Práctica: Recocido Simulado

Integrantes:

Ventura Guevara Carlos Martin

201909842

Luna Romero Emmanuel

2019

Vélez López Morelia Del Carmen

201945081

Titular del curso: Rodríguez C Lino Alberto

Periodo: Primavera 202

Introducción

Se ha realizado un programa que busca y devuelve el valor mínimo posible, utilizando como estrategia de búsqueda el algoritmo de recocido simulado.

Para este caso en particular se realizaron 2 versiones distintas del programa. Esto debido a que se solicitaba la resolución de 2 problemas que involucraban valores de entrada totalmente diferentes en comparación (uno para grafos, y otro para funciones de 3 dimensiones).

A pesar de tener 2 versiones de este programa, todo lo que se refiere al aspecto del "recocido simulado", se encuentra prácticamente idéntico en ambas.

Es importante destacar que se han proporcionado 3 ejecutables, y de los mismos se extraerá la información que los programas analizaran, siendo que el programa pensado para analizar grafos investigará uno de los 3 ejecutables otorgados (my_graph1.exe). Mientras que el programa pensado para analizar funciones tridimensionales investigará los restantes 2 ejecutables (my_func4.exe y my_func5.exe).

Desarrollo

A continuación, se comenzará a profundizar en el comportamiento y resultados obtenidos para cada caso:

Primera versión del programa (pensada para grafos):

La primera versión dispone de múltiples características diseñadas para que pueda llevar a cabo su cometido, sin embargo, se destacan las siguientes:

1. El vecindario es restringido en función del número de consultas que realizará el algoritmo y del número de vértices en el grafo.
    a. Por ejemplo: si se consulta 1000 veces el resultado del ejecutable "my_graph1.exe", y el grafo consta de 20 vértices, entonces el vecindario se restringirá en 1 vértice, por cada 1000 DIV 20 ejecuciones.
    b. Como curiosidad, es posible reducir el número de vértices que se usaran para efectuar la división entera, modificando internamente el valor de la variable "domFinal" (admitiendo valores enteros entre 1 y 19).

2. La temperatura puede ser decrementada a través del uso 4 diferentes funciones:

   a. Lineal:

   $$T = T - 1$$

   b. Exponencial:

   $$T = (T)(0.95)$$

   c. Logarítmica

   $$T = \frac{1}{(1 + Ln(i))}$$

   d. Geométrica

   $$T = 0.11^{\frac{1}{i+1}}$$

   i es el núm. de iteraciones hechas.
   Los valores como 0.95 y 0.11 fueron hallados experimentalmente.

3. La función de aceptación es la función de Metrópolis.

Resultados:

Al ejecutar el código en un total de 10 ocasiones y recopilar la información obtenida para realizar una comparativa más precisa, se han obtenido los siguientes datos:

Con 1000 ejecuciones y 20 vértices:

Decrecimiento de la temperatura	Lineal	Exponencial	Logarítmica	Geométrica
Resultado de la ejecución 1	4964	4411	6177	4532
Resultado de la ejecución 2	4435	4587	6491	4473
Resultado de la ejecución 3	4760	4457	6152	4615
Resultado de la ejecución 4	4760	3937	5763	4728
Resultado de la ejecución 5	4820	4432	5934	4474
Resultado de la ejecución 6	4803	4510	6313	4720
Resultado de la ejecución 7	4824	4734	4994	4842
Resultado de la ejecución 8	5129	4446	5491	4754
Resultado de la ejecución 9	4908	4659	5468	4221
Resultado de la ejecución 10	4368	4673	6488	4866
Resultado Promedio	4777.1	4484.6	5927.1	4622.5

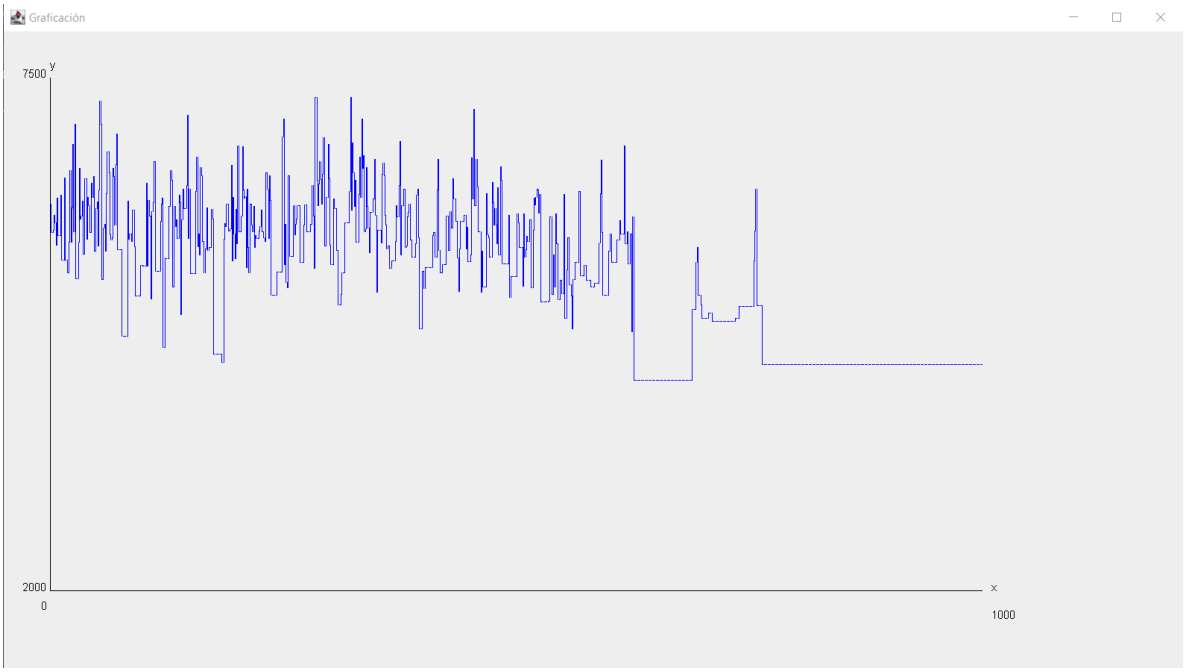
Los resultados son verificables con la siguiente tabla que muestra sus grafos correspondientes:

Decrecimiento de la temperatura	Lineal	Exponencial	Logarítmica	Geométrica
Grafo del Resultado de la ejecución 1	{ 12, 14, 2, 3, 19, 6, 10, 13, 1, 11, 9, 4, 18, 5, 16, 0, 15, 7, 8, 17 }	{ 14, 4, 11, 1, 0, 16, 3, 17, 12, 7, 5, 8, 2, 6, 10, 15, 9, 18, 19, 13 }	{ 2, 9, 6, 12, 13, 16, 18, 11, 4, 5, 8, 10, 0, 7, 3, 14, 15, 19, 1, 17 }	{ 10, 11, 6, 0, 16, 1, 12, 7, 15, 3, 19, 8, 17, 5, 18, 2, 9, 14, 13, 4 }
Grafo del Resultado de la ejecución 2	{ 14, 15, 12, 1, 6, 4, 0, 16, 17, 3, 10, 9, 11, 2, 8, 19, 7, 18, 5, 13 }	{ 12, 8, 2, 18, 9, 14, 19, 13, 3, 5, 7, 6, 0, 4, 15, 10, 1, 11, 16, 17 }	{ 17, 18, 10, 16, 12, 15, 1, 3, 6, 13, 0, 2, 5, 7, 9, 14, 11, 8, 19, 4 }	{ 11, 9, 7, 6, 1, 15, 0, 14, 10, 18, 4, 13, 19, 8, 12, 2, 17, 5, 3, 16 }
Grafo del Resultado de la ejecución 3	{ 3, 13, 7, 8, 18, 2, 6, 9, 10, 5, 16, 0, 15, 14, 1, 4, 11, 19, 12, 17 }	{ 0, 12, 6, 1, 4, 10, 7, 8, 5, 18, 9, 15, 2, 19, 14, 11, 3, 13, 17, 16 }	{ 2, 16, 10, 1, 3, 13, 7, 15, 18, 19, 9, 6, 4, 0, 17, 5, 11, 12, 8, 14 }	{ 12, 9, 5, 13, 1, 6, 18, 15, 14, 19, 16, 17, 4, 0, 7, 3, 2, 8, 10, 11 }
Grafo del Resultado de la ejecución 4	{ 14, 1, 4, 10, 18, 9, 15, 8, 19, 0, 16, 12, 2, 17, 3, 11, 5, 6, 7, 13 }	{ 16, 11, 15, 14, 9, 18, 5, 13, 10, 4, 6, 19, 3, 2, 7, 8, 1, 12, 0, 17 }	{ 11, 13, 3, 14, 5, 15, 0, 9, 17, 10, 1, 18, 8, 19, 7, 4, 2, 6, 12, 16 }	{ 4, 12, 11, 8, 19, 1, 6, 3, 13, 17, 2, 15, 18, 9, 16, 0, 7, 10, 5, 14 }
Grafo del Resultado de la ejecución 5	{ 13, 17, 4, 11, 9, 15, 2, 6, 14, 1, 0, 18, 12, 16, 5, 19, 3, 7, 8, 10 }	{ 19, 0, 4, 6, 10, 1, 14, 16, 18, 5, 13, 2, 12, 9, 15, 11, 7, 8, 17, 3 }	{ 11, 4, 19, 10, 12, 7, 0, 5, 9, 6, 1, 3, 8, 17, 13, 2, 18, 15, 16, 14 }	{ 19, 8, 12, 2, 6, 5, 11, 16, 4, 18, 7, 14, 15, 13, 9, 10, 17, 0, 1, 3 }

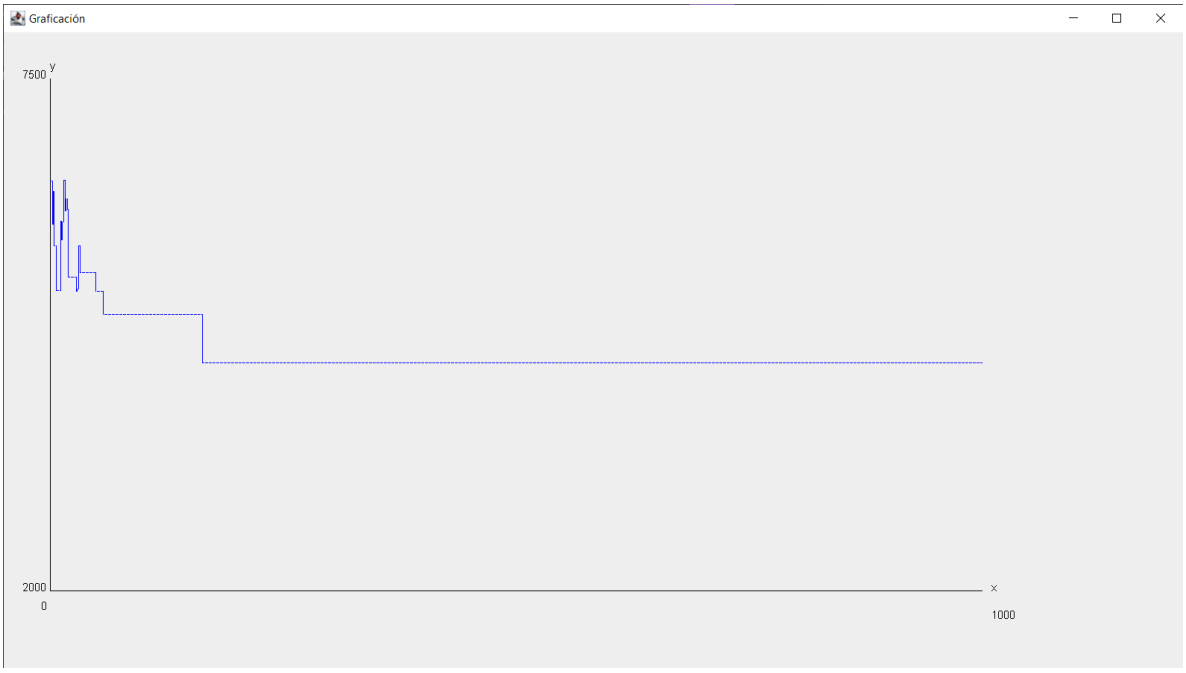
Grafo del Resultado de la ejecución 6	{ 3, 6, 15, 2, 8, 12, 18, 1, 10, 11, 13, 5, 9, 7, 16, 17, 0, 4, 14, 19 }	{ 5, 13, 6, 0, 15, 11, 16, 10, 1, 12, 8, 17, 3, 19, 7, 4, 2, 18, 9, 14 }	{ 19, 6, 2, 10, 0, 14, 16, 8, 7, 17, 15, 12, 9, 18, 13, 3, 5, 4, 11, 1 }	{ 14, 13, 2, 8, 0, 6, 1, 3, 19, 5, 15, 11, 16, 7, 10, 4, 17, 18, 9, 12 }
Grafo del Resultado de la ejecución 7	{ 13, 17, 5, 12, 6, 1, 11, 16, 4, 10, 9, 7, 8, 18, 19, 3, 2, 14, 0, 15 }	{ 10, 17, 11, 1, 0, 15, 6, 2, 3, 19, 13, 4, 16, 12, 8, 7, 9, 18, 14, 5 }	{ 5, 0, 15, 11, 6, 1, 18, 10, 2, 4, 13, 9, 14, 19, 8, 3, 7, 16, 17, 12 }	{ 18, 0, 6, 3, 10, 2, 15, 9, 16, 12, 7, 4, 8, 17, 19, 13, 5, 11, 1, 14 }
Grafo del Resultado de la ejecución 8	{ 10, 16, 6, 0, 7, 5, 19, 2, 17, 12, 8, 18, 4, 14, 9, 15, 13, 1, 11, 3 }	{ 7, 0, 16, 17, 1, 15, 11, 4, 12, 2, 18, 5, 9, 6, 14, 19, 13, 10, 3, 8 }	{ 12, 13, 6, 9, 3, 2, 19, 0, 4, 14, 16, 17, 5, 10, 7, 8, 11, 18, 15, 1 }	{ 5, 0, 17, 19, 13, 16, 11, 4, 12, 18, 10, 14, 9, 15, 2, 7, 8, 6, 1, 3 }
Grafo del Resultado de la ejecución 9	{ 19, 18, 1, 0, 6, 13, 15, 14, 7, 2, 12, 17, 8, 11, 4, 10, 3, 5, 9, 16 }	{ 16, 12, 17, 10, 7, 3, 19, 0, 5, 13, 9, 18, 2, 8, 6, 14, 15, 4, 1, 11 }	{ 9, 5, 15, 12, 0, 13, 4, 6, 1, 10, 11, 16, 2, 14, 8, 19, 7, 17, 3, 18 }	{ 19, 3, 10, 1, 14, 17, 12, 4, 7, 6, 11, 9, 18, 16, 0, 15, 2, 13, 5, 8 }
Grafo del Resultado de la ejecución 10	{ 7, 4, 5, 16, 19, 8, 2, 0, 17, 3, 13, 6, 1, 11, 15, 14, 9, 18, 10, 12 }	{ 5, 18, 8, 7, 9, 10, 13, 15, 6, 2, 12, 14, 17, 16, 19, 3, 0, 1, 4, 11 }	{ 6, 7, 15, 10, 9, 0, 8, 19, 2, 13, 18, 16, 5, 3, 12, 17, 4, 1, 11, 14 }	{ 19, 10, 6, 1, 0, 14, 9, 5, 4, 7, 15, 11, 18, 2, 17, 16, 12, 8, 13, 3 }

