

Taller Dos Estructura De Datos

Brayan Steven Hernandez Fonseca
Daniel Esteban Torres Triana
Lisbeth Tatiana Quilindo Patiño
Fundación Universitaria Konrad Lorenz

Resumen—El informe evaluó la ejecución de dos códigos, uno en Python y otro en Java, en tres máquinas diferentes. Se destacaron diferencias significativas en el tiempo de ejecución y el uso de memoria, en gran parte atribuibles a las especificaciones de hardware de cada máquina. Se observó que las máquinas con hardware más potente tuvieron un rendimiento superior. Además, se notaron pequeñas variaciones relacionadas con el sistema operativo. Se enfatizó la importancia de evaluar tanto el tiempo como el espacio en la optimización de códigos. .

I. INTRODUCCIÓN

El presente informe aborda el análisis de dos códigos desarrollados en Python y Java, respectivamente, con el objetivo de comprender y evaluar su rendimiento en términos de espacio temporal y espacial. Estos códigos se han diseñado para llevar a cabo tareas específicas: el código Python se encarga de generar datos ficticios y medir el tiempo de ejecución y el uso de memoria, mientras que el código Java realiza un proceso de Extracción, Transformación y Carga (ETL) en datos contenidos en un archivo CSV.

Este análisis se ha llevado a cabo por tres personas en tres computadoras distintas, lo que permitirá comparar y contrastar los resultados obtenidos en diferentes entornos de ejecución. El objetivo principal de este informe es proporcionar una visión detallada de la ejecución de los códigos en términos de tiempo y recursos utilizados, así como analizar las diferencias observadas entre las diferentes ejecuciones.

A lo largo de este informe, se describirá en detalle cada uno de los códigos, se presentarán los resultados de las ejecuciones realizadas en diferentes máquinas y se analizarán las posibles implicaciones de estas diferencias en el rendimiento de los códigos. Además, se destacará la importancia de evaluar el espacio temporal y espacial en el desarrollo de aplicaciones.

II. RESULTADOS

II-A. Características de las máquinas utilizadas

Las características de los equipos utilizados en el estudio representan un entorno único en términos de hardware y sistema operativo, lo que influyó en los resultados de rendimiento de los códigos evaluados. Comprender estas características es fundamental para contextualizar las diferencias observadas en los tiempos de ejecución y el uso de memoria, lo que permite una evaluación más completa del impacto de cada factor en el rendimiento de las aplicaciones. A continuación, se detallan las especificaciones de cada máquina y su relevancia en el análisis de resultados:

NOMBRES	INFORMACIÓN DE LA MÁQUINA
Tatiana Quilindo	HP RYZEN 5 BIOS: F.27 64 bits 16,384 MB RAM Windows 11
Daniel Torres	HP Core i5 BIOS: F.41 64 bits 8192 MB RAM Windows 11
Brayan Hernandez	HP Core i5 BIOS: F.58 64 bits 8192 MB RAM Windows 11 Pro

II-B. Espacio temporal y espacial

Esta sección presenta una tabla resumida con los resultados de tiempo de ejecución y uso de memoria de los códigos Python y Java en tres máquinas diferentes. La evaluación de espacio-temporal destaca cómo las diferencias en el hardware y el sistema operativo de las máquinas afectaron el rendimiento de las aplicaciones. Los resultados proporcionan información clave para futuras optimizaciones y análisis de rendimiento.

JAVA			
TIEMPO	2° TIEMPO	3° TIEMPO	TOTAL
Tatiana Quilindo			
800 ms	800 ms	1 segundo	2,segs 600 ms
megabytes			
0.000360473	0.000360766	0.000319477	1.040716000
Daniel Torres			
1 sec,200 ms	1 sec,200 ms	1 sec,500 ms	3,segs,900 ms
megabytes			
0.000365771	0.000365047	0.000365377	1.14995000
Brayan Hernandez			
2 segundos	1 sec,500 ms	2 segundos	5, sec 500 ms
megabytes			
0.000392061	0.000392512	0.000345272	1.129845000

PYTHON			
TIEMPO	2° TIEMPO	3° TIEMPO	TOTAL
Tatiana Quilindo			
36,31768513	35,86377637	39,66580343	37,28242164
megabytes			
48.49664	81.928192	62.746624	64.39048533
Daniel Torres			
109,0757642	114,9211199	61,31301403	285,3098981
megabytes			
157.24211875	170.546875	112.43359375	440.2225875
Brayan Hernandez			
53.3938336	58.45319676	43,91012763	155.7583068
megabytes			
93.3203125	1141.8867187	38.2265625	190.0000718

