licenciatura en ciencias de la computación

programación I

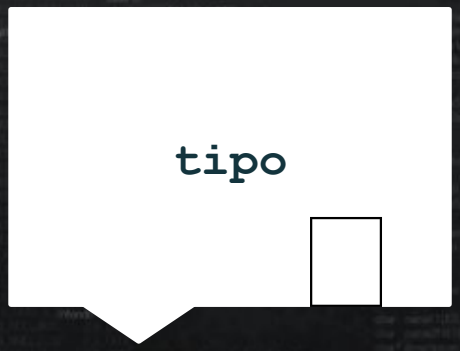
<clase> 6 = arreglos: vectores y matrices </clase>

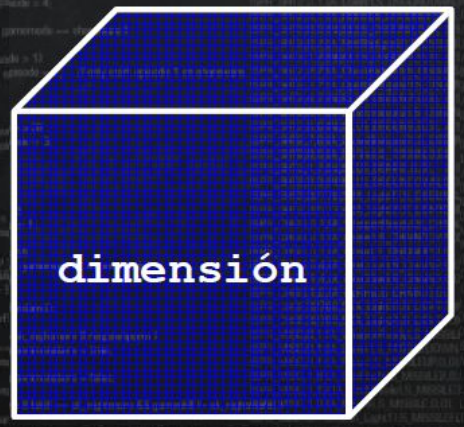
UNIVERSIDAD
NACIONAL DE CUYO

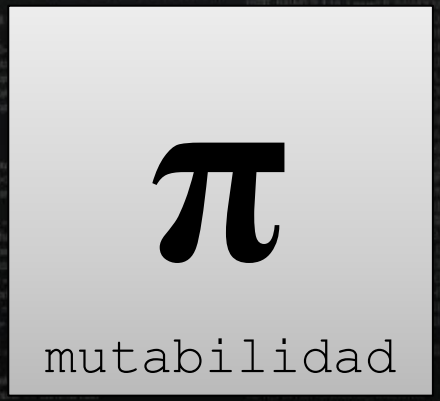
FACULTAD DE INGENIERIA
en acción continua...

Dra. Elina Pacini
Lic. Leandro Spadaro
Ing. Silvina Manganelli
Lic. Laura Noussan Lettry

datos: clasificación





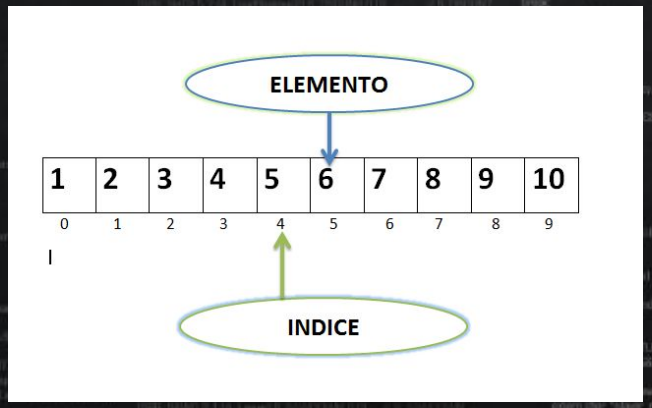


datos: dimensión y complejidad

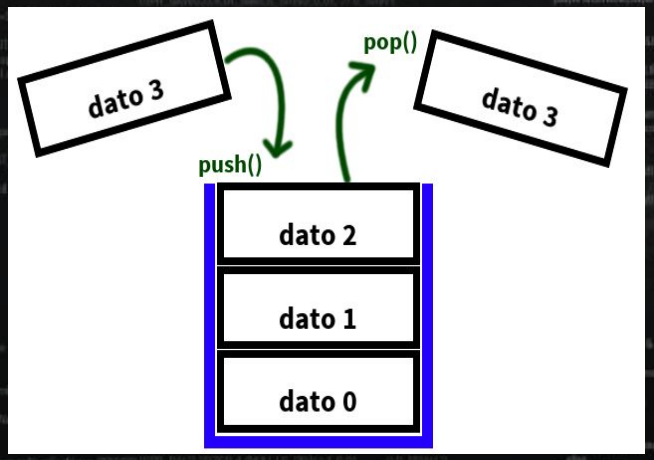
- ***escalares:*** valor único

30516

- ***multidimensionales:***
  - estáticos: arreglos (vectores, matrices)



  - dinámicos: árboles, listas, grafos, etc.



