

Comparación de algoritmos de ordenación

COUNTINGSORT - SHAKERSORT

Russbell Juan Pablo Arratia Paz

Alexander Efrain Contreras Rodriguez

Danitza Carmen Capía Quiñonez





Objetivos

IMPLEMENTAR

REGISTRAR

ANALIZAR

VISUALIZAR

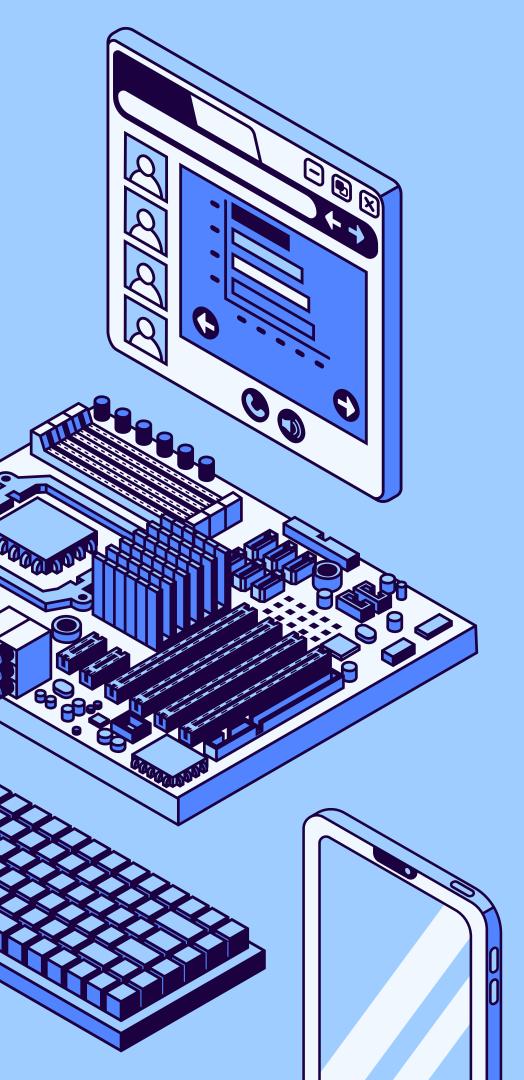
Metodologia

Comparación entre algoritmos de ordenacmiento en espacios controlados.

50 Realizaciones por tamaño.

Evaluación según complejidad.

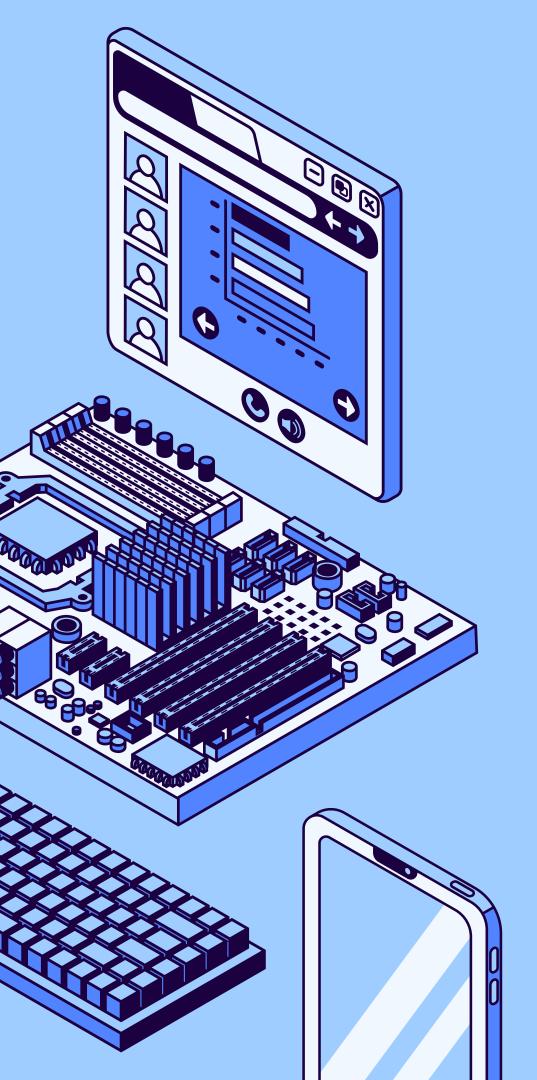




Su modo de ordenación no es por comparación, sino que cuenta cuántas veces aparece un valor en el arreglo, infomación que usa para para colocarlos en orden

Mejor, promedio y peor caso: O(n + k)

Complejidad de espacio: O(k)





