



SUITE DE OPTIMIZACIÓN LINEAL

Johan Sebastián Forero Agudelo¹, Sander Farid Sierra Reyes², Miguel Angel Rincon Morales³

Universidad de los Llanos

Villavicencio, Meta

¹jsforero.agudelo@unillanos.edu.co 160005011

²marincon.morales@unillanos.edu.co 160005033

³sfsierra@unillanos.edu.co 160005038

Abstract— Este documento presenta el desarrollo de un software didáctico para la resolución de problemas de optimización lineal mediante Método Gráfico, Método Simplex (básico, Gran M y Dual) y análisis de sensibilidad. La herramienta permite modelar problemas a partir de descripciones en lenguaje natural, construir y editar modelos matemáticos, verificar su coherencia y visualizar paso a paso los procedimientos de solución. El sistema integra visualización de iteraciones, interpretación de resultados y evaluación post-óptima para apoyar la toma de decisiones. Este manual describe su funcionamiento técnico, arquitectura y métodos implementados.

estructura general, un modelo de PL plantea una función lineal sujeta a un sistema de igualdades o desigualdades y a restricciones de no negatividad sobre las variables [1], [2]. Su versatilidad permite modelar problemas propios de la ingeniería, la logística, la economía y la administración, en los que se busca asignar eficientemente recursos limitados [3].

La importancia de la PL radica en que facilita la representación de situaciones reales como planificación de producción, distribución, transporte, mezcla de materiales, asignación de personal o presupuestos. Su formulación estructurada permite transformar estos problemas en modelos resolubles mediante algoritmos especializados, lo que convierte a la PL en una herramienta fundamental para la toma de decisiones en la industria y el sector público [1], [4].

I. INTRODUCCIÓN

La Suite de Optimización Lineal es una aplicación web diseñada para automatizar la formulación, análisis y resolución de problemas de programación lineal mediante una arquitectura que combina procesamiento con inteligencia artificial y herramientas matemáticas simbólicas. El sistema permite interpretar problemas ingresados por el usuario en texto o imagen, generar modelos matemáticos validados y ofrecer distintas representaciones formales del mismo.

El software integra un flujo completo que abarca extracción automática de variables y restricciones mediante Groq, validación simbólica con SymPy, generación de formas canónica, estándar, matricial y dual, y resolución mediante los métodos Simplex, Simplex Dual y Gran M utilizando PuLP. Además, incorpora análisis de sensibilidad para evaluar precios sombra, costos reducidos y variaciones críticas del modelo.

La interfaz web ofrece visualización detallada de tableaux, y gráficos, permitiendo al usuario seguir paso a paso el proceso de resolución y comprender cómo evoluciona la solución. Este manual describe la estructura técnica del sistema, sus componentes principales, su funcionamiento interno y el flujo operativo de la herramienta.

II. MARCO TEÓRICO

La optimización lineal, o Programación Lineal (PL), es una metodología matemática utilizada para determinar la mejor estrategia posible—generalmente maximizar ganancias o minimizar costos—cuando tanto la función objetivo como las restricciones pueden expresarse mediante relaciones lineales. En su

A. MÉTODOS DE SOLUCIÓN

Para resolver problemas de programación lineal se emplean distintos métodos clásicos. El **método gráfico** es el más sencillo y se aplica exclusivamente cuando el modelo tiene dos variables de decisión. Consiste en representar cada restricción como una recta en el plano cartesiano, identificar la región factible generada por la intersección de dichas restricciones y evaluar la función objetivo en cada vértice de esa región. Su propósito es principalmente didáctico, ya que permite visualizar conceptos como factibilidad, optimalidad y restricciones activas, aunque no es aplicable a problemas de mayor dimensión [3].

El **Método Simplex**, desarrollado por George Dantzig, es el algoritmo más utilizado para resolver modelos de programación lineal de tamaño medio y grande. Este método parte de una solución básica factible y avanza de vértice en vértice dentro de la región factible mejorando progresivamente el valor de la función objetivo mediante operaciones de pivoteo. Su estructura tabular permite seguir cada paso del proceso, lo que facilita su interpretación y lo convierte en un método eficiente, sistemático y ampliamente empleado en entornos computacionales [1], [2].

