

Capítulo 2

1. Estructura de Control Selectiva (Alternativa o de Decisión)
2. Estructura de Control Selectiva Múltiple

1. La estructura de **control selectiva** nos permite evaluar una condición y decidir cuales sentencias a ejecutar, entre dos alternativas excluyentes.

```

if Condición then
    sentencias 1      {se ejecutan si la condición es verdadera}
[else
    sentencias 2]     {se ejecutan si la condición es falsa, puede estar vacía}
  
```

Después de ejecutar la alternativa que corresponde, sigue en secuencia con la sentencia que está a continuación de la estructura selectiva.

No va un **;** antes del **ELSE**, porque estaría indicando fin de la sentencia.

Nota: las sentencias 1 y 2 pueden ser simples o compuestas (encerradas entre begin - end)

Ej1 - Si quiero calcular $f(x) = 1/x$ ¿Qué sucede si ingresa un $x = 0$?

```

Program evalua;
Var
    X : real;
begin
  Writeln('ingrese un numero real');
  readln(X);
  if X <> 0 then
    writeln(1/X:8:2)
  else
    writeln('no se puede evaluar la función con argumento cero');
end.
  
```

Ej2 - Leer un precio, si supera los \$ 100 decrementar en un 15%. Escribir el precio resultante.

```

Program descuento;
Var
    Precio : real;
begin
  Writeln('ingrese el precio');
  readln(Precio);
  if Precio > 1000 then
    Precio := Precio*0.85;
  writeln('el valor decrementado es', Precio:6:2)
end.
  
```

{ **Aclaración : no se hacen especificaciones de medidas como mts., hs, \$, lts. etc**

Ej3-Ingresar dos números e informar la distancia entre ambos. Ej : 2,7 → 5 , -2, -8 → 6

```

Program distancia;
Var
    Nro1 , Nro2, Dist : integer;
  
```

<pre> Begin readln (Nro1 , Nro2); if Nro1 > Nro2 then Dist := Nro1 - Nro2 else Dist := Nro2 - Nro1; writeln (Dist); end.</pre>	<pre> Begin readln (Nro1 , Nro2); if Nro1 > Nro2 then writeln (Nro1 - Nro2) else writeln (Nro2 - Nro1) end.</pre>	<pre> Begin readln (Nro1 , Nro2); Dist := Nro1 - Nro2 if Dist < 0 then Dist := Dist * (-1); writeln (Dist) End.</pre>
--	---	---

Ej4 - Leer un carácter que representa el tipo de pasaje aéreo (P=Primera Clase o E=Clase Económica). Imprimir qué tipo de pasaje ingresó.

Program determina;

Var

Tipo : char;

Begin

Write ('Ingrese tipo de pasaje: ');

readln (Tipo);

if Tipo = 'P' then

writeln('Primera Clase')

else {cualquier otro carácter}

writeln('Clase Económica')

end.

Notar que cualquier carácter distinto de 'P' provoca la impresión de la palabra 'Clase Económica'. Para controlar que solo se escriba si se trata de un carácter 'E', debemos evaluar nuevamente el valor de Tipo

Es posible dentro de la alternativa verdadera (then) o falsa (else) volver a evaluar una condición y elegir nuevamente entre dos alternativas excluyentes. En este caso se trata de estructuras de decisión "anidadas"

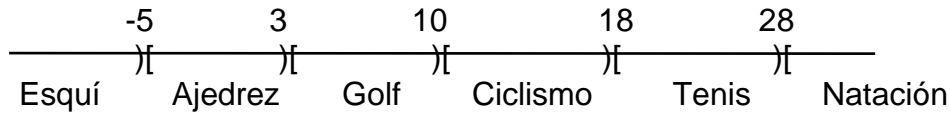
<pre> if Tipo = 'P' then writeln('Primera Clase') else {cualquier otro carácter} if Tipo = 'E' then writeln('Clase Economica') else writeln ('Carácter erróneo');</pre>	<pre> if (Tipo <> 'P') AND (Tipo <> 'E') then writeln('Caracter erróneo') else {(Tipo = 'P') o (Tipo = 'E')} if Tipo = 'P' then writeln('Primera Clase') else {Tipo = 'E'} writeln('Clase Económica');</pre>
---	--

El caracter ingresado puede ser mayúscula o minúscula, antes de comparar convertir a mayúscula utilizando la función UPCASE, que devuelve la mayúscula del carácter
 Tipo :=UPCASE (Tipo) { evitando comparar además (Tipo <> 'p') AND (Tipo <> 'e')}