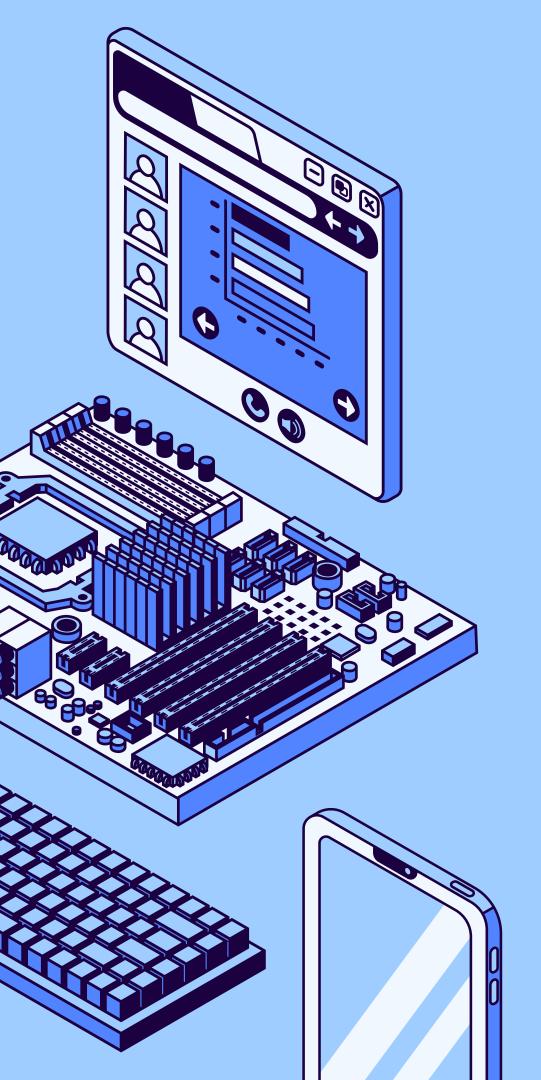
 0
 2
 2
 0
 1
 0
 1
 0
 0

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

 0
 2
 4
 5
 6
 7
 7

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

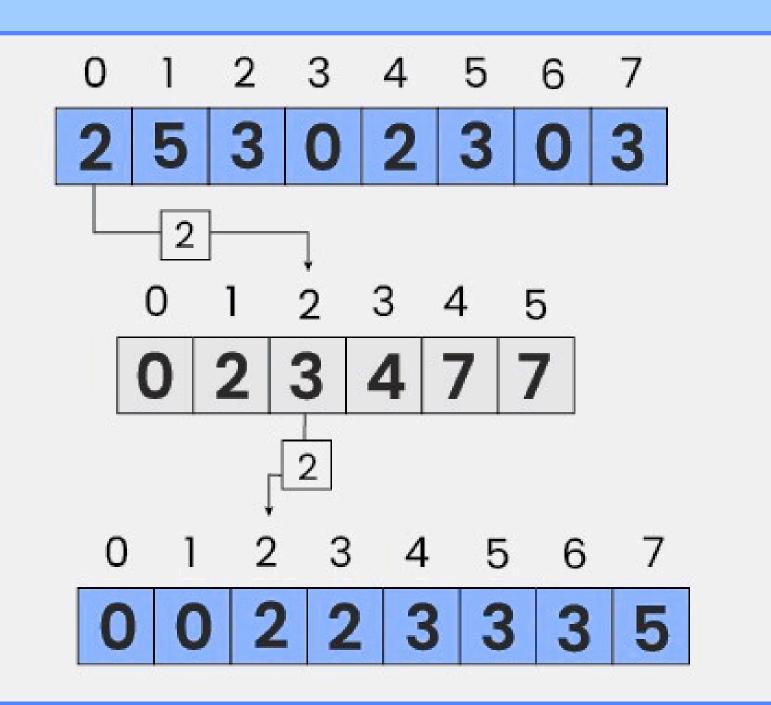


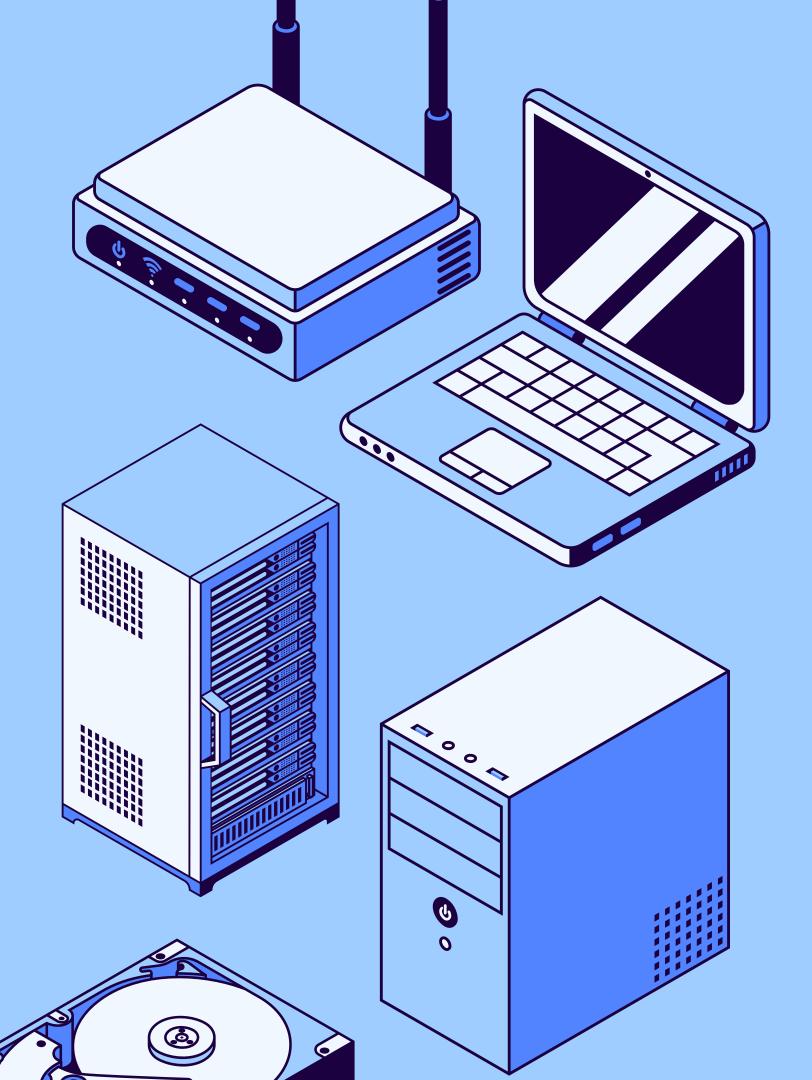


inputArray

countArray

outputArray





Intercambio Directo Bidireccional

