PROYECTO 2

202346773 – Marco Antonio Martínez Gutiérrez
202342112 – Douglas Estuardo Cetino Ciramagua

Resumen

El proyecto 2 consiste en desarrollar una plataforma web para simular la logística de la empresa LogiTrack S.A. Se requiere un backend en Java Spring Boot y un frontend en React. El diseño debe basarse completamente en programación orientada a objetos, sin el uso de bases de datos persistentes, almacenando toda la información en memoria. La configuración inicial se carga desde un archivo XML que contiene datos sobre centros de distribución, rutas, mensajeros, paquetes y solicitudes. El backend gestionará las reglas de negocio, incluyendo la asignación de mensajeros, validación de capacidades y el procesamiento de solicitudes mediante una cola de prioridad. El frontend permitirá al usuario visualizar y administrar los recursos, además de iniciar el procesamiento de los envíos. Al finalizar, se generará un archivo XML de salida con las estadísticas de la operación. Este proyecto busca la aplicación de lógica y diseño de software, incluyendo la implementación de endpoints REST y la entrega documentada en GitHub.

Palabras clave
LogiTrack, logística, API REST, POO, XML

Abstract
Project 2 involves developing a web platform to simulate the logistics of the company LogiTrack S.A. It requires a backend in Java Spring Boot and a frontend in React. The design must be based entirely on object-oriented programming (OOP), without the use of persistent databases, storing all information in memory. The initial configuration is loaded from an XML file that contains data about distribution centers, routes, messengers, packages, and requests. The backend will manage the business rules, including messenger assignment, capacity validation, and request processing through a priority queue. The frontend will allow the user to visualize and manage resources, in addition to initiating the processing of shipments. Upon completion, an output XML file will be generated with the operation's statistics. This project seeks the application of logic and software design, including the implementation of REST endpoints and documented delivery on GitHub.

Keywords

LogiTrack, logistics, API
REST, OOP, XML

Introducción

En el ámbito tecnológico actual, la gestión eficiente de la logística y la cadena de suministro es fundamental para el éxito empresarial. En este contexto, surge el Proyecto 2, enfocado en desarrollar una solución integral que simule las operaciones de la empresa LogiTrack S.A. El objetivo principal es aplicar los principios de la programación orientada a objetos (POO) para modelar un sistema logístico real.

La solución propuesta integra un backend robusto desarrollado en Java Spring Boot que expone una API REST, interactuando con un frontend dinámico creado con React. El sistema gestionará recursos como centros de distribución, rutas, mensajeros, paquetes y solicitudes, manteniendo toda la información en memoria y utilizando archivos XML para la configuración inicial y los reportes finales. Este proyecto permitirá comprender la aplicación práctica de conceptos fundamentales de ingeniería de software y el diseño de arquitecturas modulares para resolver problemas complejos de gestión operativa.

Descripción del proyecto

LogiTrack es un sistema que arranca con un archivo XML que le da toda la estructura inicial. En ese archivo, definimos la infraestructura logística: los centros de distribución, las rutas entre ellos, los mensajeros, los paquetes y las solicitudes de envío que tienen los clientes.

El sistema funciona con estos componentes principales:

- **Centros de distribución:** Los lugares donde se guardan los paquetes, con una capacidad límite.
- **Mensajeros:** La gente que reparte los paquetes, tienen una capacidad máxima de carga.
- **Rutas:** Las conexiones entre centros, con una distancia definida.
- **Paquetes:** Los artículos que se envían, con un estado (PENDIENTE, EN_TRANSITO, ENTREGADO).
- **Solicitudes:** Las tareas de los clientes (como hacer un envío normal) que el sistema debe procesar.

Una vez que cargamos el XML, el programa organiza internamente toda la información con las estructuras de datos que implementamos. Luego, nos permite interactuar con una API REST para hacer consultas, administrar recursos y procesar las solicitudes.

Análisis y postura sobre el proyecto

El enfoque que le dimos a LogiTrack tiene puntos muy sólidos. Al usar programación orientada a objetos logramos que el sistema sea súper modular, es fácil entender qué hace cada parte porque cada componente actúa como una entidad real de la logística. También decidimos

armar nuestras propias estructuras de datos, como las listas y colas de prioridad, lo que nos dio un control total sobre cómo fluye la información. Es cierto que quizás el rendimiento no sea el más capacitado para algo a gran escala porque no usamos librerías optimizadas, pero el valor real de LogiTrack está en el aprendizaje. Priorizamos entender cómo funciona "el motor" por dentro antes que la velocidad pura y eso ayuda muchísimo a entender cómo se gestionan los envíos en la vida real.

La Lógica Del Código

Para que el código fuera manejable y claro, usamos el enfoque de Programación Orientada a Objetos (POO). Creamos una clase para cada cosa (Centro, Ruta, Mensajero, etc.), lo que hizo más fácil organizar la información y ver cómo se relacionaban entre sí, algo que plasmamos en nuestro diagrama de clases.

