Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**ALGORITMOS Y ESTRUCTURA DE DATOS**

**Análisis de Algoritmos Empírico**

**Docente:** M.I Palacios Guerreros Alma Leticia

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 1261509

Contents

[INTRODUCCION 3](#_Toc53362604)

[DESARROLLO 3](#_Toc53362605)

[IMPLEMENTACION EN C 3](#_Toc53362606)

[IMPLEMENTACION EN JAVA 7](#_Toc53362607)

[TABLA COMPARATIVA 10](#_Toc53362608)

[CONCLUSIONES 10](#_Toc53362609)

[ENLACE A CARPETA CON CODIGOS (DRIVE) 10](#_Toc53362610)

# INTRODUCCION

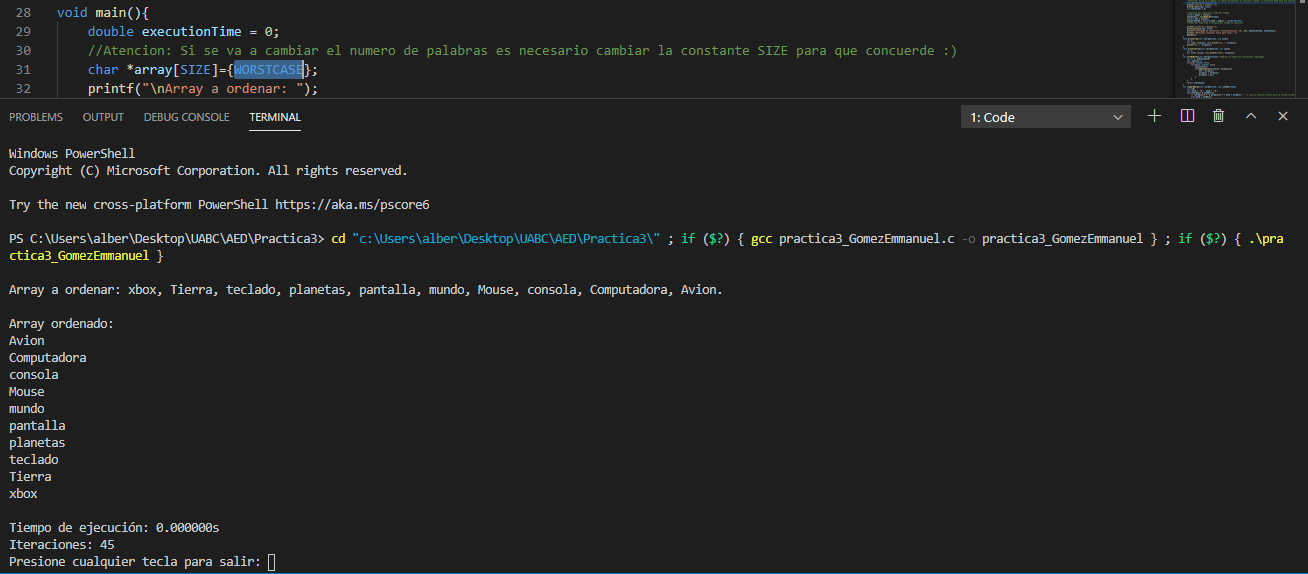
El método burbuja es un método de ordenación muy simple, pero no muy eficiente, de elementos X0, X1, X2, X3,… Xn en orden ascendente consiste en encontrar el elemento más pequeño para la primera posición, luego el segundo elemento más pequeño para la segunda posición, esto se hace comparando cada elemento del vector con el resto de los elementos, en cada ocasión que encuentre que están no están en orden, los elementos deben intercambiarse. Al terminar la primera iteración el elemento más pequeño estará en la primera posición del vector. En la siguiente iteración se comienza por el segundo elemento y al terminar en la posición 1 estará el segundo elemento más pequeño. Las iteraciones continúan hasta que se llega a la penúltima posición en la que se ordenan los últimos dos. (Método Burbuja)

