ITEDES

Educación Digital

Módulo:

Fundamentos de Ingeniería de Software

Segmento:

Algoritmos y Estructuras de Datos

Tema:

Bubble Sort

Prof. Germán C. Basisty

german.basisty@itedes.com

Índice de Contenidos

Bubble Sort 1
Índice de Contenidos 2
Bubble Sort 3
Ejemplo 3
Ejercitación 8

Bubble Sort

El **burbujeo** (**Bubble Sort** en inglés) es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran pequeñas "burbujas". También es conocido como el método del intercambio directo. Dado que solo usa comparaciones para operar elementos, se lo considera un algoritmo de comparación, siendo uno de los más sencillo de implementar.

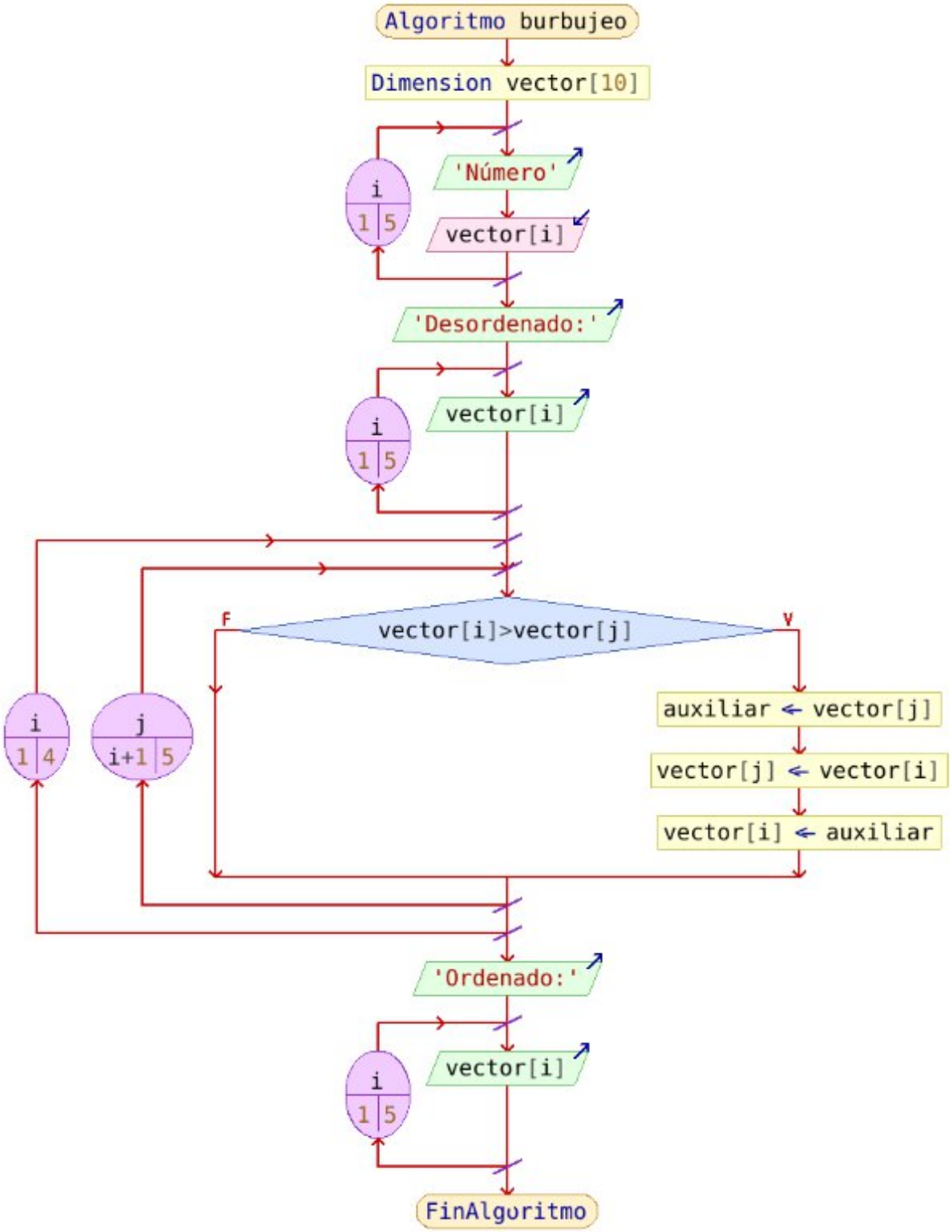
Ejemplo

Cargar un vector con 5 números, mostrarlo, ordenarlo y volverlo a mostrar.

