



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

CICLO II: Programación Básica en Java

Misión
TIC2022





El futuro digital
es de todos

MinTIC



Vigilada Mineducación

Sesión 2: Introducción a Java

ARREGLOS UNIDIMENSIONALES
VECTORES





Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Declarar e inicializar arreglos unidimensionales (vectores).
2. Utilizar los arreglos unidimensionales para almacenar datos y recuperarlos
3. Iterar a través de los arreglos unidimensionales y manipular sus datos utilizando las primitivas de programación estudiadas



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

ARREGLOS UNIDIMENSIONALES VECTORES

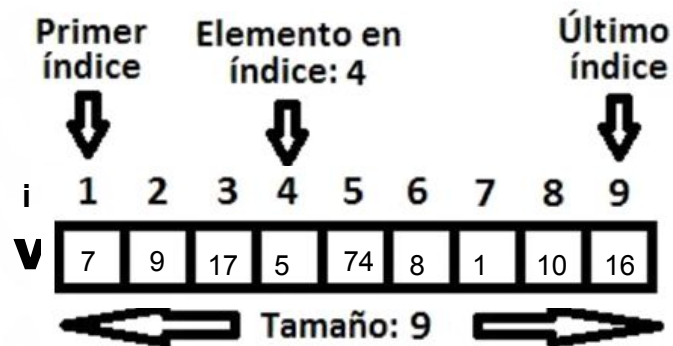




Arreglo Unidimensional: Vector

Definición - Repaso

Es un arreglo de N elementos organizados en una dimensión donde N recibe el nombre de longitud o tamaño del vector. Para hacer referencia a un elemento del vector se usa el nombre del mismo, seguido del índice (entre corchetes), el cual indica una posición en particular del vector.



$$V[4] = 5$$



Lectura y Escritura de un Vector

Pseudocódigo

Inicio

Entero i, n, V[100]

Leer n

para i: 0, n-1, 1 hacer

Leer V[i]

Finpara

para i: 0, n-1, 1 hacer

Escribir V[i]

Finpara

Fin

Java

```
public static void main(String[] args) {  
    int n, i, V[] = new int[100];  
    Scanner leer = new Scanner(System.in);  
    System.out.print("Cantidad estudiantes: ");  
    n = leer.nextInt();  
    for (i = 0; i < n; i++) {  
        System.out.print("Elemento "+(i+1)+" ": );  
        V[i] = leer.nextInt();  
    }  
    System.out.println("Elementos del vector");  
    for (i = 0; i < n; i++) {  
        System.out.println(V[i]);  
    }  
}
```



Arreglo Unidimensional: Vector

Algoritmos de Inserción – Codificar en Java

En Vector ordenado

Inicio

Entero V[cap_max], x, i, n, pos

pos ← 1

Leer x

Si (n < cap_max) entonces

MQ ($x > V[pos]$) hacer

pos ← pos + 1

Fin MQ

Para i ← n a pos paso -1 hacer

V[i + 1] ← V[i]

Fin para;

V[pos] ← x

Sino

Escribir "No hay espacio disponible"

Fin si

Fin

En Vector desordenado en la posición k

Inicio

Entero V[cap_max], x, i, n, k

Leer x, k

Si (n < cap_max) entonces

Para i ← n a k paso -1 hacer

V[i + 1] ← V[i]

Fin para

V[k] ← x

Sino

Escribir "No hay espacio disponible"

Fin si

Fin



Vector búsqueda secuencial

Pseudocódigo

```
Inicio
  Entero Pos  $\leftarrow$  1, X
  lógico existe  $\leftarrow$  F
  leer X
  MQ (Pos  $\leq$  N y Existe = F) hacer
    si (V(Pos) = X ) entonces
      existe  $\leftarrow$  V
    sino
      Pos  $\leftarrow$  Pos + 1
  fin si
fin MQ
Si (existe = V) entonces
  Escribir "Encontrado"
sino
  Escribir "No encontrado"
fin si
Fin
```