El **Método de la Gran M** es una extensión del Simplex utilizada cuando el problema incluye restricciones del tipo “ \geq ” o igualdades que impiden encontrar de inmediato una solución básica factible. Para manejar este tipo de estructuras, se introducen variables artificiales acompañadas de un coeficiente muy grande M en la función objetivo, de manera que el algoritmo penaliza su presencia y las expulsa de la base en el proceso de optimización. Esto



garantiza que el método pueda iniciarse adecuadamente incluso en modelos con configuraciones no estándar [2].

Por otro lado, el **Método Simplex Dual** opera desde una base dual factible en lugar de una primal. Es especialmente útil cuando la solución inicial cumple las condiciones del dual, o cuando cambios en las restricciones vuelven más conveniente operar desde la perspectiva dual. Su proceso también se basa en el pivoteo, pero busca restablecer factibilidad primal mientras mantiene factibilidad dual. Esto lo convierte en una alternativa potente en modelos donde las restricciones o parámetros cambian frecuentemente, como ocurre en contextos de reoptimización [4].

B. DUALIDAD Y ANÁLISIS DE SENSIBILIDAD

La dualidad es un concepto fundamental en programación lineal. Para cada problema primal existe un problema dual cuyas variables representan valores sombra o interpretaciones económicas asociadas a los recursos. El análisis del dual permite comprender cómo afectan los cambios marginales en los recursos a la solución óptima, proporcionando una perspectiva adicional sobre la estructura del problema [1], [5].

El análisis de sensibilidad, también llamado análisis post-óptimo, evalúa cómo variaciones en los coeficientes de la función objetivo, los recursos disponibles o los coeficientes de las restricciones afectan la solución óptima. Este análisis es esencial cuando los datos del modelo pueden presentar incertidumbre o cambios en el tiempo, pues permite estimar rangos válidos, identificar restricciones activas o críticas y medir la estabilidad del resultado obtenido [2], [5]. Su aplicación es relevante para la toma de decisiones, especialmente en entornos donde los parámetros no son estáticos.

C. RELEVANCIA COMPUTACIONAL

Con el avance de herramientas informáticas y librerías matemáticas, los métodos de optimización lineal pueden implementarse de manera automatizada y eficiente. Los sistemas modernos permiten formular modelos, transformarlos en representaciones equivalentes (canónica, estándar, matricial, dual), resolverlos mediante Simplex o variantes, y generar análisis de sensibilidad sin requerir cálculos manuales extensos. Esto ha permitido que la optimización lineal sea accesible tanto para estudiantes como para profesionales, integrándose en aplicaciones prácticas de ingeniería, logística y gestión de operaciones [3], [4].

III. HISTORIAS DE USUARIO

Las historias de usuario recopiladas para este proyecto representan la base funcional de la Suite de Optimización Lineal, ya que describen de manera clara cómo un usuario interactúa con el sistema en cada etapa del proceso: desde la introducción del problema (en texto o imagen), pasando por la construcción automática del modelo, la verificación matemática, la ejecución de

métodos como Simplex, Gran M o Simplex Dual, y finalizando con la interpretación de la solución y el análisis de sensibilidad. Cada historia refleja necesidades reales asociadas al uso de un software educativo y analítico, donde el objetivo no es solo resolver el modelo, sino también explicar el procedimiento de forma didáctica, facilitar la edición de parámetros y ofrecer soporte para la toma de decisiones. Por tanto, este conjunto de HU actúa como la guía funcional que determina los módulos del sistema, sus flujos de información y las capacidades de interacción previstas para el usuario.

