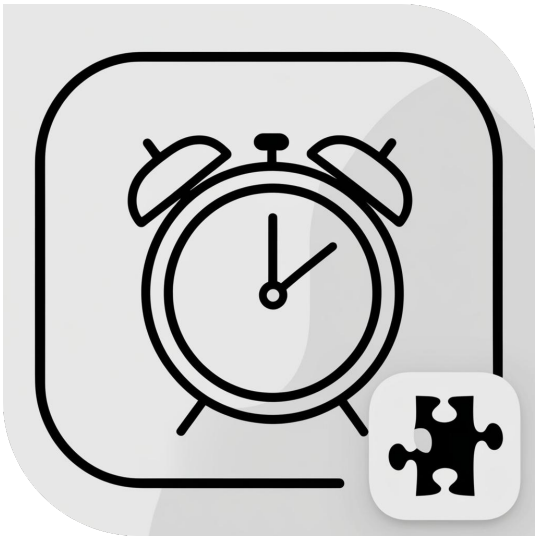
Universidad de Santiago de Chile
Facultad de Ciencia
Departamento de Matemática y Ciencia de la Computación

Ingeniería de Software II

Informe Nº4. Diseño Orientado a Objetos OMT++"

Snoozefest



Integrantes:

Gabriel Carrillo
Erick Aranda
Jose Cuellar
Elías Gangas
Vicente Rojas

Profesor:

Dino Araya

Fecha de entrega:

11 de diciembre del 2024

Índice

Índice de figuras

Índice de tablas

1. Introducción

El presente informe tiene como finalidad ofrecer una visión clara y preliminar del trabajo realizado en la etapa de diseño orientado a objetos, conforme a la metodología OMT++ aplicada al proyecto Snoozefest en curso. En este contexto, se presenta el cuarto informe de avance, en el cual se exponen las principales actividades desarrolladas, los artefactos generados y las consideraciones metodológicas adoptadas en esta fase del proceso de desarrollo de software. El documento se organiza en torno a la aplicación de un diseño que integra el paradigma Modelo-Vista-Controlador(MVC), orientado a la obtención de un modelo de clases que facilite la estructuración coherente de las funcionalidades y el comportamiento del sistema, de manera que cada capa cumpla su rol específico dentro de la arquitectura propuesta. Asimismo, se describen las trazas de eventos que contribuyen a la identificación de los métodos y las interacciones entre las clases, logrando así una caracterización precisa del comportamiento esperado. La información contenida en las secciones posteriores busca incrementar la comprensión del diseño orientado a objetos en OMT++, presentando los elementos conceptuales y técnicos de forma sistemática y coherente. De esta manera, el lector podrá obtener una perspectiva integral de las decisiones tomadas, las justificaciones técnicas y el fundamento metodológico subyacente, factores que resultan esenciales para comprender la base conceptual y operativa del diseño a implementar.

2. Diseño Orientado a Objetos

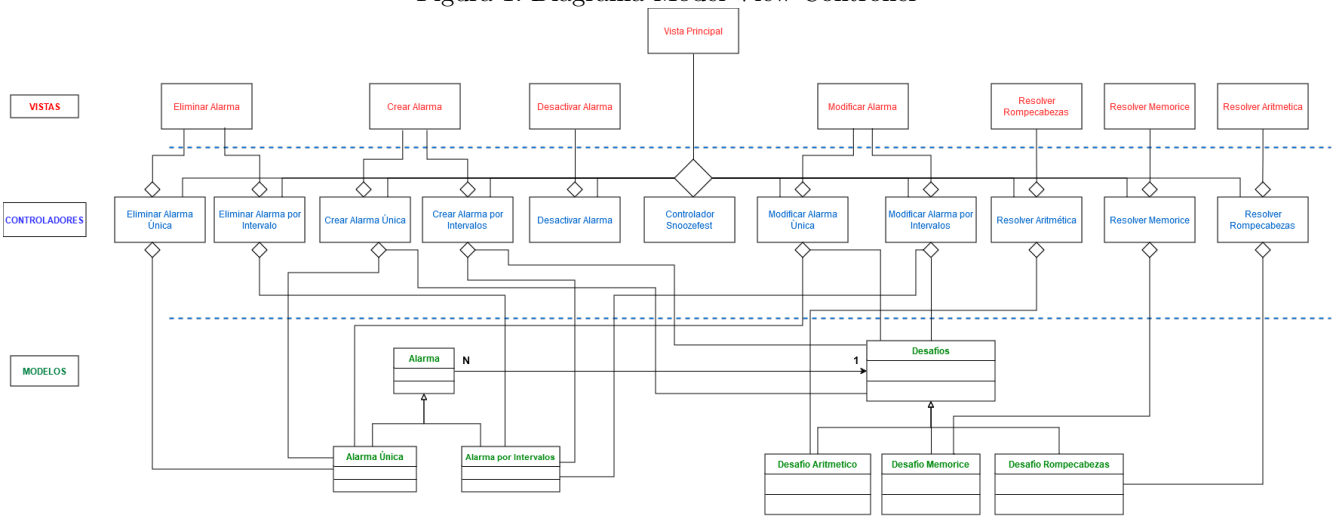
En esta sección, se describen las dos etapas principales del Diseño Orientado a Objetos bajo la metodología OMT++ . Cada etapa del diseño se construye a partir de los resultados obtenidos en etapas anteriores, como el Análisis Orientado a Objetos (AOO) o las especificaciones iniciales del sistema. El propósito de estas etapas es transformar el modelo conceptual en un diseño detallado que pueda ser implementado en un lenguaje de programación, asegurando que las clases y métodos definidos cumplan con los requisitos funcionales y no funcionales del sistema. El diseño de objetos consiste en establecer los elementos estructurales del sistema, mientras que el diseño del comportamiento se centra en definir cómo interactúan estos elementos para realizar las operaciones del sistema.