Variante del ordenamiento por selección que busca el mínimo y el máximo, posicionarlos a los extremos por pasada, termina en el medio.

Permite organizar los elementos de un arreglo o lista comparándolos y realizando intercambios.

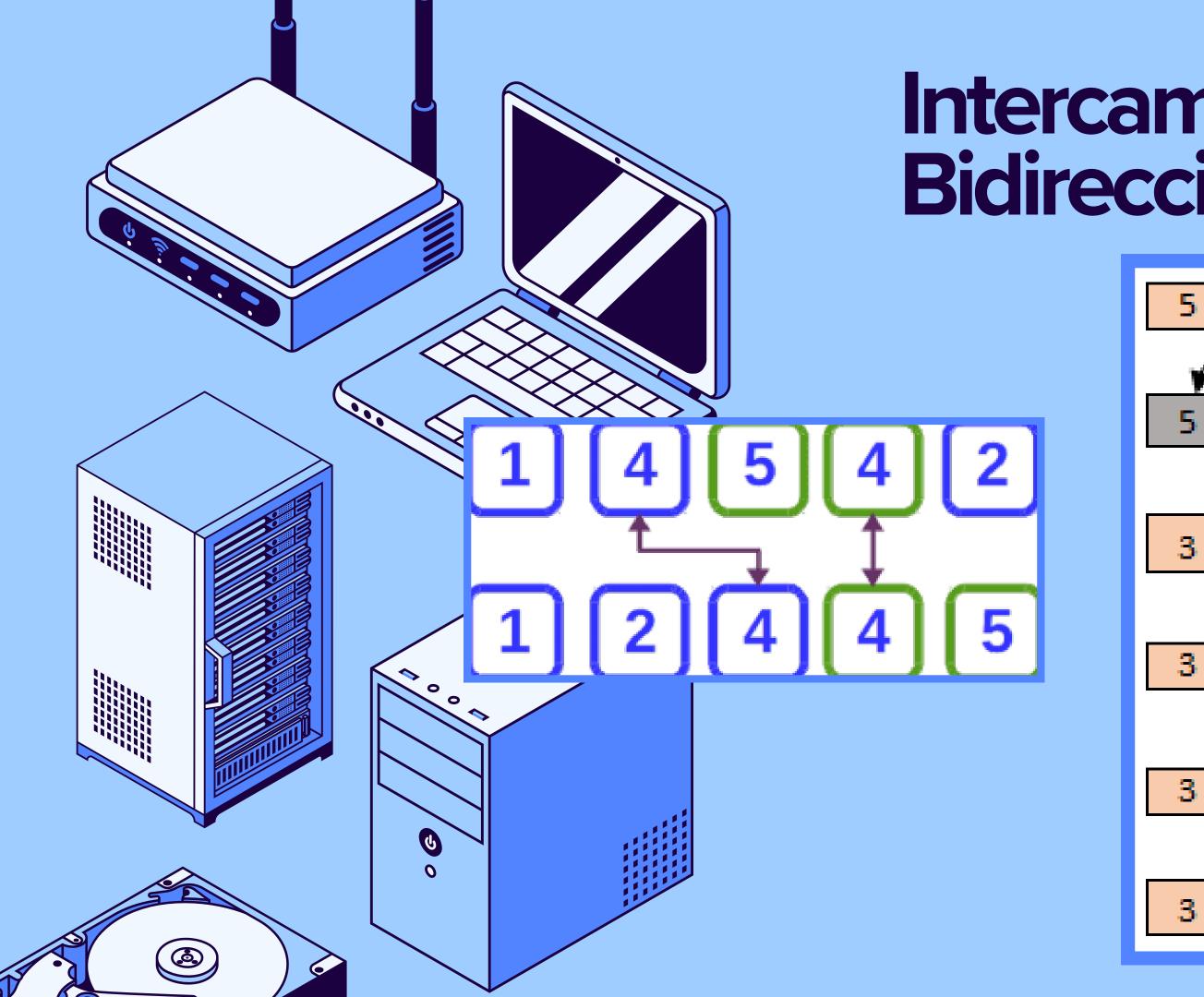
También conocido como ordenamiento burbuja bidireccional o Shaker Sort.

