

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1
SECCIÓN E
PRIMER SEMESTRE 2022
ING. NEFTALÍ DE JESÚS ALDANA CALDERÓN
AUX. SERGIO FERNANDO OTZOY GONZALEZ

SEGUNDA PRÁCTICA DE LABORATORIO

ORDENAMIENTOS

CONTENIDO

Objetivos	1
General	1
Específicos	1
Descripción.....	1
Flujo de la aplicación.....	2
Inicio	2
Ordenar.....	3
Archivo de entrada.....	1
Reportes	1
Consideraciones finales	2
Requerimientos para el desarrollo del proyecto	3

OBJETIVOS

GENERAL

- Usar hilos, manejo de archivos, algoritmos de ordenación y programación orientada a objetos para desarrollar una solución informática compleja con el lenguaje de programación Java.

ESPECÍFICOS

- Aplicar conceptos de programación orientada a objetos para crear una aplicación de ventanas en Java.
- Comprender la ejecución de tareas en paralelo implementando hilos en java.
- Comprender el manejo de archivos en java.
- Implementar algoritmos de ordenamiento.

DESCRIPCIÓN

En esta práctica el estudiante deberá determinar la velocidad y cantidad de pasos que un algoritmo de ordenamiento realiza para ordenar un set de datos. Para realizar esto, se deberá usar una librería que genera una gráfica de barras para visualizar paso a paso como el algoritmo ordena ascendente o descendentemente el set de datos. Al finalizar esta simulación, se deberá generar un informe HTML con los resultados.

FLUJO DE LA APLICACIÓN

INICIO

Antes de iniciar la simulación, el usuario deberá cargar la información desde un archivo separados por comas.

- **Examinar.** Se desplegará un cuadro de diálogo desde donde el usuario podrá navegar y seleccionar un archivo. O bien, podrá ingresar manualmente la ruta del archivo en el campo de texto *Ruta de archivo*.
- **Generar gráfica.** Se deberá especificar el título de la gráfica. Al presionar este botón la información del archivo seleccionado en el paso anterior se leerá y se creará la gráfica que se usará para visualizar la simulación de la ordenación.

Practica 2

Ruta_de_archivo

Examinar...

Título de la gráfica

Generar gráfica

00:00
Cronómetro

00
Pasos

● Ascendente
● Descendente
● Algoritmo 1
● Algoritmo 2
● Algoritmo 3

Ordenar

Ilustración 1. Ejemplo de ventana principal

ORDENAR

Para visualizar la simulación del ordenamiento se deberá seleccionar el sentido de ordenación y el tipo de algoritmo a usar. Por último, se presionará **Ordenar** para iniciar la simulación.

Al iniciar la ordenación se deberá visualizar en todo momento un cronómetro y un contador de pasos. Ambos deberán iniciar al presionar **Ordenar** y ambos deberán detenerse cuando el algoritmo de ordenación finalice.

En el cronómetro se deberá visualizar minutos, segundos y milisegundos con el formato **00:00.000**. En pasos se deberá visualizar la cantidad de movimientos que realizó el algoritmo para ordenar los valores.