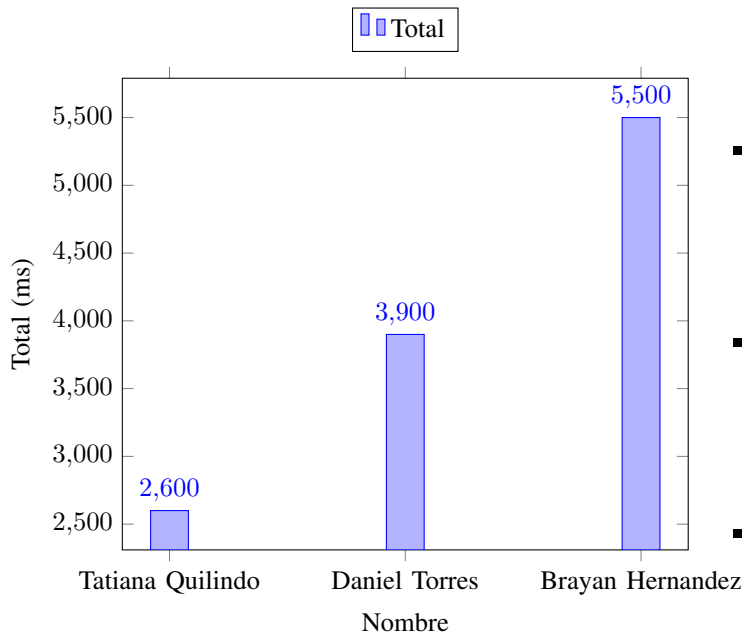


Figura 1. Gráfico de tiempos java

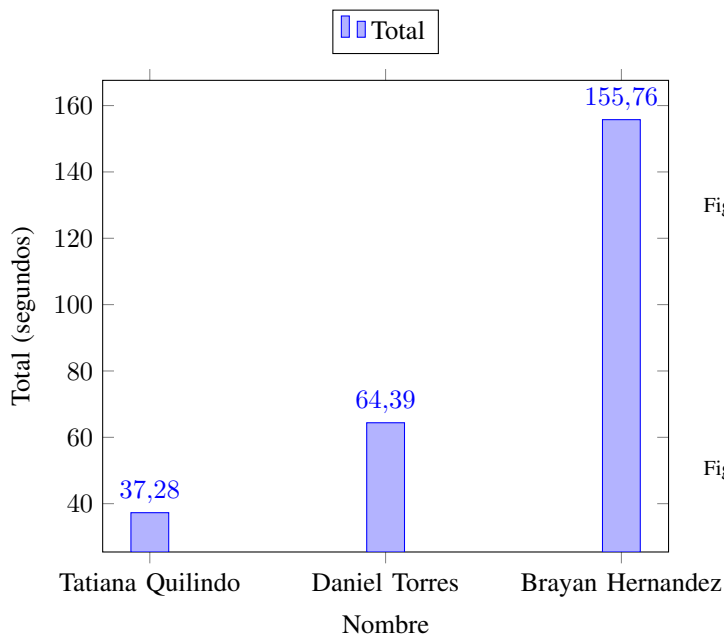


Figura 2. Gráfico de tiempos Python

- En la Máquina 1 (Tatiana Quilindo), la ejecución 1 del código Python tomó aproximadamente 36.31768513 segundos, mientras que la ejecución 1 del código Java se completó en aproximadamente 800 ms segundos.
- En la Máquina 2 (Daniel Torres), la ejecución 1 del código Python tomó aproximadamente 109.0757642 segundos, mientras que la ejecución 1 del código Java se completó en aproximadamente 1 sec, 200 ms.
- En la Máquina 3 (Brayan Hernández), la ejecución 1 del código Python tomó aproximadamente

53.39383363723755 segundos, mientras que la ejecución 1 del código Java se completó en aproximadamente 2 segundos.

- En la Máquina 1 (Tatiana Quilindo), la ejecución 2 del código Python tomó aproximadamente 35,8637763748169 segundos, mientras que la ejecución 2 del código Java se completó en aproximadamente 800 ms segundos.
- En la Máquina 2 (Daniel Torres), la ejecución 2 del código Python tomó aproximadamente 114,921119928359 segundos, mientras que la ejecución 2 del código Java se completó en aproximadamente 1 sec,200 ms.
- En la Máquina 3 (Brayan Hernández), la ejecución 2 del código Python tomó aproximadamente 58.45319676399231 segundos, mientras que la ejecución 2 del código Java se completó en aproximadamente 1 sec,500 ms.

