



GUIA TEORICA N° 6
LENGUAJE DE PROGRAMACION I
APUNTES: REGISTROS.

No temas, porque yo estoy contigo; no
Desmayes, porque yo soy tu DIOS que
Te esfuerzo; siempre te ayudaré, siempre
Te sustentaré con la diestra de mi justicia»
Isaías 41:10

Profesor : DELY M. GIL A.

VALENCIA, 2007

REGISTROS

Un *registro* es una estructura que consiste de un número fijo de componentes llamados *campos*. Los *campos* pueden ser de diferentes tipos y deben tener un *identificador de campo*.

DECLARACIÓN

```
type
  tipo_reg = record
    lista id1:tipo 1;
    lista id2:tipo 2;
    .
    .
    lista idn:tipo n
  end;
```

DECLARACIÓN

tipo_reg nombre de la estructura o dato registro
lista id lista de uno o más nombres de campos
separados por comas
tipo puede ser cualquier tipo de dato estándar
o definido por el usuario

EJEMPLO

```
Type
  Cad30      = string[30];
  Cad40      = string[40];
  Empleado  = Record
    Nombre   : cad30 ;
    edad     : byte  ;
    Domicilio : cad40 ;
    salario  : real
  end;

Fecha = Record
  Mes: (Enero,Febrero,Marzo
    ,Abril,Mayo,Junio,
    Julio,Agosto,Septiembre,Octubre,
    Noviembre,Diciembre);
  Dia  : 1..31;
  Anio : 1990..2005;
end;

VAR
  Trabajador : Empleado;
  Nacimiento : Fecha
```

ACCESO A LOS CAMPOS :

Nom bre_Reg.nombre_Campo

OPERACIONES

Asignación:

```
.....
trabajador.nombre := 'Juan Gil';
trabajador.edad   := 35;
trabajador.domicilio := 'Merida';
trabajador.sueldo := 1200000 ;
.....
```

Lectura:

```
.....
write('Nombre: ');
Readln(Trabajador.Nombre);
write('Edad: ');   Readln(Trabajador.Edad)
write('Domicilio: ');
readln(trabajador.domicilio);
write('Sueldo: ');
readln(trabajador.sueldo);
.....
```

Escritura:

```
.....
write('Nombre: ');
writeln(Trabajador.Nombre);
write('Edad:');
Write(Trabajador.Edad)
write('Domicilio: ');
writeln(trabajador.domicilio);
write('Sueldo: ');
writeln(trabajador.sueldo);
```

Sentencia WITH DO

Permite referenciar el nombre del registro en su cabecera y posteriormente para llamar a algún campo sólo se necesita el nombre del campo. "La instrucción WITH permite evitar las tediosas repeticiones del nombre de una variables tipo registro" (Hennefeld, 1989) cuando el número de campo es grande. La sintaxis es :

```
With VarRegistro do
  Begin
    { Sentencias que hacen referencia a campos de
      VarRegistro}
  End;
```

OPERACIONES

Asignación:

```
.....
with trabajador do
begin
nombre      := 'Juan Gil';
edad        := 35';
domicilio   := 'Merida';
sueldo      := 1200000 ;
.....
```

OPERACIONES

Lectura:

```
.....
with trabajador do
Begin
write('Nombre: ');
Readln(Nombre);
  write('Edad:');
  Readln(Edad)
  write('Domicilio: ');
  readln(domicilio);
  write('Sueldo: ');
  readln(sueldo);
End;
```

Escritura:

```
with trabajador do
Begin
  write('Nombre: ');
  writeln(Nombre);
  write('Edad:');
  Write(Edad)
  write('Domicilio: ');
  writeln(domicilio);
  write('Sueldo: ');
  writeln(.sueldo);
End;
```

REGISTROS ANIDADOS

Un registro con uno o más campos que son del tipo registro se llaman Registro Jerárquicos o Registros Anidados.