Java

```
public static void main(String[] args) {
    int n, i, V[] = new int[100], pos = 1, x;
    boolean sw = false;
    Scanner leer = new Scanner(System.in);
    System.out.print("Cantidad estudiantes: ");
    n = leer.nextInt();
    for (i = 0; i < n; i++) {
        System.out.print("Elemento " + (i + 1) + ": ");
        V[i] = leer.nextInt();
    }
    System.out.print("Elemento a buscar:");
    x = leer.nextInt();
    while (pos <= n && sw == false) {
        if (V[pos] == x) {
            sw = true;
        } else {
            pos = pos + 1;
        }
    }
    if (sw == true) {
        System.out.println("Encontrado");
    } else {
        System.out.println("No encontrado");
    }
}
```




Arreglo Unidimensional: Vector

Algoritmos de Eliminación – Codificar en Java

En Vector ordenado

Inicio

Entero V[cap_max], X, I, N, pos

pos ← 1

Leer X

MQ (X > V [pos]) hacer

pos ← pos + 1

Fin MQ

Para I ← pos hasta N-1 paso 1 hacer

V [I] ← V [I+1]

Fin para

N ← N-1

Fin

En Vector desordenado en la posición k

Inicio

Entero V [cap_max], I, N, K

Leer K

Para I ← K hasta N-1 paso 1 hacer

V [I] ← V [I+1]

Fin para

N ← N-1

Fin



Ordenamiento por inserción :

METODO INSERCION

```
public static void Insercion (int[] vector) {  
    for (int i = 1; i < vector.length; i++) {  
        int aux = vector[ i ];  
        int j;  
        for (j=i-1; j >=0 && vector[j] > aux; j--){  
            vector[ j+1] = vector[ j ];  
        }  
        vector[ j+1] = aux;  
    }  
}
```



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

Ejercicios para practicar





Programa Java que guarda en un array 10 números enteros que se leen por teclado. A continuación, se recorre el array y calcula cuántos números son positivos, cuantos negativos y cuántos ceros

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int[] numeros = new int[10];  
        int pos = 0, neg = 0, cero = 0; //contadores  
        int i;
```

```
        //Leemos los valores por teclado y los guardamos en el array
```

```
        System.out.println("Lectura de los elementos del array: ");
```

```
        for (i = 0; i < numeros.length; i++) {  
            System.out.print("numeros[" + i + "] = ");  
            numeros[i] = sc.nextInt();  
        }
```

```
        //se recorre el array para contar positivos, negativos y ceros
```

```
        for (i = 0; i < numeros.length; i++) {  
            if (numeros[i] > 0) {  
                pos++;  
            } else if (numeros[i] < 0) {  
                neg++;  
            } else {  
                cero++;  
            }  
        }
```

```
        //mostrar resultados
```

```
        System.out.println("Positivos: " + pos);  
        System.out.println("Negativos: " + neg);  
        System.out.println("Ceros: " + cero);
```

```
    }
```



Programa Java para leer la altura de N personas y calcular la altura media. Calcular cuántas personas tienen una altura superior a la media y cuántas tienen una altura inferior a la media. El valor de N se pide por teclado y debe ser entero positivo.

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int i, N;
```

```
        int contMas = 0, contMenos = 0;
```

```
        double media = 0;
```

```
        //Leer el número de personas
```

```
        do{
```

```
            System.out.print("Número de personas: ");
```

```
            N = sc.nextInt();
```

```
        }while(N<=0);
```

```
        //Se crea el array de tamaño N
```

```
        double[] alto = new double[N];
```

```
        //Leer alturas
```

```
        System.out.println("Lectura de la altura de las personas: ");
```

```
        for (i = 0; i < N; i++) {
```

```
            System.out.print("persona " + (i+1) + " = ");
```

```
            alto[i] = sc.nextDouble();
```

```
            media = media + alto[i]; //se suma la estatura leída para calcular la media
```

```
        }
```

```
        //Calcular la media
```

```
        media = media / N;
```

```
        //recorremos el array para ver cuantos hay más altos
```

```
        //que la media y cuantos más bajos
```

```
        for (i = 0; i < alto.length; i++) {
```

```
            if (alto[i] > media){ //si la estatura es mayor que la media
```

```
                contMas++;
```

```
            } else if (alto[i] < media){ //si es menor
```

```
                contMenos++;
```

```
            }
```

```
        }
```

```
        //Mostrar resultados
```

```
        System.out.println("Estatura media: " + media);
```

```
        System.out.println("Personas con estatura superior a la media: " + contMas);
```

```
        System.out.println("Personas con estatura inferior a la media: " + contMenos);
```

```
    }
```



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Misión
TIC 2022