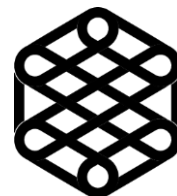




Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno



Licenciatura en Ingeniería de Software
UACM San Lorenzo Tezonco
Diseño de software

Proyecto del curso

NOMBRE DEL PRODUCTO:

Gestor de Operaciones Multitareas (GOMT).
Especificación de Requerimientos de Software

Versión: 0.1.8

18 de mayo de 2025

Grupo 301 2025-I
Equipo: Operaciones Plus
INTEGRANTE, MATRÍCULA

- **Brayan Campos**– Matrícula: 20-003-0236
- **Raymundo Ramírez** – Matrícula: 20-003-0335

Tabla de contenidos

Contenido

1 Introducción	5
1.1 Alcances	5
2 Características del producto	5
2.1 Funciones	5
2.1.1 Requisitos Funcionales	5
2.2 Entornos, Entradas y Salidas	6
2.2.1 Requisitos de Interfaz	6
2.3 Características y Restricciones Adicionales	6
2.3.1 Requisitos de Calidad en Uso	6
2.3.2 Requisitos Técnicos y de Pruebas	6
2.4 Usuarios	6
2.4.1 Requisitos de Usabilidad	6
3. Casos de Uso	6
CU01 – Modificar Tareas	6
CU02 – Modificar Operaciones	6
CU03 – Conocer Estado de Operaciones	6
CU04 – Modificar Estado de Operaciones	7
CU05 – Acerca de...	7
4. Diagramas UML	7
5. Historial de Cambios	10
6. Reactivos	10
6.2. Especificación de Requerimientos de Software	10
6.3. Propuesta de solución	11
6.4. Seguimiento del trabajo	11
6.5. Reporte de actividades	11
Roles	12
Explicación detallada de roles:	12
7. Extensión de la Práctica 2	13
7.1 Actualización de documentación	13
7.2 Propuesta de uso e interacción	13
7.3 Implementación de objetos principales (Reactivo 4.4)	13
7.4 Reporte de actividades	13
8.1 Mockups	16
16	
9. Extension de la Practica 3	
9.1.1 Descripción general	18
9.1.3 Organización del repositorio de trabajo	18
10.1 Descripción general	19

10.1.5 Especificación de Requerimientos de Software (ERS)	21
11.1.1 Descripción general	22
11.1.2 Características clave implementadas	22
11.1.3 Componentes funcionales del sistema	22
Interfaz gráfica (JavaFX 21)	22
11.1.4 Lógica funcional (Java 21)	22
11.1.5 Calidad de implementación	22

1 Introducción

La presente Especificación de Requerimientos de Software (ERS) describe un sistema gestor de operaciones multitarea con interfaz gráfica de usuario denominado **Gestor de Operaciones Multitarea (GOMT)**, identificado como `mx.edu.uacm.is.slt.ds.gomt`.

Este documento está basado en los lineamientos de la norma ISO/IEC/IEEE 29148:2018.

1.1 Alcances

Esta ERS se limita a describir una prueba de concepto del sistema, por lo que describe un sistema de baja escala. Sin embargo el diseño modular debe permitirle ser refinado y escalado a bajo costo.

La descripción de los requisitos se limita a presentar su estado y descripción para su seguimiento y desarrollo. La descripción de los requisitos incluye las características específicas que debe poseer el producto y cómo probarlas.

2 Características del producto

2.1 Funciones

2.1.1 Requisitos Funcionales

ID	Requisito	Responsable
RF01	Crear nueva tarea	Brayan Campos
RF02	Modificar tarea existente	Raymundo Ramírez
RF03	Eliminar tarea existente	Brayan Campos
RF04	Crear operación	Raymundo Ramírez
RF05	Modificar operación existente	Brayan Campos
RF06	Consultar estado de operaciones	Raymundo Ramírez
RF07	Modificar estado de operaciones	Brayan Campos
RF08	Mostrar sección "Acerca de..."	Raymundo Ramírez

2.2 Entornos, Entradas y Salidas

2.2.1 Requisitos de Interfaz

- El sistema debe estar desarrollado en **Java 21** o superior.
- La interfaz gráfica será realizada en **JavaFX 21**.
- Control de versiones mediante **GitHub**.
- Comunicaciones de equipo a través de **Discord**.

2.3 Características y Restricciones Adicionales

2.3.1 Requisitos de Calidad en Uso

- El sistema debe ser multiplataforma (Windows, Linux, macOS).
- Uso intuitivo para usuarios no técnicos.
- Documentación disponible desde la aplicación.

2.3.2 Requisitos Técnicos y de Pruebas

- Compatible con Java SDK 21.
- Pruebas básicas de integridad de operaciones.

2.4 Usuarios

2.4.1 Requisitos de Usabilidad

El sistema está dirigido a gerentes y usuarios administrativos que requieren simular o ejecutar operaciones automatizadas de forma intuitiva.

3. Casos de Uso

CU01 – Modificar Tareas

Actor: Usuario

Descripción: Permite editar tareas (nombre, precondiciones, comportamiento).

CU02 – Modificar Operaciones

Actor: Usuario

Descripción: Permite cambiar la estructura de una operación: añadir, eliminar o reordenar tareas.

CU03 – Conocer Estado de Operaciones

Actor: Usuario

Descripción: Consulta el estado actual de cualquier operación registrada en el sistema.

