

TABLA DE CONTENIDO

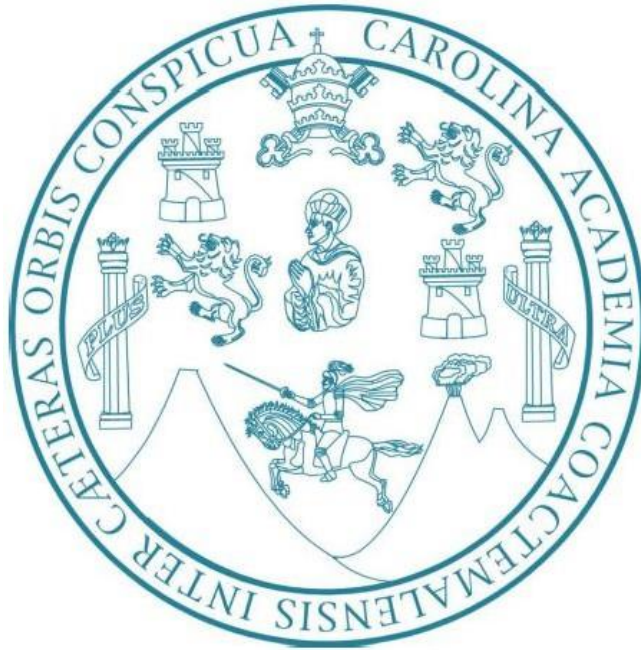
Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la Programación y
Computación 1 Sección D

Catedrático: Ing. Herman Igor Veliz Linares

Tutor académico: Carlos Roberto Jiménez Pérez



PRÁCTICA 2

TABLA DE CONTENIDO

- OBJETIVOS** 1
 - Generales 1
 - Específicos 1
- DESCRIPCION GENERAL**..... 2
 - Aplicación 2
 - Inicio 2
 - Formato del Archivo de Entrada 3
 - Opciones De Ordenamiento 4
 - Proceso De Ordenamiento 5
 - Reportes..... 7
- REQUERIMIENTOS PARA EL DESARROLLO DE LA PRÁCTICA:** 9
 - Documentacion: 9
 - Consideraciones: 9
 - Restricciones:..... 9
 - Habilidades Por Evaluar:..... 9
 - Entrega: 9

PRÁCTICA 2 - IPC1

OBJETIVOS

Generales

- ✓ Familiarizar al estudiante con el lenguaje de programación JAVA.
- ✓ Que el estudiante aplique los conocimientos adquiridos en el curso de Introducción a la Programación y computación 1.
- ✓ Elaborar la lógica para presentar una solución a la propuesta planteada.

Específicos

- ✓ Uso del lenguaje de programación Java como herramienta de desarrollo de software.
- ✓ Construcción de aplicaciones simples con interfaz gráfica.
- ✓ Implementación de sentencias de control, ciclos y vectores.
- ✓ Manejo de los algoritmos de ordenamiento (bubble sort, quicksort, shell sort).
- ✓ Implementación de hilos para la realización de múltiples tareas en paralelo.

PRÁCTICA 2 - IPC1

DESCRIPCION GENERAL

En la actualidad el manejo de datos es fundamental para las empresas líderes de cualquier mercado, ya que estos son necesarios para realizar estudios y mapeos que permitan tomar decisiones clave que buscan minimizar costos y/o maximizar las ganancias de esta compañía.

Debido a ello y sabiendo que visualizar los datos de forma gráfica es siempre una manera más fácil de interpretar estos, se le solicita cree una solución con interfaz gráfica, capaz de leer datos de un archivo con formato csv, para luego generar una gráfica representativa de los mismos, y poder aplicar distintos algoritmos de ordenamiento siendo posible clasificar los datos de mayor a menor y viceversa generando un reporte con información clave al final de cada proceso.

Los detalles se especifican a continuación

APLICACIÓN:

INICIO

En primera instancia se le solicitará al usuario la ruta del archivo csv que se desea procesar, así como un título para la gráfica.

