



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos II

Grupo: 9

No de Práctica(s): 1

Integrante(s): Díaz Hernández Marcos Bryan

*No. de Equipo de
cómputo empleado:* Elaborado en casa/Equipo personal.

No. de Lista o Brigada: 9

Semestre: 2021-1

Fecha de entrega: 3 de octubre de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la practica

El estudiante identificará la estructura de los algoritmos de ordenamiento por selección y por inserción.

Introducción

En esta práctica usted podrá visualizar la forma en que interprete los algoritmos de ordenamiento, los analice, y los traduzca a la forma en que es más simple para mí entenderlos, espero sea de su agrado.

Ejercicios de la practica:

- Ejercicio 1:

El primer ejercicio consiste en analizar los códigos y las bibliotecas que se proporcionan para resolver el segundo ejercicio.

- Dificultades en el código

Al querer ejecutar los códigos c, surge un problema con la compilación y por lo tanto no se ejecuta (Imagen 1).

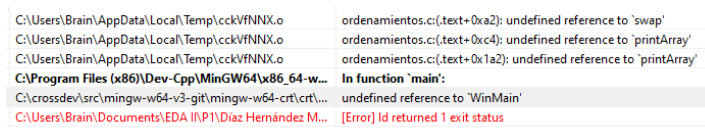


Imagen 1

Ahora el problema es que las librerías que se dan por default para el primer ejercicio tienen una forma bastante extraña de declaración de las funciones que se van a utilizar, porque se menciona en el error que no están definidas las funciones y esto es porque en las librerías (.h) se encuentra únicamente el prototipo de las funciones y en el código (.c), se encuentra el desarrollo completo de las sentencias, por lo que cuando se invocan las bibliotecas solo conocen el prototipo pero no saben cuáles son las sentencias, por lo tanto no están definidas (Imagen [2,3]).

```
void selectionSort(int arreglo[], int n);  
void insertionSort(int a[], int n);
```

Imagen 2

```
void swap(int* a, int* b);  
void printArray(int arr[],int size);  
void printSubArray(int arr[],int low, int high);
```

Imagen 3

- Solución de las dificultades:

Para poder resolverlos únicamente es necesario el decirles a las funciones que tienen que hacer y de ahí el vincular el programa principal con las librerías creadas y de esta forma el problema se elimina y podemos continuar con la ejecución de las demás actividades.

Además lo que hice como primera opción el colocar los prototipos y las sentencias de cada función en una sola biblioteca para que funcionase, esto es una alternativa y se puede aplicar lo mismo con cada biblioteca (utilidades y ordenamientos), como una segunda opción.

Es decir en utilidades colocar todas las funciones completas y en ordenamientos, colocar la biblioteca “utilidades.h”, y ahí definir las otras funciones de ordenamiento. Como están ligadas las bibliotecas al solicitarlas en el siguiente ejercicio, simplemente irán en busca de una dentro de la otra, y al estar definidas ahí las encontrara. (Imagen 1)

```
#include "utilidades.h"
#include <stdio.h>

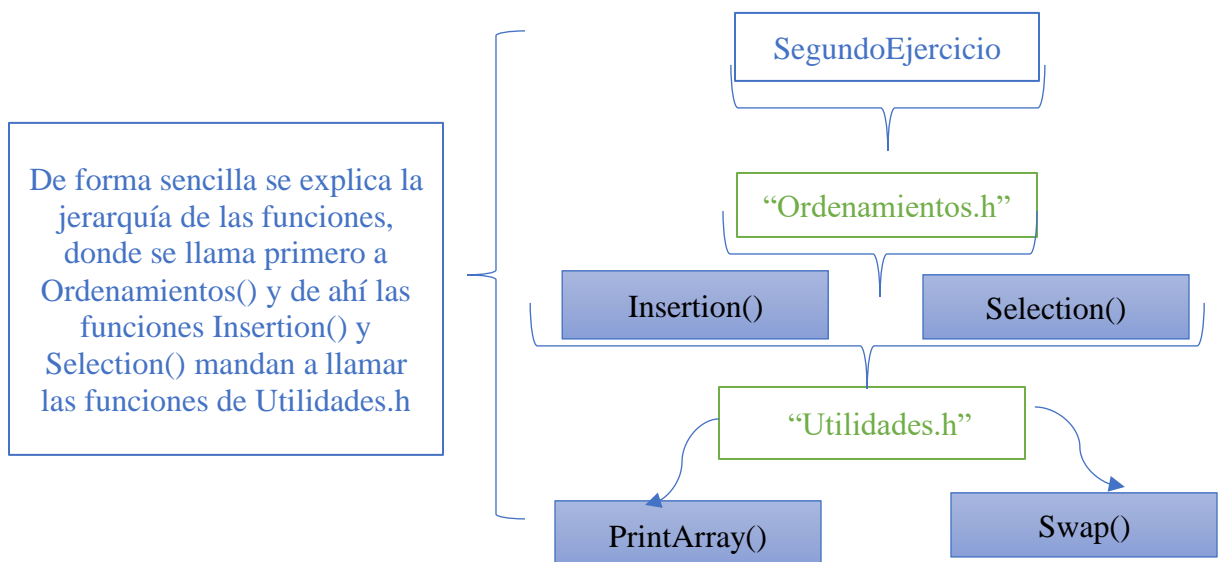
void selectionSort(int arreglo[], int n);
void insertionSort(int a[], int n);

//No estan definidas las líneas de código
// no saben lo que se supone que tendrían
```

Imagen 1

- Diagrama de funcionamiento:

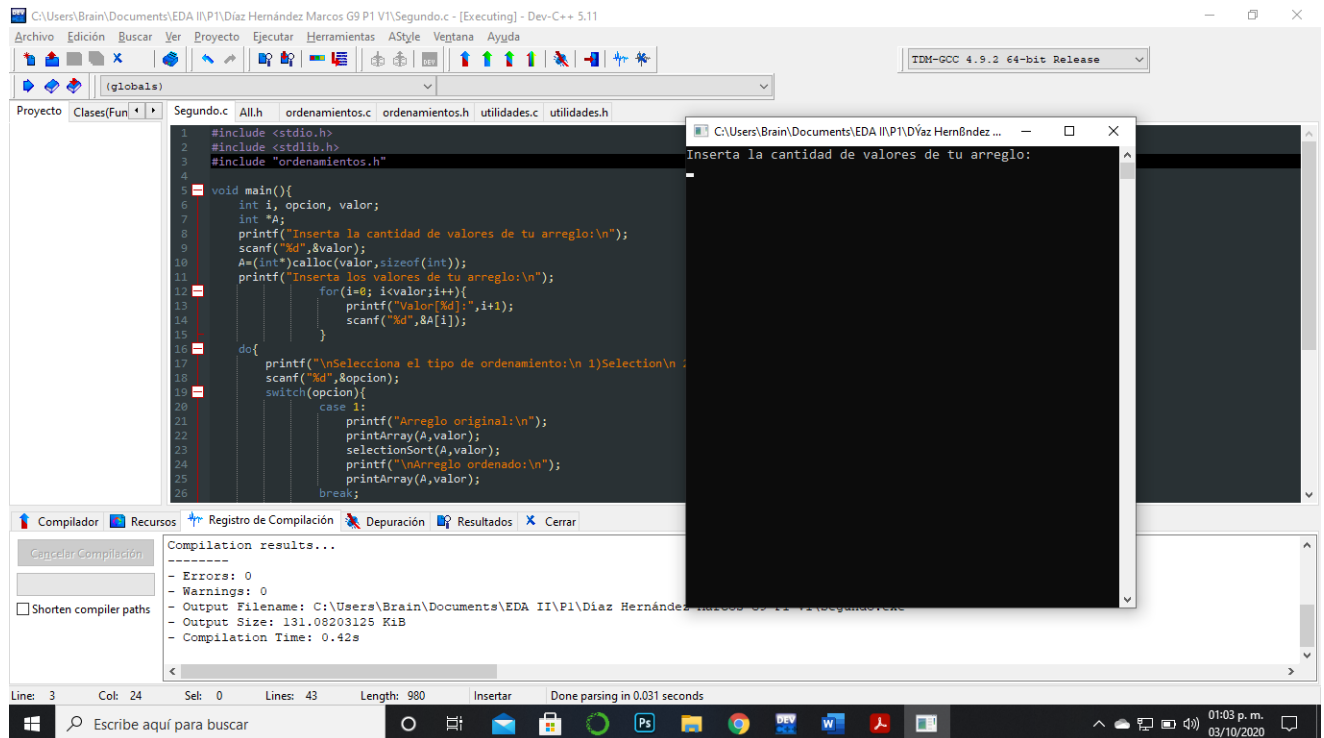
Las funciones más importantes fueron las de Insertion y Selection porque son las que utilizan las utilidades, de la siguiente forma:



- Relación con la teoría:

Esto es más que nada un repaso de las funciones y de las librerías que se pueden crear, además de entender el funcionamiento de los algoritmos solamente con el análisis, de ahí que se puede descubrir el error de las librerías, lo que siento fue el principal reto de aquí y saber cómo se relacionan para poder utilizarlas en el segundo ejercicio.

- Evidencia de implementación



- Ejercicio 2:

El segundo ejercicio consistía en realizar un pequeño menú para que el usuario pudiera seleccionar e ingresar una cantidad de elementos en un arreglo y con la selección del algoritmo de ordenamiento de su preferencia obtener la lista ordenada. Por supuesto mostrando los cambios que se realizaron al arreglo y mostrando la forma inicial y la final del algoritmo.

- Dificultades en el código

El problema principal que encontré dentro de la clase fue que no detectaba las funciones, pero eso era un problema que se tenía que resolver en el primer ejercicio, y como mencionaba anteriormente los resolví de esa manera que mencioné. Ahora con las características que se requerían tuve que repasar algunos elementos para elaborar la selección y la creación del arreglo dinámico, pero los demás no fue difícil, de hecho estuvo bastante sencillo, pero esa parte del primer ejercicio influyó mucho en este ejercicio (Imagen 1).

