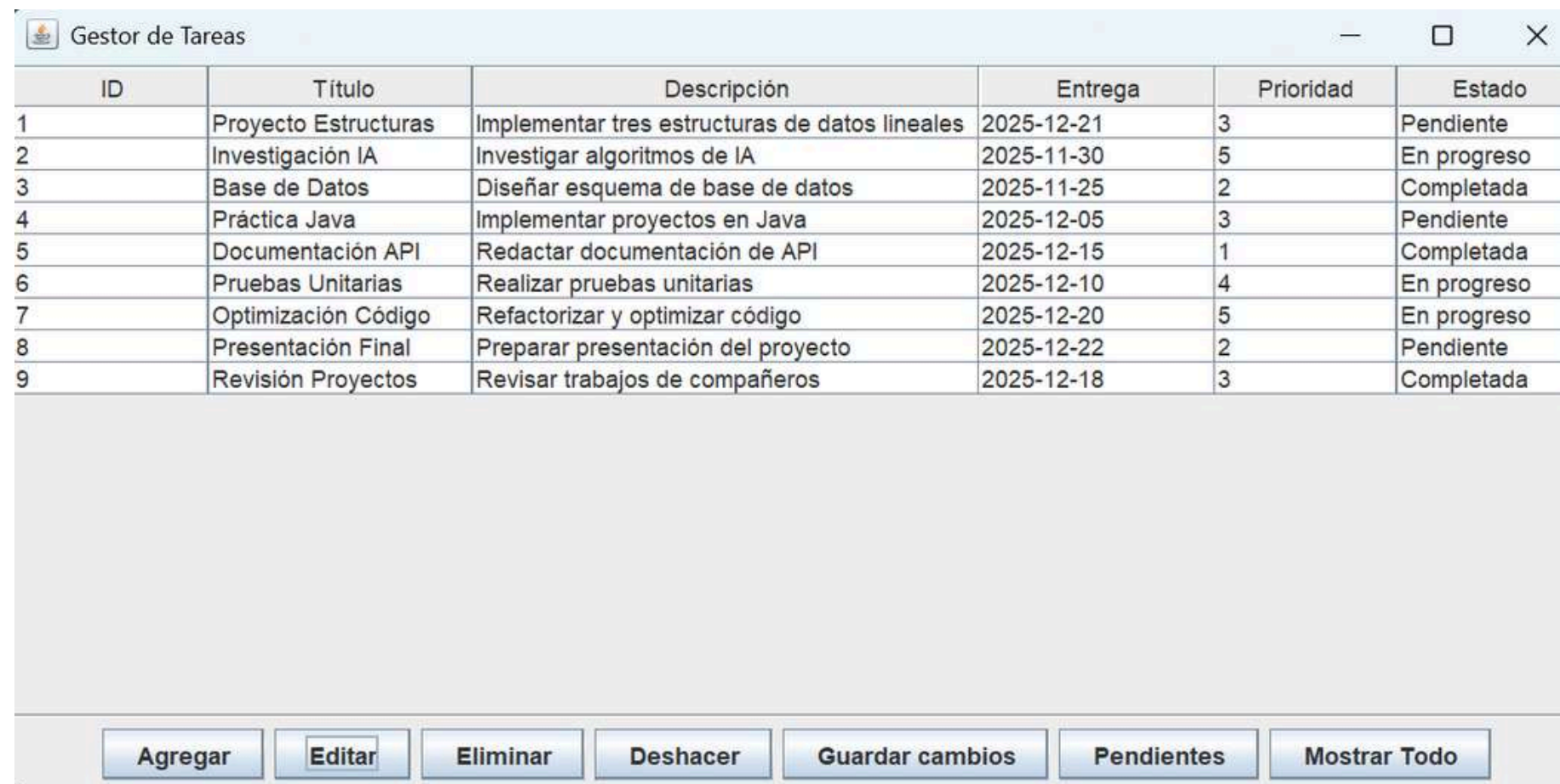
GENERAL

Desarrollar un proyecto de software en Java para gestionar tareas, por medio de una interfaz grafica sencilla y utilizando almenos 3 estructuras de datos lineales que permitan almacenar, organizar y manipular los datos necesarios

ESPECIFICOS

- Implementar un arreglo dinámico, una cola con prioridad y una pila para almacenar las tareas, ordenarlas de acuerdo a su prioridad y poder hacer y deshacer alteraciones a las tareas.
- Diseñar una interfaz grafica intuitiva para facilidad del usuario, que permita agregar las tareas del usuario, almacenarlas en un archivo y posteriormente gestionarlas realizando acciones como eliminar, editar y ordenar,

DESCRIPCIÓN



The screenshot shows a Java Swing window titled "Gestor de Tareas". It contains a table with 6 columns: ID, Título, Descripción, Entrega, Prioridad, and Estado. The table lists 9 tasks. Below the table is a row of buttons: Agregar, Editar, Eliminar, Deshacer, Guardar cambios, Pendientes, and Mostrar Todo.

