MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

ÍNDICE

1) Descripción del Problema

Desarrollar una aplicación de escritorio que permita resolver problemas de optimización no lineal usando métodos analíticos y numéricos. La aplicación debe aceptar funciones objetivo multivariables y restricciones (igualdades y desigualdades), detectar automáticamente el método más adecuado (o permitir selección manual) y proporcionar un **desarrollo paso a paso** que justifique y demuestre el cálculo (gradientes, Hessiana, sistema de Lagrange, condiciones KKT, iteraciones del gradiente, resolución QP, etc.). Debe soportar tanto soluciones simbólicas (cuando sea posible) como soluciones numéricas (cuando sea necesario), ofrecer mensajes claros en español y manejar errores de entrada.

2) Requerimientos

1. Resolución de problemas de optimización no lineal a través de cálculo diferencial.
2. Resolución de problemas de optimización no lineal a través de método de lagrange.
3. Resolución de problemas de optimización no lineal a través de Karl-Kuhn-Tucker.
4. Resolución de problemas de optimización no lineal a través de Algoritmos de PNL: Gradiente.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

5. Resolución de problemas de optimización no lineal a través de Algoritmos de PNL: Programación Cuadrática.
6. Sugerir el método adecuado para la resolución de los ejercicios planteados, diferenciado entre las opciones listadas anteriormente.

<table>
<tr><td colspan="2">DESCRIPCIÓN DEL PROBLEMA</td></tr>
<tr><td colspan="2">Desarrollar una aplicación de escritorio que permita resolver problemas de optimización no lineal usando métodos analíticos y numéricos. La aplicación debe aceptar funciones objetivo multivariables y restricciones (igualdades y desigualdades), detectar automáticamente el método más adecuado (o permitir selección manual) y proporcionar un desarrollo paso a paso que justifique y demuestre el cálculo (gradientes, Hessiana, sistema de Lagrange, condiciones KKT, iteraciones del gradiente, resolución QP, etc.). Debe soportar tanto soluciones simbólicas (cuando sea posible) como soluciones numéricas (cuando sea necesario), ofrecer mensajes claros en español y manejar errores de entrada.</td></tr>
<tr><td>REQUERIMIENTOS FUNCIONALES</td><td>REQUERIMIENTOS NO FUNCIONALES</td></tr>
<tr><td>Resolución por cada uno de los métodos listados (Cálculo Diferencial, Método de Lagrange, Condiciones de Karl-Kuhn-Tucker, Algoritmos de PNL: Gradiente y Programación Cuadrática).

Detección de la mejor solución para el problema, según los datos ingresados.

Exposición del paso a paso de la solución del ejercicio.</td><td>Exposición detallada de la solución paso a paso, de manera clara y concisa.

Interfaz gráfica amigable con el usuario, intuitiva y definida.</td></tr>
<tr><td colspan="2">OTROS ELEMENTOS A CONSIDERAR: REQUERIMIENTOS DE PROYECTO, RESTRICCIONES…</td></tr>
<tr><td colspan="2">Limitar enumeración de conjuntos activos (por ejemplo, hasta 3 desigualdades activas) para problemas con >10 desigualdades; recomendar método numérico en esos casos.</td></tr>
</table>

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

Soporte primario en 2–6 variables; problemas simbólicos con muchas variables pueden no resolverse simbólicamente.

Interfaz basada en tkinter para minimizar dependencias; si se requieren dependencias avanzadas considerar migrar a otros entornos de desarrollo.

