



Tema 9. Registros

- 9.1. Definición
- 9.2. Sintaxis
- 9.3. Operaciones de acceso
 - ▲ 9.3.1 Registro entero (Asignación)
 - ▲ 9.3.2 Campos del registro
- 9.4. Instrucción WITH
- 9.5. Registros con parte variante.



9.1. Definición

- **Definición:**

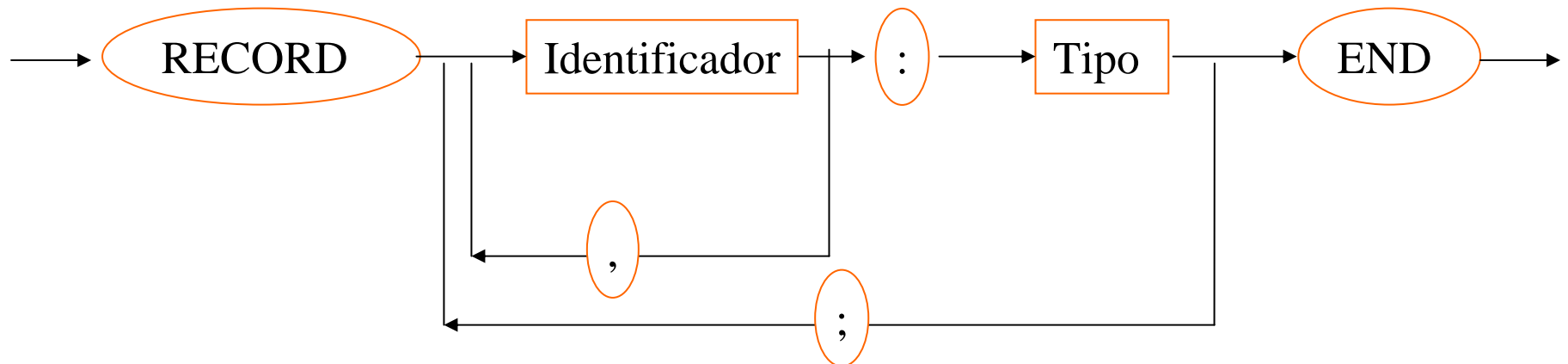
Tipo de datos estructurado que permite almacenar datos heterogéneos

- Cada uno de estos datos se denomina **campo** del registro.
- Cada campo tiene un nombre llamado *identificador de campo*.



9.2. Sintaxis

Diagrama Sintáctico





9.2. Sintaxis

- **SINTAXIS:**

TYPE

```
TNombreReg = RECORD
    idCampo1 : idTipo1;
    idCampo2 : idTipo2;
    . . .
    idCampoN : idTipoN
END; {Fin de TNombreReg}
```



