

Curso de Programación en Java



Juan Francisco Maldonado León
Arquitecto de Software



Fundamentos de **Programación**



Juan Francisco Maldonado León
Arquitecto de Software

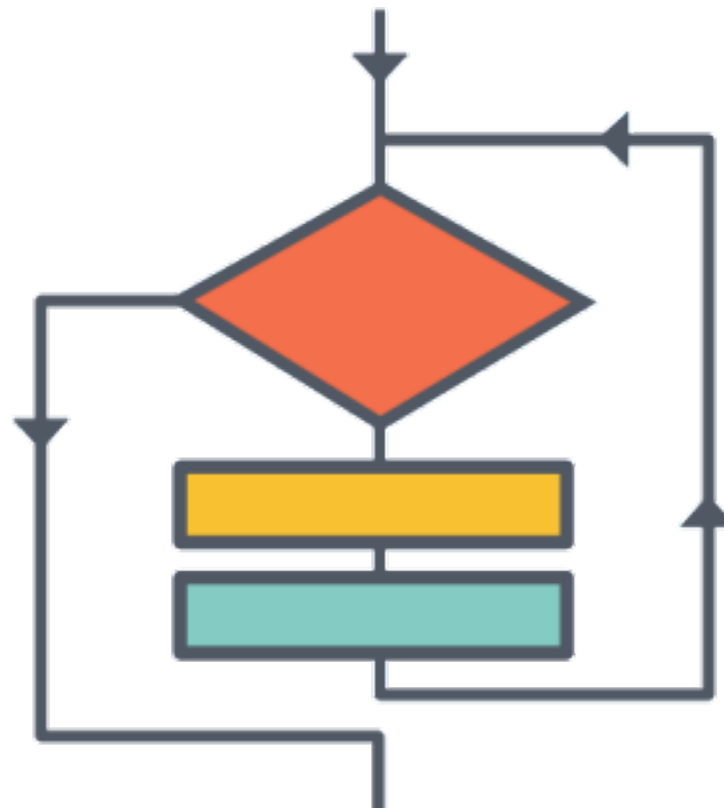


Algoritmos

Fundamentos de **Programación**

Definición

Conjunto ordenado y finito de operaciones que permite hallar la **solución de un problema**. (R.A.E)



Algoritmos

Fundamentos de **Programación**

Definición

Es un conjunto prescrito de **instrucciones** o reglas bien definidas, ordenadas y **finitas** que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una **solución**.

Algoritmos

Fundamentos de **Programación**

Componentes de un Algoritmo

- **Datos de Entrada** con los cuales operara el algoritmo.
- **Instrucciones** o pasos elementales secuenciales que componen el algoritmo.
- **Salida**, resultado que entrega el algoritmo.

Algoritmos

Fundamentos de **Programación**

Características de un Algoritmos

- Tiene que ser preciso, sin dar lugar a ambigüedades.
- Tiene que estar bien definido.
- Tiene que ser finito.
- Puede tener cero o mas datos de entrada.
- El algoritmo es independiente según donde lo implemente
- Debe producir un resultado.

Algoritmos

Fundamentos de **Programación**

Problema

Determinar si el número 5 es primo

Solución Ambigua

Divida el 5 entre sus anteriores buscando aquellos que lo dividan exactamente.

Algoritmos

Fundamentos de **Programación**

Problema

Determinar si el número 5 es primo

Solución Correcta

Divida el número 5 entre cada uno de los números enteros que lo precedan (1, 2, 3, 4). Si una de las divisiones es exacta, la respuesta es no. Si no es así, la respuesta es sí.

Algoritmos

Fundamentos de **Programación**

Problema

Determinar la suma de todos los números enteros.

Solución ?

Algoritmos

Fundamentos de **Programación**

Problema

Determinar la suma de todos los números enteros.

Solución ?

En este caso no se puede determinar un algoritmo para resolver este problema. Un algoritmo debe alcanzar la solución en un tiempo finito, situación que no se cumplirá en el ejemplo ya que los números enteros son infinitos.