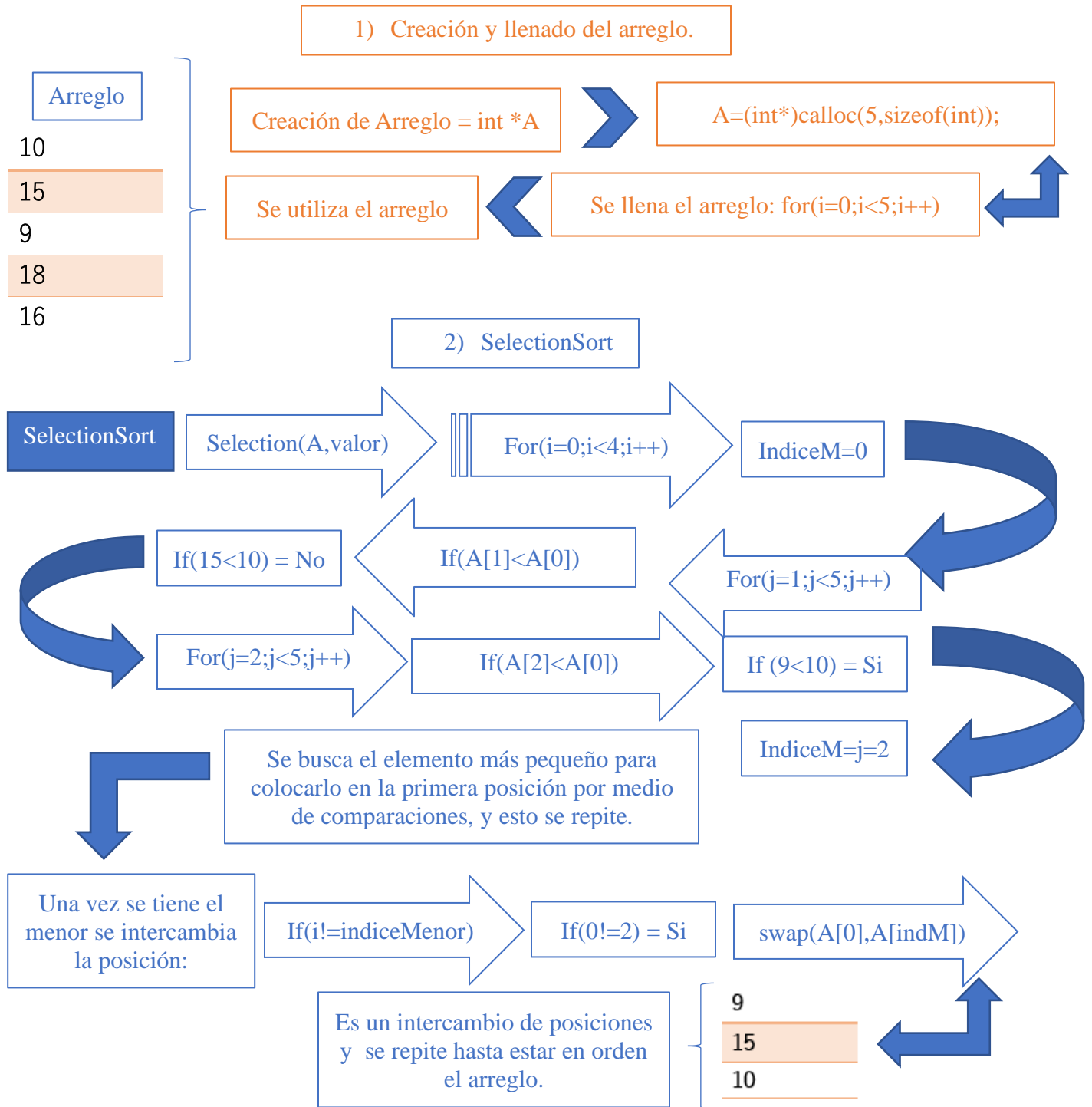
```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "ordenamientos.h"

```

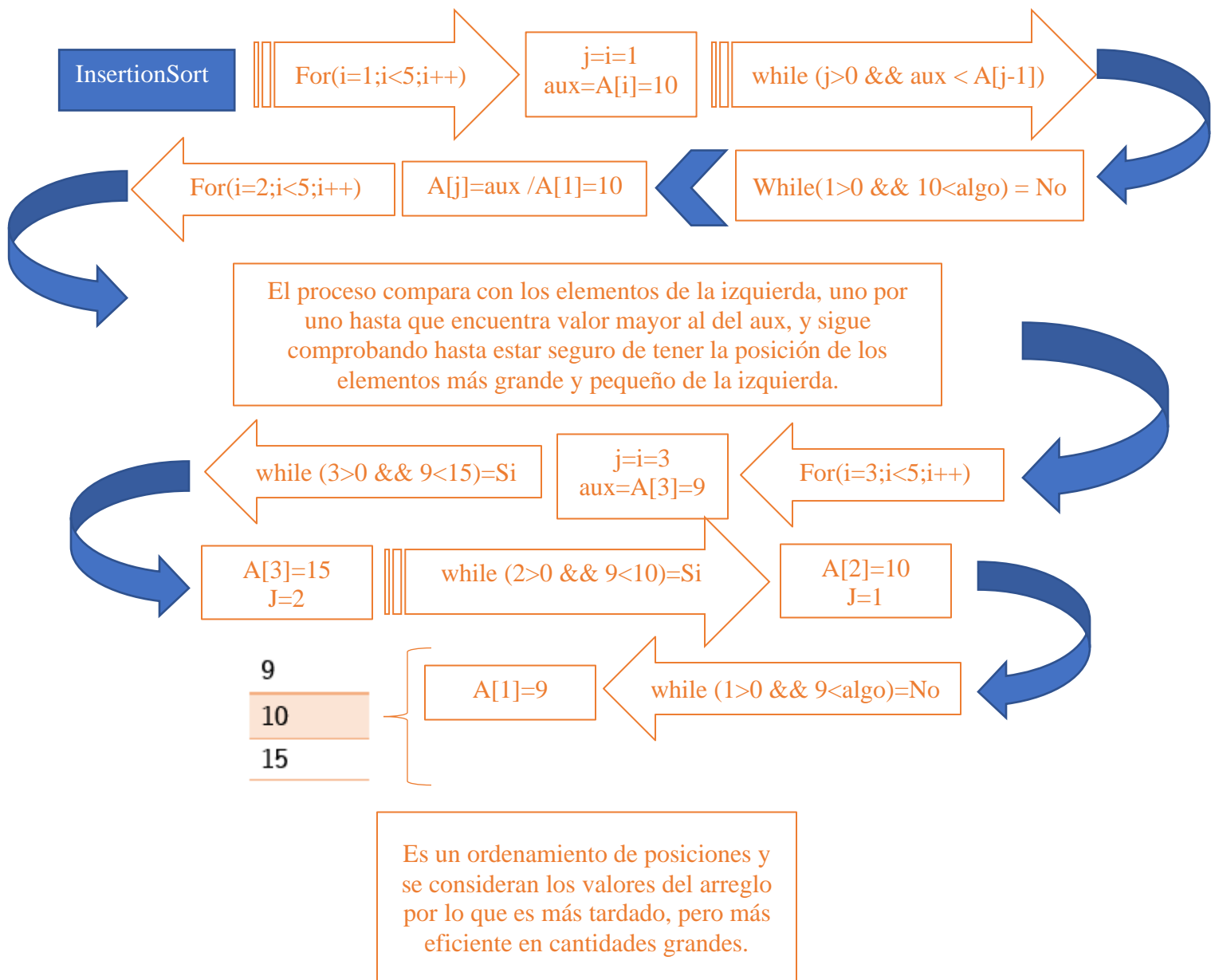
Imagen 1

- Diagrama de funcionamiento:



Por motivos didácticos utilizare el mismo arreglo como ejemplo para ambos algoritmos de ordenamiento.

3) InsertionSort



- Relación con la teoría:

Estos algoritmos fueron dos los primeros que vimos en la clase y tienen sus puntos buenos por supuesto, pero estos diagramas y el ejercicio me ayudaron a comprender mejor como es su funcionamiento e incluso describirlo por medio de un diagrama por lo que está íntimamente relacionado con la teoría que vimos, y más con el código que se ponía de cada uno de estos.

- Evidencia de implementación

The screenshot shows a C++ IDE with the following components:

- Source Code (Segundo.c):**

```

9  scanf("%d",&valor);
10 A=(int*)calloc(valor,sizeof(int));
11 printf("Inserta los valores de tu arreglo:\n");
12 for(i=0; i<valor;i++){
13     printf("Valor[%d]:",i+1);
14     scanf("%d",&A[i]);
15 }
16
17 do{
18     printf("\nSelecciona el tipo de ordenamiento:\n 1)Selection\n 2)Insertion\n");
19     scanf("%d",&opcion);
20     switch(opcion){
21         case 1:
22             printf("Arreglo original:\n");
23             printArray(A,valor);
24             selectionSort(A,valor);
25             printf("\nArreglo ordenado:\n");
26             printArray(A,valor);
27             break;
28         case 2:
29             printf("Arreglo original:\n");
30             printArray(A,valor);
31             insertionSort(A,valor);
32             printf("\nArreglo ordenado:\n");
33             printArray(A,valor);
34             break;
35     }
36 } while(opcion != 0);

```
- Compilation Results:**

```

Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Brain\Documents\EDA II\P1\Díaz Hernández Marcos G9 P1 V1\Segundo.c
- Output Size: 131.08203125 KiB
- Compilation Time: 0.42s