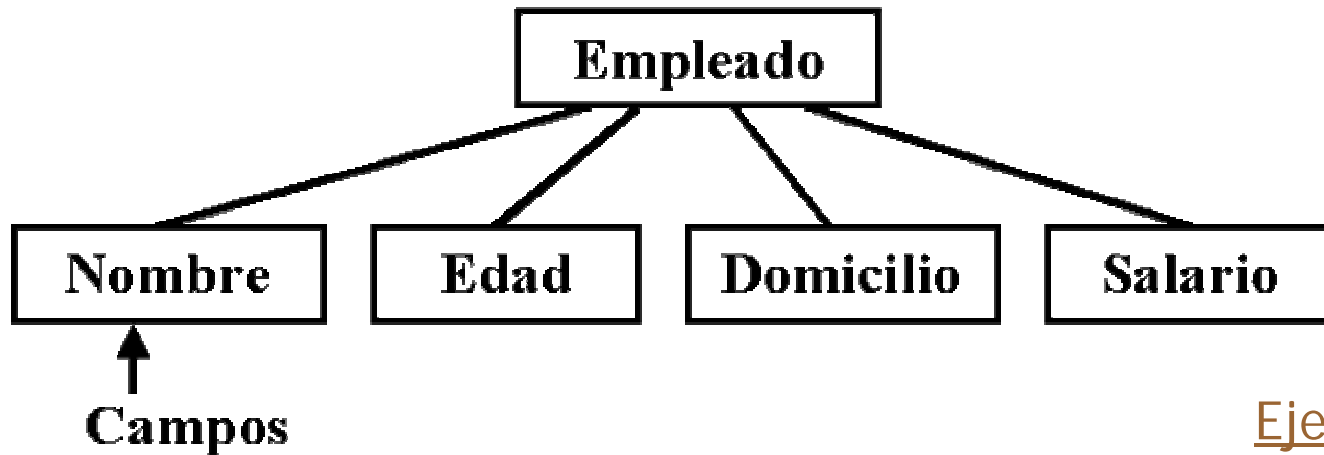
9.2. Sintaxis

- El tipo de los campos puede ser cualquier tipo predefinido de Pascal o uno definido previamente por el programador

Empleado:

Nombre	Edad	Domicilio	Salario
---------------	-------------	------------------	----------------

tipo cadena tipo entero tipo cadena tipo real



[Ejemplo 9.1](#)

[Ejemplo 9.2](#)



Ejemplo 9.1

CONST

N = 50;

INI = 16;

FIN = 65;

TYPE

TEdades = INI..FIN;

TCadena = string[N];

TEmpleado = RECORD

 Nombre : TCadena;

 Edad : TEdades;

 Domicilio : TCadena;

 Salario : real;

END;



Ejemplo 9.2

- El tipo de los campos también puede ser de tipo registro:

```
CONST
```

```
  N = 30;
```

```
  M = 10;
```

```
TYPE
```

```
  TCadena = string[N];
```

```
  TMatricula = string[M];
```

```
  TCoche = RECORD
```

```
    Marca, Modelo : TCadena;
```

```
  END; {TCoche}
```

```
  TAutomovil = RECORD
```

```
    Matricula : TMatricula;
```

```
    Nombre : TCoche; {Reg. Anidados}
```

```
    Precio : real
```

```
  END; {TAutomovil}
```



9.2. Sintaxis

- No puede utilizarse el mismo identificador para especificar dos campos distintos de un mismo registro. (Ejemplo 9.3)
- Dos campos de registros distintos pueden tener el mismo identificador. (Ejemplo 9.4)



Ejemplo 9.3

- Ejemplo:

```
CONST
```

```
    N = 30;
```

```
TYPE
```

```
    TCadena = string[N];
```

```
    TEmpleado = RECORD
```

```
        Nombre : TCadena; {Nombre del empleado}
```

```
        DNI     : integer;
```

```
        Nombre : TCadena; {Nombre de la empresa}
```



```
        {ERROR}
```

```
    END; {TEmpleado}
```



Ejemplo 9.4

CONST

N = 30;

D = 10;

INI = 1;

FIN = 8;

TYPE

TNombre = string[N];

TDNI = string[D];

TAsignaturas = INI..FIN;

TNotas = array [TAsignaturas] OF real;

TEmpleado = RECORD

Nombre, Apellido : TNombre; {Nombre}

DNI : TDNI; {del empleado}

END; {TEmpleado}

TAlumno = RECORD

Nombre : TNombre; {Nombre del alumno}

DNI : TDNI;

Notas : TNotas; {Notas}

END; {TAlumno}



9.2. Sintaxis

- Ejemplo 9.5:

Escribir una declaración de tipos que permita almacenar la información relativa a los trabajadores de una organización de forma cómoda y controlada.

La información por cada trabajador es la siguiente: nombre, dirección, edad, n^o DNI y sueldo.



Ejemplo 9.5

CONST

```
N = 50;  
IPOS = 1;  
FPOS = 8;  
INI = 18;  
FIN = 65;
```

TYPE

```
TEdad = INI..FIN;  
TDigito = '0'..'9';  
TPosicion = IPOS..FPOS;  
TNombre = string[N];  
TDNI = array [TPosicion] OF TDigito;  
TEmpleado = RECORD  
    Nombre, Direccion : TNombre;  
    Edad : TEdad;  

```

VAR

```
empleado, supervisor : TEmpleado
```



9.3. Operaciones

- **Asignación**

A las variables de tipo registro se les puede asignar **registros completos** si las dos variables son de **tipos idénticos** (Ejemplo 9.6)



Ejemplo 9.6

Basándonos en la definición de tipos del Ejemplo 9.5:

```
VAR
```

```
Empleado1, Empleado2 : TEmpleado;
```

```
BEGIN
```

```
• • •
```

```
Empleado2 := Empleado1;
```



9.3. Operaciones

Acceso a los campos de un registro

- Para **acceder** a un **campo** de una variable de tipo registro se utiliza el **operador punto** (.)