2.1. Diseño de Objetos

En esta etapa, se define la estructura del sistema a partir de los conceptos identificados durante el análisis. Se crea un modelo de objetos del diseño que incluye:

Las clases del sistema y sus atributos. Los métodos necesarios para cumplir con las operaciones. La relación entre las clases, aplicando patrones de arquitectura como MVC para organizar la solución en capas (Modelo, Vista y Controlador). El diseño de objetos garantiza una organización clara y modular, facilitando la reutilización de componentes y la mantenibilidad del sistema.

Figura 1: Diagrama Model-View-Controller



2.2. Diseño del Comportamiento

Esta etapa complementa el diseño estructural definiendo la interacción entre las clases para cumplir con las operaciones especificadas en el análisis. Utiliza herramientas como:

- **Trazas de eventos:** Representan cómo se comunican los objetos del sistema en respuesta a eventos.
- **Diagramas de secuencia (UML):** modelan las interacciones en un flujo temporal, especificando mensajes, llamadas a métodos y valores retornados.

El diseño del comportamiento asegura que cada operación del sistema esté respaldada por una secuencia de acciones coherente y eficiente, respetando las responsabilidades asignadas a cada clase. Esto garantiza que el sistema funcione de acuerdo con los requisitos y sea adaptable a cambios futuros.