Código: HU-01	Rol: Usuario
Título: Definición del Problema (NLP e Imágenes)	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero describir un problema de optimización en lenguaje natural o cargar una imagen del enunciado, para que el sistema comprenda el objetivo, variables y restricciones y prepare la base del modelo.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Entrada Multimodal: El sistema debe permitir entrada de texto libre y carga de imágenes (OCR/Visión Computarizada). • Procesamiento: Debe identificar automáticamente el objetivo (maximizar/minimizar), las decisiones (variables) y las limitaciones (restricciones). • Confirmación: El sistema debe presentar un resumen de lo entendido para validación del usuario antes de modelar. 	

Código: HU-02	Rol: Usuario
Título: Modelado Matemático y Edición	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero que el sistema genere el modelo matemático automáticamente, pero permitiéndome definir, modificar o ajustar variables y restricciones manualmente, para asegurar que la formulación sea exacta y completa.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Generación Automática: Transformar la descripción del problema en una sintaxis matemática formal (Función Objetivo, S.A.). • Edición Manual: Permitir al usuario añadir/eliminar variables y restricciones directamente. • Corrección: El usuario debe poder editar coeficientes o lógica si la interpretación automática tuvo errores. 	



Código: HU-03	Rol: Usuario
Título: Validación y Coherencia del Modelo	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero que el sistema verifique la coherencia lógica del modelo (dimensiones, tipos de variables) para asegurar que es resoluble antes de intentar ejecutar un método.	
Criterios de aceptación: <ul style="list-style-type: none"> • Chequeo de Tipos: Validar que las variables correspondan a su tipo (enteras, continuas, binarias). • Consistencia: Verificar compatibilidad dimensional en las ecuaciones. • Alertas: Notificar errores de sintaxis o in-factibilidad obvia antes del cálculo. 	

Código: HU-04	Rol: Usuario
Título: Visualización de Representaciones	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero ver el modelo en sus distintas formas (canónica, estándar, dual), para comprender mejor su estructura matemática y preparar la resolución.	
Criterios de aceptación: <ul style="list-style-type: none"> • Vistas Múltiples: Pestañas o secciones para ver el modelo "Original", "Estándar" (con holguras) y "Dual". • Formato: Utilizar notación matemática clara (LaTeX) para las ecuaciones. • Sincronización: Si se edita el modelo original, las otras vistas deben actualizarse automáticamente. 	

Código: HU-05	Rol: Usuario
Título: Selección Asistida de Métodos de Resolución	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero que el sistema analice el modelo, sugiera métodos adecuados (Simplex, Gráfico, etc.) y explique por qué otros no aplican, para elegir la técnica correcta sin errores.	

Criterios de aceptación: <ul style="list-style-type: none"> • Análisis de Elegibilidad: El sistema debe evaluar qué métodos aplican (ej: Gráfico solo para 2 variables). • Recomendación: Sugerir el método más eficiente para el tipo de problema. • Explicación Educativa: Mostrar mensajes tipo tooltip explicando "Por qué este método no está disponible" (ej: "No linealidad detectada"). • Selección: Permitir al usuario confirmar o cambiar el método sugerido. 	
---	--

Código: HU-06	Rol: Usuario
Título: Resolución Didáctica Paso a Paso	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero ejecutar el método escogido y visualizar las iteraciones o transformaciones de forma didáctica hasta llegar a la solución óptima, para entender el proceso lógico detrás del resultado.	
Criterios de aceptación: <ul style="list-style-type: none"> • Ejecución: Calcular los valores óptimos de las variables y la función objetivo. • Paso a Paso: Mostrar cada iteración (ej: Tablas Simplex, pivoteos, movimientos en gráfica). • Resultados Intermedios: Mostrar valores de variables en cada paso. • Resumen Final: Presentar claramente la solución óptima y el valor de Z (Función Objetivo). 	

