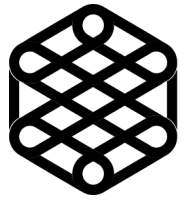




Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno



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

Proyecto del curso

NOMBRE DEL PRODUCTO: Gestor de Operaciones Multitarea
(GOMT).
Especificación de Requerimientos de Software
Versión: 0.1.0
28 de abril 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

1	Introducción	1
1.1	Alcances	1
2	Características del producto	1
2.1	Funciones	1
2.1.1	Requisitos Funcionales	1
2.2	Entornos, Entradas y Salidas	2
2.2.1	Requisitos de Interfaz	2
2.3	Características y Restricciones Adicionales	2
2.3.1	Requisitos de Calidad en Uso	2
2.3.2	Requisitos Técnicos y de Pruebas	2
2.4	Usuarios	2
2.4.1	Requisitos de Usabilidad	2
3.	Casos de Uso	2
CU01	– Modificar Tareas	2
CU02	– Modificar Operaciones	2
CU03	– Conocer Estado de Operaciones	2
CU04	– Modificar Estado de Operaciones	3
CU05	– Acerca de...	3
4.	Diagramas UML	3
4.1	Diagrama de Casos de Uso	3
5.	Historial de Cambios	5
6.1.	Designación de roles y organización del equipo	6
6.2.	Especificación de Requerimientos de Software	6
6.3.	Propuesta de solución	6
6.4.	Seguimiento del trabajo	7
6.5.	Reporte de actividades	7
	Roles	7
	Explicación detallada de roles:	8

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

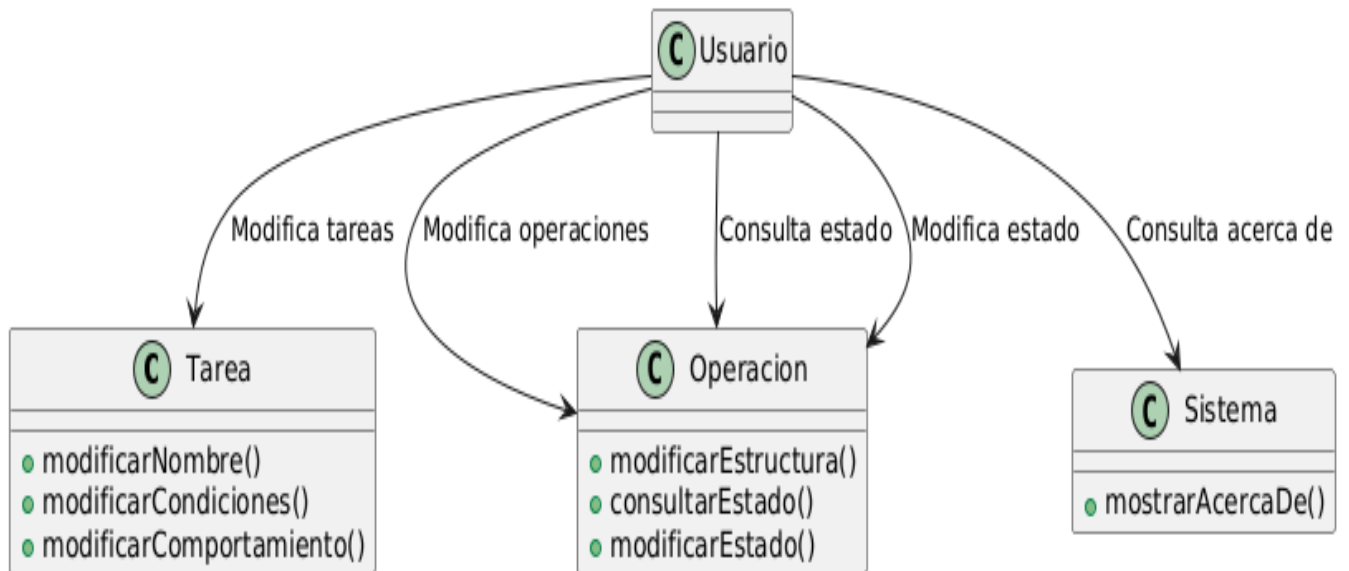
Descripción: Muestra información de la versión del software, autores y derechos.

