

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



Metodología

1. Comparación entre algoritmos de ordenamiento en espacios controlados.
2. 50 Realizaciones por tamaño.
3. Evaluación según complejidad.

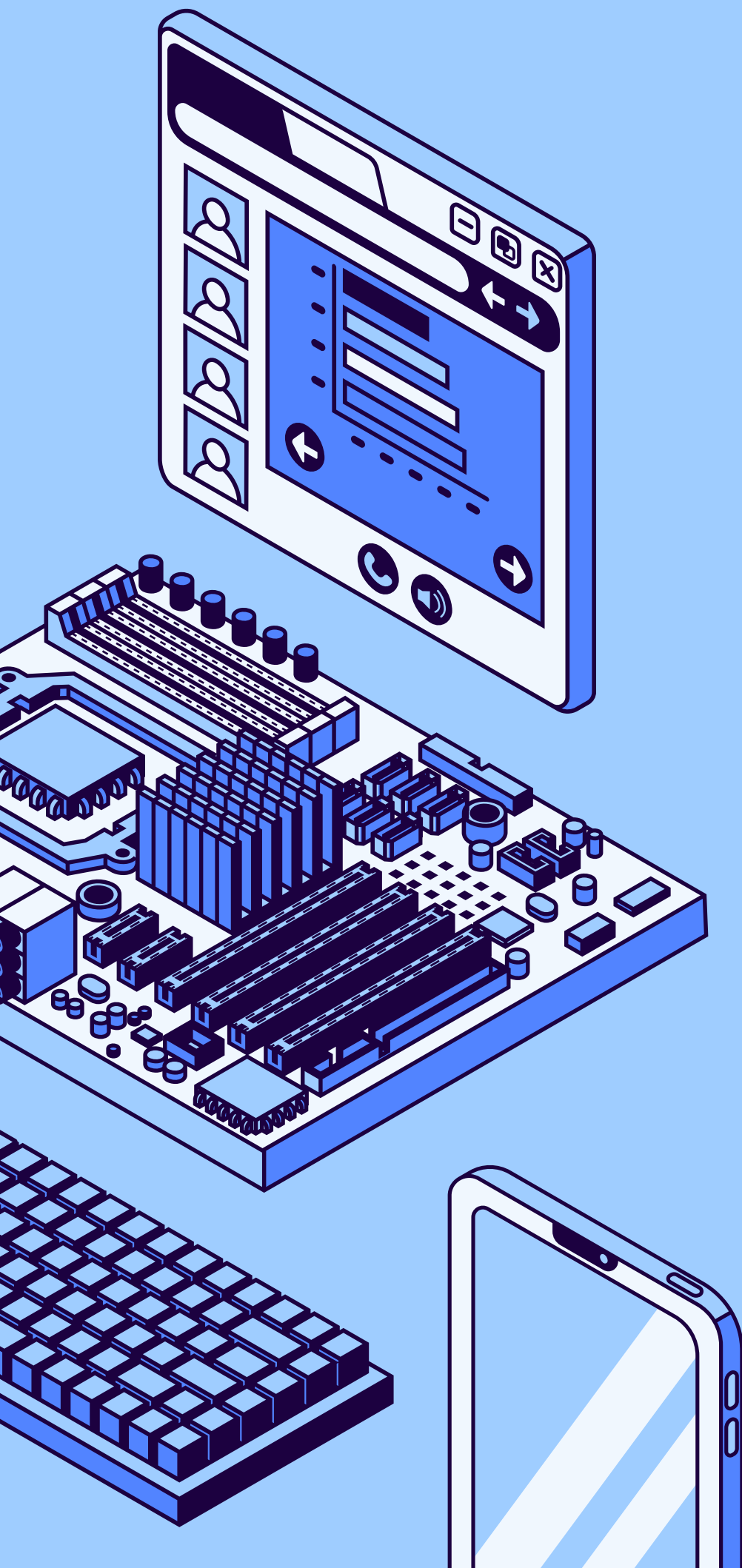


Counting Sort

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

