

ALGORITMO SELECTION SORT

Ana López
Óscar Blanco

DESCRIPCIÓN

Es un algoritmo de ordenamiento

Divide la lista en dos partes: una parte ordenada y otra desordenada

En cada iteración busca el elemento mínimo en la parte desordenada y lo intercambia por el primer elemento de la parte desordenada.

DESCRIPCIÓN TEÓRICA

Es un algoritmo de ordenamiento

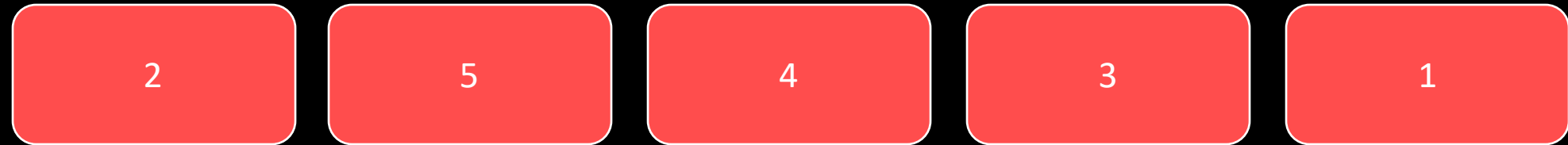
Divide la lista en dos partes: una parte ordenada y otra desordenada

En cada iteración busca el elemento mínimo en la parte desordenada y lo intercambia por el primer elemento de la parte desordenada.

IMPORTANCIA Y APLICACIONES PRÁCTICAS

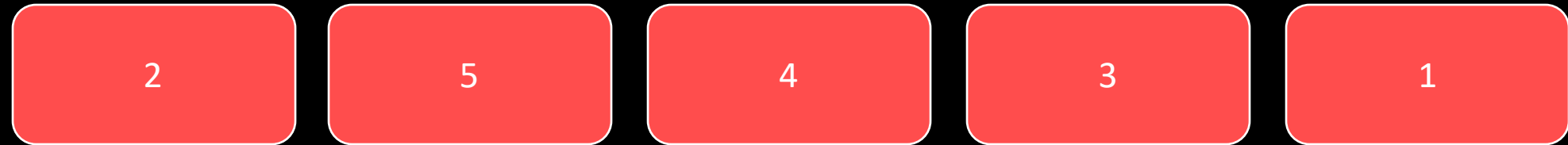
- Punto de partida para enseñar algoritmos de ordenación básica.
- Ineficiente en tiempo para conjuntos de datos grandes.
- Adecuado para listas pequeñas o parcialmente ordenadas, priorizando claridad sobre velocidad.
- Utilizado en algoritmos avanzados para identificar el elemento pivote .
- Bajo consumo de memoria.

EJEMPLO



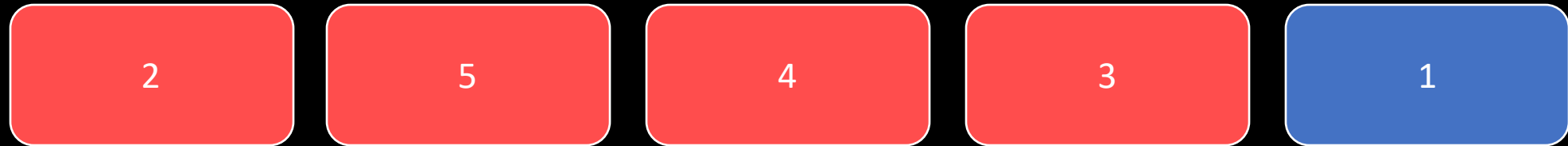
1.