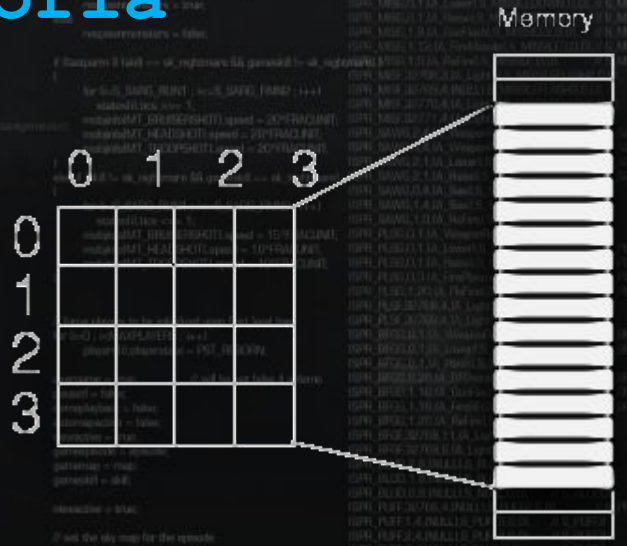
arreglos: características

a) definición estático: finitos

b) indexados

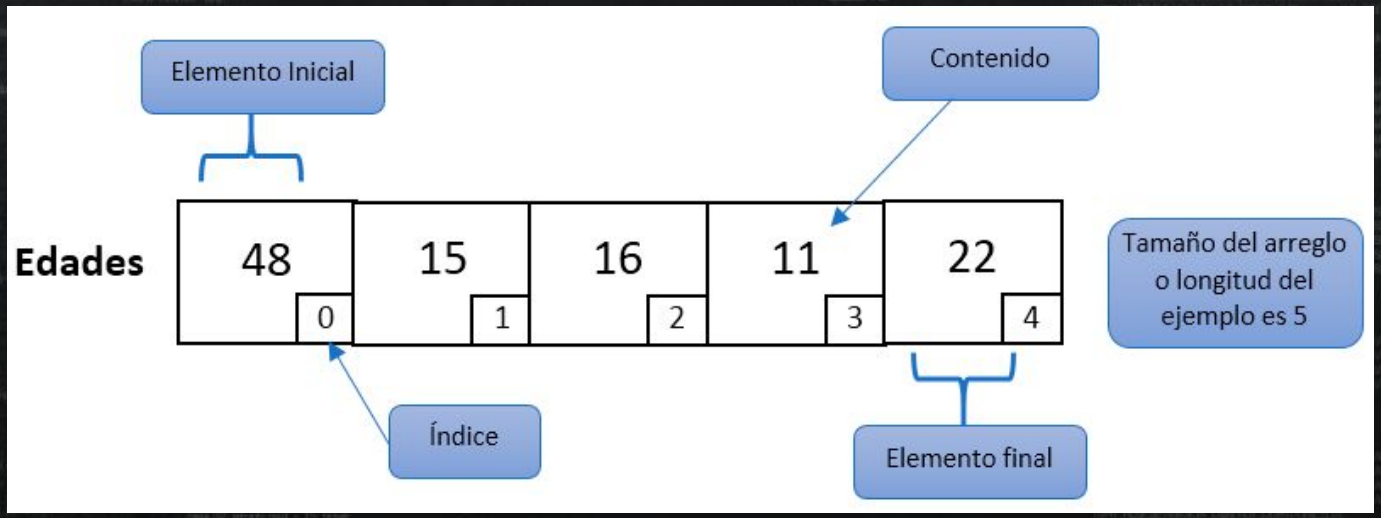
c) homogéneos

d) contiguos en memoria



vector

arreglo unidimensional



¿podría decirse que cada elemento de un vector se puede procesar como si fuese una variable simple?

(sub)índice

- **número entero** que **identifica un elemento** dentro del vector
- un vector de tamaño N posee N subíndices que se suceden de forma **creciente y monótona**
- el **valor inicial** del primer subíndice depende del lenguaje; la mayoría de los lenguajes modernos inician con el cero. Por ejemplo, si N = 4, en C, C++, Java, Ruby (y muchos más) los índices son 0, 1, 2, 3.

Temp[0]	Temp[1]	Temp[2]	Temp[3]	Temp[4]
14.0	**12.0**	**8.0**	**20.0**	**22.0**

declaración de un vector

declaración habitual

```
var
    tipo: nombre_variable[tamaño] {= valor_inicial}
```

ej.

```
var
    entero: diasSemana[7] = 0
```

declaración en PSeInt

```
Definir    nombre_variable tipo
Dimension nombre_variable(tamaño)
```

ej.

```
Definir diasSemana como Entero
Dimension diasSemana(7)
```

asignación/acceso

nombre_variable(expresión_subíndice)

```
...
Definir dias Como Real
Dimension dias(7)
Definir tempAux Como Real
Definir sabado como Entero

dias(0) = 17            //asignación con índice constante
sabado = 5
leer dias(sabado)
tempAux = dias(sabado) //acceso con índice variable
...
```

ej.

