

# BÚSQUEDA Y ORDENAMIENTO DE ALGORITMOS

## CASO PRÁCTICO: GESTIÓN DE INVENTARIO DE UNA TIENDA ONLINE

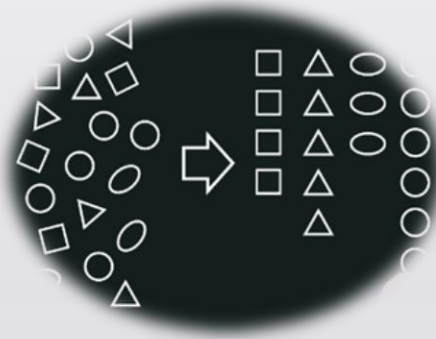
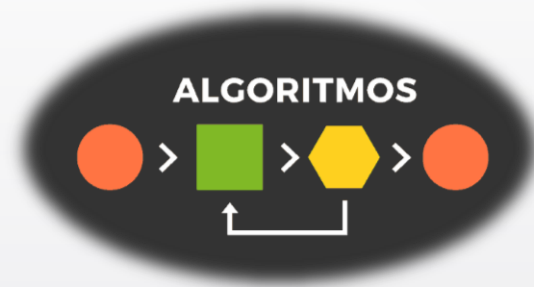
Trabajo elaborado por:

- ❖ Arjona, Martin Joel
- ❖ Arjona, Paola Yanina

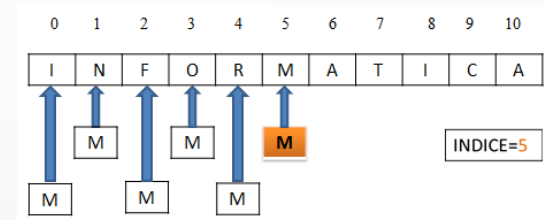


# INTRODUCCIÓN

## ALGORITMOS DE BUSQUEDA Y ORDENAMIENTO



# MARCO TEORICO

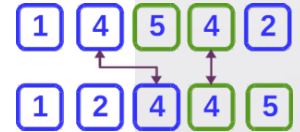


## ✓ ALGORITMOS DE BÚSQUEDA Y ORDENAMIENTO

- Permiten organizar y acceder eficientemente a grandes volúmenes de datos.
- Su correcta implementación mejora el rendimiento y optimiza soluciones..

## ✓ Ordenamiento

- Proceso de reorganizar datos para facilitar su búsqueda.
- Tipos comunes: Merge Sort, Quick Sort, Heap Sort, Bubble Sort, Insertion Sort, Selection Sort.
  - Merge Sort: eficiente, estable y usado en bibliotecas estándar como Python..



## ✓ Búsqueda

- Los algoritmos de búsqueda localizan elementos en estructuras de datos, siendo los principales la búsqueda lineal y la binaria.
- La búsqueda binaria es más eficiente, pero requiere que los datos estén ordenados.



# CASO PRÁCTICO

**OBJETIVO:** Desarrollar un sistema de gestión de inventario en Python que permita cargar datos desde un archivo CSV, ordenar los productos y realizar búsquedas de forma eficiente.

A través de la implementación de algoritmos como Merge Sort, búsqueda binaria y búsqueda lineal, se busca comparar el rendimiento entre distintos métodos de búsqueda midiendo sus tiempos de ejecución.

El programa se desarrollará en Python mediante el uso de Visual Studio Code







# RESULTADOS

## ✓ 1. Carga y estructura de datos

- Los 500 productos se cargaron correctamente desde el CSV, con tipos de datos adecuados para cada campo.

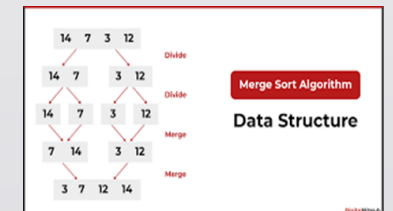


## ✓ 2. Ordenamiento de productos

- El ordenamiento por nombre (con sorted y lambda) y por precio (Merge Sort) funcionó correctamente y sin errores..

## ✓ Búsqueda por ID: Comparación de algoritmos

- La búsqueda binaria fue significativamente más rápida y estable que la búsqueda lineal en 100 pruebas repetidas.





