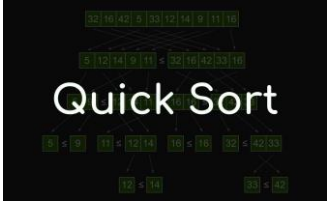
	<i>Ficha</i>		<i>Proyecto</i>	
	2236347		QuickSort	
	<i>Fecha</i>		<i>Versión</i>	<i>Código</i>
	27/10/2021		1.2	IP-01

Informe escrito, proyecto organización de números

Proyecto QuickSort

Versión: 1.2

	Ficha	Proyecto	
	2236347	QuickSort	
	Fecha	Versión	Código
	27/10/2021	1.2	IP-01

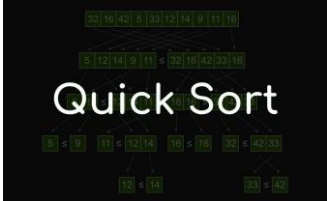
HISTORIAL DE REVISIÓN

VERSIÓN	ELABORACIÓN		REVISIÓN	
	Fecha	Responsable	Fecha	Responsable
1.0	17/10/2021	Alejandro Giraldo, Jhon Faber Muñoz	17/10/2021	Alejandro Giraldo, Jhon Faber Muñoz
1.1	23/10/2021	Alejandro Giraldo, Jhon Faber Muñoz	23/10/2021	Alejandro Giraldo, Jhon Faber Muñoz
1.2	27/10/2021	Alejandro Giraldo, Jhon Faber Muñoz	27/10/2021	Alejandro Giraldo, Jhon Faber Muñoz

CAMBIOS RESPECTO A LA VERSIÓN ANTERIOR

VERSIÓN	MODIFICACIÓN RESPECTO VERSIÓN ANTERIOR
1.0	Creación del documento y especificación de requisitos.
1.1	Se añade el diagrama de flujo.
1.2	Se agregan líneas de código totales escritas, conclusión del proyecto y resultado final.

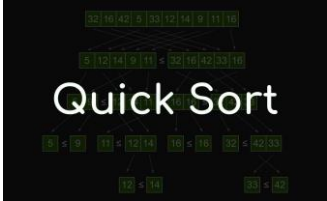
1. Requisitos.

	Ficha	Proyecto	
	2236347	QuickSort	
	Fecha	Versión	Código
	27/10/2021	1.2	IP-01

- El programa debe correr sin problemas sobre la consola.
- El programa debe leer un número natural ingresado por teclado.
- El programa debe generar una cantidad N de números enteros aleatorios entre -1000 y 999. Donde N es el número ingresado anteriormente.
- Los números generados deben ser almacenados en un archivo de texto. El programa debe de leer un archivo de texto con números, los cuales los almacena en un vector.
- El programa debe de iniciar un cronómetro a la hora de empezar a ordenar los números.
- Los números almacenados en el vector deben de ser organizados de menor a mayor utilizando el método conocido como QuickSort.
- Los números ya organizados se deben almacenar y ser visualizados en pantalla.
- Cuando los números se hayan almacenado e impreso en pantalla, el cronómetro debe detenerse.
- El programa debe de interpretar el tiempo dado por el cronómetro e igualmente presentarlo en formato HH:MM:SS.ssss. Donde HH representan las horas, MM los minutos, SS los segundos y ssss los milisegundos.

2. Algoritmo de ordenamiento QuickSort

QuickSort es un algoritmo recursivo o con posibilidades de hacerse iterativo que se basa en la ideología “Divide y vencerás”. Este algoritmo se basa en elegir un número cualquiera de los elementos a ordenar el cual llamaremos pivote. El pivote será un elemento de referencia por el cual pasarán los demás elementos para definir su orden, a la izquierda irán los menores a él y a la derecha los mayores o viceversa. Los elementos quedarán entonces en dos subgrupos, uno con los elementos mayores al pivote y otro con los elementos menores al pivote. Este proceso se repite en cada subgrupo hasta que la cantidad de elementos no sea mayor a 1.