17	14	3	12	9	28	30

```
A) días(0) = "lunes"
B) tempAux = dias
C) tempAux = días(0)
D) tempAux = días(sabado - 1)
E) tempAux = días(sabado + 5)
F) tempAux = días(días(sábado))
```

¿ ?

recorrer

```
//ej: lee 5 enteros y los escribe por pantalla
//en orden invertido

Algoritmo leo_e_invierto
    Definir i, valores como entero
    Dimension valores(5)
    Para i<-0 Hasta 4 Con Paso 1 Hacer
    leer valores(i)
    Fin Para
    i = i - 1
    Mientras i >= 0 Hacer
    escribir valores(i)
    i = i - 1
    Fin Mientras
FinAlgoritmo
```

¿Se puede usar cualquier tipo de estructura repetitiva para recorrer un vector?

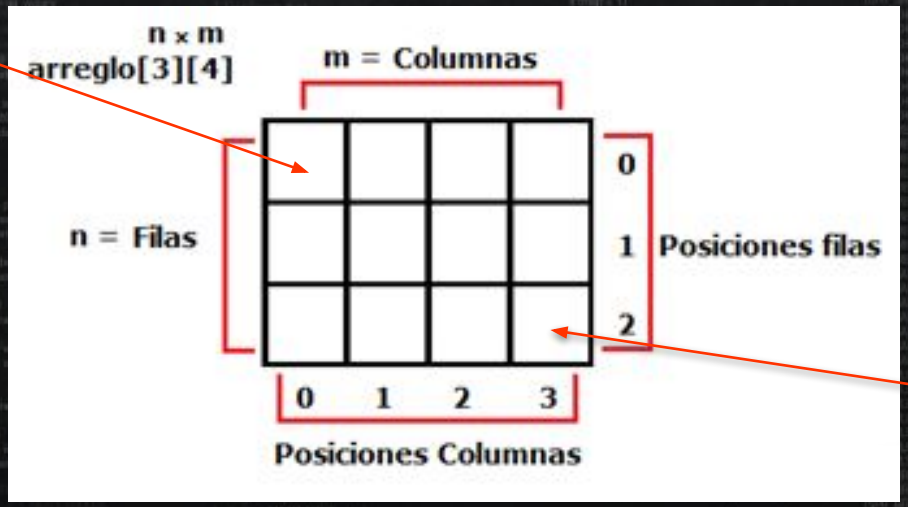
matriz

arreglo bidimensional

primer elemento subíndice [0,0]

ej. matriz A de 3x4

las filas se identifican con la primer posición del par [*i*,j]



tamaño del arreglo: 12

último elemento subíndice [2,3]

las columnas se identifican con la segunda posición del par [i, *j*]

¿que diferencia la matriz del vector?

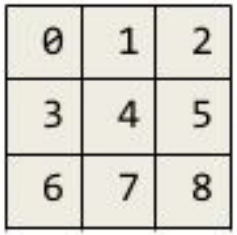
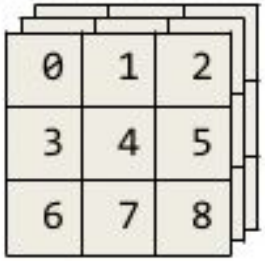
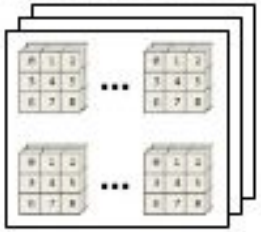
dimensión

- **cantidad de subíndices** necesarios para poder identificar un elemento
- cada dimensión tiene una **cardinalidad**, es decir una cantidad de valores subíndices posibles
- la **cantidad de elementos** de un arreglo (tamaño) es igual al producto de la cardinalidad de cada dimensión
- ej. dada la siguiente representación de una matriz