4. Diagramas UML

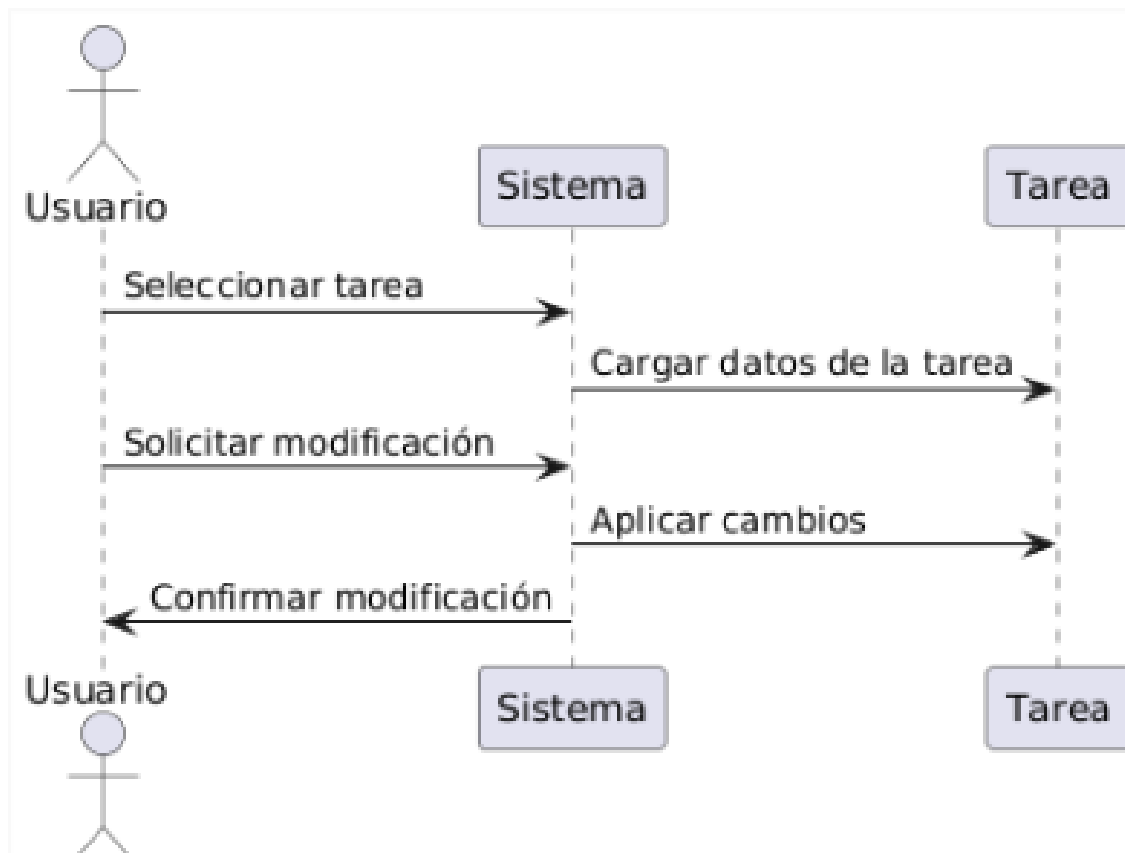
4.1 Diagrama de Casos de Uso



4.2 Diagrama de Clases

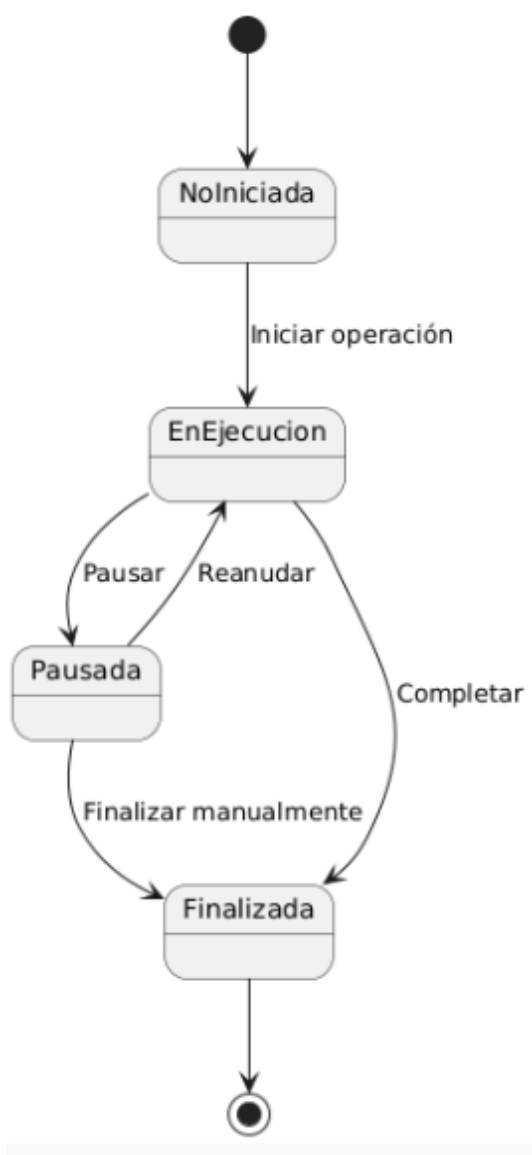


4.3 Diagrama de Secuencia



NOMBRE DEL PRODUCTO: Gestor de Operaciones Multitarea (GOMT).

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.

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").

NOMBRE DEL PRODUCTO: Gestor de Operaciones Multitarea (GOMT).

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.