CU04 – Modificar Estado de Operaciones

Actor: Usuario

Descripción: Cambia manualmente el estado de una operación (por ejemplo, de "pausada" a "ejecutando").

CU05 – Acerca de...

Actor: Usuario

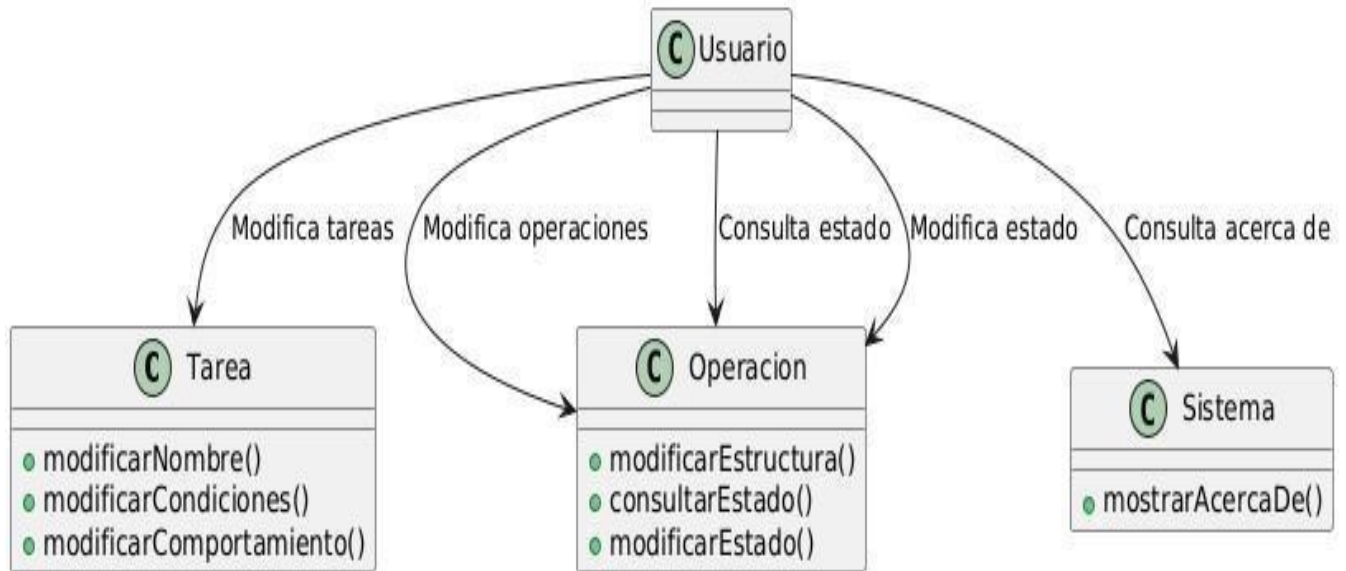
Descripción: Muestra información de la versión del software, autores y derechos.

