

República Bolivariana de Venezuela
Ministerio de Educación Superior
Instituto Universitario de Tecnología Valencia
Departamento de Informática



GUIA TEORICA N° 2 LENGUAJE DE PROGRAMACION I ESTRUCTURAS DE CONTROL: APUNTES.

Profesor : DELY M. GIL A.

VALENCIA, 2004

ESTRUCTURAS DE CONTROL

- Instrucciones Selectivas o Condicionales
IF
CASE
- Instrucciones Repetitivas
Automáticas FOR
Condicionales WHILE
REPEAT-UNTIL

Ing Del y Gil

INSTRUCCIONES COMPUESTAS

Es un conjunto de secuencias (como mínimo dos) separadas con punto y coma (;) y encerradas entre las palabras BEGIN y END. A las instrucciones compuestas también se les llama Bloques de Instrucciones, debido a que conllevan a la realización de más de una instrucción.

Nota: Debe tenerse en cuenta que el begin nunca esta acompañado de punto y coma (;), mientras que end si puede tener el punto y coma al final, dependiendo de la estructura o sentencia donde se encuentre.

Ing Del y Gil

ESTRUCTURAS DE SELECCIÓN SIMPLE

- La sentencia IF... THEN ...
If condicion Then
instrucción 1;
- La sentencia IF... THEN ...ELSE...
If condicion Then
instruccion1
*{ antes de un else no se debe colocar
punto y coma }*
Else
instruccion2;

Ing Del y Gil

Program Pares_Nones;

```
Uses CRT;  
Var  
  N , Res : integer;  
Begin  
  ClrScr; { ClearScreen }  
  Write('Ingrese un número: ');  
  Readln(N);  
  Res := N mod 2 ;  
  If res = 0 then  
    write('El numero es par')  
  else  
    write('el numero es impar');  
  Readkey  
End.
```

Ing Del y Gil

ESTRUCTURAS DE SELECCIÓN COMPUESTAS

```
If condicion Then  
  Begin  
    instruccion 1; } Instrucciones  
    instruccion2; } Compuestas  
    ...  
  End  
Else  
  Begin  
    instruccion1;  
    Instruccion2;  
  End;
```

Ing Del y Gil

SENTENCIAS IF ANIDADAS

```
If condicion1 Then  
  If condicion2 Then  
    instrucción X  
  Else  
    instrucción Y  
Else  
  Begin  
    instruccion1;  
    instruccion2;  
  End;
```

Ing Del y Gil

VARIABLES LÓGICAS (Sentencias de Asignación)

If X=Y Then Sw: = True Else Sw: = False; Sw: = X=Y;	If Resp in ['S', 's'] Then Casado: = True Else Casado: = False; Casado := Resp in ['S','s'];
If n>0 and n<10 Then Band: = True Else Band: = False	

Ing Del y Gil

SENTENCIA CASE

Selector : Pueden ser:

- √ Escalar (enumerado, subrango),
- √ Ordinal (Integer, Char, Boolean, Byte),
- √ variable, una función, una contante o una expresión.

Selector : NO Pueden ser:

- √ Real, string, longint, word

Lista de constantes : Pueden ser:

- √ Constantes
- √ Valor de subrango, o
- √ Posibles valores separados por coma.

Ing Del y Gil

SENTENCIA CASE

La sentencia CASE se utiliza para elegir entre diferentes alternativas evitando así, caer en una red de If anidados.

```

Case Selector Of
    Lista de constantes1: sentencias1;
    Lista de constantes2: sentencias2;
    ...
    Lista de constantesn: sentenciasn;
[else
    sentenciax]
End;      {End del Case}
    
```

Ing Del y Gil

COMPARACIÓN

CARACTERÍSTICAS	IF	CASE	IF EJEMPLO	CASE EJEMPLO
Evalúa variables de cualquier tipo	si	no	Todos los tipos	Integer, boolean y subrango
Permite evaluar más de una variable	si	si	If a>1 and a<8 Then s:=s+1;	1..8: s:=s +1;
Permite ejecutar un bloque de instrucciones para una alternativa	si	si	If a>1 Then Begin write('uno'); a:=2; end;	1: begin write('uno'); s:= 2; end;
Permite el uso o no de la cláusula ELSE.	si	si	If a>1 Then s:= s+1 else p:=p-1;	1,,8: S:= s+1 else write('Error');
Simplicidad en la sintaxis cuando hay más de una alternativa	no	si	If a>1 and a<<8 Then s:=s+1;	1..8: s:=s +1;
Mejor para evaluar una sola alternativa	si	no	Su mejor aplicación	Se estaría subutilizando

Ing Del y Gil

SENTENCIA CASE

Tenemos una Variable P que puede tener uno de los cuatro valores 1,2,3,4 y se desea escribir un mensaje para cada valor distinto de P.

