



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

## Estructura de datos

Grupo N°3

### INTEGRANTES:

- Xavier Cordova
- Juan Gallardo
- Nicole Lara
- Camila Naspud
- Adrian Paez

## Quicksort



Charles Antony Richard Hoare

# Quicksort

EL ORDENAMIENTO RÁPIDO

# 01

## DEFINICIONES

- Quicksort fue diseñado en los años 60s por Charles Antony Richard Hoare que un científico en computación.
- Quicksort se basa en la técnica de divide y vencerás, que permite, en promedio, ordenar  $n$  elementos en un tiempo proporcional a  $n \log n$
- El método de ordenamiento Quicksort es actualmente el más eficiente y veloz de los métodos de ordenación interna.
- Conocido como método rápido y de ordenamiento por partición.



## 02

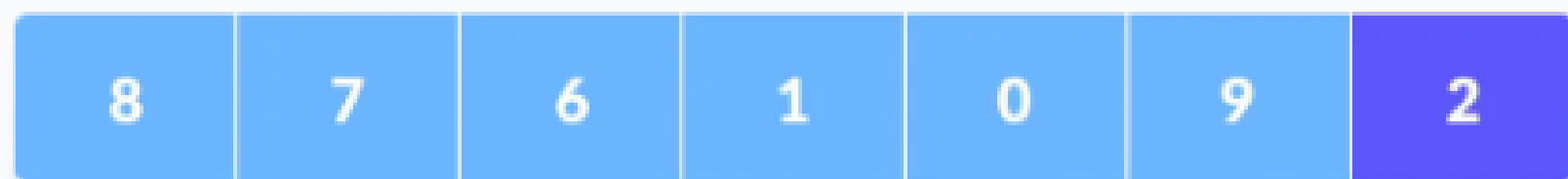
## DESCRIPCIÓN DEL MÉTODO

Lo que hace este algoritmo es dividir recurvisamente el vector en partes iguales, indicando un elemento de inicio, fin y un pivote que nos permitira segmentar nuestra lista. Una vez dividida, lo que hace, es dejar todos los mayores que el pivote a su derecha y todos los menos a su izquierda. Al final el algoritmo, nuestros elementos estan ordenados

01

## PASO 1: SELECCIONAR EL ELEMENTO PIVOTE

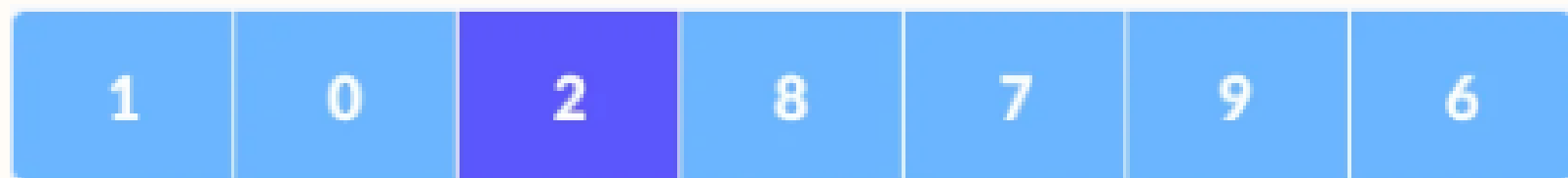
Seleccionaremos el elemento más a la derecha del array como elemento pivote.