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

Complejidad de espacio: $O(k)$



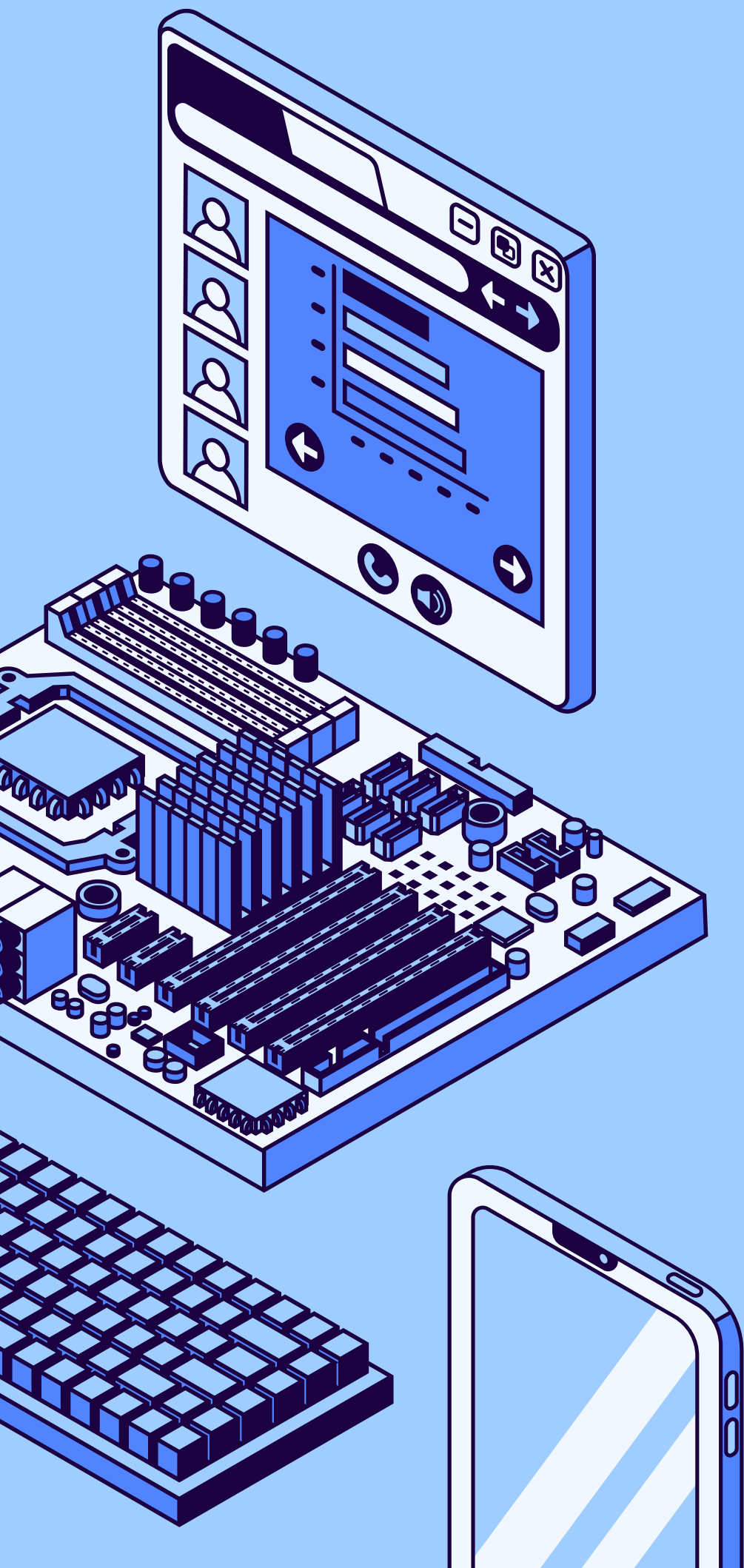
Intercambio Directo Bidireccional

Algoritmo de ordenamiento interno que permite organizar los elementos de un arreglo o lista comparándolos y realizando intercambios.

