

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.7 – Arrays unidimensionales (Vectores)

- Un **array** es un conjunto finito y ordenado de elementos homogéneos
 - Cada elemento puede ser identificado
 - Son del mismo tipo de datos
- El **array unidimensional o vector** es el tipo de array más simple.

notas[0] notas[1] notas[2] notas[3] notas[n]

Subíndice designa la posición del elemento en el vector

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.7 – Arrays unidimensionales (Vectores)

- Las **operaciones** que se pueden realizar con vectores son: asignación, lectura, escritura, recorrido, actualización (añadir, borrar, insertar), ordenación, búsqueda.
- La notación algorítmica que utilizaremos es la siguiente:


```
Array [dimensiones] de <tipo de dato> : <nombre del array>
```

```
Array[0..100] de entero : numero
```
- Asignación:


```
<nombre del array> [subíndice] <- <valor>
```

```
numero[0] <- 5
```

```
desde i = 1 hasta 5 hacer
```

```
  numero[i] <- 8
```

```
fin_desde
```

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.7 – Arrays unidimensionales (Vectores)

- Lectura:


```
leer (<nombre del array>[subíndice])
```

```
leer (numero[0])
```
- Escritura:


```
escribir (<nombre del array>[subíndice])
```

```
escribir ("el número es: "numero[0]);
```

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.7 – Arrays unidimensionales (Vectores)

- Insertión:


```
/* Se desea insertar nuevo_elemento en la posición p */
```

```
/* i corresponde al índice del último elemento*/
```

```
mientras (i >= p) hacer
```

```
  autos[i+1]<-autos[i]
```

```
  i <- i-1
```

```
fin_mientras
```

```
autos[p]<- nuevo_elemento
```

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.7 – Arrays unidimensionales (Vectores)

- eliminar:


```
/* Se desea eliminar elemento en la posición j */
```

```
/* n corresponde al índice del último elemento*/
```

```
desde (i <- j) hasta n-1 hacer
```

```
  autos[i]<-autos[i+1]
```

```
fin_desde
```

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.8 – Arrays bidimensionales (Matrices)

- Un **array bidimensional** se puede considerar como un vector de vectores. Es por consiguiente un conjunto de elementos homogéneos y ordenados en el que se necesita especificar dos subíndices para poder identificar cada elemento del array.

	notas[0,0]	notas[0,1]	notas[0,j]
Fila 1	notas[1,0]	notas[1,1]	notas[1,j]
Fila 2	notas[2,0]	notas[2,1]	notas[2,j]
.....
Fila i	notas[i,0]	notas[i,1]	notas[i,j]
	Columna 1	Columna 2	Columna j

Desarrollado por Ricardo Soto De Giorgis Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.8 – Arrays bidimensionales(Matrices)

- Las **operaciones** que se pueden realizar con matrices son: asignación, lectura, escritura, recorrido, actualización(añadir, borrar, insertar), ordenación, búsqueda.
- La notación algorítmica que utilizaremos es la siguiente:

```
Array[dimensión fila, dimensión columna] de <tipo de dato>:<nombre del array>
```


```
Array[0..100,0..50] de entero:numero
```

- Asignación:

```
<nombre del array> [subíndice fila,subíndice columna] <- <valor>
```

```
numero[0,0] <- 5
```

```
desde i = 1 hasta 5 hacer  
  desde j = 1 hasta 5 hacer  
    numero[i,j] <- 8  
  fin_desde  
fin_desde
```



Desarrollado por Ricardo Soto del Giorgio Escuela de Ingeniería Informática ICI 241 – Estructura de Datos

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO Escuela de Ingeniería Informática

Capítulo 1 - Algoritmos

1.8 – Arrays bidimensionales(Matrices)

- Lectura:

```
leer (<nombre del array>[subíndice fila, subíndice columna])
```

```
leer (numero[0,0])
```

- Escritura:

```
escribir (<nombre del array>[subíndice fila, subíndice columna])
```

```
escribir ("el número es: "numero[0,0]);
```



Desarrollado por Ricardo Soto del Giorgio Escuela de Ingeniería Informática ICI 241 – Estructura de Datos