Resolución por Cálculo Diferencial	Aplicar cálculo diferencial para encontrar puntos críticos de la función objetivo, calcular gradientes y evaluar la Hessiana permite identificar mínimos, máximos y puntos de silla de forma simbólica cuando sea posible. Al automatizar este procedimiento, el sistema ofrece soluciones exactas y comprobables; además reporta las condiciones necesarias y la clasificación por segunda derivada, lo que facilita validar si la solución hallada es un óptimo local sin necesidad de recurrir inmediatamente a métodos numéricos.
Método de Lagrange (igualdades)	Para problemas con restricciones de igualdad, el uso de multiplicadores de Lagrange construye un Lagrangiano que incorpora las restricciones y resuelve el sistema resultante de ecuaciones. El módulo implementará Lagrange para múltiples igualdades, devolverá las soluciones simbólicas cuando existan y explicará por qué y cómo se usaron los multiplicadores, mostrando las ecuaciones y la sustitución directa que verifica la factibilidad y la estacionariedad de las soluciones.
Condiciones de Karush–Kuhn–Tucker (desigualdades)	Cuando las restricciones incluyen desigualdades, el sistema aplicará las condiciones KKT: reformulación de desigualdades, complementariedad y no negatividad de multiplicadores. Se soportará una estrategia de conjuntos activos (active-set) para explorar subconjuntos candidatos de restricciones activas, verificar factibilidad y multiplicadores, y, cuando sea necesario, recomendar y usar un solucionador numérico. Esto asegura un tratamiento riguroso de problemas con fronteras y condiciones de desigualdad.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

Facilitar la gestión del invernadero para aquellos que no están familiarizados con la tarea.	Para aquellos que no tienen experiencia en el cuidado de un invernadero, la gestión de las condiciones ambientales puede resultar difícil. Con un sistema de control automatizado, la gestión del invernadero se vuelve mucho más fácil, ya que el sistema se encarga de monitorear y ajustar automáticamente las condiciones ambientales para garantizar el crecimiento saludable de las plantas.
Algoritmos de PNL Gradiente (iterativo)	Algoritmos de PNL — Gradiente (iterativo) Se incluirá un algoritmo de descenso por gradiente con búsqueda en línea tipo Armijo (backtracking) y parámetros ajustables (alpha, tolerancia, máximo de iteraciones, punto inicial). El método se usará como alternativa numérica cuando la resolución simbólica no sea viable o para problemas grandes; registrará el historial de iteraciones, norma del gradiente y reducción del valor objetivo, permitiendo auditar la convergencia y ajustar parámetros para mejorar desempeño.
Programación Cuadrática (QP)	Para funciones cuadráticas detectadas automáticamente, el sistema extraerá la matriz Q y el vector c, verificará convexidad (autovalores) y resolverá el sistema KKT para igualdades o la ecuación $Qx = -c$ sin restricciones, con fallback por pseudo-inversa si Q es singular. Esto proporciona soluciones cerradas eficientes para problemas QP y permite diagnosticar rápidamente si el problema es convexo y por tanto tiene mínimo global garantizado.
Detección automática de la mejor solución y motivo de selección	El sistema analizará la estructura de la función (grado, términos cuadráticos) y las restricciones (presencia y tipo: igualdades o desigualdades) para recomendar el método más apropiado, acompañando la recomendación con una razón clara y breve (por ejemplo: “no hay restricciones y la función es cuadrática → Programación Cuadrática”). Esta explicación ayuda al usuario a entender por qué se eligió ese método y cuándo forzar una alternativa manual.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

Exposición paso a paso de la solución	Cada procedimiento mostrará un desarrollo numerado y legible: qué se calculó en cada paso, las ecuaciones generadas, el sistema a resolver, las sustituciones realizadas y las verificaciones de factibilidad y segunda orden. Las expresiones se representarán en una sola línea usando ** para potencias, evitando formatos multilínea que confundan. Esto garantiza trazabilidad completa y facilita aprendizaje y auditoría.
Exposición detallada, clara y concisa	La documentación y salida del cálculo serán explicaciones cortas y directas que acompañen cada bloque matemático (por ejemplo: "Calculo gradiente ∇f, luego resuelvo $\nabla f=0$ para obtener puntos críticos; a continuación evalúo Hessiana para clasificar"). Se prioriza lenguaje en español neutro, sin jerga innecesaria, de modo que tanto usuarios técnicos como novatos puedan seguir el razonamiento.
Interfaz gráfica amigable e intuitiva	La GUI presentará campos claros para la función y restricciones, opciones para usar la sugerencia automática o seleccionar método manual, parámetros editables para algoritmos numéricos. y botones de ejemplo. El panel de resultados mostrará el paso a paso numerado, con controles para copiar/exportar texto. La interfaz minimizará errores de entrada con validaciones y mensajes de error descriptivos en español.

