

Empecemos...



Empecemos...



Empecemos...

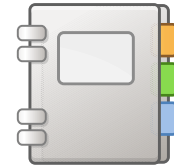


Empecemos...



¿Qué otras cosas identificamos así?

Vectores



Introducción a la Programación

Indice

- Vectores
Operaciones de vectores

Vectores

- Vectores: Introducción

Con lo aprendido hasta ahora resolvamos los siguientes problemas:

- Dados 500 números enteros, mostrarlos ordenados de menor a mayor.
- Dados 1000 nombres, mostrar cuantas veces se repite cada nombre.

Vectores

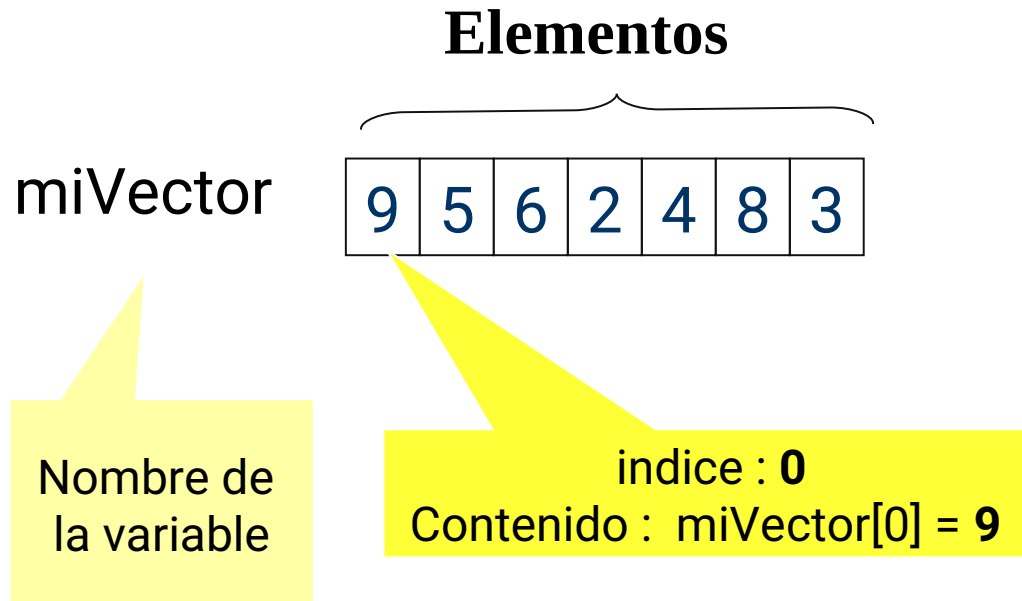
- Es posible resolver estos problemas? Por qué?
- Una de las principales dificultades que se observan con estos problemas es que para la resolución de ambos es necesario almacenar la totalidad de los datos a procesar.
- No sería eficiente crear n cantidad de variables para guardar estos datos.
- Para realizar programas y/o algoritmos que nos permitan resolver los problemas planteados, usaremos una nueva estructura de datos denominada **vectores**.

Vectores

- Es un arreglo unidimensional.
- Es una estructura de datos en la cual se almacena un conjunto de datos de **un mismo tipo**. Es decir que un arreglo es una lista de n elementos que posee las siguientes características:
 - ♦ Se identifica por un único nombre de variable
 - ♦ Sus elementos se almacenan en posiciones contiguas de memoria
 - ♦ Se accede a cada uno de sus elementos en forma aleatoria (por medio de un **índice**)

T1	Aprestamiento.....	2	..
T2	Números del 0 al 10.....	16	≡
T3	Suma (1).....	50	..
T4	Resta (1).....	60	⊗
T5	Suma y resta (1).....	72	∴
T6	Números hasta 100	78	≡
T7	Suma (2).....	92	≡
T8	Resta (2)	102	∴
T9	Suma y resta (2).....	112	≡
T10	Numeración maya.....	126	÷
T11	Longitud	130	≡
T12	Geometría.....	136	≡
T13	Gráfica	144	∴
---	---	---	∴

Vectores



Vectores - Declaración

- Una estructura de datos es considerada un tipo de dato, por lo que cumplen con las mismas reglas de una variable.

- Sintaxis Declaración

tipo_dato[] nombre

- Sintaxis Inicialización

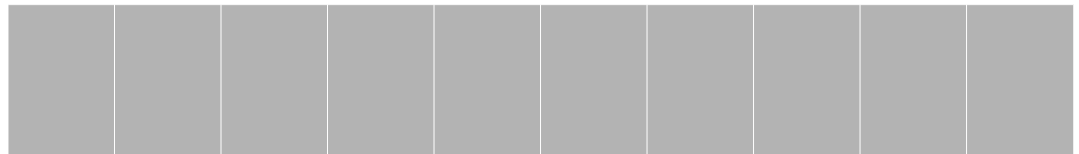
Nombre = new tipo_dato[tamaño]

Tamaño es la cantidad de elementos, es de tipo **entero**

- Ejemplo

int[] notas

notas = new int[10]



Vectores – Asignación de valor a elementos

- La representación de un elemento en un vector se hace mediante el *identificador* del vector seguido del índice entre corchetes:

identificador[indice]