4. Diagramas UML

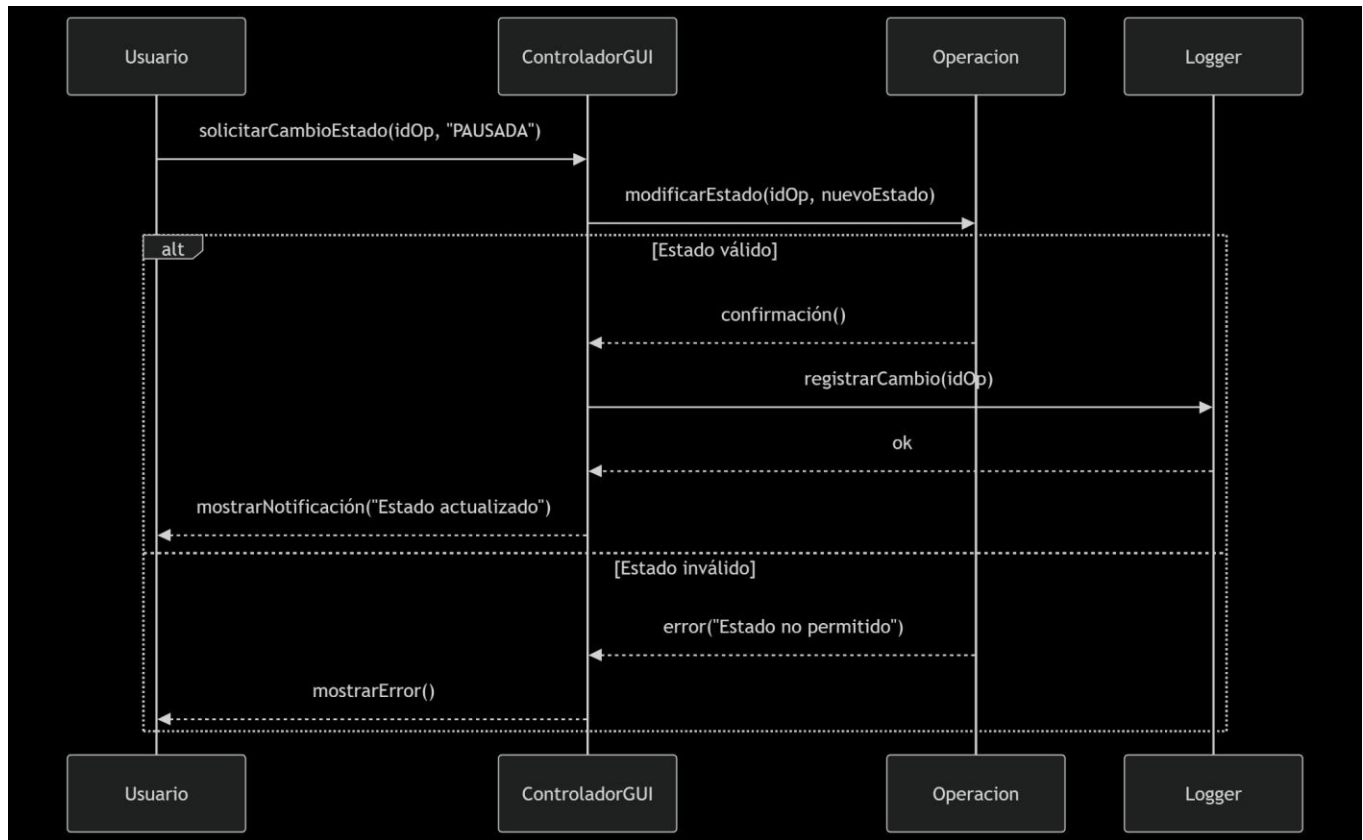
4.1 Diagrama de Casos de Uso



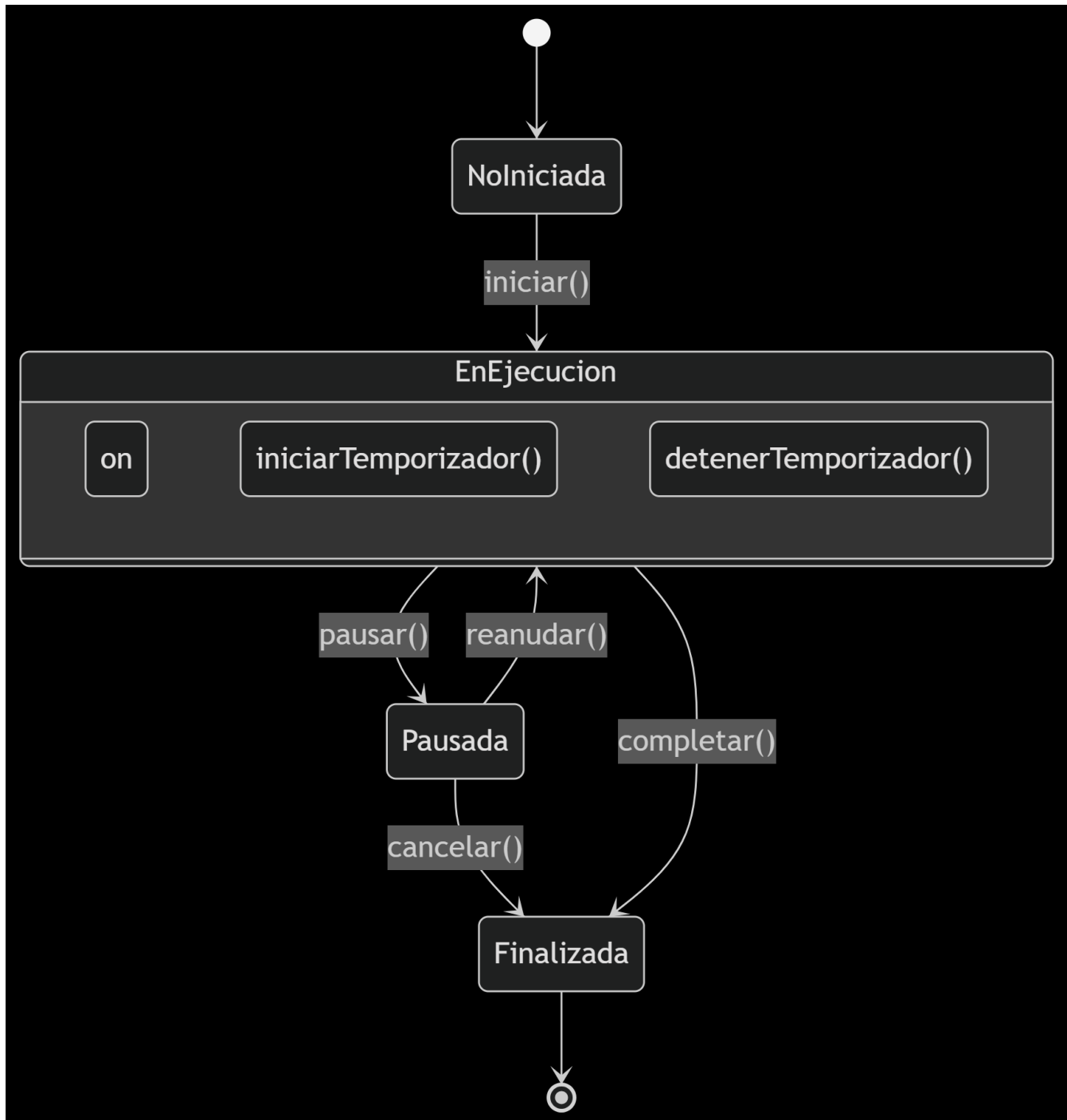
4.2 Diagrama de Clases



4.3 Diagrama de Secuencia



4.4 Diagrama de Estados



5. Historial de Cambios

Versión	Fecha	Cambio
0.1.0	21 de abril de 2025	Documento inicial, integración de casos de uso y diagramas UML.
0.1.1	28 de abril de 2025	Repositorio GitHub
0.1.3	06 de mayo 2025	Interfaces del sistema planteamientos
0.1.6	10 de mayo 2025	Interfaces Mockups
0.1.8	18 de mayo 2025	Actualización de documento