3) Historias de Usuario

Nombre	**Req. 1 – Resolución Integral de Optimización.**
Resumen	Optimizar funciones objetivo con o sin restricciones mediante un sistema automático capaz de analizar la estructura del problema y seleccionar el método matemático más adecuado. El sistema compara las características de la función, el tipo de restricciones y el grado de la expresión, y con esta información determina si debe emplear cálculo diferencial, multiplicadores de Lagrange, condiciones KKT, descenso por gradiente o programación

	cuadrática. Una vez identificado el método apropiado, genera la solución junto con el desarrollo paso a paso que explica cada operación realizada. Esto permite resolver de forma exacta o aproximada problemas de optimización no lineales bajo cualquier combinación de igualdades, desigualdades o ausencia de restricciones.
Entradas	
Función objetivo	
Variables detectadas	
Restricciones de igualdad	
Restricciones de desigualdad	
Estructura cuadrática (si existe)	
Parámetros numéricos (iteraciones, tolerancia, punto inicial)	
Salidas	
Cálculo Diferencial: puntos críticos y clasificación mediante Hessiana	
Lagrange: solución factible con multiplicadores y verificación de igualdades	
KKT: solución candidata con conjunto activo y evaluación de multiplicadores	
Gradiente: aproximación numérica al mínimo con historial iterativo	
Programación Cuadrática: solución matricial y validación de convexidad	
Paso a paso detallado del procedimiento aplicado	

Nombre	**Req. 2 – Selección automática**
Resumen	Analizar la función y restricciones ingresadas para determinar el método más adecuado, identificando el tipo de problema (sin restricciones, con igualdades, con desigualdades o cuadrático). El sistema compara la estructura matemática con los criterios establecidos y genera una recomendación clara para guiar al usuario hacia el método más eficiente.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

Entradas
Función objetivo
Restricciones ingresadas
Restricciones ingresadas
Número de variables
Salidas
Método recomendado
Justificación de la elección
Clasificación del problema

Nombre	**Req. 3 – Paso a paso.**
Resumen	Mostrar de forma clara, ordenada y comprensible cada una de las operaciones realizadas por el sistema: derivadas, gradientes, sistemas, sustituciones, comprobaciones, iteraciones y resultados. Esto compara cada suboperación con criterios matemáticos reconocidos para que el usuario comprenda cómo se obtiene la solución final.
Entradas	
Datos del problema	
Operaciones simbólicas	
Iteraciones numéricas	
Salidas	
Desarrollo completo numerado	
Explicación breve de cada paso	
Interpretación final del resultado	

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

Nombre	**Req. 4 – Interfaz gráfica intuitiva.**
Resumen	Proporcionar una interfaz fácil de usar, clara y ordenada, que permite introducir funciones, restricciones, parámetros numéricos y elegir el método, ya sea recomendado o manual. El sistema compara continuamente la entrada del usuario con reglas de validación para evitar errores y garantizar una interacción fluida.
Entradas	
Funciones ingresadas	
Restricciones	
Parámetros del método	
Acciones del usuario	
Salidas	
Visualización del resultado	
Panel de pasos	
Advertencias y validaciones	