Ahora se muestra algunas de las muchas gráficas obtenidas durante las ejecuciones del programa:

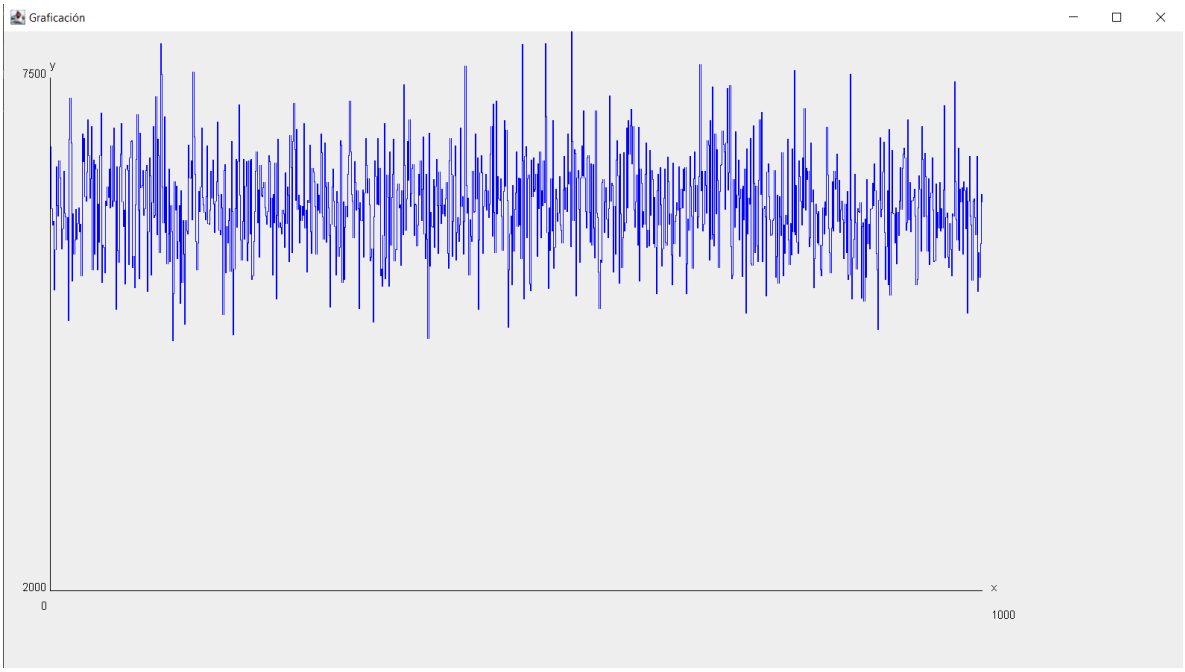
Lineal:



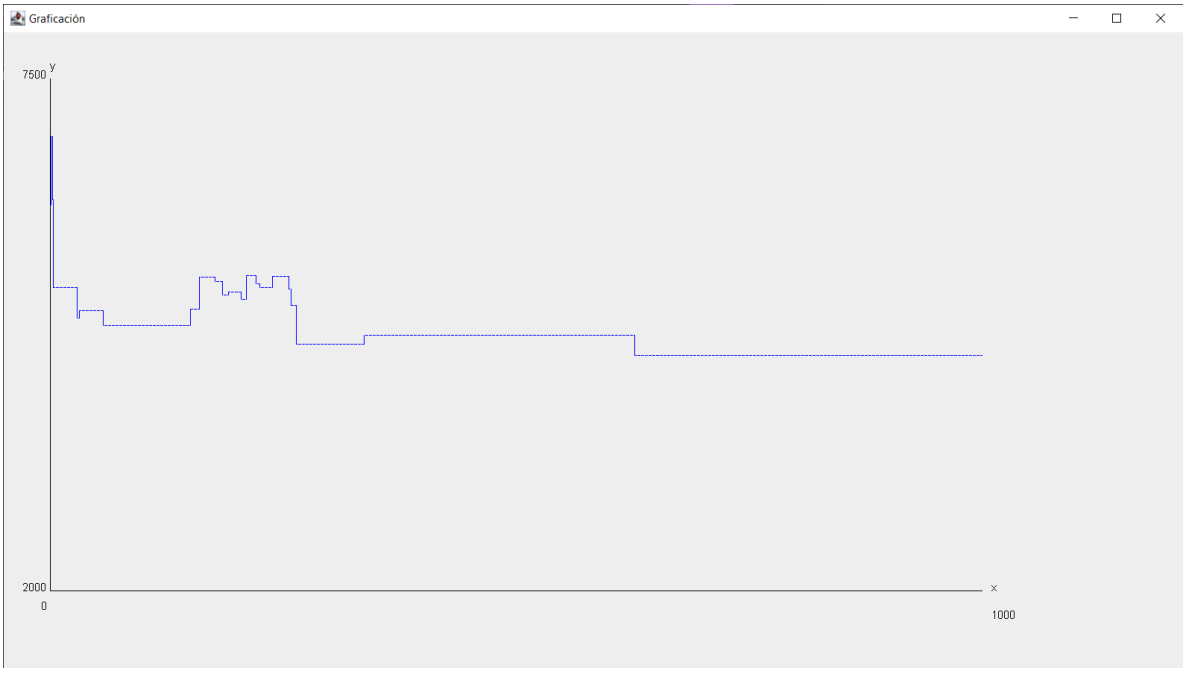
Exponencial:



Logarítmica:



Geométrica:



Segunda versión del programa (pensada para funciones de tres dimensiones):

La segunda versión del programa presenta algunas diferencias conceptuales que fueron necesarias para el correcto funcionamiento de este, sin embargo (y como ya se había mencionado antes), la estructura del recocido simulado es prácticamente idéntica a la que existe en la primera versión.

Enfocándonos ahora en sus características más destacables:

1. El vecindario es restringido de tal manera que "congelamos" el eje x momentáneamente para buscar la mejor solución con el valor de x fijo, pero el valor de y en constante movimiento.
   Por ejemplo: supongamos que ejecutamos el programa con un total de 100 iteraciones, donde 'x' e 'y' se encuentran en un intervalo entre (-25,25). Primero se generará una (x,y) aleatoria (supongamos que obtuvimos (3,8)). Pues es inmediatamente después cuando x se convertirá temporalmente en una constante, mientras que 'y' comienza a adquirir valores aleatorios (supóngase como ejemplos (3,12), (3,7), (3,19)) de modo que 'x' se mantendrá fija, e 'y' cambiará su valor durante 10 consultas.
   En la 11va ejecución, 'x' volverá a cambiar su valor, sin embargo, justo después de esto nuevamente se convertirá en una constante que no cambiará su valor durante otras 10 consultas, mientras que 'y' si lo hará.
   Este proceso se repite hasta que se alcanzan las 100 iteraciones.
2. La temperatura puede ser decrementada a través del uso 4 diferentes funciones.
   a. Lineal:
   $$T = T - 1$$
   b. Exponencial:
   $$T = (T)(0.95)$$
   c. Logarítmica
   $$T = \frac{1}{(1 + Log_{10}(i))}$$
   d. Geométrica
   $$T = 0.12^{\frac{1}{i+1}}$$

   i es el núm. de iteraciones hechas.
   Los valores como 0.95 y 0.12 fueron hallados experimentalmente.

3. La función de aceptación es la función de Metrópolis.

Resultados:

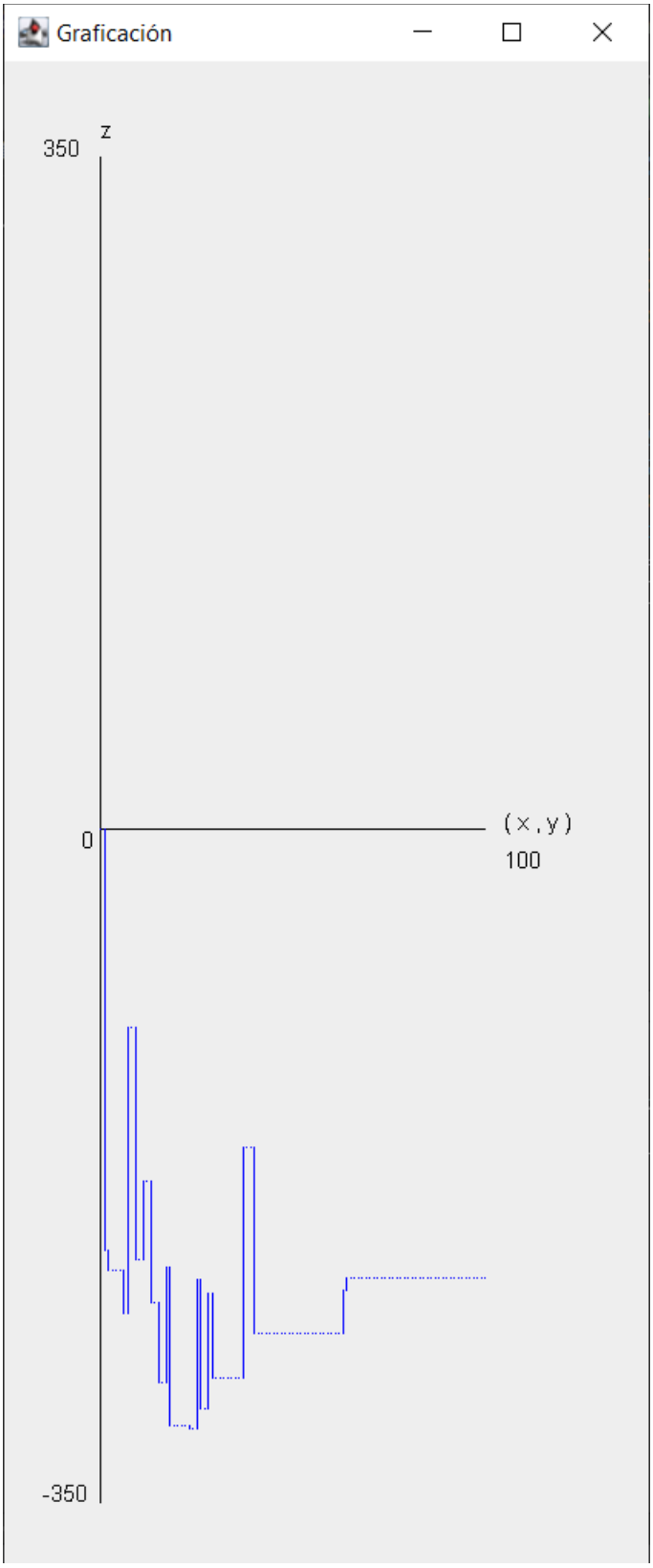
En este caso, se ejecutó el programa para los 2 ejecutables restantes (my_func4.exe y my_func5.exe). Por lo que se mostrarán los resultados obtenidos para cada ejecutable.