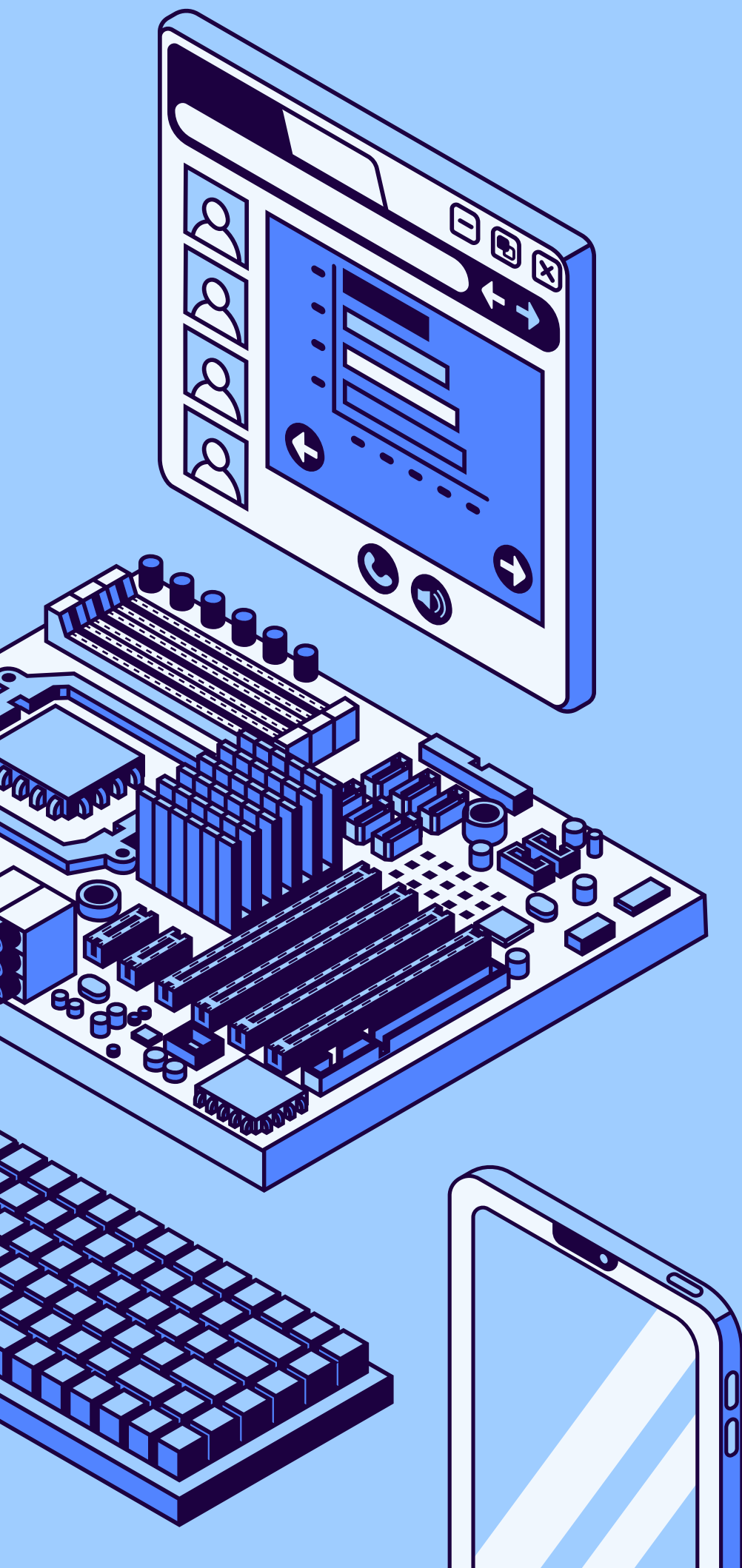
También conocido como ordenamiento burbuja bidireccional o Shaker Sort.

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

Complejidad de espacio: $O(k)$



Counting Sort



28	1	24	6	23	11	13	12	5	13	29	7	14	9	10	15	10	1	3	6	8	15	3	10	21	7	14	15	6	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

0	2	2	4	4	5	8	10	11	12	15	16	17	19	21	4	0	0	0	0	0	1	0	1	1	0	1	0	0	1	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

0	0	2	2	4	4	5	8	10	11	12	15	16	17	19	21	25	25	25	25	25	25	26	26	27	28	28	29	29	29	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

1	1	3	3	5	6	6																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

0	2	2	4	4	5	7	8	10	11	12	15	16	17	19	21	25	25	25	25	25	25	26	26	27	28	28	29	29	29	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

							6	7	7	8	9	10	10	10	11	12	13	13	14	14	15	15	15	15	21	23	24	26	29
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29



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

Mejor caso: $O(n)$

Peor caso: $O(n^2)$

Complejidad promedio: $O(n^2)$

Complejidad de espacio: $O(1)$

Intercambio Directo Bidireccional

Resultados

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



A stylized illustration of digital technology components. At the top is a computer monitor displaying a web interface with a sidebar of four user avatars, a central content area with a list, and a bottom navigation bar with icons for home, search, and a right-pointing arrow. Below the monitor is a detailed circuit board with various components like RAM modules, a CPU, and capacitors. In the bottom left is a keyboard with blue keys. In the bottom right is a smartphone with a white frame and a blue screen. The entire scene is set against a light blue background with a large, faint white 'C' in the top right corner.