Figura 2: Traza Crear Alarma Única

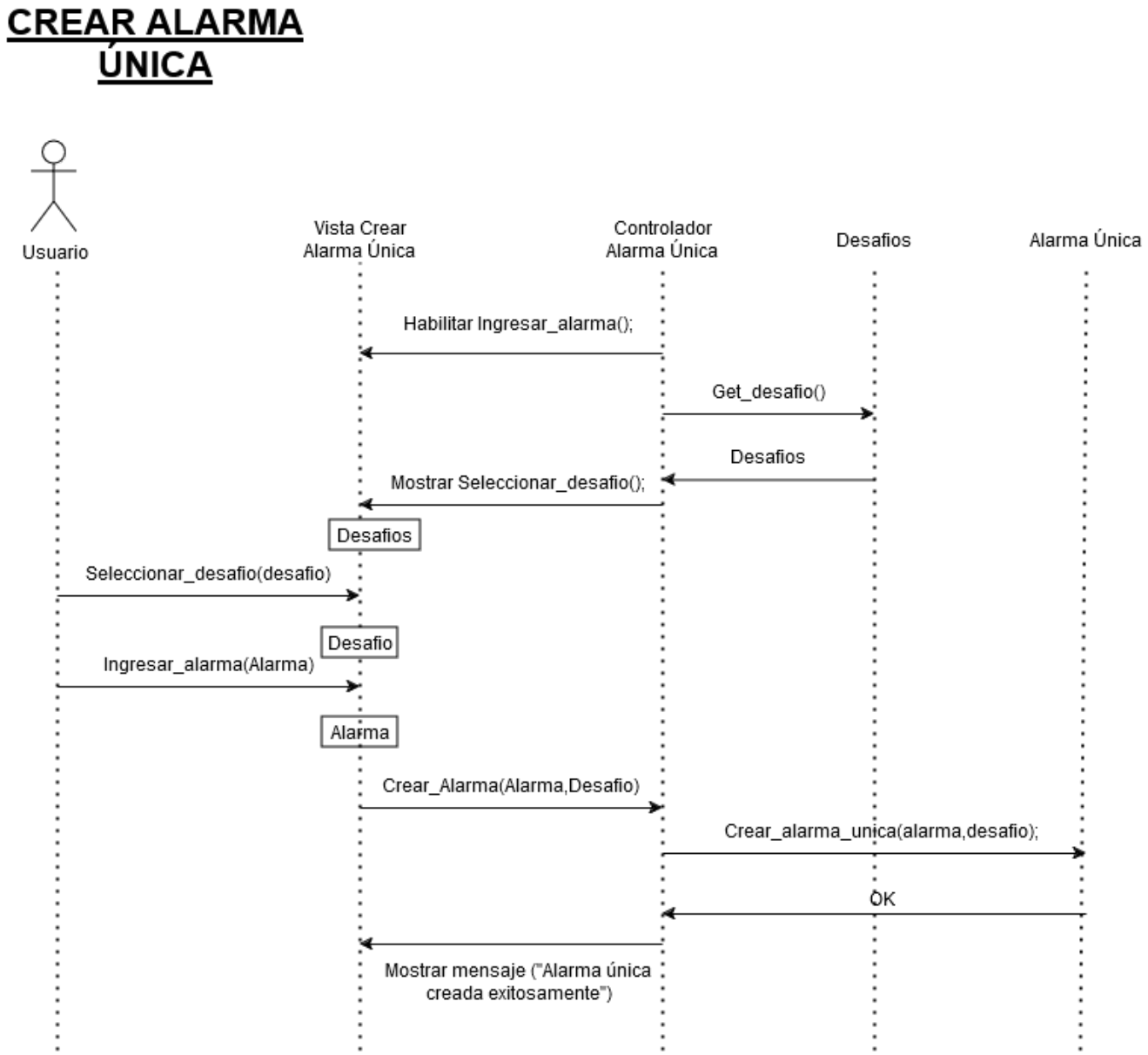


Figura 3: Traza Modificar Alarma Única

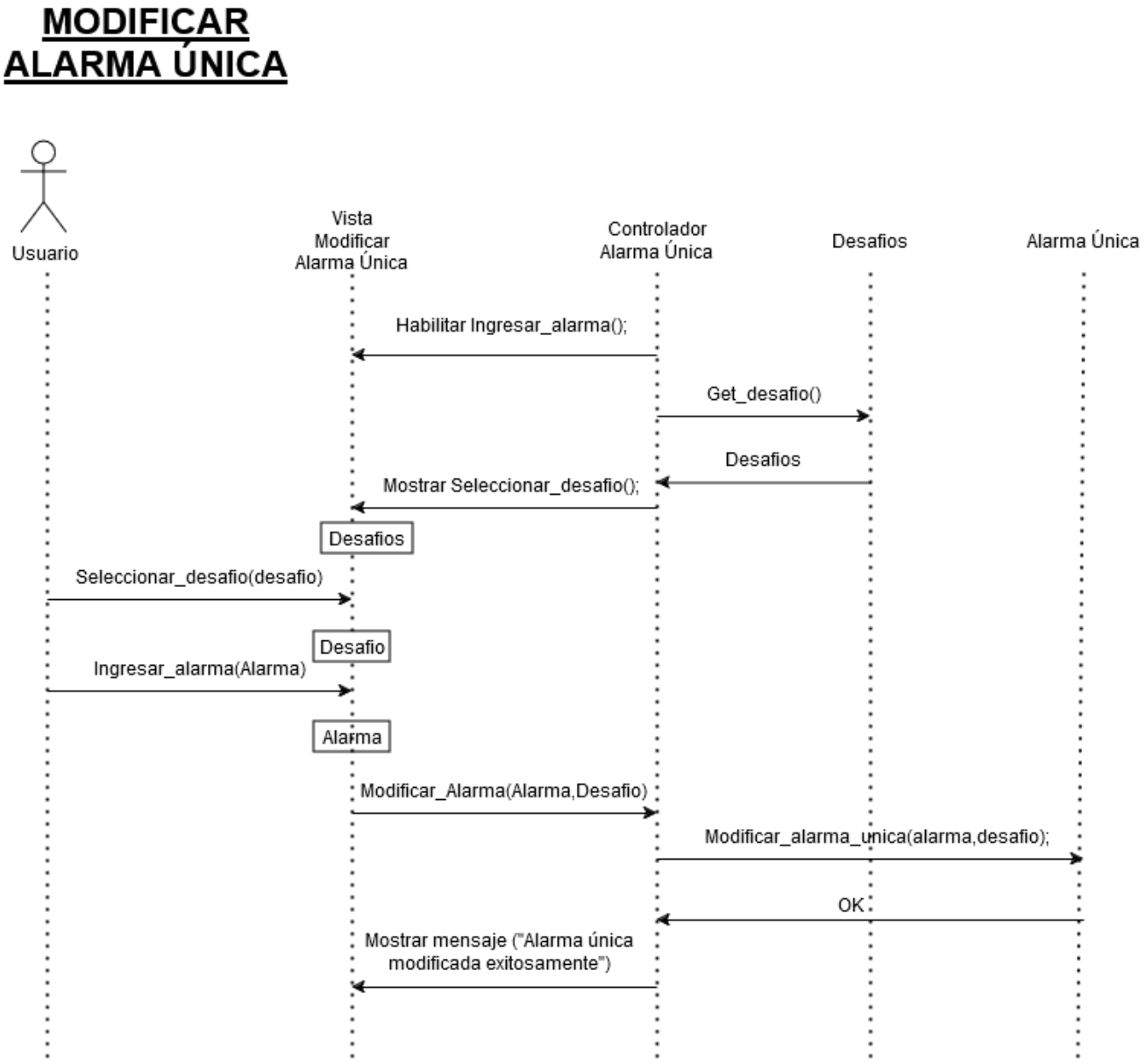


Figura 4: Traza Eliminar Alarma Única

"ELIMINAR ALARMA UNICA"