Mejor caso: O(n)

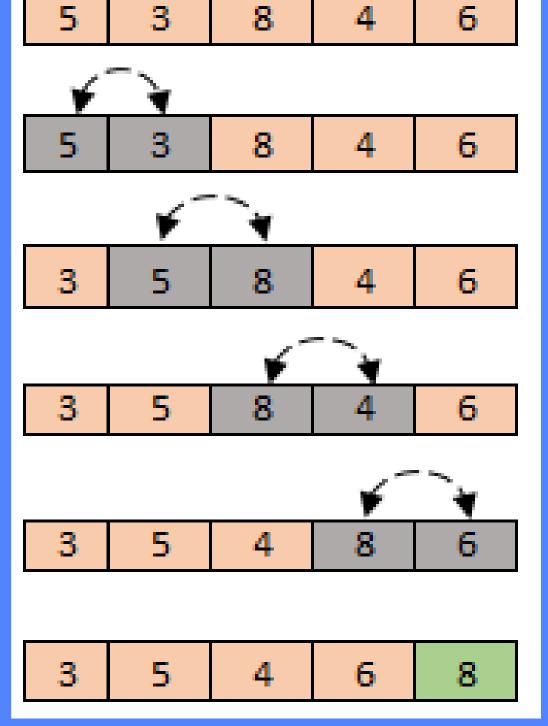
Peor caso: O(n²)

Complejidad promedio: O(n²)

Complejidad de espacio: O(1)



Intercambio Directo Bidireccional



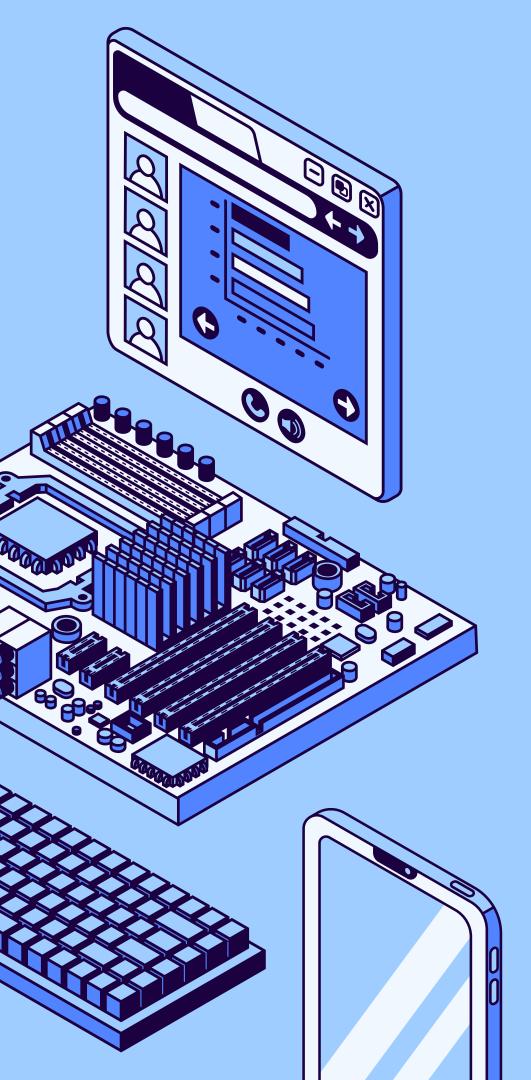
Resultados

Luego de la ejecusión de las pruebas con ambos algoritmos, se obtuvieron los resultados a continuación, seguidos de un análisis comparativo.



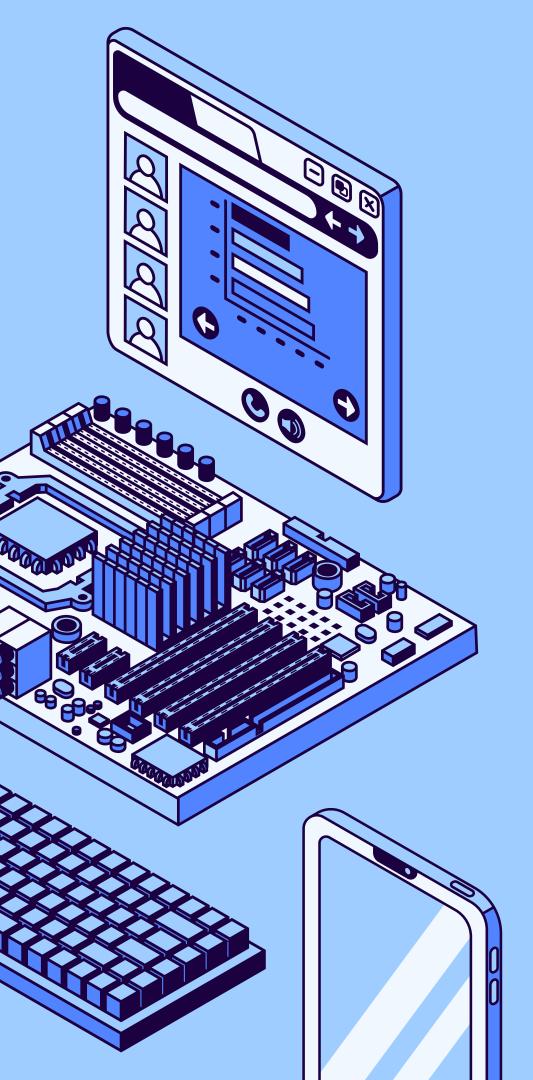


Característica de datos		Ti	iemj	ро е	n m	icro	seg	und	os		Tiempo
n=1000, rango=1000	1 2 3 4 5 6 7 8 9 10						10	Promedio			
Arreglo aleatorio	0	0	0	0	0	0	0	0	0	0	0
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo casi ordenado	0	0	0	0	0	0	0	0	0	0	0
Arreglo con duplicados	0	0	0	0	0	0	0	0	0	0	0
											0