A stylized illustration of digital technology components. At the top is a computer monitor displaying a web interface with a sidebar of four user avatars, a central content area with a list of items, and a bottom navigation bar with icons for home, search, and a right arrow. Below the monitor is a detailed circuit board with various components like RAM modules, a CPU, and capacitors. In the bottom left is a keyboard with blue keys. In the bottom right is a smartphone with a white frame and a blue screen. The entire scene is set against a light blue background with a large, faint white 'C' in the top right corner.

A stylized illustration of digital technology components. At the top is a computer monitor displaying a web interface with a sidebar of four user avatars, a central content area with a list of items, and a bottom navigation bar with icons for home, search, and a right arrow. Below the monitor is a detailed circuit board with various components like RAM modules, a CPU, and capacitors. In the bottom left is a portion of a keyboard with blue keys. In the bottom right is a smartphone with a white frame and a blue screen. The entire illustration is set against a light blue background and uses a consistent blue and white color palette with black outlines.

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

[illegible]

A stylized illustration of digital technology components. At the top is a computer monitor displaying a web interface with a sidebar of four user avatars, a main content area with a list of items, and a bottom navigation bar with icons for home, search, and a right-pointing arrow. Below the monitor is a detailed circuit board with various components like RAM sticks, a CPU, and capacitors. In the bottom left is a portion of a keyboard with blue keys. In the bottom right is a smartphone with a white frame and a blue screen. The entire illustration is set against a light blue background and uses a consistent blue and white color palette with black outlines.

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

[illegible]



Característica de datos	Tiempo en microsegundos										Tiempo Promedio
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	
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
Arreglo con duplicados	0.998	0.998	0.997	0.998	0.998	0.997	0.998	0.997	0.997	0.997	0.9975
											0.87782

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	Comparaciones										Comparaciones Promedio
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	
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 6

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

Característica de datos	Intercambios										Intercambios Promedio
n=1000, rango=1000	1	2	3	4	5	6	7	8	9	10	
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

n=1000, rango=1000	Counting sort	Selección Directa Bidireccional	Diferencia
Arreglo aleatorio	0	1.0969	1.0969
Arreglo ascendente	0	0	0
Arreglo descendente	0	1.9954	1.9954
Arreglo casi ordenado	0	0.2993	0.2993
Arreglo con duplicados	0	0.9975	0.9975
Promedio	0	0.87782	0.87782

Análisis

Tabla 7

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

Característica de datos	Tiempo en microsegundos										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	121.297	122.211	121.246	120.711	121.676	121.674	122.2617
Arreglo ascendente	0	0	0	0	0	0	0	0	0	0	0
Arreglo descendente	193.503	194.485	193.5	193.178	193.571	192.37	194.02	193.537	192.486	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	12.966	13.0171
Arreglo con duplicados	118.715	119.705	119.706	118.273	119.335	118.666	120.708	116.706	118.714	117.689	118.8217
											89.4851

n=10000, rango=10000	Counting sort	Selección Directa Bidireccional	Diferencia
Arreglo aleatorio	0.0996	122.2617	122.1621
Arreglo ascendente	0.0997	0	0.0997
Arreglo descendente	0.0997	193.325	193.2253
Arreglo casi ordenado	0.0996	13.0171	12.9175
Arreglo con duplicados	0.1992	118.8217	118.6225
Promedio	0.11956	89.4851	89.36554