```
- Execution Output:**

```

235
Selecciona el tipo de ordenamiento:
1)Selection
2)Insertion
2
Arreglo original:
52345 5675 467 67 85 987 7896 467 35634 235

Iteracion numero 1
5675 52345 467 67 85 987 7896 467 35634 235

Iteracion numero 2
467 5675 52345 67 85 987 7896 467 35634 235

Iteracion numero 3
67 467 5675 52345 85 987 7896 467 35634 235

Iteracion numero 4
67 85 467 5675 52345 987 7896 467 35634 235

Iteracion numero 5
67 85 467 987 5675 52345 7896 467 35634 235

Iteracion numero 6
67 85 467 987 5675 7896 52345 467 35634 235

Iteracion numero 7
67 85 467 467 987 5675 7896 52345 35634 235

Iteracion numero 8
67 85 467 467 987 5675 7896 35634 52345 235

Iteracion numero 9
67 85 235 467 467 987 5675 7896 35634 52345

Arreglo ordenado:
67 85 235 467 467 987 5675 7896 35634 52345

Selecciona el tipo de ordenamiento:

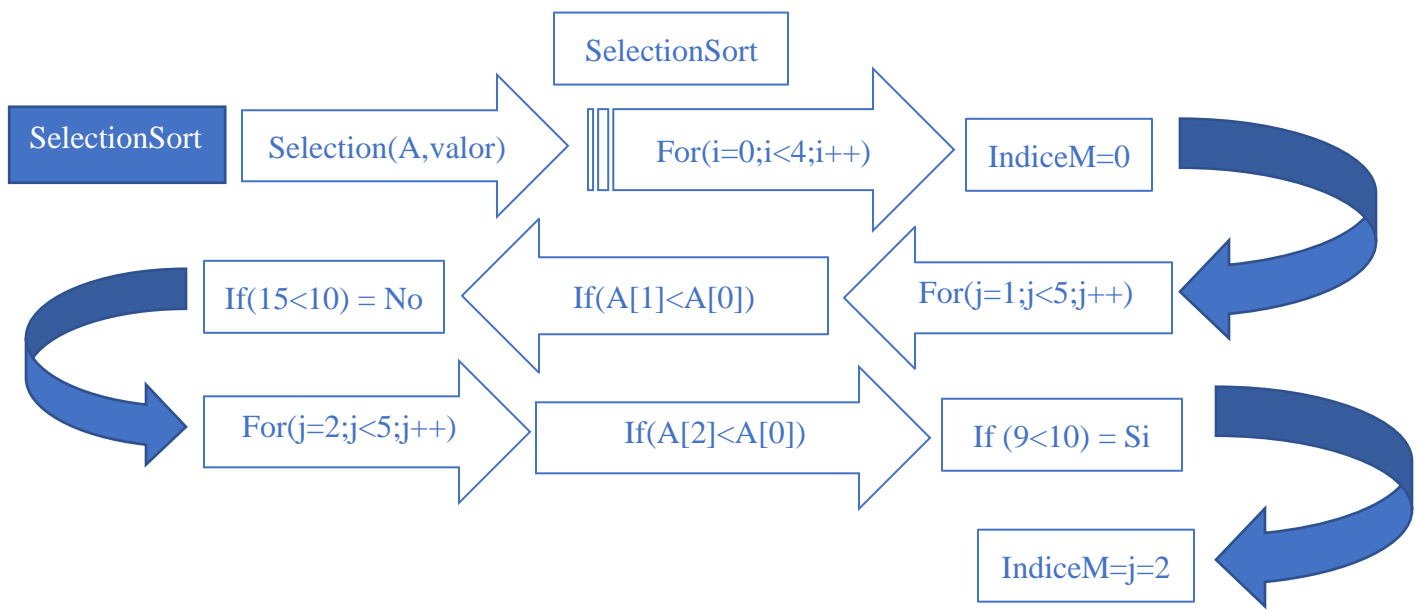
```

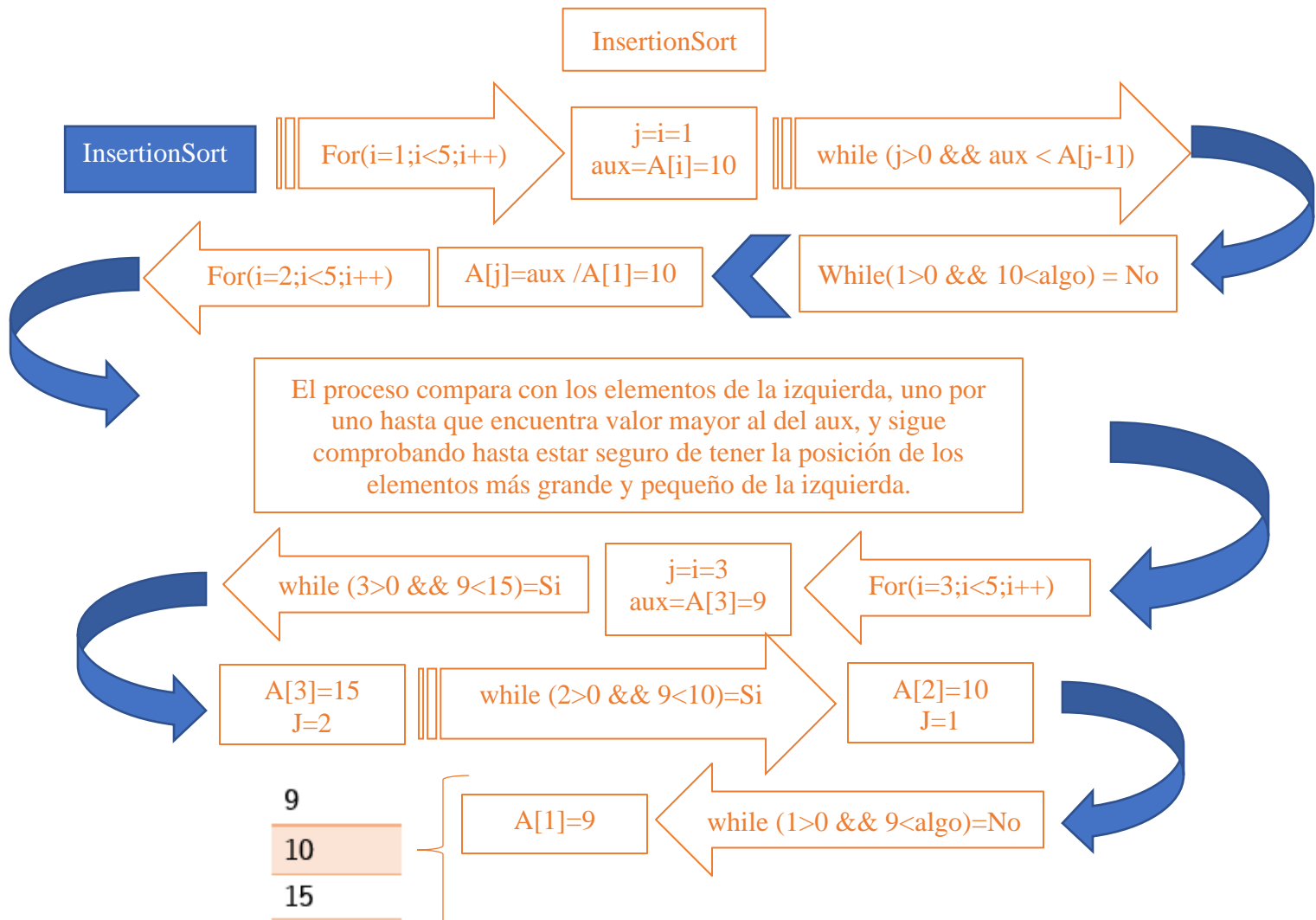
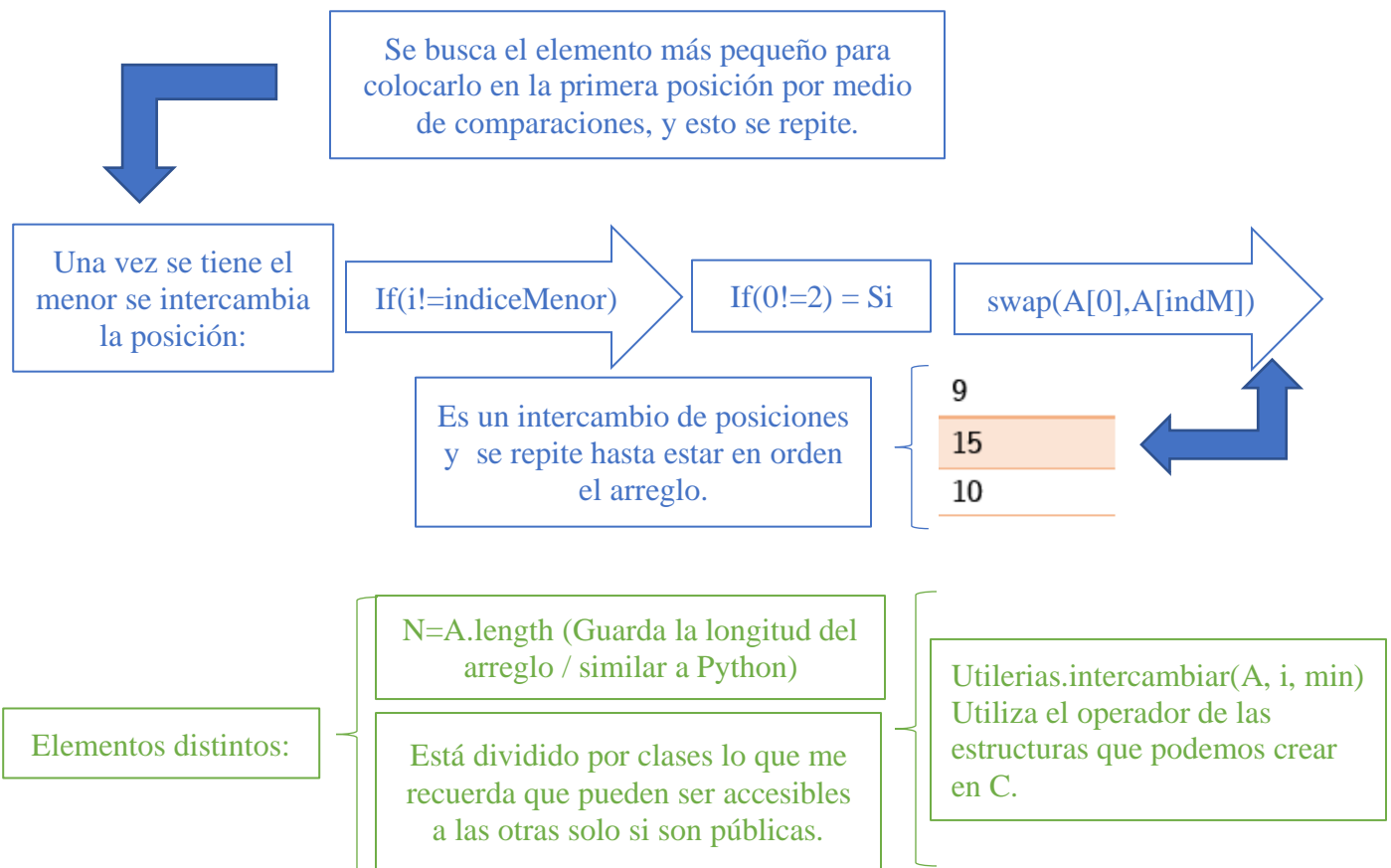
- Ejercicio 3:

En este ejercicio, se pide el ejecutar el proyecto correspondiente a Ejercicio 3, Practica1, bueno es describir lo que sucede en el programa y añadir los comentarios pertinentes a la ejecución y a elementos que me parezcan interesante o desconocidos.

- Diagrama de funcionamiento:

En esta sección del ejercicio pondría los diagramas pero ya los he elaborado anteriormente aunque solo voy a colocarlos con las diferencias que tienen entre Java y C, para que tenga yo una idea más clara de los que sucede en el programa.



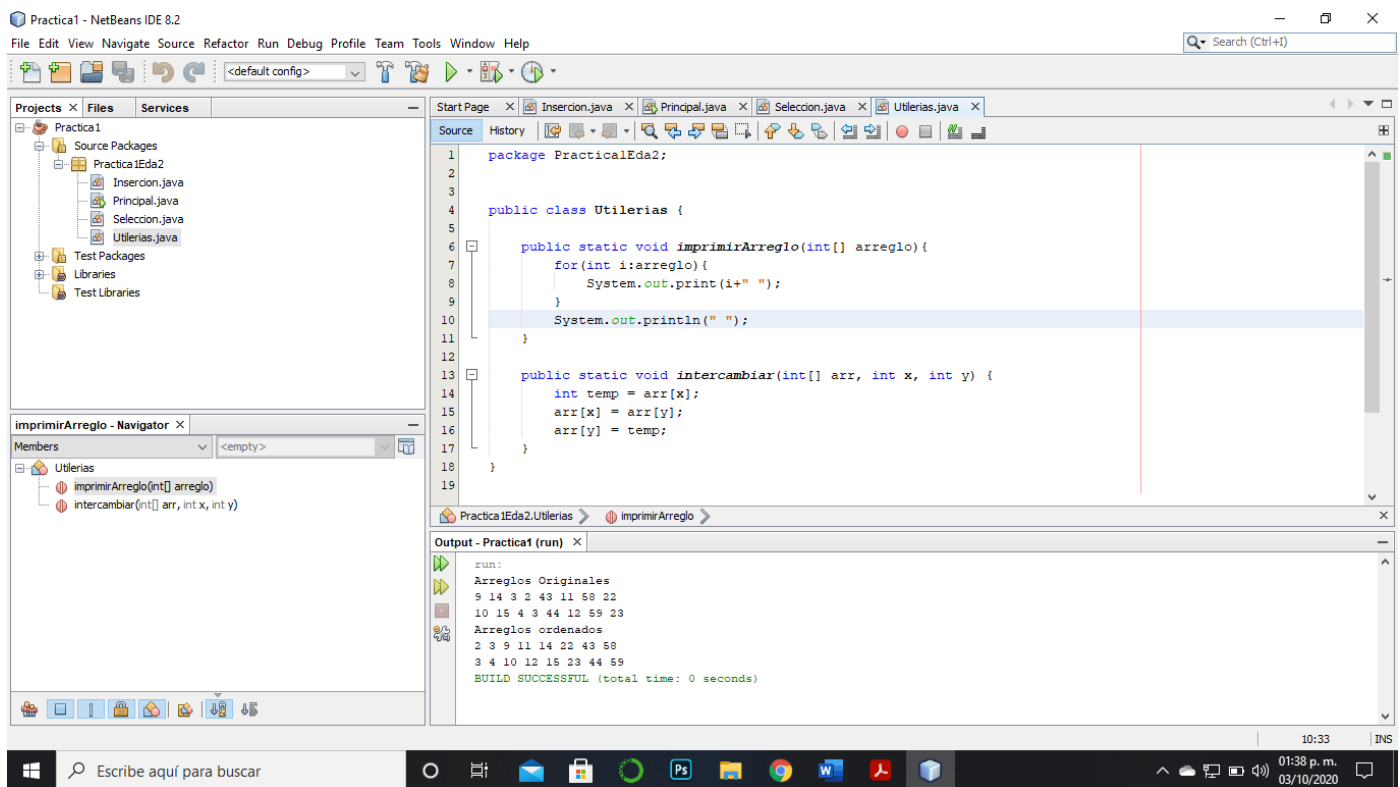


- Relación con la teoría:

Bueno esto es como un viaje entre dimensiones paralelas, lo que sucede es que las dimensiones no necesariamente tienen una interpretación del tiempo ni de los hechos iguales, y lo digo no porque vea muchas series o algo así, sino que esto es parte de POO y EDA, la situación aquí es que en mi clase de POO, casi no hemos tocado JAVA, además que vamos atrasados una clase, pero pronto voy a disponer de mi tiempo libre a estudiar y programar en JAVA.

Sin embargo, esto es totalmente relacionado con los algoritmos de ordenamiento solo que implementados en dos paradigmas distintos y siendo honesto el tener varias pestañas que separen las funciones es mucho más agradable que estar bajando sobre líneas de código.

- Evidencia de implementación



```
package PracticalEda2;

public class Utilerias {

    public static void imprimirArreglo(int[] arreglo) {
        for(int i:arreglo){
            System.out.print(i+" ");
        }
        System.out.println(" ");
    }

    public static void intercambiar(int[] arr, int x, int y) {
        int temp = arr[x];
        arr[x] = arr[y];
        arr[y] = temp;
    }
}
```

run:

Arreglos Originales
9 14 3 2 43 11 58 22
10 15 4 3 44 12 59 23
Arreglos ordenados
2 3 9 11 14 22 43 58
3 4 10 12 15 23 44 59
BUILD SUCCESSFUL (total time: 0 seconds)

- Ejercicio 4:

El último ejercicio consistía en colocar contadores/instrucciones que nos ayudaran a poder llevar un control de las diferentes operaciones que se llevan a cabo en el programa al momento de la ejecución además de que todas estas instrucciones serán usadas para el análisis de complejidad de cada algoritmo.

- Dificultades en el código

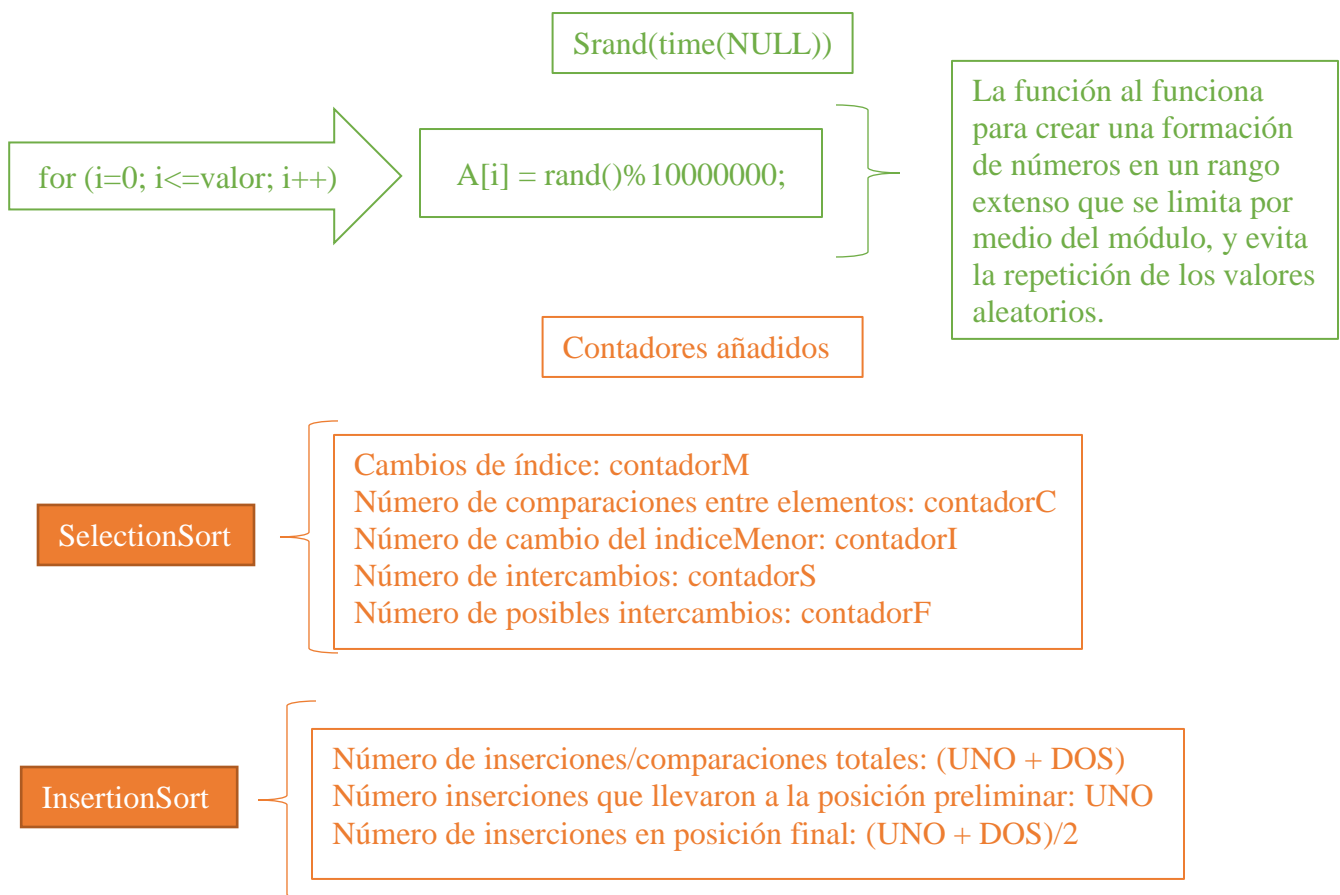
Hubo una parte donde me entretuve como una hora pensando, entre buscando como solucionarlo y haciendo otras cosas para poder despejar la mente y tener más claridad, para poder encontrar la solución, pero no la encontré.

