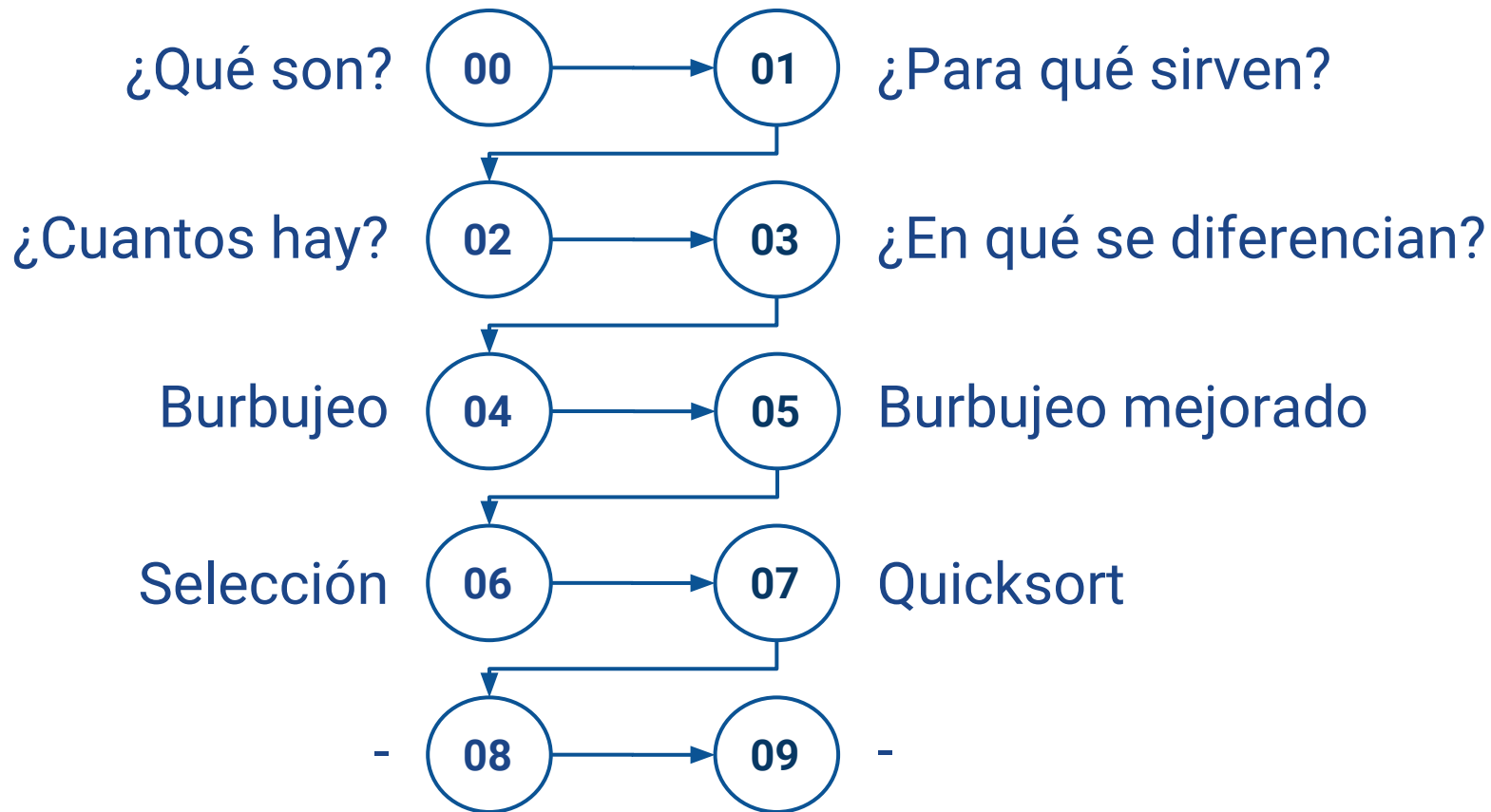


Algoritmos de Ordenamiento

Programación y Laboratorio I

Algoritmos de ordenamiento



¿Qué son?

Los algoritmos de ordenamiento son un conjunto de instrucciones que buscan organizar elementos en un orden particular (**criterio de ordenamiento**).