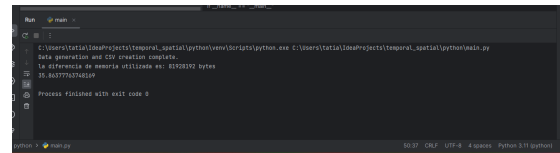


Figura 3. Maquina de Tatiana en Python.

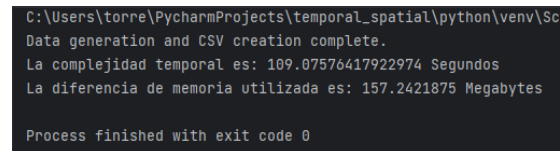


Figura 4. Maquina de Daniel en Python.

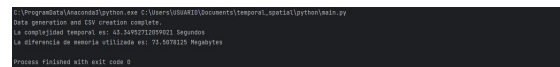


Figura 5. Maquina de Brayan en Python.

- En la Máquina 1 (Tatiana Quilindo), la ejecución 3 del código Python tomó aproximadamente 39,6658034324645 segundos, mientras que la ejecución 3 del código Java se completó en aproximadamente 1 segundo.
- En la Máquina 2 (Daniel Torres), la ejecución 3 del código Python tomó aproximadamente 61,3130140304565 segundos, mientras que la ejecución 3 del código Java se completó en aproximadamente 3 segc,900 ms.
- En la Máquina 3 (Brayan Hernández), la ejecución 3 del código Python tomó aproximadamente

43.91012763977051 segundos, mientras que la ejecución 3 del código Java se completó en aproximadamente 5, sec 500 ms.

En el proceso de ejecución de los códigos Python y Java en las tres máquinas respectivas, se identificaron varios factores que influyeron en las diferencias observadas en los resultados de tiempo de ejecución y uso de memoria:

PC 1: Tatiana Quilindo

El PC de Tatiana Quilindo, equipado con un procesador RYZEN 5 y 16,384 MB de RAM, presenta un hardware más robusto en comparación con las otras dos máquinas. Esto podría haber contribuido a tiempos de ejecución más cortos y un uso de memoria más eficiente. Además, utiliza Windows 11, un sistema operativo moderno que puede ofrecer una mejor administración de recursos en comparación con versiones anteriores. Conforme a las especificaciones de la máquina, tuvo menor cantidad de procesos en segundo plano, lo que podría haber influido en el rendimiento.

PC 2: Daniel Torres

El PC de Daniel Torres cuenta con un procesador Core i5 de 11 generación y 8192 MB de RAM. Aunque es sólida, sus especificaciones son ligeramente menos potentes que las de Tatiana. Esto podría haber resultado en tiempos de ejecución ligeramente más largos y un uso de memoria moderado. Al igual que Tatiana, Daniel utiliza Windows 11, lo que proporciona un entorno moderno y eficiente para la ejecución de aplicaciones.

PC 3: Brayan Hernandez

El PC de Brayan Hernandez tiene 8192 MB de RAM y un procesador Core i5 de 8 generación. Estas especificaciones sugieren un rendimiento comparable en términos de tiempo de ejecución y uso de memoria. Brayan utiliza Windows 11 Pro, que, si bien ofrece características adicionales en comparación con Windows 11 estándar, no debería haber influido significativamente en los resultados.

En resumen, las diferencias en las especificaciones de hardware, como la cantidad de RAM y el tipo de procesador, pueden haber sido el factor principal que contribuyó a las variaciones en el tiempo de ejecución y el uso de memoria.

II-C. Descripción de las partes de los códigos

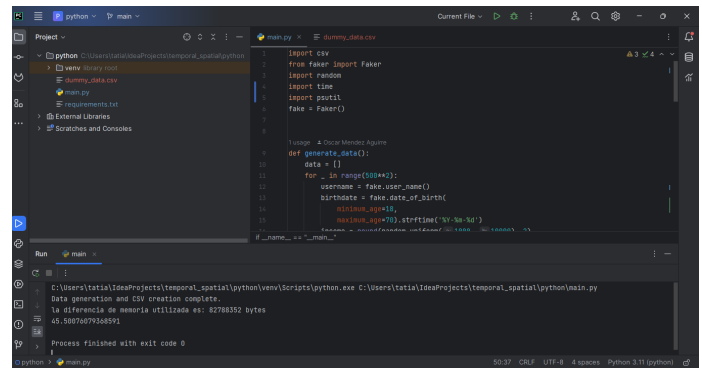


Figura 6. Código Python

El código Python se centra en la generación de datos ficticios y la medición del tiempo de ejecución y el uso de memoria durante el proceso: Utiliza la biblioteca "Faker" para generar datos ficticios, incluyendo nombres de usuario, fechas de nacimiento, ingresos, deudas, género, número de hijos y países. Estos datos se almacenan en una lista. Se inicia un temporizador antes de la generación de datos y se detiene después de completarse. El tiempo transcurrido se calcula y muestra en segundos. Esto permite evaluar el rendimiento temporal del proceso.