El problema vino en el algoritmo de Insertion, porque quería contar las veces que los números cambiaban de posición, pero mi error fue el considerar los algoritmos como similares, y al final me di cuenta de que el Insertion hace inserciones y para cada número que cambia de posición hace dos operaciones, por lo que este último detalle me llevo a una forma de estimar los números que se movían, no decir cuales eran, sino cuantas veces se intercambiaron de lugar.

Igual intente el usar otro arreglo y compararlos uno que fingiera ser el original sin cambios y el otro ya ordenado, pero al comparar las posiciones no me dieron lo esperado, además que los números aleatorios no me permitían tener un estudio más claro de lo que sucedía en cada ejecución.

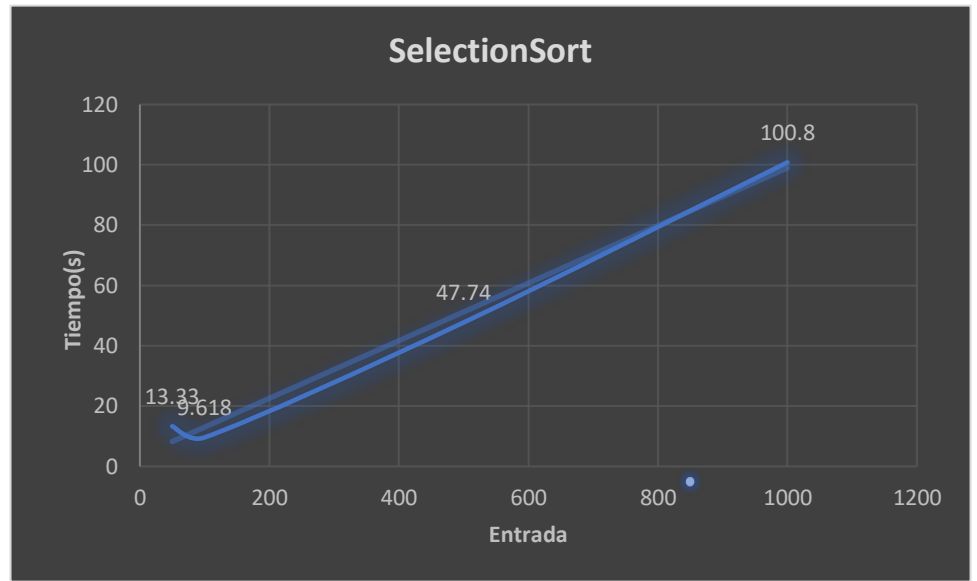
- Diagrama de funcionamiento:

El programa solo contiene contadores, para poder verificar las operaciones realizadas, pero voy a agregar la función de números aleatorios aquí para que se pueda ver como realice la función y que ayuda bastante a la solución de los casos de 50,100,500 y 1000, elementos.



- Análisis de la complejidad
 - SelectionSort

ENTRADA	TIEMPO(s)
50	13.33
100	9.618
500	47.74
1000	100.8



Operacio- nes	DATOS
1255	50
4950	100
124750	500
499500	1000



En base a las gráficas y a las pruebas realizadas, en algunas ocasiones los tiempos de ejecución tendían a ser menores, y esto es debido al procesador de la computadora y el sistema operativo de la misma, además del rendimiento.

En este caso SelectionSort parece tener una complejidad lineal, que tiende a ser cuadrática, esto va en función de los valores que tiene que ordenar y al se aleatorios pues se tiene un caso promedio, con $O(n^2)$, eso en la primera gráfica, ya que los tiempos de ejecución se duplican en base a la cantidad de la entrada.

En la segunda gráfica se tienen los datos que se dan y las operaciones que se realizan con estos, de hecho, igual se puede ver un crecimiento equivalente de 50 a 100, pero cuando llega a 500 miembros del arreglo, la proporción se pierde y comienza a ser mas compleja, por lo que en grandes cantidades es más complejo de utilizar.

- InsertionSort

ENTRADA	TIEMPO(s)
50	5.544
100	11.24
500	32.5
1000	94.64



Operaciones	DATOS
515	50
2285	100
62772	500
250001	1000



En la primer grafica es posible ver la tendencia lineal/cuadrática de la función, debido a que los tiempos en base a la entrada, son proporcionales, de hecho son mas uniformes que los del SelectionSort, ya que considera a los elementos anteriores ya ordenados, por lo que en cantidades pequeñas puede obtener ventaja, pero la grafica de Insertion esta mas inclinada a ser cuadrática por lo que la complejidad tiende a ser $O(n^2)$.

En cuanto a la segunda grafica que se basa en las operaciones por entrada, estas se cuatuplican aproximadamente por cada duplicación (50-100), pero de 100 a 500, se vuelve 30 veces lo de 100 elementos de entrada, y al doble de 500, se cuatuplica, por lo que las operaciones se multiplican al subir la entrada, con un tendencia lineal, lo que sería algo lógico si se considera que hace comparaciones entre los miembros que ya están ordenados.

- Relación con la teoría:

En este momento esta es la parte mas relevante de los algoritmos y creo es un elemento importante de que sean utilizados o no, y esto es porque la complejidad de los algoritmos dicta que tan eficiente es utilizarlo o no, además de ser un tema del semestre anterior, toma su verdadero peso al estar dentro de los algoritmos. Además que nos permite darnos una idea de como se realizan los análisis superficiales de los algoritmos cuando se busca su complejidad.

- Evidencias de implementación

```

C:\Users\Brain\Documents\EDA I\PT1\Cuarto.exe
Iteracion numero 47
1290 1870 3199 3936 4187 4702 7087 7666 7705 8151 8370 8451 9199 9274 9622 10045 11047 11900 12110 13079 14394 17378 184
87 18532 20253 20719 20940 21278 21648 21699 23526 23646 24396 24526 24706 25287 25944 26241 26576 27065 28122 28338 284
42 28610 29023 30050 31132 31480 32385 32262

Iteracion numero 48
1290 1870 3199 3936 4187 4702 7087 7666 7705 8151 8370 8451 9199 9274 9622 10045 11047 11900 12110 13979 14394 17378 184
87 18532 20253 20719 20940 21278 21648 21699 23526 23646 24396 24526 24706 25287 25944 26241 26576 27065 28122 28338 284
42 28610 29023 30050 31132 31480 32385 32262

Iteracion numero 49
1290 1870 3199 3936 4187 4702 7087 7666 7705 8151 8370 8451 9199 9274 9622 10045 11047 11900 12110 13979 14394 17378 184
87 18532 20253 20719 20940 21278 21648 21699 23526 23646 24396 24526 24706 25287 25944 26241 26576 27065 28122 28338 284
42 28610 29023 30050 31132 31480 32262 32385

Cambios de indice i: 49
Numero de comparaciones entre elementos: 1225
Numero de cambio del indiceMenor: 155
Numero de intercambios: 46
Numero de posibles intercambios: 49

Arreglo ordenado:
1290 1870 3199 3936 4187 4702 7087 7666 7705 8151 8370 8451 9199 9274 9622 10045 11047 11900 12110 13979 14394 17378 184
87 18532 20253 20719 20940 21278 21648 21699 23526 23646 24396 24526 24706 25287 25944 26241 26576 27065 28122 28338 284
42 28610 29023 30050 31132 31480 32262 32385

-----
Process exited after 13.33 seconds with return value 1
Presione una tecla para continuar . . .

```

50 -SelectionSort

```

C:\Users\Brain\Documents\EDA I\PT1\Cuarto.exe
4517 14694 14892 14904 16019 17502 18287 19651 19713 20018 20842 22355 23204 25224 25470 26105 26475 26530 28311 28754 2
9285 30710 31294 32004 6442 19451 29653

Iteracion numero 47
134 242 1242 2971 3028 3915 5151 5435 5779 6060 6169 6435 6442 7500 8476 9105 9801 9974 10284 10820 11027 11076 11144 13
201 14517 14694 14892 14904 16019 17502 18287 19651 19713 20018 20842 22355 23204 25224 25470 26105 26475 26530 28311 28
754 29285 30710 31294 32004 19451 29653

Iteracion numero 48
134 242 1242 2971 3028 3915 5151 5435 5779 6060 6169 6435 6442 7500 8476 9105 9801 9974 10284 10820 11027 11076 11144 13
201 14517 14694 14892 14904 16019 17502 18287 19451 19651 19713 20018 20842 22355 23204 25224 25470 26105 26475 26530 28
311 28754 29285 30710 31294 32004 29653

Iteracion numero 49
134 242 1242 2971 3028 3915 5151 5435 5779 6060 6169 6435 6442 7500 8476 9105 9801 9974 10284 10820 11027 11076 11144 13
201 14517 14694 14892 14904 16019 17502 18287 19451 19651 19713 20018 20842 22355 23204 25224 25470 26105 26475 26530 28
311 28754 29285 29653 30710 31294 32004

Numero de inserciones/comparaciones totales: 515
Numero inserciones que llevaron a la posicion preliminar: 466
Numero de inserciones en posicion final: 257

Arreglo ordenado:
134 242 1242 2971 3028 3915 5151 5435 5779 6060 6169 6435 6442 7500 8476 9105 9801 9974 10284 10820 11027 11076 11144 13
201 14517 14694 14892 14904 16019 17502 18287 19451 19651 19713 20018 20842 22355 23204 25224 25470 26105 26475 26530 28
311 28754 29285 29653 30710 31294 32004

-----
Process exited after 5.544 seconds with return value 1
Presione una tecla para continuar . . .

