Entregado por: Víctor Mora - Dennys Mosquera
2020-2

DISEÑO DE EXPERIMENTOS
Informe

Link repositorio: https://github.com/dennvm09/ProyectoFinal-ArquiHard

1. Identificación del problema

Con este experimento se desea analizar y emitir un juicio sobre aspectos del comportamiento o desempeño de un sistema de cómputo aplicando la metodología de conducción de experimentos y la teoría de la arquitectura y organización de computadores. Específicamente, cuantificar y explicar el impacto en el desempeño que tienen 5 versiones equivalentes de algoritmos que varían en la localidad espacial al recorrer sus pixeles de una imagen. También se desea estudiar el impacto del tamaño de las imágenes en el tiempo de respuesta normalizado en la ejecución de la operación de invertir el color y el impacto que tiene el sistema de memoria del sistema. También sería interesante comparar dos lenguajes de programación distintos.

2. Identificación de factores

Los siguientes son los factores que afectan de manera directa nuestra variable respuesta. Para este experimento se debían hacer las mismas ejecuciones en distintos equipos, para evitar una gran variabilidad en los datos, se optó por tener un ambiente “limpio”, es decir, solo se tenía en ejecución el IDE y el archivo en Excel para el registro de los datos. Asimismo, se decidió tener ambos equipos conectados a la energía.

Factores Primarios:

- **Tamaño de imagen (en pixeles):**
  - 400 x 400 px
  - 700 x 700 px
  - 1000 x 1000 px
  - 1300 x 1300 px
  - 1600 x 1600 px
  - 1900 x 1900 px
  - 2200 x 2200 px
  - 2500 x 2500 px
- **Lenguaje de programación:**
  - Java.
  - C#.
- **Algoritmo.**
- **Tiempo de ejecución (medido en ms).**

Factores Secundarios:

- **Sistema operativo:**
  - Windows
- **Equipo**

- Una manera de reducir los efectos del equipo, (que en nuestro caso fueron computadores portátiles), es tenerlos conectados directamente a la energía durante la duración del experimento.

- **Ejecuciones internas del equipo**
  - Sus efectos pueden controlarse teniendo la menor cantidad de procesos ejecutándose paralelamente con el experimento.
- **IDE's:**
  - Visual Studio 2019
  - Eclipse

  Realmente no se tiene certeza de que el uso de diferentes IDE's genera recursos adicionales que resultan afectando el tiempo de ejecución de cada algoritmo.

3. **Experimentación preliminar**

   Antes de tomar los datos finales, se realizaron ejecuciones para conocer el comportamiento de los algoritmos. Esto ayudó a que se modificaran los tamaños de algunas imágenes, debido a que en un principio se tenían tamaños muy pequeños, que a la final resultaban generando ruido. También, se logró identificar el efecto que generaba tener el computador conectado a la energía; cuál sería el tamaño de la muestra; y el orden y/o combinaciones que se realizarían para la toma de datos final.

   a. **Diseño experimental:** Usaremos el ***diseño factorial completo*** pues utilizaremos todos los factores y sus niveles.

4. **Experimentar**

   El experimento fue realizado a través de la consola de Windows y con la mínima cantidad de aplicaciones corriendo en segundo plano, de manera que esto no afectara el desarrollo del experimento. Como resultado, el experimento transcurrió sin contratiempos y de la manera como se esperaba que se desarrollara.

   Los datos obtenidos fueron consignados en una hoja de Excel con el encabezado que se ve en la ***ilustración 1.*** Es válido dar a conocer que por cada algoritmo se hicieron 3 repeticiones por tamaño de imagen. Para un total de 480 datos registrados, dado que se utilizaron dos equipos diferentes. Al sacar la varianza, su resultado da a entender que no hay una gran variabilidad entre datos, por lo cual no fue necesario tomar más muestras.