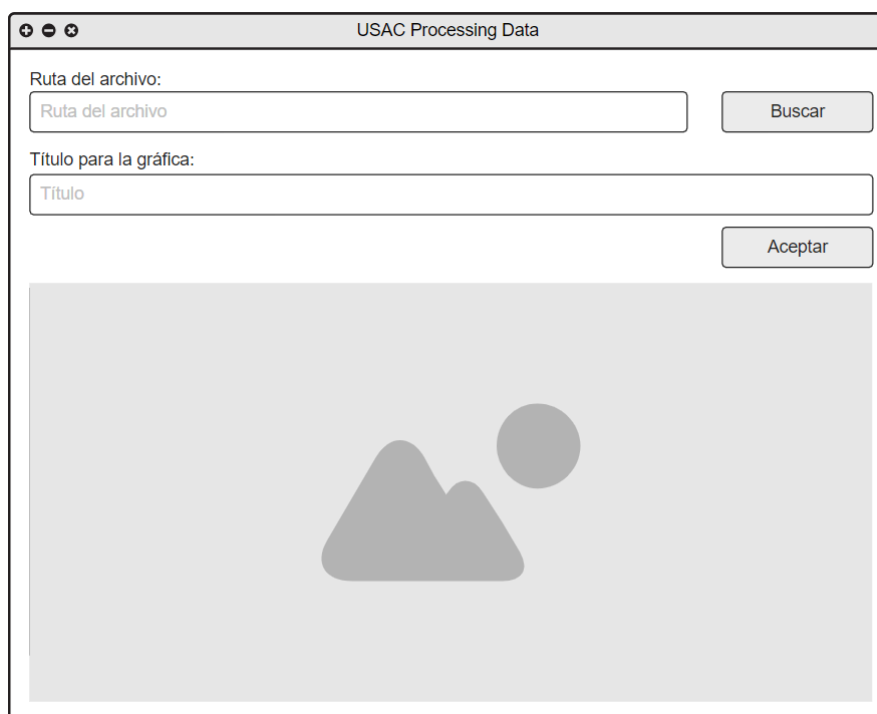


Figura 1: Interfaz con fines ilustrativos, el estudiante puede implementar un diseño diferente.

- **Botón Buscar:** Despliega un cuadro de diálogo desde donde el usuario podrá navegar por sus archivos y buscar el documento con formato csv que desea analizar. También es posible ingresar la ruta manualmente utilizando el campo de texto que se encuentra al lado.
- **Botón Aceptar:** Al pulsar el botón aceptar, la solución procede a leer el archivo que únicamente constará de dos columnas, la primera columna tendrá las categorías y la segunda el conteo total por categoría.

PRÁCTICA 2 - IPC1

FORMATO DEL ARCHIVO DE ENTRADA

El archivo como tal tendrá el siguiente formato:

	A	B
1	UFO Shape	Count
2	Blimp	597
3	Boomerang	1502
4	Bullet	10
5	Bullet/Misil	825
6	Changing	2090
7	Chevron	1505
8	Cigar	3146
9	Circle	1505
10	Cone	767

Figura 2: Archivo como tabla.

```
file.csv
1 UFO Shape,Count
2 Blimp,597
3 Boomerang,1502
4 Bullet,10
5 Bullet/Misil,825
6 Changing,2090
7 Chevron,1505
8 Cigar,3146
9 Circle,1505
10 Cone,767
```

Figura 3: Archivo como texto plano.

En la primera línea estarán los encabezados que deberá utilizar para nombrar los ejes "X" y "Y" de la gráfica, quedando la vista del programa de la siguiente manera:

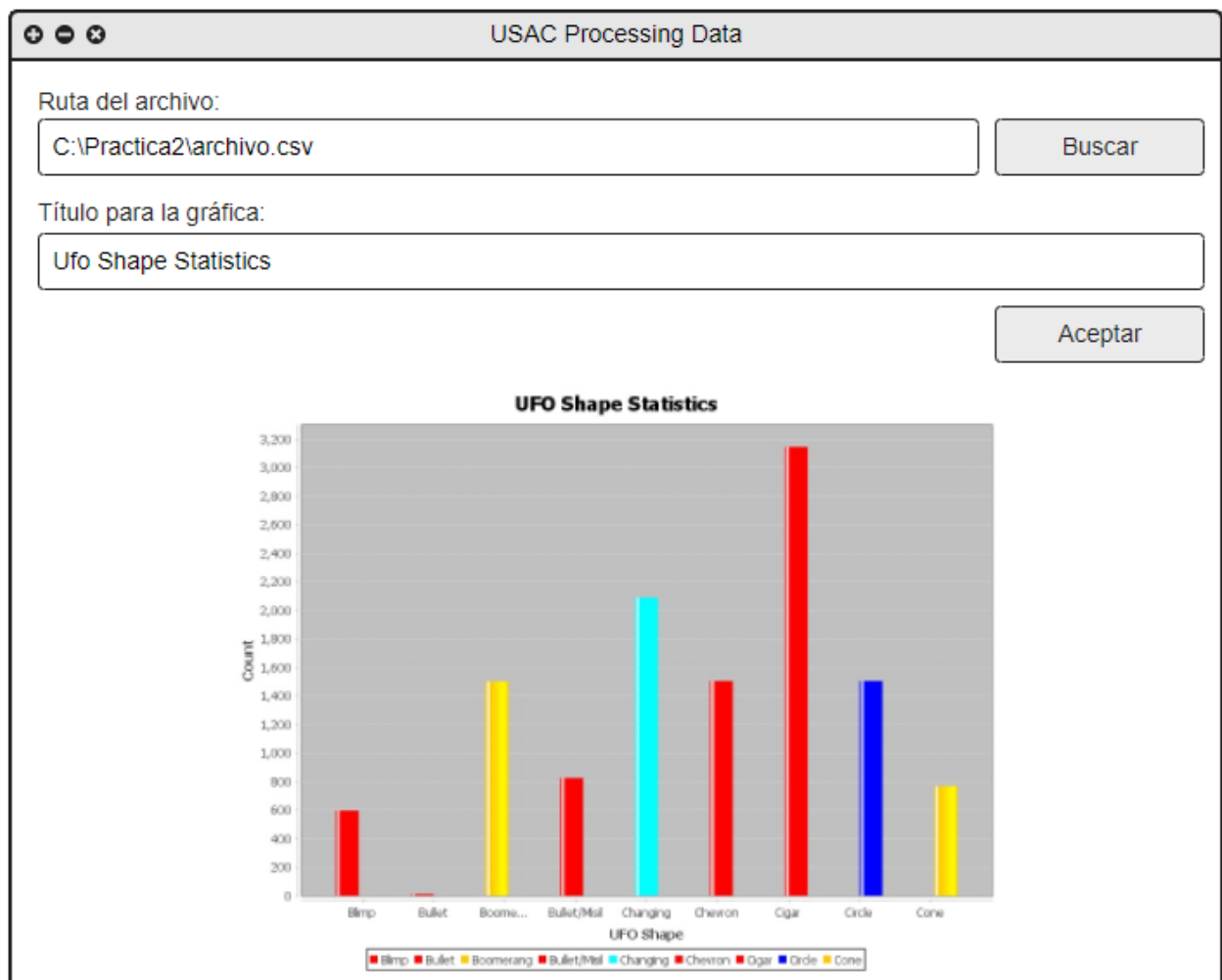


Figura 4: Grafica resultante del archivo de entrada.

PRÁCTICA 2 - IPC1

OPCIONES DE ORDENAMIENTO

Una vez cargado el archivo y sea visible la gráfica resultante, se deberán desplegar opciones para poder realizar el ordenamiento de los datos.

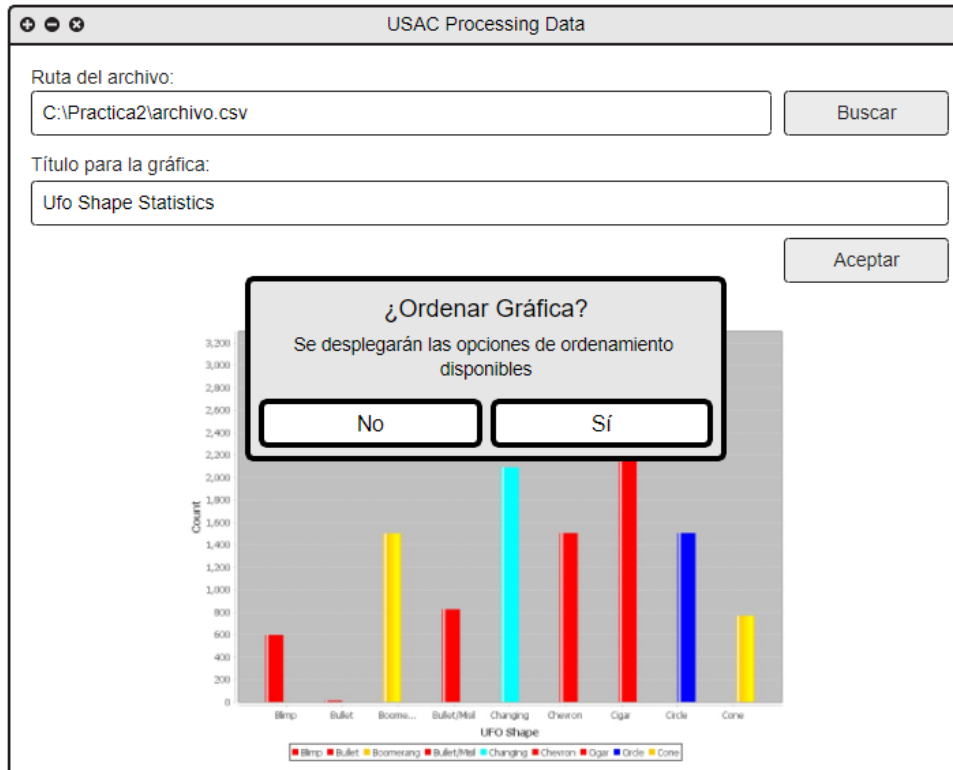


Figura 5: Interfaz con fines ilustrativos del flujo del programa, el diseño puede variar.