IDENTIFICAR EL ELEMENTO MÍNIMO EN LA PARTE DESORDENADA



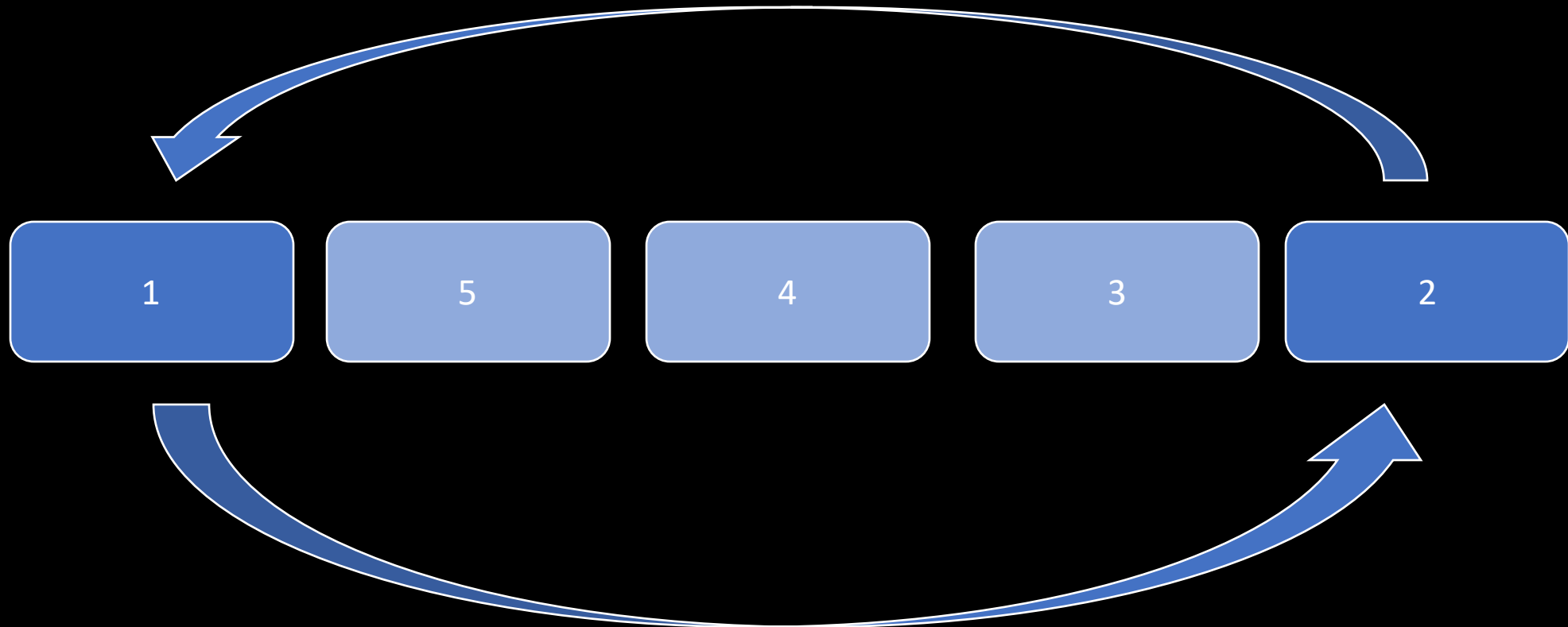
1.

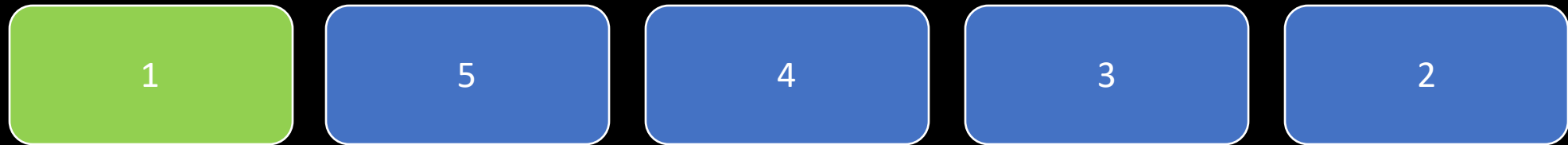
IDENTIFICAR EL ELEMENTO MÍNIMO EN LA PARTE DESORDENADA