# RESULTADOS

matplotlib



## ✓ 4. Visualización gráfica y análisis

- Los gráficos mostraron tiempos bajos y constantes para búsqueda binaria, mientras que la búsqueda lineal presentó mayor variabilidad y picos altos.

Figura 1. Comparación de tiempos de búsqueda para el producto con ID "P250"

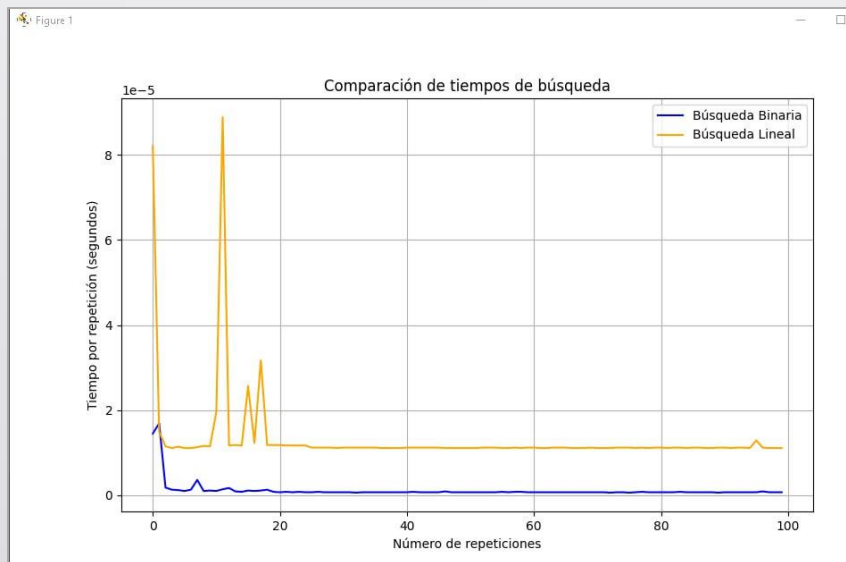
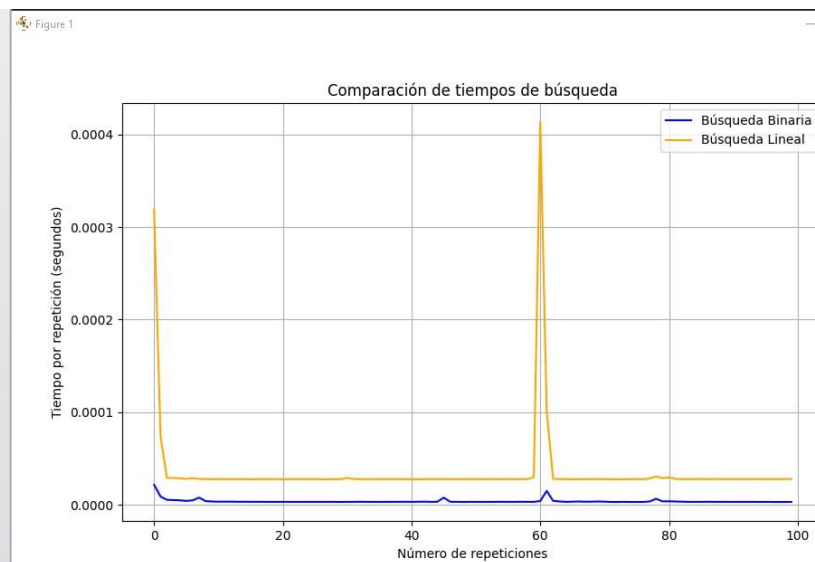


Figura 2. Comparación de tiempos de búsqueda para el producto con ID "P480"





# CONCLUSIONES



- ✓ Este trabajo reforzó conocimientos clave sobre algoritmos de búsqueda y ordenamiento aplicados a la gestión de inventarios.
- ✓ Se confirmó que la búsqueda binaria es mucho más eficiente que la lineal para grandes volúmenes, siempre que la lista esté ordenada.
- ✓ Se adquirió experiencia práctica en el manejo de archivos CSV y el uso de bibliotecas Python como csv, matplotlib y time.
- ✓ Las dificultades iniciales con sintaxis y configuración se superaron con investigación y pruebas constantes.

Como mejora, se podrían agregar funciones de filtrado, exportación de resultados y comparar con otros algoritmos o estructuras de datos.



Gracias!