

## 8. REGISTROS

Sean A y B dos conjuntos, el **producto cartesiano** se define como  $A \times B = \{ (x,y) \mid x \in A \wedge y \in B \}$

### Ejemplos

1. Si  $A = \{ 0, 1 \}$  y  $B = \{ a, b \}$  entonces  $A \times B = \{ (0,a), (0,b), (1,a), (1,b) \}$
2. Complejos =  $\mathbb{R} \times \mathbb{R} = \{ (x,y) \mid x \in \mathbb{R} \wedge y \in \mathbb{R} \}$
3.  $B = \text{Complejos} \times \mathbb{N} = \{ (x,y) \mid x \in \text{Complejos} \wedge y \in \mathbb{N} \}$
4. Racionales =  $\mathbb{N} \times \mathbb{N} = \{ (x,y) \mid x \in \mathbb{R} \wedge y \in \mathbb{R} \}$

Sea  $\{ A_i \}_{i=1,2,\dots,n}$  una familia de n conjuntos se define el **producto cartesiano generalizado** de la familia como  $A_1 \times A_2 \times \dots \times A_n = \{ (a_1, a_2, \dots, a_n) \mid a_1 \in A_1, a_2 \in A_2, \dots, a_n \in A_n \}$ .

### Ejemplos

1. Si  $A = \{ 0, 1 \}$ ,  $B = \{ a, b \}$  y  $C = \{ x, y, z \}$  entonces
 
$$A \times B \times C = \{ (0,a,x), (0,a,y), (0,a,z), (0,b,x), (0,b,y), (0,b,z), (1,a,x), (1,a,y), (1,a,z), (1,b,x), (1,b,y), (1,b,z) \}$$
2. Estudiante = Cadena  $\times \mathbb{Z} \times \mathbb{R}^5 = \{ (\text{nombre}, \text{codigo}, \text{notas}) \mid \text{nombre} \in \text{cadena}, \text{codigo} \in \mathbb{Z}, \text{notas} \in \mathbb{R}^5 \}$
3. Caja =  $\mathbb{R} \times \mathbb{R} \times \mathbb{R} = \{ (\text{alto}, \text{ancho}, \text{largo}) \mid \text{alto} \in \mathbb{R}, \text{ancho} \in \mathbb{R}, \text{largo} \in \mathbb{R} \}$

Cuando todos los conjuntos de la familia  $\{ A_i \}_{i=1,2,\dots,n}$  son iguales, es decir,  $A = A_1 = A_2 = \dots = A_n$  se dice que  $A_1 \times A_2 \times \dots \times A_n$  es el **espacio n-dimensional** del conjunto A, y se nota  $A^n$ .

En programación un **registro** es el concepto que permite representar

el producto cartesiano generalizado, excluyendo los espacios n-dimensionales que generalmente son representados mediante arreglos. En otras palabras, un registro es un tipo de dato creado por el programador, compuesto por otros tipos de datos, ya sean básicos, arreglos, matrices y/o registros previamente declarados. A cada componente de un registro se le conoce con el nombre de **campo**. Un registro se declara de la siguiente manera:

```

tipo< nombre> como
  < nombre1> : < tipo1>
  < nombre2> : < tipo2>
  .
  .
  .
  < nombren> : < tipon>
fin_tipo

```

donde,

< nombre>: es el nombre que se le pone al tipo que se está declarando.

< nombre<sub>i</sub>> : es el nombre del i-esimo campo del tipo.

< tipo<sub>i</sub>> : es el tipo del i-esimo campo del tipo.

### Ejemplos.

1. Estudiante = { (nombre, codigo, notas) | nombre ∈ cadena, codigo ∈ Z, notas ∈ R<sup>5</sup>}

```

tipo Estudiante como
  nombre : arreglo[100] de carácter
  codigo : entero
  notas : arreglo[5] de real
fin_tipo

```

2. Caja = R×R×R = { (alto, ancho, largo) | alto ∈ R, ancho ∈ R, largo ∈ R }

```
tipo Caja como  
  alto : real  
  ancho : real  
    largo : real  
fin_tipo
```

3. Complejo =  $\mathbb{R} \times \mathbb{R} = \{ (x, y) \mid x \in \mathbb{R}, y \in \mathbb{R} \}$

```
tipo Complejo como  
  x : real  
  y : real  
fin_tipo
```

Los registros se declaran al comienzo del programa, después de la declaración de constantes y de la palabra reservada **tipos** y antes de la declaración de cualquier función y/o procedimiento. El siguiente esquema muestra la forma general de un programa escrito en el pseudo lenguaje utilizado en este libro:

```
constantes  
  <declaración constantes>  
  
tipos  
  <declaración registros>  
  
  <declaración funciones y/o  
  procedimientos>  
  
procedimiento principal()  
variables  
  <declaración variables procedimiento  
  principal>  
inicio  
  <algoritmo principal>  
fin_procedimiento
```

Cuando se ha declarado un registro, es decir un nuevo tipo, se pueden declarar variables del mismo.