Código: HU-07	Rol: Usuario
Título: Análisis de Sensibilidad y Escenarios	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero ajustar parámetros del modelo (costos, RHS) y re-calcular para analizar cómo cambios en el entorno afectan la solución óptima y visualizar las variaciones.	
Criterios de aceptación: <ul style="list-style-type: none"> • Modificación de Parámetros: Interfaz para cambiar coeficientes (C_j) o disponibilidades (b_i) sobre el modelo resuelto. • Re-optimización: Calcular la nueva solución partiendo de la base actual (sin resolver desde cero si es posible). • Métricas de Sensibilidad: Mostrar precios sombra (dual prices), costos reducidos y rangos de holgura. • Comparativa: Tabla o gráfico que muestre "Antes vs. Después" del cambio. 	



Código: HU-08	Rol: Usuario
Título: Interpretación Inteligente y Recomendaciones	
Prioridad: Alta	Tipo: Funcional
Descripción: Como usuario, quiero que la IA interprete la solución, detecte riesgos o sesgos y me dé una recomendación final fundamentada, para tomar decisiones robustas más allá del número matemático.	
Criterios de aceptación: <ul style="list-style-type: none"> • Resumen en Lenguaje Natural: Explicación textual de qué significa la solución (ej: "Debes producir 50 unidades de X"). • Detección de Sesgos: Alertas sobre datos que parezcan anómalos o poco realistas. • Análisis de Riesgo: Evaluar qué tan frágil es la solución ante cambios pequeños. • Recomendación Estratégica: Un párrafo final de conclusión para la toma de decisiones gerenciales. 	

IV. DIAGRAMA DE ARQUITECTURA

El diagrama de arquitectura de la Suite de Optimización Lineal expone la estructura interna del sistema, donde se integran tecnologías como Django, SymPy, PuLP, Groq API, y un frontend interactivo basado en HTML/JS. Esta arquitectura refleja el recorrido técnico que sigue un problema: primero es recibido por la API, luego procesado por el módulo de Inteligencia Artificial para extraer variables, restricciones y función objetivo, validado simbólicamente para garantizar coherencia matemática, transformado a diferentes representaciones (canónica, estándar, dual, matricial) y finalmente resuelto mediante los métodos disponibles. La arquitectura también ilustra cómo el frontend consume los resultados para mostrar tableaux, gráficas, iteraciones y análisis de sensibilidad. Este diagrama permite comprender cómo se coordinan cada uno de estos módulos para ofrecer una experiencia fluida, transparente y explicativa al usuario.

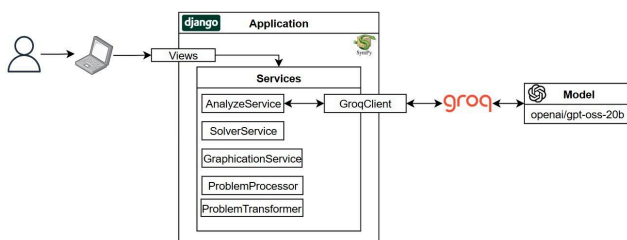


Figura 1. Diagrama de Arquitectura

V. RESULTADOS

La ejecución del sistema permitió validar el comportamiento de cada uno de los métodos implementados, verificando que los cálculos, transformaciones y evaluaciones realizadas siguen el flujo esperado. Durante las pruebas se analizó el proceso completo: desde la definición del problema inicial, la construcción del modelo canónico y la generación de los modelos derivados, hasta la ejecución individual de cada método del sistema. Esto permitió confirmar la estabilidad, precisión y correcta interacción entre módulos.

El sistema demostró consistencia tanto en los valores calculados como en la lógica aplicada en cada etapa. Además, se verificó que los algoritmos responden adecuadamente ante variaciones en los parámetros y producen resultados reproducibles en ejecuciones consecutivas. La estructura del flujo se mantuvo estable incluso bajo condiciones límite, indicando una correcta implementación.

• PROBLEMA EVALUADO

A continuación se presenta el problema ingresado para la ejecución de pruebas. Este constituye el punto de partida para las transformaciones posteriores y permite verificar la exactitud de cada método respecto al caso base.