Un punto clave es que no usamos las listas nativas del lenguaje. Tuvimos que crear nuestras propias listas anidadas, nodos, y colas de prioridad para simular cómo se administra la información en un sistema real de este tipo.

Tenemos un par de algoritmos importantes:

- **Gestión de Solicitudes:** Las peticiones de los clientes van a una cola de prioridad. El sistema siempre saca primero la tarea más urgente y revisa si hay recursos disponibles para hacerla (mensajero, ruta, capacidad).
- **Gestión de Estados:** Cuando un paquete se mueve o se entrega, el algoritmo se asegura de que el estado del paquete y del mensajero se actualicen correctamente y que no se puedan revertir los cambios.

Interacción y Funcionalidades

El sistema no es solo un programa que corre solo, está pensado para que el usuario pueda interactuar con él a través del *frontend* en React que consume nuestra API REST. Esto nos da control total sobre la simulación:

- **Cargar Archivo:** La primera opción, es cargar el XML inicial que contiene toda la configuración y las instrucciones a seguir.
- **Gestión de Centros de Distribución:** Aquí podemos ver detalles específicos. Podemos listar todos los centros, buscar uno por su ID para ver cuántos paquetes, mensajeros tiene y su capacidad actual.
- **Gestión de Mensajeros:** Nos permite buscar un mensajero por su ID para ver su estado y capacidad, reasignarlo a otro

centro o listar todos los mensajeros disponibles.

- **Gestión de Paquetes:** Podemos crear nuevos paquetes, listar los que ya están registrados o actualizar sus datos y estado.
- **Gestión de Solicitudes:** Aquí es donde agregamos nuevas tareas a la cola de prioridad, procesamos la solicitud más urgente en ese momento, o procesamos un número específico de solicitudes N en orden de prioridad.

Estructura de Archivos: XML de Entrada y Salida

El corazón de la comunicación del sistema son los archivos XML. Tienen un formato específico que debemos respetar, tanto para leer la entrada como para generar la salida.

Archivo de Entrada

El archivo de entrada (logitrack.xml) es fundamental. Contiene la lista de recursos iniciales (centros, rutas, mensajeros, paquetes y solicitudes pendientes). No se puede modificar después de cargarlo y el *backend* debe validar que todo esté bien estructurado.

Archivo de Salida

Al final de la ejecución, el sistema genera un archivo de salida que resume todo lo que pasó. Este archivo incluye estadísticas detalladas del estado final de cada centro, como la cantidad total de paquetes y mensajeros disponibles, y el estado final de todas las solicitudes y paquetes.

Conclusiones

Desarrollar LogiTrack fue la oportunidad perfecta para poner en práctica todo lo que vimos en teoría sobre sistemas logísticos, POO y estructuras de datos. Con este proyecto nos dimos cuenta de que en sistemas tan complejos, si el código no está bien organizado y las reglas de negocio no se gestionan con cuidado, todo se cae. Como grupo, vemos de suma importancia analizar primero qué hay debajo de cada programa para maximizar el aprendizaje, y eso lo pudimos observar mediante la creación del proyecto. Este proyecto nos deja pensando en cómo llevar estos modelos a algo más grande con bases de datos reales y nos marca el camino para empezar a explorar herramientas más potentes en el futuro, como frameworks de ruteo avanzados o sistemas de gestión de bases de datos.