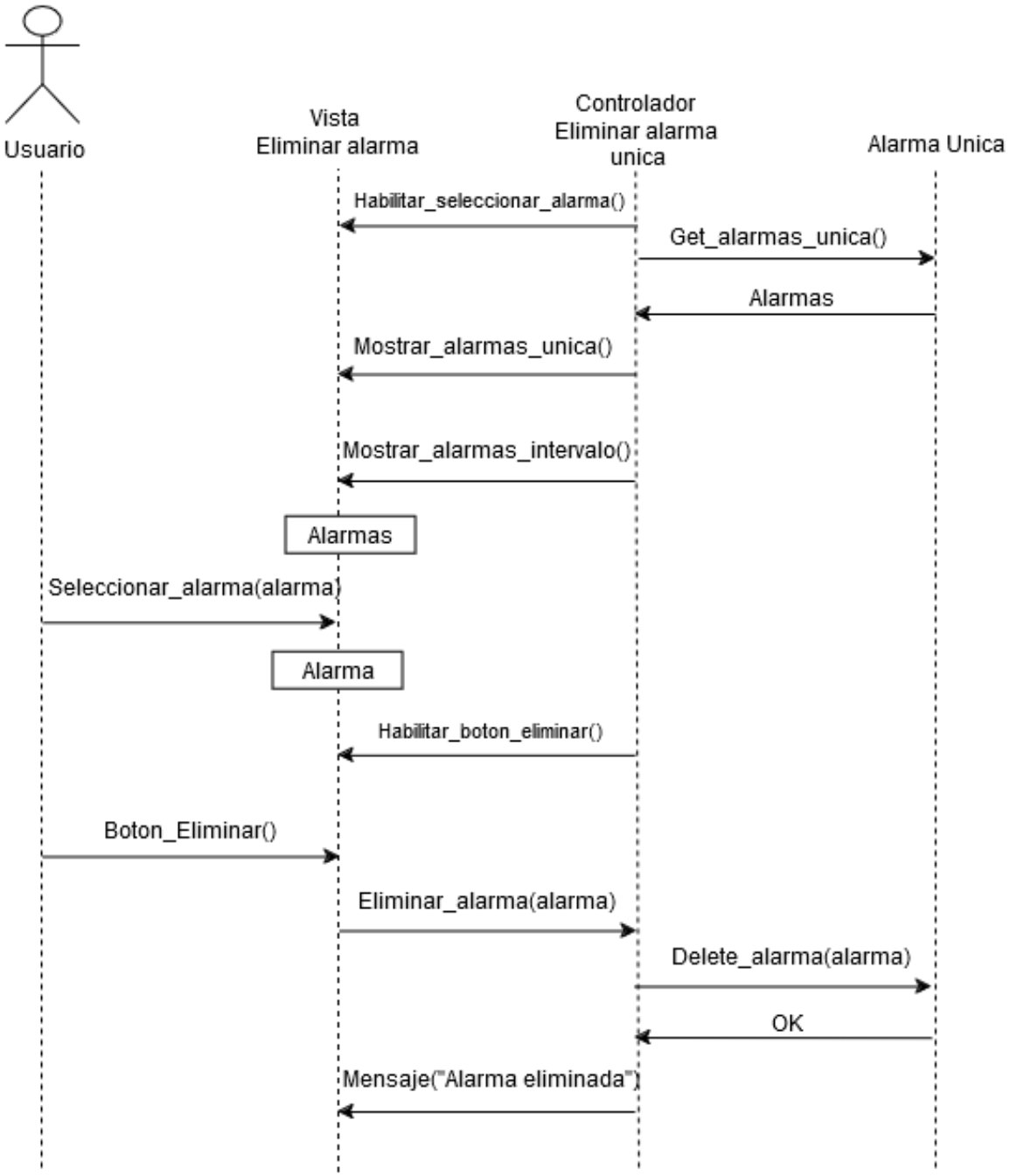


Figura 5: Traza Crear Alarma por Intervalos

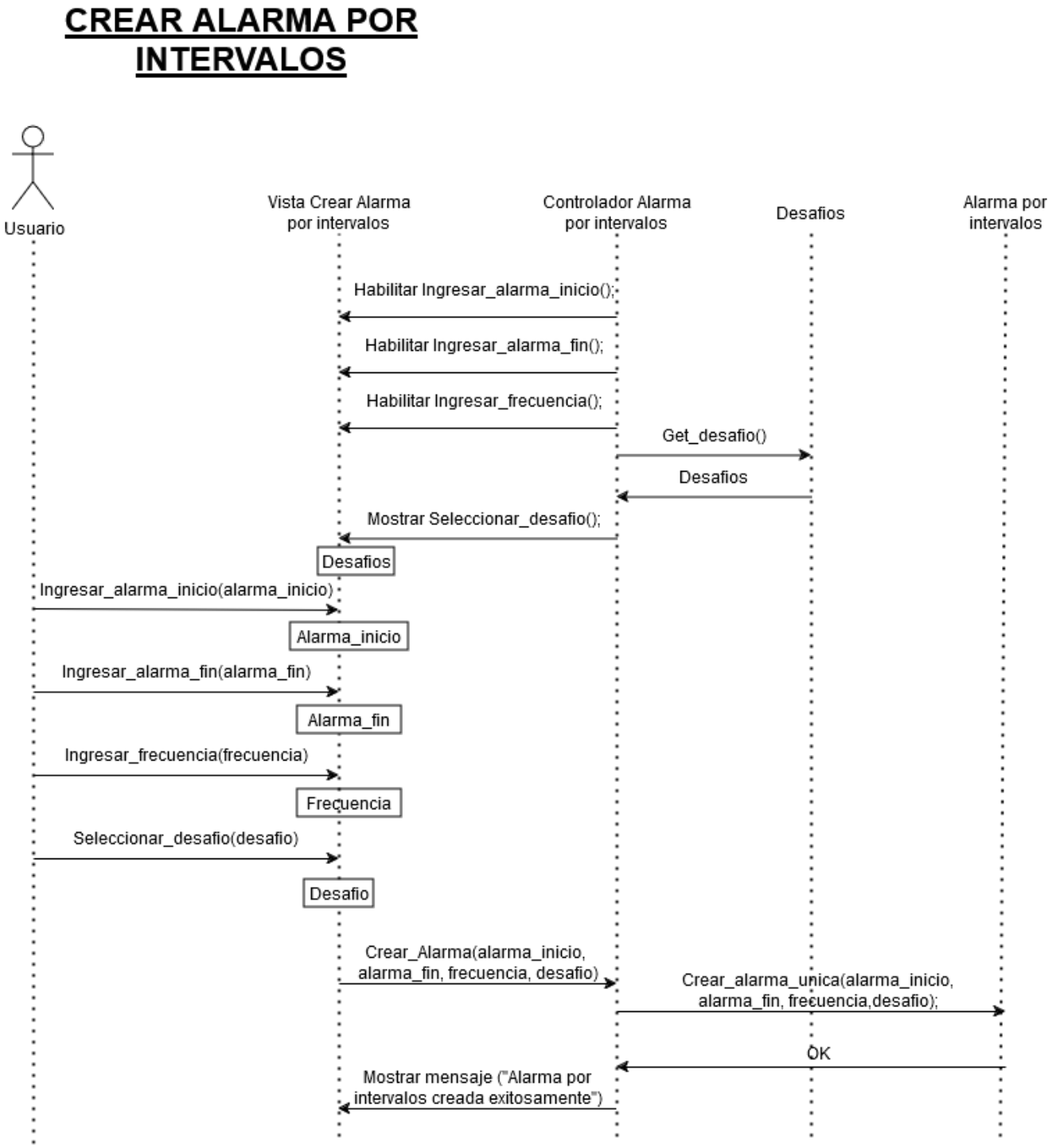