Ingresar Problema

Descripción (Texto)

Unos grandes almacenes desean liquidar 200 camisas y 100 pantalones de la temporada anterior, para ello lanzan dos ofertas: A y B.
La oferta A consiste en un lote de una camisa y un pantalón que se venden a \$ 30 dólares.
La oferta B consiste en un lote de tres camisas y un pantalón que se vende a \$ 50 dólares.
No se desea ofrecer menos de 20 lotes de la oferta A y menos de 10 de la B.
¿Cuántos lotes ha de vender de cada tipo para maximizar la ganancia?

Analizar Problema

Figura 2. Problema ingresado



• MODELO CANÓNICO GENERADO

El sistema convierte automáticamente el problema ingresado a su modelo canónico, asegurando que todas las ecuaciones cumplan la estructura formal requerida por los métodos posteriores. Se verificó que la eliminación de desigualdades, incorporación de variables de holgura o artificiales, y normalización de la función objetivo se realizaron correctamente.

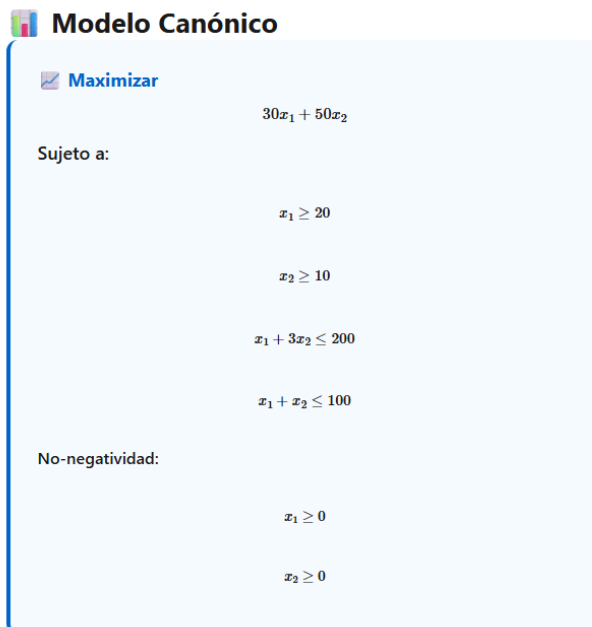


Figura 3. Modelo canónico.

• MODELOS GENERADOS PARA CADA MÉTODO

Cada método implementado opera sobre un modelo específico derivado del modelo canónico. Se comprobó que los modelos generados se ajustan a las reglas de transformación definidas y conservan la coherencia matemática del problema.

• Resultados Método Gráfico:



Figuras 4, 5. Procedimiento del Método Simplex

• Modelo y Resultados Método Simplex:



Solución
MÉTODO USADO
SIMPLEX

VALOR ÓPTIMO
4000

Ecuaciones con Variables de Holgura

$$\begin{aligned} -x_1 + s_1 &= -20 \\ -x_2 + s_2 &= -10 \\ x_1 + 3x_2 + s_3 &= 200 \\ x_1 + x_2 + s_4 &= 100 \end{aligned}$$

Valores de Variables

x_1	x_2
50	50

Procedimiento Aplicado

Método Simplex: 2 iteraciones hasta optimalidad

Iteraciones del Método (2)

Iteración 1

Iteración 2

Operación Pivote

Variable Entrante: **x_1**

Variable Saliente: **s_4**

Elemento Pivote: **0.6667**

Fila Pivote: 3, Columna Pivote: 0

Tabla Antes del Pivote:

Base	x_1	x_2	s_1	s_2	s_3	s_4	RHS
s_1	-1	0	1	0	0	0	-20
s_2	0.3333	0	0	1	0.3333	0	56.6667
x_2	0.3333	1	0	0	0.3333	0	66.6667
s_4	0.6667	0	0	0	-0.3333	1	33.3333
Z	-13.3333	0	0	0	16.6667	0	3333.3333

Tabla Después del Pivote:

Base	x_1	x_2	s_1	s_2	s_3	s_4	RHS
s_1	0	0	1	0	-0.5	1.5	30
s_2	0	0	0	1	0.5	-0.5	40
x_2	0	1	0	0	0.5	-0.5	50
x_1	1	0	0	0	-0.5	1.5	50
Z	0	0	0	0	10	20	4000