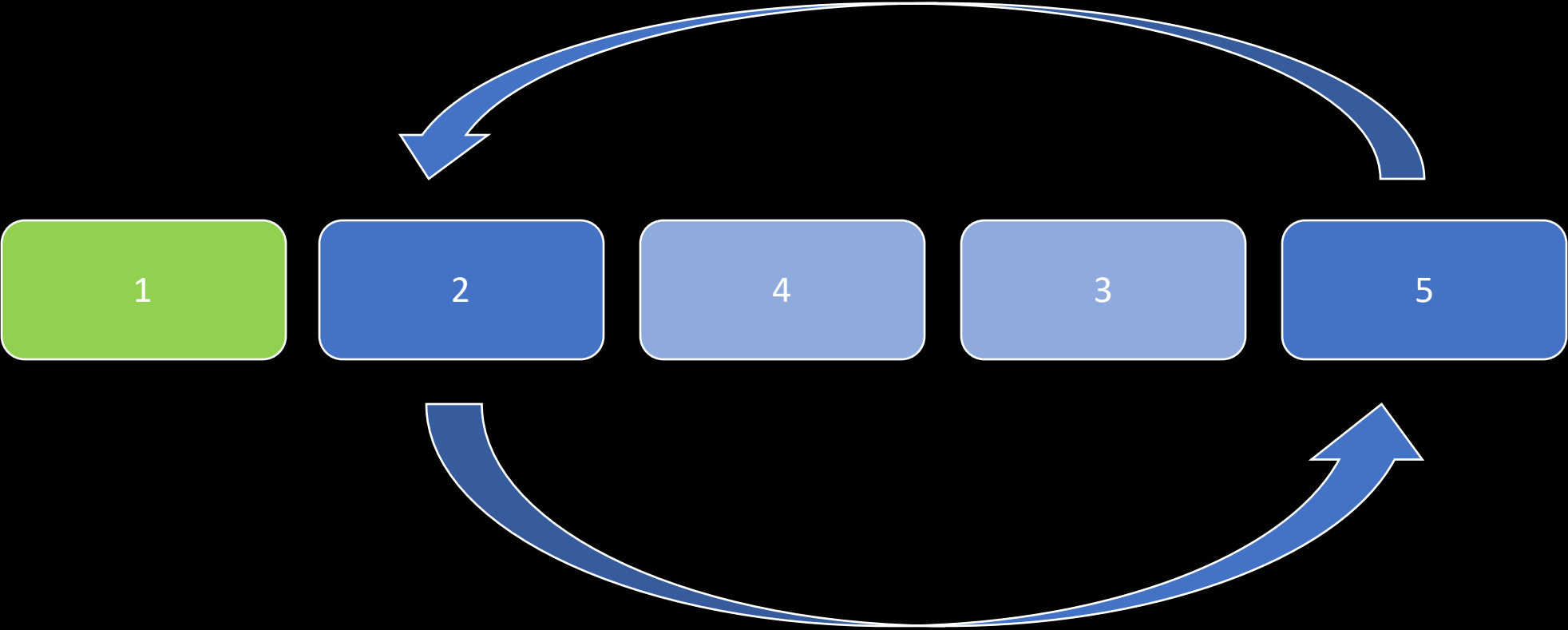


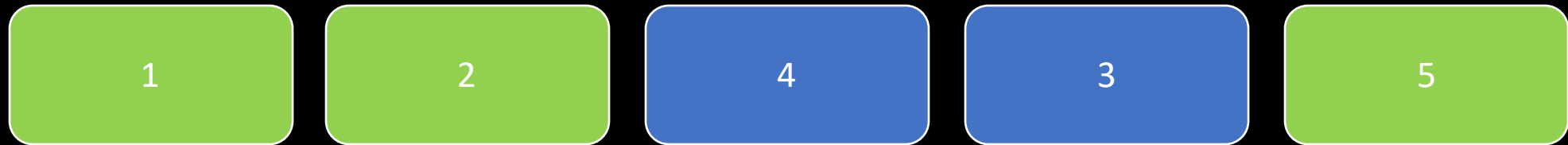
2.