Algoritmos

Fundamentos de **Programación**

Ingredientes

- Sartén
- 2 papas
- Aceite
- Sal



Receta de cocina

- Pelar las dos papas.
- Cortar papas en bastones.
- Verter Aceite en sartén
- Calentar aceite en sartén 5 min.
- Poner papas en sartén durante 10 min
- Sacar papas
- Poner un cucharada de sal

Algoritmos

Fundamentos de **Programación**



Algoritmos

Fundamentos de **Programación**

Si en particular, el problema va a ser resuelto utilizando una computadora, el conjunto de **pasos elementales** conocidos es muy reducido, lo que implica un alto grado de detalle para los algoritmos.

Se considera entonces como un paso elemental aquel que no puede volver a ser dividido en otros más simples.

Algoritmos

Fundamentos de **Programación**

La **eficiencia** es el factor que mide la calidad y rendimiento de un algoritmo.



Capacidad para realizar o cumplir adecuadamente una función con el menor esfuerzo.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

- Lenguaje Natural.
- Pseudocódigo.
- Diagramas de Flujo
- Lenguajes de programación

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Lenguaje Natural.

- Impreciso puede tener ambigüedades
- No universal, distintas personas describirán el proceso de manera diferente.
- No estructurado, la descripción no esta expresada en función de componentes simples

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Pseudocódigo.

Inicio

Si numero es mayor a cero entonces
 imprimir “Positivo”

Si no
 imprimir “Negativo”

Fin

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

- Representación gráfica de un algoritmo
- Universal UML
- Estimula el pensamiento analítico en el momento de estudiar un algoritmo.

Algoritmos

Fundamentos de **Programación**

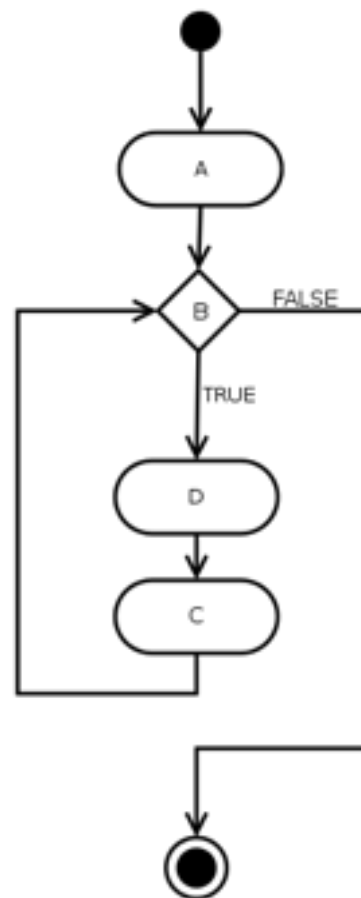


Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.



Algoritmos

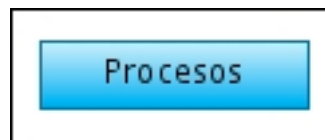
Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.



Inicio o fin del programa



Pasos, procesos o líneas de instrucción de programa de computo



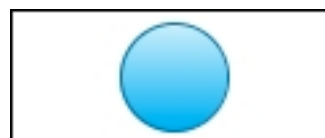
Operaciones de entrada y salida



Toma de decisiones y Ramificación



Envía datos a la impresora



Conector para unir el flujo a otra parte del diagrama.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Reglas para la creación de Diagramas

Los Diagramas de flujo deben escribirse de arriba hacia abajo, y/o de izquierda a derecha.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección que fluye la información procesos, se deben de utilizar solamente líneas de flujo horizontal o verticales (nunca diagonales)

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Se debe evitar el cruce de líneas, para lo cual se quisiera separar el flujo del diagrama a un sitio distinto, se pudiera realizar utilizando los conectores. Se debe tener en cuenta que solo se vana utilizar conectores cuando sea estrictamente necesario.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

No deben quedar líneas de flujo sin conectar

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Todo texto escrito dentro de un símbolo debe ser legible, preciso, evitando el uso de muchas palabras.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Todos los símbolos pueden tener más de una línea de entrada, a excepción del símbolo final.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Diagramas de Flujo.

Solo los símbolos de decisión pueden y deben tener mas de una línea de flujo de salida.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Hacer un diagrama de flujo que permita leer 2 números diferentes y nos diga cual es el mayor de los 2 números.

Algoritmos

Fundamentos de **Programación**

Como expresar algoritmos.