NombreVariableRegistro.NombreCampo

Donde **.NombreCampo** => Selector de Campo

Ejemplo 9.7



Ejemplo 9.7

```
VAR
  Empleado1, Empleado2: TEmpleado;
BEGIN
  ...

  Empleado1.Nombre := 'Juan Pérez';
  readln(Empleado2. Direccion);
  Empleado1.Edad := Empleado2.Edad;

  ...
```



9.3. Operaciones

Acceso a los campos de un registro

- La vigencia de un nombre de campo es la misma que la del registro en el que se ha definido.
- Con un campo de una variable de tipo registro se pueden usar todos los operadores que se utilizan con variables de su mismo tipo.



9.3. Operaciones

Acceso a los campos de un registro

- Asignación

La asignación de valores a los campos de una variable de tipo registro se hace igual que la asignación en variables del tipo de datos del campo selector. ([Ejemplo 9.8](#))

- Lectura y Escritura

Las operaciones de **lectura** y **escritura** de una variable de tipo registro se hacen **campo a campo**. ([Ejemplo 9.9](#))



Ejemplo 9.8

```
VAR
  Empleado1, Empleado2:TEmpleado;
BEGIN
  ...
  Empleado1.Sueldo := 1250.0;
  Empleado1.Nombre := 'Juan Jose';
  Empleado2.Sueldo := Empleado1.Sueldo + 500.0;
  Empleado2.Edad := 31;
  Empleado2.DNI[3] := '5';
  ...
```



Ejemplo 9.9

LECTURA

```
VAR
    Empleado1: TEmpleado;
BEGIN
    ....
    readln(Empleado1.Nombre);
    readln(Empleado1.Direccion);
    readln(Empleado1.Edad);
    FOR i:=Ini TO Fin DO
        read(Empleado1.DNI[i]);
    readln(Empleado1.Sueldo);
    ....
```



Ejemplo 9.9

ESCRITURA

VAR

```
Empleado1: TEmpleado;
```

BEGIN

```
.....
```

```
writeln('El nombre es: ', Empleado1.Nombre);
```

```
writeln('vive en: ', Empleado1.Direccion);
```

```
writeln('Tiene ', Empleado1.Edad, ' años');
```

```
writeln('Su documento nacional de identidad es:
```

```
' );
```

```
FOR i:=INI TO FIN DO
```

```
    write(Empleado1.DNI[i]);
```

```
writeln('Gana ', Empleado1.Sueldo, '€/mes');
```

```
.....
```



9.3. Operaciones

Variables de tipo registro como parámetros:

- Se pueden utilizar como parámetros de subprogramas.
- El tipo de los parámetros formales debe coincidir con el tipo de los parámetros reales.
- **CUIDADO!!** ⇒ Las funciones **no** pueden devolver una variable de tipo registro.

(Ejemplo 9.10)



Ejemplo 9.10

TYPE

```
TEjemplo = RECORD
```

```
  X : integer;
```

```
  Y : real;
```

```
END;
```

VAR

```
  uno, dos : TEjemplo;
```

```
PROCEDURE Prueba1(a: TEjemplo; VAR b:TEjemplo);
```

```
BEGIN
```

```
  . . .
```

```
END;
```

```
FUNCTION Prueba2(m: TEjemplo) : TEjemplo;
```



9.4. Instrucción WITH

Instrucción WITH

Evita la repetición del nombre de la variable de tipo registro cuando utilizamos seguidamente varios campos de un registro.

