

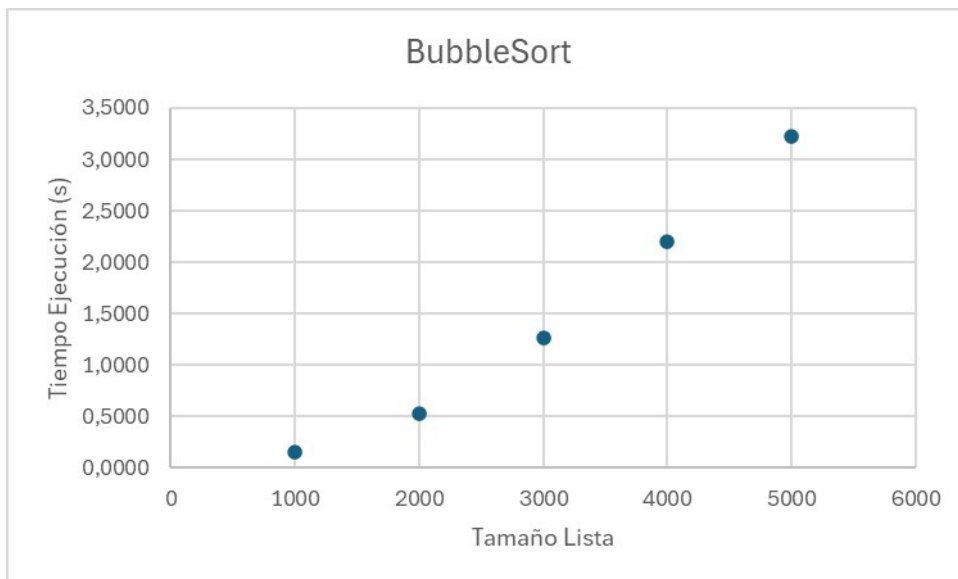
## Quiz 4

Guillermo Sánchez Cedeño

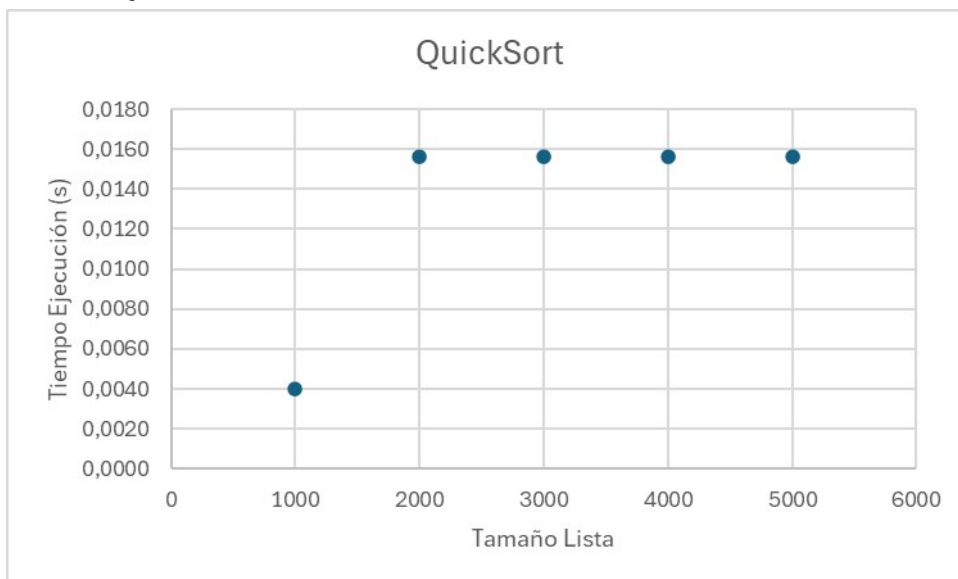
Luis Catro Montenegro

### Problema 1

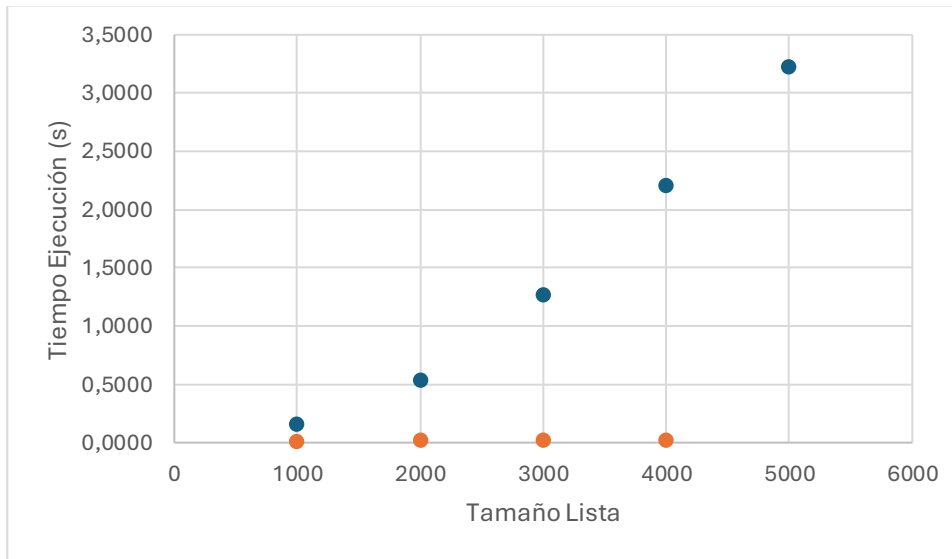
#### Grafica BubbleSort



#### Grafica QuickSort



## QuickSort vs BubbleSort



Azul: BubbleSort

Naranja: QuickSort

5. Responda las siguientes preguntas

5a. ¿Cuál algoritmo es más rápido y por qué?

QuickSort es más rápido ya que el nivel de complejidad de su BigO es menor además de que su implementación es recursiva, cosa que no se ve en el BubbleSort, en donde su complejidad aumenta de forma exponencial debido al doble ciclo for.

5b. ¿El tiempo de ejecución será el mismo si la implementación del algoritmo es iterativa o recursiva?

No, si se usa recursividad generalmente la complejidad no sube exponencialmente, lo que hace que el algoritmo dure menos en ejecución.

5c. ¿Es posible que exista un algoritmo de ordenamiento que sea muy eficiente en consumo de recursos pero que a la vez sea relativamente rápido?

Sí, el QuickSort si se trabaja con listas de mediano tamaño llega a ser muchísimo más eficiente que muchos otros algoritmos en términos de tiempo ejecución y uso de recursos, igualmente si hablamos del MergeSort que tiene un nivel de complejidad similar.

5d. Suponga que se planea ejecutar el algoritmo en un sistema computacional con extremadamente bajos recursos de memoria. ¿Cuál de los dos algoritmos de ordenamiento escogería y por qué?