The image shows a screenshot of the YouTube channel page for 'Bizarrap', which has 15.7 million subscribers. The page is set to the 'VIDEOS' tab. Under the heading 'Videos subidos', there are four video thumbnails. A sorting menu is open on the right, showing options to sort videos by 'Más popular', 'Fecha de carga (más antiguos)', and 'Fecha de carga (más recientes)'. The 'Fecha de carga (más recientes)' option is currently selected and highlighted with a grey background. An orange rounded rectangle highlights the entire sorting menu area.

Bizarrap ●
15.7 M de suscriptores

SUSCRIBIRSE

PÁGINA PRINCIPAL VIDEOS LISTAS DE REPRODUCCIÓN COMUNIDAD CANALES MÁS INFORMACIÓN >

Videos subidos

QUEVEDO || BZRP Music
Sesión #52 3:24

VILLANO ANTILLANO || BZRP
Música Sesión #51 3:08

PAULO LONDRA || BZRP
Música Sesión #49 4:18

RESIDENTE || BZRP Music
Sesión #48 8:40

ORDENAR POR

- Más popular
- Fecha de carga (más antiguos)
- Fecha de carga (más recientes)

¿Para qué sirven?

El ordenamiento cumple un rol muy importante, ya sea como un fin en sí (ej. Ordenar una lista de mayor a menor) o como parte de otros procedimientos más complejos permitiendo simplificar la solución al contar con un conjunto ordenado de datos.

¿Cuántos hay?

Bubble sort

Heap sort

Selection sort

Counting sort

Insertion sort

Radix sort

Merge sort

Bucket sort

Quick sort

...

Clasificación

- La cantidad de **intercambios**: esta es la cantidad de veces que el algoritmo intercambia elementos.
- El número de **comparaciones**: este es el número de veces que el algoritmo compara elementos.
- La cantidad de **espacio** adicional requerido: algunos algoritmos pueden ordenar una lista sin la necesidad de crear listas nuevas.
- Utilizan **recursividad** o no: algunos algoritmos usan técnicas recursivas y otros no.

Burbujeo

- Se basa en recorrer el array ("realizar una pasada") un cierto número de veces, comparando pares de valores que ocupan posiciones adyacentes (0-1, 1-2, ...).
- Si ambos datos no están ordenados, se intercambian.
- Esta operación se repite $n-1$ veces, siendo n el tamaño del conjunto de datos de entrada.

Burbujeo mejorado

Existe una forma muy obvia para mejorar el algoritmo de la burbuja.

Basta con tener en cuenta la posibilidad de que el conjunto está ordenado en algún paso intermedio.

Si el bucle interno no necesita realizar ningún intercambio en alguna pasada, el conjunto estará ya ordenado.

Selección

- Buscas el elemento más pequeño de la lista.
- Lo intercambias con el elemento ubicado en la primera posición de la lista.
- Buscas el segundo elemento más pequeño de la lista.
- Lo intercambias con el elemento que ocupa la segunda posición en la lista.
- Repites este proceso hasta que hayas ordenado toda la lista.

Quicksort

El método Quicksort debe su nombre a que este método, en general, puede ordenar una lista de datos mucho más rápido que cualquier otro método.

Se basa en la estrategia típica de "divide y vencerás", la lista a ordenar se divide en dos partes: una contendrá todos los valores menores o iguales a un cierto valor (que se suele denominar pivote) y la otra con los valores mayores que dicho valor.

Quicksort

El primer paso es dividir la lista original en dos sublistas y un valor que sirve de separación, esto es, el pivote.

Así, la lista se dividirá en tres partes:

- La parte izquierda, que contendrá valores inferiores o iguales al pivote
- El pivote
- La parte derecha, que contiene valores superiores o iguales al pivote

El primer paso es dividir la lista original en dos sublistas y un valor que sirve de separación, esto es, el pivote.

Así, la lista se dividirá en tres partes:

- La parte izquierda, contendrá valores inferiores o iguales al pivote
- El pivote
- La parte derecha, que contiene valores superiores o iguales al pivote

Quicksort

A continuación, para cada uno de estas listas volveremos a escoger un pivote y repetiremos la misma operación.

Llegará un momento en que las listas resultantes tengan o bien un elemento o bien ninguno (si ya no existe un elemento con el que compararlos).

El resto de los elementos habrán sido separados como pivotes en algún punto del algoritmo y ya se encontrarán en su posición correcta.

Quicksort

