

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.

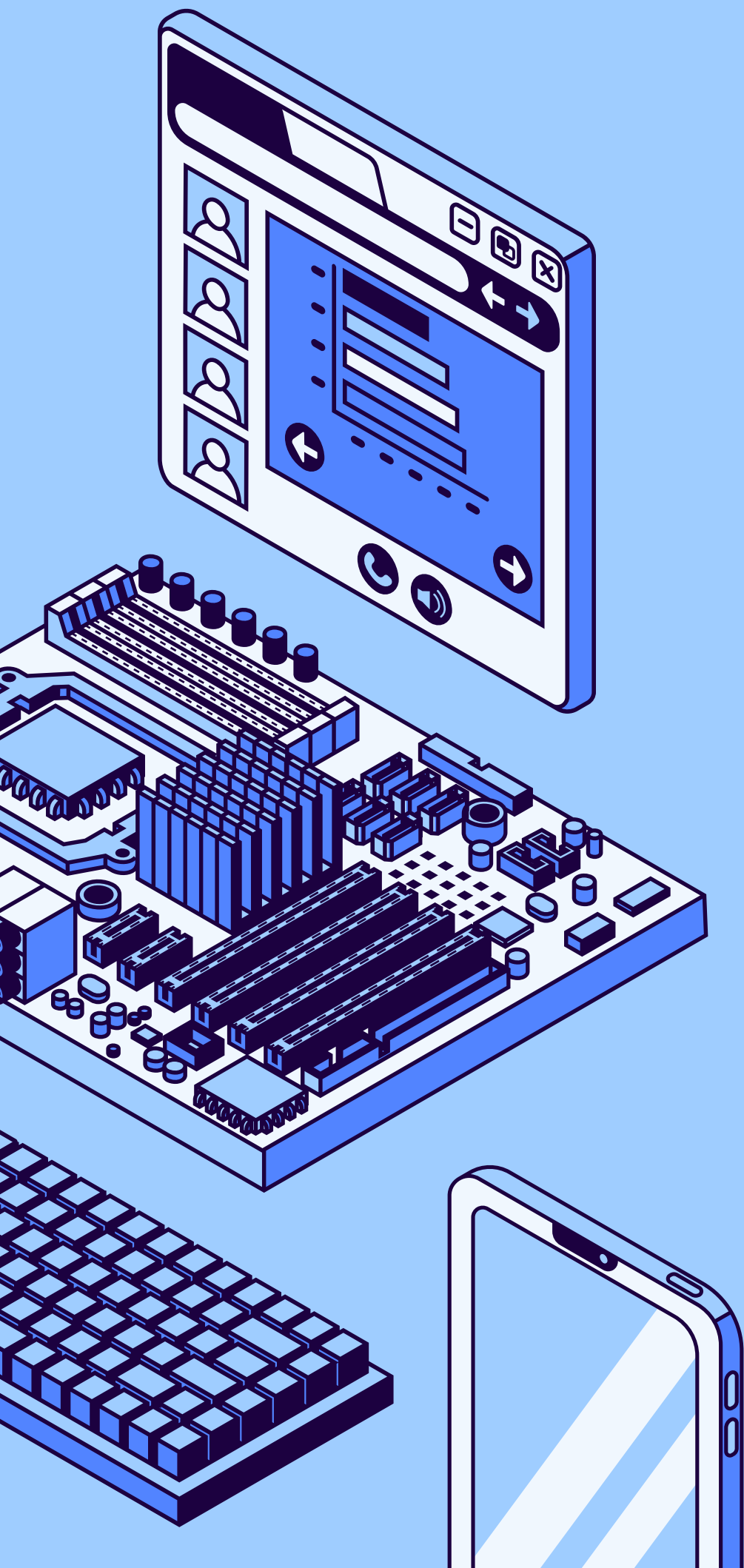


Counting Sort

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