Pseudocódigo.

- Representación estructurada de un algoritmo
- Basada en convenciones notacionales.
- Parecido al código que finalmente se escribirá en el computador
- Está pensado para ser leído por humanos.

Pre y Poscondiciones de un Algoritmo

Precondición es la información que se conoce como verdadera antes de comenzar el algoritmo.

Postcondición es la información que se conoce como verdadera al concluir el algoritmo si se cumple adecuadamente el requerimiento pedido.

Algoritmos

Fundamentos de **Programación**

Tipos de algoritmos.

Algoritmos Cualitativos.

Son aquellos en los que se describen los pasos utilizando palabras, por ejemplo recetas de comidas, guía para armar un mueble, etc.

Algoritmos

Fundamentos de **Programación**

Tipos de algoritmos.

Algoritmos Cualitativos.

¿Como hacer una llamada telefónica ?. Condición: De un teléfono publico. El algoritmo finaliza cuando se realice la llamada.

Algoritmos

Fundamentos de **Programación**

Tipos de algoritmos.

Algoritmos Cuantitativos.

Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. por ejemplo calcular la raíz cuadrada de un número, determinar el factorial, etc.

Algoritmos

Fundamentos de **Programación**

Tipos de algoritmos.

Algoritmos Cuantitativos.

Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso. por ejemplo calcular la raíz cuadrada de un número, determinar el factorial, etc.

Algoritmos

Fundamentos de **Programación**

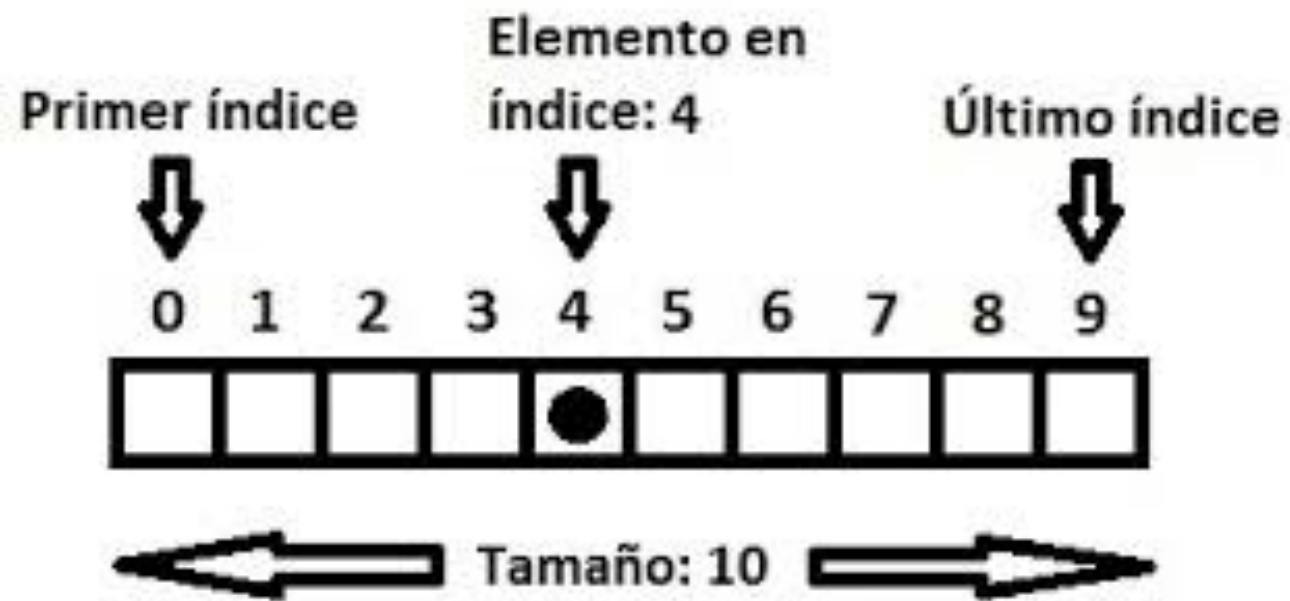
Arreglos

Un Arreglo es una colección en secuencia, de tamaño fijo de elementos de un mismo tipo.