# DESARROLLO

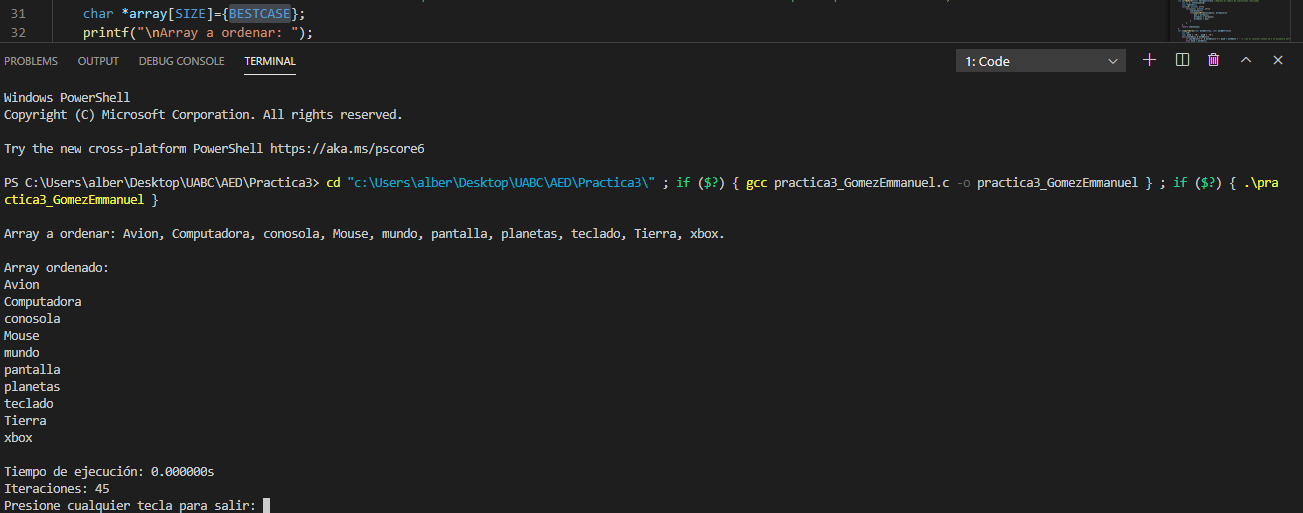
Diseñe e implemente el algoritmo en lenguaje C. Utilizando inicialmente un arreglo de 10 cadenas, determine el tiempo que tarda el algoritmo con los siguientes datos:

# IMPLEMENTACION EN C

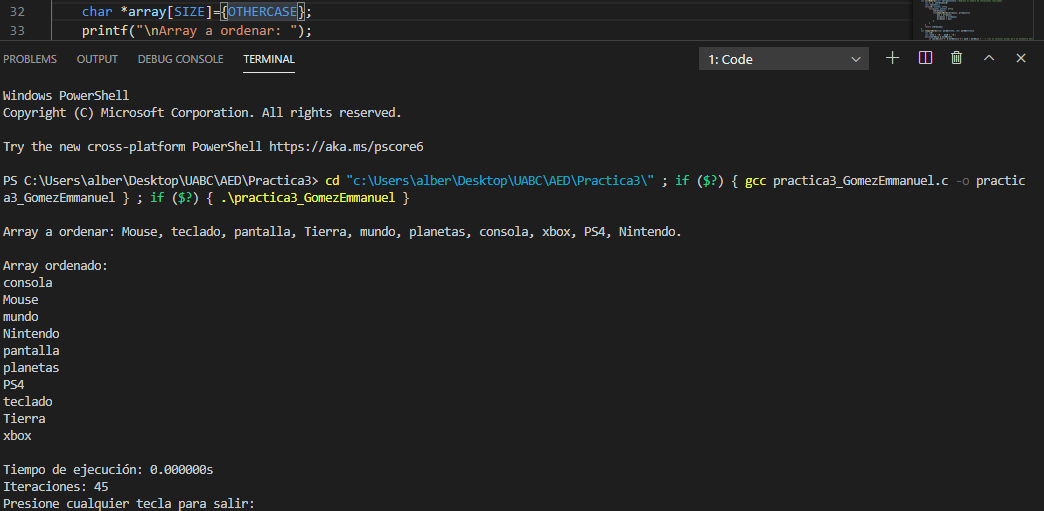
1. El tiempo y la cantidad de iteraciones para el peor de los casos



