

ALGORITMOS Y ESTRUCTURAS DE DATOS

(100000SI42)

Avance del proyecto

Sistema de Gestión de Tareas Pendientes

Docente :

Luis Alberto Loo Parian

Estudiante :

Daniel Esteban Huamán Lázaro

Año :

2025

1. INTRODUCCIÓN
2. CAPÍTULO 1 – ASPECTOS GENERALES
 - 1.1 Descripción de la Organización
 - 1.2 Misión y Visión
 - 1.3 Arquitectura Tecnológica Actual
3. CAPÍTULO 2 – MARCO TEÓRICO
 - 2.1 Tecnologías para Desarrollo de Software
 - 2.2 Arquitecturas de Software
 - 2.3 Metodologías de Desarrollo de Software
 - 2.4 Aplicaciones de Algoritmos con Estructuras de Datos
4. CAPÍTULO 3 – GESTIÓN DEL PROYECTO
 - 3.1 Objetivos
 - 3.2 Descripción del Problema
 - 3.3 Alcance del Proyecto
5. CAPÍTULO 4 – IMPLEMENTACIÓN DE ESTRATEGIA
 - 4.1 Lista de Requerimientos
 - 4.2 Prototipo Preliminar
 - 4.3 Prototipo Funcional
 - 4.4 Desarrollo de la Aplicación
6. CAPÍTULO 5 – CIERRE Y DOCUMENTACIÓN
 - 5.1 Versión Final del Producto
 - 5.2 Manuales de Usuario
7. CONCLUSIONES
8. RECOMENDACIONES
9. BIBLIOGRAFÍA
10. ANEXOS

1. INTRODUCCIÓN

El presente documento describe el avance del proyecto titulado "**Sistema de Gestión de Tareas Pendientes**". Este proyecto tiene como objetivo implementar un programa que permita al usuario gestionar tareas mediante estructuras de datos dinámicas. Este avance forma parte del curso **Algoritmos y Estructuras de Datos**, y busca aplicar conceptos teóricos de estructuras de datos en un caso práctico.

CAPÍTULO 1 – ASPECTOS GENERALES

1.1 Descripción de la Organización

Este proyecto se desarrolla en el contexto del curso universitario "Algoritmos y Estructuras de Datos", orientado a la aplicación de conceptos teóricos en soluciones prácticas mediante la programación en Java.

1.2 Misión y Visión

Misión: Desarrollar soluciones de software eficientes que apliquen conceptos de estructuras de datos para la gestión de información dinámica.

Visión: Convertirse en una referencia académica en la implementación de estructuras de datos aplicadas a problemas del mundo real.

1.3 Arquitectura Tecnológica Actual

- **Hardware:** Computadora personal con especificaciones estándar para desarrollo de software.
- **Software:**
 - Lenguaje de Programación: Java
 - Entorno de Desarrollo: Apache NetBeans
 - Sistema Operativo: Zorin OS (Linux)

CAPÍTULO 2 – MARCO TEÓRICO

2.1 Tecnologías para Desarrollo de Software

El sistema está desarrollado en Java, un lenguaje orientado a objetos que permite gestionar estructuras de datos de manera eficiente. Se utiliza Apache NetBeans como entorno de desarrollo integrado (IDE) debido a su compatibilidad con proyectos en Java y sus herramientas de depuración. Zorin OS, un sistema operativo basado en Linux, proporciona un entorno estable para el desarrollo.

2.2 Arquitecturas de Software

Se emplea una arquitectura basada en el patrón MVC (Modelo-Vista-Controlador), que separa la lógica de negocio, la interfaz de usuario y el acceso a los datos.

2.3 Metodologías de Desarrollo de Software

Se utiliza una metodología ágil con iteraciones cortas para validar el avance del proyecto y ajustar funcionalidades según el feedback recibido.

2.4 Aplicaciones de Algoritmos con Estructuras de Datos

El sistema implementa listas enlazadas para gestionar tareas, permitiendo operaciones eficientes de inserción, eliminación, y recorrido.