Algoritmos

Fundamentos de **Programación**

Arreglos



Algoritmos

Fundamentos de **Programación**

Arreglos

| [0] | [1] | [2] | [3] | [4] |
|-----|-----|-----|-----|-----|
| 5 | 8 | 10 | 9 | 2 |

Nombre del Arreglo : Números

Largo del arreglo : 5

Ejemplo

Numeros [0] = 5

Numeros [3] = 9

Algoritmos

Fundamentos de **Programación**

Arreglos

| [0] | [1] | [2] | [3] | [4] |
|-----|-----|-----|-----|-----|
| 5 | 8 | 10 | 9 | 2 |

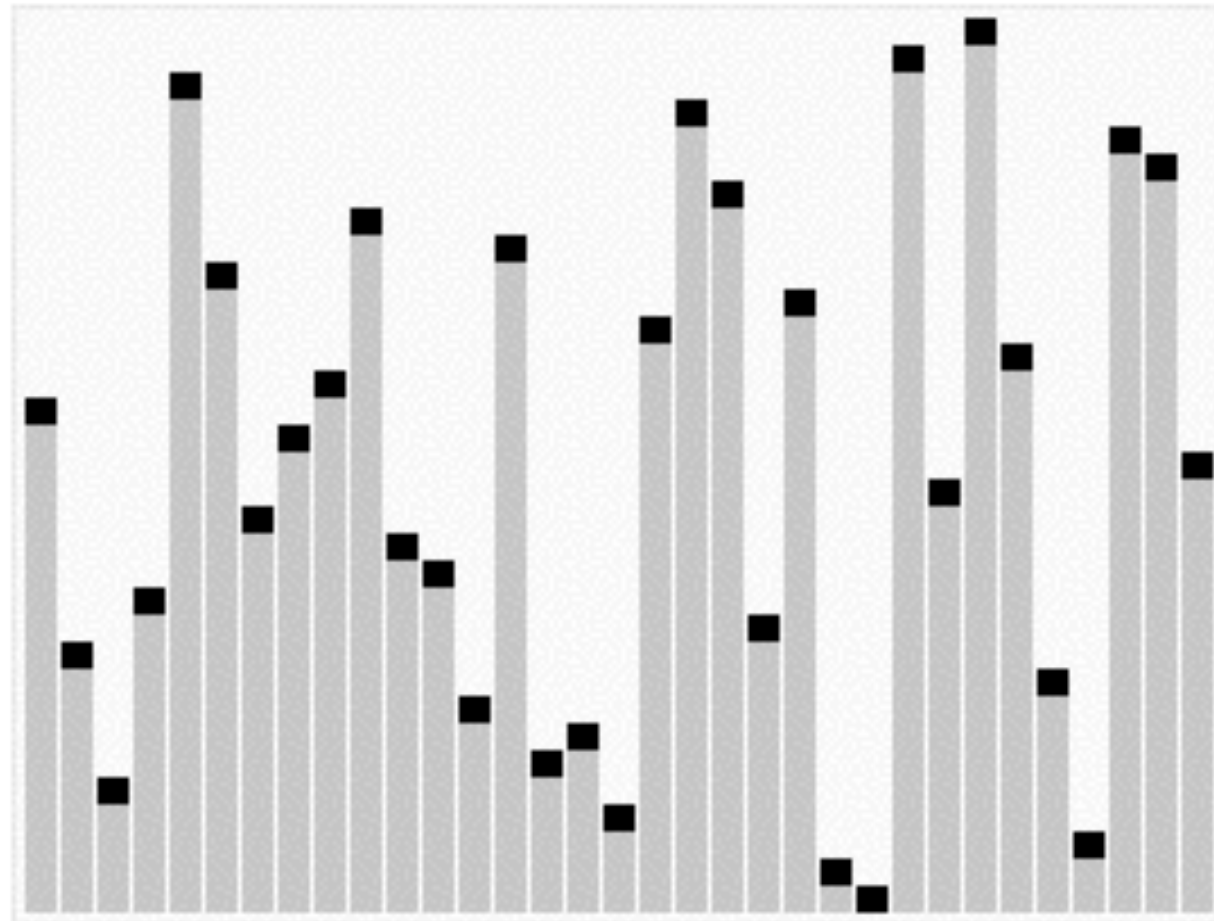
Escribir el algoritmo que permita imprimir cada uno de los elementos del arreglo.