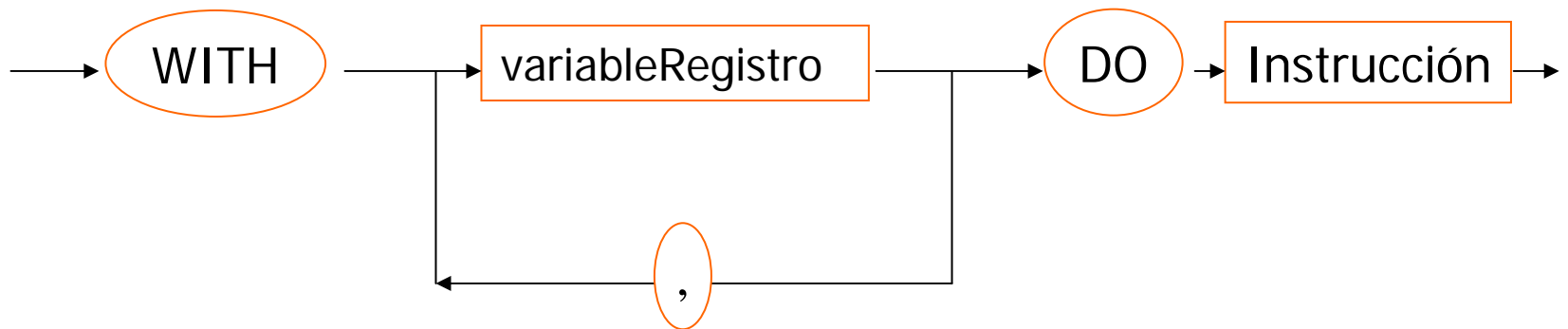
- Sintaxis:

WITH variableRegistro **DO**
Instrucción | secuencia de instrucciones



9.4. Instrucción WITH

- **Diagrama Sintáctico**





Ejemplo 9.11

```
CONST
```

```
  N = 30;  
  INI = 18;  
  FIN = 65;
```

```
TYPE
```

```
  TCadena = string[N];  
  TEdades = INI..FIN;  
  TFicha = RECORD  
      Nombre: TCadena;  
      Edad: TEdades;  
      Sueldo: real;
```

```
  END;
```

```
VAR
```

```
  Profesor : TFicha;
```

```
BEGIN
```

```
  ...  
  WITH Profesor DO  
    writeln(Nombre, Edad, Sueldo);
```

```
  |||
```

```
  writeln(Profesor.Nombre, Profesor.Edad, Profesor.Sueldo);
```



9.4. Instrucción WITH

- Los identificadores que no sean selectores de campo se tratan como si no existiese la instrucción WITH.
- No debe usarse la instrucción WITH con variables que tengan campos con identificadores iguales.