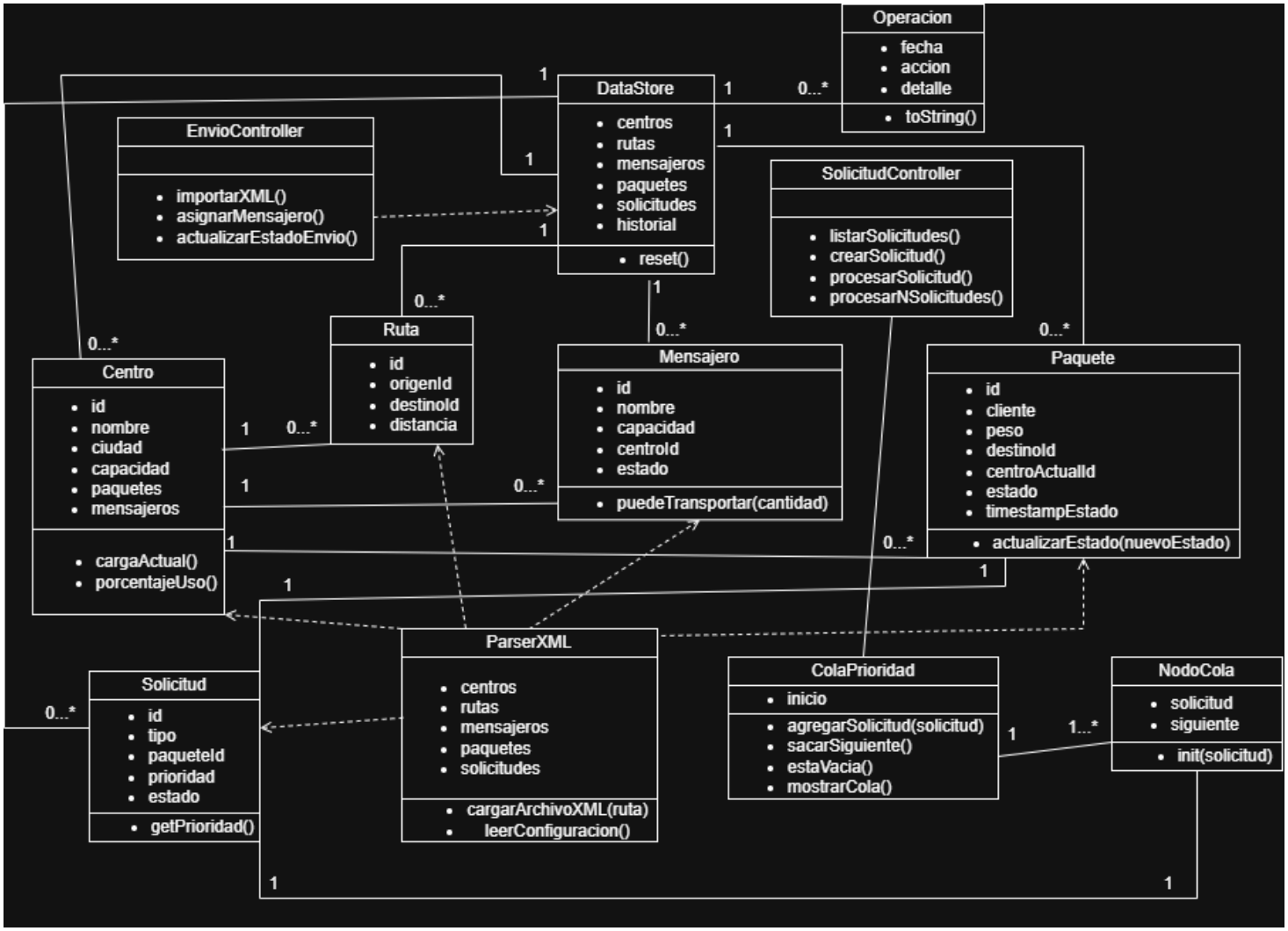

Figura 1. Diagramas de Clases.
Fuente: elaboración propia

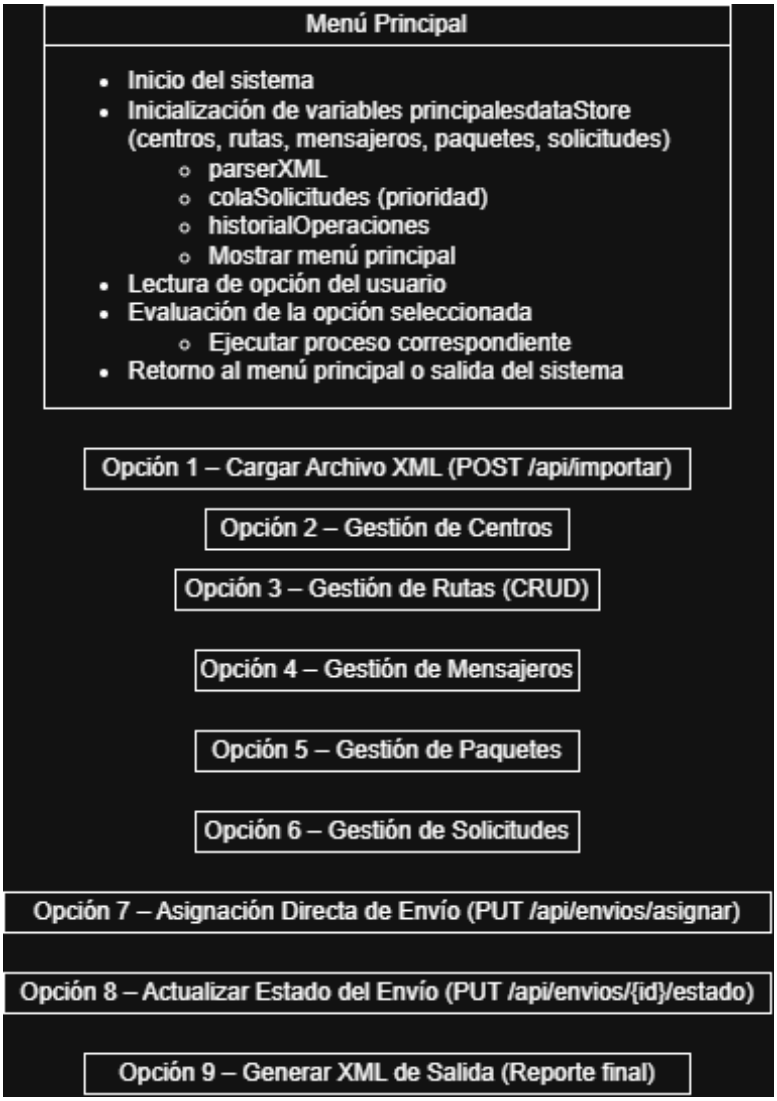

Figura 2 Diagramas de Actividades 0.
Fuente: elaboración propia