02

## PASO 2. REORGANIZAR EL ARRAY

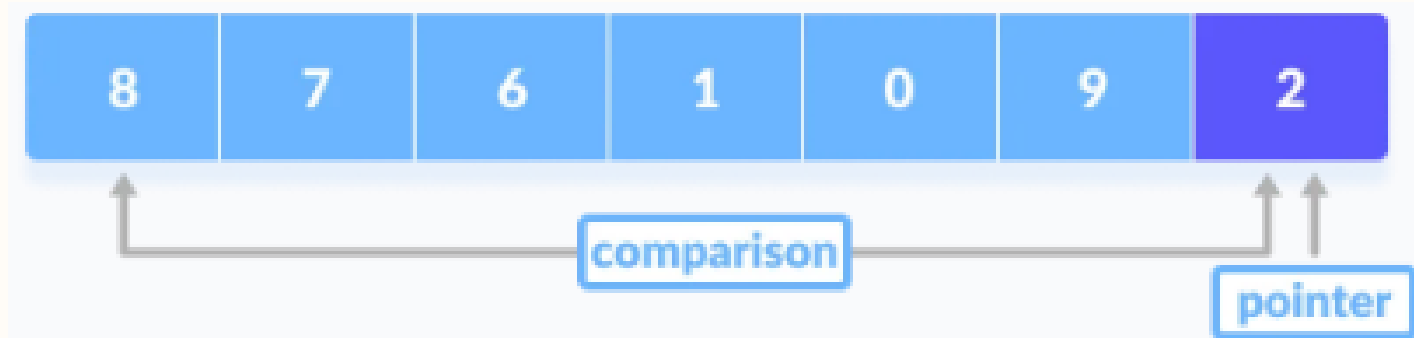
Ahora los elementos de la matriz se reorganizan de manera que los elementos que son más pequeños que el pivote se ponen a la izquierda y los elementos mayores que el pivote se ponen a la derecha.