9.5. Registros con parte variante

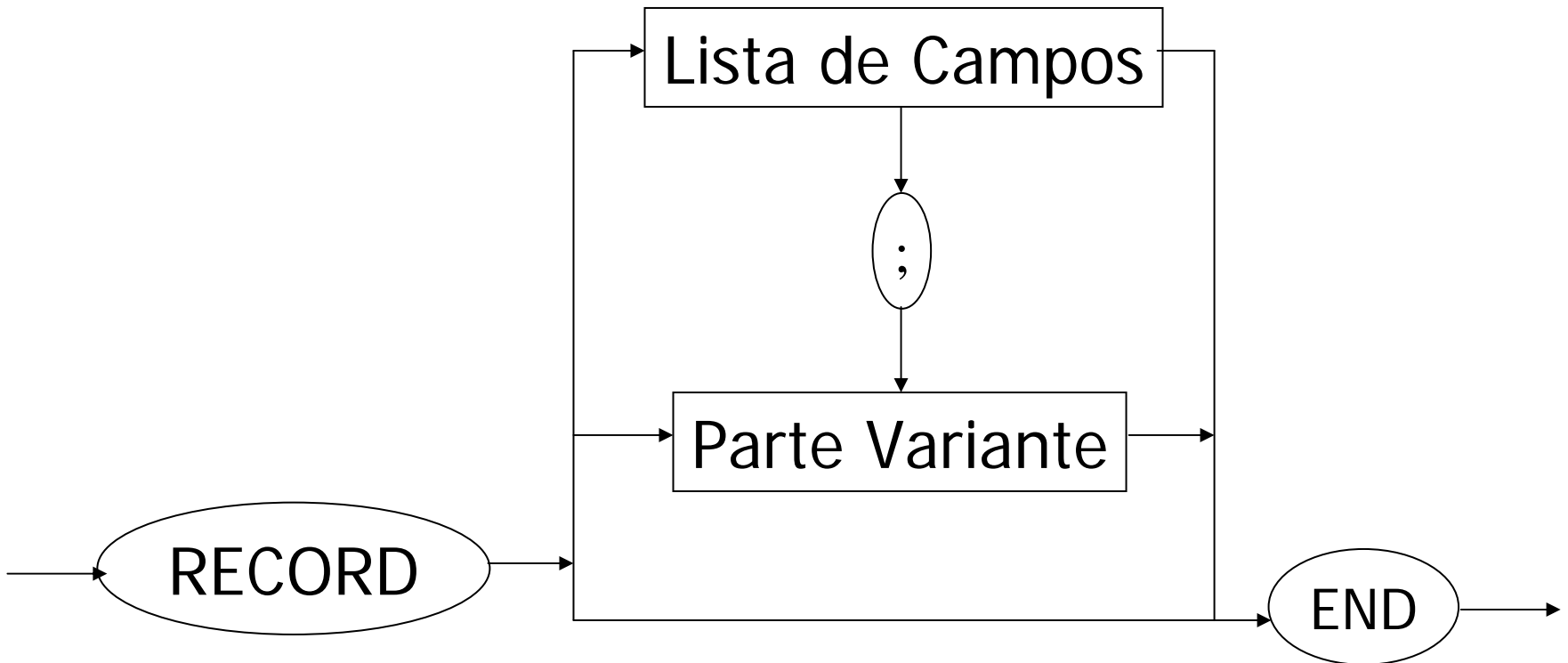
Registros con partes variantes

- Se usan para representar información con **una parte común** (parte fija) y con **otra parte** que puede contener **campos diferentes** (parte variante).
- Los registros con parte variante se declaran de un modo similar a los registros fijos excepto que hay un campo de indicador especial (**campo discriminante**) con una o más opciones (variantes).



9.5. Registros con parte variante

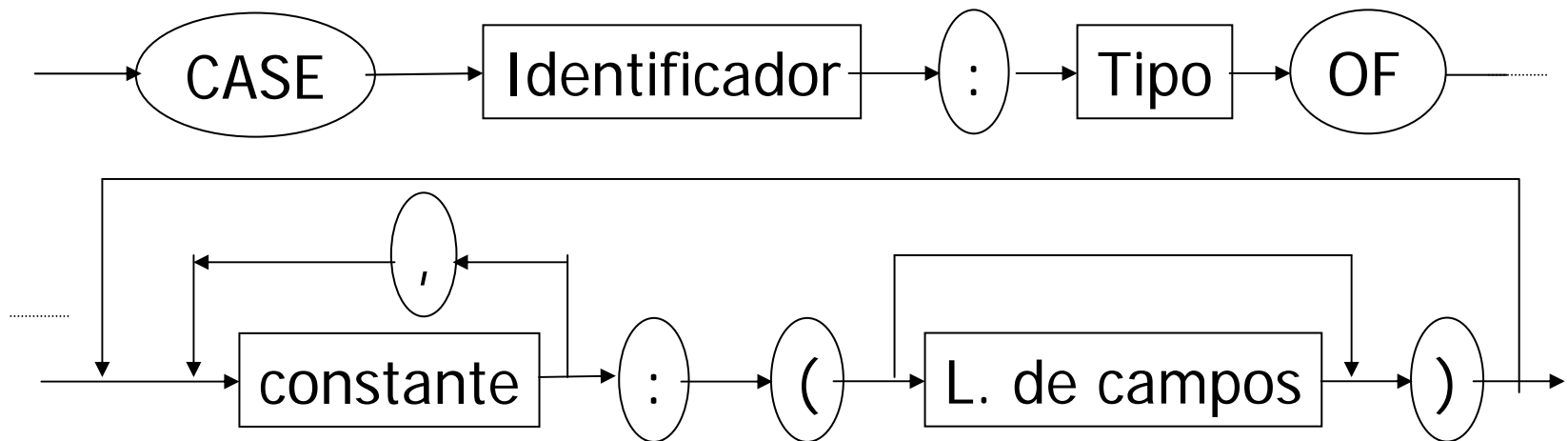
Diagrama Sintáctico





9.5. Registros con parte variante

Parte Variante





Ejemplo 9.12

TYPE

```
TSelector = ...;
```

```
TRegVar = RECORD
```

```
  Lista_Campos_Fijos;
```

```
  CASE Selector : TSelector OF
```

```
    listaConstantes1 : (listaCampos);
```

```
    ...
```

```
    listaConstantesN : (listaCampos)
```

```
  END; {TRegVar}
```

TSelector: Tipo Ordinal



9.5. Registros con parte variante

Parte fija:

- Igual

Parte variante:

- Campos del registro que sólo forman parte de algunos ejemplares del mismo.
- El número y los tipos de los campos variantes pueden cambiar durante la ejecución del programa.
- Se reserva memoria para el máximo número de campos.