1. El tiempo y la cantidad de iteraciones para el mejor de los casos



1. El tiempo y la cantidad de iteraciones para cualquier otro caso

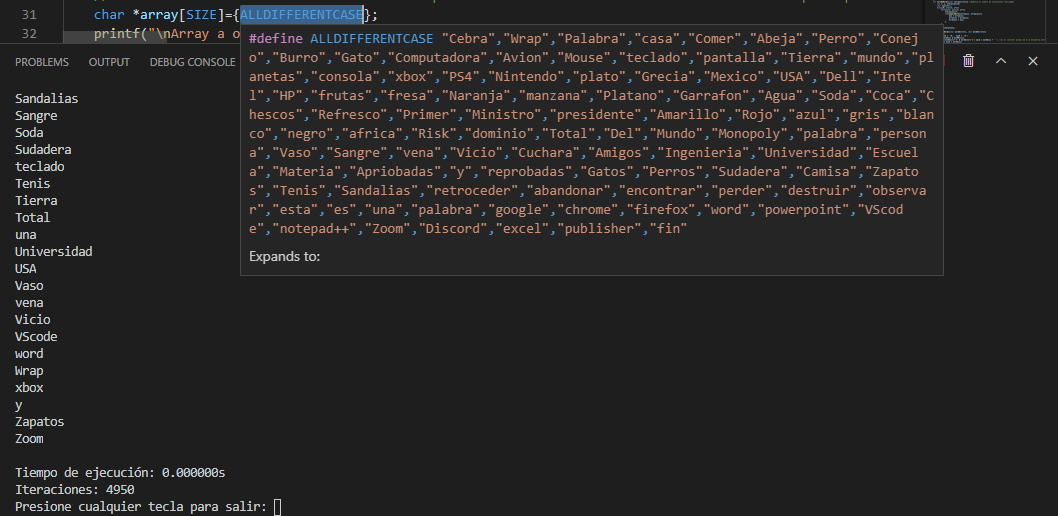


1. ¿Qué pasa si todos los datos son iguales?

Debido a que al ser todos iguales el algoritmo necesita comparar toda la cadena para poder darse cuenta de que ambas son iguales y, por lo tanto, aunque no sea necesario hacer cambios, el tiempo de ejecución aumenta dependiendo del tamaño de las cadenas.

Para representar mejor este ejemplo, se hicieron dos ejecuciones: Una cuando el array tiene 100 cadenas diferentes y otra cuando las 100 cadenas del array son exactamente iguales.

Array de 100 elementos diferentes



Array de 100 elementos iguales



1. ¿Qué pasa al cambiar el tipo de datos del vector de cadenas a entero?

Debido que al manejar un entero tiene el mismo peso que manejar una cadena de 1 char, al manejar enteros el numero de comparaciones, al igual que el tiempo de ejecución disminuye.

1. ¿Qué pasa al aumentar el tamaño del arreglo?

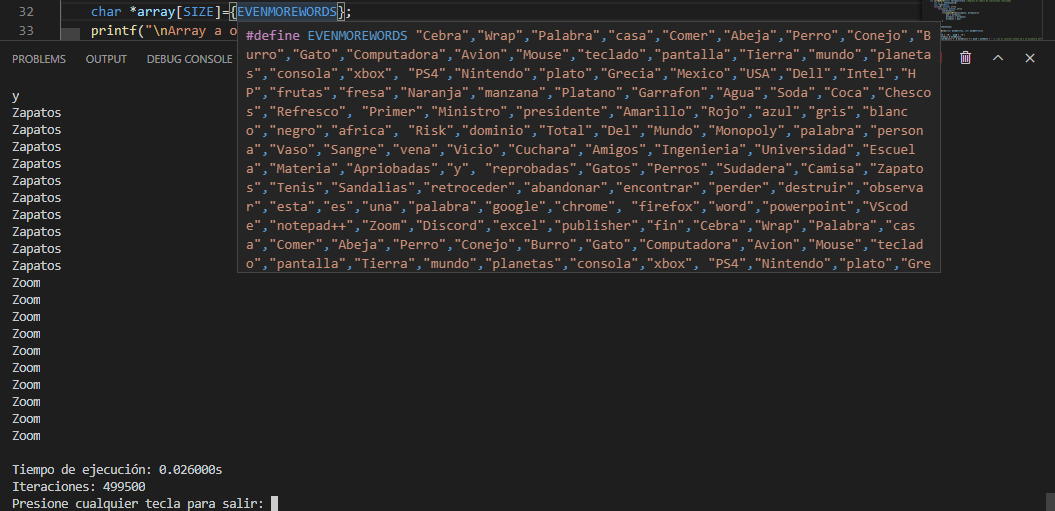
El número de comparaciones aumenta, así como el tiempo de ejecución.

