

## Capítulo 9: Registros

Tipo estructurado que permite almacenar un conjunto de valores de distinto tipo. Cada una de las componentes del registro se denomina campo, puede ser de tipo simple o estructurado.

### Declaración

```
TipoReg = Record
  idcampo1: tipo1;
  idcampo2: tipo2;
  ....
End;
```

Ejemplo de un registro dónde se almacenan los datos de una fecha:

#### Type

```
TregFecha = record
  Mes, Dia: byte;
  Anio: word;
end,
```

30	8	2004
Dia	Mes	Anio

Ejemplo de un registro dónde se almacenan los datos de una persona:

#### Type

```
St20 = string[20];
TregP = Record
  Nombre: St20;
  Edad: byte;
  Peso, Altura: Real;
End;
Var
  Per1, Per2: TregP;
```

'Pía'	20	50	1.6
Nombre	Edad	Peso	Altura

### Acceso

Para acceder a la información almacenada en un registro: *VariableTipoRegistro.Campo*

Como es un tipo estructurado, las operaciones se realizan sobre los campos y no sobre la totalidad de la variable →

```
Readln(Per1);
With Per1 Do
  Begin
    Readln(Nombre); Readln(Altura);
  End;
}
Readln(Per1.Nombre);
Readln(Per1.Altura);
-----
Per2.Peso := 50;
Per1.Peso := Per2.Peso;
Per2 := Per1; {se permite asignación entre registros si son del mismo tipo}
Writeln(Per2.Altura);
```

No se permite la comparación entre registros, solamente entre campos.

```
If Per1.Altura < Per2.Altura then
  Writeln(Per1.Nombre, ' es mas bajo que ', Per2.Nombre);
```

### Campos de tipo estructurado: registros y arreglos

- **Campo registro** Ejemplo: Fecha de nacimiento. *Escribir el año de nacimiento*

```
TregPer = Record
```

```

    Nombre: St20;
    FechNac: TregFecha; {registros anidados}
End;
Var
    Per: TregPer;
.....
Write ('Nacio en el año', Per.FechNac.Año);

```

- **Campo arreglo** Ejemplo: 5 mediciones de peso (5 meses). *Leer la 3ra medición*

```

TregPers = Record
    Nombre: St20;
    Pesos: array[1..5] of real; {el tipo arreglo podría definirse previamente}
End;
Var
    Pers: TregPers;
.....
readln (Pers.Pesos[3]);

```

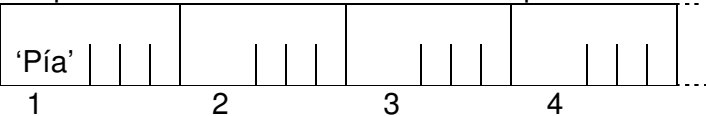
### **Registros como parte de otra estructura**

Si se desea almacenar los datos de un conjunto de personas es posible armar un arreglo de registros, donde en cada uno de ellos se puede almacenar los datos de una persona.

```

TV = array[1..MAXELEM] of TregP;

```



⇒ Lectura de un arreglo, no se conoce la cantidad de elementos:

```

Procedure LeeVector(Var V: TV; Var N: Byte);
Var
    Res: char;
Begin
    N := 0;
    Write('Ingresa datos (S/N)'); Readln(Res);
    While (RES = 'S') and (N < MaxElem) do
        Begin
            N := N+1;
            With V[N] do
                Begin
                    Readln(Nombre);
                    Readln(Edad);
                    Readln(Peso);
                    Readln(Altura);
                End;
            Write('Ingresa datos (S/N)'); Readln(Res);
        End;
    End;
End;

```

⇒ Consulta el Peso a partir del Nombre

```

Procedure Consulta(V:TV; N: Byte; Nom: St20; Var Peso: Real);
Var
    J: Byte;

```

```
Begin
  J := 1;
  While(J <= N) and (V[J].Nombre <> Nom) do
    J := J+1;
  If J <= N then
    Peso := V[J].Peso
  else
    Peso := 0;
End;
```

Notar que es posible reemplazar un conjunto de arreglos paralelos por un arreglo de registros (un campo por cada arreglo).