6. Reactivos

6.1. Designación de roles y organización del equipo

- **Nombre del equipo:** Operaciones Plus
- **Integrantes:**
 - Brayan Campos (20-003-0236) ○
 - Raymundo Ramírez (20-003-0335) •
 - Líderes del equipo:** ○ Brayan Campos ○ Raymundo Ramírez •
 - Umbral de votación:** Mayoría simple •
 - Entorno de trabajo definido:** ○
 - Java 21, JavaFX 21. ○ IDE: NetBeans.
- Control de versiones: GitHub. ○
- Comunicación: Discord

6.2. Especificación de Requerimientos de Software

- Nombre del producto: Gestor de Operaciones Multitarea (GOMT).
- Versión: 0.1.0.
- Identificador: mx.edu.uacm.is.slt.ds.gomt.
- Integrantes y matrículas documentados.
- Responsabilidad de requisitos distribuida entre Brayan y Raymundo.
- Se agregaron los casos de uso adicionales requeridos:
 - Modificar tareas. ○ Modificar operaciones. ○ Conocer estado de operaciones. ○ Modificar estado de operaciones.
 - Información "Acerca de..."

- Se actualizó el historial de cambios.
- Se actualizó la tabla de contenidos.

6.3. Propuesta de solución

Se elaboró la propuesta de solución:

1. **Casos de uso documentados:**
(5 casos de uso definidos y ligados a requisitos funcionales).
2. **Diagramas de clases:**
(Tarea, Operación, Sistema y sus relaciones con el Usuario).
3. **Diagramas de secuencia:**
(Ejemplo detallado para el caso "Modificar tarea").
4. **Diagramas de estados:**
(Estados de una operación: NoIniciada, EnEjecucion, Pausada, Finalizada).

6.4. Seguimiento del trabajo

Se documentaron las actividades del equipo para el avance de las Alfas:

- **Requisitos Aceptable:** Integración de los requisitos nuevos.
- **Producto de software Arquitectura elegida (parcial):** Diagramas de clases básicos realizados.
- **Trabajo Bajo control:** GitHub usado para seguimiento de versiones.
- **Equip: Colaborando:** Organización y trabajo en conjunto (Discord).
- **Forma de trabajo Colocada:** Entorno y flujo de trabajo establecidos.

Se ligaron las actividades de los integrantes directamente con el avance de los estados de las Alfas.

6.5. Reporte de actividades

El reporte de actividades generado incluye:

- Portada oficial (equipo, integrantes, producto, versión, identificador).
- Definición de roles y organización del equipo.
- Objetivos de la etapa actual basados en el diseño inicial de solución.
- Esquema de versionamiento basado en Semantic Versioning.
- Inclusión completa de la ERS (este documento).
- Documentación completa de la propuesta de solución (casos de uso, diagramas).
- Registro de seguimiento de las Alfas de Essence.
- Comentarios generales:
 - El equipo logró organizarse y realizar las actividades propuestas.
- Comentarios individuales:
 - Brayan: "Participé activamente en los diagramas UML y los requisitos".
 - Raymundo: "Me encargué del modelado de clases y verificación de requisitos".

Roles

Integrante	Matrícula	Rol General	Rol Técnico	Rol en Documentación
Brayan Campos	20-003-0236	Líder del equipo	Responsable de ERS, Diagramas UML	Redacción y organización de la documentación
Raymundo Ramirez	20-003.0335	Líder del equipo	Responsable de Propuesta de Solución, Seguimiento de Alfas	Coordinador de cambios y control de versiones

Explicación detallada de roles:

- **Líder del equipo:**
Ambos integrantes fungen como líderes formales para la coordinación del proyecto.
- **Responsable de ERS:**
 - Brayan gestiona la actualización, ajustes de requisitos y supervisión de versiones de la ERS.
- **Responsable de Propuesta de Solución:**
 - Raymundo desarrolla los aspectos de alto nivel del diseño, como casos de uso, arquitectura general y diagramas.
- **Seguimiento de Alfas:**
 - Raymundo lleva el registro de progreso en las listas de estados de Essence (Trabajo, Requisitos, Producto de Software).
- **Control de versiones:**
 - Raymundo se encarga de administrar el repositorio en GitHub, validando los commits y merges.
- **Responsable de diagramas UML:**
 - Brayan genera los diagramas de Casos de Uso, Clases, Secuencias y Estados.
- **Responsable de Documentación:**
 - Brayan organiza el reporte final, incluyendo portadas, tablas de contenidos automáticas y normas de formato académico.

7. Extensión de la Práctica 2

7.1 Actualización de documentación

Se actualizaron los siguientes elementos conforme al diseño iterativo:

- Casos de uso: revisados y extendidos.
- Diagramas de secuencia: se agregaron secuencias para los nuevos mecanismos de edición.
- Diagramas de clases: se refinó la relación entre Operación y Tarea.
- Diagramas de estados: se añadieron estados intermedios y excepcionales.
- ERS: se actualizó con nuevos requisitos y responsables.