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

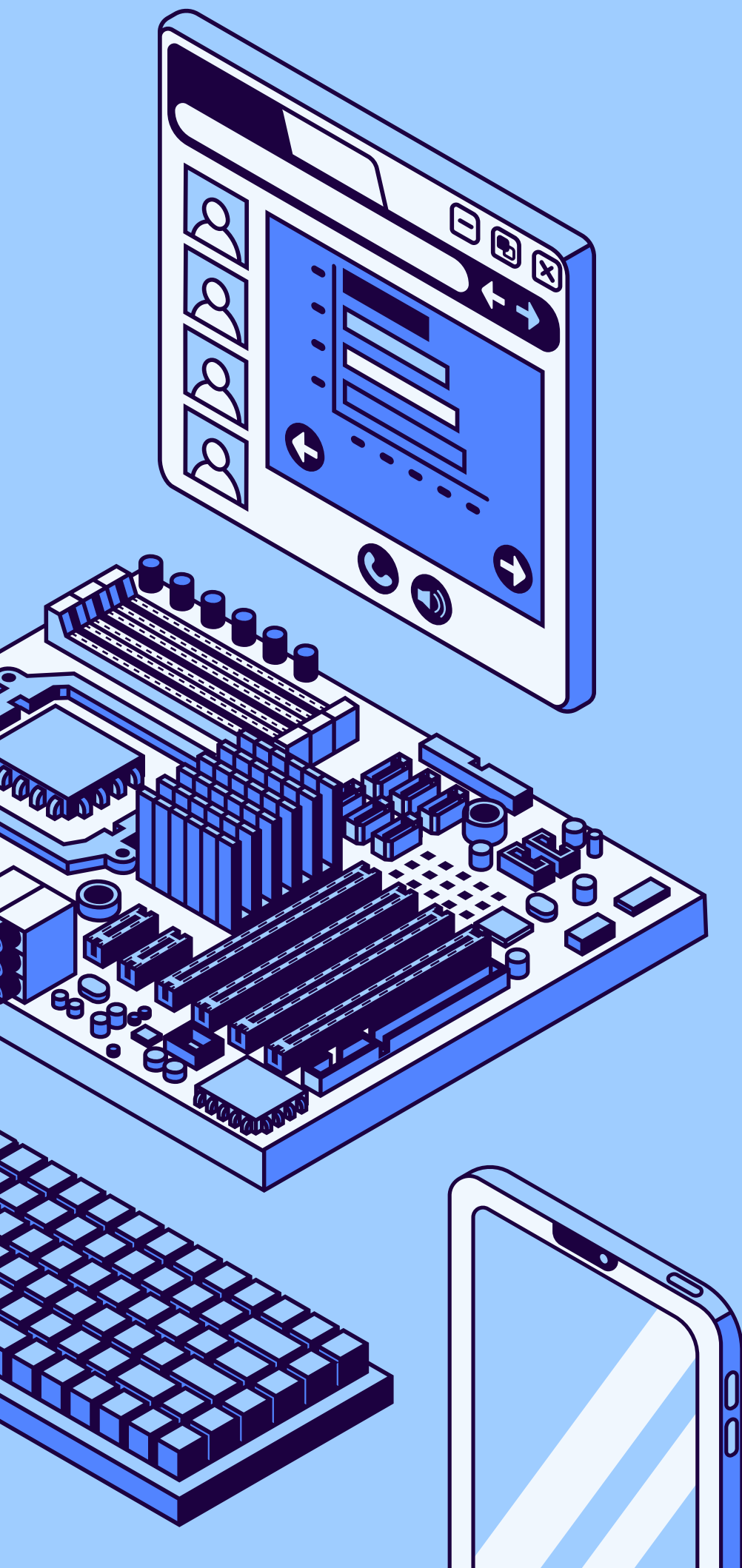
Complejidad de espacio: $O(k)$



Counting Sort



Counting Sort



inputArray