Si P = 1 Escribir Color Rojo
 Si P = 2 Escribir Color Verde
 Si P = 3 Escribir No hay colores
 Si P = 4 Escribir Se agotó el color

(*****Ejercicios*****)

```
Read(Valor);
Case Valor of
    '+' : Res: = Res + 5;
    '-' : Res: = Res - 5;
    '*' : Res: = Res * 5;
    '/' : Res: = Res /5
Else
    write('Operador inválido')
end;      {End del Case}
```

(*****Ejercicios*****)

```
Case caracter_leido of
    '0'..'9' : write('Es una Cifra');
    'a', 'e', 'i', 'o', 'u' : write('Es una vocal-Minúscula');
    'A', 'E', 'I', 'O', 'U' : write('Es una vocal-Mayúscula');
    'a'..'z' : write('Es una letra
                    Minúscula');
    'A'..'Z' : write('Es una letra
                    Mayúscula');

    #0..#31 : write('Es un código de
                    Control')
Else
    write('Es otro carácter')
end;      {End del Case}
```

SENTENCIA CASE

Menú

N: Nombre , D: Dirección
 T: Teléfono , C: Ciudad F: Fin

{*****Programa Pascal*****}

Program Agenda;

Var

Seleccion : Char;

...

Begin

...

Readln(Seleccion);

Case Seleccion of

'N', 'n' : Begin

Write(' Ingrese Nombre: ');

Inst. { Readln(Nombre)
 Compuesta } End;

'D', 'd' : Begin

Write(' Ingrese Dirección: ');

Readln(Direccion)

End;

'T', 't' : Begin

Write(' Ingrese Teléfono: ');

Readln(Telefono)

End;

'F', 'f' : Begin

ClrScr;

Write('Ha salido del sistema ');

Readkey

End

End { End del Case}

End. {Fin del programa}

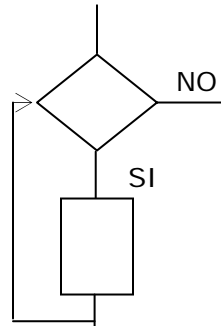
Si las instrucciones
son **compuestas**
se debe encerrar
entre
BEGIN- END

ESTRUCTURAS REPETITIVAS

Una estructura de control que permite la repetición de una serie determinada de sentencia se denomina BUCLE (LAZO o CICLO).

- La sentencia WHILE


```
While expresión lógica Do
  Begin
    instrucción 1;
    instrucción 2;
    ...
    instrucción N;
  End;
```



Ing Del y Gil

TERMINACIÓN DE LOS BUCLES

- Bucles controlados por contador
 1. Inicializar el contador

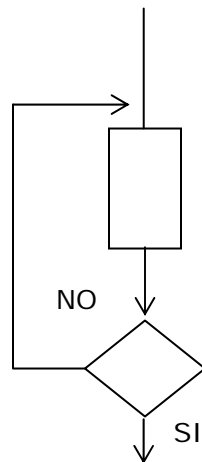
```
While contador < valor_final do
  Begin
    ...
    incremento del contador en uno;
  End;
```

Ing Del y Gil

ESTRUCTURAS REPETITIVAS

- La sentencia REPEAT-UNTIL


```
Repeat
  instrucción 1;
  instrucción 2;
  ...
  instrucción N;
until expresión lógica;
```



Ing Del y Gil

TERMINACIÓN DE LOS BUCLES

- Bucles controlados por condición


```
Band: = True;
While Band = True do {While Band Do}
  Begin
    ...
    If condición Then
      Band: = False;
  End;
```

Ing Del y Gil

TERMINACIÓN DE LOS BUCLES

```
Contador := Valor_Inicial;
Repeat
    ...
    incremento del contador en uno;
Until Contador = Valor_Final;

(*****Condicional*****)
```

```
Band := True;
Repeat
    ...
    Band := Condicion;
Until Not Band;
```