Ej5 – Un complejo hotelero 'all-inclusive' aconseja a sus huéspedes distintas actividades dependiendo de la temperatura. Dada la siguiente tabla de temperaturas y deportes implementar un algoritmo que lea una temperatura y establezca el correspondiente deporte mostrando el nombre por pantalla

```

TEMPERATURA < -5° → esquí
-5° <= TEMPERATURA < 3° → ajedrez
3° <= TEMPERATURA < 10° → golf
10° <= TEMPERATURA < 18° → ciclismo
18° <= TEMPERATURA < 28° → tenis
28° <= TEMPERATURA → natación
```

Posibles soluciones:

- ✓ analizar en forma ascendente o descendente los intervalos
- ✓ partir a la mitad, considerando dos grupos, y analizar cada uno de ellos.

Partiendo de la declaración de la variable Temp entera, y ya leída
{se presentan las dos soluciones ¿cuál es más conveniente?}

```

.....
if Temp < -5 then
  writeln ('Esquí')
else { Temp >= -5}
  if Temp < 3 then
    writeln ('Ajedrez')
  else
    if Temp < 10 then
      writeln ('Golf')
    else
      if Temp < 18 then
        writeln ('Ciclismo')
      else
        if Temp < 28 then
          writeln ('Tenis')
        else
          writeln ('Natación');

```

```

.....
if Temp < 10 then
  if Temp < -5 then
    writeln ('Esquí')
  else
    if Temp < 3 then
      writeln ('Ajedrez')
    else
      writeln ('Golf')
else
  if Temp < 18 then
    writeln ('Ciclismo')
  else
    if Temp < 28 then
      writeln ('Tenis')
    else
      writeln ('Natación');

```

Otra solución posible sería evaluar en forma individual cada uno de los intervalos

```

if Temp < -5 then
  writeln ('Esquí');
if (-5<=Temp) and (Temp < 3) then
  writeln ('Ajedrez');
if (3<=Temp) and (Temp < 10) then
  writeln ('Golf');
if (10<=Temp) and (Temp < 18) then
  writeln ('Ciclismo');
if (18<=Temp) and (Temp < 28) then
  writeln ('Tenis');
if (28<=Temp) then
  writeln ('Natación');

```

Esta forma requiere mas tiempo de ejecución, ya que evalúa todas las opciones y en todas ellas hay que describir el intervalo completo, pues ninguna situación resulta por defecto. Cualquiera de ellas que resulte verdadera no evita la evaluación de las demás

De lo visto anteriormente se resume:

- N estructuras de decisión anidadas permiten **n+1** alternativas excluyentes
- N estructuras de decisión secuenciales permiten **n** alternativas

Ej6 - Se quiere calcular el precio que debe abonar para asegurar un automotor. Los parámetros que determinan el costo son:

- ☞ Tipo de vehículo : C- Comercial (\$200) ; P- Particular (\$ 100)
- ☞ Tipo de seguro : 1- Todo riesgo (+30%) ; 2- Básico (sin incremento)
- ☞ Accidentes en el período anterior : S ; N (– 5%)
- ☞ Edad del conductor (más de 65 años + 10%)

Se parte de un importe inicial (depende del tipo de vehículo) el cual se va incrementando y/o bonificando según los diferentes condicionamientos (tipo de seguro, accidentes, edad).

Program seguro;

Var

Importe : real;

TipoVehic, Accid : char;

TipoSeg, Edad : byte;

Begin

Write('ingrese tipo de vehiculo: C-comercial; P-particular'); readln (TipoVehic);

Write('tuvo accidentes en el periodo anterior: S/N'); readln (Accid);

Write('ingrese tipo de seguro: 1-todo riesgo; 2-basico'); readln (TipoSeg);

Write('ingrese edad'); readln (Edad);

if TipoVehic = 'C' then

Importe :=200

else

Importe :=100;

if TipoSeg = 1 then

Importe := Importe*1.3;

if Accid = 'N' then

Importe := Importe* 0.95;

if Edad >65 then

Importe := Importe*1.1;

Writeln ('el valor del seguro es', Importe);

End.