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

countArray

0	1	2	3	4	5
0	2	3	4	7	7

outputArray

0	1	2	3	4	5	6	7
0	0	2	2	3	3	3	5

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: $O(n)$

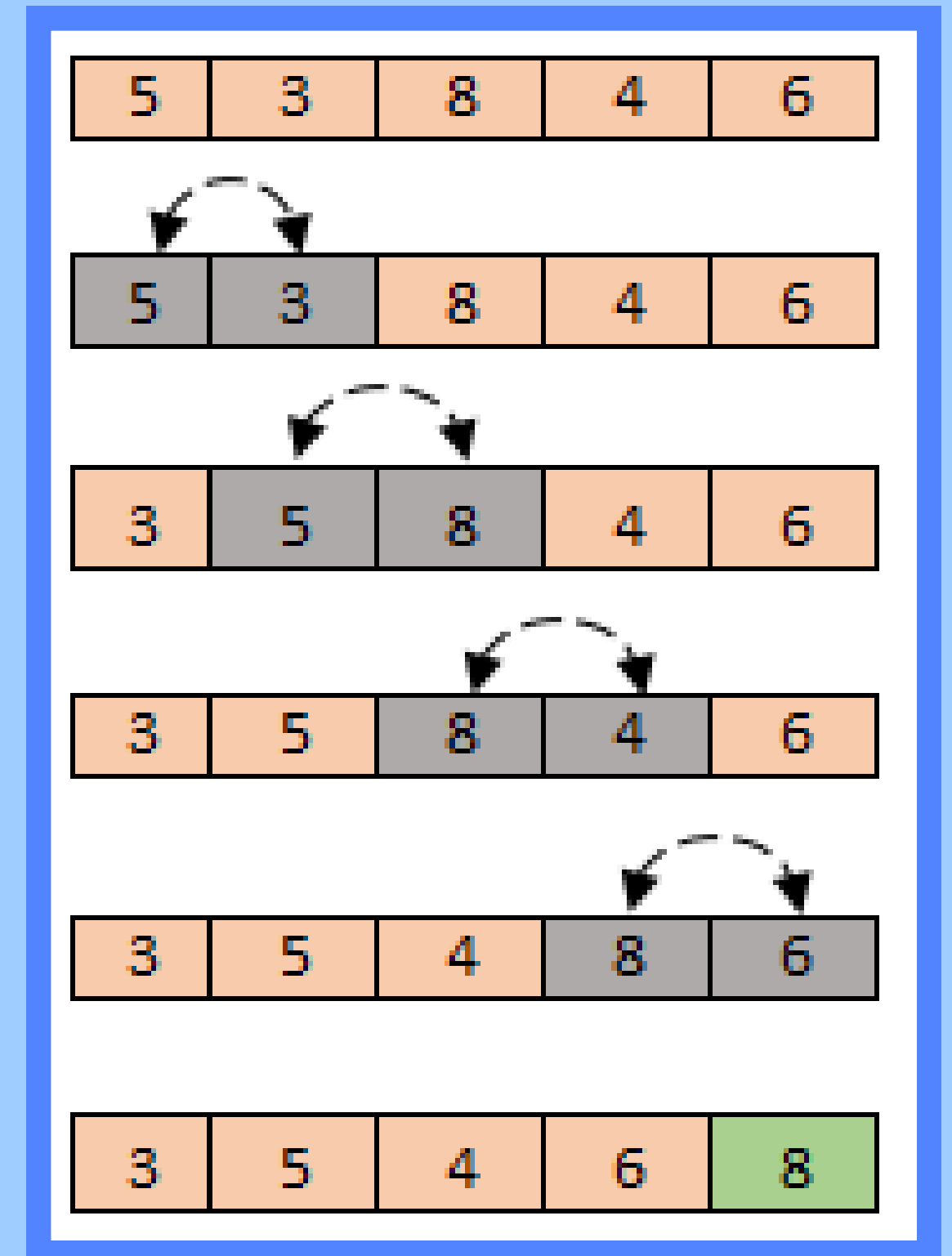
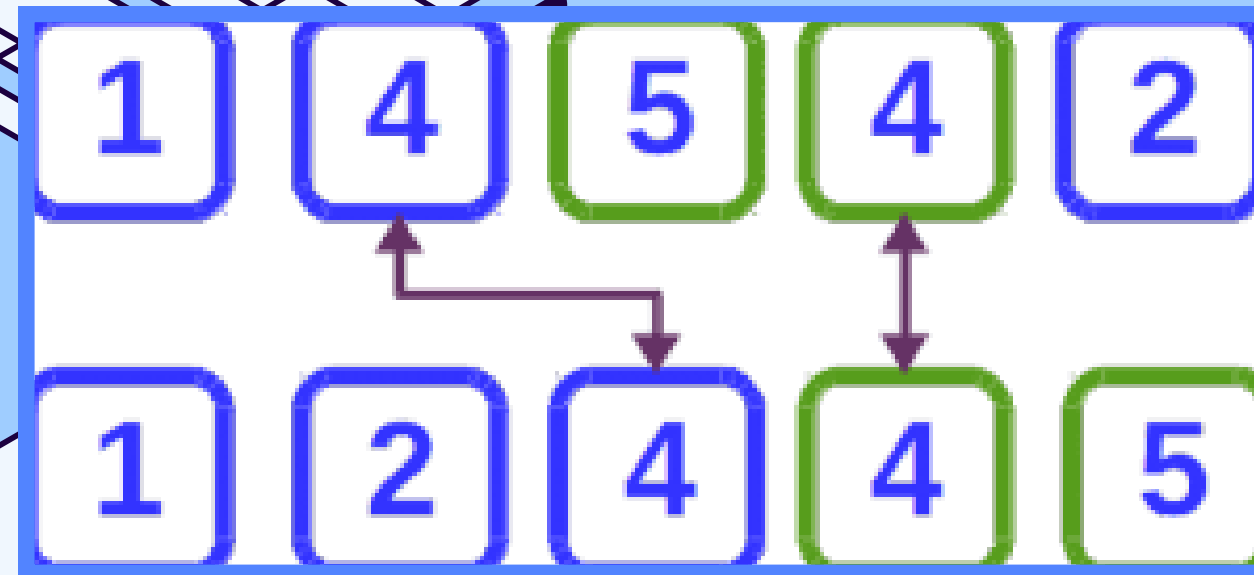
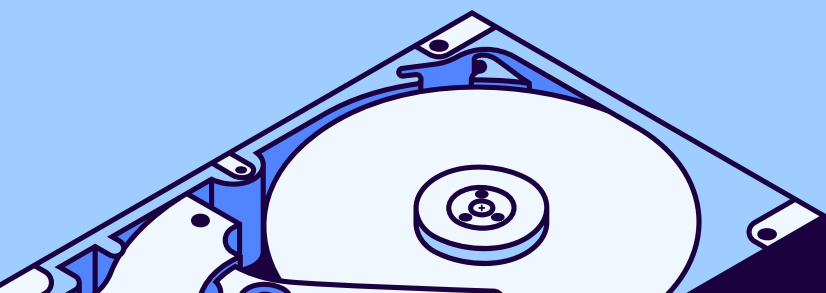
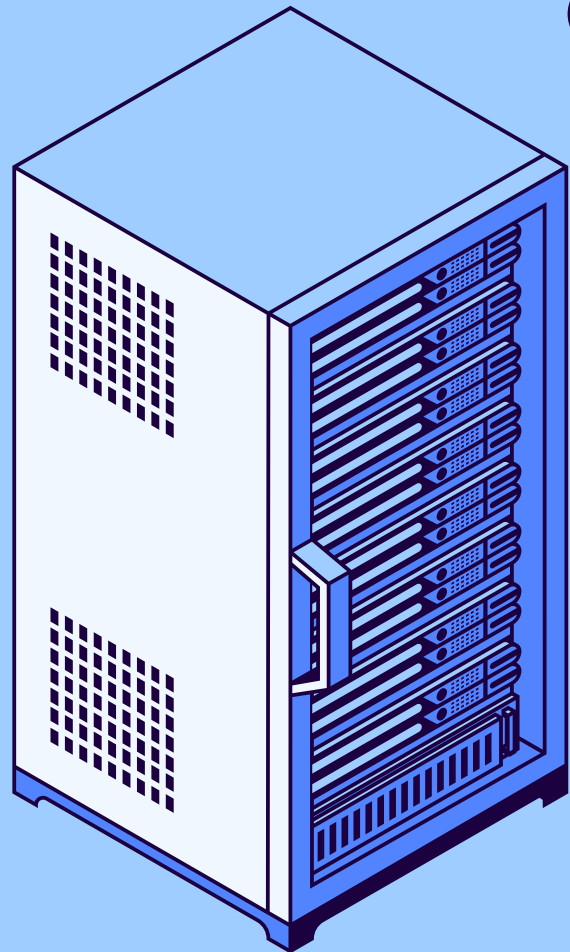
Peor caso: $O(n^2)$

