	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01 PÁGINA: 1 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016
			VIGENCIA: 2016

LABORATORIO DE OPTIMIZACIÓN

UNIDAD ACADÉMICA: INGENIERÍA DE SISTEMAS

CURSO: OPTIMIZACIÓN

PRÁCTICA Nº 02: Programación Lineal y Método Gráfico

1. OBJETIVO

- *Diseñar y desarrollar un programa en Python que construya el modelo matemático de programación lineal para problemas de optimización, mostrando la solución mediante el método gráfico.*

2. CONSULTA PREVIA

Conceptos básicos de Programación Lineal, métodos de optimización, método Gráfico para resolver problemas de Programación Lineal, Herramientas informáticas para la programación en Python.

3. FUNDAMENTO TEÓRICO

La Programación Lineal es una técnica matemática utilizada para la optimización, es decir, la maximización o minimización de una función objetivo lineal sujeta a un conjunto de restricciones también lineales. Esta metodología es ampliamente aplicada en la investigación operativa y la economía para resolver problemas en los que los recursos son limitados y deben ser asignados de manera eficiente.


Componentes de un Modelo de Programación Lineal:

- **Variables de decisión:** *Representan las cantidades que el modelo intenta determinar. Estas variables son los "factores controlables" del problema, como la cantidad de productos a fabricar, la cantidad de horas de trabajo a asignar, etc.*
- **Función objetivo:** *Es la función lineal que se desea maximizar o minimizar. Por ejemplo, maximizar las ganancias o minimizar los costos. Se expresa en función de las variables de decisión.*
- **Restricciones:** *Son las limitaciones o condiciones que deben cumplir las variables de decisión. Estas restricciones suelen representar recursos limitados como tiempo, dinero, materiales, etc. Se expresan como ecuaciones o inecuaciones lineales.*
- **Región factible:** *Es el conjunto de todas las posibles soluciones que satisfacen las restricciones. Gráficamente, en problemas de dos variables, esta región es un polígono (convexo) en el plano cartesiano.*

ELABORADO POR:
REVISADO POR:

CARGO:

FECHA: 22/02/2023

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01	PÁGINA: 2 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016	
			VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN

Método Gráfico en Programación Lineal: El Método Gráfico es una técnica visual utilizada para resolver problemas de programación lineal con dos variables de decisión. Aunque es limitado en su aplicabilidad (solo funciona con dos variables), es muy útil para ilustrar los conceptos básicos de la PL y la naturaleza de las soluciones óptimas.

Pasos del Método Gráfico:

- **Paso 1: Graficar las Restricciones:**

Cada restricción se grafica como una recta en el plano cartesiano. Las inecuaciones dividen el plano en dos regiones, una que satisface la restricción y otra que no.

- **Paso 2: Determinar la Región Factible:**

La intersección de todas las regiones que satisfacen las restricciones define la región factible, que es donde se encuentran todas las soluciones posibles que cumplen con las restricciones.

- **Paso 3: Evaluar la Función Objetivo:**

Se evalúa la función objetivo en los vértices de la región factible. En problemas lineales, la solución óptima siempre se encuentra en un vértice de la región factible.

- **Paso 4: Seleccionar la Solución Óptima:**

Comparando los valores de la función objetivo en los vértices, se selecciona el valor que maximiza o minimiza la función, dependiendo del objetivo del problema.


Herramientas Matemáticas para la Implementación en Python:

Numpy: Es una biblioteca fundamental en Python para realizar cálculos numéricos. Proporciona soporte para grandes matrices y vectores multidimensionales, así como una colección de funciones matemáticas para operar con estos datos de manera eficiente. En el contexto de programación lineal, numpy facilita la manipulación de matrices que representan las restricciones y la función objetivo.

Matplotlib: Es una biblioteca de Python para la creación de gráficos estáticos, animados e interactivos. Es especialmente útil en el método gráfico para visualizar las rectas que representan las restricciones, la región factible, y la función objetivo en el plano cartesiano. La capacidad de graficar y sombrear regiones hace que matplotlib sea esencial para mostrar soluciones gráficas en problemas de programación lineal.