Algoritmos

Fundamentos de **Programación**

Tipos de algoritmos.

Algoritmos de Ordenamiento.



Algoritmos

Fundamentos de **Programación**

Algoritmos de Ordenamiento.

Muchas actividades humanas requieren que en ellas las diferentes colecciones de elementos utilizados se coloquen en un orden específico.

Por esta circunstancia una de las tareas que realizan más frecuentemente las computadoras en el procesamiento de datos es la ordenación.

Algoritmos

Fundamentos de **Programación**

Algoritmos de Ordenamiento.

La ordenación o clasificación de datos (sort, en inglés) es una operación consistente en disponer un conjunto de datos en algún determinado orden con respecto a uno de los campos de elementos del conjunto.

Algoritmos

Fundamentos de **Programación**

Algoritmos de Ordenamiento.

En terminología de ordenación, el elemento por el cual está ordenado un conjunto de datos (o se está buscando) se denomina **clave**

El algoritmo de ordenación A será más eficiente que el B, si requiere menor número de comparaciones.

Algoritmos

Fundamentos de **Programación**

Algoritmos de Ordenamiento.

Los métodos de ordenación se suelen dividir en dos grandes grupos:

- **Ordenamiento Directos**
 - burbuja
 - selección
 - inserción
- **Ordenamiento Indirectos** (avanzados)
 - Shell,
 - ordenación rápida,
 - ordenación por mezcla, Radixsort

Algoritmos

Fundamentos de **Programación**

Ordenación por intercambio.

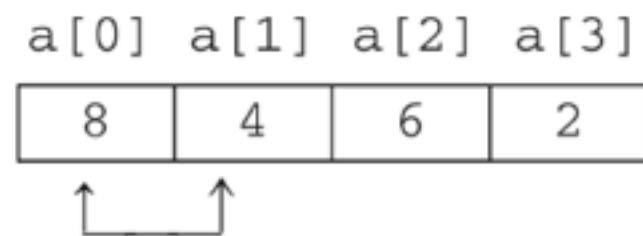
El algoritmo de ordenación por intercambio, ordena los elementos de una lista en orden ascendente.

Este algoritmo se basa en la lectura sucesiva de la lista a ordenar, comparando el elemento inferior de la lista con los restantes y efectuando intercambio de posiciones cuando el orden resultante de la comparación no sea el correcto.

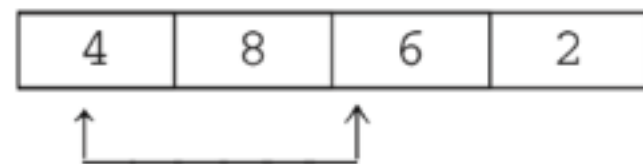
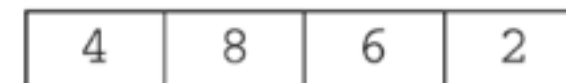
Algoritmos

Fundamentos de **Programación**

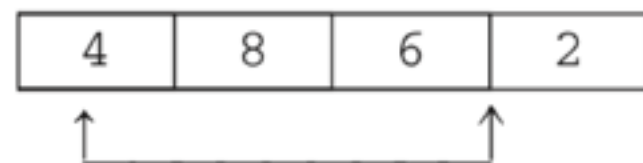
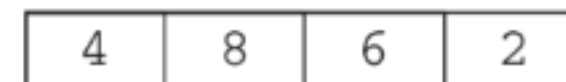
Ordenación por intercambio.