7.2 Propuesta de uso e interacción

Se diseñaron los siguientes mock-ups para las interfaces de usuario (IU):

- Edición de Operaciones: formulario con árbol de tareas.
- Edición de Tareas: editor de nombre, tipo y comportamiento.
- Panel de control: vista resumen con botones de acción para iniciar/pausar.
- Acerca de: pantalla emergente con nombre del producto, versión, autores y contacto.

7.3 Implementación de objetos principales (Reactivo 4.4)

Se implementaron las siguientes clases principales:

- Clase `Operacion` con atributos: nombre, lista de tareas, estado.
- Clase `Tarea` con atributos: nombre, duración estimada, precondiciones.
- Clase `GestorOperaciones`: punto de entrada y controlador del sistema.
- Clase `ControladorUI`: responsable de inicializar y gestionar la IGU (JavaFX).

Las clases auxiliares fueron esbozadas como interfaces sin comportamiento por el momento.

7.4 Reporte de actividades

Durante esta etapa, el equipo mantuvo el flujo de trabajo colaborativo a través de GitHub y Discord. Se utilizaron listas de verificación para los estados de las Alfas. Se reforzaron los siguientes estados:

- Requisitos::Aceptable
- Producto de software::Arquitectura elegida
- Trabajo::Concluido (parcial)
- Forma de trabajo::Trabajando bien

Comentarios finales:

- Brayan: Me enfoqué en actualizar los diagramas UML y asegurar la trazabilidad de requisitos.

- Raymundo: Realicé la implementación base y enfoqué esfuerzos en la arquitectura general y control de versiones.

8 Interfaces Gráficas

Con base en los requisitos funcionales y casos de uso del sistema *Gestor de Operaciones Multitarea (GOMT)*, lo más adecuado es diseñar una interfaz modular, clara y funcional que respalde tanto la edición como el control de operaciones y tareas.

1. Panel Principal / Panel de Control

Objetivo: Punto central de navegación. Muestra operaciones activas, botones de control y resumen del sistema.

Componentes sugeridos:

- **Barra de menú:** Acceso a “Tareas”, “Operaciones”, “Acerca de...”, “Salir”.
- **Lista de operaciones:** Tabla con nombre, estado (Ejecutando, Pausada, Finalizada).
- **Botones:**
 - Iniciar operación
 - Pausar operación
 - Finalizar operación
- **Indicadores:** Estado general del sistema (por ejemplo: número de tareas activas).

2. Editor de Operaciones

Objetivo: Crear o modificar operaciones y asignarles tareas.

Componentes sugeridos:

- **Campo de texto:** Nombre de la operación.

- **Lista jerárquica** (tipo árbol): Vista estructurada de tareas asignadas a la operación.
- **Botones:**
 - Agregar tarea
 - Eliminar tarea
 - Reordenar tareas (con flechas o arrastrar y soltar)
- **Botón guardar / cancelar**

3. Editor de Tareas

Objetivo: Crear o editar una tarea individual.

Componentes sugeridos:

- **Campo de texto:** Nombre de la tarea.
- **ComboBox / desplegable:** Tipo de tarea o categoría.
- **Área de texto:** Descripción / comportamiento de la tarea.
- **Campo numérico:** Duración estimada.
- **Checkboxes:** Precondiciones.
- **Botones:**
 - Guardar tarea
 - Cancelar

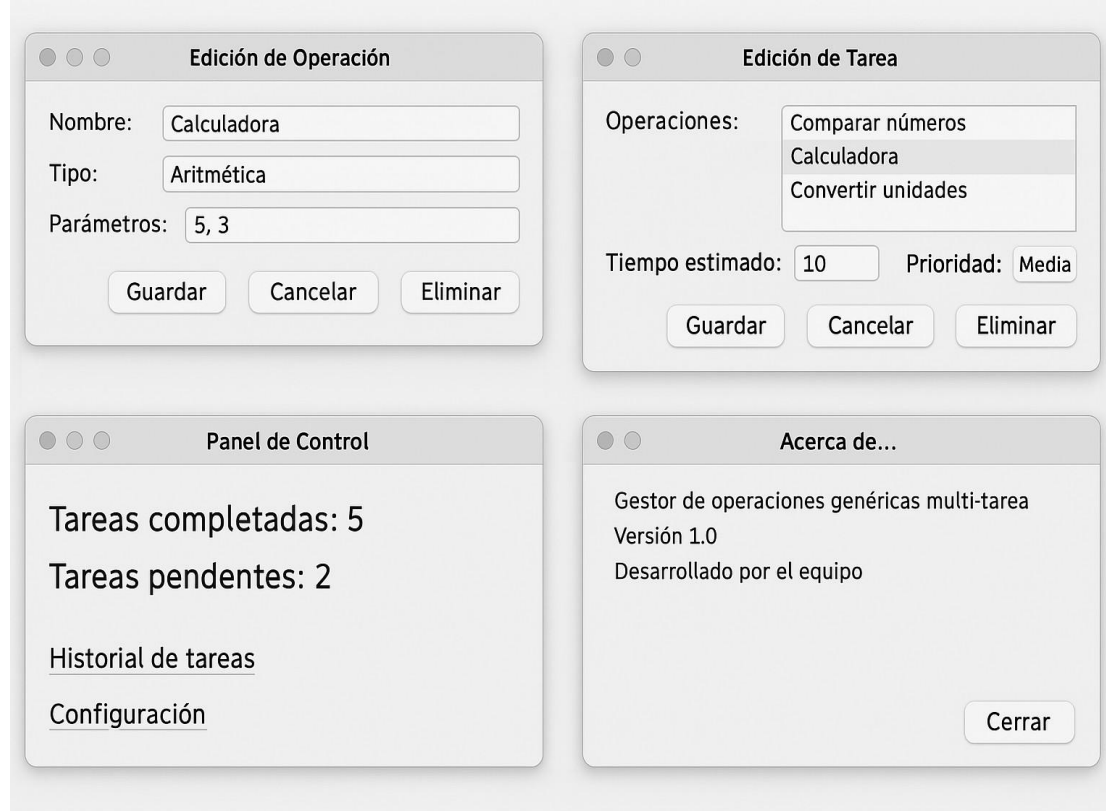
4. Ventana “Acerca de...”