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

4) Diagramas de Casos de Uso

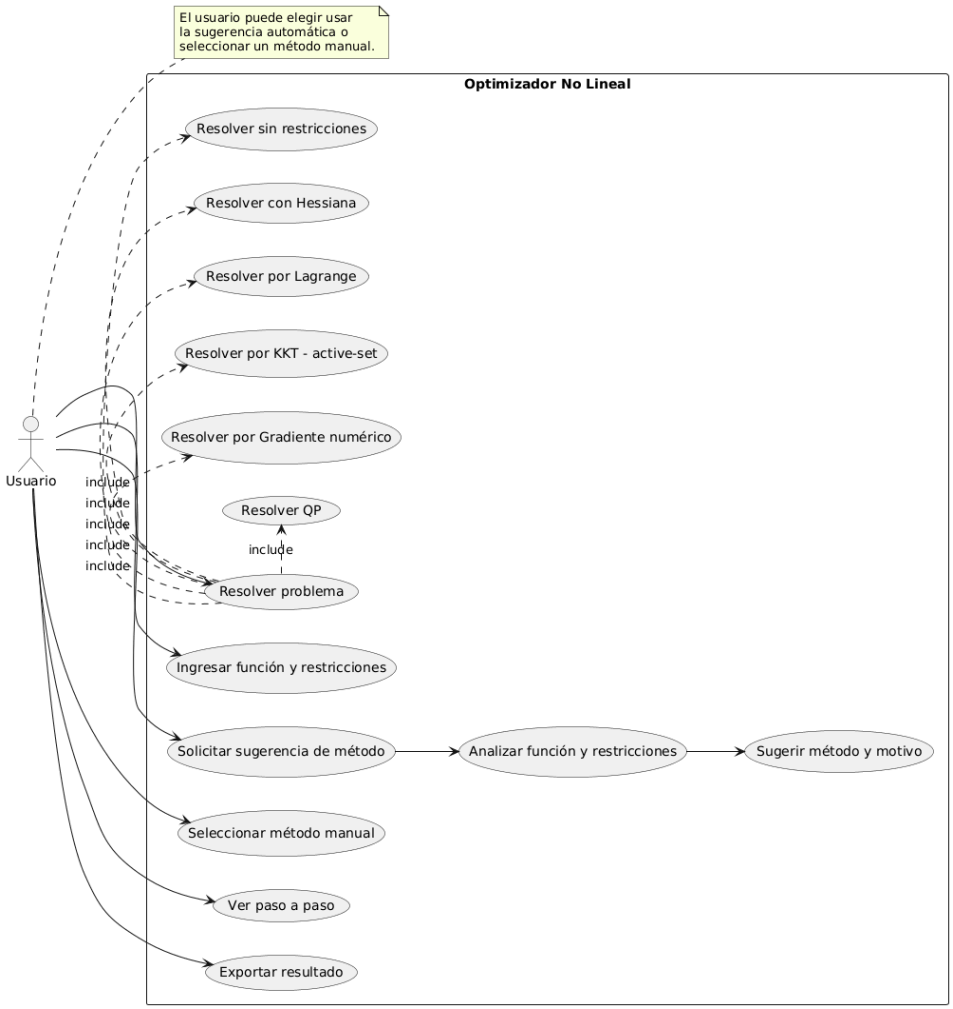

Figura 1. Diagrama Casos de Uso. Fuente: Elaboración propia con uso de PLANTUML

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

5) Arquitectura del Software