CAPÍTULO 3 – GESTIÓN DEL PROYECTO

3.1 Objetivos

Objetivo General

Desarrollar un sistema en Java que permita gestionar tareas pendientes utilizando estructuras de datos dinámicas (colas con prioridad o listas enlazadas).

Objetivos Específicos

- Implementar una estructura de datos dinámica adecuada para la gestión de tareas.
- Permitir al usuario agregar, listar y completar tareas.
- Ordenar las tareas según su prioridad (Alta, Media, Baja).
- Validar la funcionalidad del sistema mediante pruebas simples.

3.2 Descripción del Problema

El proyecto surge de la necesidad de organizar tareas en un entorno dinámico donde las prioridades cambian constantemente. La falta de un sistema de gestión eficiente puede generar desorganización y pérdida de control sobre las tareas pendientes.

3.3 Alcance del Proyecto

El sistema permitirá:

- Registrar tareas con descripciones y prioridades.
- Visualizar la lista de tareas pendientes.
- Completar (eliminar) tareas según su prioridad.
- Ordenar y buscar tareas de forma eficiente.

CAPÍTULO 4 – IMPLEMENTACIÓN DE ESTRATEGIA

4.1 Lista de Requerimientos

- **Requisitos Funcionales:**
 - Agregar tareas.
 - Listar tareas ordenadas por prioridad.
 - Completar y eliminar tareas.
 - Buscar tareas específicas.
- **Requisitos No Funcionales:**
 - Interfaz de usuario simple y clara.
 - Respuesta rápida para operaciones básicas.

4.2 Prototipo Preliminar

Se desarrolló un prototipo básico con funcionalidades de inserción y listado de tareas en la consola.

4.3 Prototipo Funcional

Se ha avanzado en la implementación de tareas de actualización, eliminación y ordenamiento por prioridad. El sistema ahora cuenta con un menú interactivo mejorado, con mensajes de confirmación, sugerencias automáticas para tareas similares y una interfaz de usuario más amigable. Se incluyen capturas de pantalla del sistema en funcionamiento.

4.4 Desarrollo de la Aplicación

La aplicación final integra todos los módulos siguiendo la arquitectura MVC, con una estructura de paquetes para model, controller, view y clases. Se asegura el correcto funcionamiento mediante pruebas de validación y manejo de errores. Se considera la posibilidad de implementar una interfaz gráfica en futuras versiones.

CAPÍTULO 5 – CIERRE Y DOCUMENTACIÓN

5.1 Versión Final del Producto

El sistema está completamente operativo, permitiendo la gestión eficiente de tareas. Se destaca la implementación de una búsqueda flexible, gestión de prioridades y una experiencia de usuario mejorada.

5.2 Manuales de Usuario

Se incluye un manual básico para el usuario final con instrucciones sobre cómo usar el sistema, incluyendo cómo agregar, listar, buscar y completar tareas.

CONCLUSIONES

El proyecto ha permitido aplicar conceptos teóricos de estructuras de datos en un entorno práctico, mejorando la comprensión de su funcionamiento y utilidad en la resolución de problemas reales. La implementación de listas enlazadas y la estructura MVC han facilitado la organización del código y la escalabilidad del sistema.