se realiza *intercambio*

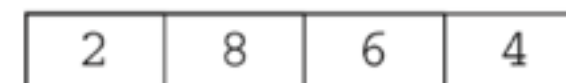


no se realiza *intercambio*



Lista inicial

se realiza *intercambio*



Lista resultante

Algoritmos

Fundamentos de **Programación**

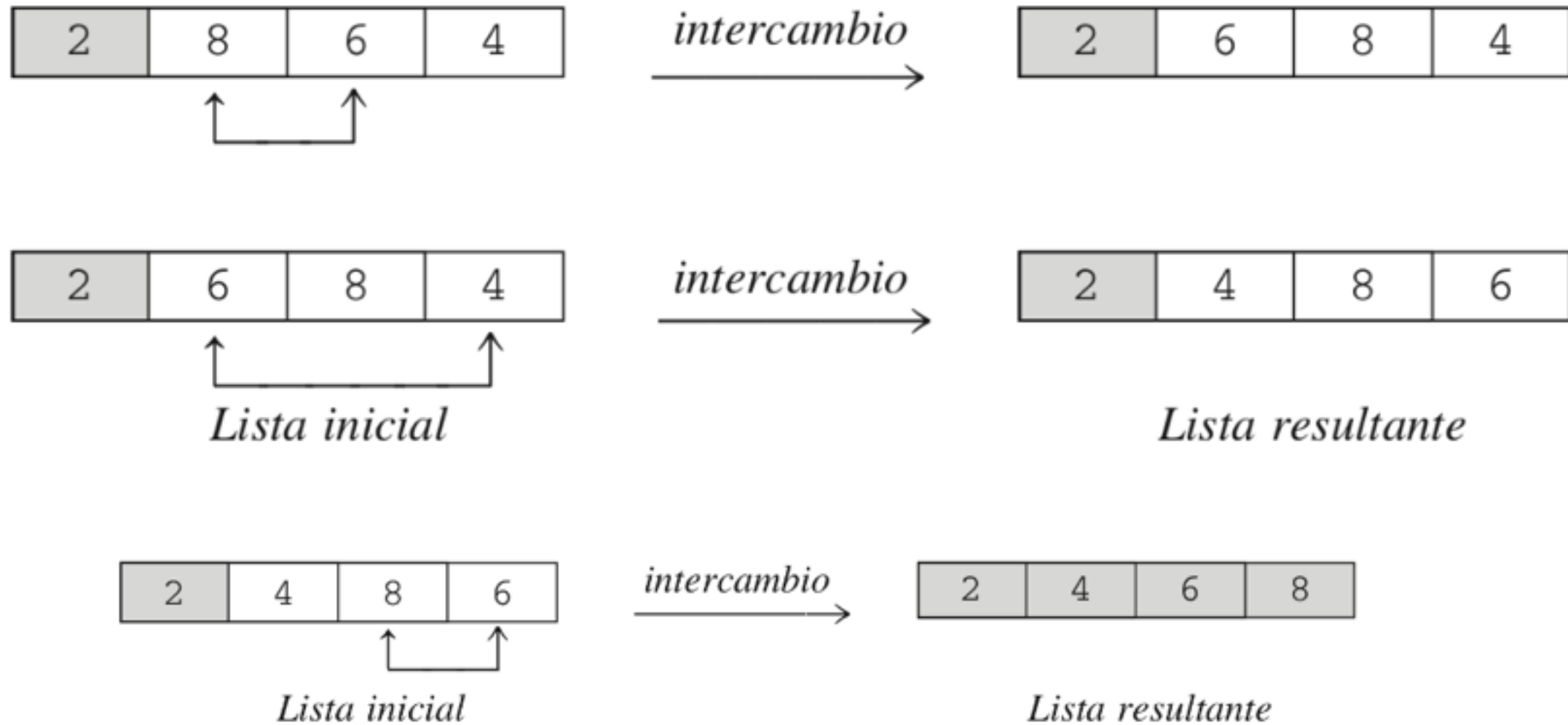
Ordenación por intercambio.

En cada comparación se comprueba si el elemento siguiente es más pequeño que el elemento de índice 0, en ese caso se intercambian.

Algoritmos

Fundamentos de **Programación**

Ordenación por intercambio.



Algoritmos

Fundamentos de **Programación**

Ordenación por selección

- Buscar el menor elemento dentro de una lista.
- Intercambiar el menor elemento a la primera posición de la lista.
- Continuar este proceso encontrando o seleccionando el elemento más pequeño de los restantes elementos de la lista, intercambiándolos adecuadamente.

Algoritmos

Fundamentos de **Programación**

Ordenación por selección

| | |
|--|---|
| | 8 |
| | 5 |
| | 2 |
| | 6 |
| | 9 |
| | 3 |
| | 1 |
| | 4 |
| | 0 |
| | 7 |