Type

```
Empleado = Record
    Nombre   : cad30 ;
    edad     : byte ;
    Domicilio : cad40 ;
    salario  : real
    fecha_ing : fecha;
            fecha_Nac : fecha
end;
```

Var

```
Trabajador : Empleado;
```

ASIGNACIÓN Y ACCESO

```
Trabajador.fecha_ing.dia := 15;
```

```
With trabajador do
```

```
    With fecha_ing do
```

```
        Dia := 15
```

```
With trabajador,fecha_ing do
```

```
    Dia := 15
```

REGISTROS COMO PARÁMETROS

Los registros pueden ser usados como parámetros en funciones y procedimientos. El uso de registros abrevia la lista de parámetros.

a.- Parámetros por valor. Ejemplo: Escribir un procedimiento que reciba como parámetro de entrada un registro completo que contenga los datos estadísticos de calificaciones de una clase: Notas Inferior y superior (puntuaciones de 0 a 100), media y desviación estándar.

b.- Parámetros por Referencia: Escribir un procedimiento que lea los valores de cada campo de un registro de Stock de un almacén y lo devuelva al programa principal.

ARREGLOS DE REGISTROS

Los registros se agrupan en conjuntos conocidos como arrays de registros.

CAMPOS DE REGISTROS TIPO ARREGLOS

El tipo del cualquier campo de un registro puede ser la estructura definida por el Usuario, en este caso un VECTOR

Declaración del Registro

```
TYPE
  Estadística = record
    Inferior, Superior: 0..100;
    Media, desviacion: real;
  End;
VAR
  Clase: Estadística;
  (***)Procedimiento(***)
Procedure DatosClase(Aula: Estadística);
  Begin
    With Aula do
      Begin
        Writeln('Puntuación Mayor: ', Superior:3);
        Writeln('Puntuación Menor: ', Inferior:3);
        Writeln('Media Aritmética: ', Media:3);
        Writeln('Puntuación Menor: ', Desviacion:3:1);
      End;
    End;
  End;
```