```

50 – InsertionSort

```
C:\Users\Brain\Documents\EDA I\NP1\Cuarto.exe

Iteracion numero 98
472 569 818 1103 1508 1682 1894 2031 2681 3017 3358 3387 4366 4410 4682 4723 5609 5812 6082 6780 7018 7114 7833 8155
2 9188 9226 9869 10281 10592 10939 11797 11967 12142 12175 12370 12777 12793 12844 13459 14090 14158 14661 14915
5602 16097 16586 16703 16987 17247 17708 17868 18624 20225 20494 20779 20780 21410 21552 21928 22309 22747 22897
4329 24608 24685 24701 25188 25219 25235 25244 25377 26245 26779 26994 27361 27777 28478 28537 29202 29202 29388
9802 29809 30058 30406 30740 30831 30874 30906 31007 31428 31641 32114 32141 32173 32493

Iteracion numero 99
472 569 818 1103 1508 1682 1894 2031 2681 3017 3358 3387 4366 4410 4682 4723 5609 5812 6082 6780 7018 7114 7833 8155
2 9188 9226 9869 10281 10592 10939 11797 11967 12142 12175 12370 12777 12793 12844 13459 14090 14158 14661 14915
5602 16097 16586 16703 16987 17247 17708 17868 18624 20225 20494 20779 20780 21410 21552 21928 22309 22747 22897
4329 24608 24685 24701 25188 25219 25235 25244 25377 26245 26779 26994 27361 27777 28478 28537 29202 29202 29388
9802 29809 30058 30406 30740 30831 30874 30906 31007 31428 31641 32114 32141 32173 32493

Cambios de indice i: 99
Numero de comparaciones entre elementos: 4950
Numero de cambio del indiceMenor: 299
Numero de intercambios: 94
Numero de posibles intercambios: 99

Arreglo ordenado:
472 569 818 1103 1508 1682 1894 2031 2681 3017 3358 3387 4366 4410 4682 4723 5609 5812 6082 6780 7018 7114 7833 8155
2 9188 9226 9869 10281 10592 10939 11797 11967 12142 12175 12370 12777 12793 12844 13459 14090 14158 14661 14915
5602 16097 16586 16703 16987 17247 17708 17868 18624 20225 20494 20779 20780 21410 21552 21928 22309 22747 22897
4329 24608 24685 24701 25188 25219 25235 25244 25377 26245 26779 26994 27361 27777 28478 28537 29202 29202 29388
9802 29809 30058 30406 30740 30831 30874 30906 31007 31428 31641 32114 32141 32173 32493

-----
Process exited after 9.618 seconds with return value 1
Presione una tecla para continuar . . .
```

100 - SelectionSort

```
C:\Users\Brain\Documents\EDA I\NP1\Cuarto.exe

27149 27755 28464 28819 29039 29917 30254 30649 31097 31580 32168 32690 28663 19180

Iteracion numero 98
87 242 307 412 446 758 1183 1217 1500 1619 1978 2426 2568 2732 3891 3934 4074 4824 5164 5317 6368 6452 7321 7713 8225 8511
86 9360 9917 10122 10935 10970 12599 12776 13021 13528 13867 14102 14120 14303 14507 14809 15435 15752 15885 15910 16075
16101 16247 16366 16506 16510 17185 17460 17651 17834 18409 18527 18796 18842 18874 19167 19753 20644 20793 21104 21437
21460 21698 21766 22400 22467 22473 22956 23093 23188 23245 23606 23832 24274 25480 25520 25625 26336 26422 26596 26961
27149 27755 28464 28663 28819 29039 29917 30254 30649 31097 31580 32168 32690 19180

Iteracion numero 99
87 242 307 412 446 758 1183 1217 1500 1619 1978 2426 2568 2732 3891 3934 4074 4824 5164 5317 6368 6452 7321 7713 8225 8511
86 9360 9917 10122 10935 10970 12599 12776 13021 13528 13867 14102 14120 14303 14507 14809 15435 15752 15885 15910 16075
16101 16247 16366 16506 16510 17185 17460 17651 17834 18409 18527 18796 18842 18874 19167 19180 19753 20644 20793 21104
21437 21460 21698 21766 22400 22467 22473 22956 23093 23188 23245 23606 23832 24274 25480 25520 25625 26336 26422 26596
26961 27149 27755 28464 28663 28819 29039 29917 30254 30649 31097 31580 32168 32690

Numero de inserciones/comparaciones totales: 2285
Numero inserciones que llevaron a la posicion preliminar: 2186
Numero de inserciones en posicion final: 1142

Arreglo ordenado:
87 242 307 412 446 758 1183 1217 1500 1619 1978 2426 2568 2732 3891 3934 4074 4824 5164 5317 6368 6452 7321 7713 8225 8511
86 9360 9917 10122 10935 10970 12599 12776 13021 13528 13867 14102 14120 14303 14507 14809 15435 15752 15885 15910 16075
16101 16247 16366 16506 16510 17185 17460 17651 17834 18409 18527 18796 18842 18874 19167 19180 19753 20644 20793 21104
21437 21460 21698 21766 22400 22467 22473 22956 23093 23188 23245 23606 23832 24274 25480 25520 25625 26336 26422 26596
26961 27149 27755 28464 28663 28819 29039 29917 30254 30649 31097 31580 32168 32690

-----
Process exited after 11.24 seconds with return value 1
Presione una tecla para continuar . . .
```

100 - InsertionSort

```
C:\Users\Brain\Documents\EDA I\NP1\Cuarto.exe

1 28606 28618 28869 28951 28990 29013 29106 29216 29434 29476 29714 29734 29741 29772 29818 29837 29878 29909 29940
5 30029 30068 30074 30140 30161 30198 30257 30452 30499 30544 30658 30777 30822 31202 31252 31280 31403 31551 31580
4 31810 31847 31910 31989 32010 32199 32262 32427 32475 32555 32556 32607 32709 32766

Cambios de indice i: 499
Numero de comparaciones entre elementos: 124750
Numero de cambio del indiceMenor: 2576
Numero de intercambios: 494
Numero de posibles intercambios: 499

Arreglo ordenado:
16 41 54 79 181 213 226 302 469 507 754 870 892 916 1149 1190 1223 1246 1261 1413 1572 1646 1650 1727 1730 1752 1753
6 1839 1921 1927 1980 2086 2230 2253 2427 2516 2690 2701 2721 2793 3042 3111 3181 3187 3226 3276 3384 3498 3529 3530
9 3766 3766 3923 3974 4114 4214 4300 4302 4329 4467 4528 4594 4661 4666 4691 4987 5025 5066 5137 5143 5240 5377 5379
4 5482 5523 5524 5593 5974 5979 5994 6104 6157 6190 6201 6273 6390 6457 6465 6487 6537 6543 6558 6620 6639 6722 6724
2 7100 7265 7301 7760 7802 7925 8030 8300 8301 8506 8542 8548 8671 8674 8858 8861 8891 8918 9187 9220 9241 9324 9324
0 9474 9501 9510 9544 9564 9583 9589 9641 9799 9803 9950 9954 10015 10030 10065 10124 10127 10256 10269 10283 10313
4 10488 10509 10518 10553 10600 10611 10622 10642 10844 10947 10960 11036 11146 11321 11324 11379 11441 11454 11454
8 11529 11537 11584 11614 11647 11751 11773 11881 11890 11921 12185 12207 12345 12669 12769 12850 12894 12938 12938
9 13053 13116 13193 13352 13391 13431 13432 13458 13480 13504 13631 13672 13775 13800 13846 13878 14002 14060 14141
2 14153 14173 14183 14219 14238 14243 14342 14375 14377 14445 14526 14555 14557 14564 14620 14665 14724 14746 14841
3 15001 15025 15079 15227 15264 15320 15334 15359 15371 15396 15533 15567 15567 15661 15680 15738 15744 15871 15919
5 15986 16138 16216 16226 16347 16413 16572 16588 16626 16644 16673 16711 16715 16771 16809 16841 16927 16951 16951
2 17212 17367 17430 17473 17530 17552 17663 17723 17764 17779 17802 17841 17878 18027 18167 18179 18196 18237 18237
8 18345 18349 18360 18405 18565 18594 18732 18812 18834 18893 18932 18966 19007 19029 19083 19091 19218 19413 19413
9 19566 19794 19796 19877 19911 19943 19979 20236 20297 20520 20631 20733 20825 21087 21103 21198 21293 21374 21374
5 21463 21549 21622 21741 21833 21844 21913 22144 22205 22465 22612 22707 22720 22938 22953 23017 23025 23150 23150
4 23278 23338 23368 23434 23579 23686 23731 23751 23850 23853 23873 23913 24026 24042 24102 24137 24164 24168 24241
4 24635 24685 24861 24953 25031 25032 25123 25250 25312 25320 25343 25346 25349 25394 25478 25512 25546 25561 25561
8 25777 25816 25948 26021 26022 26108 26155 26188 26237 26307 26313 26357 26367 26387 26440 26452 26717 26748 26808
6 26857 26960 27072 27080 27103 27185 27196 27207 27230 27240 27261 27288 27317 27383 27435 27466 27471 27493 27593
1 27593 27641 27715 27717 27760 27768 27808 27902 27918 27925 27947 27962 27967 28030 28085 28087 28121 28121
1 28606 28618 28869 28951 28990 29013 29106 29216 29434 29476 29714 29734 29741 29772 29818 29837 29878 29909 29940
5 30029 30068 30074 30140 30161 30198 30257 30452 30499 30544 30658 30777 30822 31202 31252 31280 31403 31551 31580
4 31810 31847 31910 31989 32010 32199 32262 32427 32475 32555 32556 32607 32709 32766

-----
Process exited after 47.74 seconds with return value 1
Presione una tecla para continuar . . .
```