Supongamos que queremos considerar un incremento del 8% en edades entre 18 y 20

- Si es > de 65 años incremento 10 %

- Si es < de 21 años incremento 8 %

Es necesario controlar tres posibilidades diferentes y totalmente excluyentes, con incremento del 10%, del 8% y sin incremento. Encajamos o anidamos las estructuras selectivas

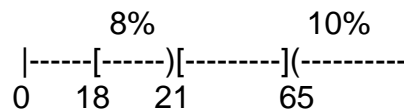
if Edad > 65 then

Importe := Importe * 1.1

else

if Edad < 21 then

Importe := Importe * 1.08;



Si se cumple Edad > 65 se incrementa el importe en un 10 % y no se evalúa Edad < 21, en cambio si lo planteamos secuencialmente, quedaría:

if Edad > 65 then

Importe := Importe * 1.1;

if Edad < 21 then

Importe := Importe * 1.08;

Si se cumple $Edad > 65$, después de incrementar el 10 %, igual se evalúa $Edad < 21$, que en dicho caso sería falso. Por lo tanto, conviene encajar o anidar las estructuras selectivas ya que las dos alternativas planteadas son excluyentes y se harían evaluaciones de más

Es importante destacar que no todas las condiciones son dependientes, por ejemplo el tipo de vehículo no incide sobre el tipo de seguro. Estas relaciones deben ser analizadas para determinar si la estructura alternativa es independiente una de otra, o conviene anidarla.

Observaciones:

a. Si al anidar estructuras selectivas, la estructura interna tiene la alternativa falsa vacía como se muestra a continuación:

<pre> If cond1 then If cond2 then Sentencia a; Else Sentencia b; </pre>	<u>Se interpreta</u> <pre> If cond1 then If cond2 then Sentencia a; Else Sentencia b; </pre>
---	--

Debe indicarse expresamente que la alternativa **else** pertenece a la estructura externa, delimitando con **Begin-End** la estructura interna, de lo contrario se asocia al **if** más próximo.

<u>Debe expresarse</u> <pre> If cond1 then begin if cond2 then Sentencia a; end Else Sentencia b; </pre>
--

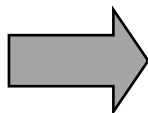
b. Evitar el abuso de IF cuando se usa variables booleanas.

Ejemplo: Sea *Cumple* una variable booleana, X e Y variables del mismo tipo y se desea almacenar True en *Cumple* si ambas almacenen el mismo valor y False en caso contrario. Es más simple sustituir la estructura de decisión, por la asignación a la variable booleana del resultado de la relación que se desea verificar.

```

If X = Y then
  Cumple := True;
else
  Cumple := False;

```



```
Cumple := X = Y;
```

2. Estructura de decisión Generalizada (o Múltiple)

Cuando en una estructura selectiva anidada, las distintas alternativas dependen del valor resultante de una expresión de tipo ordinal, se puede utilizar la estructura Case

```

Case  expresión of
    Listas de constantes 1 : Sentencias1;
    Listas de constantes 2 : Sentencias2;
    :
    :
    Listas de constantes n : Sentenciasn;
[ else
    SentenciasK; ]
End;

```

El valor que resulta de la expresión es buscado en las listas de constantes, ejecutándose las respectivas sentencias. En caso de no encontrarlo se ejecutan las SentenciasK correspondientes a la alternativa "else". Esta alternativa puede estar vacía (opcional, se expresa entre corchetes)

Ej7 - Leer el numero de mes (1..12) e informar el trimestre correspondiente en el año

Program Trimestres;

Var

Mes :byte;

Begin

Writeln('ingrese el mes');

Readln(Mes);

If Mes <= 3 then

Writeln ('1er Trimestre')

else

If Mes <= 6 then

Writeln ('2do Trimestre')

else

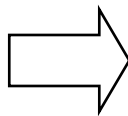
If Mes <= 9 then

Writeln ('3er Trimestre')

else

Writeln ('4to Trimestre');

End.



Case Mes of

1..3 : Escribir ('1er Trimestre');

4..6: Escribir ('2do Trimestre');

7..9: Escribir ('3er Trimestre');

10..12: Escribir ('4to Trimestre');

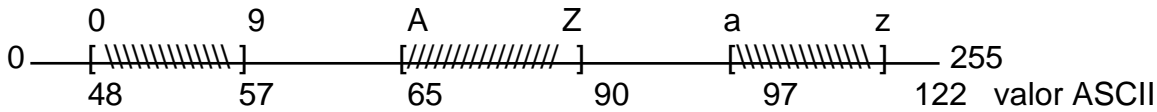
end;

- ✓ El tipo de la expresión debe ser ordinal (entero, carácter, booleano, subrango (*))
- ✓ Se evalúa el resultado de la expresión y se busca dicho valor en las listas de constantes, si se encuentra se ejecutan las sentencias asociadas. De no encontrarse se ejecuta la opción else o sea la sentenciasK. En ambos casos continua la ejecución en la sentencia que sigue al End
- ✓ Las alternativas son totalmente excluyentes, por lo tanto el mismo valor constante no puede estar en más de una lista.