Análisis

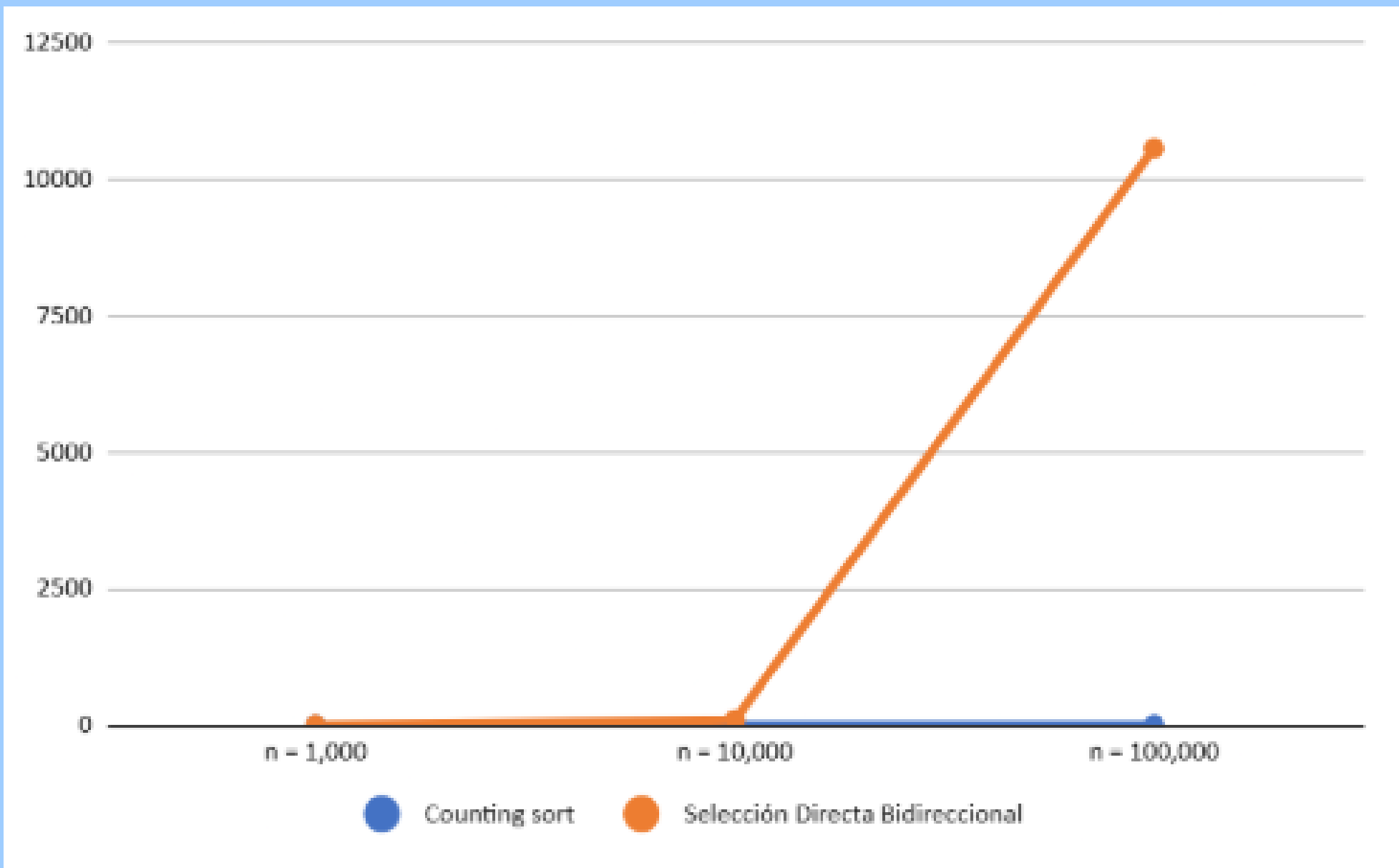
Tabla 10

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

Característica de datos	Tiempo en microsegundos										Tiempo
n=100000, rango=10000	1	2	3	4	5	6	7	8	9	10	Promedio
Arreglo aleatorio	16134.9	16289.2	16179.8	16217.3	16100.6	16092.5	16272.3	16379.9	16325.3	16365.4	16235.72
Arreglo ascendente	0	0	0	0	0	0	0.997	0	0	0	0.0997
Arreglo descendente	19457.4	19767.3	19849.7	19889.3	19805.4	19761.2	19665	19693.1	19630.2	19799.7	19731.83
Arreglo casi ordenado	403.386	410.171	381.516	381.859	399.585	386.045	376.038	384.639	388.91	392.779	390.4928
Arreglo con duplicados	16403.5	16563.6	16476.9	16365.8	16572.8	16377.2	16288.2	16278.7	16449.7	16193	16396.94
											10551.0165

n=100000, rango=100000	Counting sort	Selección Directa Bidireccional	Diferencia
Arreglo aleatorio	0.7979	16235.72	16234.9221
Arreglo ascendente	0.8973	0.0997	0.7976
Arreglo descendente	0.8972	19731.83	19730.9328
Arreglo casi ordenado	0.5982	390.4928	389.8946
Arreglo con duplicados	0.4983	16396.94	16396.4417
Promedio	0.73778	10551.0165	10550.27872

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

An illustration on the left side of the slide. It features a blue laptop with a white keyboard and a black screen, positioned above a blue server rack with multiple horizontal slots. The background is a solid light blue.

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