Pseudocódigo

```
Algoritmo burbujeo
	Dimension vector(10)

	Para i <- 1 hasta 5
		Escribir "Número"
		Leer vector(i)
	FinPara

	Escribir "Desordenado:"
	Para i <- 1 hasta 5
		Escribir vector(i)
	FinPara

	Para i <- 1 hasta 4
		para j <- i + 1 hasta 5
			si vector(i) > vector(j)
				auxiliar = vector(j)
				vector(j) = vector(i)
				vector(i) = auxiliar
			FinSi
		FinPara
	FinPara

	Escribir "Ordenado:"
	Para i <- 1 hasta 5
		Escribir vector(i)
	FinPara
FinAlgoritmo
```

Diagrama de Flujo



Python

```python
from random import randrange

_SIZE = 5

vector = []

for i in range(_SIZE):
    vector.append(randrange(100))

for i in range(_SIZE):
    print(vector[i])

print()

for i in range(_SIZE - 1):
    for j in range(i + 1, _SIZE):
        if vector[i] > vector[j]:
            auxiliary = vector[j]
            vector[j] = vector[i]
            vector[i] = auxiliary

for i in range(_SIZE):
    print(vector[i])
```

Java

```java
import java.util.ArrayList;
import java.util.concurrent.ThreadLocalRandom;

public class Burbujeo {
    public static void main(String args[]) {
        ArrayList<Integer> vector = new ArrayList<Integer>();

        for(int i = 0; i < 5; i++)
            vector.add(ThreadLocalRandom.current().nextInt(0, 1000 +
1));

        for(int i = 0; i < vector.size() -1; i++) {
            for(int k = i; k < vector.size(); k++) {
                if(vector.get(i) > vector.get(k)) {
                    Integer auxiliar = vector.get(k);
                    vector.set(k, vector.get(i));
                    vector.set(i, auxiliar);
                }
            }
        }

        System.out.println(vector);
    }
```

```
}
```

C#

```
using System;
using System.Collections.Generic;

namespace burbujeo.net
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> numeros = new List<int>();

            Random rnd = new Random();

            for(int i = 0; i < 5; i++)
                numeros.Add(rnd.Next(0, 101));

            for(int i = 0; i < 5; i++)
                Console.WriteLine(numeros[i]);

            for(int i = 0; i < 4; i++)
            {
                for(int j = i + 1; j < 5; j++)
                {
                    if(numeros[i] > numeros[j])
                    {
                        int auxiliar = numeros[i];
                        numeros[i] = numeros[j];
                        numeros[j] = auxiliar;
                    }
                }
            }

            Console.WriteLine();

            for(int i = 0; i < 5; i++)
                Console.WriteLine(numeros[i]);
        }
    }
}
```

JavaScript

```
let numeros = [];

for(let i = 0; i < 5; i++)
    numeros.push(Math.floor(Math.random() * 100));

for(let i = 0; i < 5; i++)
    console.log(numeros[i]);

for(let i = 0; i < 4; i++) {
    for(let j = i + 1; j < 5; j++) {
        if(numeros[i] > numeros[j]) {
            let auxiliar = numeros[i];
            numeros[i] = numeros[j];
            numeros[j] = auxiliar;
        }
    }
}

console.log();

for(let i = 0; i < 5; i++)
    console.log(numeros[i]);
```

Este algoritmo realiza el ordenamiento o reordenamiento de una lista a de n valores, en este caso de n términos numerados del 0 al n-1; consta de dos bucles anidados, uno con el índice i, que da un tamaño menor al recorrido de la burbuja en sentido inverso de 2 a n, y un segundo bucle con el índice j, con un recorrido desde 0 hasta n-i, para cada iteración del primer bucle, que indica el lugar de la burbuja.

La burbuja son dos términos de la lista seguidos, j y j+1, que se comparan: ***si el primero es mayor que el segundo sus valores se intercambian***.

Esta comparación se repite en el centro de los dos bucles, dando lugar a la postre a una lista ordenada. Puede verse que el número de repeticiones solo depende de n y no del orden de los términos, esto es, si pasamos al algoritmo una lista ya ordenada, realizará todas las comparaciones exactamente igual que para una lista no ordenada. Esta es una característica de este algoritmo.

Ejercitación

1) Desarrollar un programa que cargue un vector con números aleatorios y ordenarlo de forma **ascendende** utilizando **bubble sort**. Comparar diferencias de rendimiento para vectores de 10, 100, 1000, y 10000 posiciones. Resolver en Java, Python, C# y JavaScript. Presentar diagrama de flujo y pseudocódigo.

2) Desarrollar un programa que cargue un vector con números aleatorios y ordenarlo de forma **descendende** utilizando **bubble sort**. Comparar diferencias de rendimiento para vectores de 10, 100, 1000, y 10000 posiciones. Resolver en Java, Python, C# y JavaScript. Presentar diagrama de flujo y pseudocódigo.