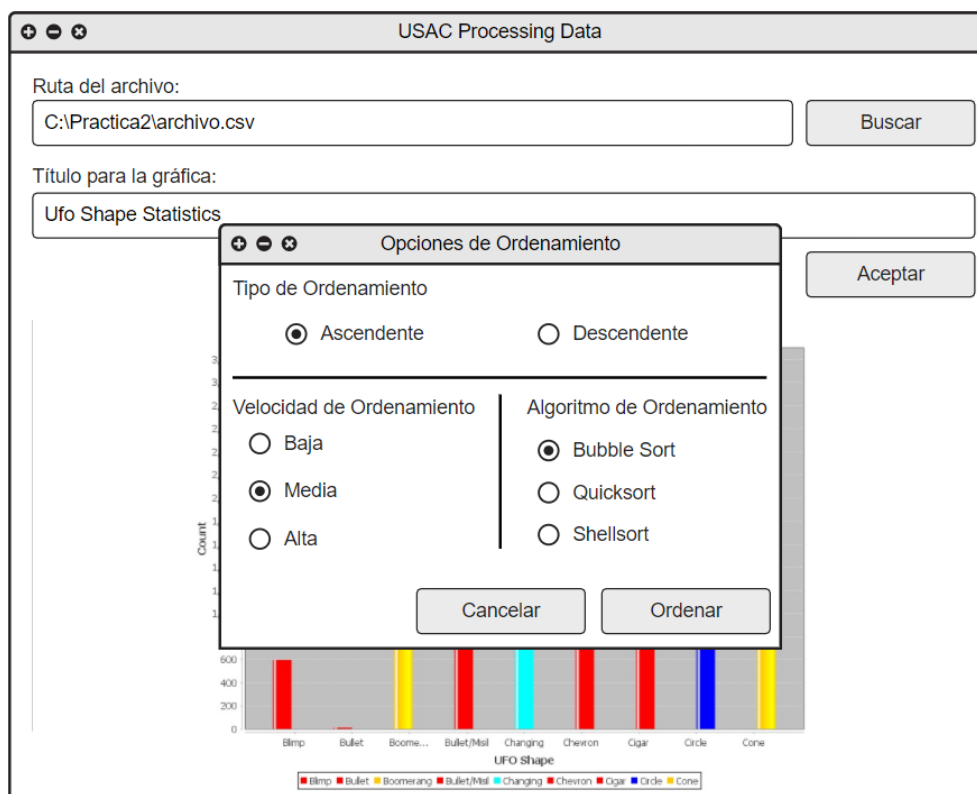


Figura 6: Ventana con las opciones de ordenamiento

PRÁCTICA 2 - IPC1

- **Tipo de Ordenamiento:** El usuario podrá elegir si desea que los datos sean ordenados ascendente o descendente de izquierda a derecha.
- **Velocidad de Ordenamiento:** El usuario será capaz de definir la velocidad a la que ocurrirá el ordenamiento. Queda a discreción del estudiante la definición de velocidad alta, media o baja, sin embargo, debe notarse la diferencia entre cada una de estas.
- **Algoritmo de Ordenamiento:** El usuario podrá elegir el algoritmo mediante el cual se realizará el ordenamiento, en este caso tiene las opciones Bubble Sort, Quicksort y Shellsort.

PROCESO DE ORDENAMIENTO

Una vez definidos todos los parámetros de ordenamiento, el programa iniciará con el proceso. Para ello, irá mostrando paso a paso cómo es aplicado el algoritmo seleccionado a la velocidad elegida (utilizar hilos). Queda a discreción del estudiante si realiza este proceso en la misma ventana principal, o en una ventana diferente, sin embargo, debe poder visualizarse los siguiente:

- **Nombre del algoritmo a aplicar.**
- **Velocidad seleccionada**
- **Si el ordenamiento se hará de forma ascendente o descendente**
- **El tiempo que el algoritmo lleva en ejecución. (formato "minutos: segundos: milisegundos" 00:00:000)**
- **Defina un algoritmo para calcular el tiempo únicamente tomando el valor de los milisegundos, el primero en enviar este algoritmo al auxiliar tiene la bonificación.**
- **Cantidad de pasos o movimientos realizados**