Objetivo: Mostrar los créditos y versión del software.

Componentes sugeridos:

- Nombre del sistema: *Gestor de Operaciones Multitarea (GOMT)*
- Versión: 0.2.0 o más actual
- Autores: Brayan Campos, Raymundo Ramírez
- Matrículas
- Universidad y curso
- Icono o logotipo del proyecto
- Botón: Cerrar

8.1 Mockups



Edición de Operación

Nombre:

Tipo:

Parámetros:

Edición de Tarea

Operaciones:

Suma
Multiplicación
Concatenar

Tiempo: Prioridad:

Panel de Control

Dashboard

Tareas completadas: 5
Operaciones actuales: 3

Historial

[Ver historial](#)

Configuración

[Ajustes](#)

Acerca de...

Versión 1.0

Gestor de operaciones genéricas multi-tarea

Equipo:

Laura Martínez
Sergio Perez
Ana Gómez

Edición de Operación

Nombre:

Tipo:

Parámetros:

Edición de Tarea

Operaciones:

Operacione 1
Operacione 2
Operacione 3

Tiempo:

Prioridad:

Panel de Control

Estadísticas

Configuración

Acerca de...

Historial de tareas

Acerca de

Gestor de operaciones
versión 1.0

9 Extensión Practica 3

9.1 Reactivo 4.1 – Seguimiento de la forma de trabajo

9.1.1 Descripción general

Durante esta tercera etapa del proyecto “**Gestor de Operaciones Multitarea (GOMT)**”, el equipo **Operaciones Plus**, conformado por **Brayan Campos** y **Raymundo Ramírez**, mantuvo el desarrollo colaborativo con base en la organización, herramientas y entorno de trabajo definidos en la Práctica 2, conforme a las secciones **2.4 (recursos)** y **2.1.1 (organización del equipo)**.

9.1.2 Herramientas y entorno establecidos

Recurso	Estado durante la etapa actual
Lenguaje de programación	Java 21
Framework GUI	JavaFX 21
Entorno de desarrollo	NetBeans
Control de versiones	GitHub (repositorio público)
Comunicación del equipo	Discord (canales por temas)
Repositorio de documentación	Dentro del mismo repositorio de GitHub

9.1.3 Organización del repositorio de trabajo

El equipo estructuró su repositorio en GitHub del siguiente modo como se mostrará en el link:

<https://github.com/BrayanCG1/Dise-o-de-Software.git>

9.1.4 Evidencia de seguimiento en el historial de cambios

Se observan **commits frecuentes por ambos miembros**, distribuidos de forma equitativa:

- Brayan: actualización de diagramas UML, edición de la ERS, documentación
- Raymundo: implementación de clases, control de versiones, arquitectura general.

9.1.5 Organización del equipo

Integrante	Rol general	Rol técnico	Actividades
Brayan Campos	Líder	Responsable de ERS y UML	Documentación y diseño
Raymundo Ramírez	Líder	Responsable de solución y control de versiones	Implementación y arquitectura

9.1.6 Evaluación según criterios

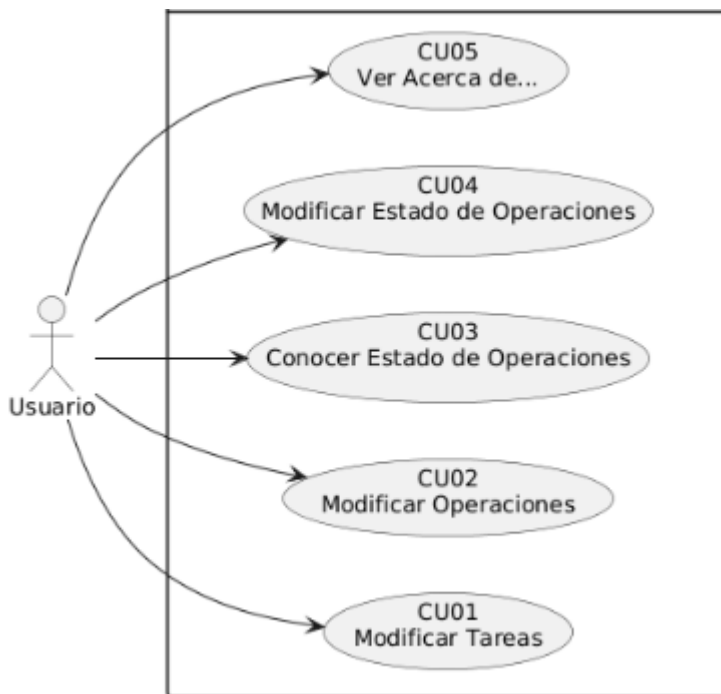
Criterio	Evidencia disponible	Cumplimiento
Cambios mantenidos en el repositorio	Sí, repositorio con historial visible	✓
Uso de herramientas y entorno definidos en la práctica anterior	Sí, todos los recursos de la sección 2.4 se usaron	✓
Participación grupal y equitativa en el desarrollo	Sí, evidenciado en commits y división de tareas	✓
Historial de documentación actualizado	Sí, incluyendo versiones de ERS y diagramas	✓

