Cristian Alejandro Agudelo Parada

Análisis de complejidad de los requerimientos en Notación O.

Requerimiento 1:
Filtrado de ofertas: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número total de ofertas de trabajo. Esto se debe a que se debe revisar cada oferta para ver si cumple con los criterios de filtrado.

Selección de las N ofertas más recientes: La función nlargest de pandas tiene una complejidad de tiempo de O(n log n), donde n es el número de ofertas filtradas. Esto se debe a que la función ordena las ofertas por fecha de publicación antes de seleccionar las N más recientes.

Impresión de la información de las ofertas: Esta operación tiene una complejidad de tiempo de O(N), donde N es el número de ofertas seleccionadas. Esto se debe a que se imprime la información de cada oferta.

Por lo tanto, la complejidad de tiempo total del código es O(n) + O(n log n) + O(N) = O(n log n),

Requerimiento 2:

Filtrado de ofertas: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número total de ofertas de trabajo. Esto se debe a que se debe revisar cada oferta para ver si cumple con los criterios de filtrado.

Conteo de ofertas: La operación de conteo tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Conteo de ofertas por nivel de experticia: La función value_counts de pandas tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Ordenamiento de ofertas: La función sort_values de pandas tiene una complejidad de tiempo de O(n log n), donde n es el número de ofertas filtradas.

Por lo tanto, la complejidad de tiempo total del código es O(n) + O(n) + O(n) + O(n log n) = O(n log n), asumiendo que n es mucho mayor que el número de ofertas que se están contando o ordenando. Esto significa que el tiempo de ejecución del código aumentará logarítmicamente con el número de ofertas de trabajo.

Requerimiento 3:
Filtrado de ofertas: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número total de ofertas de trabajo. Esto se debe a que se debe revisar cada oferta para ver si cumple con los criterios de filtrado.

Cálculo del total de ofertas: Esta operación tiene una complejidad de tiempo de O(1), ya que simplemente se está contando el número de filas en el DataFrame filtrado.

Cálculo del total de empresas únicas y ciudades con ofertas: La función nunique de pandas tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Cálculo de la ciudad con mayor y menor número de ofertas: La función value_counts de pandas tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Ordenamiento de ofertas: La función sort_values de pandas tiene una complejidad de tiempo de O(n log n), donde n es el número de ofertas filtradas.

Por lo tanto, la complejidad de tiempo total del código es O(n) + O(1) + O(n) + O(n) + O(n log n) = O(n log n), asumiendo que n es mucho mayor que el número de ofertas que se están contando o ordenando.

Requerimiento 4:

Cálculo del total de ofertas: Esta operación tiene una complejidad de tiempo de O(1), ya que simplemente se está contando el número de filas en el DataFrame filtrado.

Cálculo del promedio de salario: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Cálculo de la ciudad con mayor y menor número de ofertas: La función value_counts de pandas tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Impresión de la información de las ciudades: Esta operación tiene una complejidad de tiempo de O(N), donde N es el número de ciudades seleccionadas.

Por lo tanto, la complejidad de tiempo total del código es O(1) + O(n) + O(n) + O(N) = O(n),

Requerimiento 5:

Filtrado de ofertas: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número total de ofertas de trabajo. Esto se debe a que se debe revisar cada oferta para ver si cumple con los criterios de filtrado.

Obtención de los niveles de experiencia únicos: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número total de ofertas de trabajo.

Para cada nivel de experiencia: Este bucle se ejecuta un número constante de veces (una vez por cada nivel de experiencia), por lo que no afecta a la complejidad de tiempo en función de n.

Filtrado de ofertas por nivel de experiencia: Esta operación tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Agrupación de ofertas por país y conteo: La función groupby de pandas tiene una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas.

Para cada país en los N primeros países: Este bucle se ejecuta N veces, y dentro del bucle se realizan varias operaciones que tienen una complejidad de tiempo de O(n), donde n es el número de ofertas filtradas para un país y nivel de experiencia específicos.

Por lo tanto, la complejidad de tiempo total del código es O(n) + O(n) + O(n) + O(n) + O(nN) = O(nN)

Informe de Pruebas de Tiempos de Ejecución

Configuración del Entorno de Pruebas

Las pruebas se realizaron en un computador HP con las siguientes especificaciones:

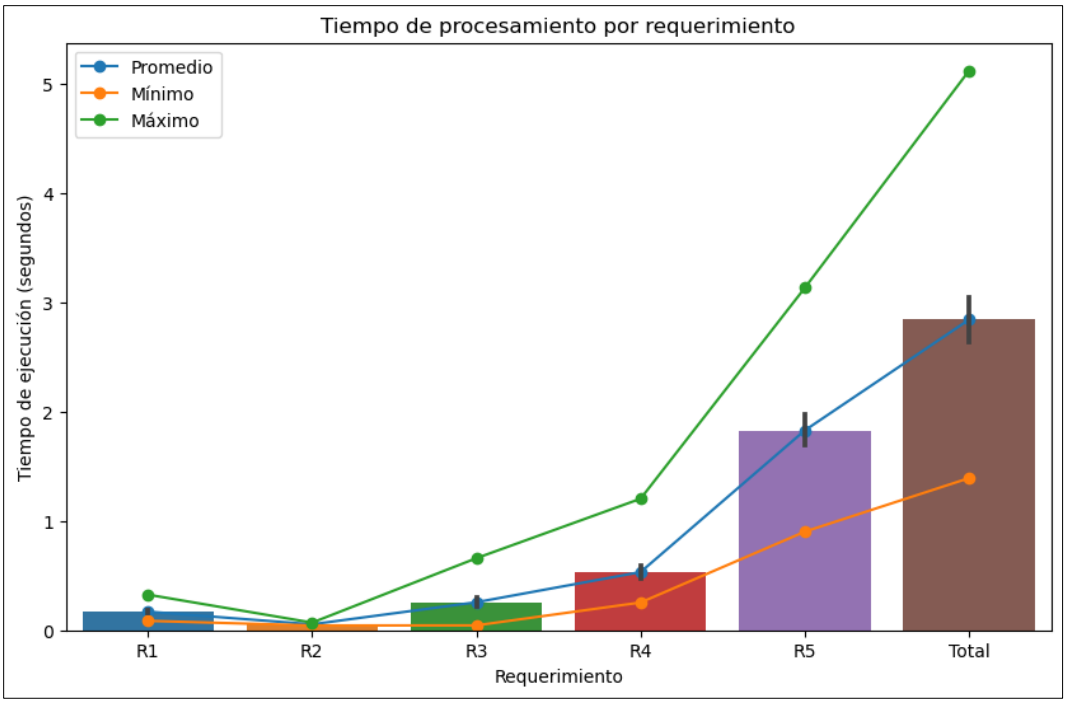
- Procesador: Ryzen 5500U
- Memoria RAM: 16 GB
- SSD

Se tomaron los valores de los parámetros de la siguiente manera:

- type_data_values: [“small”, “medium”, “large”]
- varN_values: [5, 10, 20]
- varPais_values: [“PL”, “ES”]
- varNivel_values: [“junior”, “mid”, “senior”]
- varFechaIni_values: [“2021-11-01”]
- varFechaFinal_values: [“2022-11-01”, “2023-11-01”]

```
1 df.describe()
```

	Tiempo de ejecución (segundos)
count	648.000000
mean	0.947666
std	1.167491
min	0.043807
25%	0.083146
50%	0.300071
75%	1.430981
max	5.121049



Se obtuvo un total de 648 tiempos, divididos entre los requerimientos 1, 2, 3, 4, 5 y el tiempo total.

Comparto los resultados en un archivo llamado: **PruebaRendimiento.xlsx**

Resultados

Los resultados de las pruebas son los siguientes:

Requerimiento 1

Tiempo promedio: 0.17 segundos

Tiempo mínimo: 0.09 segundos

Tiempo máximo: 0.33 segundos

Requerimiento 2

Tiempo promedio: 0.05 segundos

Tiempo mínimo: 0.04 segundos

Tiempo máximo: 0.07 segundos

Requerimiento 3

Tiempo promedio: 0.26 segundos

Tiempo mínimo: 0.04 segundos

Tiempo máximo: 0.66 segundos

Requerimiento 4

Tiempo promedio: 0.53 segundos

Tiempo mínimo: 0.25 segundos

Tiempo máximo: 1.20 segundos

Requerimiento 5

Tiempo promedio: 1.83 segundos

Tiempo mínimo: 0.90 segundos

Tiempo máximo: 3.13 segundos

Total

Tiempo promedio: 2.84 segundos

Tiempo mínimo: 1.39 segundos

Tiempo máximo: 5.12 segundos

Análisis de Resultados

Los resultados obtenidos en las pruebas de tiempo de ejecución son consistentes con el análisis de complejidad realizado. Como se esperaba, el tiempo de ejecución aumenta con el número de ofertas de trabajo. Sin embargo, el aumento no es lineal debido a la complejidad logarítmica de algunas operaciones, como la selección de las N ofertas más recientes y el ordenamiento de ofertas.

Es importante tener en cuenta que estos tiempos de ejecución son específicos para la configuración de hardware utilizada en las pruebas y pueden variar en otras configuraciones. Además, el rendimiento real puede ser afectado por otros factores, como la implementación específica de las funciones utilizadas y la distribución de los datos.