Complejidad promedio: $O(n^2)$

Complejidad de espacio: $O(1)$

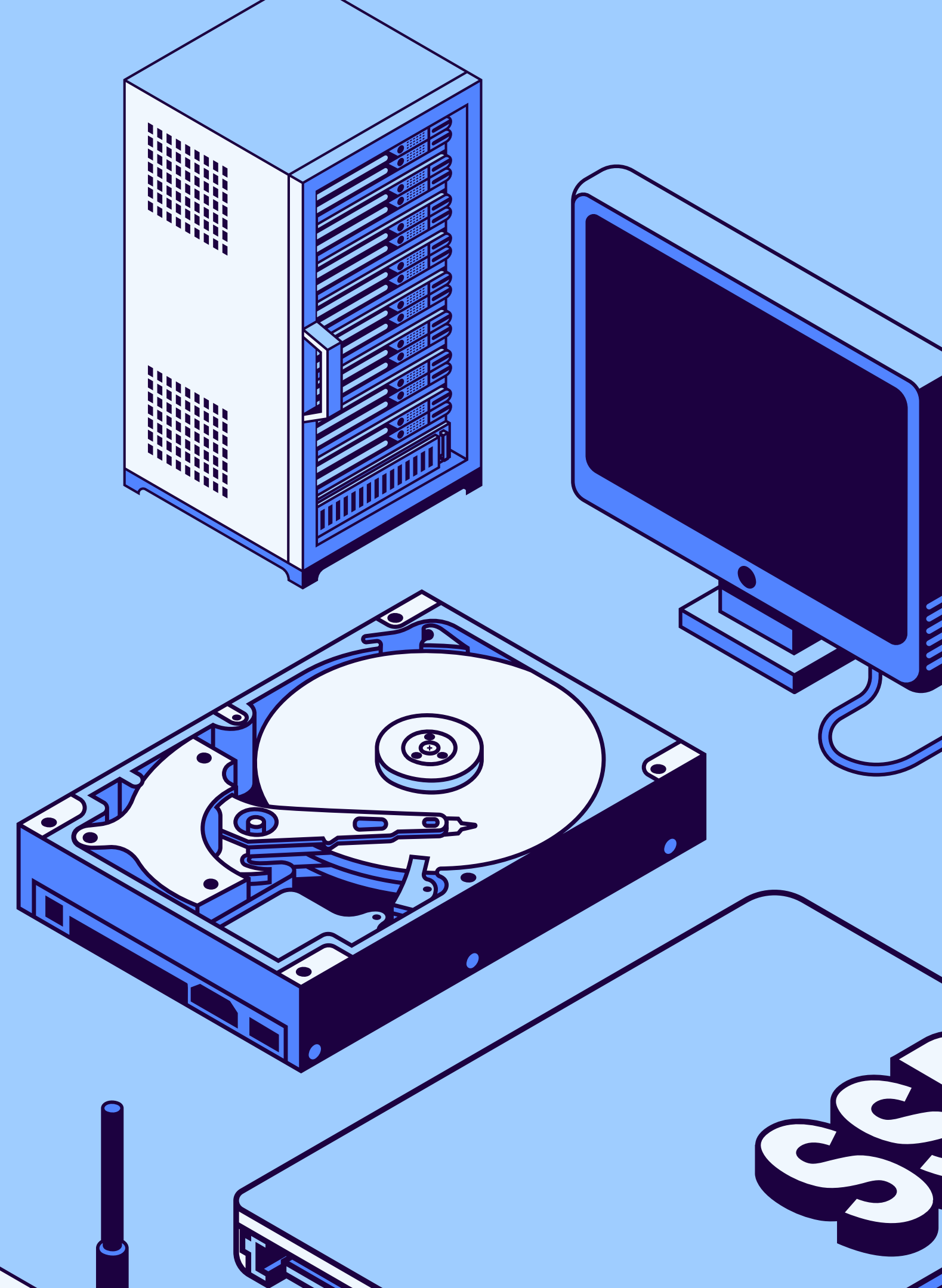


Intercambio Directo Bidireccional



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 main content area with a list of items, and a bottom navigation bar with icons for home, search, and a plus sign. 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 solid blue background.

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 three icons. 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 solid blue background.

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 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 central content area with a list and a large image, 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 sticks, a CPU, and capacitors. In the bottom left is a keyboard with square 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.

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

[illegible]



Intercambio Directo Bidireccional

Análisis

Tabla 5

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

[illegible]**Tabla 6**

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

[illegible][illegible]

Análisis

Tabla 8

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

[illegible]**Tabla 9**

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

[illegible]**Tabla 7**

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

[illegible]

Análisis

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

[illegible]