Utilizaría BubbleSort, aunque su nivel de complejidad es mucho mayor, su uso de memoria es casi nulo, pues este no genera llamadas en pila, simplemente se ejecuta sobre el arreglo ya creado.

## Problema 2

1. ¿Cuál es la diferencia entre el algoritmo de búsqueda lineal y búsqueda por interpolación?

**Búsqueda lineal:** se trata de una búsqueda secuencial, visitando cada uno de los elementos de un arreglo.

**Búsqueda por interpolación:** es una versión modificada de la búsqueda binaria, en donde se adivina un índice cerca al elemento buscado. (Utilizando una fórmula matemática)

2. Suponga que se tiene que buscar un elemento en una lista desordenada, pero se desea optimizar el tiempo de búsqueda por sobre cualquier otra métrica ¿Cómo se podría hacer eso?

Si se desea por sobre todas las cosas priorizar el tiempo de búsqueda, una estrategia favorable sería ordenar el arreglo utilizando QuickSort o MergeSort y luego buscar el elemento utilizando Interpolación.

3. Busque y explique alguna aplicación de la vida real donde el tiempo de búsqueda en una lista o en un arreglo sea crítico para que la aplicación se pueda dar.

En las bases de datos se puede implementar una columna llamada índice, que se trata de una lista ordenada, gracias a esto los tiempos entre solicitudes se reducen considerablemente.