- Asignación de valores

texto[3] = 'a'

precios[0] = 23.50

Notas[3] = 10

Notas[0] = 50

Notas[9] = 8

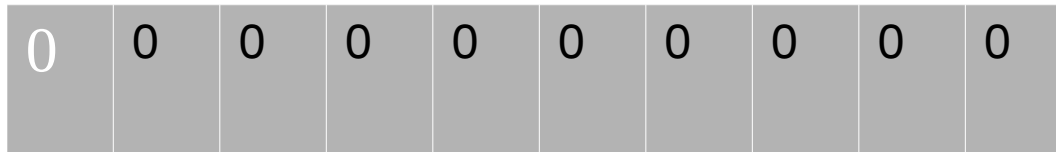
notas

50			10						8
----	--	--	----	--	--	--	--	--	---

Vectores - Operaciones

- Inicializar un vector de tipo entero con 0

```
int [] datos;    // declaracion
int i;
datos = new int[10] // inicializacion
for (i= 0; i < 10; i ++){
    datos[i] = 0;
}
```



0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Error:
Datos[10] = 0
ArrayIndexOutOfRangeException

Vectores - Operaciones

- Lectura de datos y guardarlos en un vector

```
int [] datos;    // declaracion
int i;
datos = new int[10] // inicializacion
for (i= 0; i < 10; i ++){
    read(datos[i]); //lectura
}
```

5	2	3	5	7	3	3	5	1	1
---	---	---	---	---	---	---	---	---	---

Vectores - Operaciones

- Mostrar datos de un vector

```
int [] datos;    // declaracion
int i;
datos = new int[10]; // inicializacion
for (i= 0; i < 10; i ++){
    write(datos[i]) //escritura
}
```


Ejemplos Vectores

- Cargar 10 elementos en un vector, luego sumar los números y mostrar los resultados a pantalla.

Pasos para resolver este problema:

- Leer un vector de 10 elementos
- Recorrer el vector y sumar sus elementos
- Mostrar el resultado de las sumas a pantalla

Program Sumas

```
{
Public static void main(string[] arg)
{
int[] numeros;
numeros = new int[10];
int suma =0 ;

for(int i=0; i<10; i++){
    write("Ingrese numero");
    read(numeros[i]);
    suma = suma + numeros[i];
}

for(int j=0; j<9; j++){
    write(numeros[j]+" ");
}
write(numeros[9]);
write("="+suma);
} // end main
} //end sumas
```

$$1+1+5+3+3+2+1+1+1=18$$

Vectores - Ejemplo

Algoritmo ProgramaVector

```
{  
    int[ ] sumandos;  
    int suma, i;  
    Public static void main(){  
  
        sumandos = new int[10];  
        Suma = 0;  
        for (i = 0; i <10; i++){  
            read(sumandos[i]);  
        }  
        for (i = 0; i <10; i++){  
            Suma = suma + sumandos[i];  
        }  
        write ('La suma de los números es'+ suma);  
    }  
}.
```

Declaración del tipo del arreglo

inicialización

Lectura de los elementos del arreglo

Suma de los elementos

Vectores - Ejemplos

- Dados 50 números enteros, obtener el promedio de ellos. Mostrar por pantalla dicho promedio y los números ingresados que sean mayores que el mismo.

```
Program Promedio
{
    int max = 50;
    Public static void main(){
        int[] numeros;

        Numeros = new int[max];
        Int suma =0;
        Double promedio = 0;
        for (int i=0; i<max; i++){
            write("Ingrese numero");
            read(numeros[i]);
            Suma = suma + numeros[i];
        }
        Promedio = suma / max;
        write("Promedio es "+promedio);
        For (int j=0; j<max; j++){
            If (numeros[j] > promedio){
                write(numeros[j]);
            }
        }
        write("El numero en la posicion 3 es "+numeros[2]);
    }
}
```

4	3	3	7	5
---	---	---	---	---

Suma=22
Promedio = 4.4
7 5

Vectores - Ejemplos resueltos

ProgramaPromedioMayores

BEGIN

const int max = 50;

int[] numeros;

public static void main()
{

Int suma, i;

Double promedio;

numeros = new int[max];

Suma = 0;

for(i = 0; i<max; i++){
 read(numeros[i]);
 suma = suma +numeros[i];
}

promedio = suma/max;

write ('El promedio es '
 +Promedio');

for(i = 0; i<max; i++){
 if (numeros[i] > promedio)
 write ('El número'+
 numeros[i] + es mayor al
 promedio');
}
}
}

Ejercicio

- Ingresar 100 elementos en un vector, mostrar todos los pares primero y luego todos los impares.

Ejercicio

- Ingresar 200 caracteres en un vector y luego pedir al usuario que ingrese una letra y muestre cuantas veces está dicha letra en el vector.

Vectores - Vectores Paralelos

- Dos o más arreglos que utilizan el mismo subíndice para acceder a elementos de distintos arreglos, se denominan arreglos paralelos. Estos arreglos pueden procesarse simultáneamente.

- Ejemplo:

Se tienen dos arreglos. El primero contiene nombres de personas y el segundo contiene la zona a la fecha sobre 65 puntos. Obtener a partir de estos un tercer arreglo que contenga lo que le hace falta para ganar la clase. *La zona máxima es de 65 puntos.*