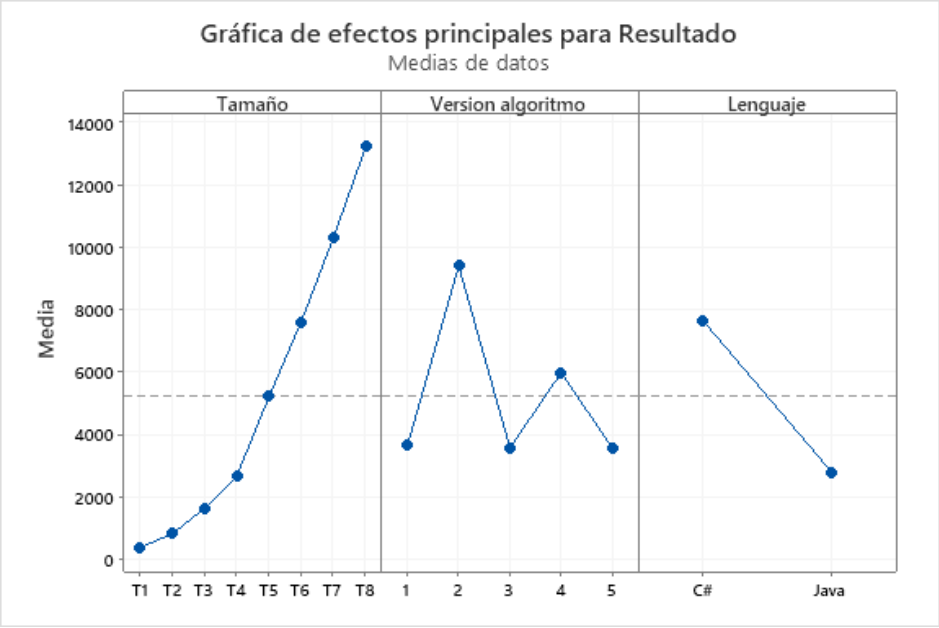
Equipo	Lenguaje	Tamaño	Version algoritmo	Resultados(milisegundos)			Resultados normalizados		
				r1	r2	r3	r1	r2	r3

Ilustración 1. Encabezado para la tabla del registro de datos obtenidos.

5. **Análisis de datos**

   Para el análisis de datos, se hizo uso de Minitab. Los datos fueron apilados por número de repetición y se hizo una distinción entre equipos, es decir, se hizo

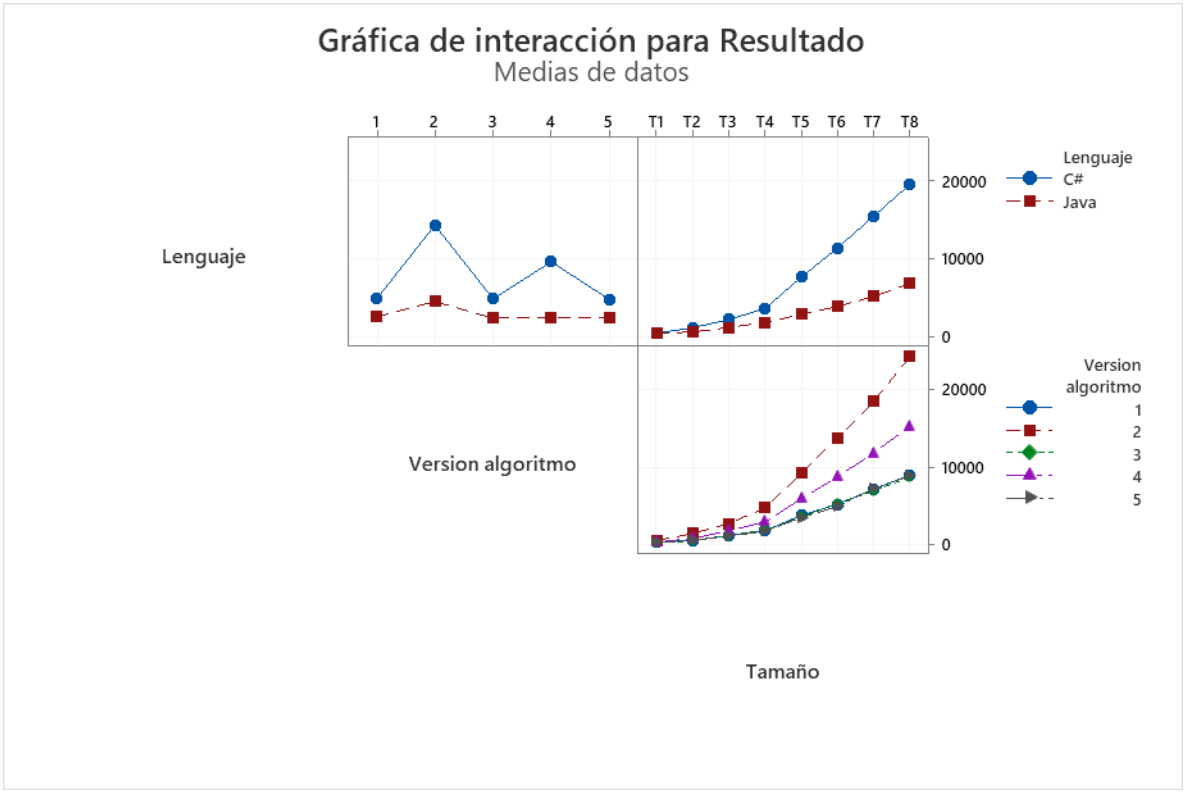
primero en el Equipo 1 y luego en el Equipo 2. Las gráficas obtenidas se pueden ver a continuación,