Ing Del y Gil

```
Program Ejem_while;
Uses
  Crt;
Const
  Esc = #27;
Type
  Cad20 = string[20];
Var
  nombre : cad20;
  tecla : char;
  cont : word;
Begin
  ClrScr;
  Cont := 1;
  tecla := '';
  While (tecla <> Esc) do
  Begin
    Write(cont, 'Nombre: ');
    Readln(Nombre);
    Inc(cont);
    tecla := Readkey;
  End
End.
```

Ing Del y Gil

TERMINACIÓN DE LOS BUCLES

```
Program Prueba;
Uses
  Crt;
Var
  veces, final : byte;
  suceda : boolean;
  Edo_Civil : Char;
Begin
  ClrScr;
  Veces := 0;
  Final := 10;
  Suceda := false;
  While (veces < Final) and (not Suceda) do
  Begin
    Write('Estado Civil: ');
    Readln(Edo_Civil);
    Veces := Veces + 1; { Inc(Veces)}
    Suceda := Edo_Civil in ['C', 'c'];
  End;
  writeln('Personas entrevistadas: ', veces);
  writeln('Solteras: ', Veces-1);
  readkey;
End.
```

.- Sumar N números donde N es un dato de entrada. Realizarlo con la estructura repetitiva WHILE y REPEAT.

Ing Del y Gil

```

Progra Ejem_Repeat;
Uses
  Crt;
Const
  Esc = #27;
Type
  Cad20 = string[20];
Var
  nombre : cad20;
  tecla   : char;
  cont    : word;
Begin
  ClrScr;
  Cont := 1;
  Repeat
    Write(cont,
           ' Nombre: ');
    Readln(Nombre);
    Inc(cont);
    tecla := Readkey
  Until (tecla = Esc)
End.

```

Ing Del y Gil

BUCLES WHILE/REPEAT ANIDADOS

```

While Cond_1 do
  Begin
    ...
    While Cond_2
      Begin
        ...
        End; {Cond_2}
      ...
    End; {Cond_1}
  ...
Repeat
  ...
  Repeat
    ...
    Until Cond_2;
  ...
  Until Cond_1

```

Ing Del y Gil

```

Program Num_No_Cero;
Var
  Positivo : Boolean;
  Num       : Integer;
{*****Programa que ingresa Números hasta que
se ingresa un cero*****}
Begin
  Cero := False;
  While (not Cero) Do {while (Cero=False) do}
  Begin
    Write('Ingrese Número:');
    Readln(Num);
    If Num =0 Then Cero := True;
    {Cero := Num = 0;}
  End.
  {Realizar el mismo ejercicio con la estructura
  Repeat-Until}

```

Ing Del y Gil

BUCLES WHILE/REPEAT ANIDADOS

```

While Cond_1 do
  Begin
    ...
    Repeat
      ...
      Until Cond_2;
    ...
  End; {Cond_1}
Repeat
  .
  .
  While Cond_2 do
  Begin
    ...
    End;
  {Cond_2}
  ...
  Until Cond_1;

```

Ing Del y Gil

ESTRUCTURAS REPETITIVAS

• La sentencia FOR-TO-DO

```
For V: = Vi to VF do
  Begin
    instrucción 1;
    instrucción 2;
    ...
    instrucción N;
  End;
```

Ing Del y Gi I

```
Program Ejem_For;
Uses          Var
  Crt;          Anno   : 1983..1989;
                Salario : Real;

Begin
  For Anno: = 1983 to 1989 do
    Begin
      Write('Ingrese Salario: ');
      Readln(Salario);
      Write('En el año ', anno:5);
      Write('he ganado: ', Salario :8:2);
      writeln;
    End {End For}
  End.
```

Ing Del y Gi I

ESTRUCTURAS REPETITIVAS

• La sentencia FOR-DOWNTO-DO

```
For V: = VF DownTo Vi do
  Begin
    instrucción 1;
    instrucción 2;
    ...
  End;
```

Ing Del y Gi I

BUCLES FOR ANIDADOS

```
For i: = 1 to 10 do
  For j: = 1 to 5 do
    Begin
      ...
    End;
  End;

For i: = 1 to 10 Do
  Begin
    For j: = 1 to 3 do
      Begin
        ...
      End;
      Readln;
    ...
  End;
```