Figuras 6, 7. Procedimiento del Método Simplex

Solución
MÉTODO USADO
GRAN M

VALOR ÓPTIMO
4000

Ecuaciones con Variables de Holgura y Artificiales

$$\begin{aligned} x_1 - s_1 + a_1 &= 20 \\ x_2 - s_2 + a_2 &= 10 \\ x_1 + 3x_2 + s_3 &= 200 \\ x_1 + x_2 + s_4 &= 100 \end{aligned}$$

Valores de Variables

x_1	x_2
50	50

Iteración 4

Operación Pivote

Variable Entrante: **s_1**

Variable Saliente: **s_4**

Elemento Pivote: **0.6667**

Fila Pivote: 3, Columna Pivote: 2

Tabla Antes del Pivote:

Base	x_1	x_2	s_1	s_2	s_3	s_4	a_1	a_2	RHS
x_1	1	0	-1	0	0	0	1	0	20
x_2	0	1	0.3333	0	0.3333	0	-0.3333	0	60
s_2	0	0	0.3333	1	0.3333	0	-0.3333	-1	50
s_4	0	0	0.6667	0	-0.3333	1	-0.6667	0	20
Z	0	0	-13.3333	0	16.6667	0	13.3333	0	3600

Tabla Después del Pivote:

Base	x_1	x_2	s_1	s_2	s_3	s_4	a_1	a_2	RHS
x_1	1	0	0	0	-0.5	1.5	0	0	50
x_2	0	1	0	0	0.5	-0.5	0	0	50
s_2	0	0	1	0.5	-0.5	0	-1	0	40
s_1	0	0	1	0	-0.5	1.5	-1	0	30
Z	0	0	0	0	10	20	0	0	4000

Figuras 8, 9. Procedimiento del Método Gran M

- **Modelo y Resultados Método Gran M:**



• **Modelo y Resultados Método Dual Simplex:**

Solución
MÉTODO USADO:
SIMPLEX DUAL

VALOR ÓPTIMO
180

Ecuaciones con Variables de Holgura

$$\begin{aligned} -x_1 - x_2 + s_1 &= -40 \\ -x_1 + s_2 &= -10 \end{aligned}$$

Valores de Variables

x_1	x_2
10	30

Iteración 2

Operación Pivote
Variable Entrante: **x_1**
Variable Saliente: **s_2**
Elemento Pivote: **-1**
Fila Pivote: 1, Columna Pivote: 0

Tabla Antes del Pivote:

Base	x_1	x_2	s_1	s_2	RHS
x_2	1	1	-1	0	40
s_2	-1	0	0	1	-10
Z	-2	0	-4	0	160

Tabla Después del Pivote:

Base	x_1	x_2	s_1	s_2	RHS
x_2	0	1	-1	1	30
x_1	1	0	0	-1	10
Z	0	0	-4	-2	180

Figuras 10, 11. Procedimiento del Método Dual

Análisis de Sensibilidad Post-Óptimo
Método: Simplex Dual

Valor óptimo: Z = 180 (mínimo)

Recurso más valioso: Restricción 1 con precio sombra $\pi = 4$. Aumentar este recurso tendría el mayor impacto en Z.

Parámetro más sensible: El coeficiente de x_2 . Pequeños cambios podrían alterar la solución óptima.

¿Qué es el Análisis de Sensibilidad? (Click para expandir)

Rangos de Coeficientes Rangos RHS **Precios Sombra** Costos Reducidos

Precios Sombra (Valores Duales)

¿Qué es un precio sombra?
El precio sombra π_j de una restricción representa el cambio en el valor óptimo Z por cada unidad adicional del recurso b_j .

Interpretación económica:

- Restricción activa (binding): El recurso se usa completamente. $\pi_j > 0$ significa que más recurso mejoraría Z.
- Restricción no activa: Hay excedente del recurso. $\pi_j = 0$ porque agregar más no ayuda.