Scipy: Es una biblioteca que extiende las capacidades de numpy y proporciona herramientas y algoritmos de alto nivel para la optimización, integración, interpolación, entre otros. En programación lineal, scipy.optimize.linprog puede resolver problemas de PL mediante métodos más avanzados como el Simplex, cuando el problema no se puede resolver gráficamente debido a la presencia de más de dos variables.

ELABORADO POR:	CARGO:	FECHA: 22/02/2023
REVISADO POR:		

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01	PÁGINA: 3 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016	
			VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN

Aplicaciones de la Programación Lineal:

- **Producción y planificación:** Optimizar la producción y el uso de recursos en fábricas para maximizar las ganancias o minimizar los costos.
- **Logística y transporte:** Determinar las rutas de transporte más eficientes para minimizar costos o tiempos.
- **Finanzas:** Asignar carteras de inversión para maximizar retornos esperados bajo riesgo controlado.
- **Agricultura:** Planificar la siembra de cultivos para maximizar el rendimiento de las tierras dadas ciertas restricciones de recursos y mercado.

Google Colab

Google Colab, o Google Colaboratory, es un entorno de desarrollo interactivo que permite a los usuarios escribir y ejecutar código Python en un navegador. Es especialmente popular en la comunidad científica y de aprendizaje automático debido a sus potentes características, facilidad de uso, y la capacidad de compartir proyectos.

Características principales:

Entorno basado en la nube:

Google Colab se ejecuta en la nube, lo que significa que no necesitas instalar nada en tu computadora. Puedes acceder y trabajar en tus cuadernos desde cualquier lugar con conexión a Internet.

Soporte para Python y bibliotecas científicas:

Colab es compatible con Python y viene preinstalado con muchas bibliotecas populares como NumPy, Pandas, Matplotlib, TensorFlow, y muchas otras, lo que facilita el desarrollo de proyectos de análisis de datos, ciencia de datos, y aprendizaje automático.

GPU y TPU gratuitas:

Google Colab permite utilizar unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPU) de forma gratuita, lo que acelera considerablemente las tareas computacionalmente intensivas, como el entrenamiento de modelos de machine learning.


Interactividad:

Puedes escribir y ejecutar bloques de código de forma interactiva, lo que es útil para la experimentación y visualización de datos. Además, puedes incluir texto, imágenes, y enlaces, lo que hace que los cuadernos sean ideales para la documentación y la enseñanza.

Colaboración en tiempo real:

Al igual que Google Docs, Google Colab permite la colaboración en tiempo real. Puedes compartir tu cuaderno con otros usuarios, quienes pueden ver y editar el código simultáneamente.

ELABORADO POR:	CARGO:	FECHA: 22/02/2023
REVISADO POR:		

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
			VERSIÓN: 01	PÁGINA: 4 de 7
	PROCESO GESTION DE APOYO A LA ACADEMIA		FECHA: 02/09/2016	
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN

Integración con Google Drive:

Colab se integra fácilmente con Google Drive, lo que permite guardar y gestionar tus cuadernos en tu cuenta de Drive. Además, puedes importar y exportar archivos entre tu Drive y Colab.

Facilidad de uso:

Es muy accesible para principiantes y expertos por igual. Su interfaz es simple e intuitiva, lo que permite a los usuarios concentrarse en el desarrollo del código sin preocuparse por la configuración del entorno.