Declaración del Registro

```
TYPE
  Stock = record
    Numeros: integer;
    Nombre :string[20];
    Precio : real;
  End;
VAR
  Item :Stock;
Procedure Leer_Registro(Var Artículo:Stock);
  Begin
    Writeln('Introducir numero de la Serie: ');
    Readln(Artículo.Numeros);
    Writeln('Ingrese descripción del Artículo: ');
    Readln(Artículo.Nombre);
    Writeln('Ingrese precio: ');
    Readln(Artículo.precio);
  End;
```

```
Program AR;
uses
  crt;
type
  ItemStock = record
    Nombre: string[30];
    Numero : integer;
    precio : real;
  end;
  Lista = array[1..100] of ItemStock;
Var
  registro : ItemStock;
  Inventario : Lista;
  I : byte;

Procedure Leer_Registro(var Item:ItemStock);

  begin
    writeln('Nombre:');
    readln(Item.Nombre);
    write('Numero de stopck');
    readln(Item.Numero);
    write('Precio');
    readln(Item.Precio);
  end;

{ *****Programa Principal*****}
begin
  for i:= 1 to 2 do
    begin
      writeln('el siguiente elemento es: ');
      { Llamada a un Procedimiento:
        Leer_Registro(Inventario[i]);}

      {*****con la sentencia with Inventario[i]*****}
      with Inventario[i] do
        begin
          writeln('Nombre:');
          readln(Nombre);
        end; }
    end;
end;
```

```
Program Registro;
{EL siguiente programa captura 10 empleados y sus
datos personales en un arreglo utilizando registros }
Uses Crt;
Const
  MaxEmpleados=10;
Type
  registro_empleado = Record
    nombre : string[30];
    profesion : string[20];
    puesto : string[20];
    sueldo : real
  end;

Var
  registro : registro_empleado;
  empleado :array[1..MaxEmpleados] of registro_empleado;
  i,col,ren :byte;
begin
  ClrScr;
  Write(' Nombre Profesión Puesto Sueldo');
  col:=1;ren:=2;
  for i:=1 to MaxEmpleados do
    begin
      With EMPLEADO[I] do
        begin
          gotoxy(col,ren);
          ReadLn(nombre);
          gotoxy(col+21,ren);
          ReadLn(profesion);
          gotoxy(col+40,ren);
          ReadLn(puesto);
          gotoxy(col+59,ren);
          ReadLn(sueldo);
          inc(ren);
          col:=1;
        end
      end;
      ReadKey
    end.
end.
```

```
Program RA;
Uses
  Crt;
Const
  MaxAlumno = 5;

Type
  cad30      = string[30];
  cad60      = string[60];
  Tipo_Nota = array[1..3] of real;
  reg_alumno = record
    Nombre   : cad30;
    Ced      : longint;
    Notas    : Tipo_Nota;
    promedio : real;
  end;
  lista     = array[1..maxAlumno] of reg_alumno;
Var
  alumno    : reg_alumno;
  alumnos   : lista;
  col,ren,i : byte;
  V_PromAsig : Tipo_Nota;

{*****}
{          Procedure Mensaje          }
{*****}
procedure mensaje(cad:cad60);

begin
  gotoxy(22,25);write(cad);
  delay(7000);
  gotoxy(22,25);clreol;
end;
```

```
{*****}
{          Procedure Valida Cadenas          }
{*****}
procedure valida_cadenas(var cadena:cad30;x,y:byte);
var band: boolean; i : byte;

begin
  repeat
    gotoxy(x,y); write(' ');
    band:= true;
    i:= 1;
    gotoxy(x,y);readln(cadena);
    repeat
      if cadena[i] in ['A'..'Z','a'..'z'] then
        inc(i) {= i:=i+1}
      else
        begin
          band:= false;
          mensaje('Nombre inv lido!!, Intente de nuevo');
        end;
    until ((i = length(cadena) + 1) or (band= false));
  until(band);
end;

{*****}
{          Procedure Leer_Reg          }
{*****}
Procedure Leer_Reg(var Reg: reg_alumno; colu,fila:byte);
var
  suma : real; i : byte;
```

```
begin
  gotoxy(colu,filas);readln(Reg.Ced);
  valida_cadenas(Reg.Nombre,colu+8,filas);
  colu:=30; suma:= 0;
  for i:= 1 to 3 do
    begin
      gotoxy(colu,filas);readln(Reg.Notas[i]);
      suma := suma + Reg.Notas[i];
      colu:= colu+10;
    end;
  Reg.promedio:= suma/3;
  gotoxy(colu,filas);write(Reg.promedio:3:2);
end;

{*****}
{          Procedure Imprimir_Reg          }
{*****}
Procedure Imprimir(A:Tipo_Nota;filas,colu:byte);
var
  i:byte;

begin
  for i:= 1 to 3 do
    begin
      gotoxy(colu,filas);write(A[i]:3:2);
      colu:= colu+10;
    end;
end;
```

```
{*****}
{          Procedure PromAsig          }
{*****}
Procedure PromAsig(Reg:lista;var B:Tipo_Nota);
var
  suma: real;
  i,j :byte;
begin
  for i:= 1 to 3 do
    begin
      suma:= 0;
      for j:= 1 to MaxAlumno do
        begin
          suma:= suma + Reg[j].Notas[i];
        end;
        B[i]:= suma/5;
      end;
    end;
end;

{*****}
{          Programa Principal          }
{*****}
Begin
  ClrScr;
  gotoxy(10,5);Write('Cedula Nombre Asignat1 Asignat2
Asignat3 Promedio');
  col:=10;ren:=6;
  For i:=1 to MaxAlumno do
    Leer_Reg(alumnos[i],col,ren+i);
    PromAsig(Alumnos,V_PromAsig);
    gotoxy(10,13);write('Promedio Asignat');
    Imprimir(V_PromAsig,13,30);
    gotoxy(15,22);
    Write('Presione una tecla para continuar....');
    readkey;
end.
```

Ejercicio:

1.- Una empresa tiene almacenados a sus vendedores en un registro. Por cada vendedor se guarda su DNI, Apellidos, Nombre, Sueldo Base, Ventas Mensuales, Total Anual y Comisión. Las ventas mensuales será un vector de 12 elementos que guardará las ventas realizadas en cada uno de los meses.. Total Anual será la suma de las ventas mensuales del vendedor. La Comisión se calculará aplicando un porcentaje variable al Total Anual del vendedor. Dicho porcentaje variará según las ventas anuales del vendedor, según la siguiente tabla:

Hasta 1.500.000	por 100
Más de 1.500.000 y hasta 2.150.000	13,75%
Más de 2.150.000 y hasta 2.900.000	16,55%
Más de 2.900.000 y hasta 3.350.000	17,6%
Más de 3.350.000	18,85%

```
program vendedores;
```

```
uses
```

```
  crt;
```

```
type
```

```
  cadena30 = string[30];
```

```
  tipo_ventas = array[1..12] of integer;
```

```
  reg_vendedor = record
```

```
    dni :longint;
```

```
    nombre,apellido,zona :cadena30;
```

```
    sueldo_base, total_anual,
```

```
    comision :real;
```

```
    ventas_mes :tipo_ventas;
```

```
  end;
```

```
reg_comision= record
    lim_sup :longint;
    porcent :real;
end;
lista_vend = array[1..2] of reg_vendedor;
lista_com = array[1..5] of reg_comision;
```

```
var
```

```
  vendedor : reg_vendedor;
```

```
  vend: lista_vend;
```

```
  comision : lista_com;
```

```
(***** procedimiento leer comision *****)
```

```
procedure leer_comision ( var com:lista_com );
```

```
var
```

```
  i: byte; {sub_indice tabla comisiones}
```

```
begin
```

```
  for i:= 1 to 5 do
```

```
    begin
```

```
      write('limite superior de venta');
```

```
      read(com[i].lim_sup);
```

```
      write('porcentaje de comision');
```

```
      read(com[i].porcent);
```

```
    end;
```

```
  end;
```

```
function calculo_comision (total_anual:real; com:lista_com): real;
```

```
var
```

```
  i: byte;{sub-indice de la lista de comisiones}
```

```
begin
```

```
  i:= 1;
```

```
  while (com[i].lim_sup < total_anual) and (i<> 5) do
```

```
    inc(i);
```

```
    calculo_comision:= com[i].porcent * total_anual;
```

```
  end;
```

```
(***** procedimiento leer vendedor *****)
procedure leer_vendedor (var reg:lista_vend;com:lista_com);
var
  i : byte; {sub_indice de las ventas mensuales}
  j : byte; {sub_indice de los vendedores}
begin
  for j:=1 to 2 do
    begin
      with reg[j] do
        begin
          write('DNI');      readln(dni);
          write('nombre');   readln(nombre);
          write('apellido'); readln(apellido);
          write('zona');     readln(zona);
          write('sueldo base'); readln(sueldo_base);
        end;
      for i:= 1 to 12 do
        begin
          read (reg[j].ventas_mes[i] );
          reg[j].total_anual:=reg[j].total_anual +
reg[j].ventas_mes[i];
        end;
      reg[j].comision:= calculo_comision (reg[j].total_anual,
com);
    end;
  end;
end;
```

```
(*****procedimiento menu *****)
procedure menu;
var
  opcion: char;
begin
  repeat
    clrscr;
    gotoxy(23,2);write ('menu principal');
    gotoxy(20,4);write ('1.- ingresar porcentaje de comision');
    gotoxy(20,6);write ('2.- ingresar datos de los vendedores. ');
    gotoxy(20,8);write ('3.- salir (fin del programa) ');
    gotoxy(24,20);write ('elija una opcion');
    repeat
      opcion:= readkey;
    until (opcion in ['1'..'3']);
    clrscr;
  case opcion of
    '1': leer_comision(comision);
    '2': leer_vendedor(vend,comision);
  end;
  until (opcion = '3');
end;
(***** procedimiento principal *****)

begin
  menu;
end.
```

```
Program Oficina;
uses Crt;
Const
N= 50;
Type
  reg_persona = record
      Ced      : LongInt;
      Nombre  : string;
      Sexo    : char;
      edad    : Byte
  End;
aPersonas = Array[1..N] of reg_persona;
Var
  Personas : aPersonas;
  Ultimo   : byte;
  Full     : Boolean;
  opc     : Char;

Procedure Incluir;
Var Temp : reg_persona;
Begin
  Write('Ingrese cedula:');
  Readln(Temp.Ced);
  write('Nombre:');
  Readln(Temp.Nombre);
  Write('Sexo:');
  readln(Temp.sexo);
  Write('Edad: '); readln(temp.edad);
  Inc(Ultimo);
  Personas[Ultimo] := Temp;
  If Ultimo = N Then Full:=true
End;
```

```
Procedure Eliminar;
Var
  Posi,i,j : Byte;
  Cedula   : Longint;
  Encontrado : boolean;
Begin
  Posi := 0;
  Write('Ingrese la Cedula:');
  Readln(Cedula);
  i := 1;
  Encontrado := False;
  While (i<=Ultimo) and (Encontrado = false) do
    Begin
      If (cedula = Personas[i].ced) Then
        Begin
          Encontrado:= true;
          Posi := i
        End
      Else
        i := i+1;
    End;
  If (encontrado = true) Then
    Begin
      For J:= Posi to (Ultimo-1) do
        Personas[j]:= Personas[j+1];
      Dec(Ultimo)
    End
  Else
    Write('El elemento no existe');
  End;
```

```
Procedure Modificar;
Var
  Cedula      : LongInt;
  i,Posi      : Byte;
  Encontrado  : boolean;
  opc         : Char;
Begin
  Write('Ingrese cedula:');
  Readln(Cedula);
  i      := 1;
  Encontrado := false;
  While (i<=Ultimo) and (Encontrado = false) do
  Begin
    If (cedula = Personas[i].ced) Then
      Begin
        Encontrado:= true;
        Posi      := i
      End
    Else
      I:= I+1;
  End;
  If (Encontrado = True) then
  Begin
    Repeat
      Writeln('1.Nombre:',Personas[i].Nombre);
      Writeln('2.Sexo  :',Personas[i].Sexo);
      Writeln('3.Edad  :',Personas[i].edad);
      Writeln('4.Salir');
      Writeln('Indique el campo a modificar o 4 para salir');
      opc:= readkey;
      Case opc of
        '1' : Begin
          write('Nombre Modificado: ');
          Readln(Personas[posi].nombre);
          End;