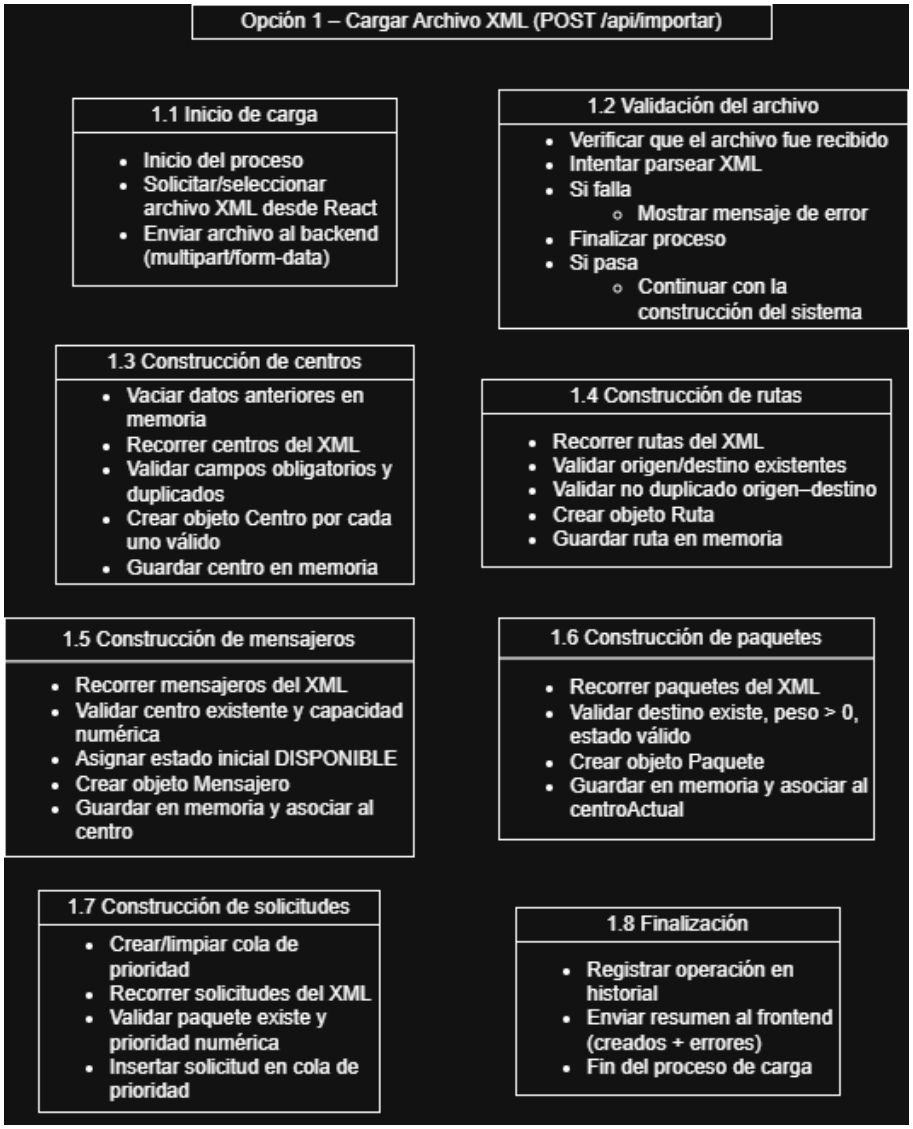

Figura 3. Diagramas de Actividades 1.
Fuente: elaboración propia

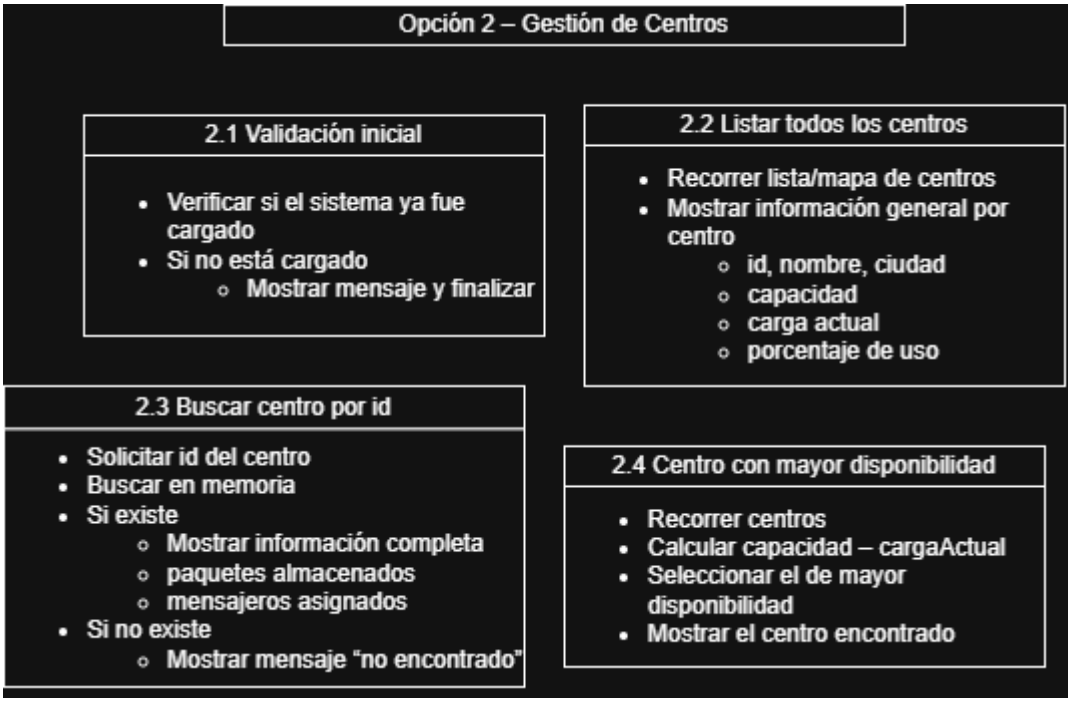

Figura 4. Diagramas de Actividades 2.