	Ficha		Proyecto	
	2236347		QuickSort	
	Fecha		Versión	Código
	27/10/2021		1.2	IP-01

En esta URL se puede ver la utilización de este algoritmo:

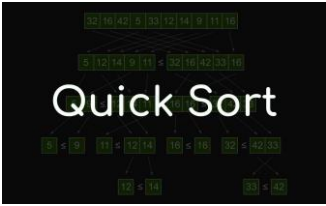
https://upload.wikimedia.org/wikipedia/commons/6/6a/Sorting_quicksort_anim.gif

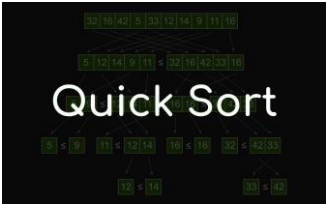
3. Diagrama de flujo.

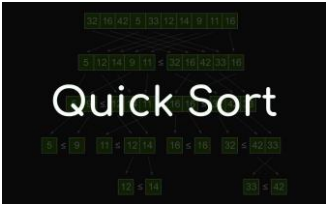
[Diagrama de flujo.drawio.png](#)

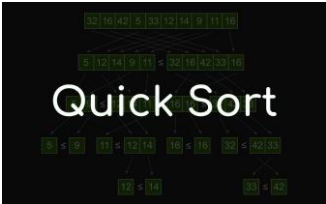
4. Detalles técnicos.


- 4.1. El proyecto se realizó en Java con ayuda de editores de texto como Visual Studio Code y Eclipse.
- 4.2. La carpeta del programa contiene 4 subcarpetas:
 - a. bin: Archivos binarios '.class' que son las clases ya compiladas de Java.
 - b. lib: Librerías utilizadas en el proyecto.
 - c. numeros: Archivos de texto con los números generados y los números ordenados.
 - d. src: Archivos de clase '.java', estas son las clases sobre las que se trabaja, para su ejecución es necesario compilarlas.
- 4.3. Se utilizó una estructura de clases, dividiendo en cada clase una funcionalidad específica y uniéndose todas en una clase App.
- 4.4. Para correr el programa, deberá tener Java versión +8 correctamente instalado en su máquina. Se recomienda su ejecución desde Visual Studio Code.
 - 4.4.1. Correr desde Visual Studio Code:

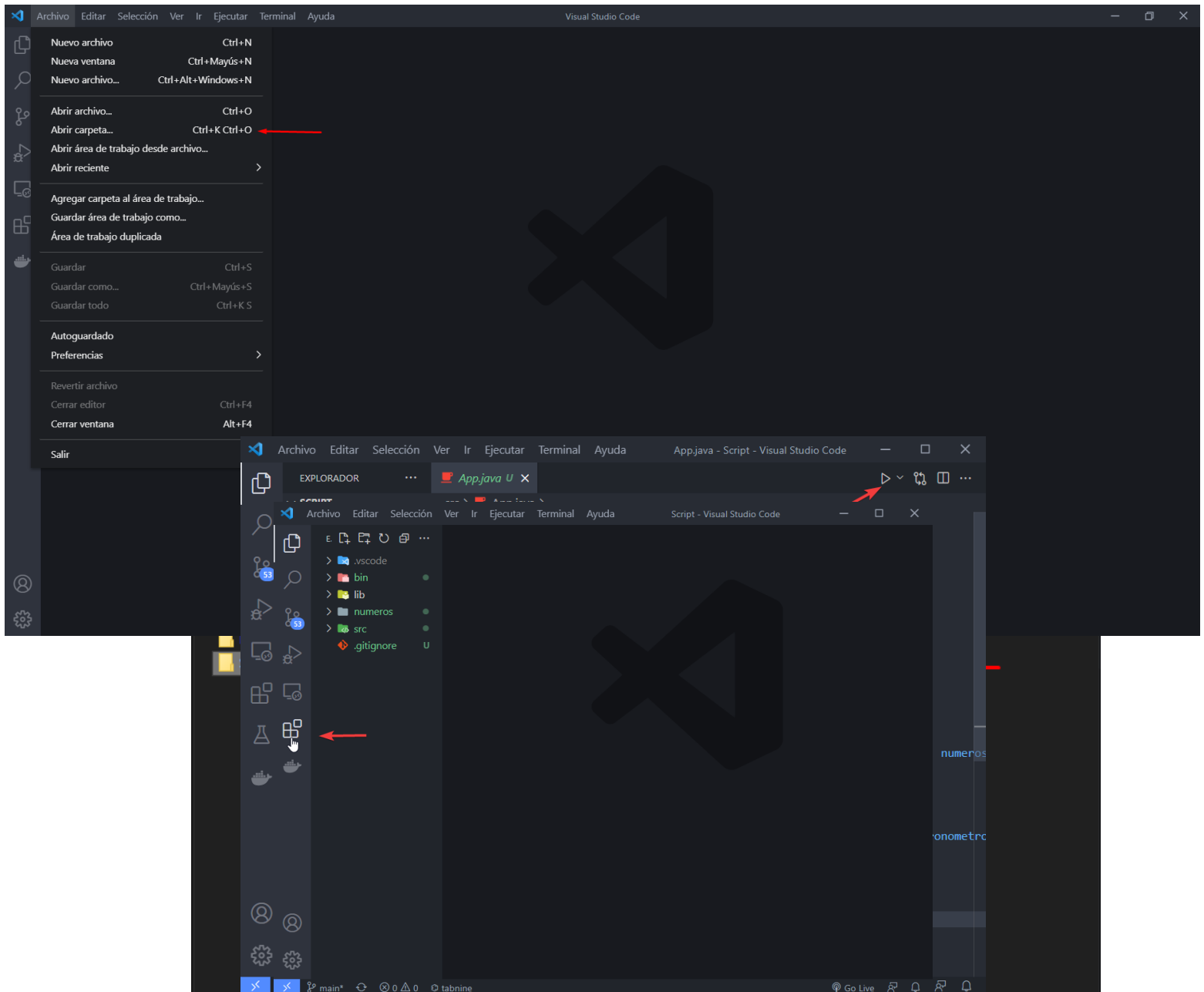


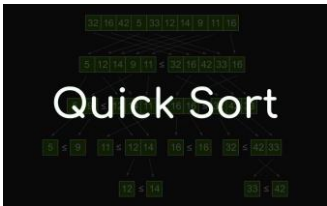
	Ficha	Proyecto	
	2236347	QuickSort	
	Fecha	Versión	Código
	27/10/2021	1.2	IP-01