Figura 6: Traza Modificar Alarma por Intervalos

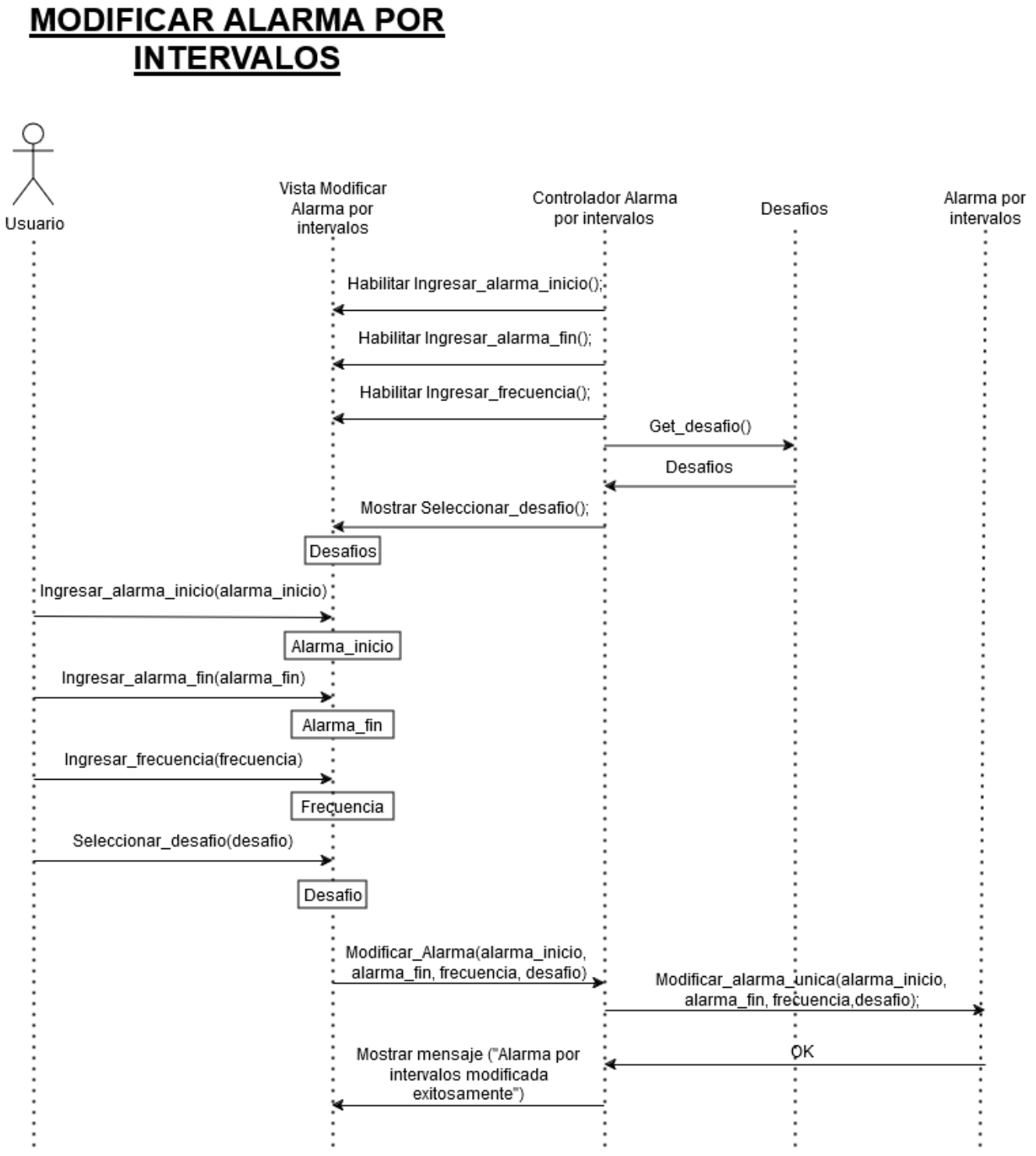


Figura 7: Traza Eliminar Alarma por Intervalos

"ELIMINAR ALARMA POR INTERVALOS"