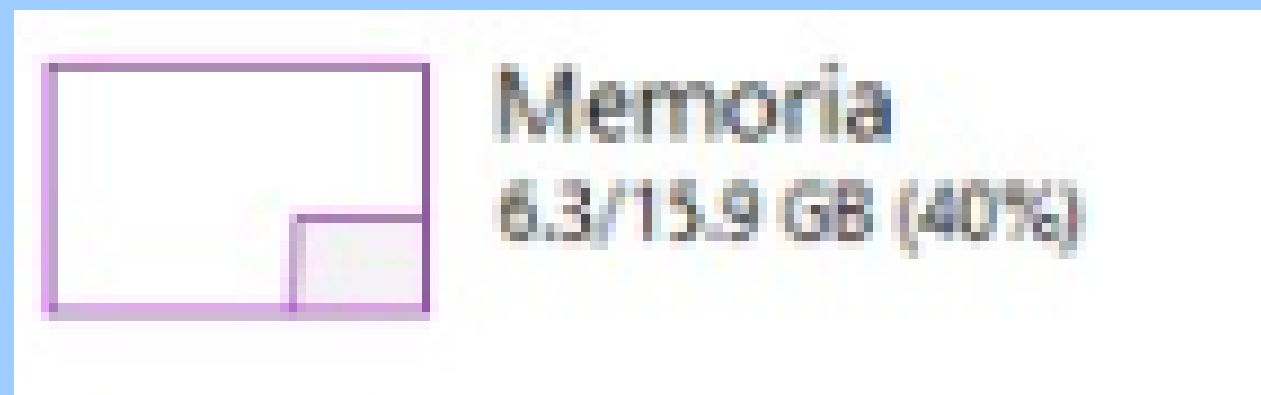
Tabla 12
Resultados de eficiencia en intercambios del algoritmo Shaking Sort cuando $n=100000$ y el rango=100000