9.5. Registros con parte variante

Campos Variantes

- Dependen del valor de una variable selectora o campo discriminante.
- El valor de la variable selectora o campo discriminante se selecciona mediante una estructura **CASE** .

Ejemplo 9.13



Ejemplo 9.13

```
CONST
  N = 40;
  INI = 1;
  FDIA = 31;
  FMES = 12;
  IANIO = 1900;
  FANIO = 2200;

TYPE
  TNombre = string[N];
  TForma = (Libro, Artículo);
  TDia    = INI..FDIA;
  TMes    = INI..FMES;
  TAnio   = IANIO..FANIO;
  TFecha  = RECORD
    Dia : TDia;
    Mes : TMes;
    Anio: TAnio;
  END;

  TPublicacion = RECORD
    {Parte fija}
    Autor   : TNombre;
    Titulo  : TNombre;
    Paginas: integer;
    {Parte variante}
    CASE Tipo : TForma OF
      Libro:(Editorial:TNombre; Anio:TAnio);
      Artículo:(Periodico:TNombre;
                Dia:TFecha);
    END; {TPublicacion}
  END;
```



Ejemplo 9.13

```
VAR
```

```
  Lib1, Art1 : TPublicacion;
```

```
BEGIN
```

```
  ...
```

```
  WITH Lib1 DO
```

```
  BEGIN
```

```
    Autor:= 'Gabriel García Márquez';
```

```
    Titulo:= 'El Coronel no tiene quien le  
             escriba';
```

```
    Paginas:= 110;
```

```
    Tipo:= Libro;
```

```
    Editorial:= 'Anagrama';
```

```
    Anio:= 1961;
```

```
  END;
```

```
  ...
```



Ejemplo 9.13

```
Art1.Autor := Lib1.Autor;  
Art1.Titulo:= 'La sirena escamada';  
Art1.Paginas:= 2;  
Art1.Tipo := Articulo;  
Art1.Periodico:= 'El Heraldo';  
WITH Art1.Dia DO  
BEGIN  
    Dia:= 31;  
    Mes:= 10;  
    Anio:= 1950;  
END;  
...
```



Ejemplo 9.13

```
PROCEDURE Muestra_datos(Datos:TPublicacion);
BEGIN
  WITH Datos DO
  BEGIN
    writeln('Autor: ',Autor);
    writeln('Título del libro: ',Titulo);
    writeln('Páginas: ',Paginas);
    IF Tipo = Libro THEN
    BEGIN
      writeln('Editorial: ',Editorial);
      writeln('Año de publicación: ',Anio);
    END
  END
END
```



Ejemplo 9.13

```
ELSE
BEGIN
    writeln('Periódico en el que apareció publicado:',
    Periodico);
    write('Fecha de publicación: ');
    escribe_fecha (Dia);
END;
END;
END;
```

Cuidado!!! Antes de acceder a la parte variante de un registro debemos verificar el valor del campo discriminante.



Aplicación de los registros

- Podemos simular arrays parcialmente llenos (APL) mediante un registro.

```
TYPE
```

```
TRango = ...;
```

```
TBase = ...;
```

```
TProducto = array [TRango] OF TBase;
```

```
TRegArray = RECORD
```

```
    tope          : integer;
```

```
    producto     : TProducto;
```

```
END;
```

```
VAR
```

```
    almacen : TRegArray;
```



Aplicación de los registros

```
BEGIN {pp}
    ...
    WITH almacen DO
    BEGIN
        tope := tope + 1;
        producto[tope] := producto[i];
    END;

    ...
    FOR j:= 1 TO almacen.tope DO
        Escribe_info(almacen.producto, j);
    ...
END. {pp}
```