



# Sopa de Letras.

Laboratorio 1: Sistemas Operativos.

Integrantes: Felipe Mellado y Cristopher Fuentes.

Paralelo 201.

## 1. Preguntas y análisis:

1.1) Tabla de tiempos de ejecución:

Palabra	Orientación	Tamaño sopa	Tiempo
banco	horizontal	200x200	0.001436948776245
Carne	vertical	200x200	0.001647949218750
carro	vertical	100x100	0.000638008117676
casa	horizontal	50x50	0.000007867813110
Cobre	horizontal	200x200	0.001923084259033
Gamer	vertical	100x100	0.000471115112305
gato	vertical	50x50	0.000581026077271
hola	vertical	50x50	0.000264883041382
Jamon	vertical	100x100	0.000867843627930
perro	horizontal	50x50	0.000051975250244
tapia	horizontal	100x100	0.000087976455688
viktor	horizontal	100x100	0.000280141830444

1.2) Palabra con mayor tiempo de ejecución:

Cobre, debido a que la sopa es del mayor tamaño

1.3) Orientación con mayor tiempo de ejecución:

Luego de crear una tabla en Excel, calculando el tiempo promedio entre horizontal y vertical, nos dimos cuenta que el código demora más tiempo en encontrar las palabras que están de forma horizontal.



#### 1.4) Propuesta para la optimización del código:

En primera instancia, creamos una función que hace uso de múltiples manejos de Strings, sumándole también el hecho de que lo que hace es buscar la palabra dentro del mismo archivo, aumentando así el tiempo de lectura y escritura. Como propuesta, lo que hicimos fué: utilizar matrices para hacer la búsqueda de la palabra dentro de la sopa de letras sin utilizar una gran cantidad de strings. Si bien, los tiempos no mejoran de manera exponencial, hay un ligero cambio entre los tiempos de ejecución del primer código y el segundo.

#### 1.5) Tabla de tiempos de ejecución (código optimizado):

Palabra	Orientación	Tamaño sopa	Tiempo [s]
banco	horizontal	200x200	0.000277042388916
Carne	vertical	200x200	0.000301122665405
carro	vertical	100x100	0.000076055526733
casa	horizontal	50x50	0.000000953674316
Cobre	horizontal	200x200	0.000132799148560
Gamer	vertical	100x100	0.000132799148560
gato	vertical	50x50	0.000019788742065
hola	vertical	50x50	0.000020027160645
Jamon	vertical	100x100	0.000076055526733
perro	horizontal	50x50	0.000010967254639
tapia	horizontal	100x100	0.000009059906006
viktor	horizontal	100x100	0.000092029571533

#### 1.6) Posibles conocimientos del curso que nos hubieran simplificado la creación del código:

Según lo que vimos en introducción al curso, éste nos puede enseñar cómo medir el rendimiento de un sistema operativo, usando la siguiente fórmula:

$$T_{Ej} : T_{CPU} + T_{RESP} + T_{ECR*} + T_{E/S} + T_{HW}$$

Donde:



- $T_{Ej}$ : Tiempo de ejecución
- $T_{CPU}$ : Tiempo de uso de CPU
- $T_{RESP}$ : Tiempo de respuesta desde que se ordena la ejecución del programa hasta que se le asigna por primera vez al procesador
- $T_{ECR*}$ : Tiempo de espera en cola ready sin cortar el tiempo de respuesta
- $T_{E/S}$ : Tiempo utilizado en ejecutar operaciones E/S y tiempo asociado a la lectura
- $T_{HW}$ : Tiempo de escritura en los distintos componentes de HW.

Con ésto, no sólo hubiésemos hecho uso de la disminución de  $T_{E/S}$  para mejorar el programa, sino que podríamos haber tratado de disminuir los 4 factores restantes para así poder mejorar el tiempo de ejecución del programa.

## 2. Conclusiones:

En conclusión, el análisis detallado de los tiempos de ejecución y la implementación de mejoras puntuales nos permitieron entender el comportamiento del código y lograr una optimización perceptible en los tiempos de búsqueda de palabras en la sopa de letras. Además, identificamos cómo conceptos de Sistemas Operativos podrían aplicarse para abordar el rendimiento de manera más integral. Este ejercicio no solo nos brindó una valiosa experiencia en la mejora de rendimiento de programas, sino que también ilustró cómo los conocimientos teóricos pueden ser aplicados de manera práctica en la optimización de sistemas y procesos.