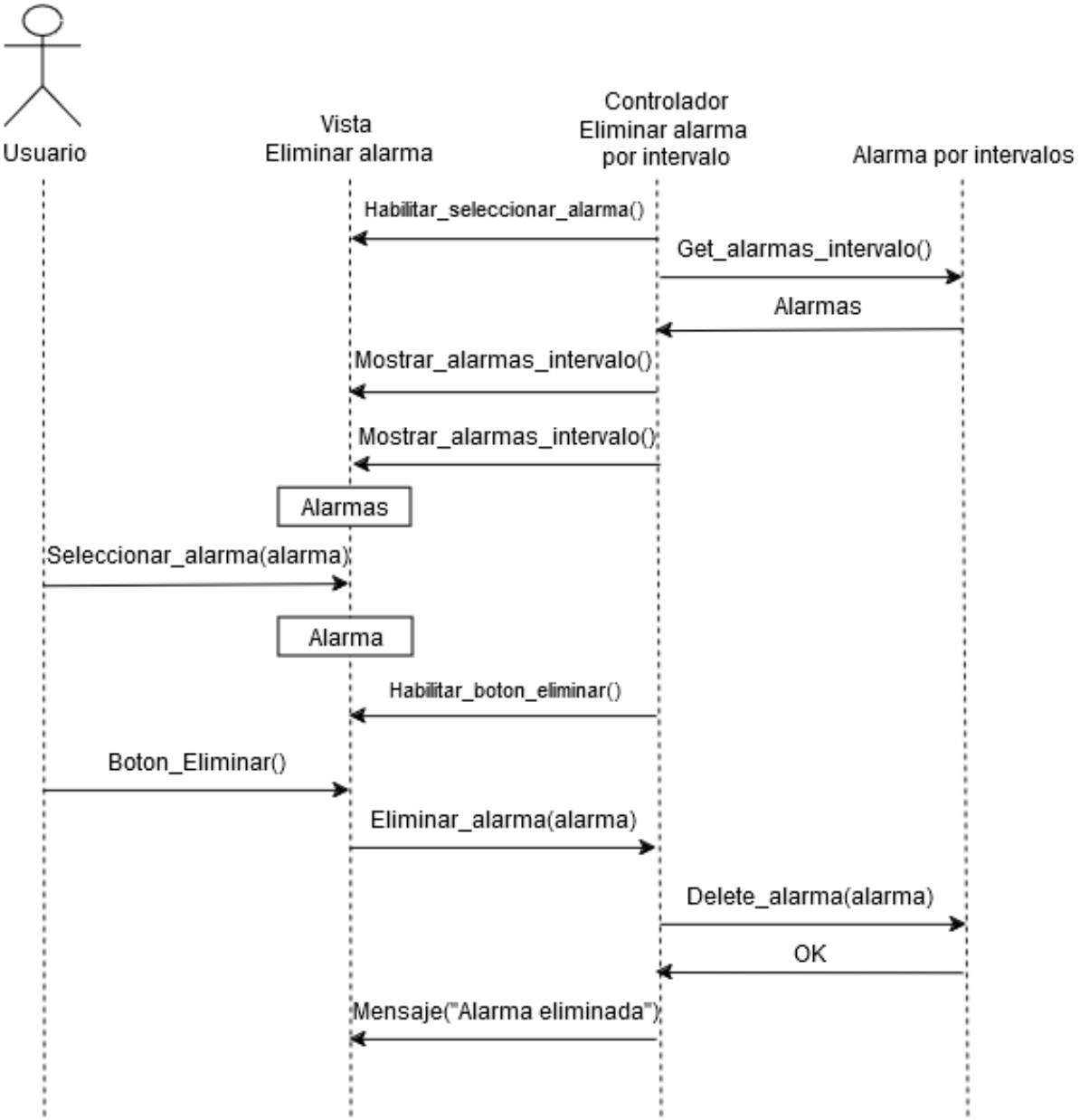


Figura 8: Traza Resolver Aritmética

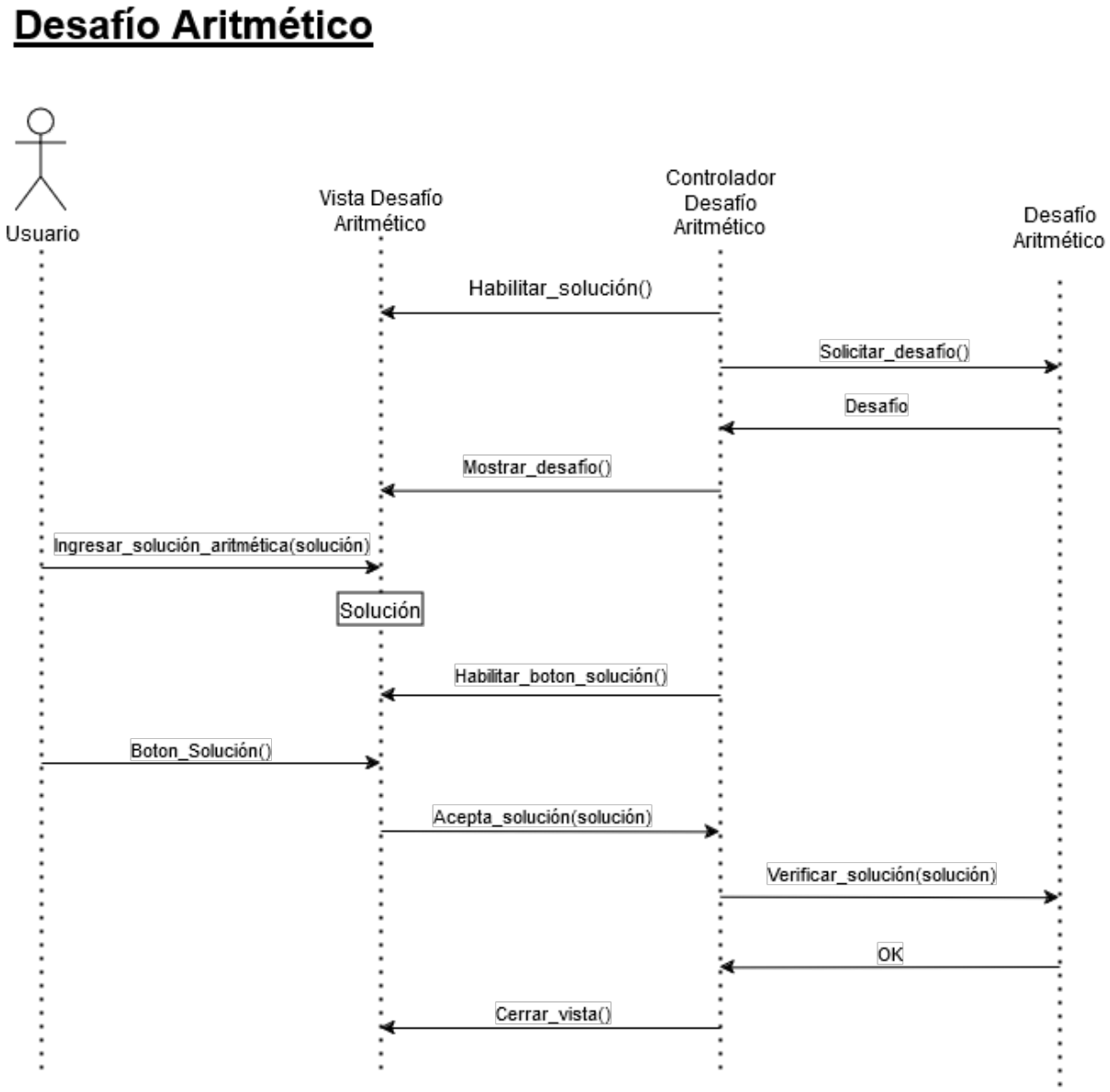


Figura 9: Traza Resolver Rompecabezas

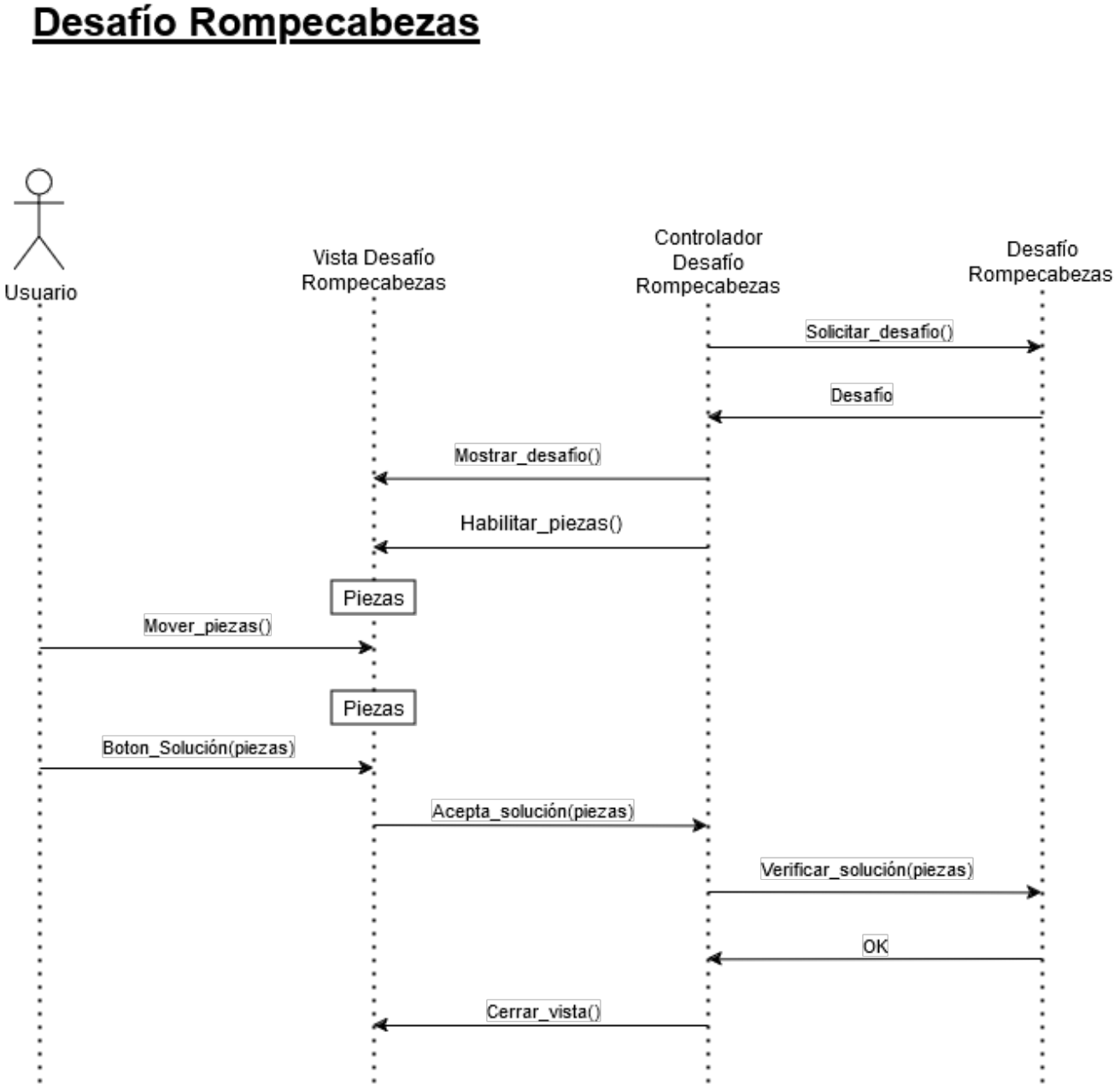


Figura 10: Traza Resolver Memorice

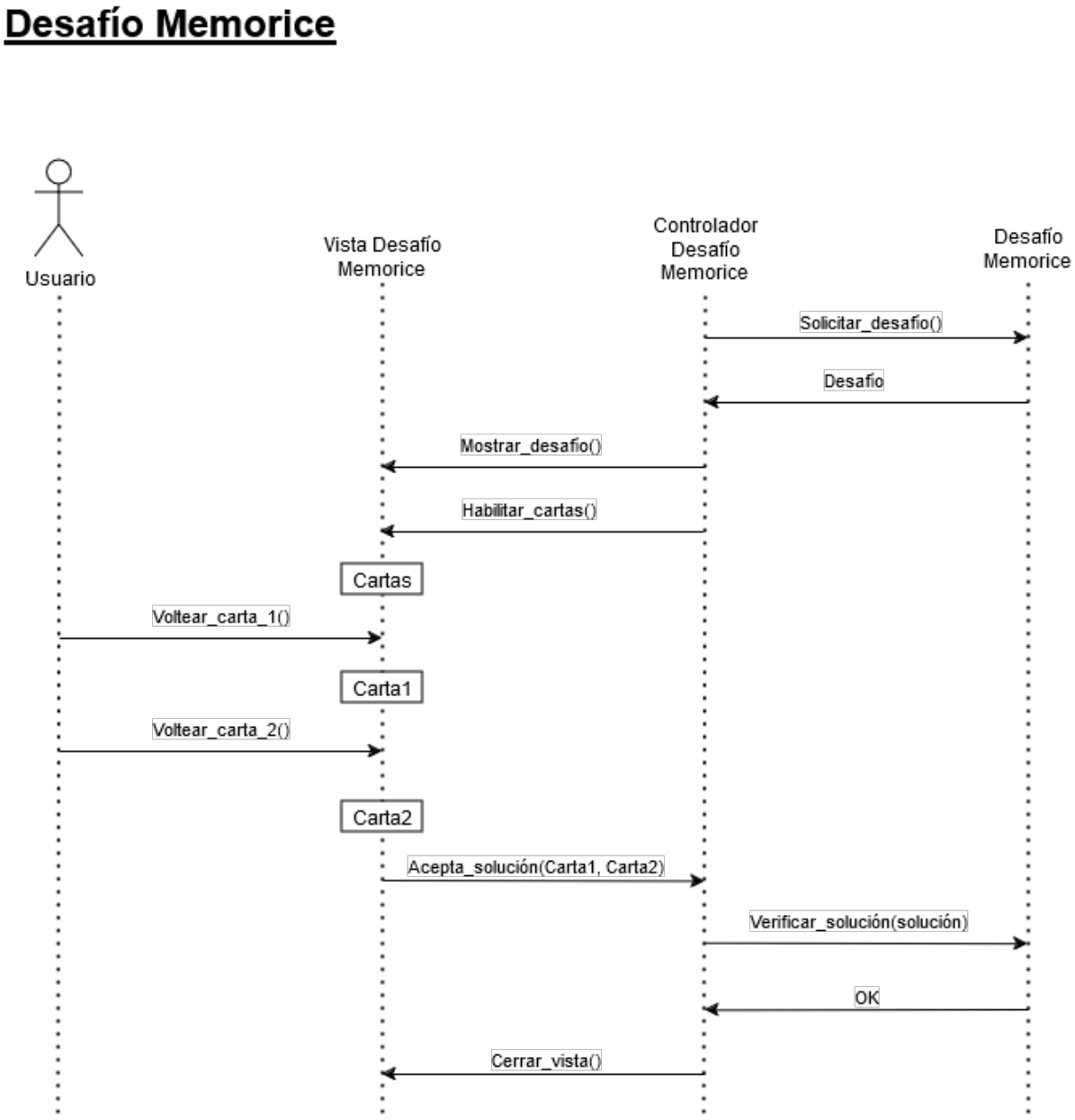
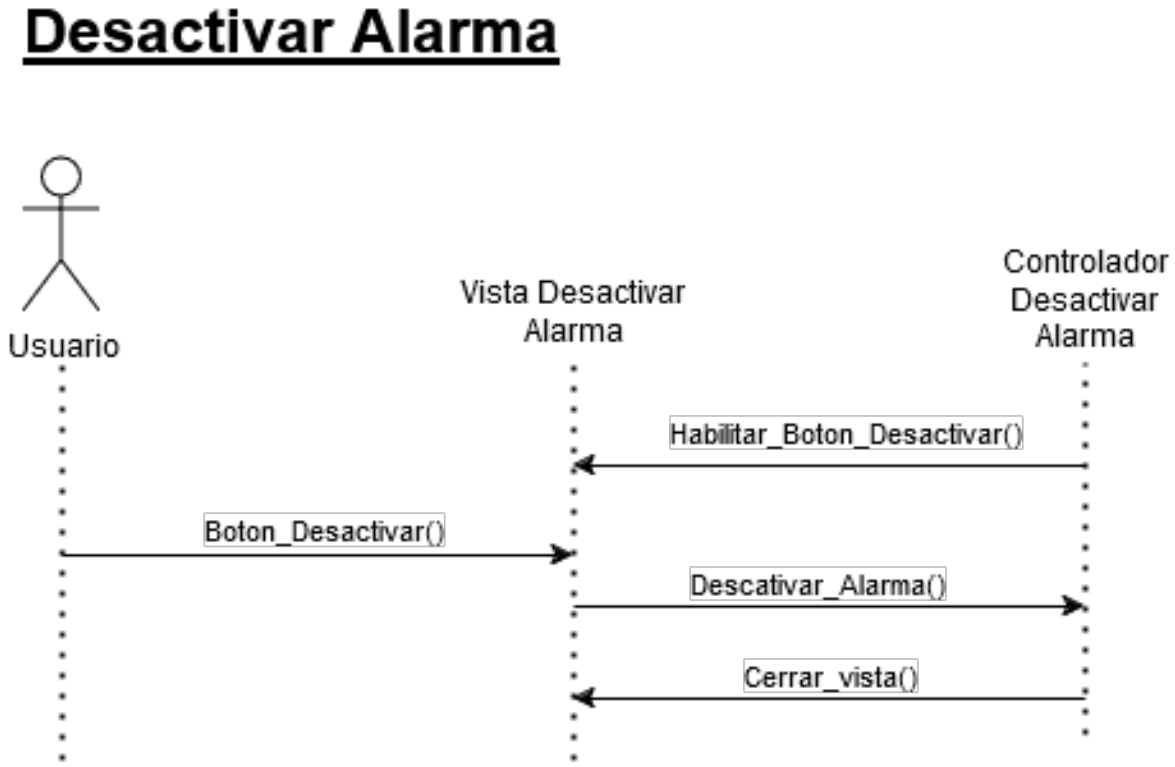


Figura 11: Traza Desactivar Alarma



3. Conclusiones

Durante el desarrollo de este avance, se detectaron falencias introducidas en el avance anterior, lo cual recalca la importancia de análisis presentado en este documento. El diagrama de diálogo resultó ser una forma útil de visualizar la interacción del usuario con el sistema, y la especificación de componenetes ayudó a visualizar de mejor forma dichos diálogos.

Con los cuatro artefactos creados durante este avance, en adición a los casos de uso obtenidos anteriormente, el proyecto avanza a la cuarta etapa, en la que se transformará lo obtenido en la etapa de análisis orientado a objetos a una forma que pueda ser implementada en un lenguaje de programación.