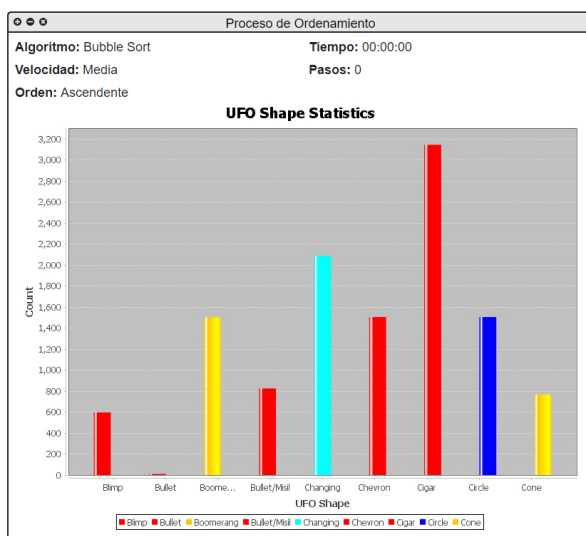


Figura 7: Gráfica sin ordenar.

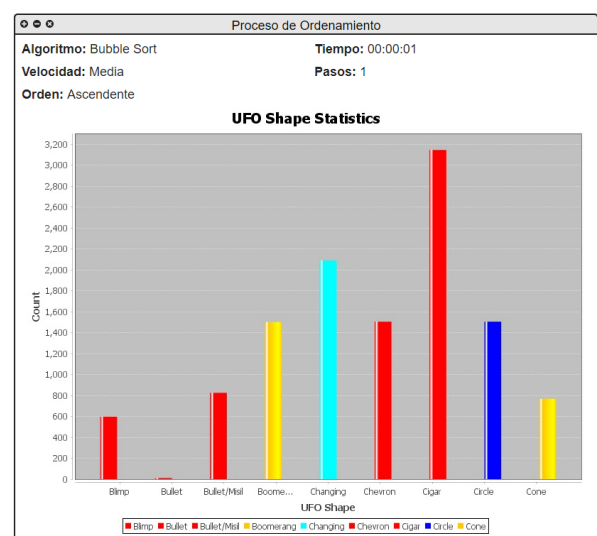


Figura 8: Proceso de ordenamiento por Bubble Sort paso 1

PRÁCTICA 2 - IPC1

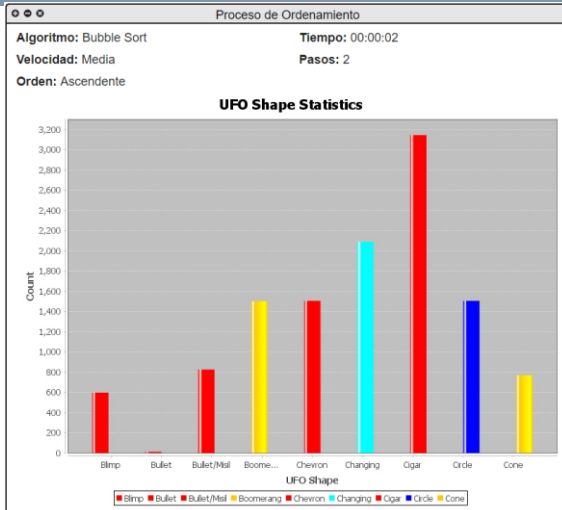


Figura 9: Proceso de ordenamiento por Bubble Sort paso 2

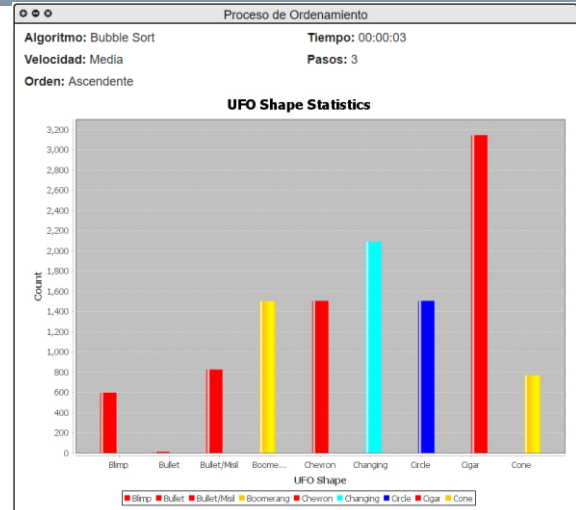


Figura 10: Proceso de ordenamiento por Bubble Sort paso 3

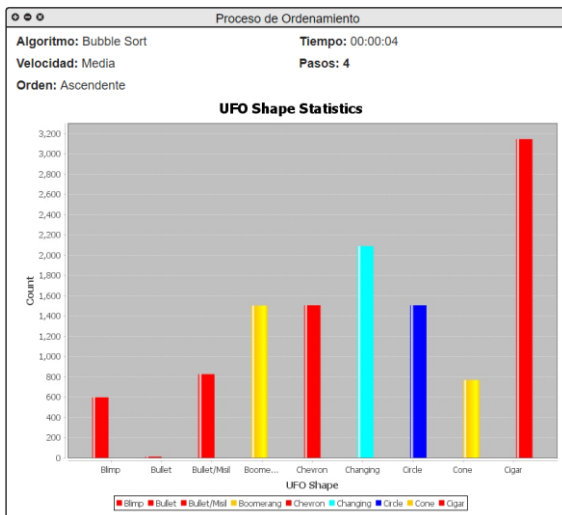


Figura 11: Proceso de ordenamiento por Bubble Sort paso 4

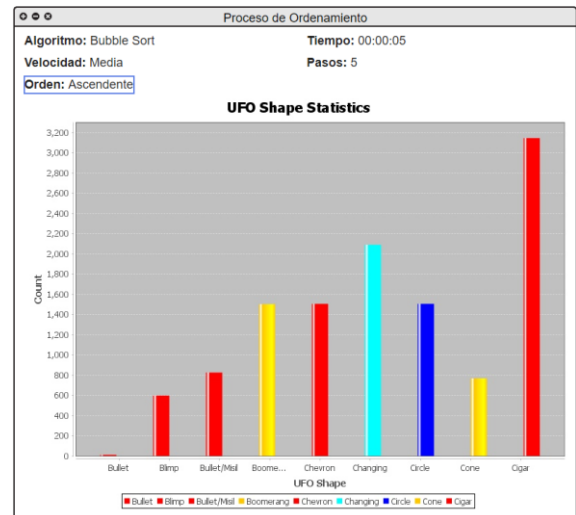