(*)Subrango: solo para ordinales, por ejemplo : 0..9 ó 'A'..'Z'

Ej8 - Ingresar un caracter e imprimir si es

{	Numero {09}✓
	Letra {A....Z, a.....z}
	Separador {, ; . b }
	otro { #, *, \$,/ }



Program Caracteres;

Var

Car : char;

Begin

Writeln('ingrese un caracter');

Readln (Car);

If ('0'<=CAR) AND (CAR<= '9') then

writeln('Número')

else

If ('A' <= Car) AND (Car <= 'Z') OR ('a' <= Car) AND (Car <= 'z') then

Writeln('Letra')

else

If(Car= ',') OR (Car = ';') OR (Car = ' ') then

Writeln('Separador')

else

Writeln('Otro');

End.

Utilizando la estructura CASE

.....

Writeln('ingrese un caracter');

Readln(Car)

Case Car of

'0'..'9' : Writeln ('número');

'A'..'Z','a'..'z': Writeln ('Letra');

',' , ';' , ' ' : Writeln ('Separador');

else

Writeln ('otros');

End;

Ej9 - Leer Dia y Mes, informar a que estación pertenece la fecha leída

Los meses que íntegramente caen dentro de una estación, se agrupan.

La dificultad se presenta en aquellos meses que abarcan dos estaciones distintas, para ello se debe analizar el día de la fecha ingresada. Por lo tanto, estos casos se resuelven aparte.

Program Estaciones;

Var

Dia, Mes : byte;

Begin

Writeln('ingrese día y mes');

Readln (Dia , Mes);

```

Case Mes of
  1 , 2 : Writeln('verano');
  4 , 5 : Writeln ('otoño');
  7 , 8 : Writeln ('invierno');
  10 , 11 : Writeln ( 'primavera');
  3 : If DIA <= 20 then
    Writeln ('verano')
  else
    Writeln ('otoño');
  6 : If DIA <= 20 then
    Writeln ('otoño')
  else
    Writeln ('invierno');
  9 : If DIA <= 20 then
    Writeln ('invierno')
  else
    Writeln ('primavera');
  12 : If DIA <= 20 then
    Writeln ('primavera')
  else
    Writeln ('verano');
End;
End.

```

Ej10 - Leer dos números N1 y N2 (pueden tomar valor cero o uno). Informar si ambos son 1 ó ambos son 0 ,ó son distintos.

```

Program Numeros;
Var
  N1, N2 : 0..1;  {precondición : solo toma solo dos valores 0 o 1}
Begin
  Writeln('ingrese dos números');
  Readln (N1 , N2);
  Case N1 + N2 of {el selector del SEGUN pude ser sea una suma}
    0 : Writeln('ambos son cero');
    1 : Writeln ('son distintos');
    2 : Writeln ('ambos son uno');
  End;
End.

```


Ej11 - Leer Dia, Mes y Anio, informar si la fecha es correcta

Program FechaOk;

Var

 Dia , Mes , CantDias : byte;

 Anio: word;

Begin

 Writeln('ingrese día, mes y año');

 Readln (Dia , Mes , Anio);

 If (1<= Mes) and (Mes <= 12) Then

 Begin

 If (Mes =4) or (Mes =6) or (Mes= 9) or (Mes=11) Then

 CantDias := 30

 else

 If Mes = 2 then

 If Anio mod 4 = 0 Then

 CantDias := 29

 else

 CantDias := 28

 else

 CantDias :=31

 If (Dia >= 1) and (Dia <= CantDias) Then

 Writeln('fecha correcta')

 else

 Writeln('día erróneo, fecha incorrecta')

 End

else

 Writeln('mes erróneo, fecha incorrecta');

End.

Utilizando la estructura CASE , se reemplazaría el código del recuadro por

Case Mes of

 4 , 6 , 9 , 11 : CantDias :=30

 2 : if Anio mod 4 = 0 then

 CantDias := 29

 else

 CantDias :=28

 else

 CantDias := 31

End;