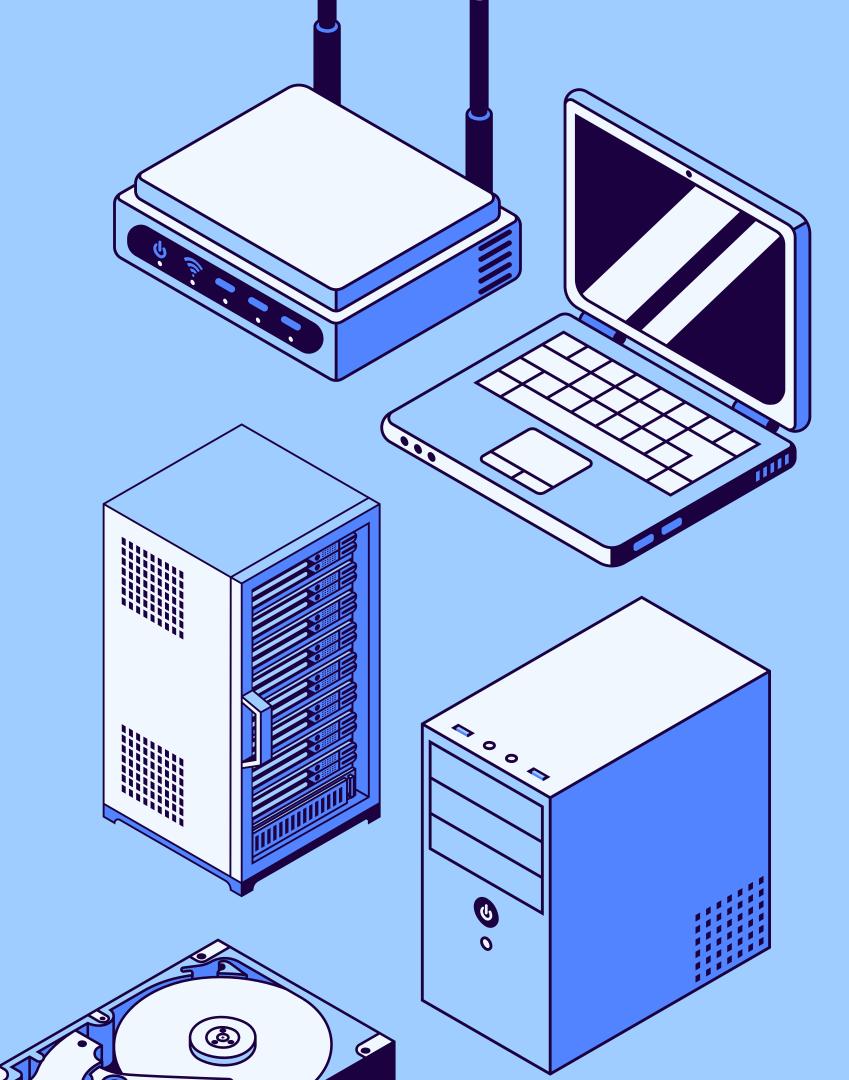
Resultados de eficiencia del algoritmo Counting Sort cuando n=10000 y el rango=10000

Característica de datos			Т	iempo	en mic	rosegur	ıdos				Tiempo
n=10000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	0	0	0	0	0	0	0	0.996	0	0	0.0996
Arreglo ascendente	0	0	0	0	0	0.997	0	0	0	0	0.0997
Arreglo descendente	0	0	0	0	0.997	0	0	0	0	0	0.0997
Arreglo casi ordenado	0	0.996	0	0	0	0	0	0	0	0	0.0996
Arreglo con duplicados	0	0	0	0	0.996	0	0	0	0.996	0	0.1992
											0.11956



Resultados de eficiencia del algoritmo Counting Sort cuando n=100000 y el rango=100000

Característica de datos				Tiempo	en micr	osegun	dos				Tiempo
n=100000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	0.998	0.998	0.997	0	0	0.996	0.998	0.997	0.997	0.998	0.7979
Arreglo ascendente	0.997	0.997	0.997	0.997	0.997	0	0.997	0.997	0.997	0.997	0.8973
Arreglo descendente	0.997	0.996	0.996	0.997	0.998	0.996	0.998	0	0.997	0.997	0.8972
Arreglo casi ordenado	0	0.997	0.997	0.996	0	0.997	0	0.997	0.998	0	0.5982
Arreglo con duplicados	0	0.997	0	0	0	0	0.996	0.998	0.996	0.996	0.4983
											0.73778



Intercambio Directo Bidireccional

Análisis

Tabla 5 Resultados de eficiencia en comparaciones del algoritmo Shaking Sort cuando n=1000 y el rango=1000