Fuente: elaboración propia

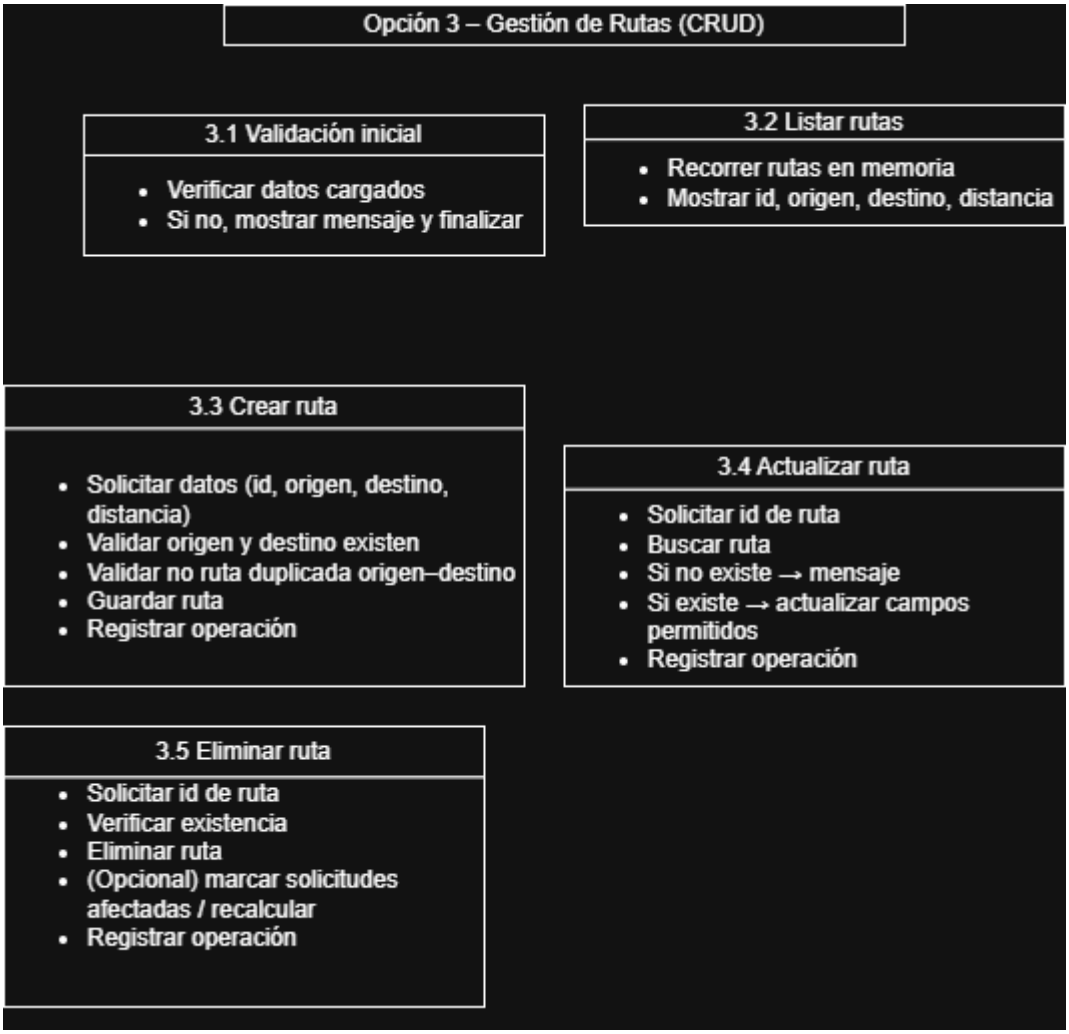

Figura 5. Diagramas de Actividades 3.
Fuente: elaboración propia

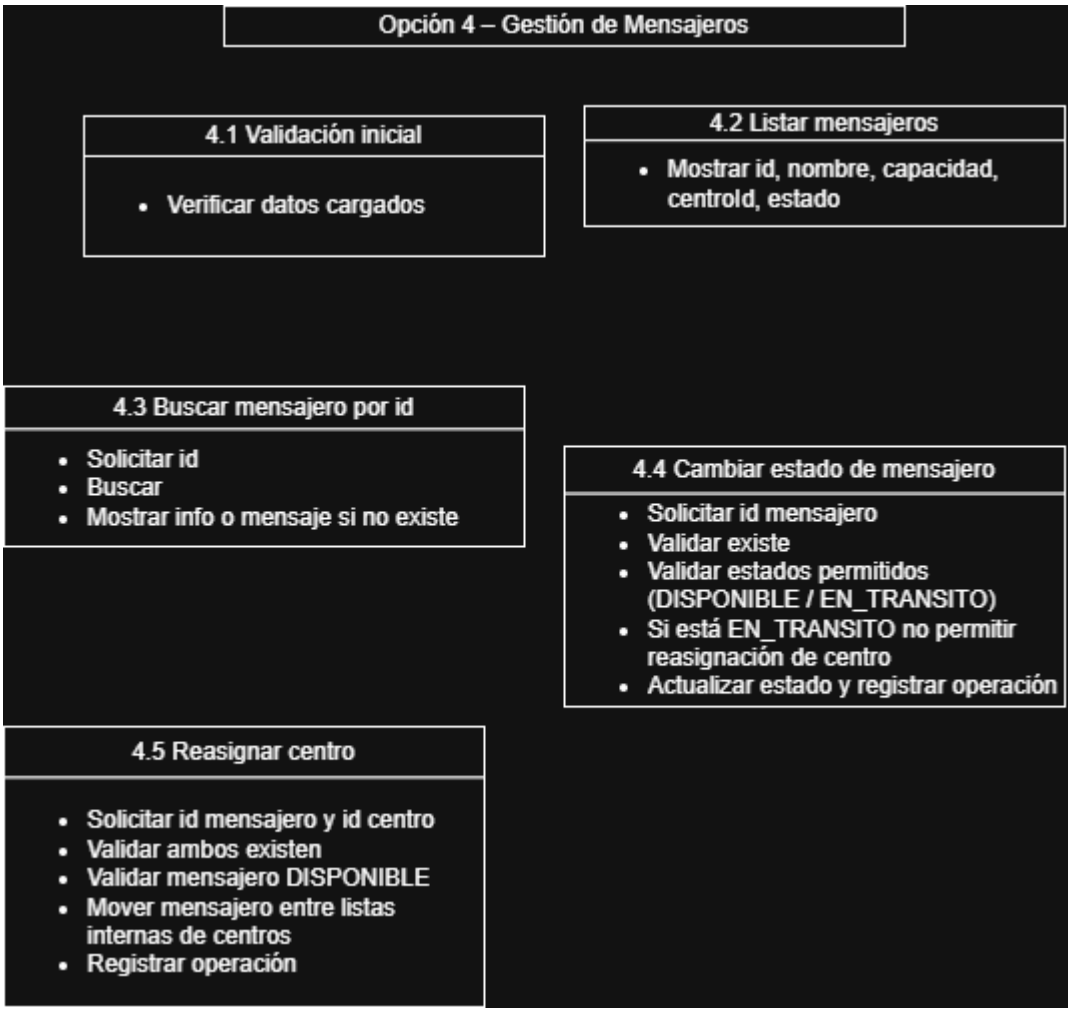