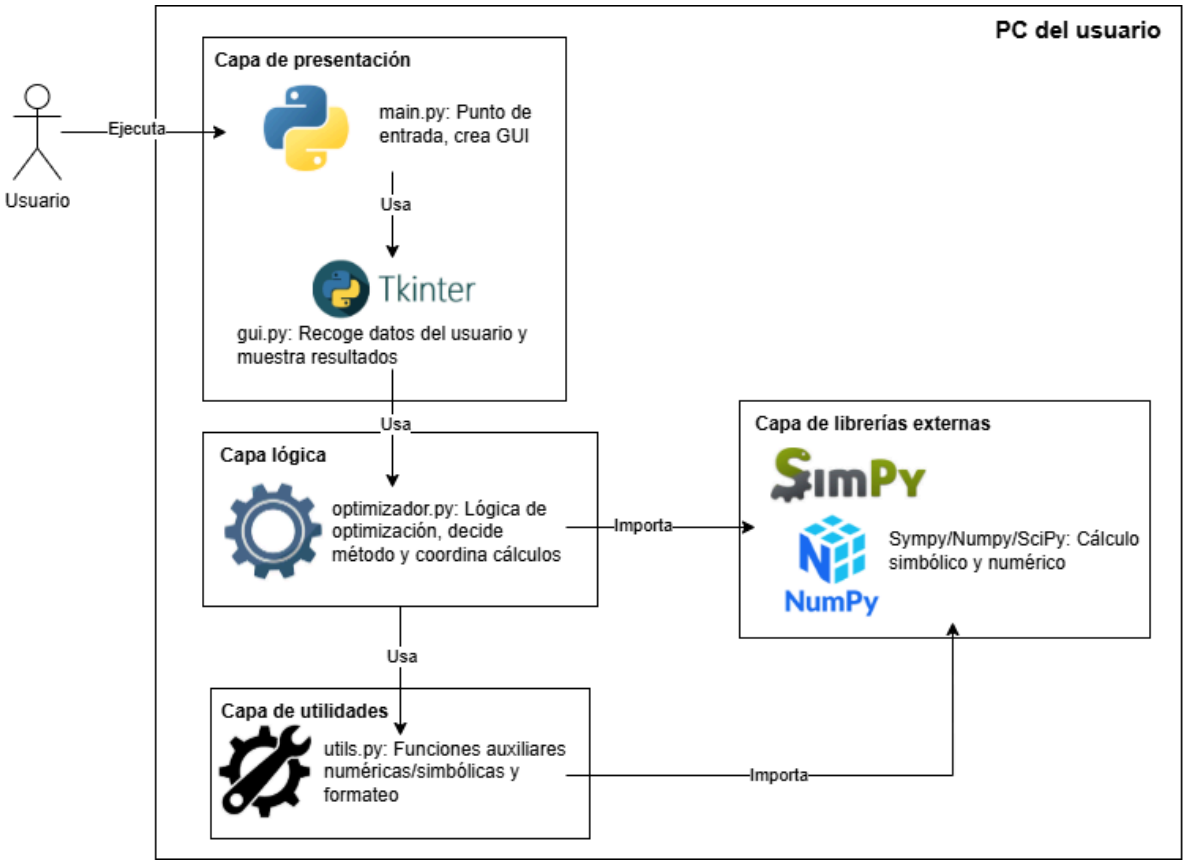

Figura 2. Diagrama de arquitectura . Fuente: Elaboración propia.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

6) Diagrama de Clases

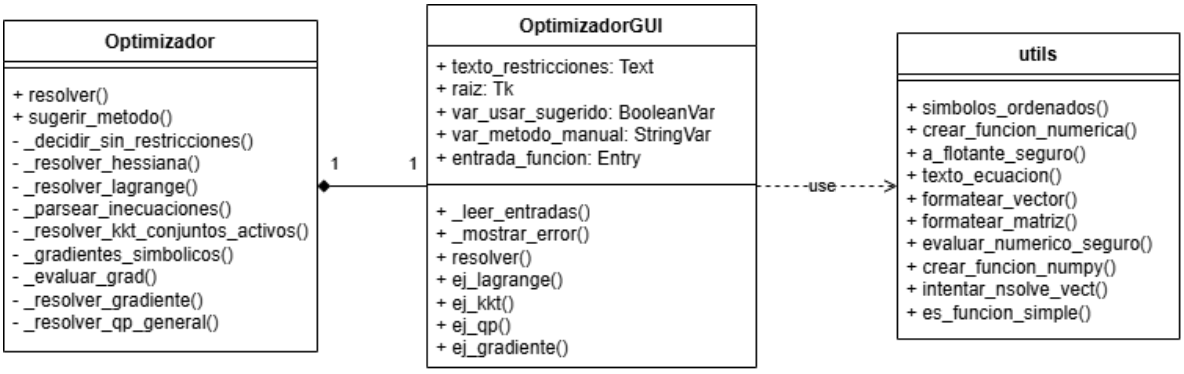

Figura 3. Diagrama Clases. Fuente: Elaboración propia.