500 - SelectionSort


```
C:\Users\Brain\Documents\EDA I\PI\Cuarto.exe
869 30954 30963 30979 31051 31074 31127 31209 31252 31303 31340 31353 31396 31422 31450 31619 31624 31754 31781 32121 32
230 32239 32262 32377 32419 32461 32470 32476 32557 32586 32619 32740

Numero de inserciones/comparaciones totales: 62772
Numero inserciones que llevaron a la posicion preliminar: 62273
Numero de inserciones en posicion final: 31386

Arreglo ordenado:
34 106 191 192 254 276 327 393 513 612 629 696 710 942 1489 1748 1749 1781 1795 1891 1921 1947 1979 2034 2039 2046 2051
2062 2142 2176 2306 2316 2385 2427 2615 2702 2718 2785 2798 2910 2967 3090 3178 3299 3306 3331 3458 3460 3462 3479 3490
3516 3583 3604 3692 3703 3714 3752 3770 3832 3877 3980 4022 4172 4263 4337 4488 4539 4548 4551 4569 4592 4676 4703 4733
4735 4753 4856 4975 5004 5134 5174 5234 5238 5287 5381 5446 5524 5562 5616 5668 5735 5851 5893 5895 6349 6481 6540 6544
6583 6589 6771 6793 6884 6959 6985 6995 7050 7132 7233 7251 7453 7504 7800 7805 7873 7877 7919 7920 8067 8067 8083 8083
8265 8329 8369 8394 8427 8441 8497 8522 8671 8673 8680 8725 8739 9084 9086 9163 9206 9210 9218 9238 9270 9524 9625 9638
9708 9732 9749 9793 9811 9832 9843 9972 10098 10143 10170 10182 10203 10424 10449 10485 10488 10604 10744 10873 10884 10
886 11024 11048 11135 11142 11146 11252 11257 11368 11400 11445 11539 11732 11742 11758 11770 11961 11968 12026 12170 12
193 12242 12369 12431 12450 12681 12726 12762 12788 12805 12864 12879 12890 12915 12933 13041 13171 13257 13370 13395 13
398 13441 13451 13703 13733 13753 14257 14375 14480 14487 14631 14645 14665 14728 14827 14891 14971 15055 15099 15106 15
213 15292 15353 15363 15425 15477 15627 15649 15727 15979 16088 16097 16126 16129 16235 16425 16462 16535 16559 16764 16
773 16776 16921 16986 17016 17076 17135 17181 17236 17442 17576 17752 17779 17803 17925 18015 18036 18090 18141 18152 18
354 18464 18466 18475 18563 18576 18608 18697 18711 18771 18817 18831 18969 19005 19048 19091 19113 19277 19353 19415 19
420 19425 19563 19575 19582 19730 19760 19986 19993 19994 19998 20095 20118 20264 20414 20538 20616 20724 20820 20876 20
883 20923 20947 20977 21025 21032 21039 21070 21071 21152 21213 21282 21358 21374 21388 21531 21589 21660 21764 21
801 21985 21991 22018 22054 22070 22183 22221 22351 22390 22445 22453 22768 22775 22808 22866 22867 22909 22934 22939 23
021 23050 23053 23072 23085 23087 23203 23263 23314 23384 23500 23561 23570 23637 23649 23677 23882 23965 23967 24018 24
117 24223 24369 24449 24585 24643 24678 24681 24833 24849 24890 24958 25107 25219 25265 25312 25349 25364 25391 25437 25
458 25478 25498 25525 25691 25735 25736 25962 25997 26230 26231 26363 26420 26568 26696 26762 26806 26841 26877 27022 27
026 27045 27053 27130 27153 27185 27355 27402 27592 27622 27630 27648 27651 27668 27702 27752 27857 27894 27978 28154 28
159 28226 28455 28464 28601 28671 28769 28851 28912 28954 28985 29096 29123 29176 29204 29322 29374 29397 29547 29824 29
902 29934 30048 30054 30070 30119 30130 30256 30299 30421 30436 30502 30528 30553 30563 30646 30691 30732 30749 30760 30
869 30954 30963 30979 31051 31074 31127 31209 31252 31303 31340 31353 31396 31422 31450 31619 31624 31754 31781 32121 32
230 32239 32262 32377 32419 32461 32470 32476 32557 32586 32619 32740

-----
Process exited after 32.5 seconds with return value 1
Presione una tecla para continuar . . .
```

500 – InsertionSort

```
C:\Users\Brain\Documents\EDA I\PI\Cuarto.exe
44 32656 32662 32672 32763

Cambios de indice i: 999
Numero de comparaciones entre elementos: 499500
Numero de cambio del indiceMenor: 5759
Numero de intercambios: 995
Numero de posibles intercambios: 999

Arreglo ordenado:
3 30 86 88 135 191 218 236 237 281 282 284 339 375 394 454 492 518 550 623 628 703 736 740 774 778 868 891 1005 1021 103
2 1033 1082 1082 1163 1165 1187 1195 1219 1231 1258 1324 1328 1347 1425 1433 1468 1472 1519 1569 1608 1621 1624 1732 173
4 1775 1777 1794 1812 1815 1817 1984 2001 2044 2082 2103 2137 2148 2150 2190 2273 2278 2289 2290 2352 2377 2417 2462 247
7 2493 2506 2534 2556 2558 2595 2636 2637 2650 2669 2750 2766 2859 2891 2898 2925 2946 2947 3056 3059 3142 3150 3151 318
3 3301 3342 3353 3393 3440 3478 3565 3584 3603 3681 3750 3769 3782 3847 3874 3895 3911 3955 3960 3986 3987 4072 4076 419
1 4268 4275 4313 4380 4386 4420 4438 4493 4505 4530 4649 4677 4678 4717 4795 4833 4846 4865 4872 4878 4907 4913 5038 504
9 5064 5070 5103 5137 5143 5193 5201 5201 5216 5218 5233 5286 5322 5346 5384 5391 5398 5400 5455 5484 5512 5520 5532 565
3 5661 5687 5715 5717 5816 5855 5888 6089 6103 6108 6116 6147 6167 6203 6247 6272 6274 6281 6337 6357 6383 6386 6427 652
6 6556 6557 6566 6568 6578 6622 6672 6686 6735 6736 6783 6803 6831 6839 6846 6940 6980 6996 7002 7025 7026 7163 7184 725
9 7261 7297 7297 7352 7361 7433 7498 7552 7618 7637 7641 7643 7678 7688 7730 7755 7786 7819 7830 7866 7980 7982 8073 808
1 8119 8157 8160 8211 8216 8249 8284 8286 8324 8355 8359 8368 8429 8443 8457 8493 8493 8546 8551 8556 8629 8671 8707 873
1 8741 8770 8779 8780 8780 8846 8885 8955 8982 8985 9029 9049 9134 9155 9156 9231 9426 9428 9525 9525 9804 9817 9818 984
4 9893 9916 9915 9940 9955 10104 10130 10137 10195 10195 10355 10371 10385 10391 10407 10433 10450 10463 10468 1048
44 32656 32662 32672 32763

-----
Process exited after 100.8 seconds with return value 1
Presione una tecla para continuar . . .
```

1000 – SelectionSort

```
55 18871 18877 18886 18996 19039 19041 19079 19109 19125 19139 19144 19145 19151 19152 19169 19174 19286 19298 19348 193
50 19437 19446 19459 19467 19472 19493 19517 19520 19606 19635 19744 19775 19801 19804 19838 19914 19916 19965 20051 200
68 20073 20107 20116 20125 20181 20193 20199 20216 20287 20344 20365 20462 20509 20521 20535 20612 20628 20657 20693 207
97 20711 20793 20793 20799 20823 20839 20874 20886 20932 20935 20945 21141 21142 21163 21223 21232 21326 21343 21358 213
60 21367 21426 21472 21489 21495 21566 21626 21646 21665 21665 21699 21710 21714 21736 21753 21760 21797 21838 21886 218
92 21899 21913 21937 21976 22005 22014 22025 22090 22149 22238 22277 22284 22300 22385 22446 22479 22546 22582 22617 226
81 22688 22712 22747 22764 22816 22904 22951 22956 22971 22994 23075 23102 23102 23109 23204 23208 23222 23234 23261 23295 233
24 23326 23357 23380 23410 23445 23489 23516 23521 23544 23556 23584 23640 23678 23707 23715 23729 23864 23957 23986 240
28 24048 24060 24084 24133 24160 24192 24219 24256 24287 24319 24340 24351 24351 24353 24355 24372 24388 24411 24411 244
17 24502 24527 24530 24550 24597 24599 24717 24735 24758 24760 24790 24931 24950 24952 24970 24976 25111 25110 25157 252
20 25333 25450 25450 25490 25555 25581 25644 25719 25734 25744 25796 25816 25827 25877 25959 25988 26024 26060 26080 261
45 26146 26153 26159 26170 26170 26201 26214 26215 26232 26248 26301 26329 26478 26559 26568 26590 26617 26626 26733 267
42 26747 26762 26830 26838 26892 26897 26904 26936 27006 27042 27049 27064 27071 27091 27116 27170 27220 27224 27263 272
83 27318 27343 27371 27376 27464 27492 27519 27528 27558 27595 27598 27605 27619 27641 27696 27768 27813 27847 278
67 27898 27931 27938 27965 27984 28001 28039 28098 28109 28145 28164 28244 28281 28346 28360 28400 28408 28477 28484 285
18 28545 28606 28663 28720 28737 28740 28751 28820 28870 28905 28958 28976 29001 29013 29018 29073 29077 29169 29179 292
22 29238 29241 29414 29475 29492 29502 29584 29632 29661 29666 29685 29696 29722 29774 29786 29820 29829 29830 29836 299
20 29965 30004 30019 30050 30085 30105 30109 30156 30159 30159 30169 30181 30208 30240 30246 30281 30321 30350 30394 304
97 30412 30482 30515 30536 30555 30560 30596 30601 30624 30696 30700 30795 30795 30797 30801 30935 30936 30964 31030 310
69 31156 31181 31182 31187 31204 31224 31225 31239 31313 31371 31434 31449 31494 31525 31535 31595 31666 31706 31713 317
63 31864 31899 31933 31961 32019 32043 32067 32096 32130 32256 32260 32282 32351 32429 32440 32446 32468 32471 32556 326
44 32656 32662 32672 32763

-----
Process exited after 100.8 seconds with return value 1
Presione una tecla para continuar . . .
```