```

```
        '2' : Begin
          write('Sexo Modificado: ');
          Readln(Personas[posi].sexo);
          End;
        '3' : Begin
          write('Edad: ');
          Readln(Personas[posi].edad);
          End;
      Until(opc = '4');
    End
  Else
    Write('El elemento no existe')
  End;

Procedure Consultar;
Var
  Cedula      : LongInt;
  Encontrado  : boolean;
  i,positi    : Byte;
Begin
  Write('Ingrese cedula:');
  Readln(Cedula);
  i      := 1;
  Encontrado := false;
  While (i<=Ultimo) and (Encontrado = false) do
  Begin
    If (cedula = Personas[i].ced) Then
      Begin
        Encontrado := true;
        Posi      := i
      End
    Else
      I:= I+1;
  End;
  End;

```

```
If (Encontrado = True) then
  Begin
    Writeln('1.Nombre:',Personas[i].Nombre);
    Writeln('2.Sexo :',Personas[i].Sexo);
    Writeln('3.Edad :',Personas[i].edad);
  End
Else Writeln('El elemento no existe');
Writeln('Presione tecla para terminar'); Readkey;
End;
```

```
Procedure Menu;
Begin
  Ultimo := 0;
  Repeat
    ClrScr;
    Writeln('Menu');
    writeln('1.- Insertar. ');
    writeln('2.- Modificar. ');
    writeln('3.- Consultar. ');
    writeln('4.- Eliminar. ');
    writeln('5.- Salir. ');
    opc := readkey;
  Case opc of
    '1' : If (Full= True) Then
      Writeln('No se puede incluir')
    Else
      Incluir;
    '2' : Modificar;
    '3' : Consultar;
    '4' : Eliminar
  End;
  Until (opc='5')
End;
```

```
(*****Programa Principal*****)
Begin
  Menu;
End.
```

- 1.- Realice la corrida en frío del programa anterior.**
- 2.- Defina las cabeceras con el paso de parámetros.**
- 3.- Desarrolle el subprograma de Búsqueda.**

TAREA (Investigar)

- √ **Tipos de datos enumerados (definición, Lectura y escritura)**
- √ **Registros variantes (definición y ejemplo)**
- √ **Unidades (definición y ejemplo).**