(Ejemplo con 1k palabras)



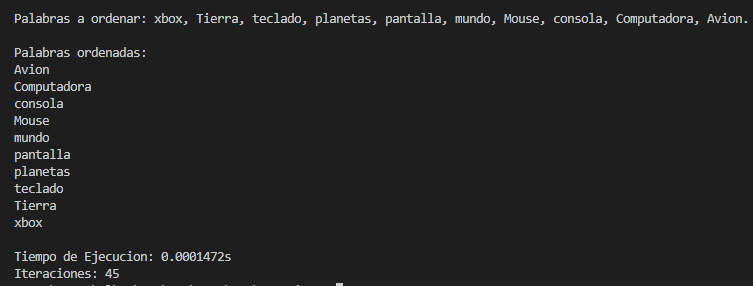
1. ¿Cómo afecta el tiempo de ejecución la carga del sistema? Compruebe abriendo varias aplicaciones simultáneamente.

(Mismas 1000 palabras, más aplicaciones abiertas)

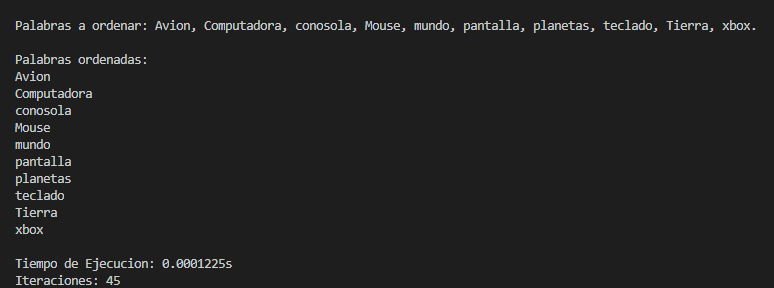


# IMPLEMENTACION EN JAVA

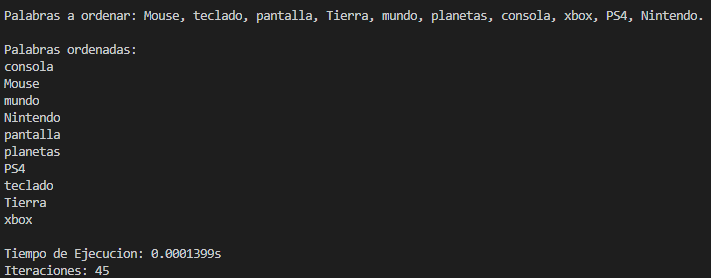
1. El tiempo y la cantidad de iteraciones para el peor de los casos



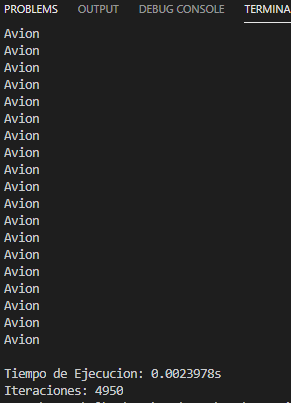
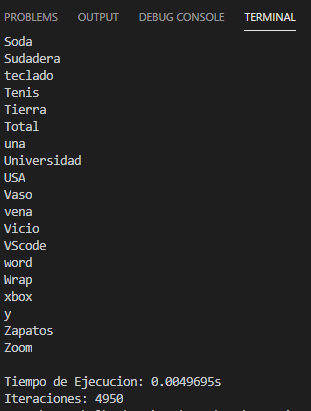
1. El tiempo y la cantidad de iteraciones para el mejor de los casos



1. El tiempo y la cantidad de iteraciones para cualquier otro caso



1. ¿Qué pasa si todos los datos son iguales?

En el caso de Java resuelve la comparación más rápida cuando ambas cadenas son iguales (Se buscó, pero no se logró comprender la información sobre como Java hace la comparación)

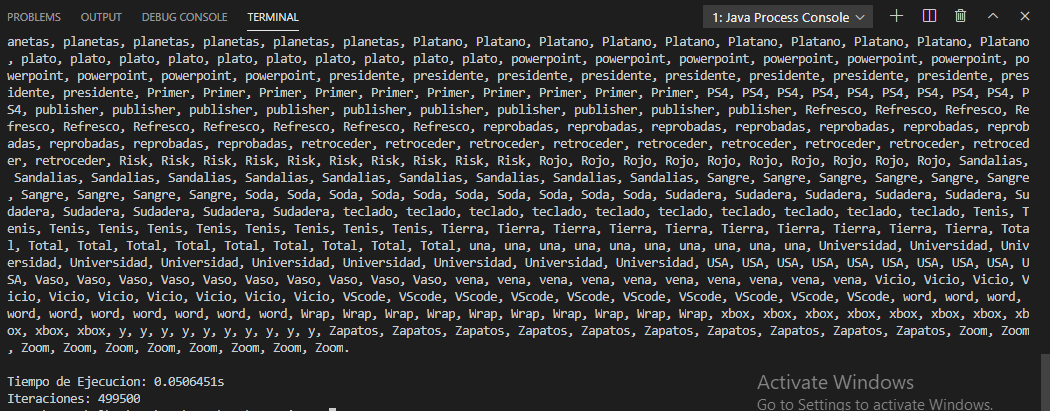
1. ¿Qué pasa al cambiar el tipo de datos del vector de cadenas a entero?