### Actividades sugeridas

- A partir de las declaraciones del ejemplo anterior
  - ⇒ Hallar el nombre de la persona más joven.
  - ⇒ Hallar el promedio de altura.

- Dadas las siguientes declaraciones de tipo y variable:

```
Type
TRF = Record
  dd, mm: byte;
  aa: word;
end;

TRExam = Record
  Fecha: TRF;
  Nota: real;
end;

TRMateria = Record
 Codigo: String [4];
  Exámenes: array [1..10] of TRExam;
  ContE: byte;
end;

TRAlumn = Record
  Matricula: string [4];
  Materias: array [1..20] of TRMateria;
end;

TVAlum = array [1..100] of TRAlumn;

Var
  V:TVAlum;
```

Se pide:

- Escribir del alumno i, la materia j, el mes del examen k.
- asignar nota 8 al último examen de la segunda materia del tercer alumno

**Ejercitación**

*Desarrollar programas Pascal que resuelvan los problemas propuestos utilizando funciones y procedimientos. Proponer juegos de datos variados y verificar su funcionamiento.*

**1.-** Considerando las siguientes declaraciones:

```

type
  St5 = string[5];
  TR1 = record
    C1: byte;
    C2: array[1..20] of real;
    C3: St5;
  end;
  TR2 = record
    C11: TR1;
    C12: char;
    C13: real;
  end;
var
  Reg, RegI: TR1;
  Reg1, RegI1: TR2;

```

Escribir las sentencias para:

- ✓ Leer REG (en C1 se almacenará la dimensión lógica de C2)
- ✓ Escribir REGI1
- ✓ Asignar todos los campos de REG a REGI.
- ✓ Asignar el campo C3 de REG al campo C3 de REGI.
- ✓ Asignar el campo C11 de REG1 al campo C11 de REGI1.
- ✓ Almacenar el promedio del campo C2 de REG al campo C13 de Reg1

No desarrollar un programa, solo las sentencias correspondientes a cada ítem.

**2.-** Programar (adaptando el código existente) los ejercicios 6, 7, 8 y 14 del Capítulo 7. Reemplazar arreglos paralelos por arreglos de registros.

**3.-** Ingresar y almacenar en un arreglo de registros los datos de N autos, por cada uno: Marca, Modelo (año de fabricación), Patente, VTV (registra si tiene o no la VTV realizada)

Para una determinada MARCA (dato que se ingresa por teclado) se pide generar un arreglo de registros con los automóviles que no tienen VTV. En el nuevo arreglo se almacenarán únicamente: Patente, Antigüedad (diferencia entre el año actual y el modelo). Mostrar el arreglo generado.

**4.-** Leer un conjunto de números reales desde un archivo de texto considerando que los ceros separan secuencias. Generar un arreglo de registros que contenga para cada secuencia:

- cantidad de positivos
- cantidad de negativos

No generar elemento en el arreglo si alguna de las dos cantidades es cero.

Mostrar el arreglo generado.

Por ejemplo: si los datos fueran 3 , -5 , 4 , 8 , -1, 0 , 17 , 8 , 3 , 0 , -90 , -45 , 7, -87 0

El resultado es

3	:	2	:	1	:	3
---	---	---	---	---	---	---

**5.-** A partir de una matriz de  $N \times M$  elementos enteros, generar un arreglo de registros que contenga los elementos distintos de 0 y su posición dentro de la matriz (fila y columna). Luego, a partir del arreglo:

- a.- Determinar el porcentaje de elementos nulos de cada fila, sobre el total de nulos
- b.- Reproducir en pantalla la matriz completa.

**6.-** A partir de una matriz de  $N \times N$  elementos reales, generar un arreglo de registros, cada uno deberá contener el número de fila y el promedio de los elementos mayores al de la diagonal respectivo, si no hubiera ninguno no generar registro para dicha fila.

Mostrar los resultados obtenidos.