Figura 12: Proceso de ordenamiento por Bubble Sort paso 5

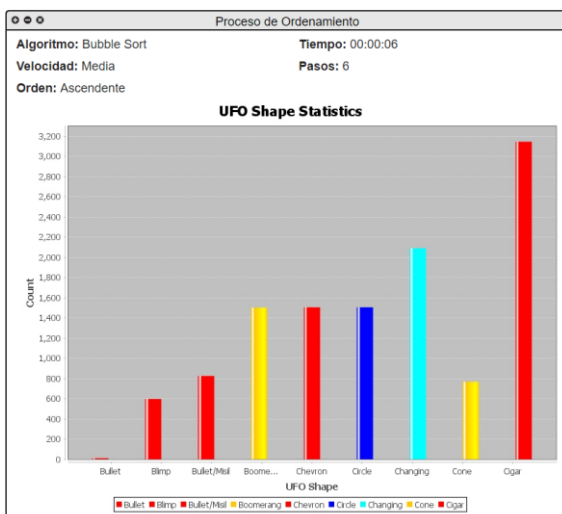


Figura 13: Proceso de ordenamiento por Bubble Sort paso 6

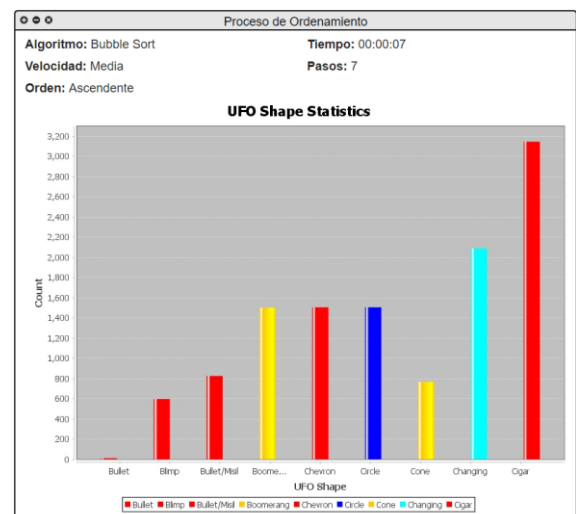


Figura 14: Proceso de ordenamiento por Bubble Sort paso 7

PRÁCTICA 2 - IPC1

Este procedimiento continuará hasta que la gráfica quede totalmente ordenada según los parámetros seleccionados. Al llegar a este punto el programa deberá generar un reporte (los detalles en la siguiente sección) y luego quedará listo para procesar otro archivo.