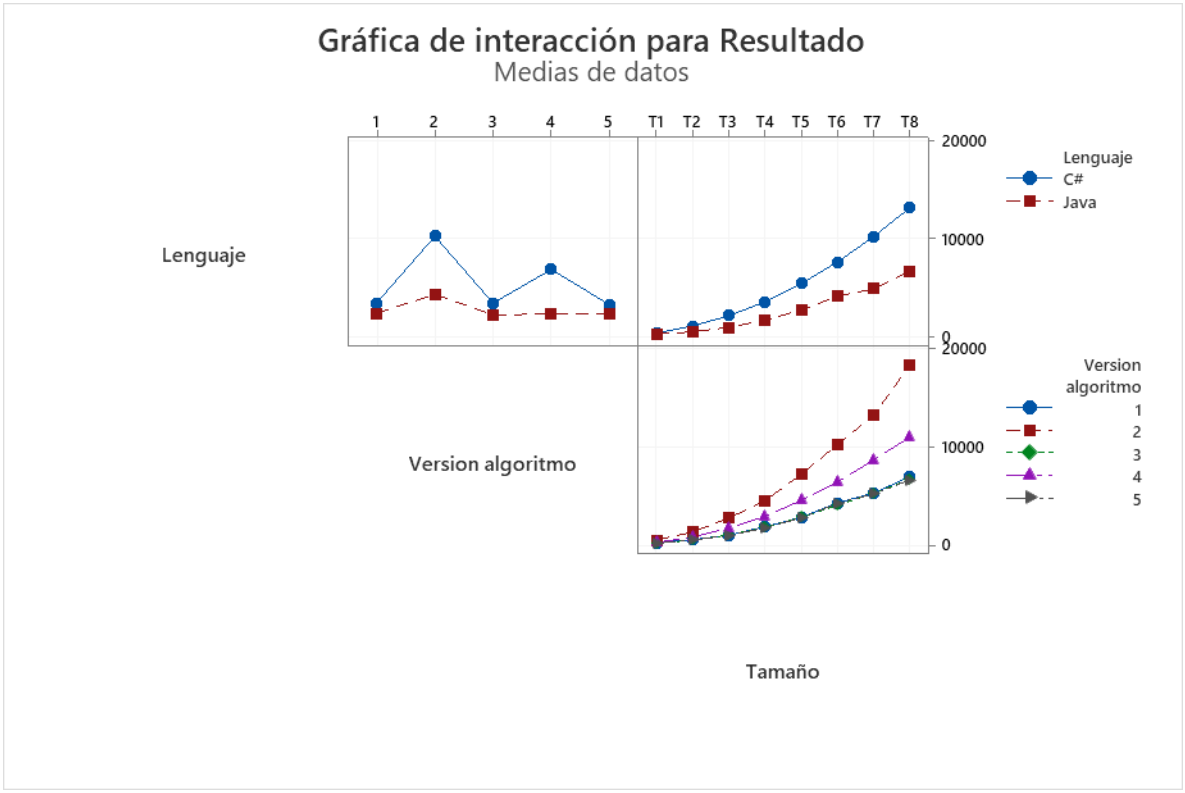
Gráfica 1. Gráfica efectos principales Equipo 1.

Gráfica 2. Gráfica efectos principales Equipo 2.

Con las gráficas ***Gráfica 1 y Gráfica 2*** se puede evidenciar que en el análisis individual de las variables no hubo mayor diferencia entre el equipo 1 y equipo 2. De igual forma, se logra apreciar que a medida que el tamaño de la imagen aumenta, el tiempo también lo hace.



Gráfica 3. Gráfica de interacción Equipo 1.



Gráfica 4. Gráfica de interacción Equipo 2.