RECOMENDACIONES

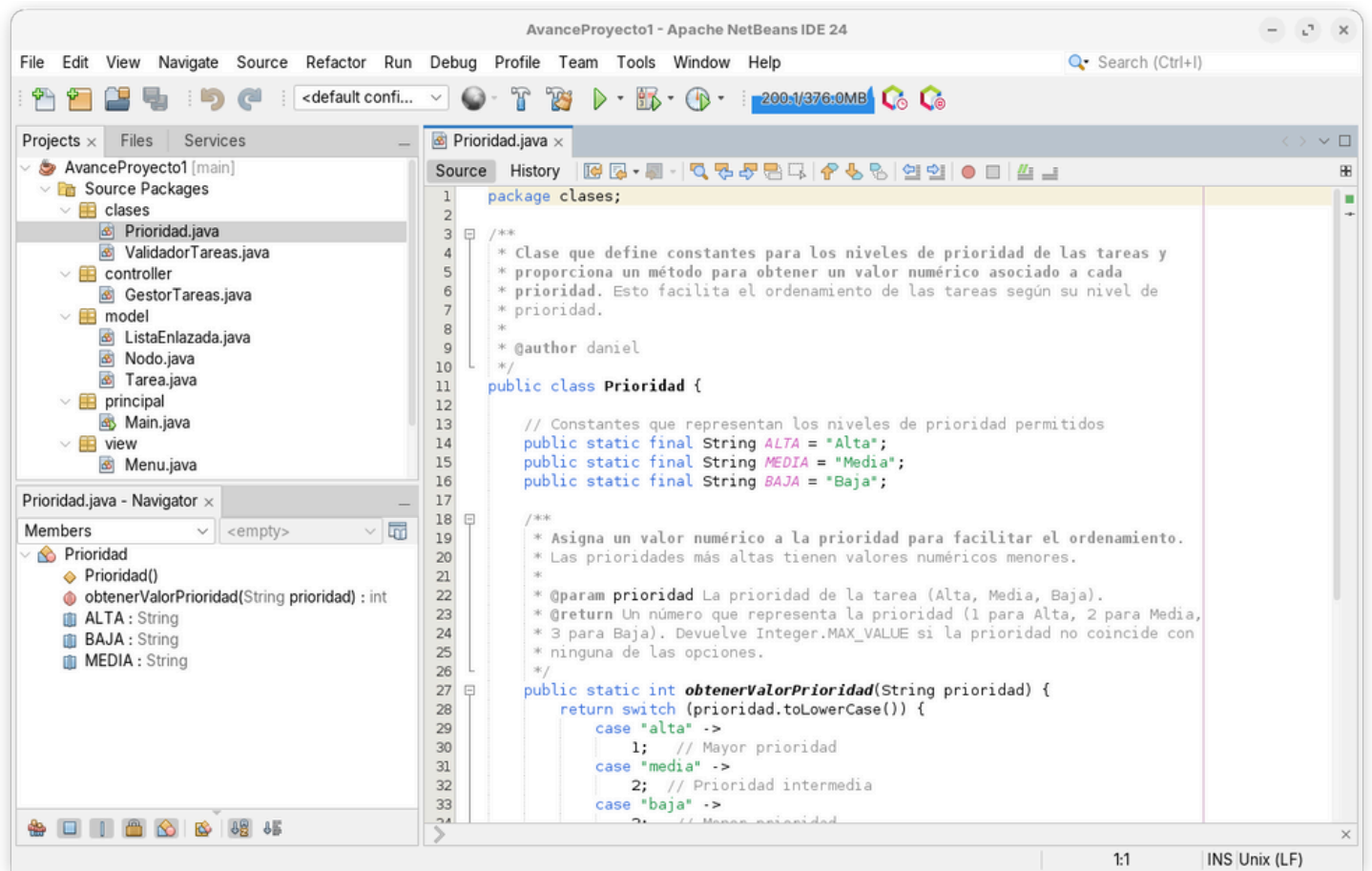
- Integrar funcionalidades adicionales como fechas límite para tareas.
- Mejorar la interfaz gráfica para mayor usabilidad en futuras versiones.
- Optimizar el sistema para manejar grandes volúmenes de tareas.

BIBLIOGRAFÍA

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*.
- Oracle Java Documentation.