ID	Título	Descripción	Entrega	Prioridad	Estado
1	Proyecto Estructuras	Implementar tres estructuras de datos lineales	2025-12-21	3	Pendiente
2	Investigación IA	Investigar algoritmos de IA	2025-11-30	5	En progreso
3	Base de Datos	Diseñar esquema de base de datos	2025-11-25	2	Completada
4	Práctica Java	Implementar proyectos en Java	2025-12-05	3	Pendiente
5	Documentación API	Redactar documentación de API	2025-12-15	1	Completada
6	Pruebas Unitarias	Realizar pruebas unitarias	2025-12-10	4	En progreso
7	Optimización Código	Refactorizar y optimizar código	2025-12-20	5	En progreso
8	Presentación Final	Preparar presentación del proyecto	2025-12-22	2	Pendiente
9	Revisión Proyectos	Revisar trabajos de compañeros	2025-12-18	3	Completada

Agregar Editar Eliminar Deshacer Guardar cambios Pendientes Mostrar Todo

El programa en Java utiliza las librerías Java Swing para implementar una interfaz que permite al usuario realizar las acciones indicadas en cada botón para las tareas. Agregar, editar y eliminar una tarea, además deshacer una acción realizada y finalmente filtrar las tareas pendientes

ESTRUCTURAS UTILIZADAS

ARREGLO DINÁMICO

El arreglo dinámico es el que contiene toda las tareas. Permite agregar o eliminar las tareas de manera flexible.

COLA CON PRIORIDAD

La cola con prioridad contiene todas las tareas que se encuentren pendientes, por sus atributos se pueden ordenar las tareas por prioridad y fecha de entrega

PILA

La pila se utiliza para llevar un historial de las modificaciones que realiza el usuario sobre las tareas. Es muy útil para la función de deshacer acción dado que opera bajo el principio LIFO.

ARQUITECTURA DEL SISTEMA



ESTRUCTURA MVC

- **Modelo: Tarea.java & Acción.java**
Representan los objetos principales del sistema, definiendo sus atributos y comportamientos.
- **Controlador: GestorTareas.java & Persistencia.java**
Gestionan la lógica central del programa, manipulando los datos y coordinando las operaciones entre la interfaz y las clases del modelo.
- **Vista: GestorTareasUI.java**
Implementa la interfaz gráfica de usuario (GUI) que permite al usuario interactuar con el sistema.
- **Main.java**
Inicia la ejecución del programa y conecta todos los componentes para su correcto funcionamiento.

FUNCIONALIDADES PRINCIPALES

AGREGAR

Crea una tarea con sus respectivas propiedades (Título, descripción, entrega, prioridad, estado) y la agrega al arreglo dinamico y a la cola con prioridad si cumple la condición.

ELIMINAR

Elimina la tarea seleccionada del arreglo de todas las tareas y la cola de prioridad si es necesario.

EDITAR

Se selecciona cierto atributo de la tarea a editar que quiera ser modificado y guarda la tarea final con los cambios que hace el usuario

GUARDAR Y CARGAR

Las tareas se almacenan en un archivo tareas.txt en el cual se guardan los cambios y este mismo se carga cuando se ejecuta el programa.

VALIDAR PRIORIDAD

Filtrar las tareas con prioridad diferenciándolas de las de mas tareas y mostrándolas por orden de prioridad y fecha de entrega.

DESHACER

La ultima acción realizada por el usuario puede revertirse si lo desea para volver al estado anterior a la modificación.

PERSISTENCIA

Las tareas que se encuentren en el arreglo se almacenan en el archivo `tareas.txt` para garantizar que no se pierdan al terminar la ejecución. Actúa como "Base de datos", utiliza un formato que separa los atributos por un `;` como se muestra:

```
1;Proyecto Estructuras;Utilizar estructuras de datos;2025-10-19;2025-12-21;3;Pendiente
```

Al ejecutar de nuevo el programa se cargan los datos almacenados en el archivo `tareas.txt`

CONCLUSIONES

- Se aplicaron estructuras de datos lineales (listas, pilas y colas con prioridad) en un caso práctico real.
- La separación entre lógica, interfaz y persistencia facilitó la organización del código.
- El uso de Java y Swing permitió una interfaz intuitiva y funcional.

The background features a white canvas with decorative elements. In the top-left corner, there is a light blue watercolor splash. The top-right corner is adorned with a solid blue abstract shape. The bottom-left corner contains a solid blue abstract shape, and the bottom-right corner features a light blue watercolor splash. Centered on the canvas is the text '¡MUCHAS GRACIAS!' in a bold, dark blue, sans-serif font.

**¡MUCHAS
GRACIAS!**