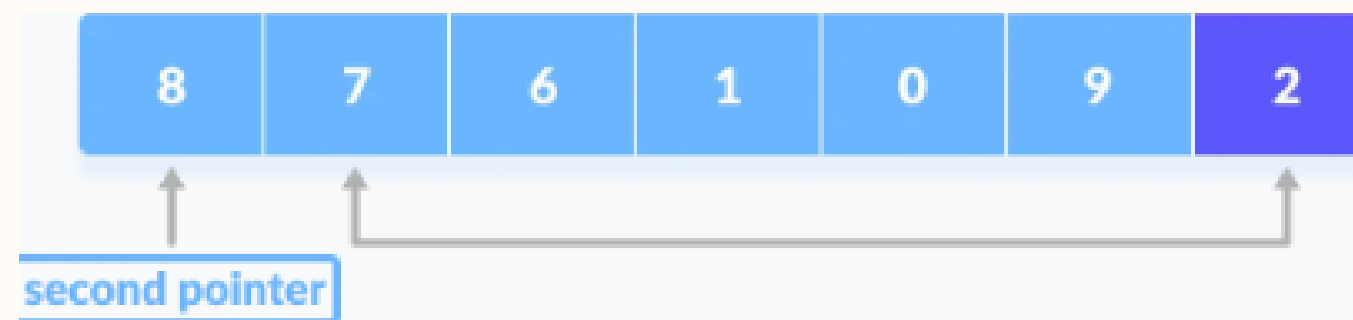
# PASOS DEL MÉTODO

## ¿CÓMO SE ORDENA EL ARRAY?

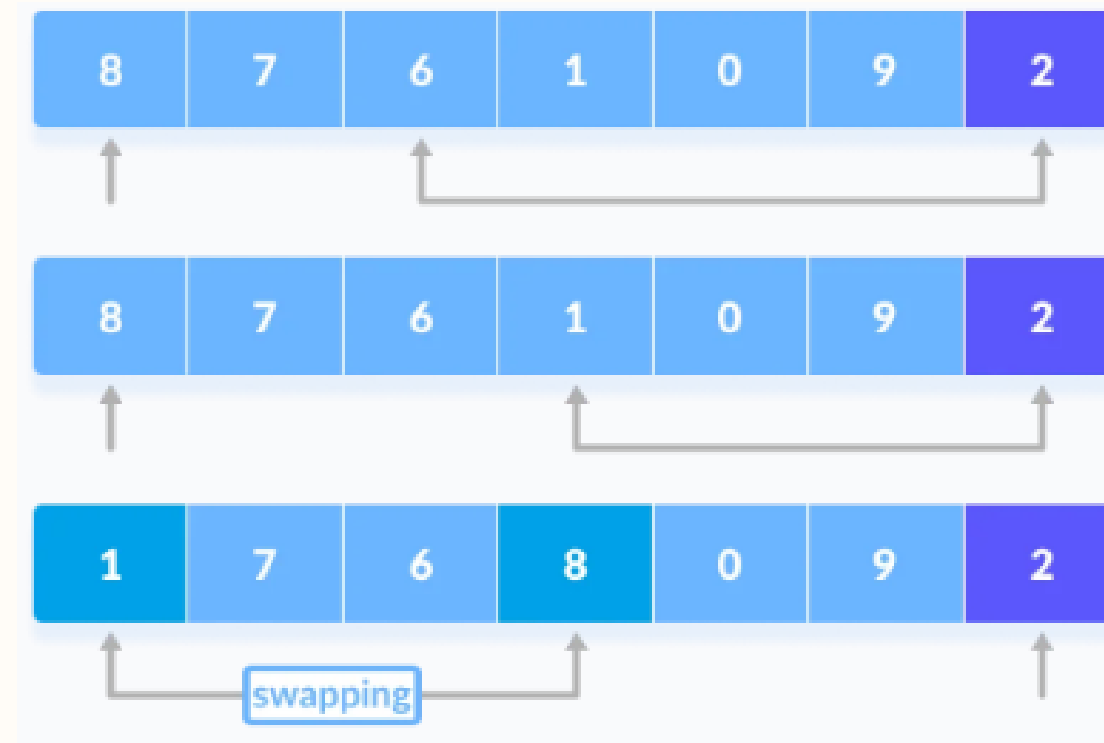
- Se fija un puntero en el elemento pivote. El elemento pivote se compara con los elementos a partir del primer índice.



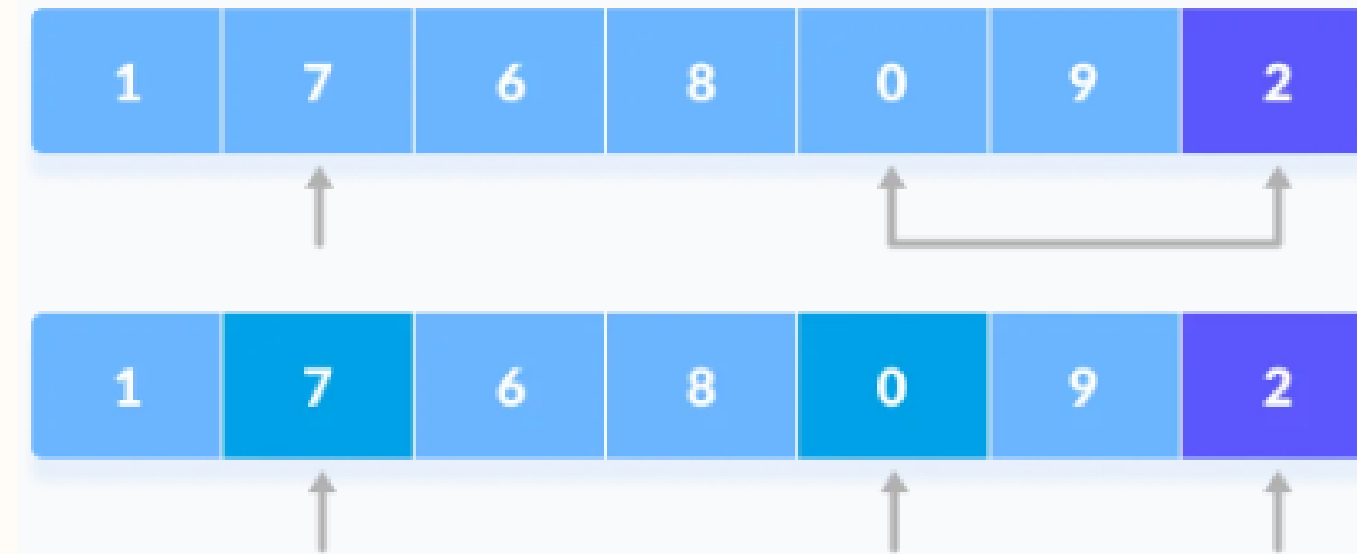
- Si el elemento es mayor que el elemento pivote, se fija un segundo puntero para ese elemento.