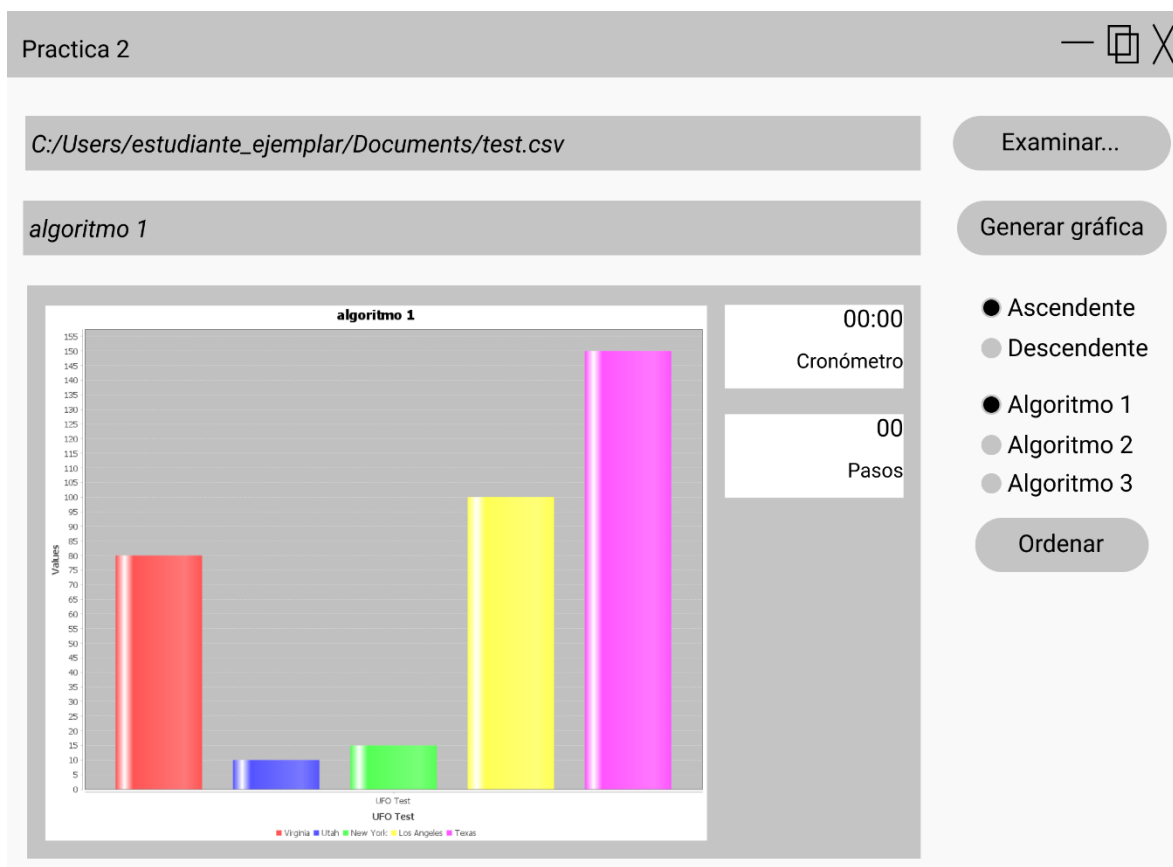


Ilustración 2. Ejemplo de ventana principal con gráfica generada

Durante la simulación se deberá visualizar paso a paso, cómo la gráfica está siendo ordenada:

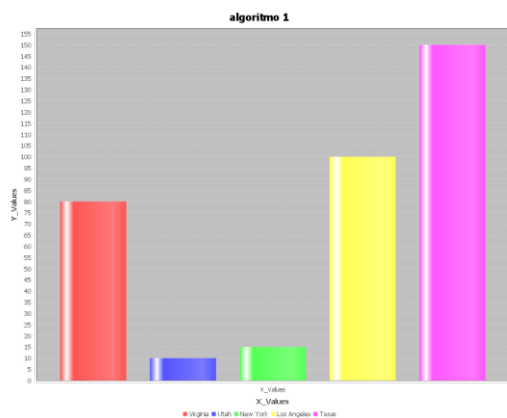


Ilustración 3. Paso 1 al ordenar

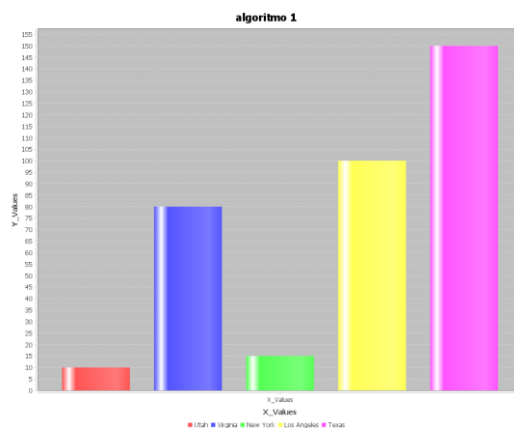


Ilustración 4. Paso 2 al ordenar

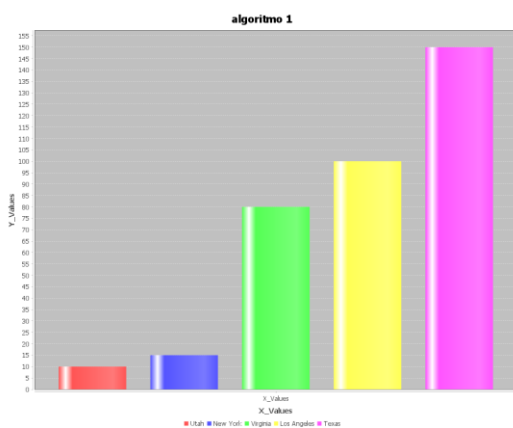


Ilustración 5. Paso 3 al ordenar

Al finalizar, el cronómetro y el contador de pasos deberá detenerse y la gráfica deberá estar totalmente ordenada. En esto punto, la aplicación quedará lista para simular otro ordenamiento con el mismo set de datos o bien cargar otro archivo.

ARCHIVO DE ENTRADA

Se usará un archivo separado por comas como archivo de entrada.

	A	B
1	X_Values	Y_Values
2	Virginia	80
3	Utah	10
4	New York	15
5	Los Angeles	100
6	Texas	150

Ilustración 4. Ejemplo archivo de entrada en formato de tabla

```
X_Values,Y_Values
Virginia,80
Utah,10
New York,15
Los Angeles,100
Texas,150
```

Ilustración 5. Ejemplo archivo de entrada como texto plano

En la primera fila estarán los encabezados que corresponderán a los ejes X y Y, respectivamente. En las filas consecutivas se encontrarán los valores para ambos ejes de la gráfica.

REPORTES

Al finalizar la simulación del ordenamiento se generará un archivo HTML en el mismo lugar en dónde se encuentra el archivo .jar. Este archivo deberá contener la siguiente información:

- Nombre y carné del estudiante
- Nombre del algoritmo usado
- Tiempo transcurrido
- Cantidad de pasos o movimientos
- En una tabla, los datos leídos desde el archivo csv (datos no ordenados)
- La gráfica inicial (datos no ordenados)
- En otra tabla, los datos ordenados
- La gráfica final con los datos ordenados

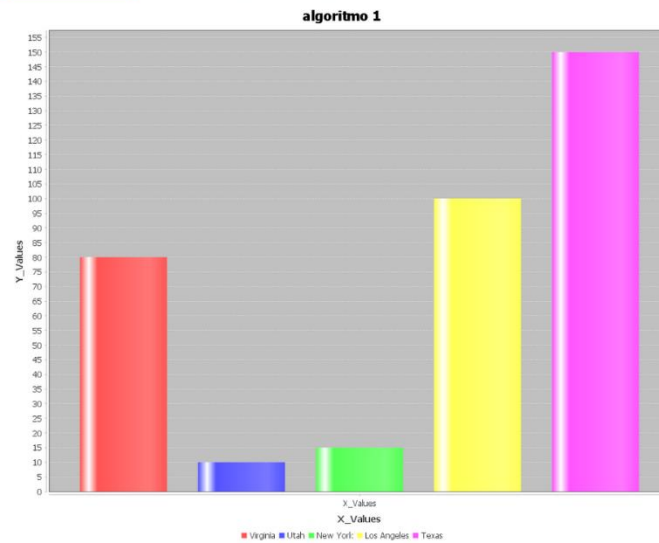
El reporte llevará como nombre, el título de la gráfica y la hora en la que fue generado: ***título_hora_minuto_segundo.html***