Figura 6. Diagramas de Actividades 4.
Fuente: elaboración propia

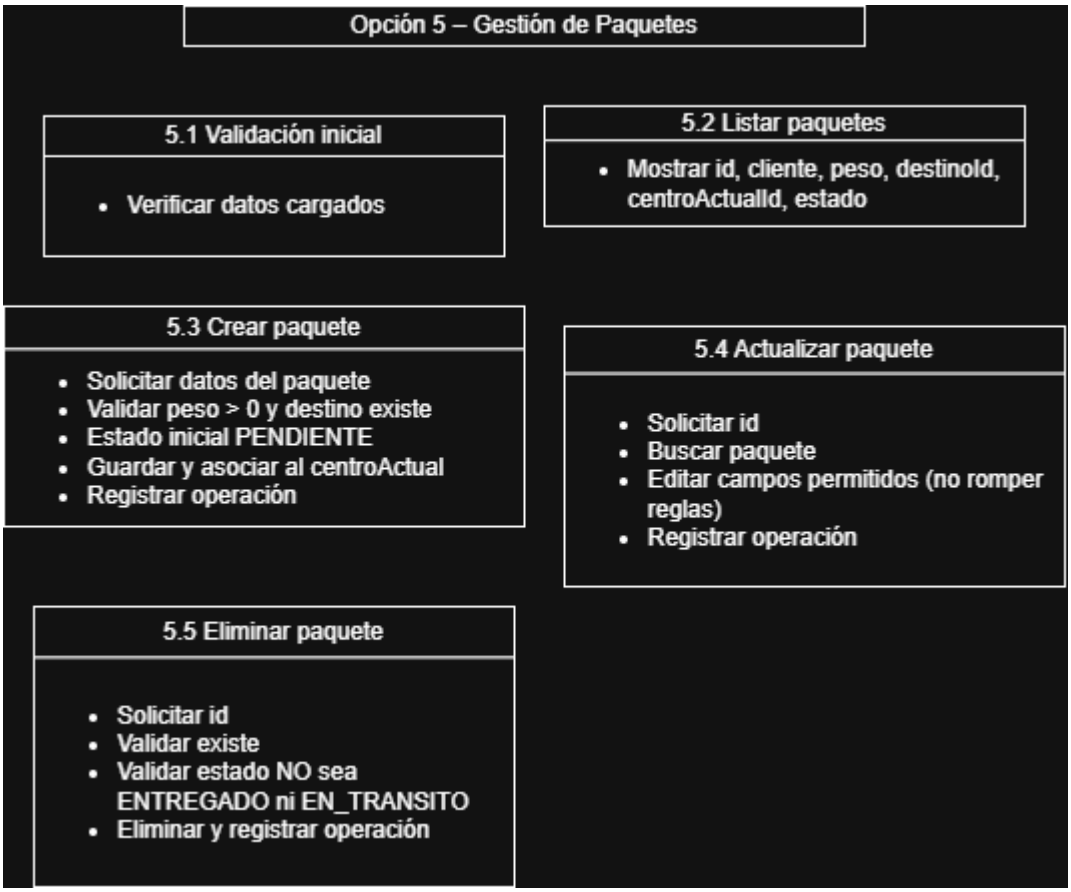

Figura 7. Diagramas de Actividades 5.
Fuente: elaboración propia

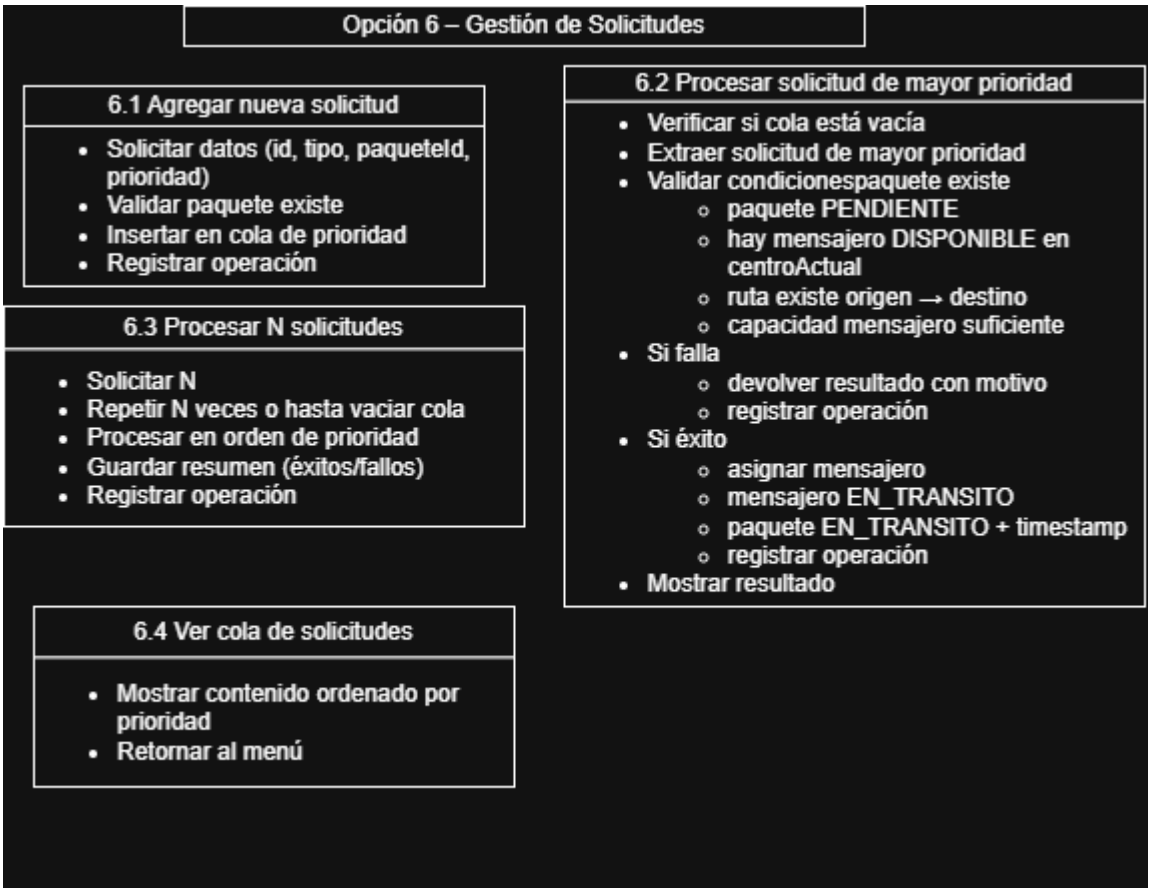