- Ahora, el pivote se compara con otros elementos. Si se llega a un elemento menor que el elemento pivote, el elemento menor se intercambia con el elemento mayor encontrado anteriormente.



- De nuevo, se repite el proceso para establecer el siguiente elemento mayor como segundo puntero. Y, se intercambia con otro elemento más pequeño.

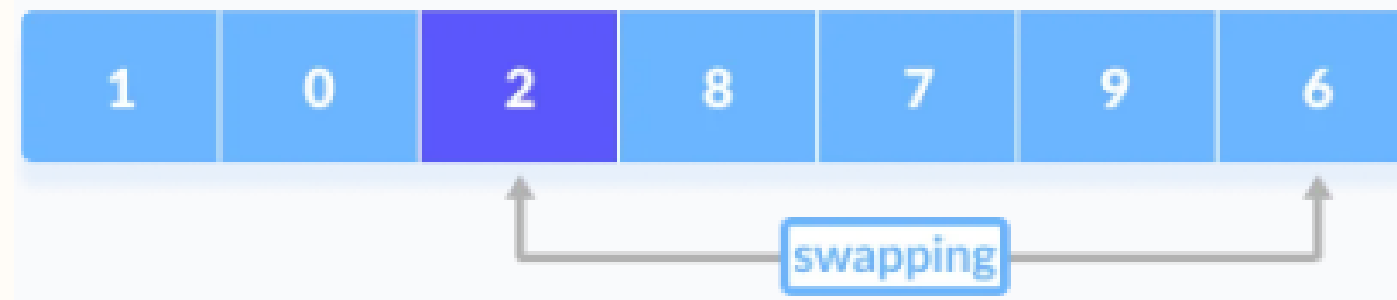


- El proceso continúa hasta que se alcanza el penúltimo elemento.





- Finalmente, el elemento pivote se intercambia con el segundo puntero.

