

# Algoritmos Avanzados

SANTIAGO RODRIGUEZ GUTIERREZ  
LUIS ANTONIO SALINAS GONZALEZ  
ROBERTO CASTRO GOMEZ

DECEMBER 2025

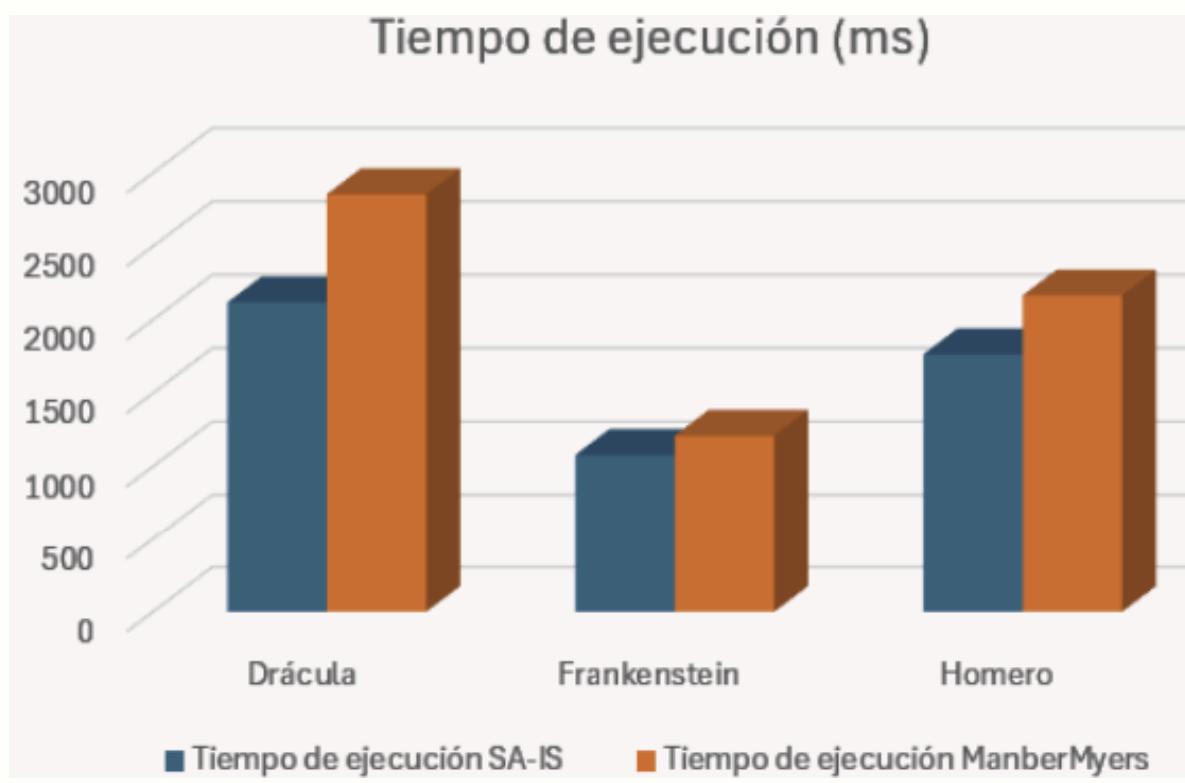
## INTRODUCCION

Este póster resume los hallazgos de tres actividades clave del curso TC2038, donde se implementaron y analizaron algoritmos avanzados en tres áreas fundamentales:

- Construcción de arreglos de sufijos (E1)
- Compresión y rutas óptimas mediante Huffman y Dijkstra (E2)
- Comparación de Textos (E3)

Ambas evidencias permiten comparar eficiencia, optimización y toma de decisiones algorítmicas en problemas reales.

## EVIDENCIA 1



### SA-IS VS Manber & Myers

- SA-IS es más rápido, pero tiene un mayor consumo de memoria, ya que requiere estructuras auxiliares más grandes.
- SA-IS tiene complejidad  $O(n)$  y mostró mejores tiempos en las pruebas realizadas.
- Manber & Myers, aunque es  $O(n \log n)$ , usa menos memoria y tiene constante más pequeña.
- Cambiar diccionarios por vectores booleanos para clasificar posiciones L/S reduce significativamente el uso de memoria.
- Pasar parámetros por referencia en C++ disminuye duplicación de datos, haciendo ambas implementaciones más eficientes.

## EVIDENCIA 2

### Compresion de Huffman

- El algoritmo de Huffman logra una compresión promedio del 68% del tamaño original
- En los 30 archivos analizados (3 idiomas), la compresión promedio fue 0.68, es decir, el archivo comprimido ocupa solo 68% del original.
- El promedio de bits por símbolo fue 4.6.
- El español fue el idioma con mejor compresión, seguido del inglés y, por último, el francés.
- Esto confirma que la entropía del lenguaje influye directamente en la eficiencia del algoritmo.

### Algoritmo de Dijkstra

- El uso de Dijkstra modificado reduce tiempos reales en rutas con transbordos.
- Se añadió una penalización de 4 minutos por cambio de línea, lo que permite calcular rutas más realistas para el metro.
- La complejidad del algoritmo sigue siendo  $O(E \log V)$  y mantiene alta eficiencia incluso con el costo por transbordo.

filename	language	original_size	compressed_size	compression_factor	avg_bits_per_symbol
es_wiki_Intel.es	es	53131	34351	0.6465340385085919	4.3992190369410995
es_wiki_Seg.es	es	442311	257559	0.5823029497344628	4.538214800018581
es_wiki_Mones.es	es	26615	19774	0.7429644937065565	4.4522080242358337

language	huff_file	decoder	runs	mean_seconds	stdev_seconds
es	es_wiki_Inteligencia_artificial.txt.huff	inverse	500	0.0220890894997546	0.0014242692622442993
es	es_wiki_Inteligencia_artificial.txt.huff	tree	500	0.010518007470003796	0.0003506593455199867
fr	fr_wiki_Seconde_Guerre_mondiale.txt.huff	inverse	500	0.07262100719595764	0.0017308568415582621
fr	fr_wiki_Seconde_Guerre_mondiale.txt.huff	tree	500	0.03511164456798724	0.0007719805348986377
en	en_wiki_Climate_change.txt.huff	inverse	500	0.026842964537972875	0.0007620545459739062
en	en_wiki_Climate_change.txt.huff	tree	500	0.013055433061970689	0.0004798579721874541

## EVIDENCIA 3

- Los embeddings modernos permiten medir similitud basada en significado y no solo en palabras.
- La matriz de similitud global permite identificar relaciones entre todos los textos de la colección.
- La implementación de un heatmap facilita ver qué partes de ambos textos se parecen más.
- La actividad demuestra cómo los modelos de lenguaje pueden utilizarse para análisis semántico real en colecciones grandes de textos.