Característica de datos					Compa	raciones					Comparaci
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	ones Promedio
Arreglo aleatorio	334237	333139	344761	336587	326490	338249	338183	333004	329866	336394	335091
Arreglo ascendente	999	999	999	999	999	999	999	999	999	999	999
Arreglo descendente	499500	499500	499500	499500	499500	499500	499500	499500	499500	499500	499500
Arreglo casi ordenado	32422	32120	39323	33018	35296	33967	32995	31589	33385	35246	33936.1
Arreglo con duplicados	324961	329229	337692	335710	333954	335103	343241	329841	328885	329581	332819.7
											240469.16

Tabla 6Resultados de eficiencia en intercambios del algoritmo Shaking Sort cuando n=1000 y el rango=1000

Característica de datos				Intercambios							
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	248227	249526	257723	254145	245695	253613	250161	250598	249430	250270	250938.8
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	499500	499500	499500	499500	499500	499500	499500	499500	499500	499500	499500
Arreglo casi ordenado	28894	29258	35984	29894	32524	31248	29716	28865	29540	32316	30823.9
Arreglo con duplicados	240739	242823	250053	251216	244556	248399	256239	245466	240072	245488	246505.1
											205553.56

Característica de datos			T	iempo e	n micro	seguno	dos				Tiempo	
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	Promedio	
Arreglo aleatorio	0.997	1.995	0.998	0.997	0.997	0.996	0.997	0.997	0.997	0.998	1.0969	
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0	
Arreglo descendente	1.994	1.995	1.994	1.995	2.002	1.994	1.994	1.994	1.996	1.996	1.9954	
Arreglo casi ordenado	0	0.998	0.998	0	0	0	0	0.997	0	0	0.2993	
	0.998	0.998	0.997	0.998	0.998	0.997	0.998	0.997	0.997	0.997	0.9975	
											0.87782	

Análisis

Tabla 8Resultados de eficiencia en comparaciones del algoritmo Shaking Sort cuando n=10000 y el rango=10000

Característica de datos					Compar	aciones					Comparaciones
n=10000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Amagla glastonia	332278	336947	334818	334642	331316	335171	333680	332039	33215	33442	33374660
Arreglo aleatorio	15	40	49	46	63	85	18	78	081	025	333/4000
Arreglo ascendente	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
Arreglo descendente	499950	499950	499950	499950	499950	499950	499950	499950	49995	49995	49995000
Arregio descendente	00	00	00	00	00	00	00	00	000	000	49993000
Arreglo casi ordenado	334218	331875	325818	338143	331952	315991	328438	345122	32417	32819	3303937.8
Arregio casi ordenado	4	9	1	7	4	7	2	6	70	98	3303737.0
Arreglo con duplicados	330089	331613	334472	329757	330066	331908	334646	326587	33365	33159	33143940.3
Arregio con auplicados	55	97	44	20	01	91	85	56	663	491	33143740.3
											23965507.42

Tabla 9Resultados de eficiencia en intercambios del algoritmo Shaking Sort cuando n=10000 y el rango=10000

Característica de datos				i	Intercan	nbios				·	Intercambios
n=10000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	2495545	25307	25139	25209	24876	25061	24937	24900	25004	25134	25052765.5
mregio diediorio	2	285	135	058	890	789	929	256	865	996	20002700.0
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	4999500	49995	49995	49995	49995	49995	49995	49995	49995	49995	49995000
Arregio descendente	0	000	000	000	000	000	000	000	000	000	47773000
Annagla aggi andangda		32221	31493	32740	32065	30490	31833	33377	31304	31711	3195761.4
Arreglo casi ordenado	3233710	92	30	86	64	70	20	64	50	28	3193/01.4
Arreglo con duplicados	2437020	24565	24768	24408	24406	24470	24730	24163	24720	24527	24513075.4
Arregio con auplicados	3	471	930	090	380	061	948	023	058	590	243130/3.4
											20551320.46

Tabla 7Resultados de eficiencia en tiempo del algoritmo Shaking Sort cuando n=10000 y el rango=10000

Característica de datos				Tiemp	o en mio	crosegun	idos				Tiempo
n=10000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	121.674	125.501	122.367	124.26	121.29 7	122.21 1	121.24 6	120.71 1	121.67 6	121.6 74	122,2617
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	193.503	194.485	193.5	193.17 8	193.57 1	192.37	194.02	193.53 7	192.48 6	192.6	193.325
Arreglo casi ordenado	12.964	12.965	13.478	12.967	12.966	12.965	12.967	13.966	11.967	12.96 6	13.0171
Arreglo con duplicados	118.715	119.705	119.706	118.27 3	119.33 5	118.66 6	120.70 8	116.70 6	118.71 4	117.6 89	118.8217
											89.4851

Análisis

Tabla 11
Resultados de eficiencia en comparaciones del algoritmo Shaking Sort cuando n=100000 y el rango=100000