Algoritmo 1

Tiempo	00:05.999
Pasos	2
Orden	Ascendente

Datos desordenados

X_Values	Virginia	Utah	New York	Los Angeles	Texas
Y_Values	80	10	15	100	150



Datos ordenados

X_Values	Utah	New York	Virginia	Los Angeles	Texas
Y_Values	10	15	80	100	150

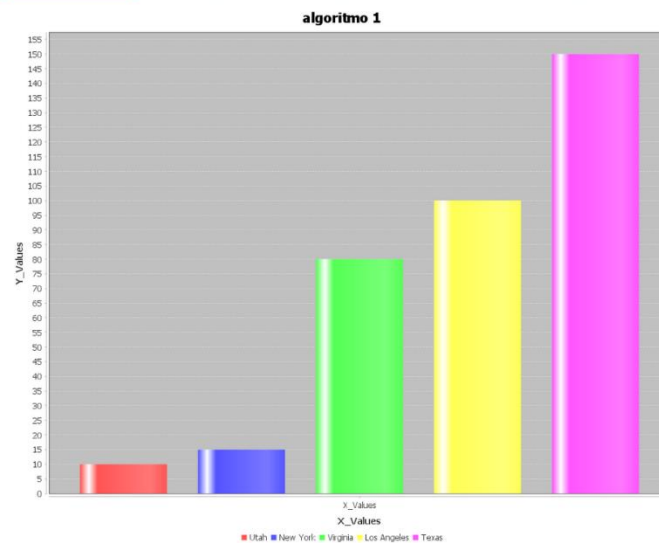


Ilustración 6. Ejemplo de reporte HTML

CONSIDERACIONES FINALES

1. Se deberá implementar 3 de cualquiera de estos algoritmos de ordenación:
 - a. Bubblesort
 - b. Shellsort
 - c. Quicksort
 - d. Insertion sort
 - e. Merge sort

El estudiante deberá investigar el funcionamiento de los algoritmos que implementará, teniendo la medida y seriedad del caso para adaptarlo a su propia solución en código.

2. Se recomienda usar la librería JFreeChart para generar la gráfica dentro de la aplicación y para mostrarla en el reporte HTML. La usará un gráfico de barras para mostrar la información.
3. El tamaño de los arreglos que manejarán la información leída de los archivos dependerá de la cantidad de datos (filas, exceptuando el encabezado) que esos archivos contengan.
4. Las imágenes usadas en este enunciado son un ejemplo. El diseño de la interfaz y del reporte HTML queda a discreción del estudiante.
5. Toda interacción con la aplicación deberá ser por medio de la interfaz gráfica.
6. Deberá usar hilos para generar la solución del problema propuesto.

REQUERIMIENTOS PARA EL DESARROLLO DEL PROYECTO

- Documentación
 1. Manual técnico. Descripción de los métodos o funciones usados en el programa a través de capturas del código, pseudocódigo o diagramas de flujo.
 2. Manual de usuario. Describir cómo funciona la aplicación y como el usuario interactúa con ella, incluir capturas de pantalla de la aplicación.
- Restricciones
 1. La aplicación deberá ser desarrollada completamente en Java
 2. No se permite el uso de estructuras que implemente la interface `AbstractCollection` de Java (`ArrayList`, `LinkedList`, etc)
 3. El IDE por utilizar queda a discreción del estudiante.
 4. Copias obtendrán nota 0 y serán reportadas a la Escuela de Ciencias y Sistemas.
 5. Se calificará únicamente sobre el JAR enviado por lo que es obligatorio generarlo.
 6. Cualquier otra librería que el estudiante quiera usar se deberá consultarlo primero con el auxiliar.
- Entregables
 1. **Fecha de entrega:** 20/03/2022 antes de las 23:59 PM. No se aceptarán entregas luego de esa hora. No se aceptarán entregas por correo.
 2. Adjuntar el código fuente, el archivo `.jar` y los manuales en un archivo comprimido con el siguiente formato: `[IPC1]Practica2_carnet`.
 3. Subir el archivo comprimido en la tarea asignada en UEDI.