54	99
83	0
3	1

identifiquemos a) cantidad de dimensiones, b) cardinalidad de las dimensiones y c) el tamaño del arreglo

dimensiones: terminología

dimensiones	representación	terminología	índices
1		vector	[i]
2		matriz	[i,j]
3		arreglo tridimensional	[i,j,k]
N		arreglo ndimensional	[i,j,k...n]

declaración de una matriz

declaración habitual

```
var
   tipo: nombre_variable[d1tam{,dntam}] {= valor_inicial}
```

ej.

```
var
     entero: ubicaciones[3,30,20]
```

declaración en PSeInt

```
Definir    nombre_variable tipo
Dimension nombre_variable(d1tam,...,dntam)
```

ej.

```
Dimension notasAlumnos(30,5)
Definir notasAlumnos como Real
```

asignación/acceso

nombre_variable[expr_sub1{,expr_subN}]

```
...
Definir MAXFILAS, MAXASIENTOS, LIBRE, RESERVADO, OCUPADO como Entero
Definir ubicaciones como Entero
Definir fila, asiento, estado como Entero
MAXFILAS = 20, MAXASIENTOS=30
Dimension ubicaciones[MAXFILAS,MAXASIENTOS]
LIBRE = 0, RESERVADO = 1, OCUPADO = 2
...
    inicializarLibre(ubicaciones)
    ubicaciones[0,0] = RESERVADO//para el dueño del teatro
    escribir "indique fila y asiento"
    leer fila, asiento      //asignación mediante una función
    si fila >= 0 y fila <= MAXFILAS-1 y asiento >=0 y asiento <= MAXASIENTOS-1
        si ubicaciones(fila, asiento) = LIBRE entonces
            ubicaciones[fila, asiento] = OCUPADO
...
```

A. `ubicaciones[0] = RESERVADO`
B. `ubicaciones[7,0] = RESERVADO`
C. `ubicaciones[MAXASIENTOS,MAXFILAS] = LIBRE`
D. `ubicaciones[6,9] = 5`
E. `estado = ubicaciones[MAXFILAS-1,MAXASIENTOS-1]`

¿ ?

recorrer

```
//ej: lee 5 enteros y los escribe por pantalla
//en orden invertido

Algoritmo leo_e_invierto
    definir i, j, matriz como Entero
    dimension matriz(3,3)
    Para i<-0 Hasta 2 Con Paso 1 Hacer
     Para j<-0 Hasta 2 Con Paso 1 Hacer
         leer matriz(i,j)
     Fin Para
    Fin Para
    Para i<-2 Hasta 0 Con Paso -1 Hacer
     Para j<-2 Hasta 0 Con Paso -1 Hacer
         escribir matriz(i,j)
     Fin Para
    Fin Para
FinAlgoritmo
```

¿existe alguna relación entre la dimensión y la cantidad de estructuras repetitivas anidadas?

Uso en subprogramas

```
Funcion mayor <- buscarMayor (notas)
    Definir mayor, i Como Entero
    mayor = notas(0)
    Para i<-1 Hasta 9 Con Paso 1 Hacer
        Si notas(i) > mayor entonces
            mayor = notas(i)
        FinSi
    Fin Para
FinFuncion


SubProceso duplicarVector(original, duplicado)
    Definir i Como Entero
    Para i<-0 Hasta 9 Con Paso 1 Hacer
        duplicado(i) = original(i)
    Fin Para
FinSubproceso
```

manejo de cadenas

Una cadena de caracteres es una secuencia ordenada (de longitud arbitraria, pero finita) de elementos que pertenecen a un cierto lenguaje formal o alfabeto (letras, números u otros símbolos). Los caracteres se guardan uno a continuación de otro para fines de eficiencia en el acceso.

Operaciones comunes con cadenas:

- asignación
- concatenación
- búsqueda dentro de una cadena
- extracción (caracter o subcadena)

funciones en PSeInt

- Longitud(cadena)
- SubCadena(cadena,desde,hasta)
- Concatenar(cadena,cadena)
- ConvertirANumero(cadena)
- ConvertirATexto(numero)
- Mayusculas(cadena)
- Minusculas(cadena)

cadenas en PSeInt

```
Algoritmo separarPalabras
    Definir frase, palabra Como Cadena
    Definir i como Entero

    palabra=""
    frase = "hola mundo cruel"
    Escribir "La longitud de la frase es: " Longitud(frase)

    para i=0 hasta longitud(frase) hacer
    si(subcadena(frase,i,i) != " ") entonces
        palabra=concatenar(palabra, subcadena(frase,i,i))
    SiNo
        Escribir "la palabra es: " palabra
            palabra=""
    FinSi
    FinPara

    Escribir "la palabra es: " palabra
FinAlgoritmo
```

buenas prácticas

- controlar el límite de los subíndices durante el recorrido de un arreglo para evitar errores (ej. desbordamientos)
- definir tamaños acordes al problema resuelto para hacer un uso racional de la memoria
- investigar en los lenguajes reales
  - política de inicio de subíndices
  - tipo de paso por parámetros
  - funciones de búsqueda/ordenamiento

¿Preguntas?