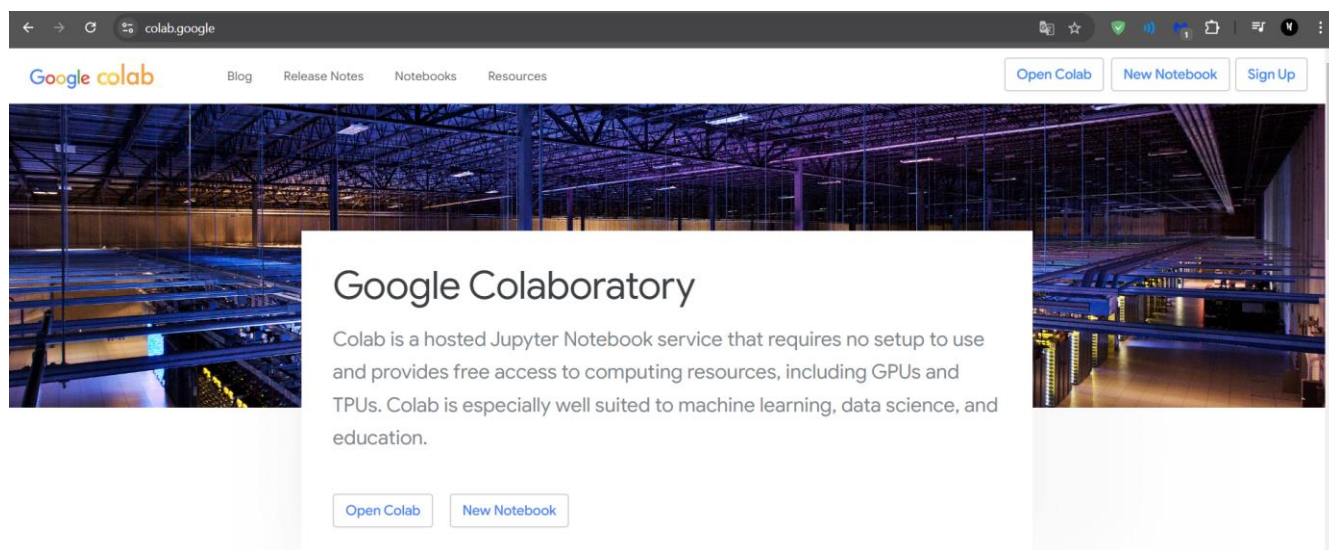


Figura 1. Google colab

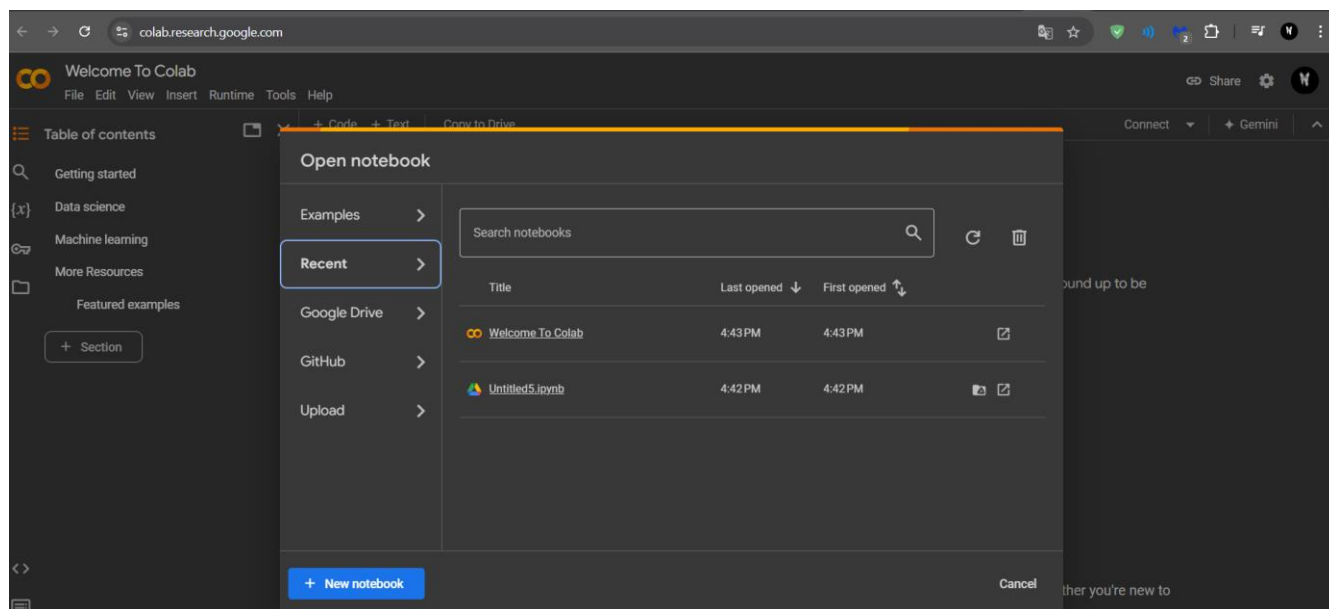

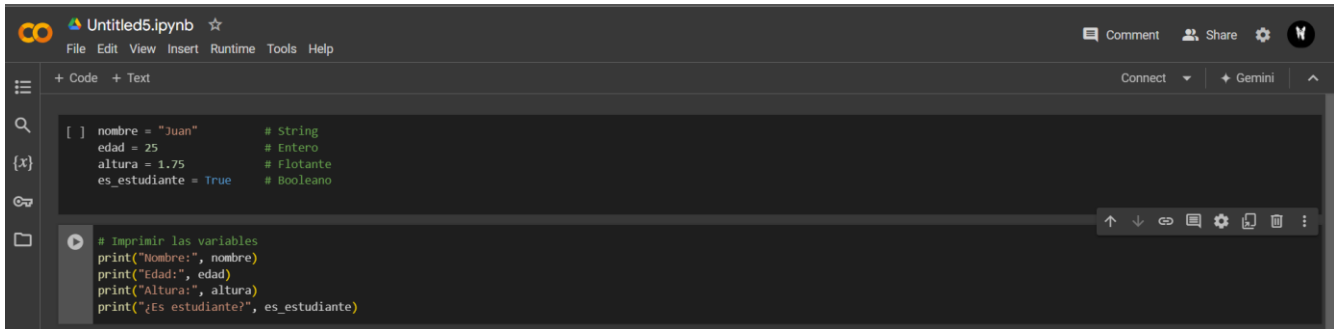


Figura 2. Nuevo libro en Google colab

ELABORADO POR:	CARGO:	FECHA: 22/02/2023
REVISADO POR:		

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01	PÁGINA: 5 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016	
			VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN



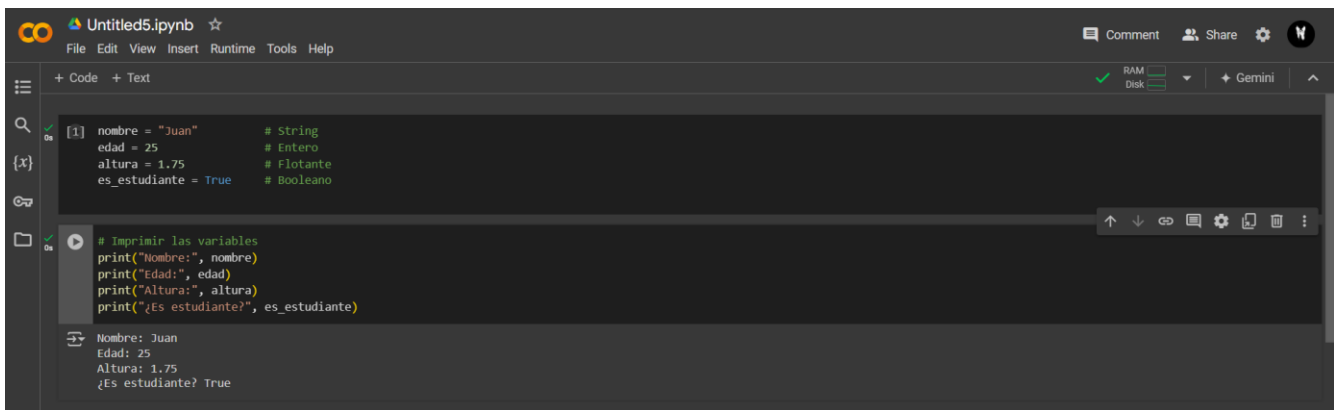
```

[ ] nombre = "Juan"      # String
    edad = 25           # Entero
    altura = 1.75       # Flotante
    es_estudiante = True # Booleano

# Imprimir las variables
print("Nombre:", nombre)
print("Edad:", edad)
print("Altura:", altura)
print("¿Es estudiante?", es_estudiante)

```

Figura 3. Layers de código



```

[1] nombre = "Juan"      # String
    edad = 25           # Entero
    altura = 1.75       # Flotante
    es_estudiante = True # Booleano

# Imprimir las variables
print("Nombre:", nombre)
print("Edad:", edad)
print("Altura:", altura)
print("¿Es estudiante?", es_estudiante)

```

Nombre: Juan
Edad: 25
Altura: 1.75
¿Es estudiante? True

Figura 4. Ejecución de código

4. EQUIPOS, MATERIALES Y REACTIVOS

Equipos	Materiales	Sustancias y/o Reactivos
Computador	Software Visual Studio code, Python3, Google Colab, bibliotecas (pandas, matplotlib, numpy)	

5. PROCEDIMIENTO O METODOLOGÍA

En esta práctica de laboratorio, los estudiantes diseñarán y desarrollarán un programa en Python que permita construir y resolver un modelo matemático de programación lineal utilizando el método gráfico. Se trabajará en grupos de dos estudiantes, empleando Visual Studio Code para el desarrollo del código y Google Colab para la ejecución y visualización de resultados.