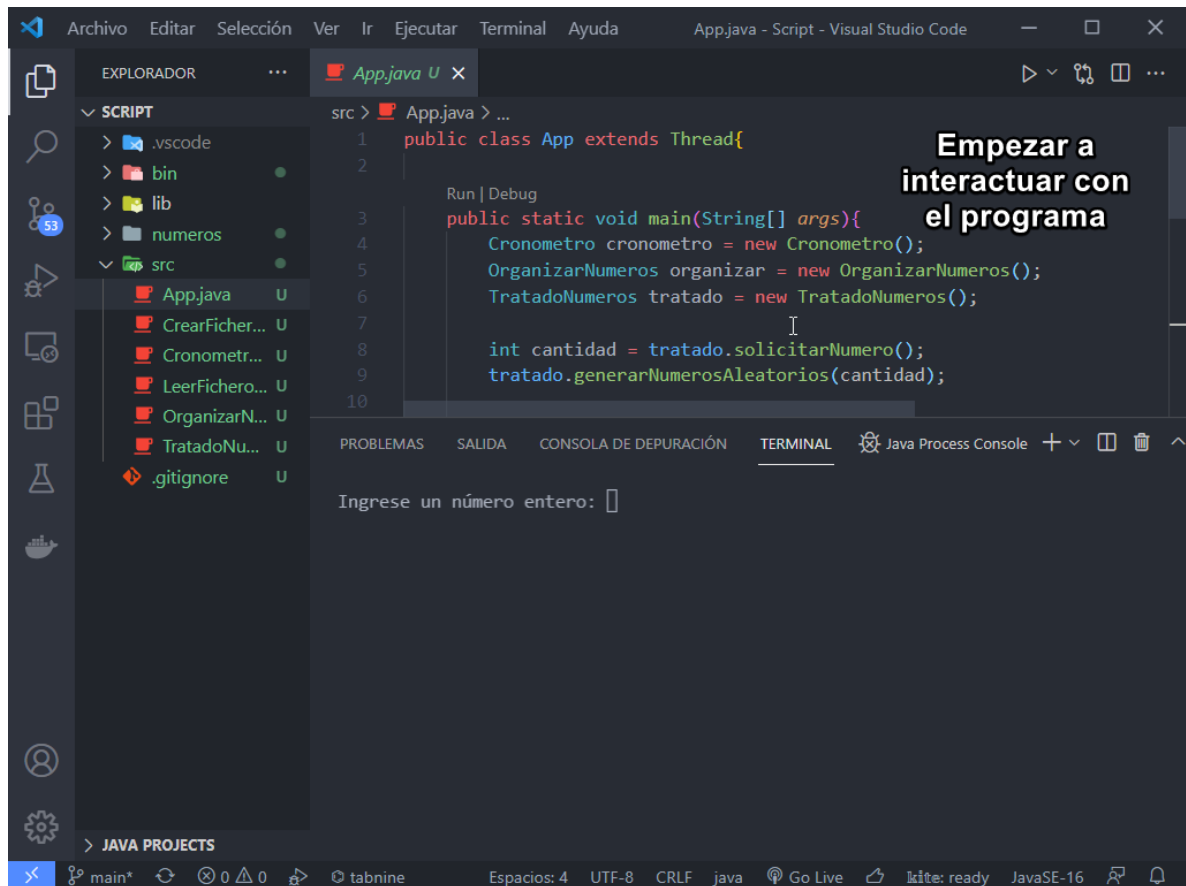


	<i>Ficha</i>	<i>Proyecto</i>	
	2236347	QuickSort	
	<i>Fecha</i>	<i>Versión</i>	<i>Código</i>
	27/10/2021	1.2	IP-01

	Ficha		Proyecto	
	2236347		QuickSort	
	Fecha	Versión	Código	
	27/10/2021	1.2	IP-01	



	Ficha		Proyecto	
	2236347		QuickSort	
	Fecha		Versión	Código
	27/10/2021		1.2	IP-01



Empezar a interactuar con el programa

```

1 public class App extends Thread{
2
3
4     Run | Debug
5     public static void main(String[] args){
6         Cronometro cronometro = new Cronometro();
7         OrganizarNumeros organizar = new OrganizarNumeros();
8         TratadoNumeros tratado = new TratadoNumeros();
9
10        int cantidad = tratado.solicitarNumero();
11        tratado.generarNumerosAleatorios(cantidad);


```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Java Process Console

Ingrese un número entero:

4.5. El programa en total cuenta con 214 líneas de código, distribuidas de la siguiente manera:


- Clase App: 16 líneas.
- Clase CrearFichero: 39 líneas.
- Clase Cronometro: 32 líneas.
- Clase LeerFichero: 54 líneas
- Clase OrganizarNumeros: 49 líneas.
- Clase TratadoNumeros: 34 líneas.

	Ficha	Proyecto	
	2236347	QuickSort	
	Fecha	Versión	Código
	27/10/2021	1.2	IP-01

- 4.6. En la anteriormente mencionada carpeta 'numeros', se encuentran los archivos de texto con los números generados aleatoriamente y los números generados ya ordenados, el almacenamiento de éste último se maneja al final del proceso, donde se le pregunta al usuario si desea conservar o no éste archivo.