## Ejemplos.

1. **z1 : Complejo.** En este caso la variable **z1** tiene dos campos **x** e **y** de tipo real.
2. **x : Caja.** En este caso la variable **x** tiene tres campos de tipo real, **alto**, **ancho** y **largo**.
3. **e : Estudiante.** En este caso la variable **e** tiene tres campos, el primero es **nombre** el cual es de tipo cadena de caracteres de tamaño cien, el segundo es **codigo** de tipo entero y el último es **notas** que es un arreglo de tamaño cinco de reales.

Para obtener y/o modificar el valor de cualquier campo de un registro se hace de la siguiente manera:

<variable>.<campo>

donde,

<variable> : es el nombre de la variable de tipo registro.

<campo> : es el nombre del campo del registro, del cual se quiere obtener el valor.

## Ejemplos.

1. Si **x** es una variable de tipo **Caja**

```

x.alto := 10
           /* modifica el campo alto de x y lo
           deja en 10 */
x.ancho := (x.ancho / 2.0)
           /* divide el ancho de la caja x por dos
           */
escribir( x.largo )
           /* escribe en pantalla el largo de la
           caja x */
volumen := x.largo*x.alto*x.ancho
           /* calcula el volumen de la caja x*/
```

2. Si **z1** es una variable del tipo **Complejo**

```

norma := z1.x * z1.x + z1.y * z1.y
        /* calcula la norma de z1 */
leer (z1.x)
        /* lee la parte x del complejo z1 */
escribir (z1.y)
        /* imprime la parte y del complejo z1
*/

```

### 3. Si **e** es una variable del tipo **Estudiante**

```

e.notas[0] := dato * 0.15
        /* obtiene el 15% de dato y lo almacena
en la primer nota      del estudiante e */
        /* las siguientes 5 lineas calculan el
promedio de las notas de e */
promedio := 0.0
para (i := 0 hasta 4) hacer
        promedio := promedio + e.notas[i]
fin_para
promedio := promedio / 5.0

```

Como cualquier otra variable, una variable de tipo registro puede ser pasada como argumento a una función y/o procedimiento por valor o por referencia.

**Ejemplo.** Desarrollar un programa que calcule la nota definitiva más alta de un conjunto de 40 estudiantes, cada uno de los cuales tiene nombre, código y cinco notas parciales.

**Algoritmo:**

—

## constantes

```
/* el numero de caracteres maximo en
el nombre */
TAMANO_NOMBRE = 100
/* el numero de estudiantes del curso */
N = 40
/* el numero de notas parciales de cada
estudiante */
NUM_NOTAS = 5
```

## tipos

```
/* El estudiante es un registro con
nombre, codigo y cinco notas */
```

### tipo Estudiante como

**nombre** :

**arreglo**[TAMANO\_NOMBRE] **de**  
carácter

**codigo** : entero

**notas** : **arreglo**[NUM\_NOTAS] **de**  
real

### fin\_tipo

```
/* leer los datos de un estudiante, la
variable e es pasada
por referencia, lo que implica que la
variable enviada como argumento sera
modificada */
```

**procedimiento** LeerEstudiante( **ref** e :  
Estudiante )

### variables

**i** : entero

### inicio

```
/* lee el nombre del estudiante */
escribir( "Digite el nombre del
estudiante:")
leer( e.nombre )
/* lee el codigo del estudiante */
escribir( "Digite el codigo del
estudiante:")
leer( e.codigo )
```

```

/* lee las cinco notas del estudiante */
para ( i := 0 hasta NUM_NOTAS-1)
hacer
    escribir( "Digite la nota ")
    escribir(i)
    escribir("-esima del estudiante "),
    escribir(e.nombre)
    leer( e.notas[i] )
fin_para
fin_procedimiento

/* calcula la nota definitiva de un
estudiante, la variable e
es pasada por valor lo que implica que la
variable enviada como argumento no
sera modificada en esta función */

funcion definitiva( e : Estudiante ) : real
variables
    i : entero
    promedio : real
inicio
    /* suma las notas parciales */
    promedio := 0.0
    para ( i := 0 hasta NUM_NOTAS-1)
hacer
        promedio := promedio + e.notas[i]
    fin_para
    /* divide la suma de notas parciales por el
numero de notas*/
    promedio := promedio / NUM_NOTAS
    /* retorna la nota definitiva del estudiante
*/
    retornar promedio
fin_funcion

/* calcula el maximo de una colección de
N reales */

funcion maximo( A : arreglo[N] de real )
: real
variables

```

```

max : real
i : entero
inicio
    max := A[0]
    para (i := 1 hasta N-1) hacer
        si (max < A[i]) entonces
            max := A[i]
        fin_si
    fin_para
    retornar max
fin_funcion

/* algoritmo principal */
procedimiento principal()
variables
    curso : arreglo[N] de Estudiante
    definitivas : arreglo[N] de real
    i : entero
    max : real
inicio
    para i := 0 hasta N-1 hacer
        escribir("Datos del estudiante ", i, "-
esimo")
        escribir (cambioLinea)
        /* lee los datos del i-esimo
estudiantes del curso */
        LeerEstudiante( curso[i] )
        /* calcula la definitiva del i-esimo
estudiante */
        definitivas[i] := Definitiva( curso[i] )
    fin_para
    /* calcula la nota maxima definitiva */
    max := maximo( definitivas )
    escribir( "La nota maxima definitiva
fue: ", max ) fin_procedimiento

```