Figura 8. Diagramas de Actividades 6.
Fuente: elaboración propia

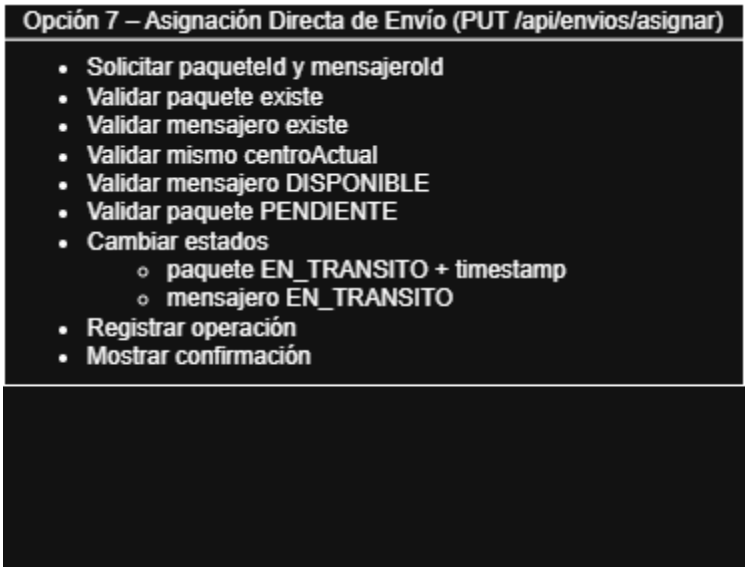

Figura 9. Diagramas de Actividades 7.
Fuente: elaboración propia