5. Plan de pruebas

PLAN DE PRUEBAS			
Nr o.	Cantidad de números	Resultado 1	Resultado 2
1	100	Tiempo demorado(hh:mm:ss.ssss): 00:00:00.0013	Tiempo demorado(hh:mm:ss.ssss): 00:00:00.0013
2	3000	Tiempo demorado(hh:mm:ss.ssss): 00:00:00.0956	Tiempo demorado(hh:mm:ss.ssss): 00:00:01.0011
3	50000	Tiempo demorado(hh:mm:ss.ssss): 00:00:04.0653	Tiempo demorado(hh:mm:ss.ssss): 00:00:03.0570
4	15000	Tiempo demorado(hh:mm:ss.ssss): 00:00:01.0118	Tiempo demorado(hh:mm:ss.ssss): 00:00:01.0192
5	100000	Tiempo demorado(hh:mm:ss.ssss): 00:00:03.0206	Tiempo demorado(hh:mm:ss.ssss): 00:00:03.0972
6	400000	Tiempo demorado(hh:mm:ss.ssss): 00:00:04.0440	Tiempo demorado(hh:mm:ss.ssss): 00:00:06.0304
7	800	Tiempo demorado(hh:mm:ss.ssss): 00:00:00.0924	Tiempo demorado(hh:mm:ss.ssss): 00:00:02.0637
8	1000000	Tiempo demorado(hh:mm:ss.ssss): 00:00:05.0208	Tiempo demorado(hh:mm:ss.ssss): 00:00:06.0130
9	10000000	Tiempo demorado(hh:mm:ss.ssss): 00:00:45.0178	Tiempo demorado(hh:mm:ss.ssss): 00:00:44.0134
10	100000000	Tiempo demorado(hh:mm:ss.ssss): 00:06:47.0385	Tiempo demorado(hh:mm:ss.ssss): 00:06:59.0820

	Ficha		Proyecto	
	2236347		QuickSort	
	Fecha		Versión	Código
	27/10/2021		1.2	IP-01

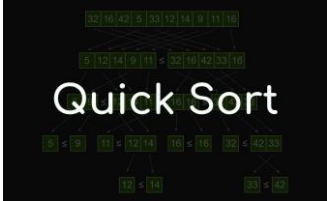
6. Diagrama de Gantt

COMPETENCIA																											
Construir un sistema de información que cumpla con los requisitos del cliente.																											
RESULTADO																											
Realizar la codificación del software planteado en Java, enfocado al cumplimiento de los requisitos.																											
GENERALIDADES																											
Trimestre IV: Entregable de proyecto para Hugo Hernán Henao Hernández.																											
OBSERVACIONES																											
Definir actividades pendientes según resultado y estrategias.																											
Temáticas	Octubre																										
	Semana 1													Semana 2													
	17	18	19	20	21	22	23	24	25	26	27	28		17	18	19	20	21	22	23	24	25	26	27	28		
Desarrollar diagrama de Gantt																											
Desarrollar documento de planeación de pruebas PSP																											
Generación de informe.																											
Analizar requisitos																											
Realizar documento con estimación de costos.																											
Crear un diagrama de flujo.																											
Alistar entorno de desarrollo.																											
Codificar																											
Ejecutar pruebas planeadas.																											
Hallar líneas de código totales realizadas																											
Revisión y pruebas adicionales.																											
Entrega del proyecto.																											

Este documento se puede encontrar en la carpeta del proyecto en el subdirectorío Documentos/Diagrama de Gantt.xlsx

7. Estimación de costos

La estimación de costos para este proyecto se puede visualizar en la carpeta Documentos/Presupuesto.xlsx. En este se da un valor detallado a cada proceso de fabricación del software, dando un presupuesto inicial de \$565.760 COP.

	Ficha		Proyecto	
	2236347		QuickSort	
	Fecha		Versión	Código
	27/10/2021		1.2	IP-01

8. Trabajo colaborativo

El trabajo se realizó entre Alejandro Giraldo Herrera Y Jhon Faber Muñoz Restrepo de forma simultánea, manteniendo una comunicación constante en canales como WhatsApp y Google Meet.