Con estas gráficas ***Gráfica 3 y Gráfica 4*** se puede observar la interacción que hay entre los diferentes factores primarios y como se dijo anteriormente, no hay gran distinción entre el Equipo 1 y el Equipo 2. Si se mira la gráfica Tamaño vs Versión algoritmo, se nota como los algoritmos 1, 3 y 5 se extrapolan, y casi que tienen un comportamiento igual, siendo estos los algoritmos con menor tiempo. Hay que mencionar, además que independientemente del algoritmo, los tiempos fueron menores con el lenguaje de programación Java, y esto se puede deber a que el IDE Visual Studio 2019, donde se ejecutó el programa en lenguaje C#, consuma mayores recursos al momento de realizar una ejecución, sin embargo, no se tiene certeza de eso.

Lo anterior, también se ve reflejado con el método de Tukey en la ***Tabla 1,*** donde con un nivel de confianza del 95% muestra que el lenguaje C# es el más demorado. Ahora bien, también coincide en el análisis que se hizo con las gráficas ***Gráfica 1 y Gráfica 2,*** donde se decía que la versión 2 de los algoritmos es el que toma mayor tiempo de ejecución. Entonces, este versus entre la versión del algoritmo y el lenguaje de programación nos arroja que la combinación entre el algoritmo 2 y el lenguaje C# es quien toma mayor tiempo de ejecución, y la combinación entre el algoritmo 3 y el lenguaje Java es el de menor tiempo.

Comparaciones por parejas de Tukey: Version algoritmo*Lenguaje

Agrupar información utilizando el método de Tukey y una confianza de 95%

Version algoritmo*Lenguaje	N	Media	Agrupación			
2 C#	24	14328,1	A			
4 C#	24	9601,4		B		
3 C#	24	4826,0			C	
1 C#	24	4825,7			C	
5 C#	24	4748,8			C	
2 Java	24	4478,5				D
1 Java	24	2497,9					E
5 Java	24	2367,2					E
4 Java	24	2356,5					E
3 Java	24	2317,8					E

Las medias que no comparten una letra son significativamente diferentes.

*Tabla 1. Comparación por parejas de Tukey: Versión algoritmo*Lenguaje*

Comparaciones por parejas de Fisher: Lenguaje

Agrupar información utilizando el método LSD de Fisher y una confianza de 95%

Lenguaje	N	Media	Agrupación	
C#	120	7665,99	A	
Java	120	2803,57		B

Las medias que no comparten una letra son significativamente diferentes.

Tabla 2. Comparación por parejas Fisher: Lenguaje

Comparaciones por parejas de Tukey: Lenguaje

Agrupar información utilizando el método de Tukey y una confianza de 95%

Lenguaje	N	Media	Agrupación
C#	120	7665,99	A
Java	120	2803,57	B

Las medias que no comparten una letra son significativamente diferentes.

Tabla 3. Comparación por parejas Tukey: Lenguaje

6. Conclusiones

Luego de realizar el experimento se puede afirmar que, por un lado, la hipótesis de que a medida que el tamaño de la imagen aumenta también aumenta el tiempo de ejecución está en lo cierto, y esto pasa independientemente de la versión del algoritmo. Por otro lado, los resultados permiten deducir que los algoritmos 1, 3 y 5, tienen un comportamiento similar, y resultan ser los más rápidos en ejecución, esto puede ser debido a la manera en que se accede a cada uno de los pixeles de la imagen, donde utilizan solo dos loops para hacer la reversión de colores. En cambio, los algoritmos 2 y 4 utilizan loops por cada uno de los colores, ocasionando que el tiempo de ejecución sea mayor.

También, se pudo comprobar que sí existe una variación en el tiempo cuando se ejecuta el programa teniendo el computador conectado a la energía, sin embargo, sigue existiendo una diferencia significativa con los diferentes leguajes de programación, siendo Java mucho más rápido que C#, y esto puede ser debido a que el IDE utilizado para ejecutar C# consume mayores recursos a la hora de la ejecución. Lo anterior se puede deducir observando los resultados de las tablas ***Tabla 2 y Tabla 3.***

Por último, se puede concluir que la similitud de resultados en ambos equipos se puede deber a que ambos se ejecutaron en ambientes con la mínima cantidad de recursos ejecutándose y conectados a la energía. De igual manera, pudo ser porque los dos equipos cuentan con la misma capacidad de memoria y procesador. También pudo ser porque el experimento se realizó a la misma vez, es decir mismo día y misma hora, y la temperatura y hora pueden ser factores que influyen indirectamente en el funcionamiento del equipo, afectando el resultado del experimento. A la final, detectamos que hay distintas variables no controlables, que afectan el desempeño a la hora de la ejecución, dando una variabilidad en los resultados.