Característica de datos				(Compar	aciones					Comparaciones
n=100000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	33203	33356	33289	33414	33252	33342	33302	33416	33385	33339	3333038226
Affegio aleatorio	92667	84504	23079	77241	98834	59086	18507	52737	10630	64971	3333036220
Arreglo ascendente	99999	99999	99999	99999	99999	99999	99999	99999	99999	99999	99999
Arreglo descendente	49999	49999	49999	49999	49999	49999	49999	49999	49999	49999	4999950000
Arregio descendente	50000	50000	50000	50000	50000	50000	50000	50000	50000	50000	4999930000
Arreglo casi ordenado	94806	94219	93633	93798	95074	93666	92477	94008	93984	95551	94122286.2
Arregio casi ordenado	409	496	657	885	710	773	740	696	769	727	94122200.2
Arreglo con duplicados	33042	33178	33131	33123	33139	33059	33231	33167	32980	33089	3311439213
Arregio con duplicados	93398	15112	40224	55437	22906	46646	78018	92441	01670	46280	3311439213
											2347729945

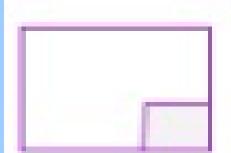
Tabla 12 Resultados de eficiencia en intercambios del algoritmo Shaking Sort cuando n=100000 y el rango=100000

Característica de datos					Interca	mbios					Intercambios
n=100000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	24906 09760	24977 93616	24968 18361	25050 56319	24905 15827	24983 41152		25061 21431	25049 88956		2498333775
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	49999 50000	49999 50000	49999 50000	49999 50000		49999 50000		49999 50000	49999 50000	49999 50000	4999950000
Arreglo casi ordenado	88315 567	87881 261	87196 260	87384 084	88612 166	87268 520	86142 056	87500 946	87517 226	89032 390	87685047.6
Arreglo con duplicados	24481 23970	24553 22650	24519 55697	24511 60399	24525 97105	24470 75137		24534 34377			2450731985
											2007340162

Tabla 10Resultados de eficiencia en tiempo del algoritmo Shaking Sort cuando n=100000 y el rango=100000

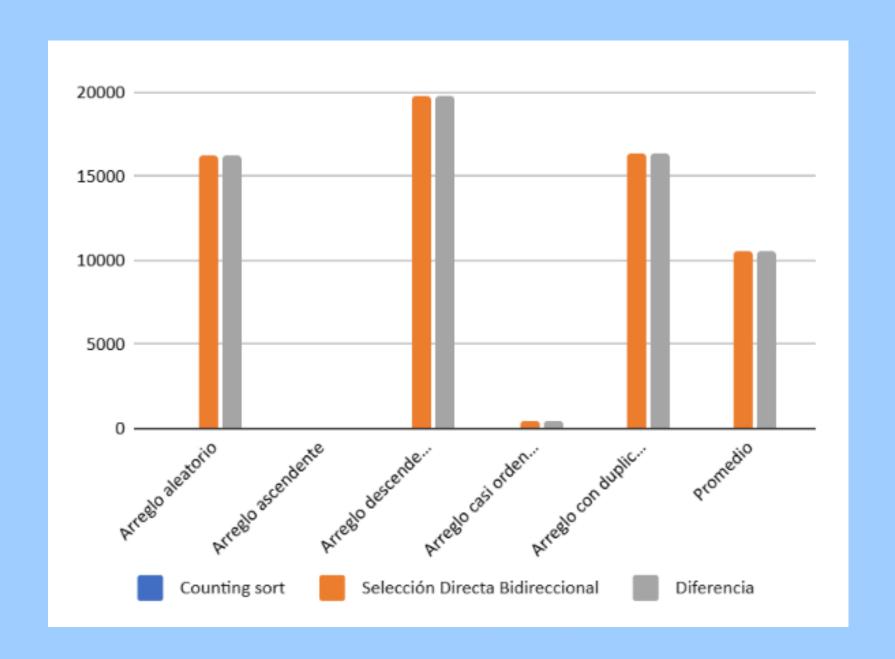
Característica de datos				Tiemp	o en mi	crosegu	ndos				Tiempo
n=100000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	16134.	16289.	16179.	16217.	16100.	16092.	16272.	16379.	16325.	16365.	16235.72
Arregio aleatorio	9	2	8	3	6	5	3	9	3	4	10235.72
Arreglo ascendente	0	0	0	0	0	0	0.997	0	0	0	0.0997
Arreglo descendente	19457.	19767.	19849.	19889.	19805.	19761.		19693.	19630.	19799.	19731.83
Arregio descendente	4	3	7	3	4	2	19665	1	2	7	19/31.63
Arreglo casi ordenado	403.38	410.17	381.51	381.85	399.58	386.04	376.03	384.63		392.77	390,4928
Arregio casi ordenado	6	1	6	9	5	5	8	9	388.91	9	390.4926
Arreale con dunlicados	16403.	16563.	16476.	16365.	16572.	16377.	16288.	16278.	16449.		16396.94
Arreglo con duplicados	5	6	9	8	8	2	2	7	7	16193	10390.94
											10551.0165