Ing Del y Gi I

```
Program Sumar;
{Programa que suma 5 números}
Const
  Maximo = 5;
Var
  Numero, Suma : Integer;
  Promedio : real;

Begin
  Suma:=0;
  For i:= 1 to Maximo do
    Begin
      Write(' Ingrese Numero: '); Read(Numero);
      Suma := Suma + Numero
    End;
  Write('La suma es : ', Suma);
  Promedio := Suma/ Maximo; {FUERA DEL CICLO}
  Write('Promedio: ';Promedio:0:2);
  Readkey;
End.
```

Ing Del y Gil

Ejm.:

Leer y visualizar 10 números no negativos. Si encuentra un número negativo antes de haber leído los 10 números se termina el bucle.

```
Begin
  For i:= 1 to 10 do
    Begin
      Readln(N);
      If n<0 Then
        break;
    End {End For}
End.
```

Ing Del y Gil

SENTENCIAS DE CONTROL INCONDICIONAL

Pascal 7.0 incluye 4 procedimientos - BREAK, CONTINUE, EXIT, HALT – que permiten bifurcar a determinadas zonas de un programa de un modo incondicional. Break y Continue son exclusivos de la versión 7.0 mientras que Exit y Halt ya existían en versiones anteriores. Las sentencias Break y Continue deben aparecer en el interior de un bucle (for, while o repeat) y permiten terminar el bucle mientras que Exit y Halt pueden aparecer en cualquier parte de un programa

Ing Del y Gil

Ejm.:

```
Var
  T,total: integer;
  ....
Begin
  For i:= 1 to 100 do
    Begin
      Readln(T);
      If T=0 Then Continue;
      Total:= total +1; {Inc(t)}
      Writeln(La suma total es: ',total);
    End {End For}
End.
```