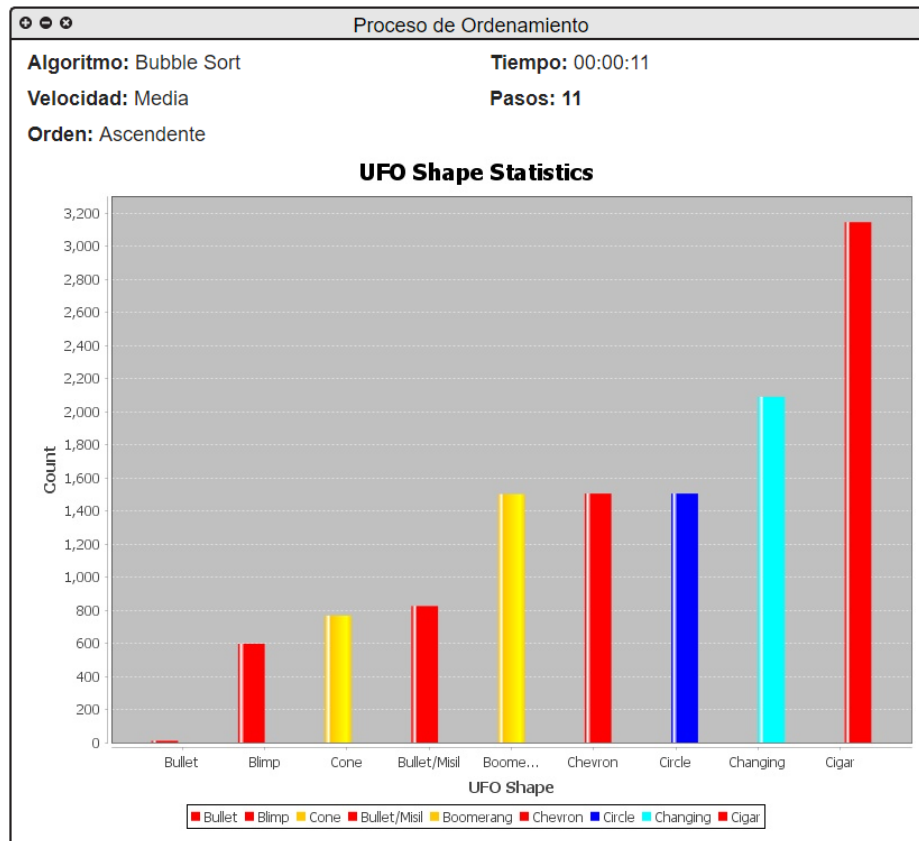


Figura 15: Gráfico ordenado de manera ascendente mediante el algoritmo bubble sort.

REPORTES

Una vez el proceso de ordenamiento termine, se deberá generar un archivo HTML, apoyado en CSS para darle estilo, que contenga la siguiente información:

- **Nombre y carné del estudiante que generó el reporte.**
- **Algoritmo de ordenamiento utilizado, velocidad a la que se realizó el ordenamiento, si el ordenamiento fue ascendente o descendente, el tiempo que tomó el ordenamiento y la cantidad de pasos que realizó.**
- **Dato mínimo y dato máximo dentro del conjunto de datos.**
- **Los datos leídos del archivo csv en forma de tabla (no ordenados).**
- **Gráfica inicial de donde se partió (con los datos no ordenados).**
- **Los datos ordenados en forma de tabla.**
- **Gráfica final con los datos ya ordenados**

Queda a discreción del estudiante la organización visual de estos datos, sin embargo, el reporte debe ser claro, ordenado y contar con cada punto expuesto anteriormente.