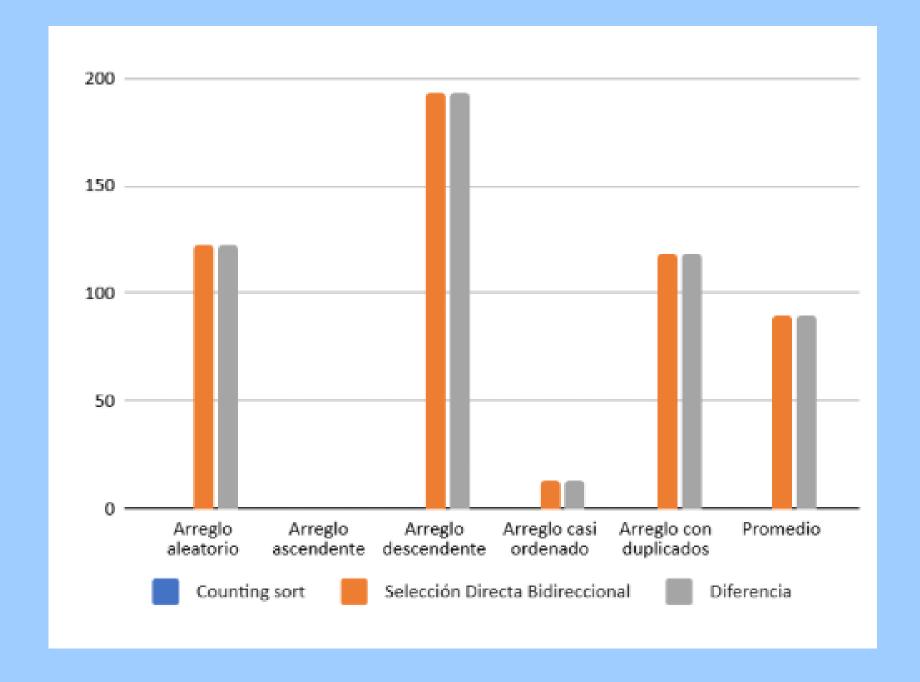
n=10000000, rango=200 n	1	2	3	4	5	Promedio
Arreglo k > n	6.94	5.759	5.798	5.909	5.798	6.040658



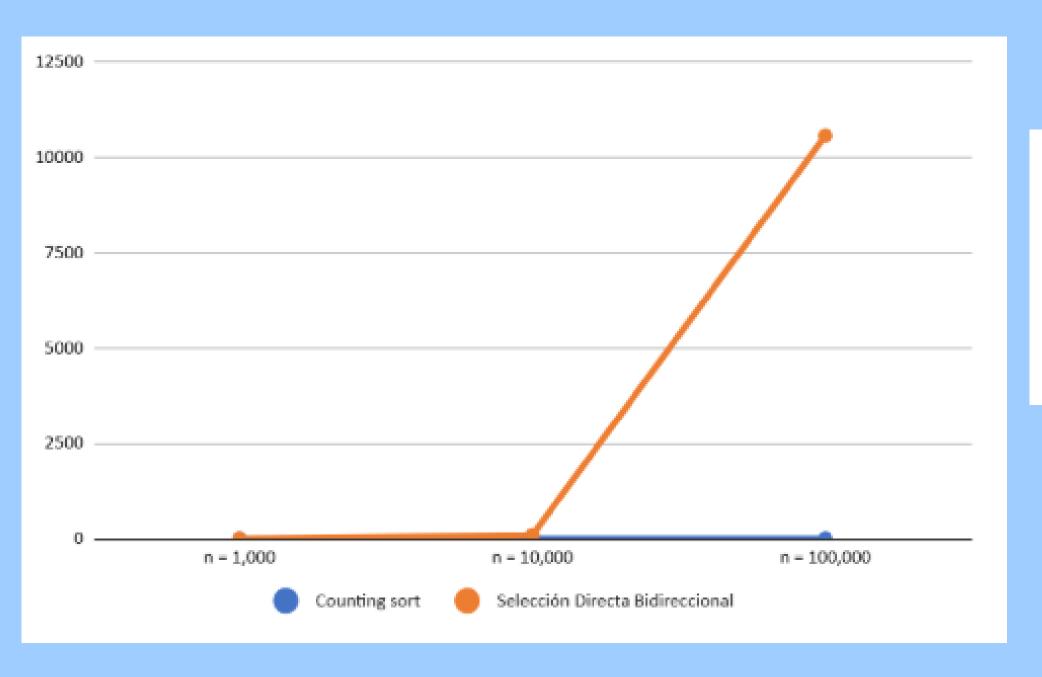
Memoria 6.3/15.9 GB (40%)







Análisis



Tamaño de n = rango	Counting sort	Selección Directa Bidireccional	
n = 1,000	0	0.87782	
n = 10,000	0.11956	89.4851	
n = 100,000	0.73778	10551.0165	



Conclusiones

En síntesis, el algoritmo
Counting Sort al ser
temporalmente lineal, tiene
mayor eficacia respecto al
algoritmo ShakerSort, ya
que este último es de costo
computacional cuadrática.

En conclusión, el costo computacional de Counting Sort es considerablemente más bajo que el de la Selección Directa Bidireccional. Counting Sort mantiene un comportamiento estable y eficiente incluso con grandes volúmenes de datos, mientras que la Selección Directa Bidireccional presenta un aumento drástico del tiempo de ejecución conforme crece n. Por lo tanto, desde el punto de vista del costo computacional, Counting Sort es mucho más eficiente y escalable, aunque requiere un poco más de memoria auxiliar.

Gracias