Pasa exactamente igual que con el lenguaje C, el tiempo disminuye.

1. ¿Qué pasa al aumentar el tamaño del arreglo?

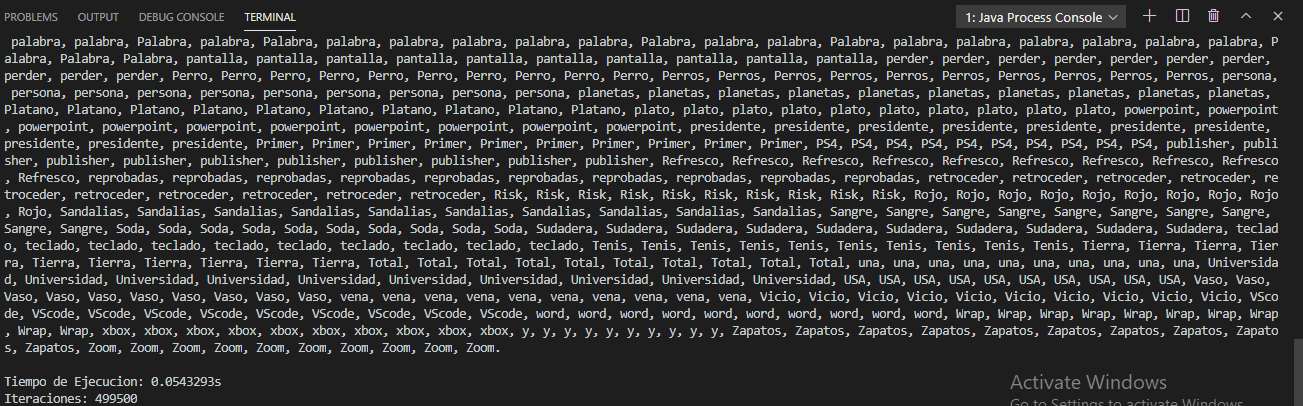
El número de comparaciones aumenta, así como el tiempo de ejecución.

(Ejemplo con 1k palabras, las mismas 1k usadas en el programa en C)

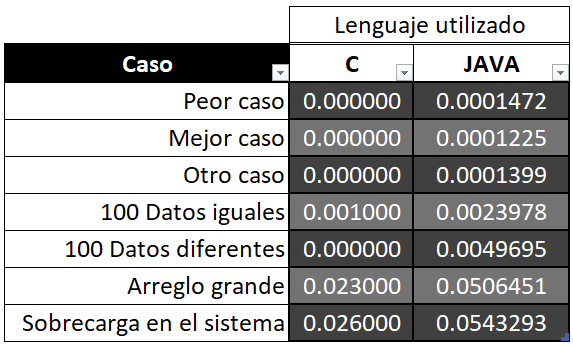


1. ¿Cómo afecta el tiempo de ejecución la carga del sistema? Compruebe abriendo varias aplicaciones simultáneamente.

(Mismas 1000 palabras, más aplicaciones abiertas)



# TABLA COMPARATIVA



# CONCLUSIONES

Lo primero que captó mi atención en cuanto al programa escrito en C es como la medición del tiempo es muy imprecisa cuando se trata de tiempos muy cortos, en cambio, en Java el tiempo puede ser medido hasta en nanosegundos. También el hecho de que en Java la operación compareTo es notoriamente más rápida cuando se trata de cadenas exactamente iguales, lo cual contrasta bastante al programa que hice en C. Ya que, en C debido a ser iguales, se hacen muchas más comparaciones incrementando substancialmente el tiempo de ejecución. Quede bastante intrigado por aprender el algoritmo utilizado para comparar dos cadenas en Java.

# ENLACE A CARPETA CON CODIGOS (DRIVE)

<https://drive.google.com/drive/folders/16-aOKBQ7uIC1xRNVOrr5xjhcPoGqSo2b?usp=sharing>