MANUAL TÉCNICO DE PROGRAMA PARA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN NO LINEAL

7) Diagramas de Secuencia

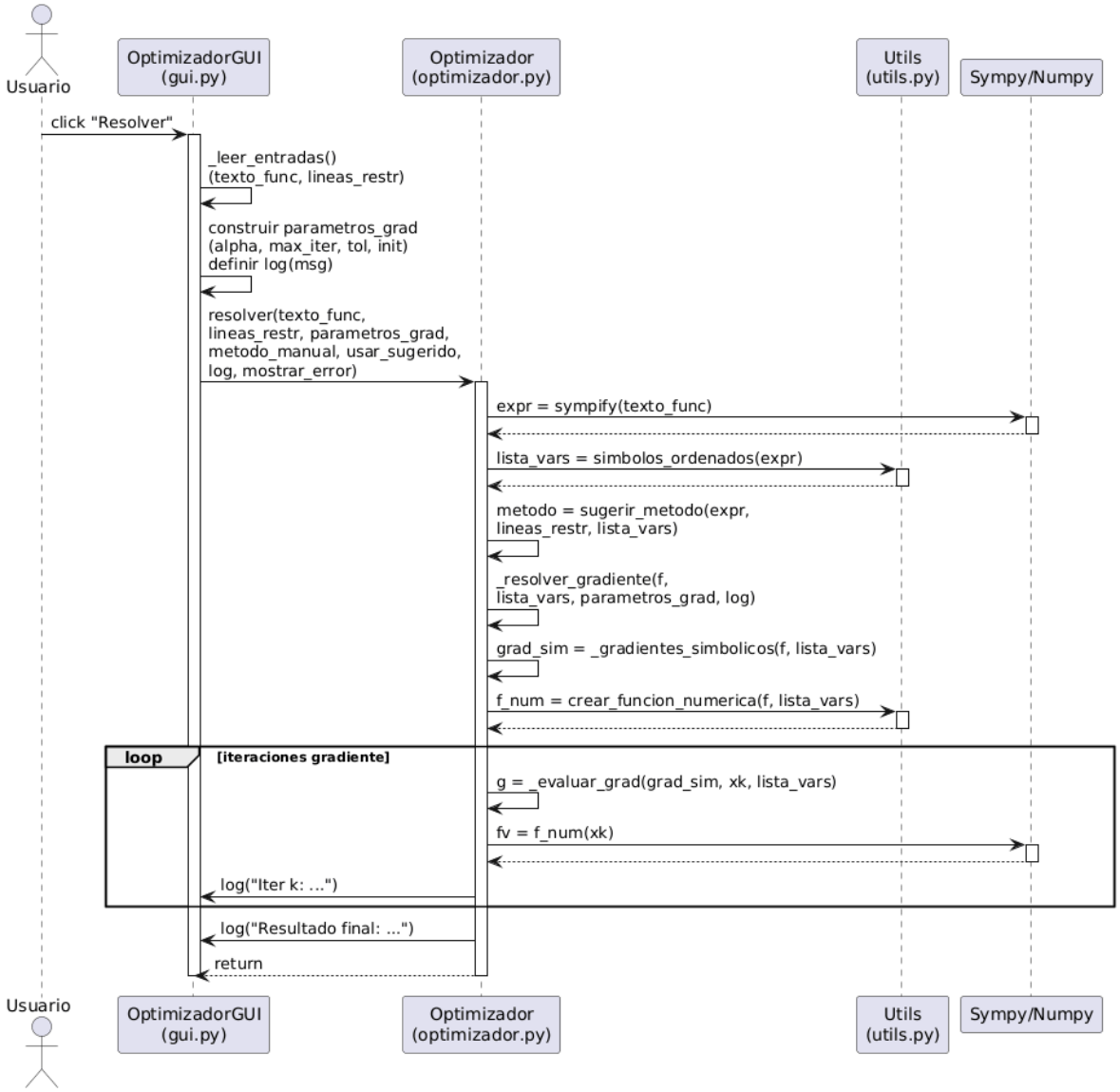

Figura 4. Diagrama de secuencia. Fuente: Elaboración propia con uso de PLANTUML.