10. Reactivo 4.2 – Gestión de diseño y documentación

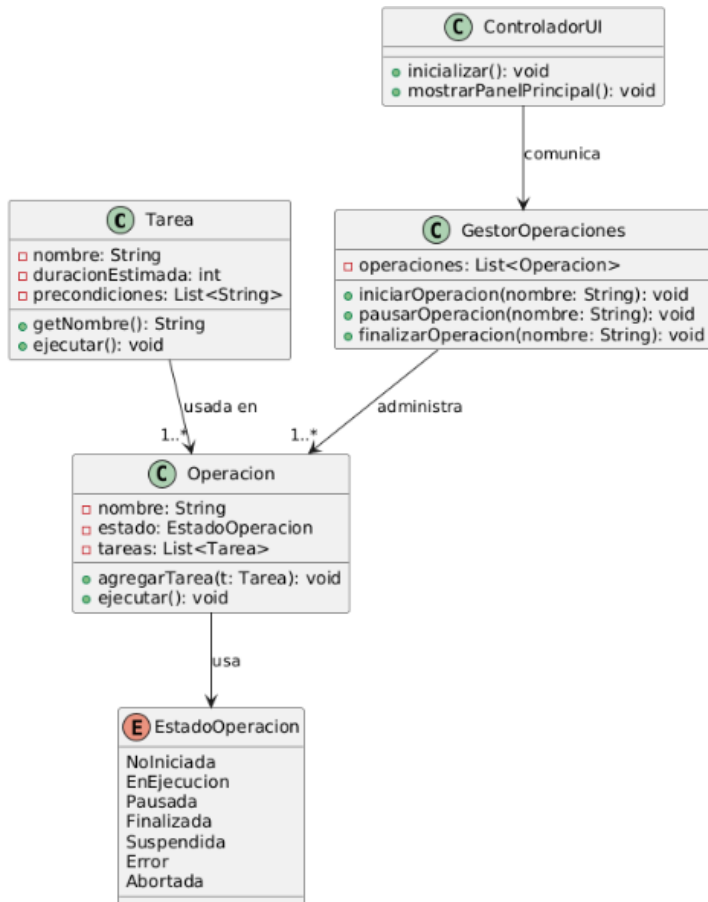
10.1 Descripción general

Durante esta etapa, el equipo **Operaciones Plus** (Brayan Campos y Raymundo Ramírez) actualizó y amplió toda la documentación de diseño a partir del trabajo base establecido en la Práctica 2. Las actualizaciones reflejan los ajustes realizados en la arquitectura, requisitos y mecanismos de operación del sistema GOMT.