En ‘my_func4.exe’ con 100 ejecuciones y con un intervalo de (-25,25) para ‘x’ e ‘y’:

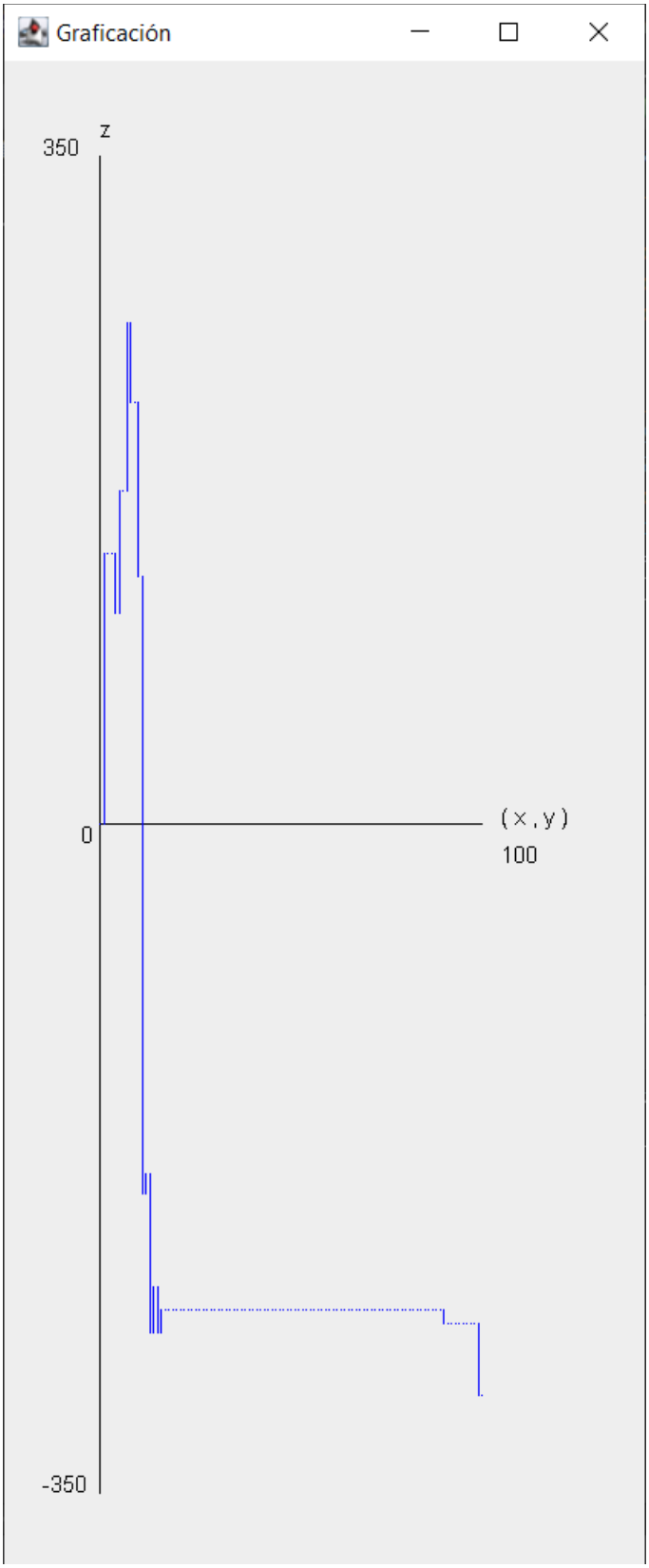
Decrecimiento de la temperatura	Lineal	Exponencial	Logarítmica	Geométrica
Resultado de la ejecución 1	-233.87419	-299.19893	-13.00376	-301.90727
Resultado de la ejecución 2	-296.87375	-285.58608	288.19968	-295.21446
Resultado de la ejecución 3	-279.76962	-306.00461	-256.69384	-293.03968
Resultado de la ejecución 4	-280.9347	-319.96705	75.77309	-299.67439
Resultado de la ejecución 5	-318.84767	-286.57301	47.33789	-293.69044
Resultado de la ejecución 6	-272.95369	-314.26279	170.57158	-317.21099
Resultado de la ejecución 7	-272.77848	-287.04476	360.32083	-309.18306
Resultado de la ejecución 8	-250.40968	-296.00519	504.346	-257.69082
Resultado de la ejecución 9	-311.95058	-290.29838	-171.86098	-300.55134
Resultado de la ejecución 10	-274.66608	-284.14952	-287.47252	-322.97802
Resultado Promedio	-279.305844	-296.909032	71.751797	-299.114047

Ahora se muestra algunas gráficas de su comportamiento durante su ejecución:

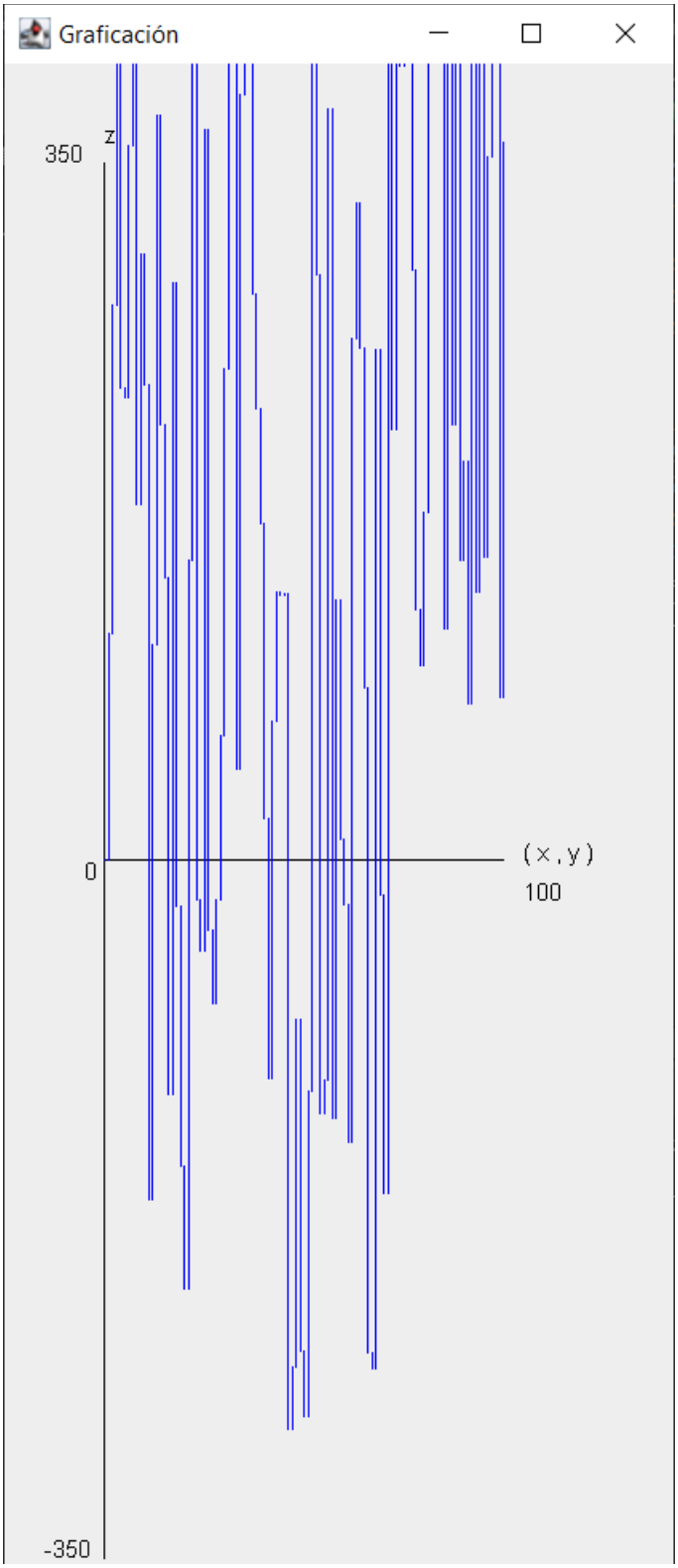
Lineal:



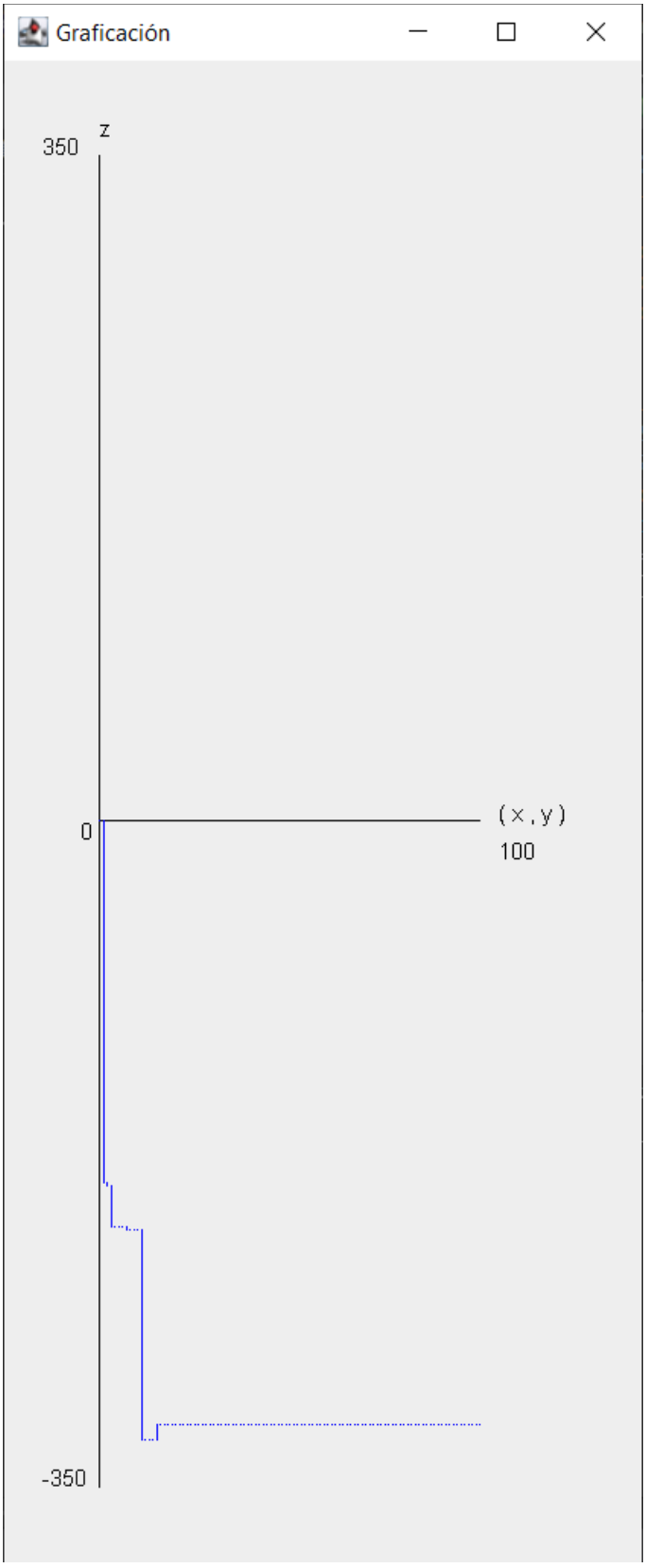
Exponencial:



Logarítmica:



Geométrica:

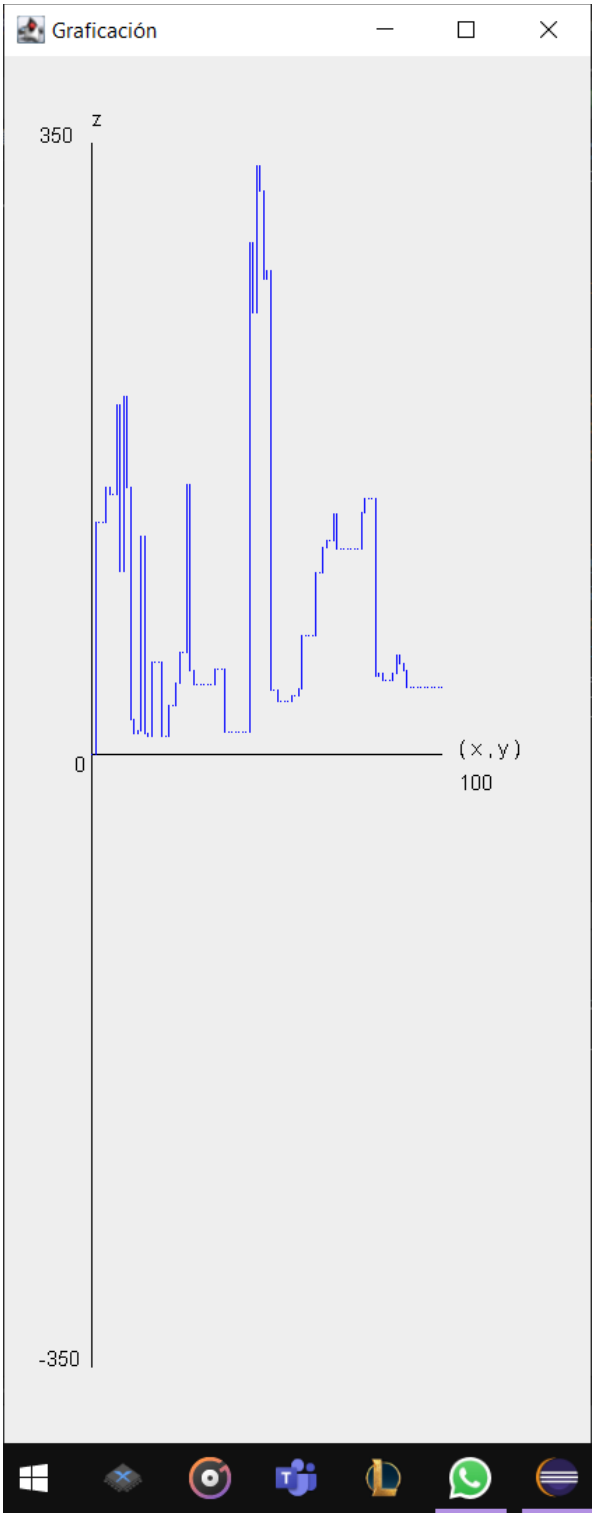


En ‘my_func5.exe’ con 100 ejecuciones y con un intervalo de (-25,25) para ‘x’ e ‘y’:

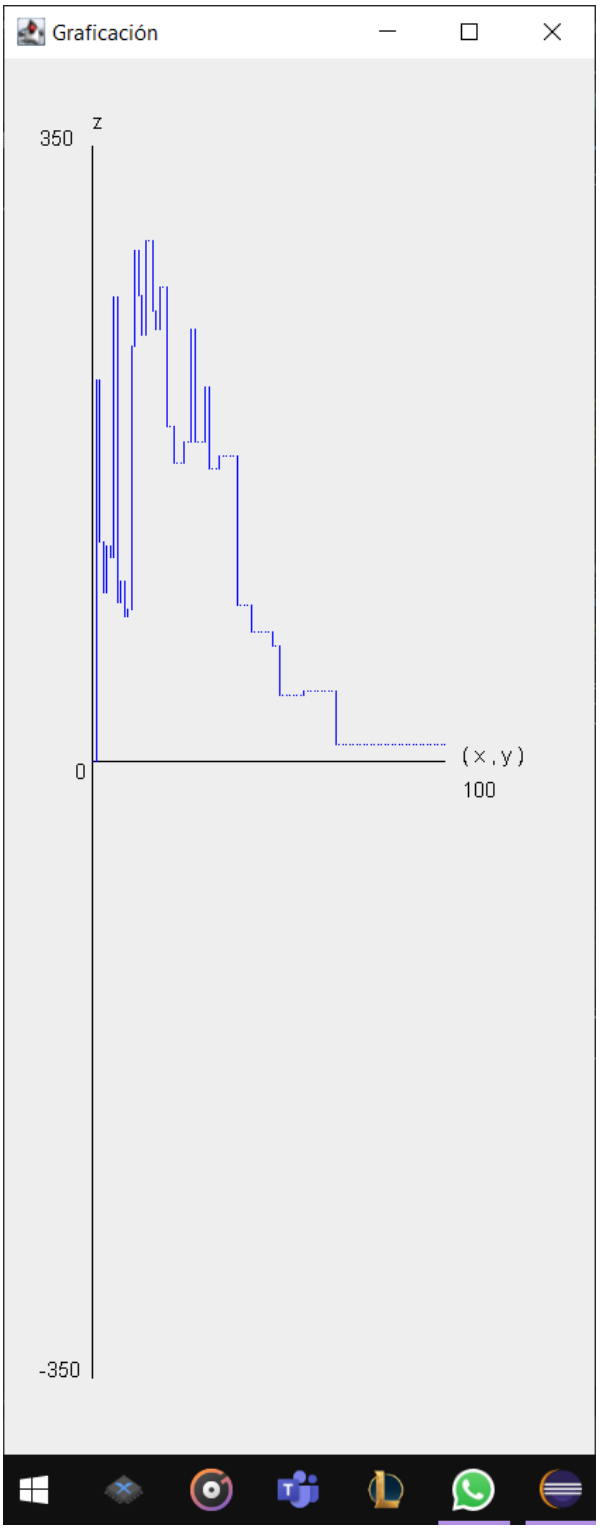
Decrecimiento de la temperatura	Lineal	Exponencial	Logarítmica	Geométrica
Resultado de la ejecución 1	39.64974	8.98632	446.79506	86.09104
Resultado de la ejecución 2	55.16394	10.50356	221.62605	45.76002
Resultado de la ejecución 3	71.47751	21.42511	64.72611	1.01
Resultado de la ejecución 4	61.5454	15.20671	338.49512	32.33069
Resultado de la ejecución 5	48.10466	30.76101	422.47178	11.60471
Resultado de la ejecución 6	20.52861	19.36809	121.67418	25.49595
Resultado de la ejecución 7	117.52223	14.23738	535.75144	12.09586
Resultado de la ejecución 8	24.65627	35.26872	164.88203	22.03403
Resultado de la ejecución 9	61.52777	21.73021	218.25676	19.09903
Resultado de la ejecución 10	94.29975	22.14594	219.32776	10.98946
Resultado Promedio	59.447588	19.963305	275.400629	26.651079

Ahora se muestra algunas gráficas de su comportamiento durante su ejecución:

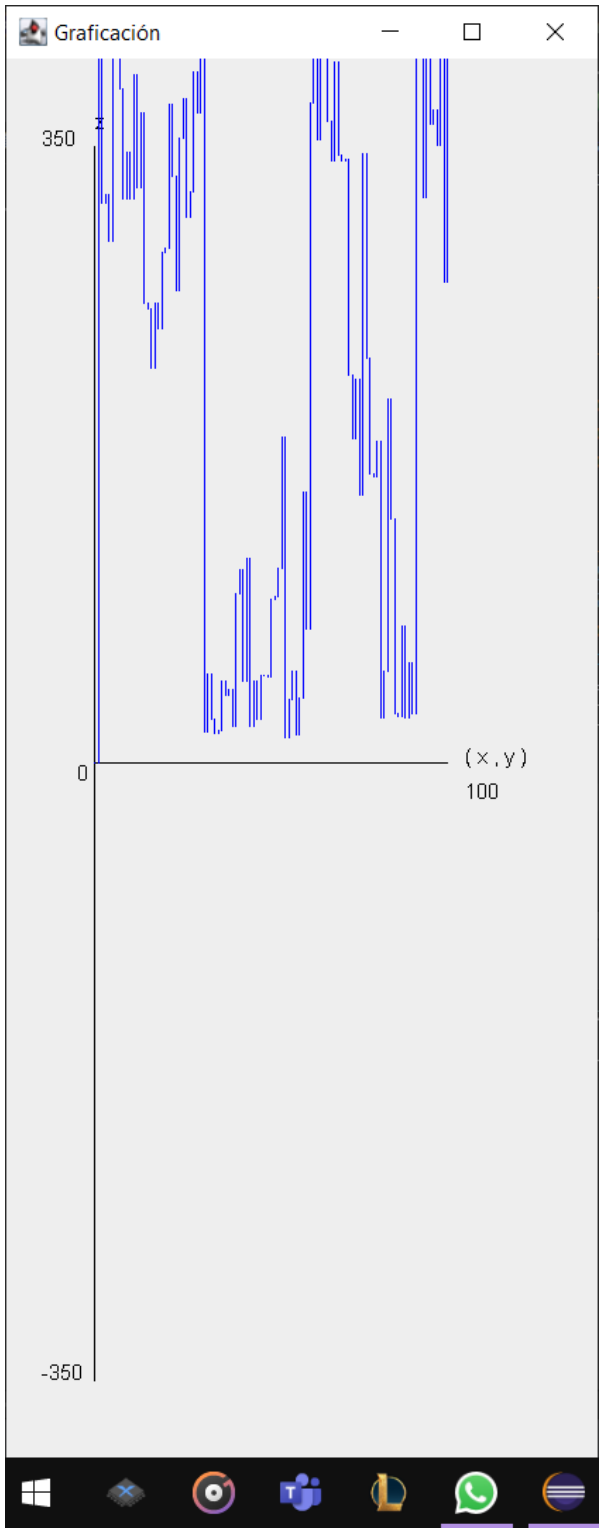
Lineal:



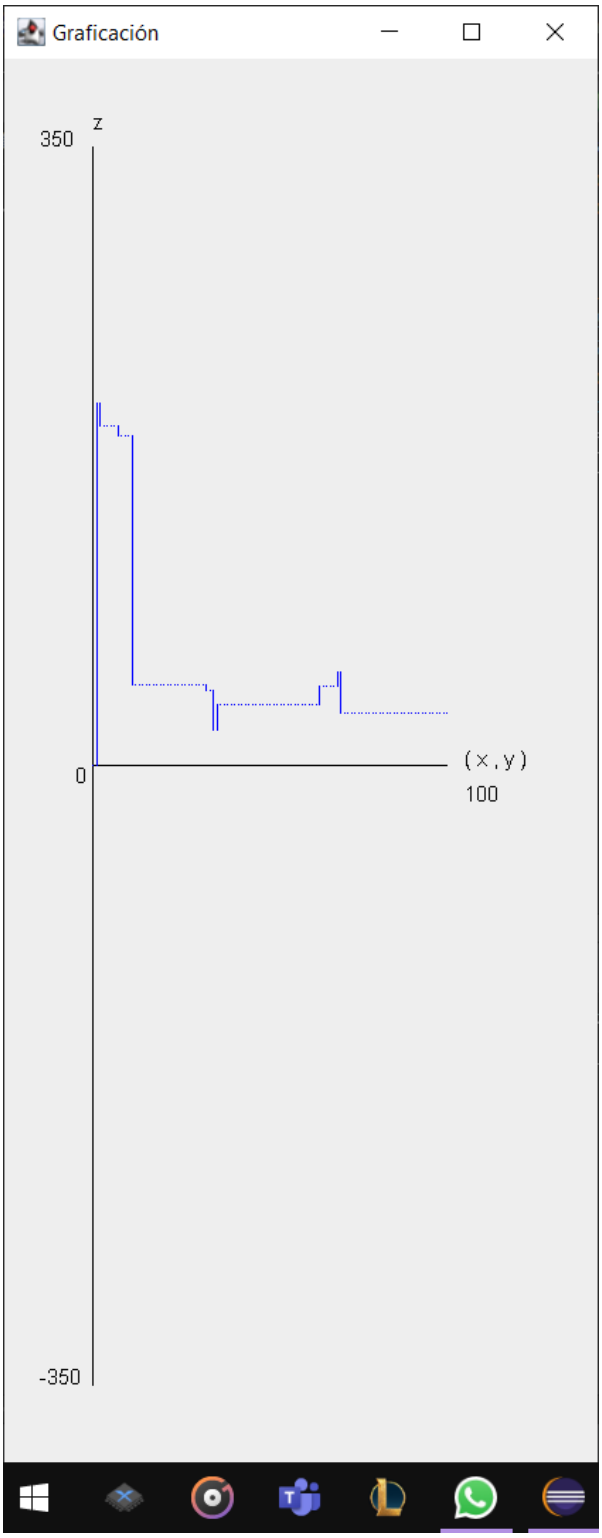
Exponencial:



Logarítmica:



Geométrica:



Conclusión:

Con la realización de esta práctica se ha podido realizar una mejor interpretación sobre el proceso que sigue un algoritmo de recocido simulado, desde como es que

se busca un número dentro de un vecindario cercano, hasta el como determinar si el programa debería o no aceptar un valor peor en cada caso que se presente.

También es importante destacar el conocimiento adquirido referente al lenguaje de programación utilizado (en este caso Java), puesto que el encontrar la forma de por ejemplo realizar permutaciones aleatorias de un grafo, o como hacer que un programa reduzca el tamaño de nuestro dominio, son cosas que, si bien son aspectos secundarios en esta actividad, no por ello dejan de ser nuevas habilidades adquiridas.