ANEXOS

- Capturas de pantalla del sistema en funcionamiento en Apache NetBeans sobre Zorin OS.



```
daniel@daniel-desktop: ~$ neofetch

`osssssssssssssssssso`
.OSSSSSSSSSSSSSSSSSSSO.
.+000000000000000000000000+.

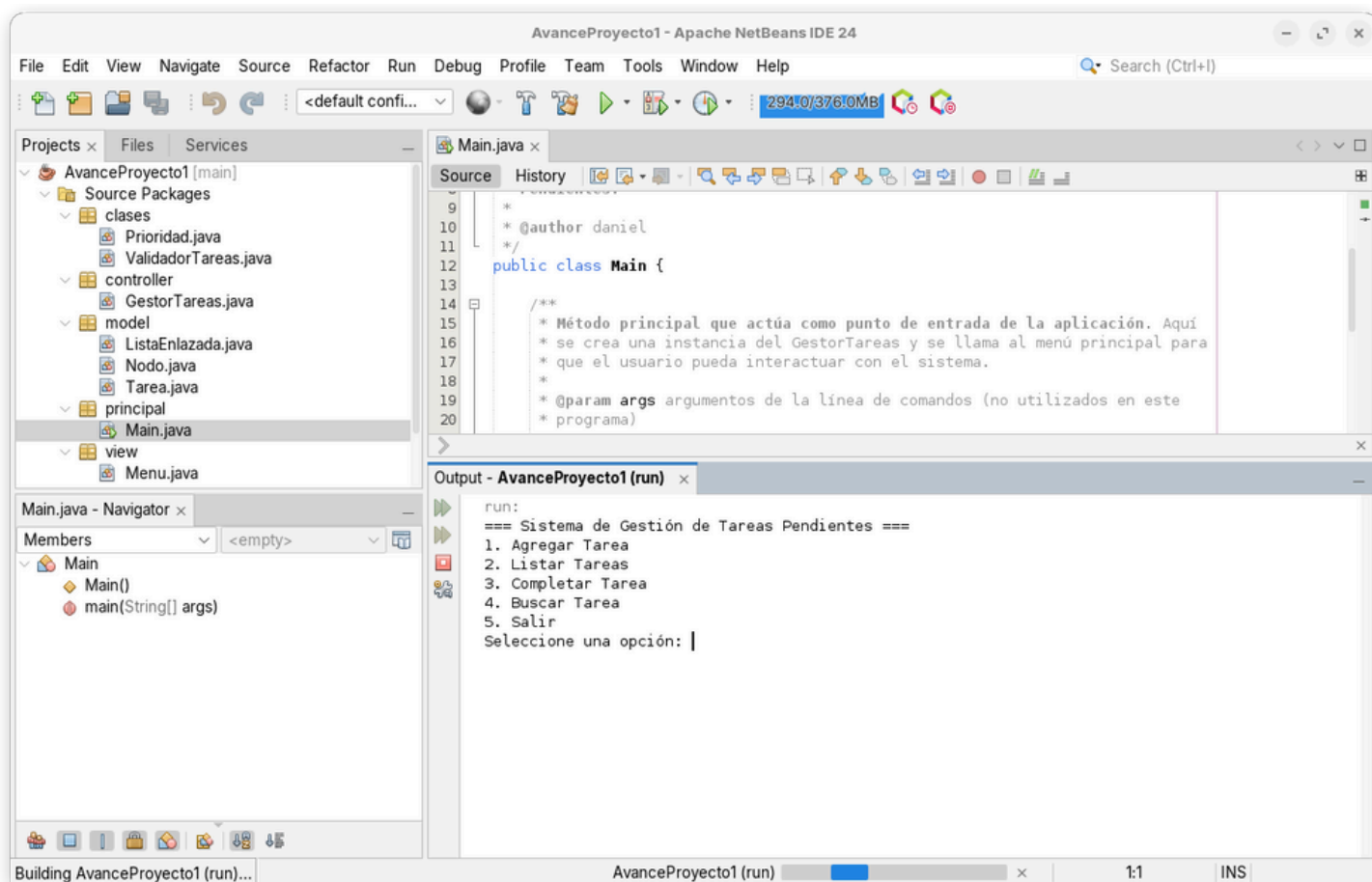