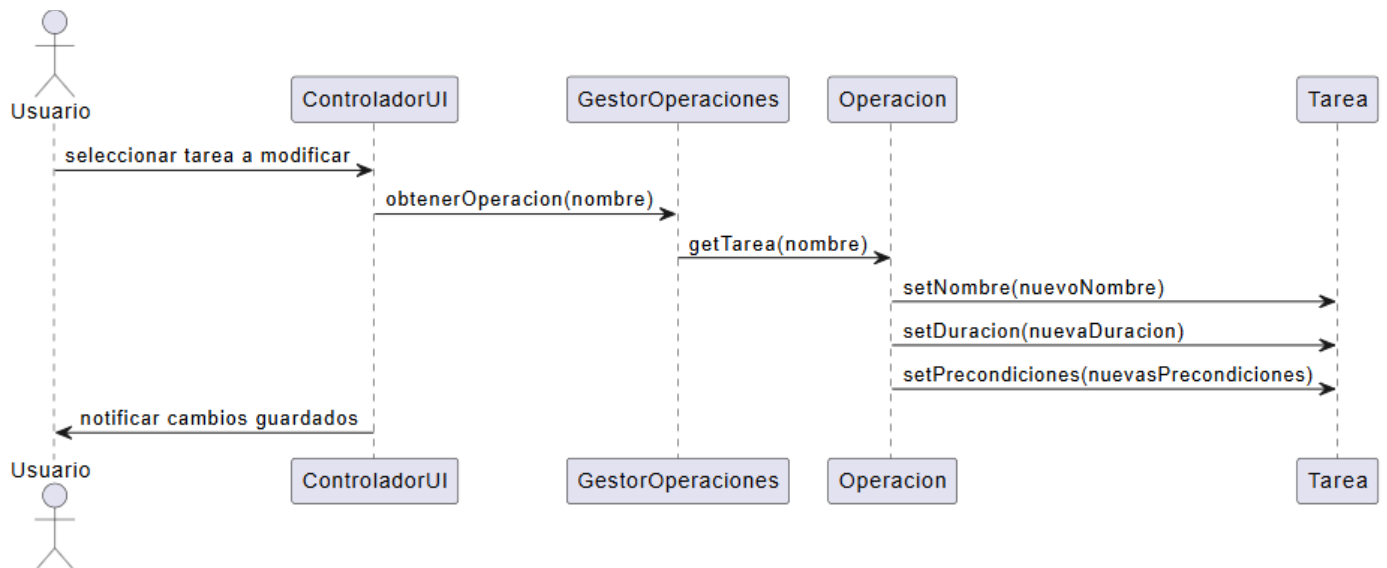
10.1.1 Formatos de Casos de Uso



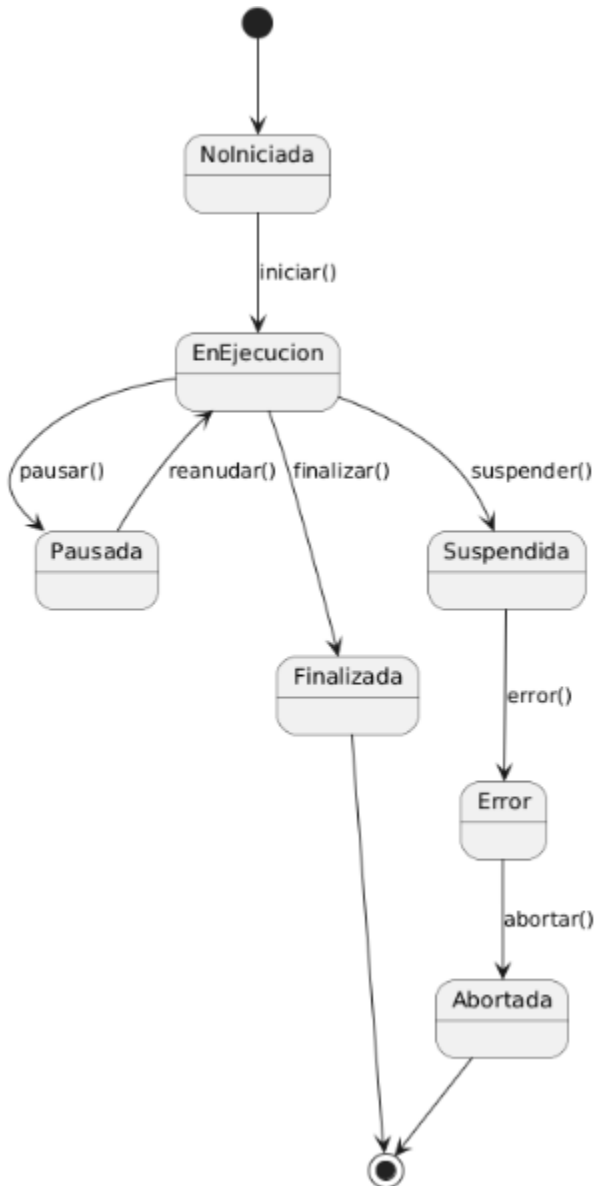
10.1.2 Diagramas de Clase



10.1.3 Diagramas de Secuencia



10.1.4 Diagrama de Estados



10.1.5 Especificación de Requerimientos de Software (ERS)

- **Versión actual:** 0.1.8
- **Cambios realizados:**
 - Se reasignaron responsabilidades de RFs
 - Se clarificaron las condiciones de uso de cada función
 - Se adaptaron descripciones conforme a los mockups e interfaces implementadas
 - Se consolidaron características técnicas y pruebas mínimas

11. Reactivo 4.3 – Mínimo Producto Viable

11.1.1 Descripción general

El equipo **Operaciones Plus** desarrolló una primera versión funcional del **Gestor de Operaciones Multitarea (GOMT)** como **mínimo producto viable (MPV)**. Esta versión cumple con los requisitos funcionales más relevantes definidos en la **ERS v0.1.8**, priorizando aquellas características fundamentales para validar el concepto de gestión de operaciones configurables.

11.1.2 Características clave implementadas

De acuerdo con la **ERS**, se abordaron los siguientes **requisitos funcionales críticos (prioridad alta ≥ 4)**:

ID	Función	Estado en el MPV	Responsable
RF01	Crear nueva tarea	✓ Implementado	Brayan Campos
RF02	Modificar tarea existente	✓ Implementado	Raymundo Ramírez
RF04	Crear operación	✓ Implementado	Raymundo Ramírez
RF05	Modificar operación existente	✓ Implementado	Brayan Campos
RF06	Consultar estado de operaciones	✓ Implementado	Raymundo Ramírez
RF07	Modificar estado de operaciones	✓ Implementado	Brayan Campos
RF08	Mostrar sección "Acerca de..."	✓ Implementado	Raymundo Ramírez

11.1.3 Componentes funcionales del sistema

Interfaz gráfica (JavaFX 21)

- **Panel de control principal:** muestra la lista de operaciones y botones de control.
- **Editor de tareas:** formulario para nombre, duración, descripción y precondiciones.
- **Editor de operaciones:** vista jerárquica (tipo árbol) para agregar, modificar o reordenar tareas.
- **Ventana “Acerca de...”:** información de versión, autores, curso y UACM.

11.1.4 Lógica funcional (Java 21)

- Clases `Tarea`, `Operacion`, `GestorOperaciones`, `ControladorUI` implementadas.
- Transiciones de estados entre operaciones (`NoIniciada`, `EnEjecucion`, `Pausada`, `Finalizada`, etc.).
- Controladores conectados a eventos de la GUI.

11.1.5 Calidad de implementación

- Código estructurado con buenas prácticas (nombres claros, separación de lógica y vista).
- Pruebas básicas manuales de integridad: creación, edición y control de operaciones funcionales.
- Flujo de trabajo validado conforme a los casos de uso documentados.