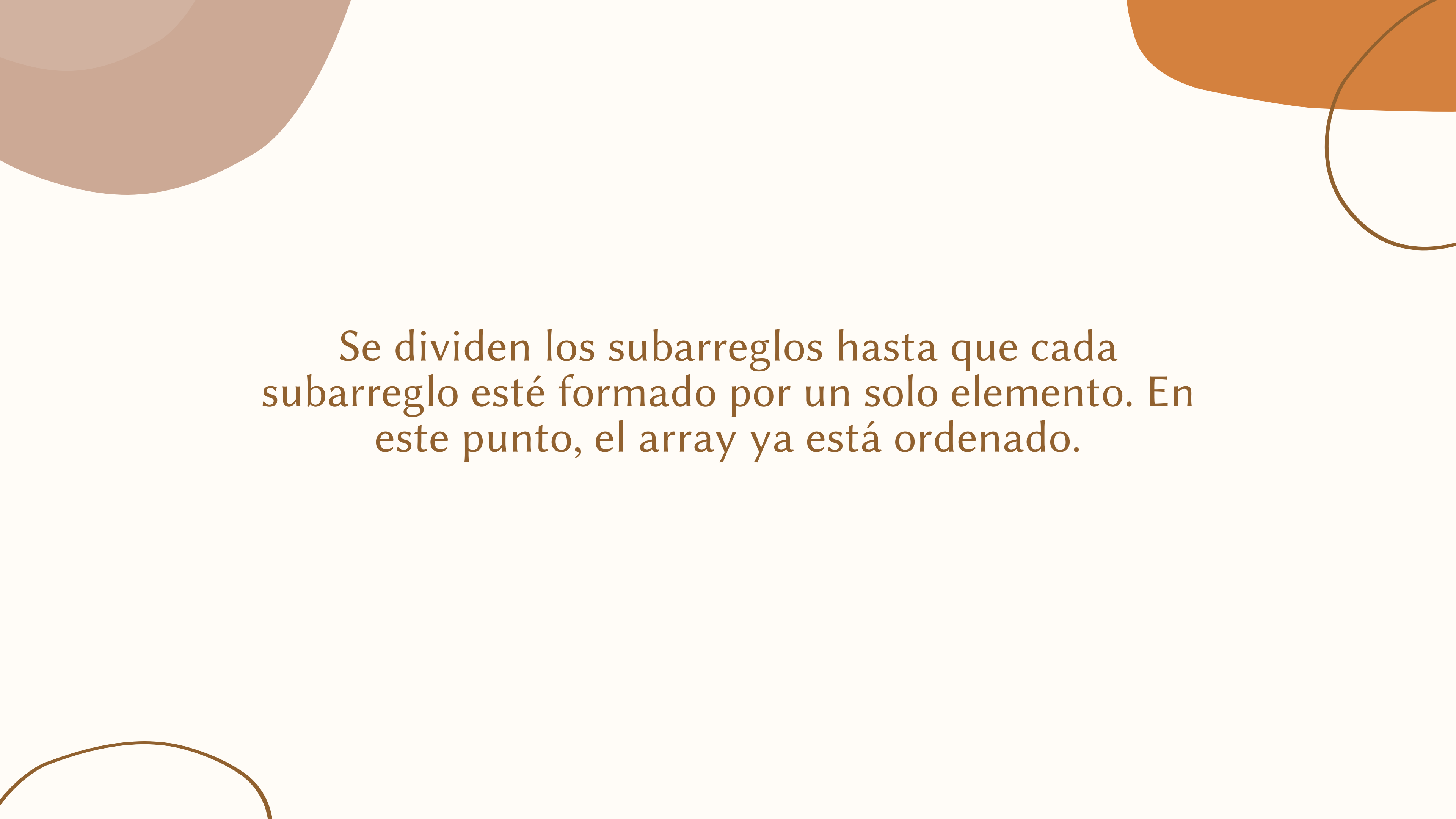


### PASO 3: DIVIVIR SUBARREGLOS

03

Se vuelven a elegir los elementos de pivote para las subpartes izquierda y derecha por separado. Y se repite el paso 2.





Se dividen los subarreglos hasta que cada subarreglo esté formado por un solo elemento. En este punto, el array ya está ordenado.

# Ventajas

- Requiere de pocos recursos en comparación a otros métodos de ordenamiento
- Ciclo interno es extremadamente corto
- En la mayoría de los casos, se requiere aproximadamente  $N \log N$  operaciones
- No se requiere de espacio adicional durante ejecución

# Desventajas

- Se complica la implementación si la recursión no es posible
- Un simple error en la implementación puede pasar sin detección, lo que provocaría un rendimiento pésimo.
- No es útil para aplicaciones de entrada dinámica, donde se requiere reordenar una lista de elementos con nuevos valores
- Se pierde el orden relativo de los elementos idénticos

# Aplicaciones para usar Quicksort

- Para ordenar una lista de números o nombres
- Utilización antes de implementar una búsqueda binaria
- Utilizado como el método de ordenamiento en tarjetas gráficas

## 04 CONCLUSIÓN

El método de ordenamiento Quicksort es un método muy rápido ya que divide al arreglo en pequeños subgrupos y los va comprando hasta reducir los subgrupos a la mínima cantidad evitando hacer comparaciones innecesarias y pasadas de mas.