Paso 1. Configuración del Entorno de Desarrollo

- Configurar Visual Studio Code con las extensiones necesarias para Python.
- Crear un nuevo proyecto en Visual Studio Code y organizar las carpetas para el código y los gráficos.

ELABORADO POR:
REVISADO POR:

CARGO:

FECHA: 22/02/2023

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01	PÁGINA: 6 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016	
			VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN

Paso 2. Desarrollo del Modelo Matemático

- Identificar y formular las funciones objetivo y las restricciones del problema de optimización.
- Implementar las ecuaciones matemáticas y restricciones en el código Python, asegurando que el programa pueda manejar **modelos matemáticos de al menos 2 restricciones**.

Paso 3. Visualización del Método Gráfico

El programa **debe visualizar el plano cartesiano** con las inecuaciones y la región factible de soluciones. Esto implica:

- Dibujar las líneas correspondientes a cada restricción en el plano cartesiano.
- Sombrear la región factible de soluciones, que representa el conjunto de puntos que satisfacen todas las restricciones.

El programa debe además **mostrar la evaluación de los vértices** de la región factible. Estos vértices representan los puntos donde se intersectan las restricciones, y son candidatos potenciales para la solución óptima. Finalmente, el programa debe **determinar y mostrar la solución óptima** para la correspondiente toma de decisiones, seleccionando el vértice que optimiza la función objetivo, ya sea maximización o minimización.

Paso 4. Ejecución y Pruebas en Google Colab

- Subir el código a Google Colab para ejecutar y visualizar los gráficos interactivos.
- Probar el programa con diferentes conjuntos de datos para validar la correcta implementación del método gráfico, asegurándose de que cumpla con las especificaciones de manejo de restricciones, visualización gráfica y determinación de la solución óptima.

Paso 5. Documentación del Proceso


Elaborar un informe técnico siguiendo el formato Working Paper de IEEE, que documente todo el proceso de desarrollo, resultados y conclusiones. El informe debe contener los siguientes elementos mínimos:

- Título de la práctica
- Autores
- Introducción
- Referente teórico
- Procedimiento
- Resultados
- Conclusiones
- Bibliografía

6. RESULTADOS

Como evidencia del desarrollo de esta práctica. Subir un archivo comprimido al aula virtual antes de la fecha límite establecida que contenga el código fuente del programa y un informe técnico del proceso realizado este archivo

ELABORADO POR:	CARGO:	FECHA: 22/02/2023
REVISADO POR:		

	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112	
	PROCESO GESTION DE APOYO A LA ACADEMIA		VERSIÓN: 01	PÁGINA: 7 de 7
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016	
			VIGENCIA: 2016	

LABORATORIO DE OPTIMIZACIÓN

será la evidencia del desarrollo de la práctica y debe reflejar tanto la implementación técnica como la comprensión teórica del método gráfico en la programación lineal.

7. BIBLIOGRAFÍA

- [1]. Murty, K. G. (1983). **Linear Programming**. John Wiley & Sons.
- [2]. Chvatal, V. (1983). **Linear Programming**. W. H. Freeman and Company.
- [3]. Papadimitriou, C. H., & Steiglitz, K. (1998). **Combinatorial Optimization: Algorithms and Complexity**. Dover Publications.
- [4]. Matplotlib Documentation. (2024). **Matplotlib: Visualization with Python**. Recuperado de <https://matplotlib.org/>
- [5]. NumPy Documentation. (2024). **NumPy: The fundamental package for scientific computing with Python**. Recuperado de <https://numpy.org/>
- [6]. Google Colab. (2024). **Google Collaboratory**. Recuperado de <https://colab.research.google.com/>

ELABORADO POR:	CARGO:	FECHA: 22/02/2023
REVISADO POR:		