Se registra el uso de memoria antes y después de la generación de datos utilizando la biblioteca "psutil". La diferencia se calcula y muestra en megabytes, proporcionando información sobre cómo se utiliza la memoria durante la ejecución. Los datos generados se escriben en un archivo CSV con encabezados.

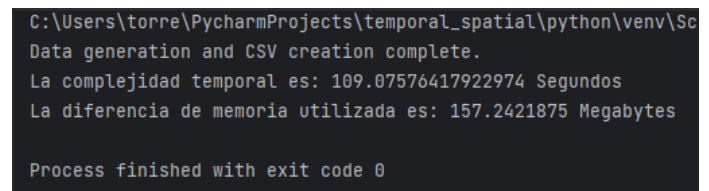


Figura 7. resultado del tiempo y memoria en megabytes del código Python

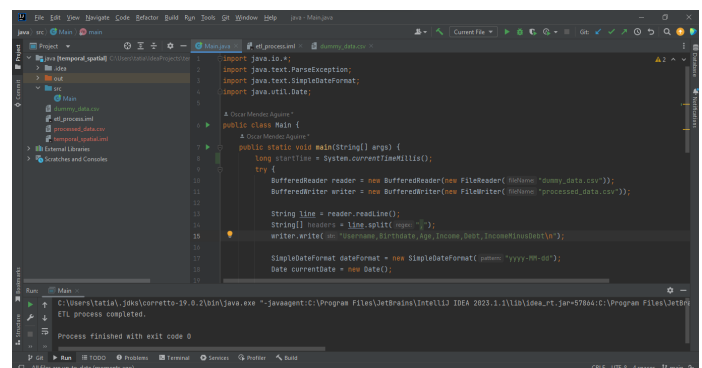


Figura 8. Código Java.

El código Java se encarga de realizar un proceso de Extracción, Transformación y Carga (ETL) en un archivo CSV de entrada llamado 'dummy data csv'.

El código utiliza clases de manejo de archivos para leer los datos del archivo CSV de entrada y escribir los datos transformados en un nuevo archivo CSV llamado 'processed data csv'. Este proceso facilita la transformación de datos y la preservación de los resultados.

Durante la lectura de cada línea del archivo CSV de entrada, se realizan diversas transformaciones en los datos. Esto incluye el cálculo de la edad a partir de la fecha de nacimiento y la diferencia entre el ingreso y la deuda para calcular 'Income-MinusDebt'. Los datos transformados se agregan a un nuevo archivo CSV. El código utiliza la clase SimpleDateFormat para manejar las fechas de nacimiento en el formato adecuado y calcular la edad de las personas.

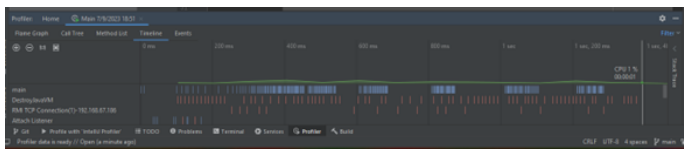


Figura 9. Resultado tiempo código Java.

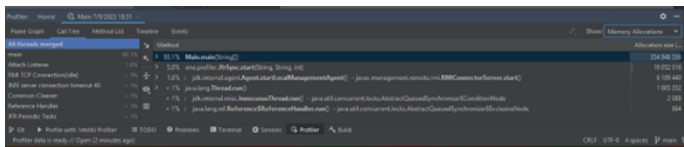


Figura 10. resultado memoria en megabytes código Java

III. CONCLUSIONES

- Observamos diferencias en el tiempo de ejecución y el uso de memoria en las tres máquinas, principalmente debido a las especificaciones de hardware.
- Las máquinas con hardware más potente mostraron mejor rendimiento, mientras que las menos potentes tuvieron tiempos de ejecución más largos y mayor uso de memoria.
- Aunque todas las máquinas usaban Windows 11, hubo ligeras variaciones, lo que sugiere que el sistema operativo puede influir en el rendimiento.
- Este ejercicio resalta la importancia de evaluar tanto el tiempo como el espacio de un código para asegurar su rendimiento.