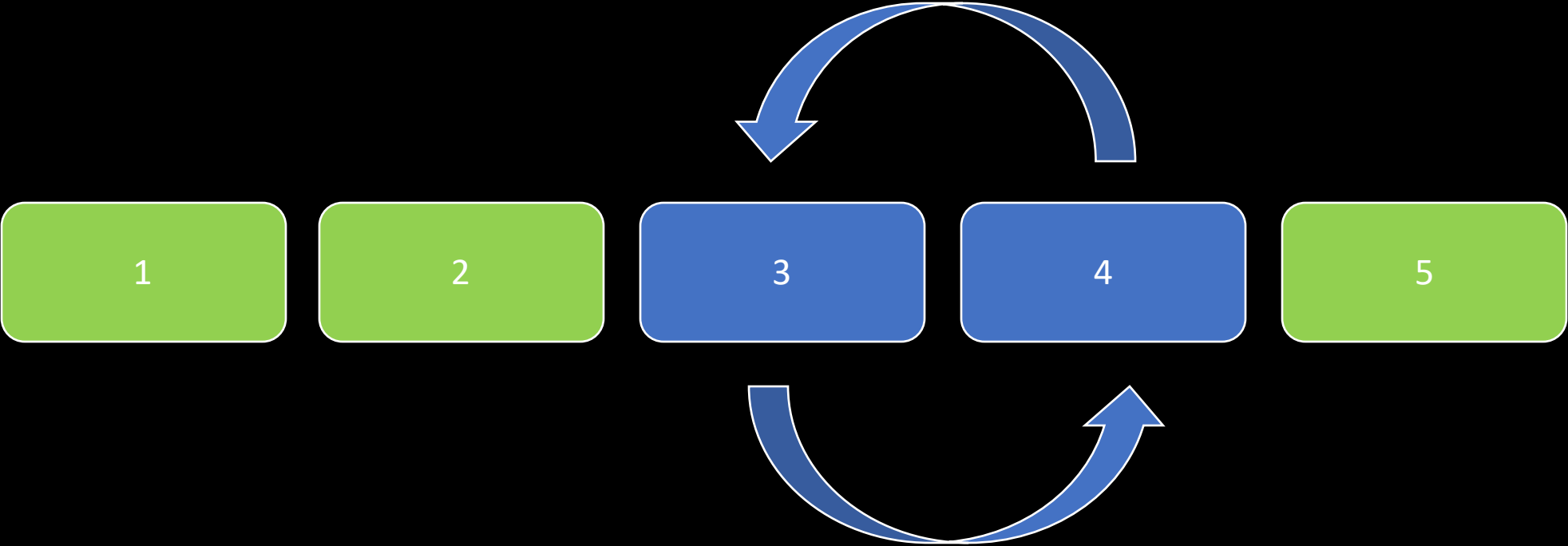
INTERCAMBIAR EL ELEMENTO MÍNIMO CON EL PRIMER ELEMENTO DE LA PARTE DESORDENADA











1

2

3

4

5

COMPLEJIDAD TEMPORAL

- ✓ MEJOR CASO: todos los números o la mayoría están ordenados.
- ✓ PEOR CASO: ningún número está ordenado o la secuencia de números está en un orden inverso, por lo que habrá que cambiar la posición en cada caso.
- ✓ CASO PROMEDIO: la mayoría de los elementos estarán desordenados, por lo que casi seguro habrá un cambio en cada posición

COMPLEJIDADESPACIAL

VENTAJAS Y DESVENTAJAS

VENTAJAS

- Fácil de implementar y entender.
- Buen algoritmo para propósitos educativos.
- No requiere memoria adicional.

DESVENTAJAS

- Ineficiente en listas grandes.
- Su rendimiento empeora significativamente a medida que aumenta el tamaño de la lista.
- Depende del orden inicial.
- No adecuado para datos sensibles al orden.

IMPLEMENTACIÓN DE JAVA

```
1 package algoritmoSeleccion;
2
3 public class seleccion {
4
5     void sort(int array[]) {
6         int numero = array.length;
7
8         for (int i = 0; i < numero-1; i++)
9         {
10             int minimo = i;
11             for (int j = i+1; j < numero; j++)
12                 if (array[j] < array[minimo])
13                     minimo = j;
14
15             int temporal = array[minimo];
16             array[minimo] = array[i];
17             array[i] = temporal;
18         }
19     }
20
21     void imprimirArray(int array[]) {
22         int numero = array.length;
23         for (int i=0; i<numero; ++i)
24             System.out.print(array[i]+" ");
25         System.out.println();
26     }
27
28     public static void main(String args[]) {
29         seleccion srt = new seleccion();
30         int array[] = {6,25,12,22,11,15,54,88,8,1,55,4,124,78,9};
31         srt.sort(array);
32         System.out.println("Array ordenado");
33         srt.imprimirArray(array);
34     }
35 }
```