Restricción 1
ACTIVA $\pi = 4$

La restricción "Restricción 1" está "activa" (binding). El precio sombra $\pi_1 = 4$.

Si aumentamos el RHS de esta restricción en 1 unidad, el costo mínimo Z "disminuirá" en 4 unidades. Relajar esta restricción tiene valor.

Restricción 2
ACTIVA $\pi = 2$

La restricción "Restricción 2" está "activa" (binding). El precio sombra $\pi_2 = 2$.

Si aumentamos el RHS de esta restricción en 1 unidad, el costo mínimo Z "disminuirá" en 2 unidades. Relajar esta restricción tiene valor.

Informe Ejecutivo con IA POWERED BY G40Q Regenerar Informe

Informe Ejecutivo - Optimización de Producción de Artículos x_1 y x_2

1. **RESUMEN EJECUTIVO**

- Objetivo:** Minimizar el costo total de producción manteniendo la demanda mínima y las restricciones de calidad.
- Resultado:** La fábrica debe producir **10 unidades de x_1** y **30 unidades de x_2** , con un costo total de **\$180**.

2. **DECISIÓN ÓPTIMA RECOMENDADA**

- Producción**
 - $x_1 = 10$ (mínimo exigido por calidad).
 - $x_2 = 30$ (resto de la producción necesaria para alcanzar 40 unidades totales).
- Acción inmediata**
 - Ajustar la línea de producción para que 30 unidades de x_2 se fabriquen simultáneamente con 10 unidades de x_1 .
 - Verificar que el equipo de control de calidad esté preparado para la producción mínima de x_1 .

3. **IMPACTO ECONÓMICO**

- Costo total mínimo:** \$180 ($6 \times 10 + 4 \times 30$).
- Ahorro comparado con una producción aleatoria:**
 - Si se produjera 40 unidades de x_1 (costo 240), la diferencia sería **\$60**.
 - Si se produjera 40 unidades de x_2 (costo 160), la diferencia sería **\$20**.
- Conclusión:** La combinación actual reduce costos en **\$20-\$60** respecto a otras estrategias de producción simples.

Figuras 12, 13. Análisis de Sensibilidad.

• **Análisis de sensibilidad**



VI. CONCLUSIONES

El sistema demostró un funcionamiento estable y coherente a lo largo de todo el flujo de procesamiento, desde la captura del problema inicial hasta la generación del modelo canónico y los modelos derivados para cada método. Las transformaciones aplicadas se realizaron sin inconsistencias, evidenciando que la estructura base del algoritmo está correctamente implementada y preparada para manejar diferentes configuraciones de entrada.

Los resultados obtenidos en cada método confirmaron la precisión de los cálculos y la correcta identificación de los elementos necesarios para avanzar entre etapas, como pivotes y variables críticas según el tipo de algoritmo empleado. Los modelos intermedios generados fueron consistentes y mantuvieron las reglas definidas para cada técnica, garantizando una ejecución reproducible y matemáticamente válida incluso en condiciones límite.

En conjunto, las pruebas permiten concluir que el sistema es funcional, confiable y adecuado para su uso en contextos académicos o de desarrollo. La modularidad y claridad en la construcción del flujo facilitan el mantenimiento y la futura ampliación del proyecto, consolidando la solución como una herramienta capaz de resolver problemas de manera determinista y verificable.

VII. REFERENCIAS

- [1] H. A. Taha, *Investigación de Operaciones*, 10.^a ed., Pearson, México, 2019.
- [2] F. S. Hillier y G. J. Lieberman, *Introducción a la Investigación de Operaciones*, McGraw-Hill, México, 2018.
- [3] J. K. Sharma, *Investigación de Operaciones: Teoría y Aplicaciones*, Pearson Educación, España, 2021.
- [4] Universidad Nacional de Colombia, *Fundamentos de Programación Lineal y Optimización*, UNAL, 2020.
- [5] Universidad de Sevilla, *Análisis de Sensibilidad en Programación Lineal*, Departamento de Métodos Cuantitativos, 2022.