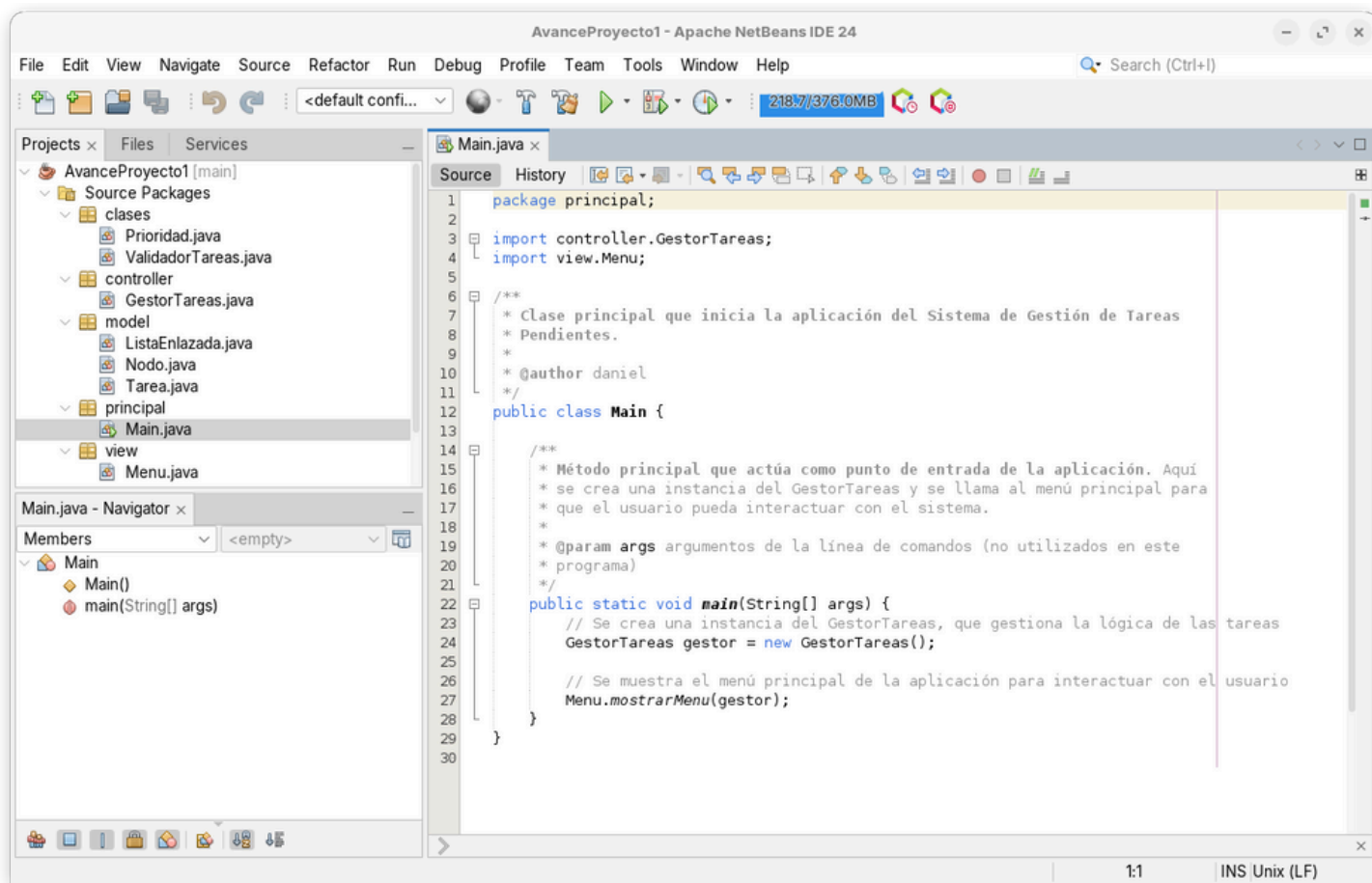
`.:.:.
`+ssssssssssssssss+.:. `.:+ssso`
.OSSSSSSSSSSSSSSSO/. `.-+OSSSSSSSO.
SSSSSSSSSSSSSO/- `.-/OSSSSSSSSSSSS
.OSSSSSSSO/- `.-/OSSSSSSSSSSSSSO.
`+sss+.:. `.:+SSSSSSSSSSSSSSSS+`
`.:. :.:.

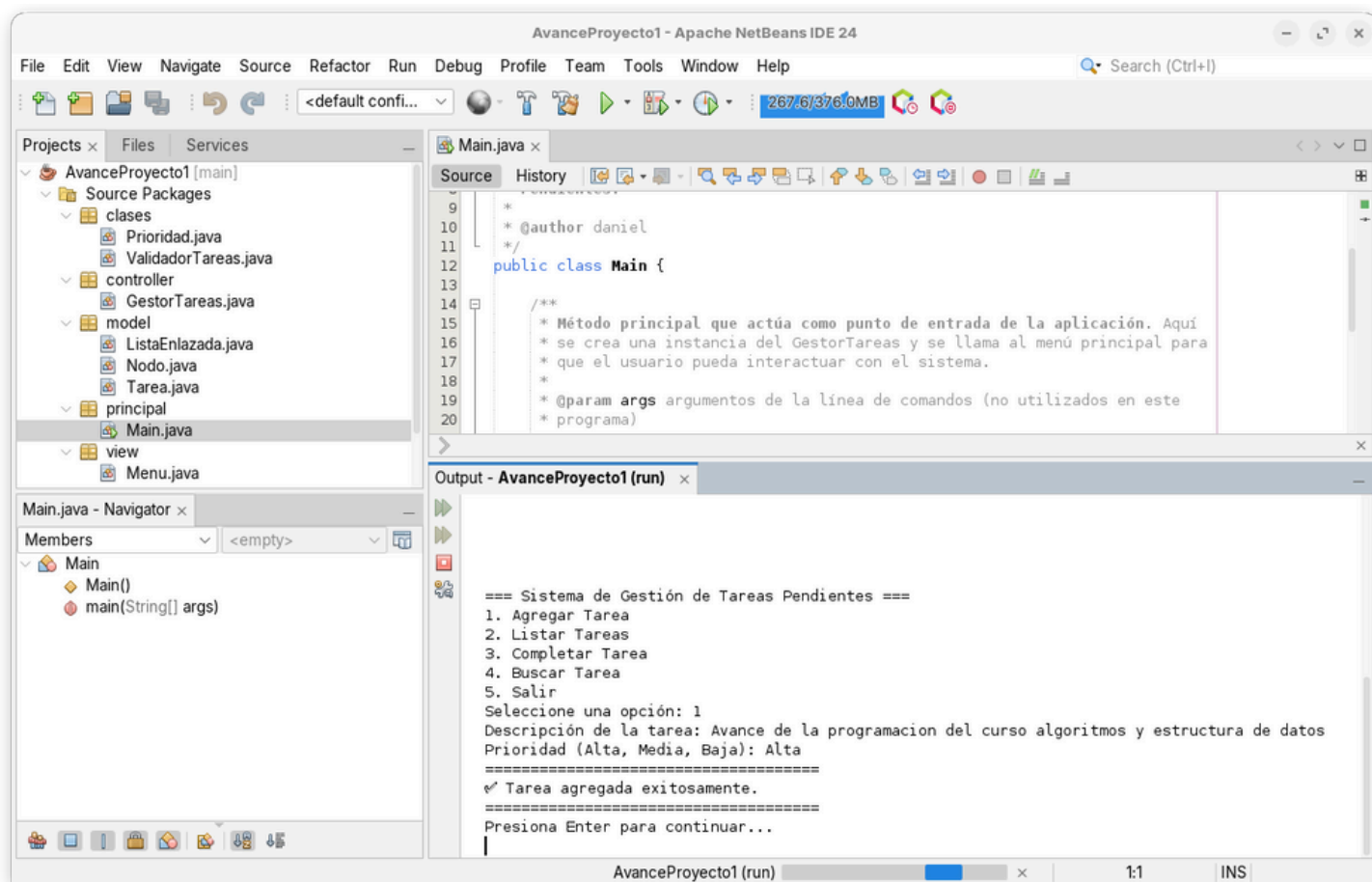