1000 – SelectionSort

```
C:\Users\Brain\Documents\EDA I\PI\Cuatro.exe
2 32001 32025 32038 32172 32198 32342 32353 32380 32427 32515 32541 32560 32572 32583 32603 32626 32627 32638 32645 3265
7 32679 32727 32752

Numero de inserciones/comparaciones totales: 250001
Numero inserciones que llevaron a la posicion preliminar: 249002
Numero de inserciones en posicion final: 125000

Arreglo ordenado:
1 8 88 91 102 114 118 140 154 199 206 208 247 266 295 302 323 328 339 373 386 435 476 487 487 499 572 611 625 627 653 66
0 696 715 724 836 853 862 863 866 929 938 1028 1085 1085 1224 1246 1266 1296 1308 1324 1337 1337 1351 1386 1392 1412 150
1 1570 1572 1601 1606 1679 1692 1709 1728 1851 1858 1902 1911 1913 1987 2021 2041 2098 2201 2204 2236 2238 2292 2303 231
5 2352 2391 2411 2443 2518 2520 2522 2598 2701 2709 2737 2738 2748 2749 2752 2756 2758 2823 2837 2839 2881 2919 2943 296
1 3225 3292 3345 3354 3362 3378 3387 3401 3416 3437 3510 3579 3602 3608 3620 3621 3647 3653 3720 3736 3746 3769 3840 384
4 3872 3891 3940 3954 4034 4047 4073 4087 4145 4165 4195 4233 4233 4301 4303 4422 4460 4487 4525 4545 4722 4729 4732 477
5 4810 4825 4832 4908 4937 4980 4996 5014 5070 5072 5089 5145 5184 5237 5269 5308 5330 5347 5381 5391 5406 5438 5553 560
4 5657 5692 5696 5758 5812 5813 5820 5834 5839 5853 5871 5951 5954 5955 6057 6215 6285 6331 6364 6375 6378 6388 6410 645
4 6480 6482 6483 6548 6575 6601 6609 6690 6856 6938 6941 6956 6984 7029 7091 7147 7277 7371 7417 7435 7483 7510 7525 757
0 7645 7693 7758 7787 7808 7830 7832 7841 7980 8044 8089 8092 8141 8154 8173 8278 8292 8336 8344 8350 8388 8433 8637 865
9 8679 8730 8765 8864 8892 8892 8911 8971 8991 8993 9050 9128 9150 9157 9208 9214 9233 9239 9297 9345 9359 9381 9445 949
4 9516 9521 9529 9572 9647 9648 9654 9717 9762 9774 9814 9823 9864 9891 9894 9909 9910 9930 10153 10221 10256 10294 1030
5 10341 10375 10421 10427 10458 10520 10544 10574 10652 10653 10772 10775 10819 10823 10836 10863 10873 10917 10919 1096
7 11070 11154 11201 11217 11223 11255 11255 11400 11427 11450 11465 11521 11590 11603 11775 11785 11856 11912 11968 1200
2 12065 12096 12125 12127 12148 12199 12201 12209 12236 12387 12402 12404 12409 12420 12431 12482 12516 12526 12567 1257
2 12591 12678 12696 12725 12726 12764 12772 12778 12817 12904 12905 13004 13008 13036 13047 13049 13222 13236 13295 1333
0 13402 13510 13515 13520 13529 13649 13659 13698 13712 13726 13737 13746 13779 13812 13835 13848 13887 13893 13894 1390
1 13936 13938 13990 14019 14022 14071 14074 14121 14126 14146 14165 14168 14249 14253 14255 14301 14306 14317 14344 1437
4 14384 14441 14447 14461 14462 14480 14530 14621 14628 14665 14673 14692 14743 14779 14822 14872 14889 14933 14952 1499
4 15093 15093 15106 15129 15133 15143 15185 15234 15239 15239 15258 15292 15335 15358 15380 15419 15437 15526 15534 1554
8 15559 15616 15627 15666 15759 15817 15852 15963 15969 16010 16014 16131 16140 16179 16192 16203 16229 16357 1637
7 16495 16524 16525 16555 16561 16568 16623 16639 16639 16671 16697 16742 16755 16769 16822 16841 16854 16856 16882 1688
```

1000 – InsertionSort

```
C:\Users\Brain\Documents\EDA I\PI\Cuatro.exe
8 16890 16991 16991 17072 17086 17112 17184 17191 17202 17212 17220 17230 17268 17285 17377 17399 17439 17487 17489 1756
3 17607 17650 17695 17697 17705 17729 17746 17755 17766 17772 17826 17834 17966 17992 17998 17999 18046 18063 18082 1817
9 18209 18221 18227 18291 18292 18333 18372 18379 18383 18389 18409 18533 18585 18648 18660 18700 18732 18747 18782 1879
3 18800 18859 18868 18957 18969 18978 19051 19076 19080 19120 19215 19219 19246 19261 19276 19319 19347 19349 19385 1947
3 19502 19507 19646 19671 19676 19679 19680 19697 19727 19799 19803 19835 19857 19884 19892 19897 19953 19958 19958 1998
7 20005 20098 20123 20136 20154 20159 20240 20285 20299 20319 20332 20355 20357 20389 20392 20418 20440 20473 20565 2060
4 20615 20638 20648 20658 20670 20682 20696 20753 20755 20791 20803 20836 20863 20958 21039 21041 21041 21078 21083 2110
3 21130 21167 21204 21241 21424 21503 21513 21525 21536 21537 21555 21607 21657 21666 21693 21694 21747 21749 21804 2182
8 21842 21876 21902 21913 21951 21960 21971 21989 22078 22107 22187 22235 22282 22348 22388 22418 22421 22422 22451 2246
3 22475 22534 22538 22615 22634 22637 22670 22686 22694 22740 22775 22792 22793 22840 22869 22914 22946 22951 22978 2299
3 23041 23083 23101 23141 23161 23162 23167 23170 23204 23216 23219 23244 23298 23362 23362 23456 23575 23577 23577 2359
1 23592 23650 23650 23673 23713 23732 23734 23781 23834 23837 23868 23887 23926 23963 24021 24066 24068 24070 24202 2420
3 24283 24307 24321 24384 24461 24497 24532 24537 24551 24580 24607 24607 24663 24741 24741 24745 24752 24803 24803 2483
1 24854 24907 24936 24945 24951 25072 25078 25125 25179 25329 25358 25390 25397 25401 25409 25411 25428 25441 25477 2549
6 25535 25608 25612 25639 25654 25688 25693 25795 25810 25828 25832 25849 25928 26001 26020 26050 26050 26056 26065 2606
5 26078 26147 26155 26174 26216 26273 26323 26338 26371 26428 26490 26521 26557 26563 26654 26674 26684 26718 26718 2671
9 26741 26748 26790 26799 26842 26931 27044 27074 27123 27189 27206 27216 27316 27343 27376 27403 27412 27466 27561 2756
3 27591 27600 27629 27678 27694 27697 27745 27747 27749 27831 27833 27870 27887 27916 27925 27944 27955 28001 28061 2815
9 28180 28187 28198 28340 28359 28378 28402 28475 28500 28527 28567 28651 28663 28682 28716 28753 28755 28761 28785 2880
8 28841 28902 28927 28934 29045 29139 29164 29179 29179 29212 29248 29276 29309 29314 29333 29354 29369 29377 29407 2941
2 29436 29441 29506 29510 29549 29556 29607 29617 29675 29723 29729 29761 29815 29828 29863 29867 29938 29951 29998 3002
4 30038 30046 30084 30097 30133 30167 30197 30209 30213 30242 30257 30261 30330 30391 30395 30401 30462 30474 30475 3052
7 30536 30583 30595 30642 30659 30660 30757 30845 30855 30929 30956 30957 30961 31014 31050 31051 31065 31075 31079 3108
3 31087 31092 31106 31116 31127 31177 31372 31392 31452 31528 31555 31556 31687 31755 31816 31837 31865 31885 31961 3199
2 32001 32025 32038 32172 32198 32342 32353 32380 32427 32515 32541 32560 32572 32583 32603 32626 32627 32638 32645 3265
7 32679 32727 32752

-----
Process exited after 94.64 seconds with return value 1
Presione una tecla para continuar . . .
```

1000 - InsertionSort

Conclusiones.

Todo lo bueno tiene un final por lo que este reporte igual, los objetivos se cumplieron, es decir, logre identificar las características de los ordenamientos de Selección y de Inserción, por medio de la realización de 4 ejercicios que consistían en análisis, codificación, y desarrollo de los algoritmos de ordenamiento, reforcé lo aprendido en las clases, además recordé bastantes cosas de la codificación. Por último, comprendí la importancia de las implementaciones o pruebas de estrés de los algoritmos, para el completo entendimiento de las cosas, ya que la noción de la complejidad en los algoritmos era algo teórico y se volvió más práctico, eso me ayudo bastante a comprender mejor el funcionamiento de los algoritmos.