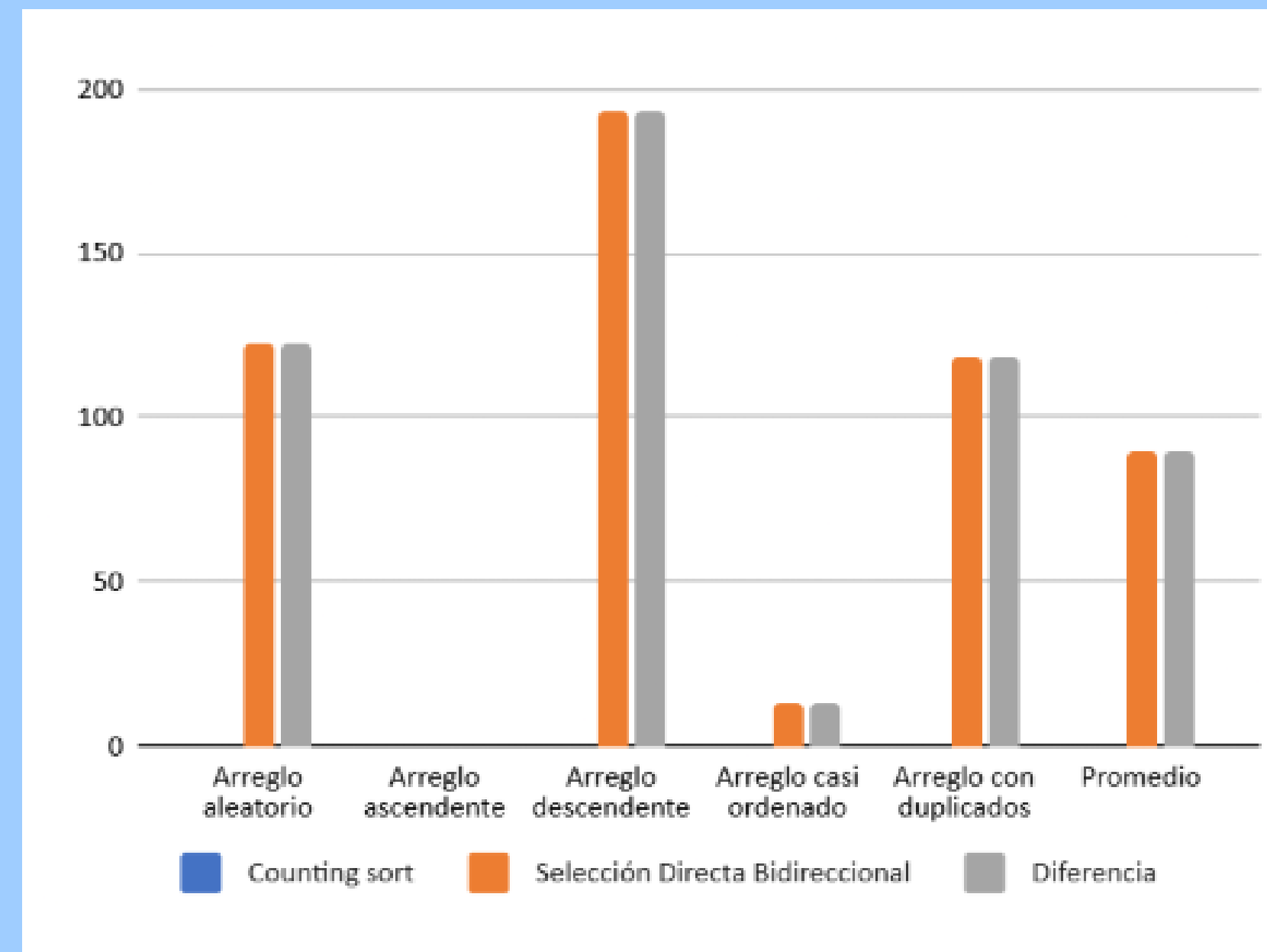
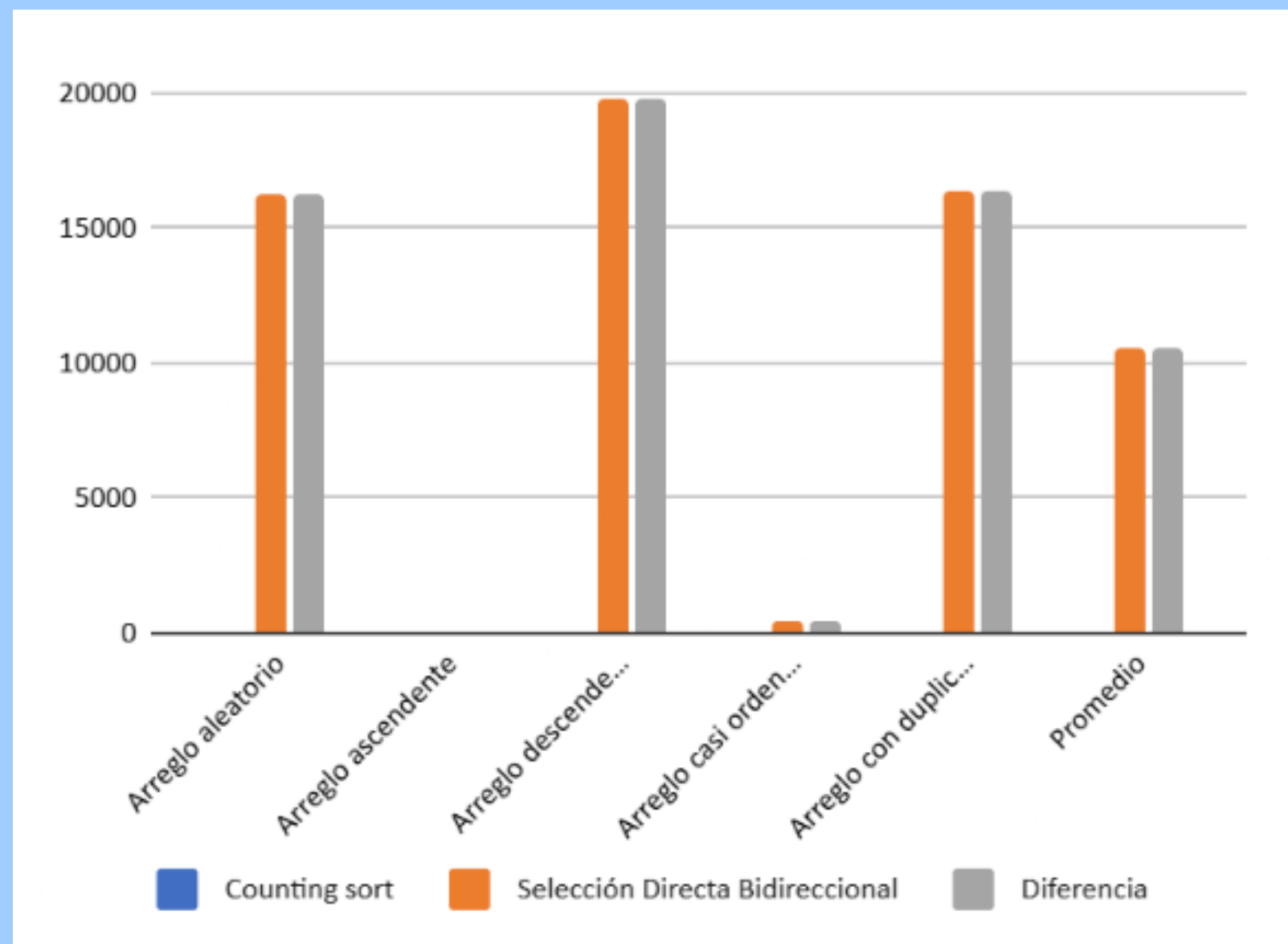
[illegible]