Ing Del y Gil

```
Program Guia1_16;
Uses
  Crt;
Var
  A,B,C,mayor      : integer;

Begin
  ClrScr;
  write('Ingrese los valores');
  write('Primer número: ':20);
  readln(A);
  write('Segundo número: ':20);
  readln(B);
  write('Tercer número: ':20);
  readln(C);
  writeln;
  If A>B Then
    if A>C Then
      mayor := A
    else
      mayor:= C
  Else
    If B>C Then
      mayor:= B
    else
      mayor:= C;
  writeln('El número mayor es: ',mayor);
  readkey
end.
```

```
Program Guia1_30;
Uses
  Crt;
Var
  i           :
  x,y         : Integer;
  resultado   :
  p           :

Begin
  ClrScr;
  writeln;
  Write('Ingrese valor x:':20);
  repeat
    {$I-}
    readln(X);
    resultado := IoResult
    {$I+}
  until (resultado=0);
  Write('Ingrese valor y:':20);
  repeat
    {$I-}
    readln(Y);
    resultado := IoResult
    {$I+}
  until (resultado=0);
  p:= 1;
  For i:= 1 to abs(y) do
    Begin
      p:= p*x;
    End;
  if y<0 Then
    p:= 1/p;    {Omite begin-End ya que hay 1 instrucción}
  writeln;
  write('Resultado: ',p:0:2);
  readkey;
end.
```

```
{ Guia 1_44 }
Program CEAT;
{ Realizado por: Michelanthony Nieto. C.I: 17.315.486
  Instituto Universitario de Tecnología de Valencia
  Materia: Lenguaje de Programación I
  Sección: 01B Turno Mañana
  Fecha: 16/03/04
  Prof: Dely Gil }
Uses
  CRT;
Var
  Porc_C, Porc_E, Porc_T           : Real;
  monto, mayor_E, mayor_O, mayor_Ch : Real;
  Zona, pago, codigo, forma      : Char;
  N, cont_E, Cont_O, Cont_N, Cont_S, cont_C : Word;
  cont_Ef, cont_T                 : Word;
  Nombre, Nombre_E, Nombre_O     : String[20];
  Resultado, l                    : Integer;
Begin
  Clrscr;
  Porc_C:=0; Porc_E:=0; Porc_T:=0; mayor_E:=0;
  mayor_O:=0; mayor_Ch:=0; cont_E:=0;
  Cont_O:=0; Cont_N:=0; Cont_S:=0;
  cont_C:=0; cont_E:=0; cont_T:=0; l:=1;
  Gotoxy (30,3); Write ('Bienvenidos a CEAL,C.A');
  Gotoxy (5,5); Write ('Cuántas ventas desea ingresar: ');
  Repeat
  { $I- }
  Gotoxy (36,5); ClrEol;
  Read (N);
  Resultado:=IOResult
  { $I+ }
  Until (resultado=0) and (n>0);
For I:=1 to N do
Begin
  ClrScr;
  Gotoxy (5,7); Write ('Ingrese monto de la venta: ');
  Repeat
  { $I- }
  Gotoxy (32,7); ClrEol;
  Read (Monto);
  Resultado_m:=IOResult
  { $I+ }
  Until (resultado=0) and (monto>0);
```

```
Gotoxy (5,9); Write ('Zona de la venta ');
Gotoxy (5,11); Write (('E)ste-(O)este-(N)orte-(S)ur: ');
Repeat
  Gotoxy (35,11); ClrEol;
  Read (Zona);
  Zona:=UpCase (Zona)
  Until (Zona in ['E','O','N','S']);
  Gotoxy (5,13); Write ('Forma de pago ');
  Gotoxy (5,15); Write (('C)heque-(E)fectivo-(T)arjeta de credito: ');
  Repeat
  Gotoxy (47,15); ClrEol;
  Read (Forma);
  Forma:=UpCase (Forma)
  Until (Forma in ['C','E','T']);
  Gotoxy (5,17); Write ('Codigo 1,2,3: ');
  Repeat
  Gotoxy (19,17); ClrEol;
  Read (Codigo);
  Codigo:=UpCase (Codigo)
  Until (Codigo in ['1','2','3']);
  Gotoxy (5,19); Write ('Ingrese el Nombre: ');
  Repeat
  Gotoxy (20,19); ClrEol;
  Read (Nombre);
  IF Nombre = '' then { If length(Nombre)=0 }
  Write ('Debe ingresar un nombre: ');
  Until (Nombre = '');
  Clrscr;
  { Total de ventas por zona }
  Case Zona of
  'E': Begin
    Inc (Cont_E);
    If Monto > Mayor_E then
    Begin
      Mayor_E:= Monto;
      Nombre_E:= Nombre
    End
  End;
  'O': Begin
    Inc (Cont_O);
    If Monto > Mayor_O then
    Begin
      Mayor_O:= Monto;
      Nombre_O:= Nombre
    End
  End
```

```
'N': Inc (Cont_N);
'S': Inc (Cont_S);
End; {Case}
{Total de pagos efectuados}
Case Pago of
'C': Begin
    Inc (cont_C);
    If monto > Mayor_Ch Then
        Mayor_Ch:= monto
    End;
'E': Inc (cont_Ef);
'T': Inc (cont_T);
End; {case}
End; {For}
Porc_C:= (cont_C * 100) /N;
Porc_E:= (cont_Ef * 100) /N;
Porc_T:= (cont_T * 100) /N;
{ Salida}
Clrscr;
Gotoxy (30,3);Write ('*****Resultados*****');
Gotoxy (5,5);Write ('Venta en la zona este: ', Cont_E);
Gotoxy (5,7);Write ('Venta en la zona oeste: ', Cont_O);
Gotoxy (5,9);Write ('Venta en la zona norte: ', Cont_N);
Gotoxy (5,11);Write ('Venta en la zon sur: ', Cont_S);
Gotoxy (5,13);
Write ('El vendedor con mayor venta en la zona este es: ',mayor_E);
Gotoxy (5,15);
Write ('El vendedor con mayor venta en la zona oeste es:
',mayor_O);
Gotoxy (5,17);
Write ('Porcentajes de pagos efectuados en cheque es:
',Porc_C:0:2);
Gotoxy (5,19);
Write ('Porcentajes de pagos efectuados en efectivo es: ',Porc_E:0:2);
Gotoxy (5,21);Write ('Porcentajes de pagos efetuados en tarjeta de
credito es: ',Porc_T:0:2);
Gotoxy (5,23);Write ('La mayor venta en cheque es:
',Mayor_Ch:0:2);
ReadKey;
end.
```

Taller: Realice el ejercicio con subprogramas, desarrolle los Módulos:

- | | |
|---------------------|------------------|
| ✓ Inicializacion | ✓ Validar_codigo |
| ✓ Validar_N | ✓ validar_nombre |
| ✓ Validar_monto | ✓ Calculo |
| ✓ validar_zona | ✓ Porcentaje |
| ✓ Validar_formaPago | ✓ Salida |

Desarrolle el programa principal con las llamadas de los subprogramas:
Program CEAL

{Definiciones de Variables globales}

{Definiciones de Subprogramas Inicializacion, alidar_n,.....,.....}

BEGIN {Programa Principal}