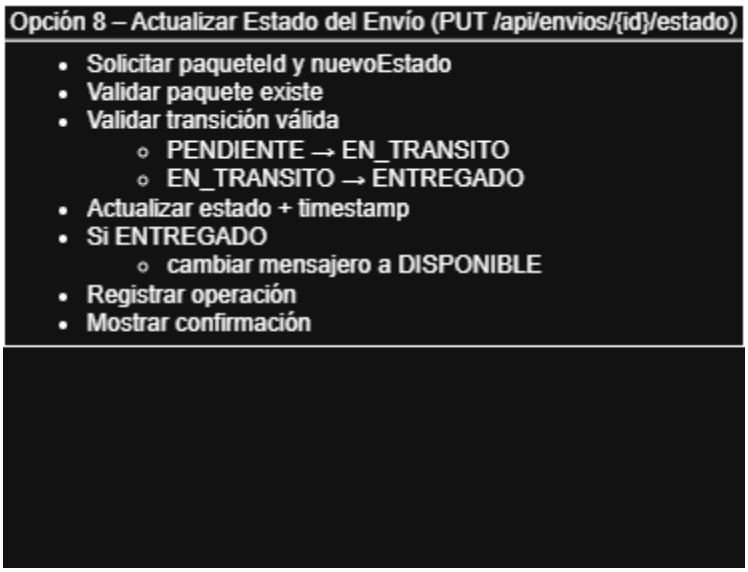

Figura 10. Diagramas de Actividades 8.
Fuente: elaboración propia

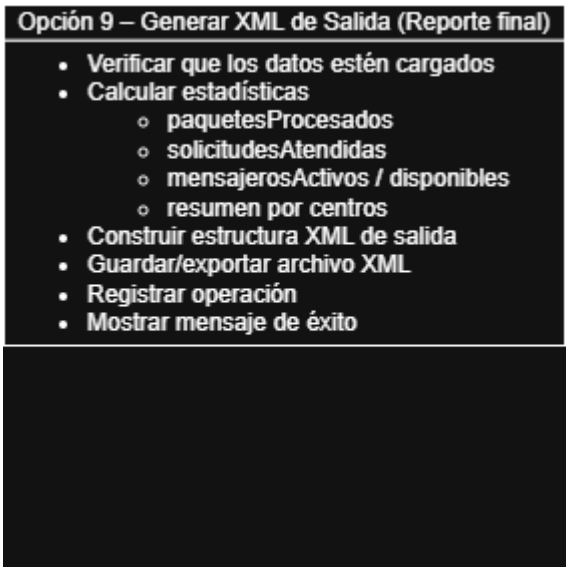

Figura 11. Diagramas de Actividades 9.
Fuente: elaboración propia