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

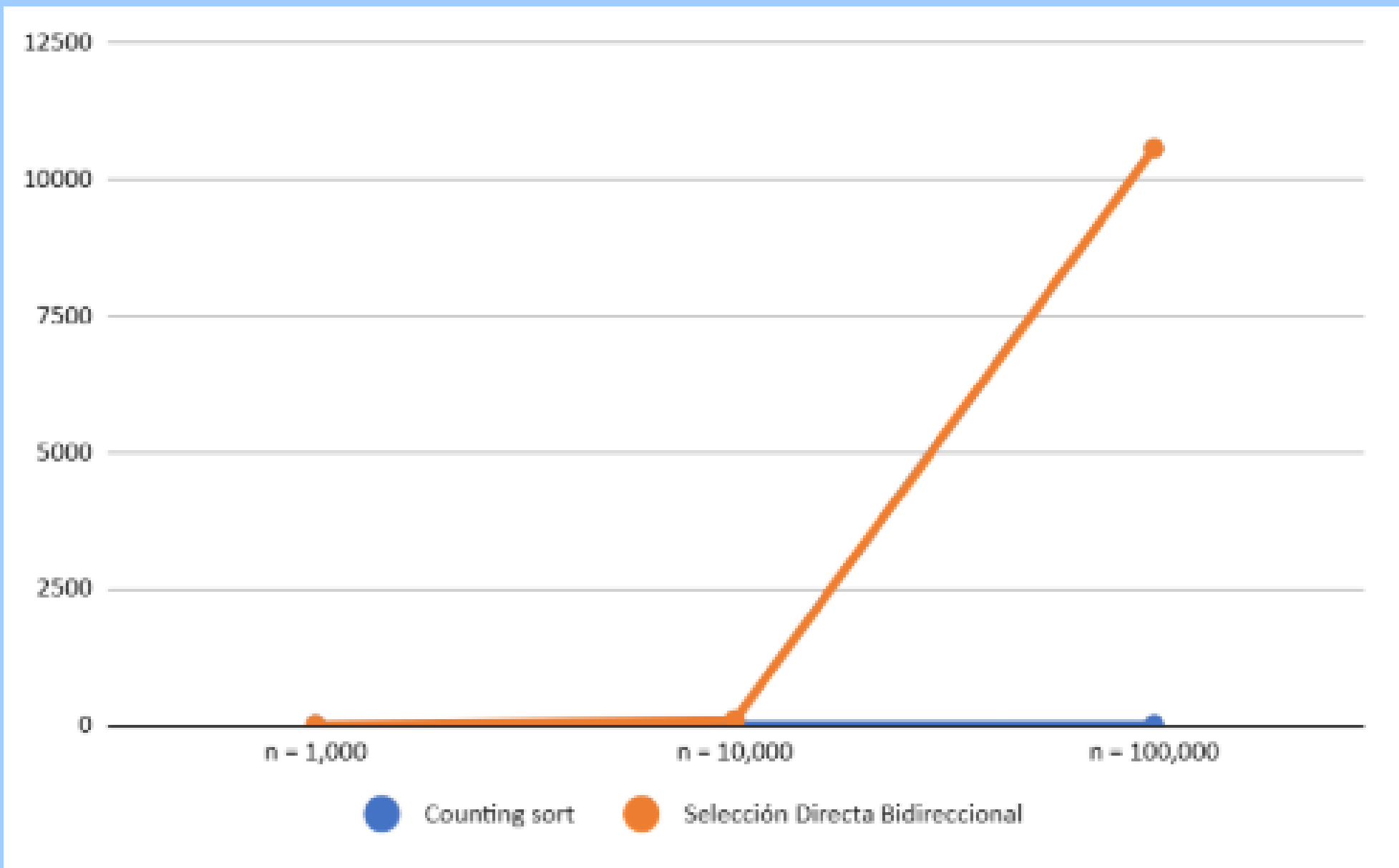
[illegible]

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	





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

A stylized illustration in the top-left corner of the slide. It features a blue laptop with a white keyboard and a black screen, positioned next to a blue server rack with multiple horizontal slots. The background is a light blue gradient.

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