PRÁCTICA 2 - IPC1

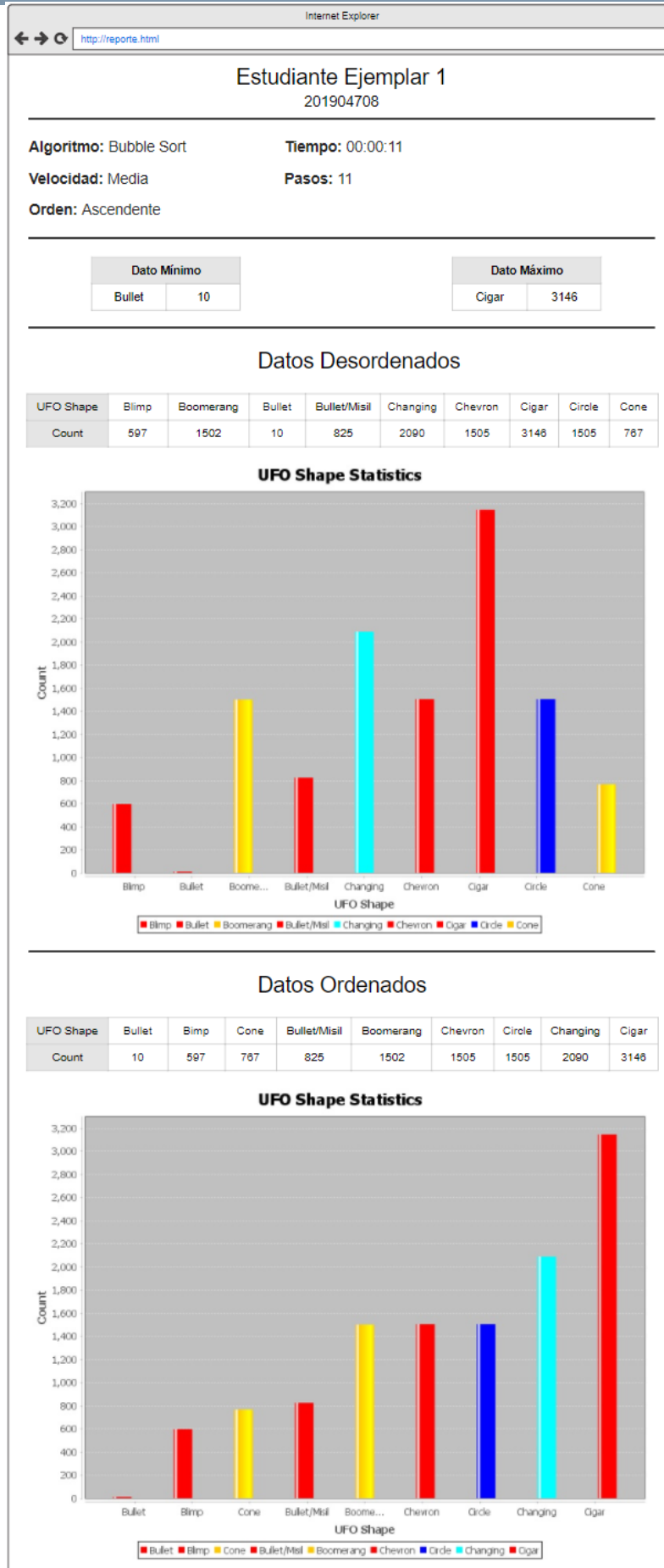


Figura 16: Reporte en HTML (ejemplo)

PRÁCTICA 2 - IPC1

REQUERIMIENTOS PARA EL DESARROLLO DE LA PRÁCTICA:

DOCUMENTACION:

- ✓ Manual Técnico (Descripción de los métodos creados) en PDF.
- ✓ Manual de Usuario (Cómo funciona la aplicación y como el usuario interactúa con ella) en PDF.

CONSIDERACIONES:

- ✓ Toda interacción con la aplicación debe ser por medio de interfaz gráfica de usuario.
- ✓ La distribución de los componentes gráficos queda a discreción del estudiante. Las ventanas propuestas solo deben ser una guía para las interfaces que decidan realizar.

RESTRICCIONES:

- ✓ La aplicación debe ser desarrollada en el lenguaje de programación Java.
- ✓ No se permite el uso de estructuras que implemente Java (List, ArrayList, LinkedList, etc.).
- ✓ No se permite utilizar código copiado o bajado de Internet.
- ✓ No se permite hacer uso de la herramienta de los IDE para construir interfaces gráficas arrastrando componentes, sino únicamente las librerías de API gráficas que proporciona Java para las ventanas que se solicitan (AWT y Swing).
- ✓ El IDE por utilizar queda a discreción del estudiante (se recomienda el uso de NetBeans)
- ✓ Copias obtendrán una nota de 0 y reporte a la Escuela de Ciencias y Sistemas.
- ✓ Se calificará únicamente del JAR enviado, por lo que es obligatorio generarlo

HABILIDADES POR EVALUAR:

- ✓ Uso de variables globales y locales.
- ✓ Uso de memoria estática
- ✓ Uso de estructuras de control y de selección
- ✓ Uso correcto de los arreglos y matrices.
- ✓ Conocimientos sobre sistemas computacionales
- ✓ Habilidad para analizar y sintetizar información
- ✓ La habilidad de comprender y realizar diagramas
- ✓ Habilidad para resolver problemas.
- ✓ Uso adecuado de hilos

ENTREGA:

- ✓ **FECHA DE ENTREGA:** 06/10/2020 antes de las 23:59 PM. No se aceptarán entregas a partir de esa hora.
- ✓ Adjuntar código fuente, archivo ejecutable JAR y documentación en un archivo .zip o rar con el formato: [IPC1]Practica2_carnet.rar. Ejemplo: [IPC1]Practica2_201900000.rar
- ✓ Subir el archivo comprimido en la tarea asignada en UEDi.