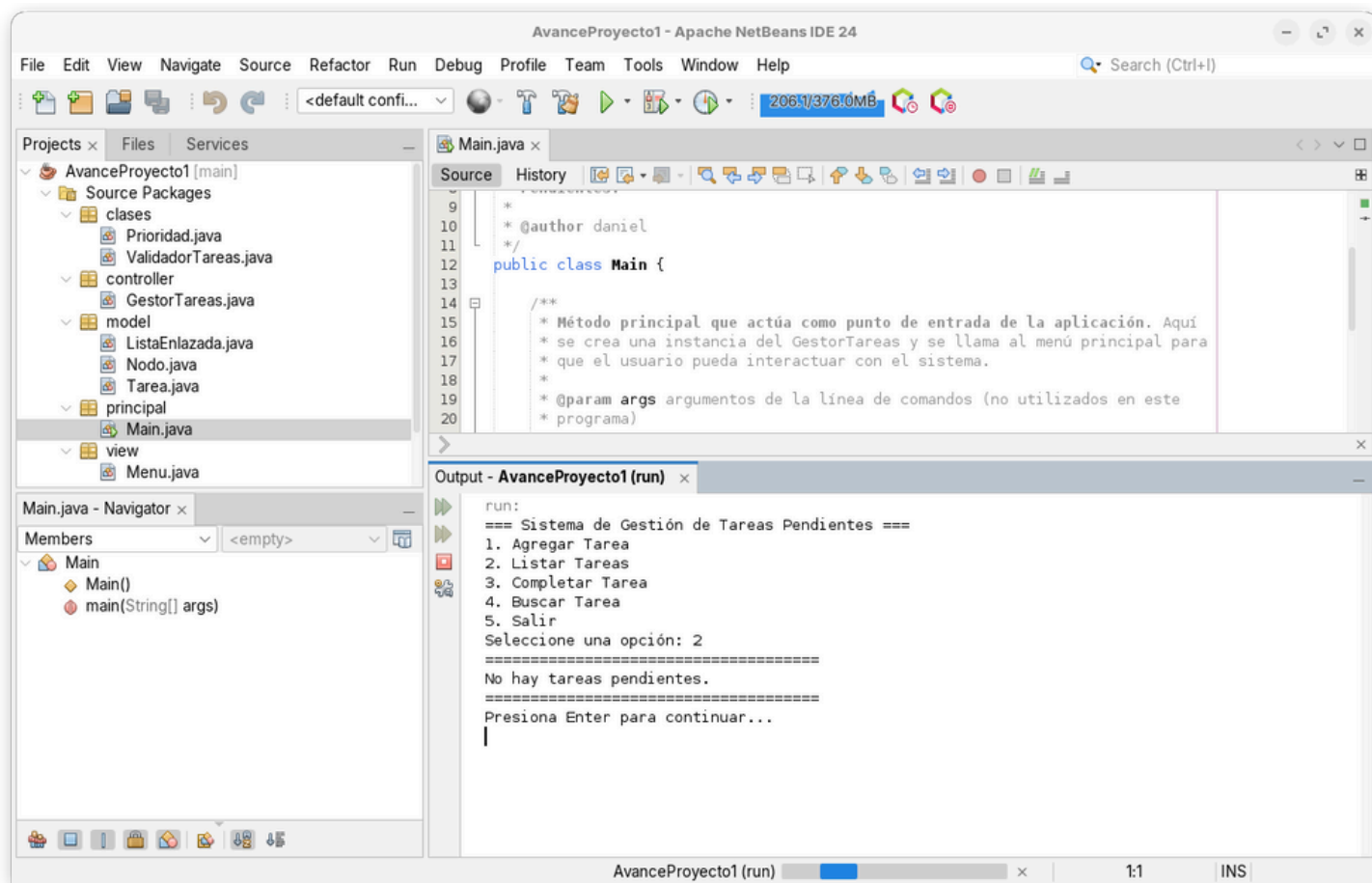
.+000000000000000000000000+.
-OSSSSSSSSSSSSSSSSSSSSSO-
`OSSSSSSSSSSSSSSSSSSSO`

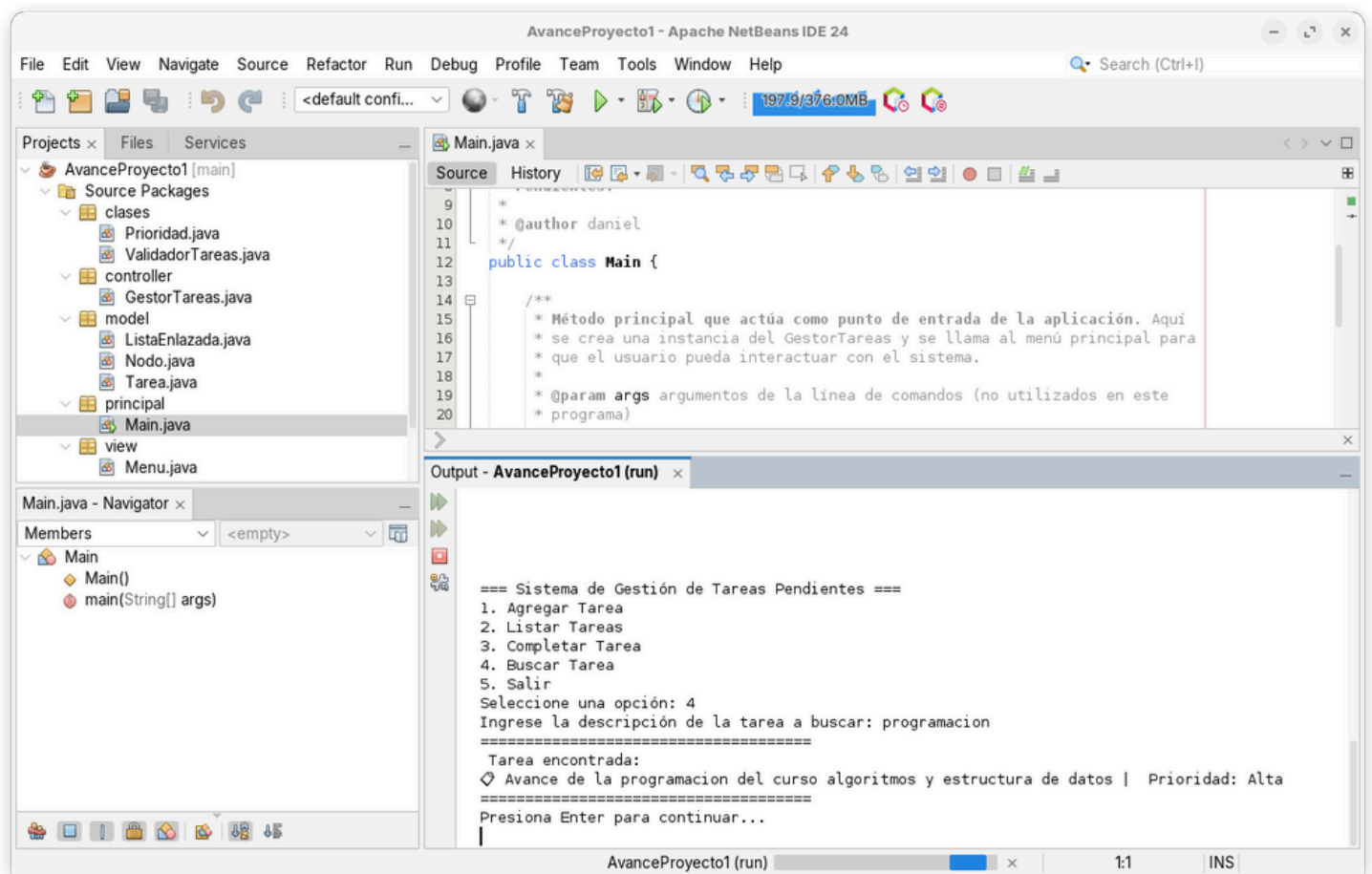
daniel@daniel-desktop
-----
OS: Zorin OS 17.2 x86_64
Kernel: 6.8.0-52-generic
Uptime: 14 hours, 53 mins
Packages: 2241 (dpkg), 113 (flatpak)
Shell: bash 5.1.16
Resolution: 1920x1080
DE: GNOME 43.9
WM: Mutter
WM Theme: ZorinBlue-Light
Theme: ZorinBlue-Light [GTK2/3]
Icons: ZorinBlue-Light [GTK2/3]
Terminal: gnome-terminal
CPU: Intel i5-10400 (12) @ 4.300GHz
GPU: NVIDIA GeForce GTX 1650
Memory: 11244MiB / 32023MiB

daniel@daniel-desktop:~$
```

- Fragmentos de código relevantes que demuestran la implementación de la lógica principal del sistema.







Link del github :

<https://github.com/DanielLazaro1555/Algoritmos-y-Estructuras-De-Datos-7056>